HydroWeb

ИНЖЕНЕРНАЯ ПЛАТФОРМА

РУКОВОДСТВО РАЗРАБОТЧИКА

Архитектура · расчётное ядро · МКЭ · API · тестирование · деплой

Учебно-методическое и техническое пособие

Версия платформы 1.1.0-alpha · 2026

Двухуровневая проверка ядра: сравнение с замкнутыми решениями статики на семи контрольных задачах Блюма и сравнение с эталонной программой

Аннотация

Настоящее пособие — техническое руководство разработчика платформы HydroWeb. Оно описывает архитектуру системы, расчётное ядро (два движка — Блюм и МКЭ), полную теорию метода конечных элементов, блок контрольных задач, программный интерфейс, стратегию тестирования, релиз и технический долг.

Документ предназначен для инженеров-программистов, сопровождающих и развивающих платформу, а также для внешних экспертов и научных партнёров, проверяющих корректность расчётного ядра. Центральная глава 6 приводит выведенные формулы, матрицы конечного элемента и результаты сверки с замкнутыми аналитическими решениями.

> **Структура документа**
> Главы 1–4 — архитектура, установка, структура, конвенции (быстрый вход в проект).
> Главы 5–7 — расчётное ядро, полная теория МКЭ и блок контрольных задач (ядро продукта).
> Главы 8–11 — API, тестирование, деплой, технический долг.

Оглавление

Аннотация ... 2
Глава 1. Архитектура проекта ... 6
Стек ... 6
Глава 2. Установка для разработки ... 7
Переменные окружения ... 7
Глава 3. Структура каталогов ... 10
Схема БД: что добавил релиз ... 14
Глава 4. Конвенции кода ... 20
Python ... 20
TypeScript ... 20
Коммиты и журнал ... 20
Размещение файлов и именование (порядок в проекте с 05.10.2026) ... 20
Глава 5. Расчётное ядро: два движка ... 22
5.1. Теория давления грунта ... 22
Принцип эффективных напряжений (Терцаги) ... 22
Активное и пассивное давление: предельное равновесие ... 22
Давление покоя (Jáky / K0) и метод Блюма ... 23
Сейсмика (Мононобе-Окабе) ... 23
5.2. Реализация ядра Блюма (core/) ... 24
coefficients.py — коэффициенты давления ... 24
soil_pressure.py — построение эпюры ... 24
Полосовая пригрузка: бытовое давление и глубина действия ... 24
Коэффициенты запаса ... 26
embedment.py — глубина заделки (бисекция) ... 26
internal_forces.py и steel_check.py ... 27
Глава 6. МКЭ-ядро: балка на упругопластическом основании ... 28
6.1. Физическая модель ... 28
6.2. Определяющие уравнения ... 29
6.3. Матрицы конечного элемента ... 29
6.4. Аналитическая верификация Hetényi ... 29
6.5. Порядки сходимости и суперсходимость ... 30
6.6. Метод начальных напряжений и стадийность ... 30
6.7. Ключевые источники ... 31
6.8. Реализация: сборка и решение (beam_solver.py) ... 31
6.9. Реализация: двусторонний закон p–y (SoilSpringLaw) ... 31
6.10. Реализация: метод начальных напряжений (Пикар) ... 32
6.11. Реализация: стадийность и коэффициенты запаса (staging.py) ... 32
6.12. Ускоренная сборка K и кэш ([0293-C1]) ... 36
6.13. Прямой Ньютон и детектор механизма ([0293-C2]) ... 37
Переключение на Ньютон по умолчанию: критерии и результат (05.10.2026, [0296-3], [0297]) ... 38
Причина сдвига: страховочная пружина Пикара ([0297], замер невязки) ... 39
Критерий (a), переформулированный ([0297]): режим ratchet_check.py --solver-switch ... 39
Что меняется для пользователя ... 40
Чистка 1.1: ручки и мёртвый код ([0377-чистка], 15.10.2026) ... 41
6.14. Управление долгим счётом (run_control) и оценка стоимости ([0293-C1, C3]) ... 41
6.15. Режимы ядра по желанию (opt-in, [0293-C5, C7, C9]) ... 42

Глава 7. Блок контрольных задач....45
Запуск....45
Допуски....45
Пользовательский модуль blum_benchmarks (Б-1…Б-7)....45
Точное интегрирование усилий Блюма ([0551-блюм-сетка-и-q-internal], [0551-блюм-узлы-сетки-показа], 29.10.2026)....47
Глава 8. API....50
POST /api/calculate/fem (Этап 5)....57
Поля запроса, возвращающие прежний расчёт ([0376]–[0393], 15.10.2026)....62
Строгий вход: отказ 422 вместо тихой потери ([0412-строгий-вход], 20–21.10.2026)....63
POST /api/calculate/fem: статус, дедлайн, семафор, задачи ([0293-C3])....66
POST /api/calculate/fem: новые входы и выходы....68
Проверки вне МКЭ-расчёта: /api/checks/*....68
Узлы крепления: POST /api/joints/check ([0462-узлы-пилот])....73
Кабинет поставщика и каталоги поставщиков ([0461-кабинет-поставщика])....74
Регистрация по ролям и заявки поставщиков ([0485-регистрация-ролей])....78
Правовые документы: редакция 2026-10-29 и журнал согласий ([0484-правовые-тексты-цены], [0499-правовые-тексты-сводка-поставщика], [0520-обращения-с-вложением-и-редакция-2026-10-27], [0531-обращения-до-пяти-вложений], [0552-правовая-редакция-2026-10-29])....80
Импорт файлов проектов в продукт: POST /api/projects/import-file ([0459-импорт-проектов-в-продукт])....83
Запрос цены поставщикам: /api/price-requests/* ([0478-запрос-цены])....84
Письма по событиям кабинета поставщика ([0562-письма-кабинета-поставщика], [0563-письма-политика-тексты-книги])....87
Продукция поставщиков в проектах: сводка спроса ([0498-продукция-поставщика-в-проектах])....88
Сравнение вариантов проекта: поле variants и ключ utilization_wall_metal ([0475-сравнение-вариантов])....90
Откуда это число: реестр оснований ([0476-почему-это-число])....92
Готовность к экспертизе ([0477-готовность-к-экспертизе])....93
Размеры котлована в плане: поле проекта planSize ([0490-доработки-интерфейса])....95
Реквизиты проекта и карточка тома ТЭО в проекте: поля projectInfo и teoCard ([0504-тэо-по-ревизии])....96
Сквозной сборщик запроса и сторож «поле проекта → запрос» ([0293-C4])....97
Клиент: статус, ход, отмена, проверки ([0293-C3, C5b, C6, C7])....98
Сессия и безопасность....99
ИИ-ассистент: API, база знаний и источники....101
ИИ-ассистент: суточный лимит и /api/assistant/quota....102
База знаний: очистка от материалов третьих лиц....103
ИИ-ассистент: клиентская часть....105
Экспорт и импорт проектов (projectExport.ts)....106
Администрирование пользователей (admin API)....106
Администрирование: журнал, принудительный выход, приглашение, состояние....107
Кабинет пользователя (routers/profile.py)....109
Пробный доступ по метке ссылки и обращения «Написать нам» ([0517-пробный-доступ-по-метке], [0520-обращения-с-вложением-и-редакция-2026-10-27], [0531-обращения-до-пяти-вложений])....110
Журнал событий беты ([0543-журнал-событий-беты], 29.10.2026)....114
Уровни доступа, шаг 1 ([0548-уровни-доступа-шаг-1], 29.10.2026)....116

Уровни доступа, шаг 2а ([0570-уровни-доступа-шаг-2а], 30.10.2026)....118
Интерфейс редактора: границы ошибок, вкладки, вердикт, шаблон проекта....121
Сметный движок: модуль hydroweb/estimate и /api/estimate/*....127
Снимок цен ФГИС ЦС и книга норм: формат и пересборка....131
Привязка шпунтового каталога к КСР....133
Площадь сечения А каталога шпунта: на 1 м стенки ([0325-каталог-А])....134
Сортамент труб для распорок и стоек РРСК....135
Сортамент двутавров и швеллеров для распорок, поясов и стоек РРСК....137
Выписки сортаментов и норм, у которых нет потребителя в продукте ([0338-каталоги-b-0708])....138
Проверка распорки по СП 16: коэффициент φ (core/strut_check.py, calc/strutCheck.ts)....139
Клиент: смета (РИМ) на вкладке «Документы»....140
Клиент: том ТЭО (DOCX)....143
Оформление выгружаемых документов (ГОСТ Р 21.101-2026)....145
Клиент: отчёт МКЭ и его приложения ([0516-отчёт-мкэ-приложения], 27.10.2026)....164
Клиент: подраздел «Анкерное крепление» записки, спецификация и отчёт МКЭ ([0529-анкеры-в-записке-и-выгрузках], 28.10.2026)....166
Подбор решений: клиентская часть....168
Вкладка «Чертёж»: панорамирование и масштаб....172
PDF чертежа: векторный вывод с текстовым слоем ([0526-pdf-чертежа-с-текстовым-слоем])....173
Глава 9. Тестирование....175
Двухуровневая проверка ядра....180
Храповик сверки (scripts/gw_compare/ratchet_check.py)....180
Три набора эталонов и инструменты сверки (05.10.2026, [0296])....182
Аудит стиля кода ([0487-чистка-кода], 24.10.2026)....186
Непрерывная интеграция на push и pull request ([0481-ci-на-push])....186
Глава 10. Релиз и деплой....189
Эксплуатация: копия тома, проверка восстановления, монитор прода, аудит зависимостей....191
Вход под нагрузкой: блокировка БД и её устранение ([0343])....194
Сборка и публикация руководств....197
Глава 11. Технический долг....199
Воды, узлы, кабинет поставщика, импорт (24.10.2026)....199
Линии 24.10.2026 на интеграции....199
Безопасность (аудит [0020])....199
Архитектура....199
Качество и процессы....200
Документы и смета....200
Расчётное ядро (МКЭ)....201

Глава 1. Архитектура проекта

Монорепозиторий: Python-бэкенд (расчётное ядро + FastAPI) и React-фронтенд (SPA), деплой в Docker за nginx.

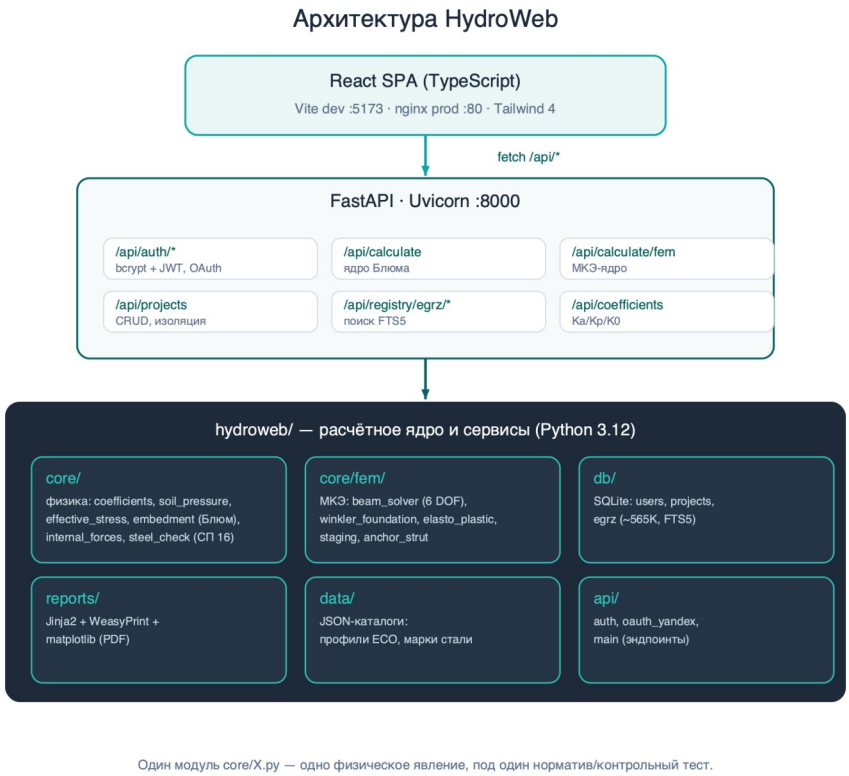


Рис. 1. *Архитектура HydroWeb: React SPA → FastAPI → расчётное ядро hydroweb/ (реальная структура проекта).*

Стек

Backend: Python 3.12, FastAPI 0.115+ / Uvicorn, Pydantic 2, numpy/scipy, SQLite + aiosqlite, bcrypt + PyJWT (JWT), openpyxl (импорт ЕГРЗ).

Frontend: React 19 + Vite, TypeScript, Tailwind 4; docx, exceljs 4.4.0 (MIT; XLSX со стилями ячеек, только динамический импорт), jspdf/xlsx (генерация документов на клиенте; jspdf — в том числе векторный PDF чертежа с текстовым слоем, [0526-pdf-чертежа-с-текстовым-слоем]). Состояние — в useState-хуках (без Redux/Zustand сознательно).

> **Принцип ядра**
>
> Один модуль core/X.py — одно физическое явление, под один норматив или контрольный тест.
>
> Все расчёты выполняются только на сервере; дублирование расчётов на фронте запрещено (глава 11).

Глава 2. Установка для разработки

Рабочая ОС — macOS (zsh) или Linux. Системные зависимости (Homebrew):

```
brew install node python@3.12
brew install pango cairo          # только для extra [tools] (WeasyPrint в скриптах
проверки выгрузок)
```

Python-окружение и зависимости:

```
cd ~/Projects/HydroWeb
python3.12 -m venv .venv && source .venv/bin/activate
pip install -c requirements.lock -e ".[dev,ingest]"   # сервер и тесты; скрипты
книг и проверки выгрузок — дополнительно ".[tools]" (matplotlib, WeasyPrint)
cd frontend && npm install && cd ..
```

Локальный запуск (два терминала, в обоих активирован venv):

```
uvicorn hydroweb.api.main:app --reload --port 8000
cd frontend && npm run dev           # Vite :5173, проксирует /api → :8000
```

Переменные окружения

Необязательны для разработки; обязательны для production:

- `HYDROWEB_JWT_SECRET` — секрет JWT (без него — эфемерный random на запуск);
- `HYDROWEB_CORS_ORIGINS` — whitelist origin'ов (CORS), НЕ «*»;
- `ADMIN_EMAILS` — список email, получающих роль admin при регистрации;
- Yandex OAuth — client id/secret и redirect URI (см. `api/oauth_yandex.py`); значения по умолчанию у YANDEX_CLIENT_ID нет — вход включён только при заданных ID и секрете. VK ID — в разработке: код готов и выключен (VK_CLIENT_ID пуст; см. `api/oauth_vk.py`), на экране входа кнопка недоступна и подписана «в разработке».
- `HYDROWEB_ASSISTANT_DAILY_LIMIT_STANDARD` — суточный лимит вопросов к ассистенту для роли standard (по умолчанию 3; 0 или отрицательное — ассистент для роли закрыт; нечисло — умолчание с предупреждением в журнале);
- `HYDROWEB_DEPLOYED_COMMIT` / `DEPLOYED_COMMIT` — штамп выкатки для вкладки «Состояние» админки (по очереди: окружение, /app/DEPLOYED_COMMIT, корень проекта; без них UI пишет «коммит не известен»).
- `HYDROWEB_MAX_CONCURRENT_CALCS` — число одновременных МКЭ-расчётов, общее для воркеров (файловые блокировки flock; по умолчанию 2, пределы 1…64); нет места — 429 с Retry-After: 10. Семафор общий на МАШИНУ: параллельные прогоны сверки и тестов получают 429 — для скриптов сверки ставить HYDROWEB_MAX_CONCURRENT_CALCS=8;
- `HYDROWEB_CALC_DEADLINE_S` — дедлайн синхронного /api/calculate/fem по умолчанию, с (240, пределы 5…600; меньше 300 с у nginx, чтобы ответ «timeout» успел дойти; поднимать на медленных машинах и в CI);
- `HYDROWEB_CALC_SLOT_DIR` — каталог файлов-«мест» семафора; по умолчанию зависит от пути к коду, поэтому независимые копии приложения на одной машине (рабочие деревья, параллельные тесты) не отнимают друг у друга места;

- HYDROWEB_MAX_ACTIVE_JOBS (20) и HYDROWEB_MAX_ACTIVE_JOBS_PER_OWNER (3) — лимиты активных фоновых задач расчёта всего и на владельца (пользователь либо IP); сверх них — 429;
- HYDROWEB_LOGIN_RATE_LIMIT / HYDROWEB_LOGIN_RATE_WINDOW_S (10 / 300), HYDROWEB_REGISTER_RATE_LIMIT / …_WINDOW_S (5 / 600), HYDROWEB_CHANGE_PASSWORD_RATE_LIMIT / …_WINDOW_S (10 / 600) — лимиты входа, регистрации и смены пароля: попыток с одного адреса на один процесс за окно, с. Не число или вне 1…100000 (окно 1…86400) — прежнее значение и предупреждение в журнале. Повышение ослабляет защиту от перебора паролей — только на время (например, занятие класса за одним NAT).
- HYDROWEB_EVENTS_EXCLUDE — адреса и домены почты команды через запятую (по умолчанию пусто): их учётные записи не идут в критерии выхода из беты вкладки «События» ([0543-журнал-событий-беты]), в таблицах журнала остаются; строка в .env.example, проброс в docker-compose.yml;
- HYDROWEB_LEVELS_ENFORCED — выключатель проверок по уровню доступа ([0570-уровни-доступа-шаг-2а], глава 8): «1» включает проверки действий по уровню (ассистент, МКЭ профессионального режима, число активных проектов) и требование входа для расчёта; пусто, «0» и любое иное значение — выключено (по умолчанию), нестандартное значение пишет предупреждение в журнал; выключено — ответы прежние. Читается один раз при запуске процесса: смена — перезапуск контейнера. Пробрасывается docker-compose.yml в форме ${HYDROWEB_LEVELS_ENFORCED:-0}, строка (закомментированная) — в .env.example;
- HYDROWEB_SWEEP_LOGIN / HYDROWEB_SWEEP_PASSWORD — логин и пароль тестовой записи для сверки (scripts/gw_compare/prod_sweep.py) и смоука (deploy/smoke_check.py). Нужны только при включённом выключателе (расчёт требует вход); пока он выключен, не нужны. В репозитории их нет, docker-compose.yml их в контейнер не пробрасывает: для смоука изнутри контейнера они передаются ключами docker exec -i -e HYDROWEB_SWEEP_LOGIN -e HYDROWEB_SWEEP_PASSWORD (подробно — глава 8, «Уровни доступа, шаг 2а»);
- HYDROWEB_TOKEN_EXPIRE_HOURS — срок сессии, ч: 336 (14 суток, было 720), скользящий (глава 8). В контейнер не пробрасывается: для прода строку добавляют в environment файла docker-compose.yml;
- HYDROWEB_LOGIN_LOCK_FAILURES (20), HYDROWEB_LOGIN_LOCK_WINDOW_S (900) и HYDROWEB_LOGIN_LOCK_S (900) — блокировка входа по учётной записи: столько неудач за окно (с) блокируют вход на срок (с); пределы 1…1000 и 1…86400, иначе умолчание и предупреждение в журнале; счётчики в базе, общие для процессов; пробрасываются docker-compose.yml;
- В контейнер переменные нагрузки и лимитов (HYDROWEB_MAX_CONCURRENT_CALCS, HYDROWEB_MAX_ACTIVE_JOBS, …_PER_OWNER, HYDROWEB_ASSISTANT_DAILY_LIMIT_STANDARD, лимиты входа, HYDROWEB_LEVELS_ENFORCED) пробрасывает docker-compose.yml в форме ${VAR:-умолчание кода} ([0343]); до этого строки в .env до приложения не доходили. Новые переменные попадают в контейнер только при его пересоздании (docker compose up -d --build в шаге 4 deploy.sh), а не при restart; если в /opt/hydroweb/.env такие строки уже записаны, после пересоздания контейнера они начнут действовать — перед выкаткой это проверяют по .env (раздел «Вход под нагрузкой» главы 10).

- `PYTHON_IMAGE` / `PYTHON_IMAGE_FALLBACK` — относятся к сборке образа, а не к работе приложения: база образа (Dockerfile: ARG PYTHON_IMAGE, по умолчанию python:3.12-slim; docker-compose.yml берёт значение из окружения или из .env рядом с ним) и образ-зеркало запасного пути deploy/deploy.sh при лимите Docker Hub (пустое значение отключает запасной путь; глава 10).
- `HYDROWEB_COST_MACHINE_FACTOR` — множитель скорости машины для оценки длительности расчёта (1 — машина калибровки cost_estimate).
- `HYDROWEB_FGIS_SNAPSHOT` — путь к снимку цен ФГИС ЦС (.json или .json.gz) для сметного движка; по умолчанию — новейший hydroweb/data/fgis/snapshot_*.json.gz. Нет файла — `/api/estimate/zones` отвечает available: false, остальные эндпоинты сметы — 503.
- `HYDROWEB_GESN_BOOK` — путь к книге норм ГЭСН-2022 (по умолчанию новейший hydroweb/data/fgis/gesn2022_*.json.gz); либо `HYDROWEB_FSNB2022_ZIP` (архив открытых данных ФСНБ-2022) и необязательный `HYDROWEB_TECHGROUPS_XML` (файл «База технологических групп»).

> **Известные грабли macOS**
>
> OSError: cannot load library 'libgobject-2.0-0' → не установлены pango/cairo (нужны WeasyPrint в скриптах проверки выгрузок, extra [tools]).
>
> Ошибки C-расширений при pip install → xcode-select --install.

Глава 3. Структура каталогов

Путь	Содержимое
`hydroweb/core/`	Расчётные модули: coefficients (Ka/Kp/K0), soil_pressure (эпюры σa/σp), effective_stress (σ'v), embedment (заделка по Блюму), internal_forces (M/Q), deflection, steel_check (СП 16), steel_grades, profiles, profile_selection, corrosion, dewatering, staging, water_*, surcharge, models, units.
`hydroweb/core/fem/`	МКЭ-ядро: beam_solver (балка 6 DOF), winkler_foundation, elasto_plastic (p–y, решатели), anchor_strut, nodal_elements, staging — см. главу 6.
`hydroweb/api/`	main (эндпоинты), auth (JWT/bcrypt, delete_account, эпоха сессий, revoke_user_sessions), login_guard (блокировка входа по учётной записи), session_renewal (скользящее продление сессии), assistant (ИИ-ассистент, суточный лимит), admin_audit (record_admin_action, журнал действий администратора), email_send, email_templates (шаблоны писем по событиям кабинета поставщика: тема и тело), supplier_notify (фоновые задачи отправки этих писем), oauth_common (state/PKCE, личности), oauth_yandex, oauth_vk.
`hydroweb/api/routers/`	Роутеры, вынесенные из main.py: profile.py (кабинет пользователя: /api/profile*, /api/auth/change-password) и admin_extra.py (журнал, принудительный выход, приглашение, состояние сервера, приём документов приглашённым) — см. главу 8.
`hydroweb/db/`	connection (схема SQLite; таблицы и столбцы, добавленные релизом, — ниже), egrz_repo (FTS5-поиск).
`hydroweb/core/ (поток ядра [0293])`	hydraulic_stability (взмыв, выпор, градиент), slope_stability (общая устойчивость: Феллениус, упрощённый Бишоп), anchor_capacity (несущая способность анкеров, СП 381 п. 7.3, прил. Л), load_combinations (нагрузочные сочетания), structure_class (допустимое значение [Kз] запаса в заделке по классу сооружения — ориентир, не норматив; те же четыре числа служат γn в [k_st] общей устойчивости); в water_pressure — слоистое поровое давление (WaterBearingLayer, StagedAquifer, LayeredPorePressure) и открытый столб воды.
`hydroweb/core/fem/ (дополнение)`	run_control (дедлайн, отмена, прогресс), newton_continuation (contact_strategy="newton" — основной режим контакта с 05.10.2026; детектор механизма), cost_estimate (оценка стоимости расчёта до запуска), mesh (opt-in сетка с узлами в характерных глубинах); кэш сборки K — в beam_solver и elasto_plastic (глава 6).
`hydroweb/api/ (дополнение)`	calc_status (статусы результата), calc_runtime (семафор, слежение за разрывом соединения, дедлайн по умолчанию), jobs (фоновые задачи, таблица calc_jobs), aquifers (вода слоями), fem_support_checks (N ⩽ N_пред), load_combinations_fem; роутеры routers/calc_jobs.py, checks_hydraulic.py, checks_stability.py, checks_anchors.py.
`hydroweb/estimate/`	Сметный движок (ресурсно-индексный метод): gesn.py (загрузчик ГЭСН-2022 и ФСБЦ_Маш, NormBook), pricebook.py (PriceBook, SnapshotPriceBook), nr_sp.py (нормативы НР/СП по приказам 812/пр и 774/пр), calc.py (позиция по норме, позиция-ресурс, ЛСР), summary.py (ОСР, ССР, главы 8 и 9, резерв, НДС), support_norms.py (кандидаты норм и нормы вне области подбора), support_selection.py (подбор нормы анкера, распорки, пояса по явным условиям), ssr_hints.py (подсказки к ВЗиС и зимнему удорожанию с цитатами приказов), engine.py (сборка ответа), registry.py (ленивая загрузка книги норм и снимка), constants.py (источник, «не учтено», допущения). Роутер — api/routers/estimate.py, раздел «Сметный движок» главы 8.

Путь	Содержимое
`hydroweb/data/ (смета, КСР)`	fgis/snapshot_2026Q3.json.gz (снимок цен, 147 из 148 ценовых зон, 2,5 МБ) и .build_report.json, fgis/gesn2022_20260709.json.gz (книга норм, 169 КБ), fgis/scope.json (область снимка), estimate/nr_sp_812_774.json (нормативы НР/СП), estimate/zone_notes_2026Q3.json (ограничения применения цен зон из примечаний источника), ksr_binding.json и ksr_excerpt.json (привязка каталога шпунта к КСР и выписка из классификатора).
`scripts/ (смета, КСР)`	fgis/ (download_splitforms.py, build_snapshot.py), estimate/ (build_norm_book.py, build_nr_sp_table.py, build_test_fixture.py), catalog/build_ksr_binding.py; команды — в разделе «Снимок цен и книга норм» главы 8.
`frontend/src/ (документы и смета)`	modules/documentation/estimate/ (EstimatePanel.tsx, buildEstimate.ts, normSelection.ts, xlsxEstimate.ts, __fixtures__ — настоящие ответы бэкенда), modules/documentation/teo/ (том ТЭО: TeoDialog.tsx, teoFinder.ts, teoVariants.ts, teoEstimate.ts, teoContent.ts, docxTeo.ts), services/estimateApi.ts + estimateTypes.ts, modules/documentation/export/ (xlsxSpecification.ts, gostForm.ts, metalSpec.ts, docxExplanatoryNote.ts, volumeSheets.ts — документы тома), modules/documentation/core/projectMark.ts (марка комплекта, обозначения, номер части раздела КР), modules/documentation/core/costEstimate.ts (ручной «Расчёт стоимости»: число оборотов, ГЭСН п. 1.5.6), reports/ (gostDoc.ts, gostStamp.ts, gostPlain.ts, reportLayout.ts, reportOptions.ts — параметры выгрузки, numFormat.ts — десятичная запятая в Word, kzLabel.ts — подпись запаса в заделке, certificateInfo.ts — реквизиты сертификата, xlsxStyle.ts, xlsxTable.ts, platformInfo.ts, platformVersion.ts), testing/ (decimalDots.ts — сторож десятичной точки, demoFixtures.ts, forbiddenStrings.ts), modules/catalog/core/ksrBinding.ts + data/ksrBinding.json, components/ReportOptionsPanel.tsx (режим листа и реквизиты отчёта на «Результатах»), modules/drawing/core/titleBlockBuilder.ts (основная надпись чертежа, форма 3).
`frontend/src/ (подбор решений)`	components/solution-finder/steps/ (WallChoiceStep.tsx — шаг «Ограждение»), components/solution-finder/results/ (SolutionTuner.tsx — настройка варианта, AssistantConclusions.tsx — выводы), services/solutionEngine/ (calcClient.ts, catalog.ts, conclusions.ts, finderProject.ts, quickInputToProject.ts, profileMarks.ts, schemeProps.ts, wallChoice.ts, supportSizing.ts, variantDraft.ts, solutionWords.ts).
`scripts/`	Скрипты по темам: gw_compare/ (сверка с эталонной программой: prod_sweep.py, храповик ratchet_check.py, refs_paths.py — пути к эталонным наборам, refs_requests/, import_gwx.py — прототип импорта файлов проектов стороннего расчётного ПО, gate_sets.py — реестр необязательных наборов (gwx_projects включён, pos нет), pos_sweep.py — пробный набор из семи расчётов, manual_examples_v8.py и manual_examples_gwx.py — аналоги примеров и контрольных задач руководства программы-эталона (версия 8.4) и их расчёт по входу из файлов, analog_2139_*.py — аналог расчёта 2139 и сверка, research/ — разовые замеры (steel_modulus_sweep.py — влияние модуля упругости стали E на сверку, pos_datum_shift_scan.py — чувствительность набора ПОС к началу отсчёта K, ry_table_v5_sweep.py — подмена таблицы Ry (табл. В.5 СП 16) только в скрипте замера, без правки продукта, snap_angles_measure.py — округление углов при импорте, c1… c5_*.py — замеры по 11 файлам проектов проектировщика и results_projects_0708/ с числами), README; ни один из них, кроме gwx_projects по флагам, не входит в храповик и базы), bench/ (solver_speed.py, cost_calibration.py), prototypes/arclen/ (прототип прямого Ньютона и сверка с Пикаром), books/ (сборка руководств и рисунков), registry/ (документы для Реестра ПО), knowledge_base/ (build_kb.py, kb_scrub.py), legal/, certification/, documents/, experiments/, maintenance/ (prod_health.py — монитор прода),

Путь	Содержимое
	estimate/, fgis/, catalog/ (генераторы сортаментов труб и двутавров/швеллеров build_pipe_catalog.py и build_beam_catalog.py, выписки таблиц стандартов в каталогах *_sources/ с протоколами проверок, скрипты извлечения и проверки выписок, confirmation_stats.py — доля ячеек, подтверждённых вторым независимым изданием); import_egrz.py лежит в корне scripts/ (его импортирует приложение).
`документы/`, `база_знаний/`	Документы проекта (руководства, реестр ПО, сертификация, нормативы, литература, проектные записки, заметки по этапам, промпты, сверка расчётов, аудиты, архив) и источники ИИ-ассистента с командой сборки базы знаний. Индекс — документы/README.md, карта переезда 05.10.2026 — документы/КАРТА_ПЕРЕМЕЩЕНИЙ.md.
`эталонные_расчёты/`, `рабочие_материалы/`, `резервные_копии/`	Эталонные отчёты и запросы набора проектировщика; рабочие выгрузки и данные ФГИС ЦС; резервные копии баз и .env (секреты). В git не хранятся, на сервер не уходят.
`frontend/src/` (дополнение)	services/projectToRequest.ts (buildFemRequest, buildBlumRequest) и projectFieldFlow.ts (реестр + сторож), femJobs.ts, checksInput.ts, aquifersPayload.ts; hooks/useCalcProgress.ts; core/calcStatus.ts, applicability.ts, supportChecks.ts, residualStrength.ts; components/StagesTab.tsx (вкладка «Этапы»), StagePanel.tsx (правая панель), stageCharts.ts (данные эпюр этапа), DenseTable.tsx (общая таблица ввода), HelpTip.tsx («?»), SubgradeKNotice.tsx, components/CalcStatusBlock.tsx, ChecksSection.tsx (HydraulicChecksPanel, StabilityPanel, AnchorCapacityPanel), AnchorRootInputs.tsx, AnchorTendonDialog.tsx (окно «Тяга анкера N»), core/supportStiffness.ts (жёсткость связи по сечению или тяге), ResidualStrengthPanel.tsx, AquiferSection.tsx, LateralCoefficientsTable.tsx.
`hydroweb/ai/`	ИИ-ассистент: knowledge_base (kb.db, SQLite + numpy, косинусный поиск), embeddings (Yandex), chunking, ingest (сборка базы), sanitize (исключение документов и вырезание названий третьих лиц), source_links (ссылки на открытые источники) — см. главу 8.
`hydroweb/data/`	eco_profiles.json, марки стали и прочие каталоги; ground_anchors_tu.json — каталог грунтовых анкеров по ТУ 25.11.23-001-22252558-2024 ([0524-каталог-анкеров-по-ту]): 23 штанги (items) и 198 записей комплектующих (components: муфты, гайки, плиты, центраторы, диаметры и массы долот), у каждой записи source со страницей ТУ; terms (срок службы, гарантия) и gaps (чего в ТУ нет). Файл генерируется: scripts/catalog/build_ground_anchors_tu.py из CSV в scripts/catalog/ground_anchor_sources/ (--check сверяет актуальность), второй проход по тексту PDF — scripts/catalog/verify_ground_anchors_tu.py (549 значений, расхождений 0).
`hydroweb/tests/`	test_<модуль>.py; test_gw_references.py (сравнение с пром. аналогом); test_fem_* (МКЭ); tests/benchmarks/ (реестр контрольных задач); test_assistant_quota.py, test_profile_account/_password/_consents/_export/_delete.py, test_admin_audit/_session_epoch/_invite/_health.py (кабинет, квота ассистента, расширенная админка); test_login_lockout.py, test_session_lifecycle.py (блокировка входа, срок и отзыв сессии), test_container_nonroot.py, test_requirements_lock.py, test_nginx_site_conf.py (контейнер, lock-файл, заголовки и CSP); test_new_project_template_fem.py с эталонным телом запроса fixtures/new_project_fem_payload.json.
`frontend/src/`	App.tsx (вкладки редактора — TabNavigation.tsx и editorTabs.ts: восемь вкладок), AppWrapper.tsx (навигация, ErrorBoundary верхнего уровня, экраны first-login и «Профиль»), components/, solution-finder/ (b2c;

Путь	Содержимое
	SolutionFinderAssistant.tsx — ассистент мастера), modules/drawing/ (план и разрез; export/ — SVG, DXF и векторный PDF: pdfExporter.ts, шрифт drawingPdfFont.generated.ts, [0526-pdf-чертежа-с-текстовым-слоем]), modules/documentation/ (записка и подраздел «Анкерное крепление» — core/anchorSection.ts, спецификация в XLSX и DOCX — export/, расчёт стоимости «ручной ввод» — core/costEstimate.ts, том ТЭО — teo/), modules/catalog/ (карточки шпунта, марок стали, сечений распорок), reports/, services/solutionEngine/, services/assistantHistory.ts, assistantQuota.ts, assistantContext.ts, assistantAccess.ts, normLinks.ts + data/openSources.ts (ассистент), projectExport.ts (перенос проектов), newProjectTemplate.ts (шаблон нового проекта), components/admin/ (девять вкладок администрирования), accessLevels.ts (названия и подписи уровней доступа), services/betaEvents.ts (отчёт браузера о выгрузках в журнал событий), ProfilePage.tsx + profileApi.ts + profileLocalData.ts (кабинет), FirstLoginScreen.tsx, ErrorBoundary.tsx + errorReport.ts, ResultsVerdict.tsx + core/resultsVerdict.ts (плашка результата), softPromptPref.ts, types.ts.
`hydroweb/importers/ (импорт файлов проектов)`	gwx.py, request_builder.py, project_model.py, messages.py, file_import.py — разбор .gwx/.gw4 в запрос расчёта и модель проекта интерфейса ([0459-импорт-проектов-в-продукт]); маршрут — api/routers/project_import.py.
`hydroweb/core/ (вода и узлы)`	water_inflow, fragment_method (приток воды в котлован; вход — api/routers/checks_water.py, [0460]) и joints (швы и болты по СП 16; вход — api/routers/joints.py, [0462]); кабинет поставщика — api/supplier_specs.py и api/routers/supplier.py ([0461]).
`hydroweb/api/ routers/ price_requests.py`	Запросы цены поставщикам ([0478-запрос-цены], [0528-экран-запросов-цены-администратора]): одиннадцать маршрутов проектировщика, поставщика и администратора; таблицы price_requests, price_request_targets, price_offers (PRICE_REQUEST_SCHEMA в db/connection.py) — глава 8.
`hydroweb/api/ routers/ supplier_usage.py`	Продукция поставщиков в проектах ([0498-продукция-поставщика-в-проектах]): сбор строк сводки из полей проекта при сохранении (record_project_usage, collect_usage), удаление строк (forget_project_usage), два маршрута сводки: поставщика и администратора (глава 8, раздел «Продукция поставщиков в проектах»); таблица supplier_usage (SUPPLIER_USAGE_SCHEMA в db/connection.py). Рядом hydroweb/api/regions.py: normalize_region приводит текст поля проекта «Регион» к названию одного из 89 субъектов Российской Федерации либо к пустому значению; в routers/supplier.py маршруты принятия условий кабинета (terms_accepted, GET /api/supplier/terms, POST /api/supplier/accept-terms).
`hydroweb/api/ routers/ supplier_applications.py`	Регистрация по ролям и заявки поставщиков ([0485-регистрация-ролей]): одиннадцать маршрутов (глава 8, раздел «Регистрация по ролям и заявки поставщиков»); таблица supplier_applications (APPLICATION_SCHEMA в db/connection.py), столбец users.account_kind; файлы карточек организации лежат в каталоге supplier_cards/ рядом с базой (том данных, не frontend/public). В routers/project_import.py из extract_upload выделен parse_form (разбор частей формы).
`frontend/src/ (регистрация по ролям)`	components/AccountKindChoice.tsx (две карточки «Я проектировщик» и «Я поставщик»), AccountKindScreen.tsx (первый вход через Яндекс), SupplierApplicationFields.tsx (поля заявки), SupplierApplicationStatus.tsx (экран «Заявка на проверке», отказ, одобрение), admin/AdminApplicationsTab.tsx (вкладка «Заявки поставщиков»), services/supplierApplication.ts (проверка до отправки и запросы), AuthPage.tsx (первый шаг регистрации), AppWrapper.tsx (экраны account-kind и supplier-status), admin/auditFormat.ts (подписи трёх действий журнала); legal.ts (LEGAL_VERSION, LEGAL_DOCS,

Путь	Содержимое
	SUPPLIER_CONSENT_LABEL).
`frontend/src/ (линии 24.10.2026)`	core/substage.ts (подэтап «k.F0» на плашке этапов); core/expertiseReadiness.ts и modules/documentation/components/ExpertiseReadiness.tsx (готовность к экспертизе); data/resultBasis.ts, core/resultBasisView.ts, components/BasisCard.tsx и useBasisCard.ts (карточка «Откуда это число»); services/variantComparison.ts, variantInputDiff.ts, variantCost.ts, components/VariantComparison.tsx и reports/generateComparisonDocx.ts (сравнение вариантов); services/priceRequestApi.ts, modules/documentation/core/priceRequestSeed.ts, components PriceRequestDialog.tsx и PriceRequestsPanel.tsx, components/supplier/SupplierRequestsList.tsx (запрос цены); services/planSizeProject.ts и hooks/usePlanSize.ts (размеры котлована в плане: поле проекта planSize); readinessPref.ts (разворот списка готовности); services/projectInfo.ts и modules/documentation/teo/teoCard.ts (реквизиты проекта и карточка тома ТЭО в проекте: поля projectInfo и teoCard), teo/teoEstimateTexts.ts (словарь формулировок сметного модуля для тома); core/stageResultsTable.ts и components/StageResultsTable.tsx (таблица по этапам, «Кз, справочно»).
`frontend/src/ (продукция поставщиков в проектах)`	services/specSummary.ts (buildSpecSummary: строки сводки по объёмам спецификации; specInputKey: отпечаток ввода, по которому сводка устаревает) и supplierUsageView.ts (подписи и числа для показа); components/supplier/SupplierUsageSection.tsx (раздел кабинета «Моя продукция в проектах» с приглашением принять условия), UsageTable.tsx (таблица и разворот по месяцам и регионам, общие для кабинета и администрирования); components/admin/AdminUsageTab.tsx (вкладка «Продукция поставщиков в проектах»); строка о сводке под таблицей расценок — modules/documentation/components/CostEstimatePanel.tsx; services/documentsProject.ts держит расценки, запас на материал и цену шпунта из сметы в проекте; services/supplierApi.ts (supplierApi.usage, terms, acceptTerms; adminUsageApi).
`.github/workflows/ci.yml`	Непрерывная интеграция на push и pull request в release/v1.1: шесть заданий, те же ворота, что перед выкаткой, — глава 9, раздел «Непрерывная интеграция».

Схема БД: что добавил релиз

Все изменения — только добавляющие (CREATE TABLE IF NOT EXISTS в _SCHEMA и идемпотентные ALTER в _MIGRATIONS файла db/connection.py), откат на прежний образ безопасен: старый код лишних таблиц и столбцов не замечает.

Когда выполняются схема и миграции ([0343]). Один раз при старте каждого процесса uvicorn (lifespan в `api/main.py` → `db/connection.init_db`), изменения фиксируются сразу. Соединение запроса (`get_async_db`) только читает PRAGMA schema_version и повторяет схему и миграции, если схема файла изменилась (новый файл или путь, восстановление таблиц restore_db.sh на работающем контейнере); init_sync (скрипты, тесты) тоже фиксирует миграции сразу. Сбой при старте пишется в журнал (DB_INIT …), процесс не останавливается, схема и миграции повторятся на первом соединении. Ошибки миграций больше не глотаются молча: «duplicate column name» — норма, любая другая — предупреждение DB_MIGRATION в журнале, и файл не считается готовым (следующее соединение повторит); тексты самих миграций не менялись. Прежде схему и миграции выполняло КАЖДОЕ соединение, и data-миграция оставляла его в транзакции с блокировкой записи до конца запроса — залп входов получал 500

«database is locked». Ожидание чужой блокировки — 30 с на каждом соединении модуля (BUSY_TIMEOUT_S; было 5 с по умолчанию sqlite3). Режим журнала прежний (не WAL): копии тома tar-ом при WAL могут молча терять зафиксированные данные — подробности в разделе «Вход под нагрузкой» главы 10.

Объект	Назначение
`admin_audit`	Журнал действий администратора: id, ts, actor_id, action, target_id, details_json; индексы по ts, actor_id, target_id. Без внешних ключей — строки переживают удаление пользователя. Пишет только record_admin_action (api/admin_audit.py); хранение AUDIT_RETENTION_DAYS = 400 суток, чистка при каждой записи.
`assistant_usage`	Счётчик вопросов к ассистенту: user_id, day (календарная дата по Москве, ГГГГ-ММ-ДД), count; первичный ключ (user_id, day), индекс по day. Строки старше 40 суток удаляются при записи; при удалении пользователя — вместе с ним (auth.delete_account). С шага 2а ([0570]) это и месячная квота ассистента по уровню (сумма count по суткам месяца); в суточную копию БД таблица не входит.
`users.session_epoch`	INTEGER NOT NULL DEFAULT 0 — эпоха сессий; токен несёт её в поле ep, «Завершить все сессии» и выход увеличивают счётчик (раздел «Сессия и безопасность»).
`login_attempts`	Счётчики неудачных входов по учётной записи (api/login_guard.py): acct_key (первичный ключ; SHA-256 нормализованной почты), failures, window_start, locked_until (unix-время, с); схема LOGIN_ATTEMPTS_SCHEMA в db/connection.py. Строка живёт не дольше окна подсчёта и срока блокировки, в копию deploy/backup_db.sh не входит.
`users.must_change_password`	INTEGER NOT NULL DEFAULT 0 — приглашённый вошёл по временному паролю и обязан его сменить; снимается set_user_password и clear_must_change_password.
`users.consents_pending`	INTEGER NOT NULL DEFAULT 0 — приглашённый ещё не принял документы; снимается POST /api/auth/invite/accept-consents (записи в consents с source='invite').
`users.account_kind`	TEXT: кто входит в систему, designer | supplier; NULL только у новой записи после первого входа через Яндекс (ей показывается выбор роли). Столбец добавляет ALTER TABLE в _SCHEMA, и в тот же момент, один раз, существующим строкам проставляется значение: supplier у роли supplier, designer у остальных (_AFTER_COLUMN_ADDED в db/connection.py; повторные запуски и повторная проверка схемы новые записи не штампуют). Роль supplier назначает одобрение заявки; до него account_kind = supplier при роли standard означает «на проверке» (auth.supplier_pending). account_kind_of(user) — запасное правило для строк без значения. Приглашённый администратором получает значение при создании.
`supplier_applications`	Заявки поставщиков ([0485-регистрация-ролей]): id, user_id, organization, inn, contact_name, phone, website, file_path (имя файла в supplier_cards/: 32 hex и тип по сигнатуре, не путь от пользователя), file_name (исходное имя для показа), file_size, status (new | approved | rejected), admin_note (причина отклонения или записка), created_at, decided_at, decided_by; индексы по (user_id, created_at) и (status, created_at) и частичный уникальный

Объект	Назначение
	индекс по user_id при status = new: одна активная заявка на пользователя. Схема только добавляющая (APPLICATION_SCHEMA). Строки и файлы удаляются вместе с учётной записью (forget_user_applications, файлы стираются после фиксации базы). Таблица входит в копию deploy/backup_db.sh; файлы карточек копирует полная копия тома (backup_data.sh).
`consents`	Журнал согласий: id, user_id, kind, version (LEGAL_DOC_VERSION на момент согласия), granted, source, accepted_at. kind: terms, pd, marketing (регистрация, приглашение, профиль), price_request_contacts (одна запись на запрос цены с передачей контактов, source = price_request:<id запроса>, пишется в одной транзакции с запросом, отказанный запрос её не пишет) и supplier_terms (принятие условий кабинета поставщика первой отправкой позиции или кнопкой «Принимаю условия» в разделе «Моя продукция в проектах», source = supplier-cabinet, одна запись на редакцию документов). Допустимые kind в запросах клиента (ConsentItem) — только terms, pd и marketing: две записи поставщика делает сервер. Прочие source: register, oauth-yandex, oauth-vk, invite, profile. Записи удаляются вместе с учётной записью, входят в выгрузку «Мои данные» и в копию backup_db.sh.
`users.signup_source`, `users.signup_source_at`, `users.trial_until`	TEXT: метка ссылки ([a-z0-9_-]{1,32}), время перехода по ней (ISO, UTC) и конец пробного периода (naive UTC); NULL у обычной записи, обычная запись без ограничений ([0517-пробный-доступ-по-метке]). Столбцы добавляют ALTER TABLE в _MIGRATIONS; резервная копия берёт users целиком по sqlite_master, поэтому новые столбцы входят в неё сами.
`feedback`	Обращения «Написать нам» ([0517-пробный-доступ-по-метке], [0520-обращения-с-вложением-и-редакция-2026-10-27]): id (32 hex у новых записей), user_id, text, created_at, kind (review | calc_issue | question | proposal; NULL читается как review), file_path, file_name, file_size (прежние столбцы единственного вложения: с [0531-обращения-до-пяти-вложений] не заполняются и не удаляются, вложения лежат в feedback_files), handled_at, handled_by; индекс по user_id. Схема FEEDBACK_SCHEMA в db/connection.py (CREATE TABLE IF NOT EXISTS) и отдельные ALTER TABLE для баз, где таблица уже была. Строки удаляются вместе с учётной записью (auth.delete_account), файлы после фиксации базы. Таблица входит в backup_db.sh (пятнадцатая); файлы вложений в дамп не входят, их копирует полная копия тома (backup_data.sh).
`feedback_files`	Вложения обращений ([0531-обращения-до-пяти-вложений], до пяти на обращение): id (32 hex), feedback_id, file_path (имя файла в feedback_files/: <32 hex>.<тип>, не путь от пользователя), file_name (очищенное исходное имя только для показа), file_size, created_at; индекс idx_feedback_files_feedback — индекс по feedback_id. Схема в FEEDBACK_SCHEMA (db/connection.py). Вложения из прежних столбцов feedback.file_path, file_name, file_size (прежние столбцы единственного файла) переносятся в feedback_files при старте процесса (_DATA_MIGRATIONS): один INSERT … SELECT … WHERE NOT EXISTS (… file_path) на запись, идемпотентно при каждом старте и после возврата из копии прежнего состава; имя берётся из file_name (нет — имя на диске), размер из file_size (нет — 0). Строки удаляются вместе с обращениями

Объект	Назначение
	(auth.delete_account, forget_user_feedback), файлы стираются с диска после фиксации базы. Таблица входит в backup_db.sh (шестнадцатая, после feedback).
`events`	Журнал событий беты ([0543-журнал-событий-беты]): id (INTEGER AUTOINCREMENT), user_id, kind (calc_fem, calc_blum, import_file, project_save, export_docx, export_xlsx, export_pdf, export_dxf, export_svg, export_xml — routers/events.KINDS), created_at (naive UTC до секунды), meta_json — только служебное по белому списку routers/events.clean_meta (project_id по форме [A-Za-z0-9_-]{1,64}, part, engine, stages, status, new и метка сервера src = no_project; любой другой ключ и значение не по форме отбрасываются: названий, имён файлов и содержимого проектов в строке нет); индексы idx_events_user_time (user_id, created_at) и idx_events_time (created_at). Схема EVENTS_SCHEMA в db/connection.py (CREATE TABLE IF NOT EXISTS). Строки по сроку не удаляются; auth.delete_account удаляет события пользователя (как calc_jobs), а при старте процесса _DATA_MIGRATIONS снимает строки пользователей, которых нет в users (удаление во время фоновой записи, возврат users из более старой копии). Запись — INSERT … SELECT … WHERE EXISTS (SELECT 1 FROM users …): удалённого пользователя фоновая запись не воскрешает. Таблица входит в backup_db.sh (семнадцатая, после feedback_files), в restore_check.sh и в выгрузку «Мои данные» (раздел events: вид и время).
`users.access_level`, `users.access_level_until`, `access_level_changes`	Уровни доступа ([0548-уровни-доступа-шаг-1]). users.access_level — идентификатор уровня (trial | free | start | pro | team | university) или NULL: администратор уровень не назначал, действует BETA_DEFAULT_LEVEL = pro (в интерфейсе «Профи (бета)»); users.access_level_until — конец срока (naive UTC), NULL — бессрочно. Столбцы добавляют ALTER TABLE в _MIGRATIONS, существующим строкам значения не проставляются. access_level_changes — журнал смен: id (INTEGER AUTOINCREMENT), user_id, changed_at, changed_by (id администратора; NULL — система), old_level, new_level, old_until, new_until, reason (обязательна, до 300 знаков); индекс idx_access_level_changes_user (user_id, id); схема ACCESS_LEVEL_SCHEMA. Строки удаляются вместе с учётной записью (auth.delete_account) и входят в выгрузку «Мои данные» (accessLevelChanges, без идентификатора администратора). В суточную копию backup_db.sh таблица не входит (список остался семнадцатью таблицами; в test_backup_scripts.py она записана в not_backed_up с причиной): уровень лежит в users и копируется вместе с ней, факт смены записан в admin_audit; журнал смен с причиной есть только в полной копии тома (backup_data.sh). Включить таблицу в копию — отдельная правка трёх скриптов, тестов и этого описания.
`calc_jobs`	Фоновые задачи МКЭ-расчёта ([0293-C3]): id, user_id (NULL — расчёт без входа), owner_key (user:<id> | ip:<адрес>), state (queued | running | done | failed | cancelled), created_at, started_at, finished_at, heartbeat_at, cancel_requested, progress_json, result_json, error_json, owner_pid; индексы по (owner_key, state) и finished_at. Создаётся лениво (CREATE TABLE IF NOT EXISTS при первом обращении) и входит в _SCHEMA. Результат хранится 1 час после завершения; задача с heartbeat_at старше 60 с помечается failed.

Объект	Назначение
`supplier_items`	Позиции кабинета поставщика ([0461-кабинет-поставщика], [0471-кабинет-поставщика-v2]): владелец (поставщик), вид (восемь категорий; прежняя «марка стали» только у созданных позиций, одобренные позиции этого вида отдаёт GET /api/steel-grades), характеристики с единицами, цена с датой и сроком, наличие, регион, документ, статус модерации и записка модератора; detail_json (срок поставки, минимальная партия, длины шпунта), history_json (лента статусов). Схема только добавляющая; удаление учётной записи удаляет позиции.
`supplier_profiles`, `supplier_demand`	Профиль организации поставщика (название, ИНН, контакты, регионы, документ-основание) и спрос (item_id, project_id: в скольких проектах выбрана одобренная позиция; число проектов, не сохранений). Добавляющие таблицы [0471]; удаляются вместе с учётной записью.
`supplier_usage`	Сводка продукции поставщиков в проектах ([0498-продукция-поставщика-в-проектах]): project_id (случайный идентификатор проекта), code (supplier:<id позиции> | catalog:<ключ встроенного каталога>), category (sheet_pile | pipe | i_beam | channel | anchor_tendon), name, unit (шт | м.п. | т), quantity, unit_price (цена за тонну у строк в тоннах, иначе NULL), price_source (manual | rim | fgis), region (название субъекта РФ или NULL), used_at (момент последнего сохранения проекта); первичный ключ (project_id, code, unit), индекс по (code, used_at). Названия проекта, имён, организации и почты проектировщика в таблице нет. Строки пересобираются при каждом сохранении проекта и удаляются вместе с проектом, позицией и учётной записью проектировщика или поставщика; таблица входит в копию deploy/backup_db.sh. Схема только добавляющая (SUPPLIER_USAGE_SCHEMA).
`projects.data_json → specSummary, costRates, materialReserveFactor, sheetPileRimPrice`	Не таблица, а необязательные поля данных проекта ([0498-продукция-поставщика-в-проектах]; types.ts, реестр projectFieldFlow.ts, вид «отчёт»: в расчёт не идут). specSummary — строки спецификации в сжатом виде (category, code, name, quantity, unit; у строк в тоннах rate, rimPrice и rimPriceSource), costRates — расценки с вкладки «Документы», materialReserveFactor — запас на материал (от 1,0 до 2,0), sheetPileRimPrice {price, source} — цена шпунта из сметы РИМ с происхождением (user | fgis). Сервер хранит их общим путём вместе с проектом и собирает из них supplier_usage.
`projects.data_json → teoCard, projectInfo, anchors.rows[].tendon`	Необязательные поля данных проекта ([0503-прототип-таблица-распорок], [0504-тэо-по-ревизии]; types.ts, реестр projectFieldFlow.ts: teoCard и projectInfo — вид «отчёт», tendon — «ui»; в расчёт не идут). teoCard — карточка тома ТЭО (тексты, цены металла, замечания и флажки вариантов по названию профиля); projectInfo — реквизиты вкладки «Документы», только отличные от умолчаний; anchors.rows[].tendon — снимок выбранной тяги анкера (id позиции каталога или supplier:<id>, обозначение, стандарт, число тяг, A, E). Сервер хранит их общим путём вместе с проектом; id поставщика в tendon учитывается спросом (supplier_demand).
`price_requests`	Запрос цены ([0478-запрос-цены]): id, user_id, project_id, category (типы supplier_items.kind, кроме прежнего steel), name, mark, quantity, unit (шт | м.п. | кг | т), comment, contact_consent, requester_org (сохраняется только при согласии), to_all, status (new

Объект	Назначение
	| answered | declined | closed), created_at, updated_at; индексы по (user_id, created_at), (user_id, project_id), (status, category). Пометка администратора «просмотрено» ([0528-экран-запросов-цены-администратора]): admin_seen_at (время; пусто — запрос новый) и admin_seen_by (кто из администраторов поставил пометку); столбцы есть в CREATE TABLE и добавляются ALTER TABLE в прежние базы, прежние запросы числятся новыми; на статус и updated_at запроса пометка не влияет.
`price_request_targets`	Адресаты запроса: первичный ключ (request_id, supplier_user_id), state (new | answered | declined), decline_reason, decided_at; индекс по (supplier_user_id, state). Записываются при создании запроса.
`price_offers`	Предложения поставщиков: id, request_id, supplier_user_id, org, price, price_unit (rub_per_t | rub_per_m | rub_per_pc | rub_per_kg), currency (RUB), price_date, valid_until, comment, created_at; уникальный индекс (request_id, supplier_user_id): повторное предложение заменяет прежнее. Схема только добавляющая; все три таблицы удаляются вместе с учётной записью (auth.delete_account, forget_user_requests), статусы затронутых запросов пересчитываются.
`projects.data_json → variants`	Не таблица, а поле данных проекта ([0475-сравнение-вариантов]): запомненные варианты для сравнения. Сервер поля не знает, хранит его общим путём вместе с проектом (глава 8, раздел о сравнении вариантов).

Глава 4. Конвенции кода

Python

- Type hints обязательны на публичных функциях; Pydantic-модели на все request/response API. Field(..., ge=0, le=...) — часть security-аудита, не убирать.
- Единицы: длины/глубины — метры, давления — кПа, моменты — кН·м/м, силы — кН/м, напряжения — МПа. Любые конверсии — только через core/units.py.
- Логирование через logging.getLogger(__name__), не print. Один core/X.py → один tests/test_X.py.
- НЕ возвращать str(e) в HTTP-ответах (generic message + stack в лог).

TypeScript

- Все доменные типы — в одном types.ts. Компоненты и файлы — PascalCase.
- Единицы в UI — на русском; технические ключи API/БД — английский snake_case (excavation_depth, не глубина_котлована).
- import type { … } для типов; import { … } для значений. НЕ использовать any (только unknown с проверкой). Тяжёлые библиотеки (docx, xlsx, jspdf) — динамический import().

Коммиты и журнал

Префиксы feat:/fix:/docs:; сообщения по-русски — норма. Журнал `CHANGELOG_DEV.md` — командная память, коммитится вместе с кодом в том же коммите; новые записи [NNNN] — сверху файла.

Размещение файлов и именование (порядок в проекте с 05.10.2026)

05.10.2026 каталоги проекта приведены в порядок: корень стал коротким, документы и материалы разложены по назначению, имена не-кодовых файлов — по-русски, скрипты — по темам. Правила закреплены в CLAUDE.md (раздел 10а), старые пути сопоставлены с новыми в документы/КАРТА_ПЕРЕМЕЩЕНИЙ.md.

- **Корень проекта** содержит только: README.md, CLAUDE.md, CHANGELOG.md, CHANGELOG_DEV.md, VERSION, Dockerfile, docker-compose.yml, pyproject.toml, .env.example, .gitignore, .dockerignore; каталоги кода и инфраструктуры hydroweb/, frontend/, scripts/, deploy/, .github/; каталоги данных и материалов документы/, база_знаний/, эталонные_расчёты/, рабочие_материалы/, резервные_копии/; служебные (.claude/, .backups/, .venv/, _корзина/ — временный архив перед удалением, _рабочие_копии/ — рабочие деревья git на время задач) и генерируемые. Новый каталог в корне не создаётся.
- **Что не едет на сервер.** Не-кодовые каталоги (документы/, база_знаний/, эталонные_расчёты/, рабочие_материалы/, резервные_копии/, _корзина/, _рабочие_копии/) исключены из rsync в deploy/deploy.sh и из образа (.dockerignore); резервные_копии/, _корзина/ и _рабочие_копии/ не попадают и в git (.gitignore), причём резервные_копии/ хранит секреты.
- **Язык имён.** Код и инфраструктура (hydroweb/, frontend/, scripts/, deploy/, .github/ и перечисленные корневые файлы) — латиницей, как требуют инструменты. Всё остальное

(документы, материалы, данные для людей) — по-русски, без пробелов (подчёркивания), без транслитерации; допустимы бренд HydroWeb и общепринятые аббревиатуры (СП, ГОСТ, МКЭ, ТЭО, API). Идентификаторы вроде k4b_13, EP-2 и номера расчётов (1949.3) остаются как есть.

- **scripts/ — по темам:** books/ (сборка руководств и рисунков), registry/, knowledge_base/, legal/, certification/, documents/, experiments/, maintenance/, gw_compare/ (сверка), estimate/, fgis/, catalog/, bench/, prototypes/. Новый скрипт кладут в тематический подкаталог; имена файлов латиницей и без транслитерации. Исключение — scripts/import_egrz.py: его импортирует приложение. Пути к эталонным наборам сверки собраны в одном модуле scripts/gw_compare/refs_paths.py.
- **Имена, которые менять нельзя,** потому что на них завязаны код и данные: имена документов в базе знаний ассистента (hydroweb/data/kb.db и hydroweb/ai/source_links.py: SP-381.1325800.2018.pdf, user_guide.md, developer_guide.md, supplier_guide.md, fem_theory.md), имена, по которым работают защитные фильтры hydroweb/ai/sanitize.py, всё под frontend/public/ (отдаётся сайтом по адресам вроде /docs/user.html), hydroweb/data/*, scripts/gw_compare/refs_requests/* (номера расчётов), данные ФГИС ЦС внутри рабочие_материалы/фгис_цс/ (имена источника — в тестах).
- **Переименование — не правка текста.** Исторические документы (аудиты, архив, заметки по этапам, промпты, письма, CHANGELOG_DEV.md) сохраняют в тексте прежние пути: они фиксируют состояние на свою дату; соответствие старых путей новым — в документы/КАРТА_ПЕРЕМЕЩЕНИЙ.md.

Глава 5. Расчётное ядро: два движка

HydroWeb содержит два независимых расчётных движка для подпорной стенки; они не конкурируют, а покрывают разные классы задач.

	Блюм (core/embedment.py)	МКЭ (core/fem/)
Модель	Предельное равновесие (жёстко-пластичный грунт)	Балка Эйлера–Бернулли на упругопластическом основании Винклера
Теория	СП 381, принцип предельного равновесия	Hetényi, Bowles §13, Zienkiewicz; СП 22, СП 16
Схема	Консоль или одна верхняя связь	Многоярусные анкеры/распорки, многоэтапная отрывка
Выход	d_min, d_design, Mmax, Q, усилие связи, kst (в интерфейсе — Кз)	Полные эпюры w/M/Q/p, реакции, огибающая по этапам, коэф. запаса
Cantilever без связи	да (замыкание через R_toe)	нет (rigid body motion — глава 6)

Связь движков не эвристическая

В пределе k → ∞ МКЭ-упругопластика символьно (sympy) сводится к кубическому уравнению Блюма $K_a \cdot (H+d)^3 = K_p \cdot d^3$.

Это математическое доказательство, что оба движка решают одну физическую задачу. Активный предел σ = σa даёт классический Ренкин $E_a = \frac{1}{2}\gamma H^2 K_a$.

5.1. Теория давления грунта

Принцип эффективных напряжений (Терцаги)

В основе расчёта — принцип эффективных напряжений К. Терцаги: полное напряжение в точке грунта складывается из эффективного (передаваемого скелетом) и порового давления воды, σ = σ′ + u. Прочность грунта и его боковое давление определяются именно эффективным напряжением. Ниже уровня грунтовых вод вертикальное эффективное напряжение накапливается со взвешенным удельным весом:

$$\sigma'_v(z) = \sum_i \gamma'_i \, h_i$$

$$\gamma' = \gamma_{sat} - \gamma_w$$

$$u(z) = \gamma_w \, (z - z_w)$$

Модель воды в ядре — «свободная» (СП 22): взвешивание γ′ в σ′_v плюс отдельное поровое давление u = γ_w·(z − z_УГВ), приложенное к стенке. «Связанная» вода (γ_sat без отдельного u) — per-project опция.

Активное и пассивное давление: предельное равновесие

Когда стенка отходит от грунта, за ней формируется активный клин обрушения (площадка 45° + ф/2); давление минимально — активное. При вдавливании стенки поднимается пассивный клин

(площадка 45° − φ/2); отпор максимален — пассивный. Теория Рэнкина (гладкая вертикальная стенка, δ = 0) даёт замкнутые выражения через круг Мора:

$$K_a = \tan^2\left(45^\circ - \frac{\varphi}{2}\right)$$

$$K_p = \tan^2\left(45^\circ + \frac{\varphi}{2}\right)$$

Для связного грунта добавляется вклад сцепления: σa = Ka·σ′v − 2c√Ka, σp = Kp·σ′v + 2c√Kp. Теория Кулона обобщает Рэнкина на наклонную грань стенки (α), уклон поверхности (β) и трение «грунт-стенка» (δ):

$$K_a = \frac{\cos^2(\varphi - \theta)}{\cos^2\theta\cos(\delta+\theta)\left[1+\sqrt{\dfrac{\sin(\varphi+\delta)\sin(\varphi-\beta)}{\cos(\delta+\theta)\cos(\theta-\beta)}}\right]^2}$$

В ядре пассивный δ зануляется при φ > 20° (консервативно, СП 22.13330 п. 9.23). При β ⩾ φ клин Кулона неустойчив — формула бросает NumericalInstabilityError.

Давление покоя (Jáky / K0) и метод Блюма

Пока стенка неподвижна, грунт находится в состоянии покоя (K0 между Ka и Kp): классическая эмпирика K0 = 1 − sin φ (Jáky); в FEM-ядре как опорное берётся K0 = ν/(1−ν)·(1 + sin β), зажатое в [σa, σp].

$$K_0 = \frac{\nu}{1-\nu}$$

Метод Блюма (СП 381) — расчёт по предельному равновесию с моделью защемлённой стенки (fixed-earth support). Ниже точки нулевого давления стенка условно защемлена; вместо распределённого отпора у носка вводится сосредоточенная эрзац-сила (Ersatzkraft) R_toe, замыкающая равновесие. Глубина заделки d находится из моментного равновесия; проектная — с запасом d = 1,2·d_min; устойчивость — kst = Mp/Ma.

$$\sum M_O = 0 \;\Rightarrow\; M_a(d) = M_p(d)$$

$$d = 1{,}2\, d_{min}$$

$$k_{st} = \frac{M_p}{M_a} \geq [k_{st}]$$

Сейсмика (Мононобе-Окабе)

Сейсмическое воздействие — псевдостатическая поправка к активному/пассивному коэффициенту через угол инерции θ = arctan(k_h/(1−k_v)) (coefficients.py: seismic_kh, seismic_inertia_angle, mononobe_okabe_kae/kpe). Горизонтальный сейсмокоэффициент:

```
k_h = A(баллы) / g · k1 · k2,   g = 9,81 м/с²
```

A — расчётное сейсмическое ускорение по балльности площадки, СП 14.13330.2018 п. 5.17, нормируется в м/с² и переводится в доли g делением на g: 7 баллов → 1,0; 8 → 2,0; 9 → 4,0; < 7 баллов → A = 0 (сейсмика не учитывается). k1, k2 — коэффициенты допускаемых повреждений и ответственности сооружения, задаются инженером. Прежняя запись таблицы A сразу в долях g (0,1 / 0,2 / 0,4) была округлением частного A/g и занижала k_h на 1,9 % (SEISMIC_A_MS2_BY_INTENSITY, [0191]).

Сейсмический угол ψ ограничен сверху: φ − β в активном давлении, φ + β в пассивном (методика эталона §2.1.3, прим. к ф. 25–26; _cap_seismic_angle) — состояние, в котором сейсмика срывает откос; дальше коэффициент остаётся на предельном значении, а не уходит в бесконечность. До [0191] в этой точке ядро бросало NumericalInstabilityError вместо продолжения счёта, как это делает эталон. Отдельно: если предел отрицателен (β > φ уже без всякой сейсмики), ошибка сохраняется — это статическая неустойчивость откоса, а не сейсмическая.

- `seismic_intensity` (FemCalculateRequest) — Optional[int], 6..9; None/6 → k_h=0, сейсмика не учитывается (дефолт, ноль регресса).
- `seismic_k1, seismic_k2` — float, default 1.0, диапазон [0, 5].
- `seismic_kv_ratio` — доля вертикального сейсмокоэффициента k_v = ratio·k_h, default 0.0, диапазон [0, 1].

5.2. Реализация ядра Блюма (core/)

Как расчёт реализован фактически. Все усилия — на 1 погонный метр стенки. Модули — в hydroweb/core/.

coefficients.py — коэффициенты давления

get_coefficients(phi, delta, alpha, beta, method) возвращает {Ka, Kp, Kac, Kpc}. Ренкин: Kac = 2√Ka, Kpc = 2√Kp; при φ ≈ 0 — точные Ka = Kp = 1 (защита от неустойчивости tan). Кулон: полные формулы клина; при β ⩾ φ — NumericalInstabilityError. Пассивный δ зануляется при φ > 20°.

soil_pressure.py — построение эпюры

calculate_pressure строит профиль (σ′v, σa, σp, u, полные σ_total). Сетка глубин — регулярный шаг 0,1 м плюс контрольные точки: z = 0, УГВ и каждая граница слоя тройкой z−ε, z, z+ε (для корректной отрисовки скачка при смене слоя). Одна точка:

- σ_v = compute_sigma_v(z, …) с учётом равномерной пригрузки;
- σa = (σ′v + qa)·Ka − c·Kac, σp = (σ′v + qp)·Kp + c·Kpc;
- **зона отрыва (zero-tension):** при σa < 0 грунтовая часть обнуляется — связный грунт не «тянет» стенку;
- добавка неравномерных пригрузок после обнуления трещины (внешняя нагрузка давит через оторванную зону);
- поровое давление u, полные σa_total = σa + u, σp_total = σp + u.

Метод полосовой пригрузки СП 22 (surcharge_method="sp22"): вклад полосы учитывается раздельно в активной и пассивной ветвях через углы плоскостей скольжения 45° ∓ φ(z)/2, где φ — послойно в точке события. С [0381] (15.10.2026) по умолчанию полоса любой ширины действует на всю высоту стенки; обрезка по дну котлована — только явным surcharge_full_depth = false.

Полосовая пригрузка: бытовое давление и глубина действия

- **В бытовом давлении тыла.** p0 = K0·(σ′v + q) в активной полосе СП 22: равномерная пригрузка сидит в σ′v и входила всегда, полосовая — только в пределы. На малом

консольном этапе стенка стоит около p0, поэтому полоса фактически не толкала её (прогиб первого этапа занижался в 1,3–3 раза).

- **Глубина действия — по каждой полосе.** С [0381] поле surcharge_full_depth без значения (None) — на всю высоту стенки у всех полос, широкая (шире 1,5·H) не обрезается по дну: по опытам сверки (серия с широкой полосой 5,5 и 10 м) обрезки по дну нет, а обрезка занижала момент на ≈5 % и прогиб на ≈13 %. Значения True и False смысла не меняют (False — обрезать все полосы по дну). Замечания о смене умолчания в ответе нет — решением 3.4 оно не предусмотрено; у старых проектов с широкой полосой момент и прогиб при пересчёте вырастут без отдельной плашки.
- **Пионерный срез.** Историческая запись: призма Кулона и полосы СП 22 удалены 22.10.2026 ([0417-срез-по-силам-движок]). Клин Кульмана считался без пригрузок, а пригрузка добавляется полосой СП 22 от натуральной поверхности; в связном грунте прибавка складывается до отсечения зоны растяжения.
- **Груз на дневной поверхности внутри самого клина** (pioneer_culmann.py, удалён 22.10.2026; [0190]). Прежде _Pa_at_Z учитывал пригрузку только за гребнем откоса (a = max(offset, crest)) — полоса рядом со стенкой при широкой берме выпадала из призмы целиком. Область учёта — пересечение полосы с проекцией клина [0, x_int], где x_int зависит от пробного угла θ, поэтому вклад пригрузки участвует В ПОИСКЕ максимума Pa(θ), а не добавляется поверх найденного результата. Проверено численно (полоса 50 кПа, берма 30 м, гребень 31 м, Z=6 м, песок φ=30°): старое правило давало 0,00 кН/м, новое — 83,33 против теоретических Ka·q·h = 83,33 (отношение 1,0000). На проде эффекта нет и не ожидается: эндпойнт всегда передаёт в призму surcharges=[] (с [0183] пригрузка идёт отдельной полосой СП 22 — см. бюллет выше), аудитом подтверждено на всех 305 вызовах призмы в полном прогоне. Правка готовит функцию к будущему прямому подключению пригрузок к самой призме.

> **Двойной учёт полосы в зажиме p0 устранён ([0215]/[0222]; пионерная часть относится к удалённой 22.10.2026 призме)**
>
> До правки зажим p0_back = max(σa_back, min(p0_back_raw, σp_back)) сравнивал p0_back_raw БЕЗ вклада полосы с σa_back, который полосу уже содержал (Ka·q, [0183]) — зажим почти всегда срабатывал и поднимал давление покоя до активного предела, внося ту же полосовую нагрузку второй раз, а K0·q поверх этого — в третий.
>
> Теперь вклад полосы входит в p0_back_raw = K0·(σ′v + q_полосы) + u ДО зажима, и зажим сравнивает сопоставимые величины. На эталоне 1655 (однородный грунт, без выемки и связей) зажим перестаёт срабатывать вовсе: p0_back_raw = 121,47 против σa_back = 93,51; приложенная асимметрия выходит намеренные 48,46 кПа вместо 68,96.
>
> Пол минимального активного у пионерной ветви брался как доля от p0_back и после первой правки начал расти вместе с полосой, хотя полоса уже входит в σa_back отдельным приращением — переведён на бытовое K0·σ′v + u без полосы (основание, на котором калибровалась константа 0,2 в [0166]).
>
> Потолок σa на шаге призмы Кульмана (min(..., p0_back)) сравнивался с p0_back, который теперь содержит полосу, хотя сама призма считается без пригрузок ([0183]) — переведён на p0_back без полосы; расхождение поймал существующий тест test_pioneer_strip_surcharge_enters_active_limit_in_band.
>
> Реализация всех трёх поправок — build_two_sided_spring_laws (core/fem/elasto_plastic.py). Замер на 56 эталонах: M 47→49, Q 48→49, Км 47→49, все пять метрик одновременно 40→41 из 56.

Коэффициенты запаса

- **Металл:** K = R_y·W/M, R_y = R_yn/γ_m, γ_m = 1,05 (СП 16.13330, табл. 3; GAMMA_M_STEEL в staging.py).
- **Заделка:** дефолт — мобилизованный резерв F_lim/F; режим «auto» (предельное равновесие при d/H > 1,4) — только по явному выбору. C [0382] (15.10.2026) основной путь консоли без крепления — зона выше точки вращения (staging._cantilever_zone_K_direct): F_lim/F по отрезку заделки от дна котлована до первой смены знака перемещения, без ворот 0,05/0,10; узел на границе входит дробной долей своего участка, поэтому Kз непрерывно зависит от ϕ и от шага сетки. Зона действует только при ровно одной прямой смене знака w (выше точки к котловану, ниже в тыл); без смены знака, при обратном повороте и при двух и более нулях путь прежний (min из резерва и моментного баланса, который не применяется, если носок отыгрывает меньше 5 % максимального прогиба или точка вращения ближе 10 % длины заделки к носку — иначе он срабатывает на численном нуле). Известные ограничения: обратный поворот (отчёты 1998.1, 1655) не решён ни зоной, ни прежним путём; при точке вращения ближе ≈ 10 % заделки к носку зона занижает Kз на 2–10 % относительно печати эталонной программы; у 1667.9 (две пары) и у бермы L2.4 Kз вышел из ±10 % на 0,3–0,9 п.п.; на расчётах «берма» и «срез» поле перемещений решения неадекватно, Kз по зоне им не доверяет. Границы зоны по точке вращения в документации эталонной программы нет — это допущение ядра, подтверждённое опытами (лестница: худшее 4,8 %; партия 7 вслепую: 6 из 6 в ±4,2 %; одно грунтовое семейство, четыре информативные точки).

embedment.py — глубина заделки (бисекция)

find_cantilever_embedment и find_propped_embedment. Для пробной d строятся две эпюры: активная на всю высоту h + d (zero-tension, вода с обеих сторон) и пассивная ниже дна (локальные координаты 0..d, без zero-tension, лицо сухое). Уравнение ΣM = 0 решается методом бисекции: границы d_low = 0,1, d_high = 3h (до 5h), невязка f = Ma − Mp — точные интегралы по кусочно-линейным эпюрам (_WallStatics.moments_about), остановка по ширине интервала d_high − d_low ⩽ 1e-6 м или по относительной невязке |f| ⩽ 1e-10·(|Ma| + |Mp|), но не меньше 1e-9 кН·м/м, до 100 итераций (с [0573]; до этого — трапеция по узлам эпюры и абсолютный допуск |f| < 0,1 кН·м/м). Внутренние усилия строятся при d_min (запас — только в геометрии d_design).

> **Q_max внутренний против R_toe у носка**
>
> Реакция заделки моделируется требованием ΣM(toe) = 0 — это создаёт в эпюре Q(z) фейковый пик у носка (концентрация R_toe), а не напряжение среза металла.
>
> Q_max (для СП 16): у консоли — |Q| только до последнего перехода через ноль перед носком, хвост у носка — реакция заделки R_toe = Q_at_toe = |Q[-1]|, она идёт в диагностику; у схемы с опорой — max|Q| по всей стенке, нижняя ветвь ниже дна котлована входит (_compute_forces_propped, с [0551-блюм-сетка-и-q-internal]; прежде эпюру резали по последнему нулю Q, и ветвь входила или нет по знаку численной невязки у пяты). Q_at_toe распорки — |Q(L)| = 0 (свободное опирание; прежде шум трапеции 0,0–0,7 кН/м). Без разделения у консоли проверка среза завышалась бы ~в 3,5 раза (исправлено в [0022]).

internal_forces.py и steel_check.py

Нетто-давление p_net(z) = σa_total(z) при z ⩽ h и σa_total − σp_total(z−h) ниже дна; Q(z) = ∫p_net, M(z) = ∫Q — точные многочлены на участках между узлами эпюр (класс _WallStatics в embedment.py, с [0551]; прежде — трапеции на равномерной сетке). compute_internal_forces (эпюры показа) берёт значения из того же _WallStatics.sample: узлы — глубины, кратные шагу 0,1 м, нижняя точка стенки, дно котлована, скачки давления (границы слоёв, УГВ) и тройка ±0,001 м вокруг распорки; узлы ближе 5 мкм — один, не меньше 10 узлов; в узлах Q и M точные, между узлами график — линейная интерполяция. Проверки СП 16 (check_profile) — пять проверок с коэффициентами γc, γn и учётом коррозии:

Проверка	Условие
Изгиб	σ = M/W ⩽ Ry·γc
Срез	τ = Q/A_web ⩽ Rs·γc, Rs = 0,58·Ry
Совместное M+Q	σ_eq = √(σ² + 3τ²) ⩽ 1,15·Ry·γc
Устойчивость стенки	λ_w = h_w/t_w ⩽ 3,5·√(E/Ry)
Устойчивость полки	λ_f = b/t ⩽ 0,5·√(E/Ry)

Глава 6. МКЭ-ядро: балка на упругопластическом основании

Новое расчётное ядро (core/fem/), совместимое с промышленным аналогом на уровне физической модели (МКЭ + основание Винклера). Задача [0023], пять этапов.

6.1. Физическая модель

Ограждение котлована моделируется как вертикальная балка Эйлера–Бернулли, опёртая на грунт, который ведёт себя как нелинейное основание Винклера (распределённые пружины с предельными давлениями) и, опционально, на дискретные крепления — анкеры и распорки. Искомая величина — поле горизонтальных перемещений стенки w(z), из которого постпроцессингом получаются эпюры момента M(z), поперечной силы Q(z), давления грунта p(z) и реакции креплений.

Каждый узел несёт три степени свободы: u (продольное), w (поперечное), θ = w′ (поворот); двухузловой элемент — 6 DOF. Поперечное поле w(x) аппроксимируется эрмитовыми кубическими функциями формы (C^1-непрерывность), продольное u(x) — линейными. Основание Винклера двустороннее: пружина «back» (тыл) активна на всей высоте стенки; пружина «front» (лицо) — только ниже дна котлована.

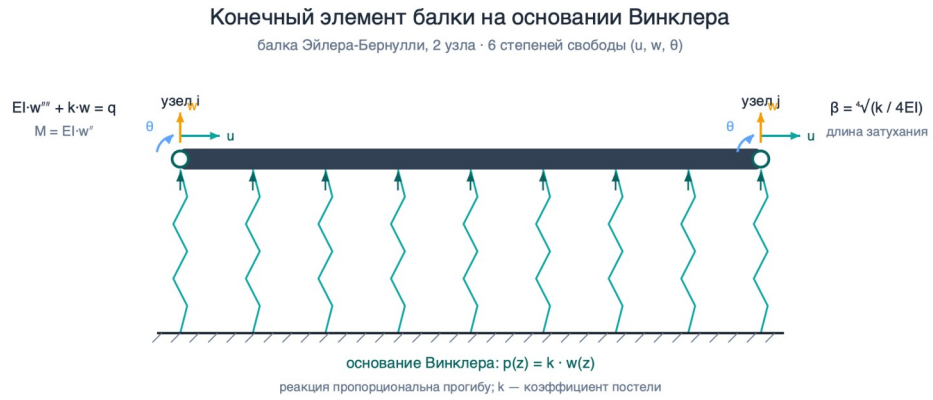


Рис. 2. *Конечный элемент балки на основании Винклера: 2 узла, 6 степеней свободы (u, w, θ), распределённые пружины грунта.*

> **Знаковая конвенция (единая по всему ядру)**
>
> z — глубина от верха стенки, ось вниз положительна;
>
> w > 0 — смещение стенки в сторону удерживаемого грунта (от котлована); w < 0 — в котлован;
>
> момент M = +EI·w″ (растяжение со стороны котлована при M > 0), со ссылкой на СП 16, Зенкевича и промышленный аналог.

6.2. Определяющие уравнения

Дифференциальное уравнение балки на упругом основании (гипотеза Винклера — реакция пропорциональна прогибу):

$$EI\frac{d^4w}{dz^4} + k\,w(z) = q(z)$$

$$p(z) = k\,w(z)$$

где k_s — коэффициент постели (кН/м³, на единицу ширины стенки), q(z) — распределённая поперечная нагрузка. Характеристическая длина (масштаб затухания):

$$\beta = \sqrt[4]{\frac{k}{4EI}}$$

Постпроцессинг усилий — через производные функций формы:

$$M = EI\frac{d^2w}{dz^2}$$

Активный и пассивный пределы (эффективные, Ренкин или Кулон), зона растяжения обнуляется:

$$\sigma_a = K_a\,\sigma'_v - 2c\sqrt{K_a} + K_a\,q$$

$$\sigma_p = K_p\,\sigma'_v + 2c\sqrt{K_p}$$

Двусторонний билинейный закон p–y: пружина back активна всегда, front — только ниже дна котлована; нетто-давление net(w) = p_back − p_front.

6.3. Матрицы конечного элемента

Матрица жёсткости изгиба (4×4, степени свободы w_1, θ_1, w_2, θ_2):

$$K_{bend} = \frac{EI}{L^3}\cdot\begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix}$$

Осевой вклад — EA/L по степени свободы u. Consistent-матрица Винклера (согласованное интегрирование по эрмитовым функциям формы, точнее lumped-пружин для упругой задачи):

$$K_{wink} = \frac{k_s \cdot L}{420}\cdot\begin{bmatrix} 156 & 22L & 54 & -13L \\ 22L & 4L^2 & 13L & -3L^2 \\ 54 & 13L & 156 & -22L \\ -13L & -3L^2 & -22L & 4L^2 \end{bmatrix}$$

Глобальная сборка — разреженная (scipy.sparse); граничные условия — методом удаления степеней свободы.

6.4. Аналитическая верификация Hetényi

Точное решение бесконечной балки на упругом основании под точечной силой P в конвенции проекта:

$$w(z) = \frac{P\beta}{2k}\,e^{-\beta|z|}(\cos\beta z + \sin\beta z)$$

$$M(z) = -\frac{P}{4\beta}\, e^{-\beta|z|}(\cos\beta z - \sin\beta z)$$

$$Q(z) = +\frac{P}{2}\, e^{-\beta|z|}\cos\beta z \cdot \operatorname{sign}(z)$$

Знак Q(0+) = +P/2 разрешён через strong-form вывод EI·w′′′′ + k·w = P·δ(z) и подтверждён sympy. Контрольные числа точности:

Тест	Что проверяется	Результат
W1	Длинная балка под P, прогиб против Hetényi	rel.err = $1{,}85\cdot10^{-5}$
W3	Порядок сходимости по прогибу w	$O(h^4)$, ratio = 16,00
W3	Порядок сходимости по моменту M	$O(h^2)$, ratio = 4,00
W3	Порядок сходимости по скачку Q	$O(h^1)$, ratio = 2,00
W4	Короткая балка β·L = 0,775 против FD-референса	max rel.err = $2{,}1\cdot10^{-4}$

6.5. Порядки сходимости и суперсходимость

Этап 1 (beam_solver.py): T1 (линейный момент) — машинная точность; T2 (квадратичный момент, полином 4-й степени по w) — узловая суперсходимость по w ~ $5\cdot10^{-13}$, порядок $O(h^2)$ по M, ratio = 4,00; T3 (осевое) — машинная точность.

Этап 2 (winkler_foundation.py): consistent integration через эрмитовы функции формы (не lumped-springs Bowles); backward compat с Этапом 1 сохранён; физические тесты W1–W4 — см. раздел 6.4.

6.6. Метод начальных напряжений и стадийность

Решатель по умолчанию с 05.10.2026 — прямой Ньютон (contact_strategy="newton", раздел 6.13). Метод начальных напряжений (Пикар, contact_strategy="initial_stress") больше не базовый и удалён из ядра 15.10.2026 ([0377]; вместе с penalty и active_set — на умолчании продукта они не вызывались, числа не изменились): запрос с таким значением поля — 422 ([0412-строгий-вход]); метод описан здесь как прежний решатель по умолчанию, на котором получены сверка с эталоном 1949.3 и базы храповика до 05.10.2026. Это метод с lumped-пружинами и ε-континуацией. Винклер элемента = 0; пружина грунта сосредоточена в узле. В сошедшемся состоянии lumped-слагаемое Cn·w сокращается, но страховочная пружина ε последнего уровня лестницы (1e-4) остаётся в каждом узле, поэтому неподвижная точка — не точное равновесие, а равновесие с этой пружиной (замер — раздел 6.13):

```
K_балки · w + ε·k_ref·trib · w = − net(w) · l      (точное равновесие — при ε → 0)
```

На пластичном узле net постоянно ⇒ вклад в жёсткость ровно ноль. Это снимает обе крайности: «corrective» при полной consistent-K давал решение ~6× жёстче, а чистый билинейный Винклер с занулением (β=0) даёт механизм. ε-континуация (исчезающая регуляризация) идёт по расписанию [1, 0.3, 0.1, …, 10^{-4}] с тёплым стартом.

Двусторонняя p–y (независимая классификация back × front), тёплый старт между этапами, огибающая усилий max|·| по всем этапам. Преддеформация w_0 в момент монтажа крепления фиксируется физично — критично для правильных усилий в анкерах/распорках.

> **Сверка с эталонной программой (эталон 1949.3)**
>
> Реальный объект: анкерное ограждение ECO 618, 3 этапа отрывки.
>
> Главные метрики в инженерных допусках: M −12,5 % / w +9 % / R −13 % / K_заделки −21 % (допуски ±15 / 20 / 15 / 25 %).
>
> Метод начальных напряжений снял ранее наблюдавшийся остаток по R/w/K одной формулировкой — за счёт драйва из полного баланса давлений. Числа этой врезки получены на решателе Пикара; на Ньютоне эталон 1949.3 в этой книге не пересчитывался.

6.7. Ключевые источники

Источник	Роль в ядре
Hetényi M., «Beams on Elastic Foundation» (1946)	Эталонные аналитические решения бесконечной балки (сверка W1–W4).
Bowles J., «Foundation Analysis and Design» §13	Коэффициент постели, схема active-set + modified Newton-Raphson.
Zienkiewicz O., «The Finite Element Method»	Эрмитова аппроксимация, consistent-интегрирование, знаковая конвенция.
СП 16.13330	Проверка металла K_металл = σ_yield·W/|M_max|; конвенция момента.
СП 22.13330 / СП 381.1325800	p–y законы, эпюры давления. Запас в заделке Kз — калибровочная величина программы, допустимое значение — ориентир, не норматив.

6.8. Реализация: сборка и решение (beam_solver.py)

Класс BeamElement2D6DOF строит локальную матрицу 6×6: осевой блок EA/L и изгибный блок $EI/L^3 \cdot [12, 6L, 4L^2, 2L^2]$ (декуплированы). При k_winkler > 0 ленивым импортом добавляется local_winkler_stiffness_6x6(k, L) — K_local = K_elastic + K_winkler. Глобальная сборка (assemble_stiffness): суммирование локальных матриц по шаблону CSR с кэшем (раздел 6.12); ячейки с нулевым значением в матрице отсутствуют.

Решение solve_beam: граничные условия — «Способом B» (удаление DOF): free_dofs = все \ fixed, редукция K[free,:][:,free], u = spsolve(K_reduced, F_reduced), восстановление полного вектора. Постпроцессинг усилий — через производные Эрмитовых функций формы: M(s) = EI·w″(s) (линейна внутри элемента, разрыв на стыке), Q = EI·w‴ (постоянна в элементе), N = EA·$(u_2 - u_1)$/L.

6.9. Реализация: двусторонний закон p–y (SoilSpringLaw)

Каждый узел несёт до двух билинейных пружин. Back (тыл) активна всегда; front (лицо) — только ниже дна котлована (z ⩾ H_котл), задаётся all-or-nothing. Закон давления compute_pressure(w):

```
p_back  = clip(p0_back + k_back·w,  σa_back, σp_back)
```

```
p_front = clip(p0_front − k_front·w, 0,       σp_front)   # или 0 если
односторонний
net     = p_back − p_front
```

При +w (стенка в тыл) back-грунт сжимается к пассивному пределу, лицо «теряется» к нулю; при −w — наоборот. Опциональная асимметричная мобилизация (mobilization="sp_fig91", СП 22 рис. 9.1) задаёт разный наклон на активной и пассивной ветвях; по умолчанию — единый k. Драйв-нагрузка: f(z) = −σa_back(z) на свободной части 0 ⩽ z ⩽ H_котл, ниже — 0.

6.10. Реализация: метод начальных напряжений (Пикар)

Решатель Пикара _solve_two_sided_initial_stress (contact_strategy="initial_stress"; до 05.10.2026 — решатель по умолчанию; удалён в [0377]). Винклер элемента = 0, вся пружина грунта сосредоточена (lumped) в узле, нелинейность отсекается корректирующей узловой силой, ε-континуация убирает жёсткое тело до мобилизации заделки. Для узла i с локальным тангентом Cn = elastic_stiffness(w) (метод удалён вместе с решателем):

```
k_lump = (Cn + ε·k_ref)·trib_i          # узловая пружина
f_i    = (Cn·w_i − net_i(w))·trib_i     # корректирующая сила
k_ref  = median(Cn > 0)                 # опорная жёсткость для ε
```

В сошедшемся состоянии слагаемые Cn·w слева и справа сокращаются, но пружина ε·k_ref·trib последнего уровня лестницы остаётся в системе, поэтому получается равновесие K_балки·w + ε·k_ref·trib·w = −net(w)·l, а не точное K_балки·w = −net(w)·l; на пластичном узле вклад в жёсткость, кроме этой пружины, равен нулю. ε-континуация идёт по расписанию (1, 0.3, 0.1, …, 1e-4) с тёплым стартом; внутри уровня — итерации Пикара с демпфированием w_new = 0,5·w_new + 0,5·w_cur.

> **Предохранители near-limit**
>
> Гомотопический откат: снапшот последнего физичного решения (|w| < 0,1·пролёт); если очередной ε уходит в нефизичный прогиб — откат к нему и стоп.
>
> Мягкий критерий: если max_iter исчерпан, но относительный Δw < 1e-3 и прогиб физичен — расчёт принимается (нелинейные узлы «дребезжат»).
>
> Страж абсурда: при |w|_max > 10·пролёт или не-finite — принудительно converged = False.
>
> Альтернативы: penalty (soft-contact, остаточная жёсткость β = 0,17) и active_set (ККТ, 4 состояния back×front) — удалены в [0377] вместе с β по слою.

6.11. Реализация: стадийность и коэффициенты запаса (staging.py)

solve_staged последовательно вызывает _solve_single_stage, передавая предыдущий снапшот как тёплый старт. На каждом этапе spring_laws пересобираются под текущую H_exc (снятие отпора при углублении — автоматически через геометрию, front-пружина исчезает для обнажённой зоны). Механизмы:

- **Тёплый старт** переносит w и состояния пружин из предыдущего этапа; маппинг — по node_idx.
- **Преддеформация при монтаже:** анкер/распорка фиксирует w0 = w[node] на старте своего этапа и реагирует на приращение прогиба, а не на абсолютное.

- **Огибающая** — поэлементный max|·| усилий по всем снапшотам (M_max, Q_max, w_max; K_заделки — минимум по этапам).

> **Q_max внутри одного этапа: поэлементно, не по узловому среднему ([0216])**
>
> Постпроцессинг даёт Q(s) кусочно-постоянной ВНУТРИ конечного элемента (Q = EI·w‴); shear_diagram для отрисовки усредняет два смежных элемента в узле — так рисует и промышленный аналог, и это даёт гладкую с виду кривую.
>
> До [0216] отчётная величина Q_max бралась как максимум именно по этому узловому среднему. Среднее двух элементных значений по модулю не больше большего из них ⟹ занижение гарантировано математически, а в узле со скачком (точечная сила связи, пластический узел) среднее ложится ровно посередине скачка.
>
> Эталон сам печатает поэлементную константу и предупреждает об этом (методика, §9.1: «перерезывающая сила вычисляется в центре конечного элемента»). Теперь Q_max = max по новому полю shear_max_element (максимум |Q| по конечным элементам, без узлового усреднения); shear_diagram для графика не менялась.
>
> Решателя правка не касается — M, w, K_заделки бит-в-бит; K_металл берёт M и N, а не Q, поэтому тоже не меняется. Q может только вырасти, поэтому ранее заниженные отчёты подтягиваются к нулю ошибки, а не уходят за него: на 56 эталонах Q 46→48/56, вышедших нет.
>
> Не путать с Q_at_toe / опорной реакцией заделки R_toe метода Блюма (см. предупреждение в §5.2) — тот же класс путаницы уже приводил к завышению проверки металла ×3,5 в [0022], но здесь речь о другом движке (МКЭ) и другой паре величин.

Дополнительные входы этапа (dataclass Stage), переопределяющие проектные дефолты только на конкретном этапе:

- `water_back_depth, water_front_depth` — абсолютные уровни воды тыла/лица НА ЭТАПЕ, м от верха стенки. None (дефолт) — берётся проектное значение soil["water_table_back/front"], как было до поэтапной воды.
- `backfill_depth` — Optional[float], ge=0, le=50 ([0194]). Глубина ПОВЕРХНОСТИ ГРУНТА ТЫЛА от верха стенки на этапе, м — аналог колонки «ИГЭ подсыпки» таблицы этапов эталонной программы. None/0.0 (дефолт) — тыл отсыпан до проектной отметки, поведение байт-в-байт с версией без подсыпки. > 0 — тыл ещё НЕ досыпан: выше этой отметки грунта за стенкой нет — ни веса, ни давления, ни пружины. Датум неподвижен и равен ПРОЕКТНОЙ (конечной) поверхности тыла: по этапам меняется только сама засыпка, а сетка, длина стенки и колонка грунта остаются теми же объектами (важно для тёплого старта). Валидация в __post_init__: backfill_depth > H_exc + 1e-9 → ValueError (тыл не может лежать ниже дна выемки).
- **Флаг затопления теперь ВЫВОДИТСЯ из уровней этапа, а не задаётся отдельно** ([0197], FLOODED_LEVEL_TOL = 0,05 м). Когда у этапа заданы ОБА уровня:

```
flooded = (|water_back_depth − water_front_depth| ≤ 0,05)  И  (water_back_depth <
H_exc − 0,1)
```

flooded=True ⟹ гидростатика тыла/лица уравновешена, нетто порового давления на стенку — ноль. Проектный soil["flooded"] остаётся ТОЛЬКО как совместимость — работает там, где у этапа нет пары уровней (узкий формат отчёта, ручной ввод без поэтапной воды). До [0197] флаг был один на весь расчёт и брался с ПОСЛЕДНЕГО этапа: там, где баланс воды поэтапный (например, этап 1 сбалансирован, этапы 2-3 нет), это зануляло реальный двухметровый перепад и занижало момент на 73-82 %. Порог 0,05 обоснован данными: из 136 пар уровней в эталонном

наборе 58 совпадают точно, наименьшая ненулевая разница — 0,100 м (1999.2/1999.3); 0,05 лежит посередине с двукратным запасом в обе стороны.

Коэффициенты запаса:

```
K_заделки  = F_lim / F,   F_lim = Σ(σp_front − σ₀)·L,  F = Σ(σx − σ₀)·L
K_металл   = σ_yield·W / |M_max|              (N = 0, СП 16 ф. 41)
K_анкер    = F_yield / |R_anchor|,   K_распорка = F_buckling / |R_strut|
```

Для консоли (нет верхнего крепления) резервная формула K_заделки на near-limit повороте переоценивает запас, поэтому с [0382] при одной прямой смене знака перемещения берётся F_lim/F по зоне выше точки вращения (непрерывная зона, дробное участие узла), а в остальных случаях — min(K_reserve, K_rotation), моментный баланс относительно точки вращения. Крепления входят в то же равновесие Ньютона кусочно-линейными пружинами (внешнего цикла по состояниям связей после [0377] нет): анкер — идеально-пластичная пружина растяжения $R \in [0, F_{yield}]$; горизонтальная распорка — по умолчанию двусторонняя (сжатие до F_buckling и растяжение до −F_buckling, [0386]), а при strut_tension = false — только сжатие $R \in [0, F_{buckling}]$ с потерей контакта ($R \leqslant 0 \Rightarrow$ выпадает).

Подэтап натяжения анкера «k.F0» ([0383-F0-подэтап-и-Км-ф105], 15.10.2026; staging._f0_plan_stage). Анкер с F0 > 0 на этапе k раньше ставился пружиной R = F0 − C·(w − w0) в одном равновесии с откопкой этапа: стенка отходила в тыл, и усилие релаксировало до 0,3–0,6·F0 (M этапа занижен на 10–42 %, Q на 10–51 %, Км завышен до ×2 — небезопасно). Теперь, по схеме программы-эталона (лестница опытов, партия 5, 8, 9), сначала решается подэтап «k.F0» на данных ПРЕДЫДУЩЕГО этапа (дно, вода, подсыпка, нагрузки, действующие связи этапа k−1), а новый анкер в нём — внешняя сила F0 (жёсткости ещё нет; усилие анкера на подэтапе = F0 точно); перемещения подэтапа фиксируются, и на этапе k анкер — пружина от этого состояния. Допущения: A1 — состав подэтапа (копия этапа k−1; прочие крепления этапа k на подэтапе не стоят); A2 — анкер на этапе 1: подэтап «1.F0» на исходном состоянии (без откопки и нагрузок, вода на уровне тыла); A3 — память грунта (с 23.10.2026, [0426-память-грунта-на-всех-этапах], память на ВСЕХ этапах, раньше только с первого этапа с F0): этап 1 — накопление пластических смещений («track»), все следующие этапы и подэтап — с памятью («apply»); при f0_substage = false память включается только с первой подсыпки лица (механизм гипотезы V8-01, elasto_plastic.apply_plastic_offsets / update_plastic_offsets, возвратное отображение; руководство 8.4, контрольная задача 9.5.3). Без памяти отклик на натяжение (разворот перемещений после откопки) не тот. В ответе подэтап — отдельная запись stages с полем substage = «F0» и label «<метка>.F0» перед записью этапа k; огибающая M, Q, w, Кз, Км и усилий связей берётся по всем записям, включая подэтапы; support_checks и governing_stage — по этапам. Расчёт без анкеров с F0 > 0 схема не трогает (побитно). Без подэтапа — f0_substage = false в запросе. Замер: лестница опытов (4 расчёта L5 — M этапа +0,4…+5,1 %, Q −0,2…+4,3 %, усилие анкера +0,6…+5,4 %), пример 2 руководства (три яруса, F0 = 100 кН) — ХУЖЕ прежнего на этапе 4 (M −14 %, верхний анкер +95 %: верхний анкер после второго натяжения почти не разгружается), так что многоярусные схемы с преднатягом схема описывает хуже, чем одиночный анкер. Не доказано: вне семьи лестницы (один шпунт, один анкер, 30° / 45°) подтверждения нет; проект вне выборки с наклонным анкером и F0 > 0 ожидается (DemoExample).

Распорка работает на растяжение ([0386-распорка-на-растяжение], 15.10.2026; StrutFEM.tension_capable, staging._installed_to_nodal_spring / _compute_reaction). Горизонтальная

распорка между стенками (struts[], inclination_angle = 0) по умолчанию — двусторонняя пружина R ∈ [−max_force, +max_force]: потери контакта нет, реакция линейна по смещению до предела в обе стороны, состояние YIELD_MIN — предел растяжения (−F_buckling), в диагностике узла R_final = F_min. Основание — опыт L6.5 лестницы опытов, этап 3: верхняя распорка в окне результатов программы-эталона показана стрелкой 13 кН против нижней (78 кН, сжатие), то есть растянута; односторонний закон (R ⩾ 0) давал M +20 %, Q −15 %, Kм −19 %, двусторонний — M 34,1 / Q 52,3 / Kм 26,32 / Kз 3,37 против печатных 34,7 / 53 / 26,57 / 3,37. Решение односторонней задачи единственно (Ньютон, выпуклый функционал; восемь стартов сходятся к одному w с точностью $1 \cdot 10^{-13}$ м), поэтому «ложной потери контакта» нет: различие — в законе связи. Допущения: (1) предел растяжения был принят симметричным пределу сжатия (max_force) без источника — СНЯТ в [0393-поправки-по-результатам-1]: опыт L11.14 (Fm 5 кН при усилии 16,9 и 12,7 кН, все числа равны опыту L6.5 с Fm 7305 кН, «K зап.» 0,30 и 0,39) показал, что max_force горизонтальной распорки, работающей на растяжение, усилие не ограничивает, а служит базой запаса K = Fm/|F| (StrutFEM.force_limited = False; K < 1 допустим: замечание этапа «распорка … перегружена», состояние связи exceeded в support_checks, ok = false; раскос, грунтовая распорка, анкер и режим strut_tension = false предел сохраняют; прежнее — strut_force_limit = true); (2) растяжение несёт только распорка, закреплённая к обвязочному поясу, — если она лишь упирается в стенку, двусторонняя связь завышает поддержку верха стенки (в L6.5 момент ниже на 18 %); для такой распорки задайте `strut_tension = false`; в интерфейсе — поле проекта «Распорка работает на растяжение» (wall.strutTension, вкладка «Анкеры» → «Распорки», по умолчанию отмечено; [0394-дефекты-скептика-1]); предел растяжения — допущение, опыт, определяющий его, не проводился; (3) раскос (inclination_angle ≠ 0, опирается на дно) и грунтовая распорка растяжения не несут при любом значении поля. Ответ: у растянутой распорки строка усилия несёт signed_force_kN и signed_force_axial_kN (растяжение +, сжатие −: соглашение строительной механики с [0512-знак-усилий-связей], до того было сжатие +, растяжение −; знак меняется только в этих полях) и tension = true — модуль force_kN прежний, старые потребители не затронуты; в огибающей строка берётся по этапу с наибольшим усилием; в warnings этапа — замечание о растяжении (⩾ 0,05 кН/м). Работа только на сжатие — поле запроса strut_tension = false; ответ в этом режиме без новых ключей.

Продольная сила в стенке (N, `axial_model`, [0195]). Осевая задача прямой балки Эйлера-Бернулли развязана с изгибной — осевой блок EA/L локальной матрицы не связан ни с прогибом, ни с поворотом, поэтому N вычисляется отдельно (_compute_axial_force) и НЕ меняет M, Q, w. Единственный потребитель — проверка металла: при N ≠ 0 вместо ф. 41 считается СП 16.13330 п. 9.1.1 (ф. 105, прочность при продольной силе с изгибом, n = cx = 1, γc = 1, как печатает эталон):

```
N ≠ 0:  K_металл = Ry / (|N|/A + |M_max|/W)   (СП 16.13330 п. 9.1.1, ф. 105)
N = 0:  K_металл = Ry·W / |M_max|             (п. 8.2.1, ф. 41 — прежняя ветка,
байт-в-байт)
```

- `wall["axial_model"]` не задан (FemCalculateRequest.axial_model = None) — модель `"forces"` (staging.DEFAULT_AXIAL_MODEL); модель `"none"` (Км по ф. 41 СП 16) задаётся явно.

- `"forces"` (дефолт с [0383]) — только внешние вертикальные силы: вертикальные составляющие усилий наклонных анкеров и раскосов (Py = P·sin α) и вертикальные проекции сосредоточенных нагрузок на стенку, БЕЗ собственного веса. Это приближение к осевой силе N0, которую печатает отчёт 8.4 (Приложение Б), а не её воспроизведение: по замеру на печатных N0 модель «forces» занижает |N| примерно на 10 %, «full» (с весом) завышает примерно на 16 %, остаток не объяснён (в том числе N0 кантилеверов −2…−4 кН/м; константой не вводится). Итоговый Км по ф. 105 на 22 анкерных отчётах лестницы опытов с такой N — в ±4,3 % печатного (медиана 1,3 %), с «full» — в ±7,9 % (медиана 3,5 %, систематически ниже печати), без N — до +17,4 % (медиана +9,4 %, небезопасно); ошибка |N| на 10 % сдвигает Км примерно на 1–2 %. Нет внешних вертикальных сил → None, ответ и Км побитно как при «none».
- `"none"` — N ≡ None, Км по ф. 41 (прежнее умолчание; явное значение в запросе возвращает его побитно).
- `"self_weight"` — только собственный вес стенки: погонная осевая нагрузка γ_steel·A (STEEL_UNIT_WEIGHT = 78,5 кН/м³, СП 16.13330/СП 20.13330), накапливается вниз как сжатие; опора по вертикали — носок.
- `"full"` — плюс вертикальные составляющие наклонных анкеров/распорок (Py = P·sin α, §2.3 методики эталона) и вертикальные проекции сосредоточенных нагрузок на стенку (`FemPointLoadInput.fy`, кН на шаг, + вниз).

Ответ несёт `axial_force` (эпюра N(z) по узлам финального этапа, List[DiagramPoint], сжатие < 0; пусто — продольной силы нет) и `N_governing` (N в том же сечении, где найден max|M|, этапа с наибольшим моментом). Усилия связей `anchor_forces / strut_forces` — МОДУЛЬ; у горизонтальной распорки, работающей на растяжение ([0386]), строка дополнена знаком (signed_force_kN, signed_force_axial_kN: растяжение +, сжатие −; tension). Без этих полей анкер — растяжение (+), распорка — сжатие (−). Разведка на 11 эталонных отчётах с полной печатной таблицей СП 16 (N0, M0, Qx0, A, Wnmin и все четыре коэффициента): ф. 105 всегда определяющая — приведённые напряжения (ф. 44) и касательные (ф. 42) не были определяющими ни разу; доля продольной силы в загружении 0,2–1,8 % на 9 отчётах из 10 и 44,1 % на одном с вырожденным изгибом (M ≈ 0). Собственный вес стенки как источник печатного N эталона — ОПРОВЕРГНУТ его же печатью (не сходится по величине); источник остаётся неизвестным. Итог разведки [0195]: инфраструктура заведена и измерена, дефолт был axial_model="none". С [0383] дефолт — "forces" (замер на лестнице опытов — выше; решение владельца 15.10.2026, п. 3.2 «подэтап k.F0 + осевая сила в Км, парой»).

6.12. Ускоренная сборка K и кэш ([0293-C1])

Замер на реальных входах показал, что 93–98 % времени итерации Пикара занимала Python-сборка матрицы жёсткости, которая от итерации не зависит (2,8 / 9,3 / 17,4 мс против 0,2–0,3 мс решения на 41 / 121 / 241 узлах). Теперь шаблон CSR и упругие матрицы элементов кэшируются по сетке и сечению (beam_solver: chain_mesh, SparseSystem); на итерации добавляются Винклер (векторно, по элементу) и узловые пружины; оркестратор собирает систему без lil_matrix, узловые члены initial_stress считаются векторно, а призма пионерного среза кэшируется по точным входам (_Pa_at_Z). Числа побитово прежние: проверено на 20 реальных расчётах и тестами; ускорение ×11 (CLAUDE.md §11), полный прогон набора — около 4 минут вместо 30.

- `scripts/bench/solver_speed.py` записывает выходы ядра без округления и времени и побитово сверяет повторный прогон с записью (доли сборки и spsolve). Переключатель `FAST_ASSEMBLY` и сборка через lil_matrix удалены ([0487-чистка-кода]); свойства сборки — совпадение с плотной сборкой малых случаев, симметрия, отсутствие нулевых ячеек, равенство систем chain_mesh и произвольной сетки — проверяет `tests/test_fem_assembly_cache.py`.
- `solve_elasto_plastic(reset_bond_states=False, outer_warm_start=False)` — opt-in: связи этапа стартуют в состоянии ELASTIC / внешняя итерация идёт от предыдущей внешней; по умолчанию выключено, результат прежний (убирает «холостые механизмы» внешнего цикла связей).

6.13. Прямой Ньютон и детектор механизма ([0293-C2])

`core/fem/newton_continuation.py` — `contact_strategy="newton"` (основной режим контакта с 05.10.2026, [0297]; до этого — opt-in). Решается та же дискретная задача R(U) = K·U + T·net(w) − S(w) − F = 0 полугладким методом Ньютона с линейным поиском по энергии (функционал выпуклый, равновесие единственно); клэмп связей встроен в S(w), поэтому внешний цикл по состояниям связей и «лок» после трёх переключений не нужны; поддержаны предписанные перемещения ($\bar{w} \neq 0$), прогрев от предыдущего этапа и асимметричная мобилизация. Регуляризация ε·k_ref финального уровня Пикара (1e-4) не вводится — решается точное уравнение.

Континуация по доле λ «нагрузки сверх покоя»: λ = 0 — стенка не смещена, λ = 1 — исходная задача. Сначала пробуется λ = 1; при неудаче шаг по λ адаптивный (удвоение / деление пополам); шаг меньше `NEWTON_DLAM_MIN` — механизм, `lambda_limit` = последний сошедшийся λ (оценка снизу). Это НЕ arc-length: путь по λ останавливается на предельной точке. λ_lim < 1 означает, что равновесия при действующей нагрузке нет: «нагрузка выше предельной на (1/λ − 1)·100 %». Исчерпанный бюджет уровней λ — численная несходимость, а не механизм. Глобальное равновесие проверяется по жёстким модам без K (у жёсткой стенки поузловой допуск тонет в округлении), шаг уточняется дефляцией по тем же модам.

Проекция жёстких мод ([0376], 15.10.2026). В float64 собранная K не аннулирует жёсткие моды точно ($K \cdot v \neq 0$ на ulp элементов $12 \cdot EI/h^3 \approx 10^{11}$), и вычисленное K·U несёт суммарную силу и момент порядка 10^{-4}…10^{-3}, которых у точного K·U быть не может; на отдельных шагах сетки (0,0125–0,025 м) невязка узла связи стояла выше допуска, и континуация печатала ложное «механизм». Теперь из вычисленного K·U вычитается проекция на жёсткие моды: $K \cdot U \leftarrow K \cdot U - W \cdot (V^T \cdot K \cdot U)$, $W = V \cdot (V^T V)^{-1}$; допуск, дефляция и проверка глобального равновесия по rest не менялись. Замер: на 684 парах «расчёт × шаг сетки» не сходились 48, теперь 0; ответ на ранее сходившихся парах сдвинулся не более чем на 0,0031 %; настоящие механизмы остаются механизмами с прежним λ_lim. Проекция действует всегда.

- ElastoPlasticSolution несёт `lambda_limit`, `mechanism`, `newton_iterations` (только у newton). API принимает `contact_strategy` только со значением newton (и без поля): поле устарело, ничего не меняет и интерфейсом не отправляется; initial_stress, penalty и active_set — 422 «Способ решения контакта: допустимо только «метод Ньютона»» ([0412-строгий-вход]; с [0377] до этого три прежних значения решались Ньютоном с пометкой в warnings). Выбора стратегии в интерфейсе нет, проект её не хранит: сохранённые проекты считаются

Ньютоном. Прогон МКЭ на контрольных задачах Блюма (страница /benchmarks/blum) идёт тем же решателем, что и продукт.

- Детектор механизма в этапном расчёте (диагностика, числа не меняет). StageResult несёт `mechanism_detected`, `lambda_limit`, `mechanism_message` и `mechanism_checked`: при несходимости основного решателя либо срабатывании near-limit предохранителя этап перепроверяется λ-тестом прямого Ньютона (≈0,3 с вместо 30–1372 с поиска равновесия, которого нет); на сходящихся этапах проверка не запускается. Перемещения, усилия, converged и warnings основного решения побитово прежние (тест). У стратегии newton поля заполняет само решение этапа. Они приходят в stages[] ответа API.
- Контрольные задачи EP-N1 (упругий режим ≡ Hetényi), EP-N2 (предельная нагрузка жёсткой сваи P_u = q_u·L·(√2 − 1), λ_lim = 0,8), EP-N3 (стенка с шарниром в голове, P_u = q_u·L²/(2a)); провайдеры — analytical_newton.py. Тесты: test_fem_newton_continuation.py, test_fem_newton_staging.py. Сверка «Пикар против Ньютона» на продуктовом пути — `scripts/prototypes/arclen/compare_newton_picard.py` (поэтапно w/M/Q/Kз/реакции; столбец «Ньютон с ε = 1e-4» показывает, что расхождение — смещение регуляризации Пикара).

Переключение на Ньютон по умолчанию: критерии и результат (05.10.2026, [0296-3], [0297])

Перед переключением владелец заранее задал четыре критерия; проверка выполнена на продуктовом пути (scripts/gw_compare/ratchet_check.py, scripts/gw_compare/extreme_embedment_check.py, результаты — scripts/gw_compare/results_core_0296_2026-10-05/). Итог первого прогона ([0296-3]): режим не переключили, потому что критерий (а) в исходной формулировке не выполнен. Затем замер невязки равновесия (документ ньютон_невязка_2026-10-05.md) объяснил сдвиг, владелец принял решение переключить, критерий (а) переформулирован и пройден на трёх наборах ([0297]): contact_strategy="newton" — режим контакта по умолчанию. Ниже таблица первого прогона, затем причина сдвига и результат по переформулированному критерию.

Критерий	Результат первого прогона ([0296-3])	Итог
(а) полный храповик на 56 расчётах (исходная формулировка): «все пять» не меньше 41, сумма выхода за 10 % не больше 1690,7 п.п., чистые пары не выходят из допуска	«Все пять» 41 → 41; чистые пары из допуска не вышли (0), пар в допуске 238 из 280 → 238 из 280; сумма выхода 1690,7 → 1692,0 п.п.; медиана |Δ| по Q 2,45 → 2,80 % при допуске гейта +0,2	не выполнен в исходной формулировке; после переформулировки выполнен (ниже)
(б) набор проектировщика не хуже (восстановленные 4/11 и настоящие 5/6)	Восстановленные: «все пять» 4 → 4, сумма 599,3 → 597,1, гейт пройден; настоящие: 6/7 → 6/7 (5/6 без дубля 1726.2)	выполнен
(в) крайние задачи: заделка 0,6–1,0 от исходной, 10–15 расчётов	18 расчётов × доли 1,0 / 0,8 / 0,6 / 0,45 × два режима = 144 прогона. В диапазоне 0,6–1,0 статус хуже у Ньютона — ни разу, лучше — дважды (2001.1 при 1,0: soft → ok; 22 при 0,6: mechanism → near_limit); расхождение Kз не больше 0,56 % (допуск 2 %). Вне диапазона, при 0,45, одно ухудшение: 1721.6 (1) near_limit → mechanism	выполнен в диапазоне; одно ухудшение вне его

Критерий	Результат первого прогона ([0296-3])	Итог
(г) время не растёт	На 56 расчётах суммарное время счёта 27 с против 3743 с (прежний контрольный замер при одновременном запуске — 97 против 435 с); на крайних — 3,7 против 138 с	выполнен

Что видно по парам в первом прогоне: сдвиг не больше 0,9 п.п. почти везде (M, Q, w в среднем вверх, Км вниз), вошедших в допуск и вышедших пар нет; выброс по прогибу — 1651.5, +1,9 п.п.

Причина сдвига: страховочная пружина Пикара ([0297], замер невязки)

Замер (64 расчёта, 168 этапов с нагрузкой; сырые результаты — scripts/gw_compare/results_newton_residual_2026-10-05/, инструмент — scripts/gw_compare/newton_residual_check.py) показал, что оба решателя решают одно дискретное уравнение R(U) = K·U + T·net(w) − S(w) − F = 0, но неподвижная точка Пикара — это уравнение плюс пружина ε·k_ref·T последнего уровня лестницы ε (1e-4), которая остаётся в каждом узле. Невязка равновесия, одна метрика у обоих режимов (максимум по свободным степеням свободы, нормированный на масштаб нагрузки): медиана по этапам $1{,}6\cdot10^{-4}$ у Пикара против $1{,}9\cdot10^{-9}$ у Ньютона, максимум $1{,}5\cdot10^{-3}$ против $1{,}6\cdot10^{-8}$, то есть Ньютон точнее по невязке примерно в 10^5 раз. На 96–100 % (медиана 99 %) невязка Пикара — сила этой пружины; свою же регуляризованную задачу Пикар решает точно.

- **Ошибка Пикара относительно точного решения уравнения (1) на тех же входах:** профиль M — медиана 0,05 %, максимум 0,83 %; профиль w — 0,03 % и 0,99 %; максимум |Q| — до 0,52 %. M, Q, w у Пикара ниже, чем у Ньютона, на всех 64 расчётах; Кз у Ньютона ниже на 61 из 64, Км — на 31 (у остальных без изменения в пределах округления ответа). Пикар занижает усилия и прогиб и завышает запасы, то есть не в запас.
- **Причинность проверена контрфактом:** с полом ε = 1e-6 вместо 1e-4 расстояние Пикара до Ньютона падает по медиане M примерно в 37 раз (0,059 → 0,0016 %), по максимуму в 50 раз.
- **Это не смена модели:** состояния контакта одни и те же (из 169 этапов различаются 4, по одному узлу или связи), равновесие единственно (холодный старт Ньютона и прогрев от решения Пикара дают одно решение), а энергия Пикара выше минимума на всех 168 этапах.

Критерий (а), переформулированный ([0297]): режим ratchet_check.py --solver-switch

Обычные гейты храповика рассчитаны на шум архитектуры (0,1–0,5 %), а при смене решателя числа законно сдвигаются на вклад пружины ε (до 1 %). Критерий в режиме --solver-switch [РСТ]: (1) все пары, бывшие в допуске, остаются в допуске без полосы предупреждения; (2) число расчётов «все пять» не меньше базы; (3) сдвиг каждой пары — относительное изменение величины продукта — не больше РСТ % (по умолчанию 1,0; сравнение с разрядностью замера 0,01 %), по нескруглённым числам (product_raw, пишутся при --local: ответ округляет M и Q до 0,1, прогиб до 0,01 мм, запасы до 0,01); (4) Σ выхода за 10 % и медианы не хуже базы более чем на вклад ε, пересчитанный в единицы этих гейтов (оценка сверху; в отчёте печатаются и фактические числа).

Набор	«Все пять»	Пар в допуске	Σ выхода за 10 %, п.п.	Наибольший сдвиг пары
Основной, 56	41 → 41	238 из 280 → 238 из 280	1690,7 → 1692,0	1,001 % (расчёт 2, прогиб; в пределах разрядности замера)
Восстановленные по скриншотам, 11	4 → 4	24 из 55 → 24 из 55 (по нескруглённым 23 → 24: расчёт 2137, Q −10,02 → −9,99 %)	599,3 → 597,6	0,57 %
Настоящие отчёты проектировщика, 9 строк	6 → 6	35 из 45 → 35 из 45	136,8 → 136,4	0,46 %

Медианы |Δ| основного набора, % (M / Q / w / Кз / Км): 2,55 / 2,45 / 3,45 / 1,85 / 2,80 → 2,55 / 2,80 / 3,45 / 1,85 / 2,90; пар, вышедших из допуска, нет ни на одном наборе. Режим --solver-switch пройден на трёх наборах; обычный режим на старой базе формально проваливается (Σ +1,3 п.п., медиана Q +0,35 п.п.) — ради этого критерий и переформулирован. Контроль all_sweep.py --local --jobs 2 на одной основе (linearDeformable): основной 41 из 56 (M 49, Q 49, w 44, Кз 47, Км 49), восстановленные 4 из 11, настоящие отчёты 5 из 6, общий итог 50 из 73. Время трёх наборов с разбором отчётов — 37,6 с у Пикара против 13,3 с у Ньютона; итераций решателя на основном наборе 87 072 против 440.

Что меняется для пользователя

- M, Q, w в среднем вверх на 0,06 % (до 1 %), Кз и Км чуть ниже («в запас»); состав пар в допуске и «все пять» не меняются.
- Мягких принятий (статус soft) на обычных расчётах нет (у Пикара на наборе их было 3 этапа из 168); расчёт быстрее.
- У проектов на пределе несущей способности заделки Ньютон прямо называет отсутствие равновесия (status = mechanism, lambda_limit < 1) там, где Пикар «держал» стенку страховочной пружиной ε и отвечал near_limit: в тестовом проекте с осушённым лицом — λ_lim = 0,988 вместо near_limit с прогибом 419 мм. Тест закреплён.
- Прогон МКЭ на контрольных задачах Блюма: Mmax консолей Б-1, Б-3, Б-4, Б-7 теперь в пределах 0,1–0,3 % от метода Блюма (у Пикара было ниже на 5–6 %); Б-2 +5,6 % (у Пикара 0,0 %); Б-5 и Б-6 без изменений. Страница сверки показывает разницу двух расчётных моделей (МКЭ против метода Блюма), а не отклонение МКЭ от замкнутого решения: замкнутая форма с [0474] есть у всех семи задач, но для МКЭ она — сопоставление моделей, не верификация. МКЭ-прогон реестра идёт на сетке 0,25 м (fem.mesh_step), поэтому у Б-6 отклонение момента от предела сетки (−9 %) — следствие шага, а не модели; подробно — глава 7.

> **Что известно о точности и что не решено**
>
> По измерениям аудита (прототип scripts/prototypes/arclen, 8 систем на текущем ядре) ответы Ньютона совпадают с Пикаром в пределах ⩽ 1 % при 0,01 с вместо 3–17 с на вызов; дрейф по набору — медиана 0,04 %, максимум 1,6 %, а 2001.1 по прогибу стоит на границе допуска (−10,4 %). Замер [0297] объяснил этот сдвиг пружиной ε Пикара (выше); решение о переключении принято

> владельцем после него.
>
> На синтетическом случае обход гарда через active_set вернул converged=True с моментом, заниженным в 5,7 раза, тогда как λ-тест показал λ_max = 0,969 (механизм, выданный за «у предела»); вход реконструирован — это гипотеза о частоте такого сценария.
>
> Не решено владельцем: что показывать вместо чисел при механизме (подмена чисел), SRF и запас по λ как критерий (λ_lim — механическая величина, а не Кз), заколы. Диагностика и статус готовы, подмены чисел нет.

Чистка 1.1: ручки и мёртвый код ([0377-чистка], 15.10.2026)

Блок 1.1 решений владельца (решения_владельца_2026-10-14). Удалены без изменения чисел продукта по умолчанию: решатели контакта initial_stress (Пикар с ε-лестницей и near-limit контуром: подшаги выемки, запасная стратегия, _guard_near_limit_stage, _attach_mechanism_check), penalty (остаточная жёсткость β = 0,17) и active_set двустороннего пути, внешний цикл по состояниям связей _solve_with_nonlinear_nodal, поле β у SoilLayer и SoilSpringLaw; слияние смежных полос пригрузки (_merged_strip_widths, P-06) — решение «широкая / узкая» принимается по собственной ширине каждой полосы; тумблер вырожденной призмы среза pioneer_degenerate_at_rest (B-04, пол под призмой не менялся).

Проверка: побитное сравнение с деревом до шага — табло лестницы (84 строки) и 92 расчёта наборов (56 + 11 + 9 + 6 + 10); единственная строка, где число сдвигается, — расчёт 2182 набора проектировщика, чей сохранённый запрос несёт contact_strategy = initial_stress (теперь Ньютон; сдвиг не более 0,23 % по w). Поля запроса после [0412-строгий-вход]: contact_strategy принимается только со значением newton, pioneer_degenerate_at_rest — только false (прочие значения — 422, подробнее — глава 8); у solve_elasto_plastic игнорируются max_outer_iter, penalty_eps, relaxation_factor, reset_bond_states, outer_warm_start. Из интерфейса убраны переключатели модели запаса «Автоматически» и «Жёстко-пластическая», допущение «давление покоя при работающей связи» и список «давление при вырожденной призме среза»: проект, сохранённый с таким значением, считается как раньше и показывает уведомление; API принимает safety_model = auto / rigidPlastic и restrained_at_rest как прежде и добавляет пометку в warnings.

6.14. Управление долгим счётом (run_control) и оценка стоимости ([0293-C1, C3])

«Крайний» расчёт не падал, а тысячи итераций искал равновесие, которого нет: минуты счёта, 504 от nginx и «зомби»-счёт после ухода клиента. `core/fem/run_control.py` добавляет кооперативное управление: `RunControl` (дедлайн, флаг отмены, счётчики, решённые этапы) лежит в `ContextVar`, который устанавливает вызывающий (`with run_control(ctl)`), — сигнатуры решателя не менялись. `check_run_control()` вызывается раз в итерацию в циклах Пикара initial_stress, penalty и active_set, в Ньютоне (раз в итерацию и раз в уровень λ), во внешнем цикле связей и между подшагами предохранителя; `note_stage_start/note_stage_done` сообщают о ходе этапов. Вне блока вызовы — пустые операции. Исключения `SolverTimeout` и `SolverCancelled` (оба от SolverInterrupted) НЕ наследуют RuntimeError/ValueError: в staging.py стоят except (ValueError, RuntimeError, LinAlgError) вокруг подшагов, и прерывание не должно быть проглочено. ПРОВЕРЕНО: с

включённым управлением M, Q, w и эпюры побитово прежние на трёх решателях (Пикар, Ньютон, схема со связью); один вызов ≈ 1,25 мкс против 3–17 мс на итерацию.

Сочетание нагрузок считает схему дважды под одним бюджетом: `RunControl.begin_pass()` разделяет проходы; таймаут на втором расчёте оставляет основной ответ целым, а в таблице сочетания — причину; таймаут на основном — второй расчёт не запускается.

`core/fem/cost_estimate.estimate_solve_cost`(n_nodes, n_stages, n_bonds, pioneer, …) — ожидаемое (`expected_s`), верхнее (`upper_s`, 90-й процентиль числа итераций) и предельное (бюджет решателя) время и число итераций Пикара: t = I·(T0 + T1·n) + [пионерный срез]·P·n + S·n·N_эт, I = N_эт · k_св · i_solve. Калибровка — `scripts/bench/cost_calibration.py` на 20 расчётах после ускорения: при известных итерациях медиана ошибки 5 % (худший ×1,34); до расчёта — 51 % (худший ×2,9, в пределах ×2 — 14 из 20), факт выше upper_s — 1 из 20. Откалибрована только стратегия initial_stress (Пикар); для Ньютона calibrated = False, оценка консервативная верхняя граница, калибровка Ньютона не сделана (до первого решённого этапа время завышено); множитель машины — HYDROWEB_COST_MACHINE_FACTOR. В прогресс фоновой задачи идёт upper_s; после первого решённого этапа оценку строит RunControl.estimate_total_s по измеренному времени этапов. Отказ или предупреждение до счёта по этой оценке (скрининг) пока не сделаны.

6.15. Режимы ядра по желанию (opt-in, [0293-C5, C7, C9])

Все режимы выключены по умолчанию, и при выключенных ответ /api/calculate/fem совпадает побитово (тесты сравнивают весь JSON). Каждый — допущение, которого норматив не задаёт целиком; допущения печатаются в ответе.

- `aquifers[]` + `aquifer_mode` (C5a). Модель в core/water_pressure.py: WaterBearingLayer, StagedAquifer, LayeredPorePressure, layered_stage_water. СП 381 п. 6.2.13 (u = γf·γw·Hw) и прил. Е рис. Е.5: внутри слоя гидростатика его пьезометрического уровня (уровень = кровля − «Напор»; ноль — обычная гидростатика от кровли); между слоями линейно (communicating, рис. Е.5 в, г; дефолт) либо ноль (separated, рис. Е.5 а, б); ниже подошвы нижнего слоя — порового давления нет, u = 0, σ'v полное ([0426-вода-под-подошвой], 23.10.2026; прежнее замирание по правилу водоупора [0187] отменено); над кровлей верхнего — допущение: гидростатика уровня, но не больше линейного перехода от нуля на поверхности. Один слой без подошвы = прежний УГВ, с подошвой = УГВ + водоупор (тесты до 1e-9). Хунки с параметром pore_profile / pore_profile_back|front (по умолчанию None) — effective_stress, soil_pressure, elasto_plastic.build_two_sided_spring_laws, staging (soil["aquifers"]: сторона со слоями считается по ним на всех этапах, её уровни этапов, УГВ и водоупор не участвуют). Поля aquifers[]: side back|front, top_depth, bottom_depth?, pressure_head ⩾ 0, start_stage, end_stage (включительно; 0 — до конца); 422 по-русски (подошва выше кровли, кровля ниже разреза, нет этапа, слои стороны пересекаются на одном этапе). В ответе — pore_pressure (эпюры по этапам, флаг applied) и пояснение в warnings. Путь Блюма слоёв не принимает; сочетание с free_standing_mode отклоняется 422 (не определено). Тесты: test_aquifers_core.py (16), test_aquifers_api.py (15), aquifersPayload.test.ts (7).
- `free_standing_mode` (C9). water_pressure.open_water_column_pressure/_resultant — гидростатика открытого столба; staging._open_water_nodal_loads — узловые силы обеих

сторон точным интегралом по грузовой длине (не на затопленном этапе; при связанной воде — предупреждение, не прикладываются); staging._compute_K_embedment_back — F_lim/F по тылу, под флагом Kз = меньший из двух сторон (только linearDeformable). Без режима уровень лица выше дна прижимается к дну, тыла выше подсыпки — к её поверхности. soil_pressure.py и elasto_plastic.py не тронуты. Тест test_free_standing_mode.py (13): замкнутая гидростатика, точная равнодействующая узловых сил, побитовый дефолт, эквивалентность ручной distributed_loads_on_wall, M = 0 при равных уровнях, 1998.1. Эталонный расчёт 1998.1 (причальная стенка) в допуск НЕ входит: этап 2 — M 3,1 / Q 1,9 / w 1,65 мм / Kз 129,6 против печатных 8,9 / 3 / 3,0 / 20,29; без режима та же постановка давала w 2392 мм и converged = false. Ответы проектировщика нужны по: постели и бытовому давлению в теле бермы на тыле, γw (10 или 9,81), методу предела для конечной бермы, стороне интегрирования Kз, источнику продольной силы (−66 кН на элемент). Не сделаны конечная берма на тыле и продольная сила; разбор scripts/gw_compare для 1998.1 неверен (prod_sweep строит неверный запрос).

- `residual_strength_active` + layers[].phi_res_deg / cohesion_res (С7). СП 381, прим. к п. 6.1.7. Активное давление тыла по φ_res / c_res слоёв (там, где заданы); пассив и давление покоя не меняются; 422 без значений, при φ_res > φ, c_res > c и при пионерном срезе. Предупреждение о превышении прогиба на МКЭ-пути ссылается на режим; при включённом режиме говорит, что снижение учтено, а заколы отдельно не моделируются; текст классического пути прежний. Внесено по прототипу zakoly/C_residual_strength_fem_PROTOTYPE.diff; значения чисел задаёт проектировщик — допущение. Тест test_fem_residual_strength.py.
- `deflection_limit_mm` ([0569-релиз-6-20-w-марка-ary], необязательное, больше 0 и не больше 1000 мм; поле проекта `wall.deflectionLimitMm`). Допустимое смещение [w], заданное проектировщиком. Влияет ТОЛЬКО на предел, с которым замечание ответа сравнивает наибольшее перемещение: расчёт, усилия, прогиб и запасы от него не зависят (тест сравнивает ответы с полем и без него: равны все ключи, кроме `warnings` и `notes`). Без поля замечание то же слово в слово (рекомендуемое min(H/100; 100 мм), `core/deflection.py`). С полем текст — «Перемещение X мм превышает заданное [w] N мм (допустимое смещение, заданное пользователем; рекомендуемое по СП 381.1325800.2018, примечание к п. 6.1.7, для этой глубины — M мм). Предельные деформации устанавливает техническое задание.» вместо «превышает рекомендуемое»; текст и строка замечания (код `deflection_exceeded`) собраны в слое API (`hydroweb/api/deflection_note.py`, вызов в `_apply_notes` файла `main.py`), ядро для этого не менялось. Русское имя поля для отказов — «допустимое смещение» (`refusal.py`); поле внесено в базовый запрос строгих тестов (`_strict_bases.py`) и в перечень `frontend/src/services/femRequestFields.json`. У ответа МКЭ нет поля `w_limit_mm` (оно только у классического `/api/calculate`, где [w] не учитывается). Клиент шлёт поле, когда `wall.deflectionLimitMm` задано и верно (`buildFemRequest`; реестр `projectFieldFlow` — вид «sent»); проверки «Подбора решений» (`buildFemCheckRequest`) шлют запрос без него: тело служит ключом кэша, а проверка судит перемещение сама (`judgeFemResponse`), и смена [w] не должна гонять МКЭ повторно. В документах замечание сжимается правилом `reports/clientWarnings.ts` до фразы «Перемещение 105,7 мм больше заданного [w] 80 мм; рекомендуемое по СП 381 — 30 мм.»; текст сервера закреплён тестом

`test_the_message_text_is_the_one_the_client_shortens_for_documents` и его копией в `clientWarningsW.test.ts`. Тесты: `test_fem_deflection_limit_field.py` (17).

- `load_combination` = none | sp381_basic | sp381_special + emergency_water_depth, water_gamma_f_back/front, γf пригрузок и нагрузок на стенку (C5b; core/load_combinations.py, api/load_combinations_fem.py). sp381_basic выполняет ВТОРОЙ расчёт на расчётные нагрузки (п. 6.2.3, ф. (6.1), табл. 6.2: пригрузки по категории 1,15 / 1,1 / 1,2; давление воды тыла 1,1 и лица 0,9 по ф. (6.2)); sp381_special — аварийное обводнение п. 6.2.25 (γf = 1,0; прим. 2 — столб ⩽ 5 м без водоупора и вод). Основной ответ — на характеристических нагрузках; таблицы «характеристическое / расчётное» — в поле load_combination (прогиб — II группа, п. 6.2.5). В ядре γf применяется к поровому давлению на стену (u_factor в calculate_pressure, water_gamma_f_back/front в законах пружин и staging; умножение только при ≠ 1,0, σ′v не меняется). Допущения L1–L7 (категория пригрузки не задана — 1,2; разгрузка благоприятна — 1,0; вода тыл 1,1 / лицо 0,9; ψ = 1,0; сейсмика не входит; ф, с не пересчитываются; столб ограничен 5 м только без водоупора и вод). 24 теста (16 без МКЭ + 8 API). В интерфейсе выбор сочетания есть только в секции фильтрационной устойчивости.
- `mesh_snap` (C1). core/fem/mesh.py: snapped_nodes (равномерная сетка того же шага + характерные глубины — дно этапов, связи, границы слоёв, уровни воды; слияние близких ключей по приоритету, без коротких элементов), wall_key_depths, nearest_node; связи — в ближайший узел, точечная нагрузка на стенку — в ближайший узел (ветка только под wall["nonuniform_mesh"]). Дефолт False — равномерная сетка, ответ побитово прежний. Тесты test_mesh.py (11), test_api_fem_mesh_snap.py (5): при ΔH = 1 мм и Δ глубины связи = 1 мм M, Q, w меняются ⩽ 1 %. По оценке аудита (гипотеза) сетка с узлами в характерных глубинах опустит базу сверки до 38–40 из 56 — подгонкой её не возвращать.

> **Что выведено в интерфейс**
>
> Через buildFemRequest из интерфейса идут: вода слоями, свободно стоящая стенка, остаточная прочность и допустимое смещение [w] (deflection_limit_mm — только предел замечания о перемещении, когда поле задано во вкладке «Ограждение»). Только через API: mesh_snap и load_combination (contact_strategy устарел: принимается только newton и ни на что не влияет, интерфейс его не отправляет); reset_bond_states и outer_warm_start у решателя игнорируются (внешний цикл по состояниям связей удалён в [0377]).

Глава 7. Блок контрольных задач

Единый реестр контрольных задач для обоих ядер (Блюм и МКЭ): эталон — замкнутое решение статики той же схемы (у метода Блюма с [0474] — на всех семи задачах). Новые возможности подтверждаются добавлением кейсов в реестр, а не разрозненными test_*.py.

Расположение — `hydroweb/tests/benchmarks/`. Ключевой модуль `registry.py` содержит dataclass BenchmarkCase и BENCHMARK_REGISTRY (не импортирует hydroweb.*). Провайдеры замкнутых решений — в `analytical.py`.

Запуск

```
pytest -m benchmark                    # только блок
pytest -m benchmark --benchmark-report # + benchmark_report.md в корне
pytest -m "not benchmark" -q           # всё, кроме блока (быстрый цикл)
```

benchmark_report.md — таблица ID / Group / Name / Status / Tolerance / Source по всему каталогу (группы BB/WK/SP/BL/EP/GW), включая ещё не реализованные (xfail/skip).

> **Артефакт, не код**
> benchmark_report.md пересоздаётся при каждом прогоне и в git не попадает (в .gitignore). Источник истины — реестр в коде.

Допуски

Линейная задача ⩽ кубики → 1e-10; узловой прогиб → 1e-9; постпроцессинг M/Q (O(h^2)) → 2e-3; Hetényi w/M → 5e-3 / 5e-2; скаляры Блюма → 0,5 %; ⩽ 2 % к аналитике — жёстко; промышленный аналог → ±10 % только после Этапа 4.

Пользовательский модуль blum_benchmarks (Б-1…Б-7)

Отдельный от pytest-фреймворка деливерабл — реестр 7 контрольных задач Блюма с узловым сравнением (σa/σp/Q/M по глубине), доступный через публичный API и веб-страницу `/benchmarks/blum` (рис. 3). Эталон каждой задачи — замкнутое решение статики той же схемы (core/blum_closed_form.py, [0474]: Б-1 — явные формулы; Б-2…Б-7 — кусочно-линейное давление Ренкина, Q кусочно-квадратичная, M кусочно-кубический, условие Блюма — корень кубики на участке пяты; корень принимается при переходе невязки «+→−», при нескольких — наибольший; невязки ⩽ 2e-16; коэффициенты проверены sympy). Прежний numpy-интегратор трапеций оставлен вторым контролем в tests/test_blum_closed_form.py (его собственная ошибка O(h) на скачках давления). Ядро против замкнутой формы (замер 30.10.2026, шаг ядра 0,05 м, [0573]): по всем скалярам семи задач не хуже 0,015 % (наибольшее — Q_internal Б-5, −0,0142 %, и F_prop Б-5, −0,0114 %: округление ответа ядра до 0,01), по узловым эпюрам Q и M — не хуже 0,07 % от максимума (Б-5 M 0,062 %: усилие в распорке для эпюры берётся из ответа, округлённого до 0,01); вердикт pass у всех семи, записей KnownDeviation нет. Без округления ответа глубина заделки совпадает с корнем замкнутой формы до 5e-7 м, а Mmax, Q_internal и F_prop — до 1e-4 % (test_embedment_bisection.py). Остаток 29.10.2026 ([0551]: скаляры до 0,110 %) снят в [0573]: бисекция останавливается по ширине интервала d (1e-6 м) и относительной

невязке вместо абсолютного допуска tol = 0,1 кН·м/м (Mmax Б-6 был −0,110 %), а невязка моментов и kst берутся точными интегралами embedment._WallStatics.moments_about вместо трапеции (Mmax Б-5 был +0,042 %); tol оставлен в сигнатуре как верхняя граница досрочной остановки. Усилия точно делает embedment._WallStatics (подраздел ниже). До [0551] (np.linspace без узла на дне котлована, трапеция через ступеньку Q в точке распорки) было: Mmax −1,2 % (Б-2), +3,0 % (Б-5), +1,6 % (Б-6), Q_internal −1,7 % и Q_at_toe +0,6 % (Б-2), узловой M Б-2 — 6,0 % от максимума; восемь записей KnownDeviation (пять у скаляров, три у узловых эпюр) сняты. Q_internal_max распорки у ядра — max|Q| по всей стенке (нижняя ветвь входит), как у замкнутой формы; прежний открытый вопрос (режет по последнему нулю Q, на схеме вне реестра −30,8 % при любом шаге) снят. Замкнутая форма распорки проверяет предпосылку свободного опирания G(h) > 0 и при нарушении ставит флаг free_earth_premise_violated (ядро на таких схемах отказывает). Аналитического эталона для Кз МКЭ нет.

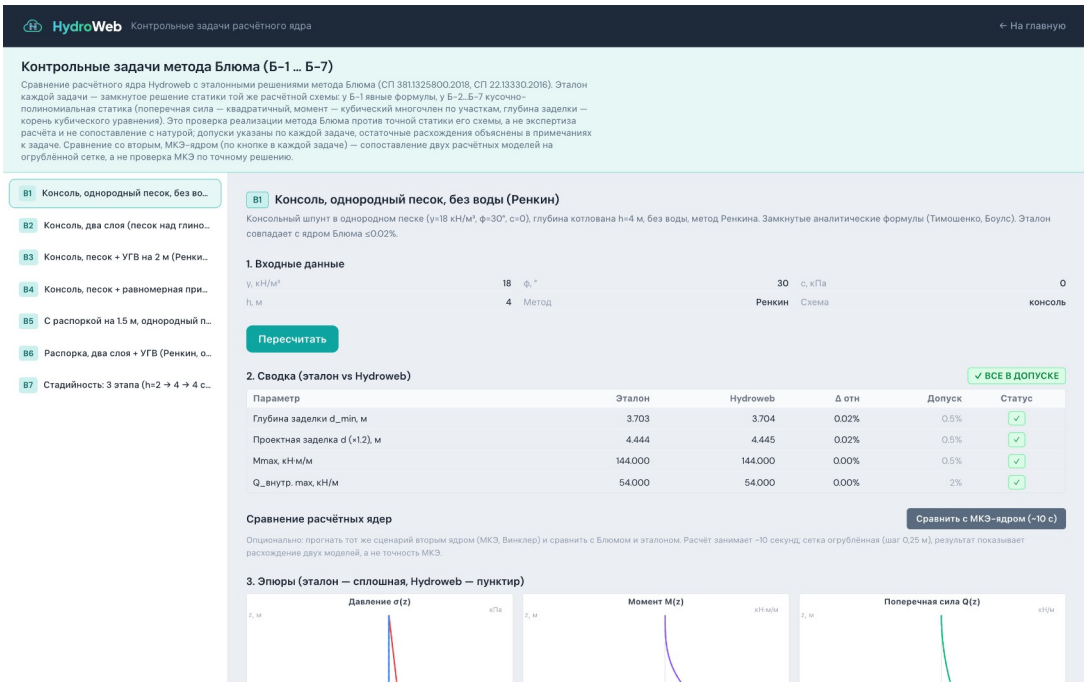


Рис. 3. *Страница «Контрольные задачи метода Блюма» (/benchmarks/blum), задача Б-1: входные данные, сводка «эталон / Hydroweb» с допусками и статусами, совмещённые эпюры.*

Pass узловой эпюры — по нормированной абсолютной погрешности |model−exact|/max(|exact|) (робастно у нулей). Допуски — потолки по умолчанию (DEFAULT_SUMMARY_TOL_PCT, NODE_TOL): скаляры 0,5 % (Q_internal 2,0, Q_at_toe и R_toe 1,0), узлы σ 0,1 %, Q 2,0 %, M 1,0 %; с [0551] поднятых под ошибку сетки допусков по задачам нет (были Б-2 Mmax и Q_internal 2,0, Q_at_toe 1,0; Б-5 Mmax 3,0; Б-6 Mmax 2,0; узловой M Б-2 и Б-6 — 2,5, Q Б-6 — 2,0). Факт: σa, σp в узлах — 0,000 %; M и Q — не хуже 0,07 % от максимума (Б-5 M 0,062 %, Q 0,014 %; Б-6 0,002 %; у Б-1…Б-4 и Б-7 — 0,000 %); остаток у Б-5 — не интегрирование: усилие в распорке для эпюры берётся из ответа ядра, F_prop округлён до 0,01 (до [0573] — 0,099 % и у Б-6 0,30 %: трапеция в уравнении моментов и допуск бисекции по d_min). Механизм KnownDeviation(причина, ожидаемая величина, полоса ±0,3 п. п.) в known_core_deviations сохранён, записей нет: величина с записью получает статус known вместо pass даже в допуске — пока факт лежит в полосе; вне полосы — fail (страница краснеет, реестр падает). Блок «известные отклонения» на странице виден только при записях. Вердикт задачи — pass / known / fail (сейчас pass и

all_passed = True у всех семи); работа механизма проверяется на подставленных записях (test_blum_benchmarks.py).

Сравнение со вторым ядром (кнопка «Сравнить с МКЭ-ядром», `POST /api/blum-benchmarks/{id}/run-fem`). МКЭ-прогон реестра идёт на ОГРУБЛЁННОЙ сетке: шаг 0,25 м (_FEM_MESH_STEP в core/blum_benchmarks.py; рабочий шаг продукта — 0,1 м), k_s = 10000 кН/м³ одно для всех задач, продуктовые умолчания не применяются. Шаг указан в ответе (fem.mesh_step) и печатается на странице рядом с числами. Расхождение Mmax с методом Блюма складывается из сетки и модели, и их нельзя смешивать (замер 15.10.2026, шаг 0,0125 м принят за предел сетки): момент при 0,25 м отличается от предела на +0,3 % у консолей Б-1, Б-3, Б-4, Б-7, на +4,0 % у Б-2, +2,6 % у Б-5 и −9,1 % у Б-6 (при 0,1 м у Б-6 +4,5 %; зависимость от шага не монотонна); это сетка и она уменьшается с шагом. Предел МКЭ у схем с распоркой остаётся ниже эталона задачи: Б-5 на 8,6 %, Б-6 на 12,5 % — это модель (упругое основание у пяты против свободного опирания в методе Блюма), сеткой не объясняется. Сравнение показывает различие двух моделей и не является проверкой МКЭ по точному решению: замкнутого решения для этой нелинейной контактной задачи нет. Тест — test_blum_benchmarks.py (test_fem_mesh_step_*).

Задачи Б-2…Б-7 активны в реестре hydroweb/tests/benchmarks/cases.py и проходят (с [0186]; эталон с [0474] — замкнутая форма; BL-2 с [0551-блюм-узлы-сетки-показа] проходит: xfail по узловому моменту снят); остаются два xfail — заглушки GW-1 и GW-2. Блок контрольных задач НЕ является гейтом для правок под сверку: он считает без продуктовых дефолтов (min_active 0, фиксированный k_s, модель запаса не задаётся). Страница /benchmarks/blum говорит то же: эталон — замкнутое решение статики той же схемы; это проверка реализации метода, а не экспертиза расчёта и не подтверждение точности на реальном объекте.

Точное интегрирование усилий Блюма ([0551-блюм-сетка-и-q-internal], [0551-блюм-узлы-сетки-показа], 29.10.2026)

Правка классического ядра в двух файлах: hydroweb/core/embedment.py (усилия и глубины экстремумов) и hydroweb/core/internal_forces.py (эпюры показа); в реестре blum_benchmarks.py — снятие записей и допусков, поднятых под ошибку сетки, и тексты. Эпюры давления (soil_pressure.py), МКЭ, политика Блюма и API не менялись.

- **Класс _WallStatics (embedment.py).** Нетто-давление p(z) = σa(z) − σp(z − h)·[z > h] берётся из тех же точек эпюр (PressureProfile.points) и между соседними узлами линейно. Узлы разбиения — все узлы активной эпюры, все узлы пассивной эпюры (сдвинутые на h), дно котлована h (пассив начинается скачком 2c√Kp), точка распорки и пята L = h + d. На участке p = a + b·t, поэтому $Q(t) = Q_0 + a·t + b·t^2/2$ и $M(t) = M_0 + Q_0·t + a·t^2/2 + b·t^3/6$ — точные многочлены; скачок Q на F — в узле z_p (Q непрерывна справа). Экстремумы: M — в нулях Q на участках (корни квадратного уравнения), |Q| — в нулях давления и на узлах, с обеих сторон скачка у распорки.
- **Обёртки.** _compute_forces_cantilever и _compute_forces_propped стали обёртками над _WallStatics; параметр step оставлен для совместимости, результат от него не зависит. Q_internal консоли — прежнее определение (|Q| до последнего нуля Q, хвост у пяты — R_toe); Q_internal распорки — max|Q| по всей стенке; Q_at_toe распорки = |Q(L)| = 0. M_max_depth и Q_internal_max_depth — глубины точек экстремума, а не узлов равномерной

сетки (в ответе M_depth, Q_internal_depth). Прежде np.linspace не имел узла на дне котлована и в точке распорки: скачок σp = 2c√Kp размазывался на шаг, а трапеция через ступеньку Q сдвигала M на F·шаг/2.

- **Эпюры показа (internal_forces.py).** compute_internal_forces берёт Q и M из _WallStatics.sample; число точек — медиана 84 → 85 (от −2 до +7). Сигнатура, поля и порядок точек прежние; изменились массивы moment и shear в ответе /api/calculate (графики, таблицы, выгрузки); скаляры от этой части не зависят.
- **Что не менялось в [0551] и сделано в [0573].** Уравнение моментов поиска заделки (_moment_residual_*, kst) брало интеграл σ·плечо трапецией по узлам профиля — для квадратичной подынтегральной это O(step²), а допуск бисекции tol = 0,1 кН·м/м был абсолютным. Оба решения затрагивали d_min и kst и сделаны отдельной правкой [0573-блюм-относительный-допуск-бисекции]: невязка моментов и kst — точные интегралы (_WallStatics.moments_about), бисекция останавливается по ширине интервала d (1e-6 м) и относительной невязке 1e-10·(|Ma| + |Mp|). На корпусе из 636 расчётов d_min меняется на сотые доли процента (медиана 0,005 %, p90 0,056 %); больше 0,1 % — в 33 расчётах, все при заделке d_min до 2 м, где невязка моментов по d меняется медленно и абсолютный допуск 0,1 кН·м/м грубее (ошибка d не больше 0,1/|dΣM/dd|; до 12 % по d_min при Mmax 0,03 кН·м/м); Mmax и Qmax консоли не меняются.
- **Корпус до и после.** 660 запросов классического пути (53 сохранённых запроса набора эталонов и 18 запросов проектировщика, каждый в четырёх вариантах — консоль или распорка, с водой и пригрузкой или сухо, — и 376 синтетических): в /api/calculate 636 ответов 200 и 24 отказа 422 («рассчитайте МКЭ» или несходимость), те же до и после. На всех 636 расчётах d_min, d_design, total_length, kst, F_prop, R_toe, iterations, converged равны прежним без округления; статус, scheme, converged, policy, число и коды замечаний равны; ни один расчёт не пересёк η = 1,0. Изменились только величины, которые считает _compute_forces_*, и производные от Mmax и Qmax: η, Ks_metal, wmax, M_depth, Q_internal_depth, числа в текстах замечаний.

Группа (M ⩾ 5 кН·м/м)	Расчётов	Mmax: медиана / p90 / макс, %	Qmax: медиана / p90 / макс, %
Консоль, класс А	212	+0,055 / +0,36 / +1,1 (все вверх)	0,05 / 0,39 / 2,7
Консоль, класс Б	148	0,32 / 0,99 / 2,5 (109 вверх, 39 вниз)	0,46 / 1,6 / 3,6
Распорка (учебный режим)	263	−1,3 / 3,2 / 7,8 (220 вниз, 43 вверх)	0,37 / 25,7 / 71

- **Класс А.** Mmax растёт во всех 212 расчётах (прежняя трапеция на сетке занижала пик). Наибольшее +1,107 % — запрос ref:1713.9 (суглинок c = 48 кПа ниже дна): класс А проверяет c ⩽ 2 кПа только в пределах H, грунт заделки порогам не подчиняется, скачок σp = 2c√Kp на дне там большой, и прежняя сетка размазывала его сильнее всего.
- **Распорка.** Mmax до 3,2 % у 90 % схем, наибольшее 7,8 % — при малых моментах (h = 3 м, распорка на полвысоты, пригрузка 20 кПа: Mmax 10,1 → 10,9; F·шаг/2 ≈ 0,8 кН·м). Qmax: в 93 из 264 расчётов изменился больше чем на 1 % (p90 25,7 %, наибольшее 71 %) — это нижняя ветвь Q, которую прежнее определение отсекало. В проекте П3 теста test_api_calculate_report_fields (две распорки, учебный режим): Qmax 24,5 → 37,0, Mmax 46,3 → 45,8, η 0,0647 → 0,0639, Q_at_toe 0,4 → 0,0.

- **Автоподбор профиля.** По замеру независимой проверки (при подготовке книги не воспроизводилось) POST /api/auto-profile, подбор по Kм при γc = 0,90 из Mmax и Qmax ядра, меняет профиль в 10 из 1002 подборов; в продуктовом режиме (консоль, без учебного режима) — в 3 из 366: η в базе у них 0,98–1,00, подбор стоит на границе запаса, и сдвиг Mmax на десятые доли процента переводит его через границу.
- **Регрессия test_gw_references.** Закреплённые числа перебазированы: GW-1 Mmax 79,13 → 77,41 (−2,2 %), Qmax 53,75 → 54,53 (+1,5 %), простая схема 120,49 → 118,94 (−1,3 %); сдвиги внутри окна ±3 %, F_prop 52,86 прежняя. Разрыв с эталонной программой: Mmax −19 %, Qmax +19 % (были −18 и +17).
- **Клиент.** `blumShearExplanation` (frontend/src/reports/blumDiagrams.ts): у схемы с опорой Qmax — наибольшая по модулю Q по всей стенке, нижняя ветвь входит; фраза «в Qmax не включено» осталась только у консоли. Фикстуры frontend/src/testing/blumApiResponses.json П2, П3, П3б обновлены ответами нового сервера на те же запросы (поля moment, shear, Mmax, Qmax, Q_at_toe, Q_internal_depth, M_depth; η и Ks_metal пересчитаны на ёмкость самой фикстуры): П3 Qmax 24,5 → 37,0 (пик на дне котлована, 4,50 м), Mmax 46,3 → 45,8; П2 Mmax 72,2 → 70,7; П3б Mmax 164,0 → 162,6, Qmax 70,3 → 71,2. Фикстуры П1, П4, П5 (консоль, отличия — десятые доли процента) не пересобирались.
- **Тесты.** `test_embedment_statics.py` (50): интегрирование ядра на эпюрах при глубине заделки замкнутой формы — до 1e-4 по Mmax, Q_internal, Q_at_toe и глубинам экстремумов (Б-1…Б-6), независимость от шага, скачок пассива на дне и скачок Q в узле распорки, нижняя ветвь Q_internal, сетка 36 схем против замкнутой формы ⩽ 0,3 %; `test_internal_forces_exact.py` (10): Q и M в каждом узле показа против замкнутой формы (до 1e-4 от максимума, Б-1…Б-6), узел на дне, кратные шагу узлы, тройка у распорки и скачок на F, консоль без узлов распорки; `test_blum_closed_form.py` (117): прежние xfail(strict) стали обычными проверками, граница скаляров — округление ответа ядра (была 0,15 %), узловых Q и M — 0,1 % от максимума (была 0,35 %); `test_embedment_bisection.py` (53): точные моменты против интегралов статики, трапеция как второй контроль (O(step²)), независимость d_min от масштаба схемы (γ, c, q в 1e-3…1e3 раз), корень против brentq и замкнутой формы, tol как верхняя граница, max_iter, граница «бисекция невозможна», остановка с converged = False при NaN и скачке невязки, kst = 1 при запасе 1,0; `test_blum_benchmarks.py` (29) и `test_api_blum_benchmarks.py` (7): у всех семи задач вердикт pass, записей нет, механизм KnownDeviation проверяется на подставленных записях. Ключи преподавателю пересобраны.

Глава 8. API

FastAPI (hydroweb/api/main.py), Pydantic-модели на все request/response. Авторизация — bcrypt + JWT в httpOnly-cookie, Yandex OAuth, роли admin, professional, standard и supplier; у учётной записи есть вид (users.account_kind: проектировщик или поставщик). CORS — whitelist через HYDROWEB_CORS_ORIGINS (не «*»).

Метод	Путь	Назначение
POST	`/api/auth/register, /login`	bcrypt + JWT; ставят cookie hw_session; вход в заблокированную учётную запись — 429 с Retry-After
GET	`/api/auth/me`	текущий пользователь по cookie (401 — сессии нет); с [0548-уровни-доступа-шаг-1] ещё accessLevel: действующий уровень с подписью, сроком, источником (assigned | expired | trial | default), признаками beta, expired, trialActive и enforced (false, пока выключатель HYDROWEB_LEVELS_ENFORCED выключен; следует за ним с [0570-уровни-доступа-шаг-2a]), матрица уровня; то же в GET /api/profile и в ответе POST /api/auth/login
POST	`/api/auth/logout`	отзывает сессию на сервере (revoke_user_sessions: эпоха +1, на всех устройствах) и гасит cookie; без действующей сессии только гасит cookie
GET	`/api/auth/oauth/yandex[/callback]`	вход через Яндекс ID; токен не передаётся в URL; state в подписанной httpOnly-куке
GET	`/api/auth/oauth/vk[/callback]`	вход через VK ID (код авторизации + PKCE, device_id); те же правила привязки
GET	`/api/auth/oauth/providers`	какие способы входа настроены на сервере (флаги)
POST	`/api/auth/register`	регистрация; необязательное поле consents [{kind: terms|pd|marketing, version}] записывается в таблицу consents (версия правовых документов LEGAL_DOC_VERSION)
GET	`/api/profiles, /coefficients`	каталоги / коэффициенты
POST	`/api/calculate`	расчёт (ядро Блюма); при включённых проверках по уровню без действующей сессии — 401 «Для расчёта войдите в учётную запись.» (выключено — как раньше)
POST	`/api/calculate-staged`	расчёт по этапам (Блюм)
POST	`/api/calculate/fem`	расчёт МКЭ-ядром (Этап 5): статус ok|soft|near_limit|mechanism|not_converged|timeout, deadline_s (240, 5…600), общий семафор (429 + Retry-After); профессиональный режим: при включённых проверках по уровню нужны вход (401) и уровень «Профи» (403 level_required); выключено — как раньше

Метод	Путь	Назначение
POST	`/api/calculate/fem/finder`	тот же расчёт, что у /api/calculate/fem (общее тело _run_fem), для мастера «Подбор решений» ([0570-уровни-доступа-шаг-2a]): запрос ограничен тем, что присылает мастер — белый список полей по снимку запросов мастера (hydroweb/tests/fixtures/finder_fem_requests.json), не больше двух этапов, одной связи и одной пригрузки, без берм, среза, подсыпки, нагрузок на стенку и водоносных слоёв; лишнее поле или схема вне границ — 422 одной строкой; границы действуют при любом положении выключателя; при включённых проверках нужны вход (401) и возможность solution_finder (есть на всех уровнях); фоновых задач у маршрута нет
POST	`/api/calculate/fem/jobs`	фоновая задача: 202 {job_id, state}; тело и права как у синхронного эндпоинта (вход и уровень «Профи» при включённых проверках); query deadline_s (по умолчанию 600, до 1800)
GET	`/api/calculate/fem/jobs/{id}`	{state, progress, result | error}; Cache-Control: no-store; задача с сессией видна только владельцу (чужая — 404)
DELETE	`/api/calculate/fem/jobs/{id}`	отмена активной задачи (флаг читает тикер раз в 0,5 с, в том числе у другого воркера); у завершённой — забыть результат
POST	`/api/checks/hydraulic`	фильтрационная устойчивость основания: взмыв, выпор, градиент по этапам (СП 381 п. 6.1.12, табл. 6.1 п. 22)
POST	`/api/checks/stability`	общая устойчивость k_st: Феллениус и упрощённый Бишоп по этапам (СП 381 пп. 6.1.21 –6.1.22, 6.4.5)
POST	`/api/checks/anchors`	несущая способность грунтовых анкеров и положение корня (СП 381 п. 7.3, прил. Л); not_checked — перечень состояний, которых проверка не выполняет, с причиной «методики в своде нет» или «в другой проверке» (срыв корня и разрыв тяги в него не входят)
POST	`/api/assistant/chat`	ИИ-ассистент: SSE-поток событий text | sources | done | error; роли standard (3 вопроса в сутки), professional и admin (без суточного лимита); 30 запросов / 10 мин на IP; при включённых проверках по уровню ещё и лимит уровня из матрицы (403 level_required: нет ассистента на уровне или лимит за период исчерпан; суточный лимит роли срабатывает первым — прежний 429)
GET	`/api/assistant/quota`	остаток суточного лимита текущего пользователя: {role, limit_per_day | null, used_today, remaining | null, reset_at}; при включённых проверках и лимите уровня — ещё level_quota {level, period, limit, used, remaining, reset_at}; Cache-Control: no-store

Метод	Путь	Назначение
GET/PUT	`/api/profile`	кабинет: данные учётной записи; смена имени (PUT, а не PATCH — в CORS проекта PATCH не разрешён)
POST	`/api/auth/change-password`	смена пароля вошедшим пользователем; новая сессионная кука, прежние токены гаснут
GET/POST	`/api/profile/consents`	согласия: состояние и история; дать/отозвать согласие на рассылку
GET	`/api/profile/export`	выгрузка «моих данных» одним JSON-файлом
POST	`/api/profile/delete`	самоудаление учётной записи (подтверждение email и, при входе по email, пароля)
POST	`/api/auth/invite/accept-consents`	приглашённый принимает документы при первом входе (любой вошедший с consents_pending)
GET	`/api/admin/users`	список пользователей (admin); кроме полей учётной записи — provider, projects_count, email_verified
POST	`/api/admin/users`	приглашение: создаёт учётную запись и возвращает временный пароль один раз (admin)
POST	`/api/admin/users/{id}/role, /active`	смена роли и активности (admin); журнал admin_audit и ADMIN_ACTION
POST	`/api/admin/users/{id}/password, /reset-link, /verify-email, /delete`	новый пароль, ссылка сброса, подтверждение email, удаление учётной записи (admin) — см. подраздел ниже
POST	`/api/admin/users/{id}/logout`	«Завершить все сессии»: увеличивает эпоху сессий пользователя (admin)
GET/PUT	`/api/admin/users/{id}/access-level`	уровень доступа пользователя (admin, [0548-уровни-доступа-шаг-1]): GET — назначенный и действующий уровень и журнал смен; PUT {level, until, reason} — назначить: одна транзакция (users + строка access_level_changes), в admin_audit действие set_access_level (уровни и срок; причина в журнал действий не пишется). Отказы русской строкой: 404 — нет пользователя, 409 — поставщик (уровней нет) или ничего не изменилось, 422 — неизвестный уровень, срок не ГГГГ-ММ-ДД, не в будущем или дальше UNTIL_MAX_YEARS = 5 лет, срок у «Бесплатного» и у уровня по умолчанию, срок не задан у «Пробного», причина пуста или длиннее REASON_MAX_CHARS = 300
GET	`/api/admin/audit`	журнал действий администратора: фильтры actor / action / target, постранично (admin)
POST	`/api/events`	событие от клиента ([0543-журнал-событий-беты]): тело {kind, meta?}, допустимы только виды export_* (остальные 422), вошедший пользователь (401 без сессии), EVENTS_RATE_LIMIT = 600 запросов за 10 минут с адреса (429); meta — типизированная модель

Метод	Путь	Назначение
		EventMeta (part, engine, project_id, status, stages, new), в строку попадает по белому списку clean_meta; для выгрузок, которые строит браузер (DOCX, XLSX, PDF, DXF, SVG, XML)
GET	`/api/admin/events/summary?from=&to=`	сводка журнала событий (только admin): без дат — последние 30 суток, сутки московские (UTC+3), период не длиннее 366 суток, даты от 01.01.2026 до завтрашних суток по Москве, иначе 422 одной русской строкой («Дата начала периода — от 01.01.2026 до 03.11.2026.»). Ответ: users[] (почта, имя, роль, trial, число событий по колонкам видов и «Вне проекта», всего, время последнего), days[], totals, criteria, notes
GET	`/api/admin/health`	состояние сервера: версия, БД, kb.db, ЕГРЗ, настроенность сервисов, использование ассистента, топ проектов (admin)
*	`/api/projects[/{id}]`	CRUD проектов (изоляция по user; версия revision и base_revision: устаревшая запись — 409, удалённый проект — 410); создание проекта при включённых проверках по уровню — не больше числа активных проектов уровня (403 level_required: «Бесплатный» — 1, «Старт» — 5)
GET	`/api/registry/egrz/*`	поиск по реестру ЕГРЗ (FTS5)
GET	`/api/estimate/zones`	ценовые зоны снимка, период, версии снимка цен и книги норм, источник; без снимка — available: false, не ошибка
GET	`/api/estimate/norms?table=05-01-012`	нормы таблицы ГЭСН-2022: шифр, наименование, единица, обобщённые ресурсы с допустимыми кодами КСР, шифры НР/СП; 404 — таблицы нет в книге
GET	`/api/estimate/support-norms?kind=…`	кандидаты норм для шпунта, распорок, поясов, анкеров и причина, по которой автоподбор по названию не делается (auto_selected всегда false); для распорок и поясов — outside_scope_candidates и outside_scope_note: нормы вне сборников 05 и 09 (сб. 29 и 39-01-021), названы кандидатами, в подбор не добавлены
GET	`/api/estimate/support-select?kind=anchor|strut|belt&…`	подбор нормы по явным условиям (anchor_type, soil_group, force_kn, length_m, borehole_diameter_mm; material=steel для распорок и поясов): норма, проверенные условия, оговорки; не подобрана — 200 со status: not_selected и причиной; параметры не того вида — 422
GET	`/api/estimate/ssr-hints?zone_id=…`	подсказки к ВЗиС (глава 8) и зимнему удорожанию (глава 9): что задаёт процент, цитаты приказов (до 15 слов), температурная зона по Приложению N 4 к 325/пр — только для 4 зон снимка; проценты не подставляются

Метод	Путь	Назначение
POST	`/api/estimate/calc`	ЛСР по позициям + ОСР + (по явным параметрам) ССР, в том числе строки ВЗиС и зимнего; 30 запросов в минуту; позиции без нормы или цены — без суммы, complete: false; у позиции-анкера с явной нормой 05-03-007/008 без selection — необязательное force_kn (кН, > 0, ⩽ 1e6): проверка предела «до N тс», превышение — предупреждение support_force_limit_exceeded без блокировки
POST	`/api/checks/water`	приток воды в котлован и требуемое понижение (безнапорный горизонт; схемы «большой колодец» и метод фрагментов, радиус влияния по Кусакину; источники по докстрингам модулей, номера пунктов не указаны); k_f слоёв из геологии
POST	`/api/joints/check`	проверка узла крепления распорки-трубы: угловые швы и болты по СП 16.13330.2017 (п. 14.1.16, 14.2.9, табл. 38–41, Г.2, Г.5, Г.6, Г.9); K, требуемые катет или число болтов, замечания и not_checked
*	`/api/supplier/items[/{id}]`, `/api/supplier/schema`	кабинет поставщика (роль supplier): позиции, submit, withdraw; схема полей восьми категорий с единицами
GET	`/api/supplier/template.xlsx?kind=`	шаблон Excel категории (лист «Позиции», «Пояснения», скрытый «Списки»)
POST	`/api/supplier/import.xlsx?kind=`	импорт из шаблона: .xlsx ⩽ 2 МБ, ⩽ 200 строк; ответ {created, rows, errors:[{row, field, line}]}, верные строки — черновики
GET	`/api/supplier/export.xlsx`	выгрузка своих позиций в раскладке шаблона плюс «Статус», «Выбрано в проектах», «Записка модератора»
*	`/api/supplier/profile`	профиль организации (GET/PUT; ИНН с контрольными цифрами)
*	`/api/admin/supplier-items[/{id}/approve|reject]`	модерация позиций поставщиков администратором; журнал supplier_item_approve|reject; письмо поставщику о решении (notify_items_decided)
POST	`/api/admin/supplier-items/bulk`	массовое решение до 100 позиций; каждая решается отдельно, отказ по одной не останавливает остальные; записка при отклонении обязательна; одно письмо на поставщика со списком решённых позиций
GET	`/api/supplier-catalog`	публичный каталог одобренных позиций поставщиков; те же позиции добавлены в /api/profiles, /api/anchor-tendons, /api/rrsk/sections с пометкой source «поставщик: …»; профили шпунта поставщиков видят и расчёты /api/calculate и /api/auto-profile ([0541-поставщик-марки-и-блюм]); у публичной строки профиля поставщика есть steel_Ryn_MPa — Ryn

Метод	Путь	Назначение
		марки профиля при его толщине проката (то же число берёт расчёт); позиция, которую нельзя показать (неполные данные, NaN или бесконечность), пропускается с записью в журнал, остальные остаются
GET	`/api/steel-grades`	справочник марок стали ([0541-поставщик-марки-и-блюм]): марки ядра (id, source, table, sp16_grade, E_MPa, интервалы толщины ranges с Ryn) и в конце одобренные марки поставщиков (source «поставщик: …», supplier_item: true, Ryn, Run, E_MPa) — по образцу /api/profiles
GET	`/api/supplier/usage`	сводка продукции поставщика в проектах ([0498-продукция-поставщика-в-проектах]): свои позиции, проекты (всего и за 90 дней), объёмы, цены, месяцы, регионы; роль supplier; 403 «Чтобы видеть сводку, примите условия использования кабинета поставщика.», пока не принята текущая редакция условий
GET	`/api/supplier/terms`	принята ли текущая редакция условий кабинета: {version, accepted} (роль supplier)
POST	`/api/supplier/accept-terms`	принять текущую редакцию условий без отправки позиции: та же запись supplier_terms, повтор записей не плодит; ответ {version, accepted: true} (роль supplier)
GET	`/api/admin/supplier-usage?kind=&limit=&offset=`	та же сводка по всем позициям, в том числе встроенного каталога, с полями source (supplier | catalog) и org; kind — категория, limit 1…1000 (по умолчанию 200), offset до 100 000; ответ {total, projects_total, items, recent_days} (admin)
POST	`/api/projects/import-file`	импорт файла проекта .gwx/.gw4 (multipart, часть file, ≤ 5 МБ): {project, warnings, unsupported, kind}; project = null — есть непереносимое
GET	`/api/price-requests/suppliers?category=&mark=`	поставщики категории с одобренной позицией для диалога запроса цены (публичный каталог идентификаторов поставщиков не отдаёт); роли admin, professional, standard
POST	`/api/price-requests`	создать запрос цены: выбранным поставщикам (supplier_ids, до 50) либо всем поставщикам категории (all_in_category, до 200); 201; письмо каждому поставщику-адресату (notify_new_price_request)
GET	`/api/price-requests/mine?project_id=&status=`	свои запросы с адресатами и предложениями
POST	`/api/price-requests/{id}/close`	закрыть свой запрос; после закрытия предложения и отказы не принимаются
GET	`/api/supplier/price-requests`	запросы, адресованные поставщику, и число новых (роль supplier)
POST	`/api/supplier/price-requests/{id}/offer`	предложение (цена, price_unit, дата цены, срок действия, комментарий); повторное заменяет

Метод	Путь	Назначение
		прежнее, пока запрос открыт; письмо проектировщику (notify_price_offer), если изменились цена, единица цены или срок действия
POST	`/api/supplier/price-requests/{id}/decline`	отказ с причиной; после предложения 409; письмо проектировщику (notify_price_decline)
GET	`/api/admin/price-requests?status=&category=&supplier=&seen=&limit=&offset=`	запросы цены для экрана администратора: фильтры по статусу, категории, поставщику (идентификатор с дефисами или без) и пометке (seen=new|seen); limit до 1000 (по умолчанию 100), offset до 100000; ответ несёт total, counts (новые и просмотренные по всем запросам, без учёта фильтров), suppliers (варианты фильтра) и requests; в строке предложения не раскрыты (offers_count, valid_until); почта проектировщика только при его согласии на передачу контактов
GET	`/api/admin/price-requests/summary`	число непросмотренных запросов цены (счётчик на вкладке; admin)
GET	`/api/admin/price-requests/{id}`	запрос цены целиком: адресаты (состояние, причина отказа), предложения и history (отправка, предложения, отказы, закрытие по времени); admin; чужой или несуществующий идентификатор — 404 одной строкой
POST	`/api/admin/price-requests/{id}/seen`	пометка «просмотрено»: тело {seen: true | false}; статус и updated_at запроса не меняются; в журнал действие price_request_seen
GET	`/api/ground-anchors-tu`	весь каталог грунтовых анкеров по ТУ 25.11.23-001-22252558-2024 (hydroweb/data/ground_anchors_tu.json; публичный, как /api/anchor-tendons; ответ /api/anchor-tendons не менялся); тип ответа GroundAnchorsTuCatalog в frontend/src/types.ts
POST	`/api/checks/struts`	проверки сечения распорки по СП 16.13330.2017: шесть блоков, список позиций 1–200 ([0530-проверки-распорки-и-пояса-в-api]; раздел «Проверки вне МКЭ-расчёта»); в ответе оба предельных осевых усилия — по устойчивости (limit_force_stability_kN) и по прочности A·Ry (limit_force_strength_kN, [0569-релиз-6-20-w-марка-ary])
POST	`/api/checks/wales`	проверки обвязочного пояса: изгиб, срез, приведённое напряжение, список позиций 1–200 ([0530-проверки-распорки-и-пояса-в-api])
POST	`/api/auth/register-supplier`	регистрация поставщика: multipart/form-data, часть data (JSON: email, password, first_name, consents, organization, inn, contact_name, phone, website) и часть file (карточка организации); учётная запись, согласия, заявка и файл одной транзакцией, сбой откатывает и файл; предел регистрации прежний (5 за 10 минут с адреса)

Метод	Путь	Назначение
POST	`/api/supplier-applications`	заявка вошедшего (первый вход через Яндекс или повторная после отказа): те же две части формы; 201; активная заявка уже есть — 409, больше 5 в сутки на пользователя — 429; запись проектировщика — 409, роль supplier и admin — 403
GET	`/api/supplier-applications/me`	последняя заявка владельца или null: данные, статус, причина решения; имени файла на диске в ответе нет
PUT	`/api/supplier-applications/me/file`	заменить файл заявки на проверке (часть file); 10 раз в час с адреса; прежний файл удаляется; заявка решена — 409
POST	`/api/auth/account-kind`	выбор «проектировщик» вошедшим без выбора (тело {kind: designer}); после заявки поставщика — 409
POST	`/api/supplier-applications/me/designer`	«Продолжить как проектировщик»: только владелец отклонённой заявки; account_kind = designer, платформа открывается той же сессией, заявка остаётся в истории; журнал supplier_application_designer; заявка на проверке, одобренная и отсутствие заявки — 409
GET	`/api/admin/supplier-applications[/summary]`	список заявок с владельцем и счётчиками по статусам (status, q, limit до 200, offset) и число новых заявок (admin)
GET	`/api/admin/supplier-applications/{id}/file`	карточка организации: только администратор, всегда вложением (attachment, nosniff, no-store)
POST	`/api/admin/supplier-applications/{id}/approve`	одобрить: роль supplier, account_kind = supplier и профиль организации из заявки (регионы, основание и почта профиля не меняются); записка необязательна; журнал supplier_application_approve; письмо владельцу заявки (notify_application_decided)
POST	`/api/admin/supplier-applications/{id}/reject`	отклонить с причиной 3–500 знаков; решение принимается один раз (повторное — 409); журнал supplier_application_reject; письмо владельцу заявки (notify_application_decided)

POST /api/calculate/fem (Этап 5)

Маппит запрос на `core/fem/staging.solve_staged`. Решатель контакта один — Ньютон (до 05.10.2026 по умолчанию был initial_stress); поле contact_strategy принимается только со значением newton и ни на что не влияет, остальные значения — 422. Жёсткость грунта задаётся либо k_subgrade (кН/м³) явно, либо через модуль деформации E → сервер выводит $k_s = E / (B\cdot(1-\nu^2))$, B = 1 м (для точной сверки с заданным значением задавать k_subgrade напрямую).

Допустимое смещение [w] в запросе ([0569-релиз-6-20-w-марка-agy]): необязательное поле `deflection_limit_mm` (мм, больше 0 и не больше 1000; поле проекта `wall.deflectionLimitMm`). Оно влияет только на предел, с которым замечание ответа

сравнивает наибольшее перемещение; расчёт, усилия, прогиб и запасы от него не зависят, а без поля замечание прежнее слово в слово (описание и тесты — раздел 6.15).

Ответ (FemCalculateResponse): эпюры финальной стадии (deflection, moment, shear, soil_pressure по узлам), огибающая (Mmax, Qmax, wmax, K_embedment, K_wall_metal и utilization_wall_metal = 1/Км, [0475-сравнение-вариантов]), огибающая эпюр envelope ([0396-отчёты-мкэ], [0399-отчёты-первая-волна-остатки]: две ветви M, Q, w в узлах последнего этапа — наибольшее и наименьшее значение по всем этапам и подэтапам, поля moment_max / moment_min, shear_max / shear_min, deflection_max / deflection_min; экстремумы M, Q, w с этапом и глубиной узла того же этапа, значения равны Mmax, Qmax, wmax; api/fem_envelope.py). Ответ Блюма несёт Ryn_check_MPa рядом с Ry_check_MPa и gamma_c; допущения и исходные данные модели model_assumptions ([0401-отчёты-мкэ-пз-тэо-чертёж]: mesh_step_m, min_active_ratio, water_model, subgrade_mode, layers[] с k_s_kNm3 и k_s_source = given | from_E | not_used; значения — те, что принял расчёт; hydroweb/api/main.py, _model_assumptions), ключ sigma_yield_used_MPa — нормативное Ryn, МПа, которое расчёт реально взял для проверки металла (из запроса либо по каталогу профиля, когда марка не задана; null — металл не проверялся): заявленное добавление ответа [0569-релиз-6-20-w-марка-ary], остальные ключи и числа ответа прежние; клиент (mapFemResponse(d, n, steel) в services/femResults.ts) записывает его в результат вместе с маркой стали и профилем (пункт «Сталь расчёта хранится с результатом» подраздела «Интерфейс редактора…»), сопротивление стали собирает в reports/steelResistance.ts, замечания ядра для клиента очищает reports/clientWarnings.ts (значок предупреждения, имена полей запроса), реакции креплений, converged, governing_stage, постадийные stages[], warnings.

Усилия креплений отдаются в двух видах: `force_kN` — нормальная к стенке составляющая, `force_axial_kN` — осевое усилие в элементе (деление на cos наклона и угла в плане, умножение на шаг). Именно осевое печатают другие расчётные комплексы; сравнивать надо его ([0171]).

Проектные настройки, влияющие на физику (все — входы проекта, дефолты не меняют прежний результат):

- `pioneer.accounting` = "forces" (умолчание) | "fromBrovka" — учёт пионерного котлована ([0417-срез-по-силам-движок], 22.10.2026): «по силам» — давление тыла и лица по ломаной поверхности среза (core/fem/broken_surface_pressure.py, ответ несёт блок broken_surface с вкладами по этапам); «по грунту» (fromBrovka, [0384]) — массы откоса в давлении нет, поверхность плоская на уровне бровки, отметки приложения пригрузок пересчитываются от бровки. Прежние значения принимаются: soil, slopeStrip и forcesBroken читаются как forces (масса откоса учитывается, как у прежней призмы), fromBrovka остаётся только явным выбором. Удалены блок γ·h, полосы СП 22 и призма Кулона (pioneer_culmann.py); интерфейс — две кнопки «По силам» / «По грунту», старый проект переводится при открытии (core/pioneerAccounting.ts).
- `pioneer.pressure_engine`, `pioneer.allow_unconfirmed` — устарели ([0417-срез-по-силам-движок]): принимаются и игнорируются; движок по ломаной поверхности единственный. Отказ 422 (одной строкой, интерфейс проверяет те же сочетания до отправки): точечная пригрузка (point_load), φ слоя среза в (0; 9,1°) (phi_range), φ = 0 вне постановки опыта L11.4 (phi_zero), а также сейсмика, берма «с отпором» (выход — berms[].method = surcharge),

backfill_slope_deg, residual_strength_active, подсыпка тыла и free_standing_mode. Одно замечание в ответе — при срезе глубже 3,6 м или угле откоса вне 15–65°. Трапеция qA ≠ qB расширяется на подполосы до построителя и считается по тому же правилу. Правило подтверждено на ограниченном числе постановок; это сравнение с расчётами эталонной программы и независимым численным интегратором, а не аналитическое решение.

- `pioneer_degenerate_at_rest` — удалённый режим «давление покоя σa = σ0, когда призма пионерного среза вырождена» ([0377]). Поле остаётся в схеме как устаревшее: `false` и отсутствие поля равны, `true` — 422 «Режим «вырожденная призма» удалён; пересчитайте без него.» ([0412-строгий-вход]; прежний интерфейс слал поле только со значением true при включённом тумблере). Включение режима было измерено на 56 эталонах и ухудшало сверку (M/Q/w/Kз/Kм 49/49/43/47/49 → 45/44/41/42/44 из 56, [0239]). Пол под призмой — `min_active_ratio` (0,2·σ0 по умолчанию, [0166]) — не менялся.
- `restrained_at_rest` — «давление покоя при работающей связи (допущение)»: на раскреплённых этапах активное давление не ниже бытового P0, per-project, по умолчанию выключено ([0170]). Это НЕ «Корректировка давлений (P0 → Pp)» программы-эталона — та автоматическое сообщение окна результатов (замена бытового давления пределом в узлах), а не настройка; имя поля прежнее, поведение не менялось ([0296]). С [0377] переключатель убран из интерфейса (проект, сохранённый с ним, считается как раньше и показывает уведомление), в warnings ответа — пометка.
- `safety_model` = linearDeformable | rigidPlastic | auto — модель запаса в заделке; дефолт linearDeformable. `auto` (предельное равновесие при d/H > 1,4) не рекомендуется: на трёх отчётах проектировщика с d/H > 1,4 даёт Kз в допуске у 0 из 3 против 3 из 3; значение принимается для совместимости со старыми проектами, при нём ответ несёт предупреждение в `warnings` ([0296]).
- `water_model` = free | bound, `surcharge_full_depth` = auto | true | false — модель воды и глубина действия пригрузки (auto, с [0381] — на всю высоту стенки у полос любой ширины).
- `aquiclude_depth` — отметка водоупора (нижней границы водоносного горизонта) от поверхности, м; по умолчанию не задан. Ниже этой отметки свободной воды нет: порового давления нет (u = 0, σ'v полное; с 23.10.2026, [0426-вода-под-подошвой] — прежде оно замирало на значении у водоупора), вес грунта идёт полным насыщенным — локальный аналог режима связанной воды. При пионерном срезе отметка приводится к бровке, как котлован, вода и бермы. Влияет прежде всего на запас в заделке: на двух эталонных расчётах с водоупором 4,5 м учёт перевёл запас из −18 % в допуск при неизменных моменте, поперечной силе и прогибе ([0187]; историческое число по прежнему правилу замирания).
- `axial_model` = (не задан → forces) | none | self_weight | full — продольная сила в стенке для Kм по ф. 105; `f0_substage` = true | false — подэтап натяжения анкера «k.F0» (§ выше; false — без подэтапа); `strut_tension` = true | false — горизонтальная распорка работает и на растяжение (по умолчанию да; false — только сжатие); `seismic_intensity/k1/k2/kv_ratio` — сейсмика Мононобе-Окабе. Оба — подробно в §5.1 и §6.11.
- `subgrade_K`, `layers[].k_prop`, `subgrade_z_origin` — жёсткость основания, растущая с глубиной, C(z) = K·z′ ([0316-k-по-слоям], [0316-k-по-слоям-правки], 07.10.2026). Необязательный режим, выключен по умолчанию: запрос без новых полей и результат прежние бит-в-бит (полные ответы 10 прежних запросов на коде до и после правки

совпадают побайтно, golden-фикстура hydroweb/tests/fixtures/k_prop_legacy_golden.json, три набора сверки по строкам равны базам). subgrade_K — один K на запрос (переключатель «Жёсткость грунта определять по» в интерфейсе, [0169]); layers[].k_prop — K слоя, кН/м⁴ (0 < K ⩽ 1e6), внутри слоя жёсткость растёт непрерывно, на границе слоёв скачком; subgrade_z_origin = "pit_bottom" (по умолчанию, прежняя семантика subgrade_K) | "ground_surface". При pit_bottom ниже дна котлована каждого этапа C = K·(z − H), выше дна у тыла остаётся постоянный k_s слоя (нужен k_subgrade или E). При ground_surface z′ — глубина узла от поверхности геологии (верх стенки плюс глубина пионерного среза, ключ depth_datum_shift словаря soil): один закон C = K·z′ по обеим граням и всей стенке, от этапа не зависит, k_s и E слоёв не читаются; допустим только вместе с K (subgrade_K или k_prop слоёв).

- **Правила режима (явные, молчаливых приоритетов нет).** k_prop задан у всех слоёв или ни у одного (иначе 422 с именами слоёв без K — режим один на расчёт); k_prop вместе с subgrade_K — 422; ground_surface без K — 422; pit_bottom с k_prop требует k_subgrade или E у слоёв — иначе 422 (в ядре те же отказы — ValueError). Узел на границе слоёв относится к верхнему слою; при ground_surface обе грани узла берут K слоя тыла (одно z′ и одно K на узел) — прежде на узле z = H на границе слоёв лицо могло взять нижний слой по случайности округления суммы толщин. Пол жёсткости 1,0 — прежняя константа [0130], не новая. Классический POST /api/calculate эти поля, как и subgrade_K, молча игнорирует.
- **Источник и что в нём не названо.** СП 381.1325800.2018 п. 6.4.14 (текст прочитан: документы/нормативы/SP-381.1325800.2018.pdf, с. 57 PDF) допускает описывать основание моделью Винклера с коэффициентом постели, растущим линейно с глубиной и различным для участков стены выше и ниже уровня экскавации; значений K и начала отсчёта глубины пункт не называет. Выбор «от поверхности геологии» — допущение проектировщика, не вывод из свода; при ground_surface различия выше и ниже уровня экскавации режим не задаёт (K различается только между слоями). Допустимый диапазон K источником не установлен и программой не проверяется (только техническая граница 0 < K ⩽ 1e6). Ответ при включённом режиме (k_prop слоёв или ground_surface) несёт одно предупреждение о принятом режиме и начале отсчёта; при ground_surface проверка правдоподобия k_s не выполняется и «расчёт выполнен с этим значением» о нём не пишется (_check_soil_plausibility(layers, k_subgrade_used=False)). Источник значений K в подписи интерфейса («табл. В.1» СП 24.13330) не проверен: текста СП 24.13330.2021 в репозитории нет, а в журнале ([0316-k-по-слоям], «Не сделано», п. 3) отмечено расхождение номера таблицы: «В.1» в подписи против «Б.1» в записи [0235]; значение по умолчанию 12000 кН/м⁴ при включении переключателя (DEFAULT_SUBGRADE_K) источника не имеет, для режима по слоям умолчание не вводилось.
- **Фронт: только типы и хранение.** SoilLayer.kProp, WallState.subgradeZOrigin, FemRequestBody.layers[].k_prop и subgrade_z_origin (types.ts); в projectFieldFlow обе строки notSent («пока только хранение»). Ввода в интерфейсе и отправки в запрос нет: проект с такими полями при сохранении и загрузке их не теряет, но расчёт по ним не ведёт (projectExport.test.ts, projectFieldFlow.test.ts). Тексты интерфейса уточнены (subgradeProportionalityTexts.test.tsx, 3): при скалярном K ниже дна коэффициенты постели ИГЭ не используются, а выше дна у тыла действует постоянный коэффициент постели ИГЭ — прежняя формулировка «ИГЭ не используются» была неточной. Строка применимости

«subgrade_k» и признак subgradeProportional описывают только скалярный K (начало отсчёта — дно котлована каждого этапа); когда отправка kProp и subgradeZOrigin будет добавлена, строку и признак нужно пересмотреть.

- **Тесты:** hydroweb/tests/test_fem_layer_kprop.py (66): дефолты и описания полей; жёсткость узла по слоям (скачок, граница, пол, сдвиг начала отсчёта, независимость от этапа и подсыпки); эквивалентность (одинаковый K при pit_bottom ≡ subgrade_K; subgrade_K при ground_surface ≡ равный k_prop); приоритеты и 422; этапы; независимая проверка математики — свободная балка EI·w⁗ + C(z)·w = q с растущей жёсткостью против scipy.integrate.solve_bvp (w_max совпадает до $1{,}1\cdot10^{-4}$ при шаге 0,05 м, допуск теста $5\cdot10^{-4}$; при скачке жёсткости ошибка убывает с шагом без «полки»); мутации (неверные законы отличаются от продукта по w_max минимум в 28 раз сильнее допуска). Узел на границе слоёв при ground_surface не зависит от шума округления (четыре значения шума, семь значений H).
- `stages[].backfill_depth`, `stages[].water_back_depth/water_front_depth` — поэтапные подсыпка тыла и уровни воды; вывод флага затопления из уровней этапа. Подробно в §6.11.

Предупреждения (`warnings[]`) — сигнал, не блокировка. Помимо near-limit прогиба (> 10 % длины стенки), выведенного k_s и правдоподобности исходных грунтов, с `[0196]` добавлено вырождение призмы пионерного среза (`pioneer_thrust_degenerate` в pioneer_culmann.py): если максимум активного распора Pa(Z) клина по высоте ⩽ 0 — сцепление вдоль линии скольжения перекрывает вес призмы — клин применён вне своей области применимости, давление тыла определяется не им, а минимальным активным (доля бытового), и результат по моменту и прогибу инженеру следует считать оценочным. Физика при этом не меняется, warning — только сигнал.

Связи на входе (`[0378]`, блоки 1.2–1.3): `depth` анкера, распорки и грунтовой распорки отсчитывается от верха стенки (при срезе — от бровки) и может быть отрицательной (не меньше −100 м): отметка выше верха ограждения. Такую связь `calculate_fem` убирает из запроса до расчёта (api/main.py: _support_input_notes) и пишет в warnings «Отметка установки … вне ограждения»; без неё числа совпадают с расчётом без связи бит в бит (опыт L6.4 лестницы опытов: оставленная связь занижала прогиб на 62 %). Там же только текст: «Недостаточное расстояние между корнями яруса анкеров № N» (spacing ⩽ 1,0 м, ANCHOR_ROOT_SPACING_WARN_M; граница между 1 и 3 м не установлена), «Некорректная отметка установки анкера № N» (глубина больше дна предыдущего этапа, на этапе 1 — больше нуля) и «Давление на грунт № N вне границ геологии» (_surcharge_outside_geology_notes: полоса type=strip начинается над откосом среза и её заглубление меньше высоты поверхности грунта в начале полосы; равномерная пригрузка не проверяется). Числа эти замечания не затрагивают. Харнес: prod_sweep.support_depth_from_wall_top и заметка импортёра import_gwx._support_outside_notes.

Умолчания ввода (`[0379]`, блок 2): δ слоя без `delta_deg` при Кулоне — 0,33·φ (было φ/2; `SoilLayer.effective_delta_deg`, доля — `delta_default_ratio` слоя либо умолчание ядра); таблица Ka/Kp ответа берёт то же δ (`_table_delta_deg`; раньше `None → 0` при расчёте с φ/2). Прежнее φ/2 — поле запроса `delta_default_ratio = 0.5` (три запроса: /api/calculate, /api/calculate-staged, /api/calculate/fem). γ_sat без `gamma_sat` у связанной воды, слоя ниже

водоупора и в устойчивости откоса — γ + 2 как у свободной воды (`SoilLayer.gamma_sat_full`). Замечание «вода у лица не задана» (`_front_water_note`) теперь и при УГВ тыла между дном и носком стенки, с подсказкой «H + d». В warnings также замечания о принятом δ (`_delta_default_notes`) и γ_sat слоёв ниже воды (`_gamma_sat_default_notes`). v вне (0; 0,5) — 422 с текстом про слой (`_localize_validation_errors`). Клиент: `nuInputIssues`, `sectionInputIssues` в projectToRequest.ts останавливают расчёт до запроса (v вне интервала; двутавр без J или A; труба без D, t).

Берма по умолчанию (`[0380]`, п. 3.1 решений владельца 15.10.2026): `berms[].method` без значения — `"wedge"` (берма с отпором, core/fem/berm_wedge.py), а не `"surcharge"` (пригрузка на дне, `staging.berm_equivalent_surcharge`). Вне области применимости (сейсмика, `free_standing_mode`, несколько берм на этапе, подсыпка тыла ниже верха бермы, верх выше дна предыдущего этапа) этап НЕ отклоняется 422, а считается «пригрузкой на дне»: `stages_with_berm_wedge(..., fallback=True)` кладёт текст в `Stage.berm_fallback_note` (он идёт в warnings этапа и ответа), `narrow_surcharge_berm_notes` добавляет прежнее предупреждение о запасе у узкой бермы. Правило B ⩾ L снято ([0426-берма-срез], 23.10.2026): берма уже призмы отпора считается тем же клином (stages_with_berm_wedge; при ширине → 0 расчёт сходится к расчёту без бермы), в замечании — «берма уже призмы отпора, предел лица снижен клином»; срез «по силам» вместе с бермой «с отпором» считается (прежде 422; build_two_sided_spring_laws, stage_pressures(cap_bottom) обрезает полосы по реальному дну). Откат поэтапный: на этапах, где отпор применим, он сохраняется. В ядре режим по умолчанию остаётся строгим (BermWedgeRejected); откат включает ключ soil["berm_wedge_fallback"], который ставит API. Прежний расчёт — поле `method = "surcharge"` у бермы.

Поля запроса, возвращающие прежний расчёт ([0376]–[0393], 15.10.2026)

Расчёт действует по одному правилу: переменных окружения, переключающих поведение, нет. Прежний расчёт выбирается явным полем запроса; без поля действует новое.

Поле запроса	Что даёт
`delta_default_ratio = 0.5`	δ слоя без значения при Кулоне — φ/2 вместо 0,33·φ
`layers[].gamma_sat`	явное значение; без него γ_sat = γ + 2 (связанная вода, слой ниже водоупора, устойчивость откоса)
`berms[].method = "surcharge"`	берма — пригрузка на дне вместо отпора по клину Кулона
`surcharge_full_depth = false`	все полосы пригрузки обрезаются по дну
`f0_substage = false`	без подэтапа «k.F0»: релаксация преднатяга в одном равновесии с откопкой; память грунта включается только с первой подсыпки лица (без подсыпки — выключена)
`axial_model = "none"`	Км по ф. 41 СП 16 (продольной силы нет) вместо ф. 105
`strut_tension = false`	распорка между стенками только на сжатие, R ∈ [0, max_force]
`strut_force_limit = true`	max_force горизонтальной распорки ограничивает усилие, R ∈ [−Fm, +Fm]
`pioneer.accounting = "fromBrovka"`	«по грунту»: массы откоса в давлении нет, вместо «по силам»

Поле запроса	Что даёт
`legacy_blum_policy = true`	учебный режим Блюма: без отказа для схем со связями, срезом и этапами со связями, без класса А/Б и плашек; прогиб и Кз Блюма в ответе и вердикте; автоподбор с γc = 1,15
`water_mode = "pore_pressure"`	общая устойчивость: вес γ_sat и давление u·l вместо взвешенных весов (γ_sat − γw, порового давления нет)
`stages[].fill_soil`	подсыпка лица: грунт, поднимающий дно на этапе; без поля подъём дна — 422. ввод — столбец «ИГЭ подсыпки» таблицы этапов (StagesTab.tsx, поле этапа fillLayerId → projectToRequest.fillSoilPayload)
`pile_spacing`	шаг стоек (wall_type = beam | pipe); не присланный — 1,5 м

Переключателя нет у решений, не меняющих числа или убирающих мёртвый код ([0377]): удалённые решатели Пикара, penalty и active_set, слияние полос и тумблер вырожденной призмы — поля contact_strategy и pioneer_degenerate_at_rest принимаются только со значениями newton и false и ничего не меняют, остальные значения — 422 ([0412-строгий-вход]). Режимы «по силам (по ломаной)» и «по грунту от бровки» ([0384]) включаются явно и умолчание не меняют.

Подсыпка лица ([0426-подсыпка-лица], 23.10.2026). `stages[].fill_soil` — грунт «ИГЭ подсыпки» на этапе, где дно котлована поднимается (справка 8.4, п. 6.4.4). Колонка лица — подсыпка с историей этапов над природным грунтом ниже самой глубокой откопки (staging.plan_face_fill); пружина нового слоя начинается с нулевого давления при положении стенки в момент отсыпки; заделка для Кз — стенка ниже дна природного грунта. Импорт: Filling стороны котлована → fill_soil (scripts/gw_compare/import_gwx.py); во фронтенде поле этапа fillLayerId (types.ts) выбирается в столбце «ИГЭ подсыпки» таблицы этапов (StagesTab.tsx) и уходит в fill_soil через projectToRequest.fillSoilPayload. Не поддерживается: откопка глубже прежнего дна после подсыпки лица (механизм не описан; 422); бермы из грунта подсыпки не достроены.

Память грунта на всех этапах ([0426-память-грунта-на-всех-этапах], 23.10.2026). Закон давления с гистерезисом (возвратное отображение) действует на всех этапах, а не только в схеме анкера F0; контрольная задача 9.5.3 руководства 8.4 — давления 30, 40, 20, 20, 30 кПа при смещениях 0, 3, 1, 0, 1 см (закон без памяти: 30, 40, 40, 30, 40). Тест test_soil_memory_staging.py; часть сверенных чисел в тестах обновлена: прежние закрепляли значения без памяти.

Строгий вход: отказ 422 вместо тихой потери ([0412-строгий-вход], 20–21.10.2026)

Принцип «либо считается по заданному, либо отказ»: ввод, который расчёт прежде молча терял (нагрузка, связь, берма), на котором падал с HTTP 500 или зависал, теперь отклоняется кодом 422 с одной русской строкой. Расчётное ядро (hydroweb/core) не менялось; на допустимом вводе ответы 200 прежние побитно (корпус из 5756 тел состояний интерфейса: различий нет; не считая времени расчёта и проектов с вертикальной силой Fy ≠ 0, ниже). Общие механизмы: api/strict.py (StrictBody), api/refusal.py (тексты отказов и защита тела запроса), api/stage_refs.py (правила ссылок на этапы — одни для МКЭ и проверки общей устойчивости).

- **Схема строгая.** `StrictBody` (extra="forbid") стоит у FemCalculateRequest и всех вложенных схем, у запросов Блюма (/api/calculate, /api/auto-profile, /api/coefficients), этапов (/api/calculate-staged) и проверок (/api/checks/*): неизвестное поле — 422 «Неизвестное поле «x»: расчёт его не читает — уберите.». Устаревшие `contact_strategy` и `pioneer_degenerate_at_rest` остаются в схеме как deprecated: принимаются только newton и false и ничего не меняют (прежний интерфейс слал их в каждом запросе), остальное — 422; новый интерфейс их не шлёт. Перечень полей запроса МКЭ — frontend/src/services/femRequestFields.json (scripts/maintenance/dump_fem_request_fields.py); равенство схеме проверяет test_strict_input.py.
- **Формат отказа.** Запись ответа — `detail: [{type, loc, msg}]` (поля input и ctx убраны: сырой ввод в ответ не идёт). `msg` — ровно одна русская строка не длиннее 20 слов; английских имён полей и служебных слов в ней нет (названия полей — _FIELD_RU, значений перечней — _VALUE_RU в refusal.py, тесты перебирают все схемы), введённое значение усечено до 40 символов. Связь, берма, нагрузка и слой названы по параметрам («Анкер на глубине 1,8 м», «Слой «Суглинок»»), а не по номеру в запросе: в запрос идут только включённые строки, и номер не совпал бы с номером в таблице. Исключение по длине — отказ политики классического Блюма (класс B, [0390]). Стандартные ответы фреймворка и предзапрос CORS с чужого сайта (403 «Запрос с этого сайта не допускается.») тоже по-русски.

Что отклоняется (422)	Строка отказа (пример)
Этап установки анкера, распорки, грунтовой распорки, пригрузки, нагрузки на стенку, бермы больше числа этапов; снятие не позже установки (у бермы — по-прежнему замечание berm_removal); повтор обозначения связи	«Анкер на глубине 1,8 м: этап установки 3 больше числа этапов (2).»
Берма: верх не выше дна самого глубокого этапа или нет ширины основания; подсыпка тыла ниже дна этапа или ниже подошвы геологии	«Берма с верхом 4 м не выше дна котлована ни на одном этапе — берма не действует.»
Срез «по грунту» вместе с подсыпкой тыла (на любом этапе; прежде HTTP 500); при срезах «по силам», «полосами», «от бровки» подсыпка не глубже бровки (прежде молча не действовала)	«Этап 1: пионерный срез «по грунту» и подсыпка тыла вместе не поддерживаются: уберите одно из них.»
Подсыпка лица (stages[].fill_soil, [0426-подсыпка-лица], 23.10.2026): подъём дна без грунта; грунт без подъёма или на первом этапе; подсыпка вместе со срезом, бермами, K по глубине, водоносными слоями, свободно стоящей стенкой, сейсмикой; откопка глубже прежнего дна после подсыпки лица	«Этап N: откопка глубже прежнего дна после подсыпки лица: сочетание не поддерживается.»
Преднатяг анкера выше предельного усилия (прежде ValueError → 500)	«Анкер на глубине 1,8 м: преднатяг 700 кН больше предельного усилия 300 кН.»
Пригрузка: вид кроме uniform и strip; равномерная с двумя краями (intensity_far ≠ intensity); полоса без ширины; равномерная с отступом или шириной. Трапеция полосы (qA ≠ qB) больше не отказ: qA у ближнего к стенке края, qB у дальнего, 160 подполос (expand_trapezoid_strips) [0417-трапеция-скала-1022]	«Пригрузка 30 кПа: у равномерной нагрузки один край — для трапеции задайте полосу.»
Нагрузка на стенку: точечная ниже низа стенки (прежде	«Нагрузка на стенку на отметке 12 м ниже

Что отклоняется (422)	Строка отказа (пример)
прикладывалась к носку), распределённая целиком ниже низа или нулевой длины	низа стенки (11 м).»
Единицы и границы: ф ⩽ 85° и с ⩽ 20 000 кПа (PHI_MAX_DEG, COHESION_MAX_KPA в core/models.py — границы ввода, не норма; прежде 50° и 500 кПа; в схемах запроса, проверках устойчивости и анкеров, мастере), sigma_yield_MPa 1…2000 МПа, section_W 1…10^7 см³, section_J 1…10^9 см⁴/м; булево в числовом поле; NaN и ±Infinity	«Предел текучести 355000 вне диапазона 1 – 2 000 МПа: введите в МПа, не в кПа.»
Металл: двутавр без момента сопротивления; стойки и профиль вне каталога без предела текучести (пока проверка металла включена)	—
Классический Блюм и этапы Блюма: суммарная толщина слоёв не больше глубины котлована; признак проекта вне перечня политики [0390]; степень агрессивности среды вне справочника; повтор опоры этапа	—

- **Тело запроса.** `BodyGuardMiddleware` (чистый ASGI) разбирает тело всех POST/PUT/PATCH под /api/ до FastAPI: не больше 10 МБ (как client_max_body_size в nginx), вложенность не глубже 64 уровней, в объекте не больше 2000 полей, в списке не больше 50 000 элементов, во всём теле не больше 50 000 объектов и списков и 200 000 значений (счёт скобок и запятых по байтам до разбора: «Запрос не принят: слишком много объектов, списков или значений.»); невалидный UTF-8, не JSON, повтор ключа и одиночный суррогат — 422. `Content-Type: application/json` обязателен для непустого тела (в том числе на /api/calculate/fem и /api/calculate/fem/jobs; допустимы регистр, charset и суффикс +json, второй заголовок Content-Type — отказ): без типа тело принять нельзя, иначе отключилась бы защита FastAPI от запросов с чужого сайта. Допустимое тело на пределе числа узлов разбирается за 0,11–0,17 с (замер четвёртого раунда проверки).
- **Этап снятия.** `remove_stage` больше числа этапов — «до конца», как 0 (так считал расчёт и раньше; интерфейс сам доводит ссылки до n + 2 при удалении этапа); отказ — только для remove_stage ⩽ install_stage при remove_stage ≠ 0; install_stage > n остаётся отказом.
- **Вертикальная сила точечной нагрузки на стенку.** `buildFemRequest` кладёт `fy` в `point_loads_on_wall` (нулевая не шлётся; в реестре projectFieldFlow строка стала sent): сервер включает её в продольную силу стенки и запас по металлу Км (на примере проверки Км 20,58 → 13,28 при Fy = 500), Mmax и Qmax не меняются. Числа меняются только у проектов с Fy ≠ 0, Км — в запас; в репозитории таких проектов нет.
- **Интерфейс.** `stageReferenceIssues` (core/uiDefaults.ts) вызывается в doCalculateFem до отправки и находит те же ошибки, что сервер, с названием строки по её номеру в таблице («Анкер № 2: преднатяг 700 кН больше предельного усилия 500 кН.»); новой плашки нет — строки идут в существующий список ошибок, запрос не уходит. qB повторяет qA и отдельно не вводится (ячейка qB не редактируется). Остаются отказом сервера после отправки: износ коррозией больше толщины профиля (каталоги толщин интерфейса и сервера расходятся, например ECO 618 — 10,9 против 10,6 мм) и режим «по силам (по ломаной)» вне подтверждённой геометрии среза. Панели проверок (общая устойчивость, анкеры, фильтрация), подбор профиля, смета, вход и кабинет показывают первую строку отказа сервера (apiDetailText), а не «HTTP 422». Усилия связей в проверку общей устойчивости идут только у тех связей, что есть в supports.

- **Импорт проекта.** `resetDerivedResults` в App.tsx очищает результаты МКЭ и Блюма, ошибки, плашку отказа политики Блюма, ход счёта (считающийся расчёт отменяется), реквизиты прежнего проекта (setDocInfo(null)) и параметры выгрузки, после чего импорт ставит реквизиты из файла (normalizeProjectInfo(proj.projectInfo)) и карточку тома ТЭО (normalizeTeoCard(proj.teoCard)); «поколение» проекта растёт, поэтому расчёт, начатый до импорта, своего результата в новый проект не кладёт, а открытая вкладка перемонтируется и теряет результаты своих панелей. Новый проект и переключение проекта заменяют редактор целиком (key в AppWrapper).
- **Проверка и тесты.** Плохие запросы проверяющих (наборы 245, 75, 69, 166 и 170 проб): 0 × HTTP 500, 0 × 400, каждый отказ по-русски; перебор сочетаний возможностей, видов среза и числа этапов (scripts/maintenance/strict_input_combos.py): 9224 запроса, до правки 513 × 500, после — 0; 6216 состояний интерфейса: 0 × 500, ложных блокировок предпроверкой 0. Тесты: test_strict_input.py, test_strict_input_generative.py, test_strict_input_r2.py, test_strict_input_r4.py, test_refusal.py, test_combinations_never_500.py; vitest — uiDefaults, projectToRequest, checksInput, App.importReset. Мутационные сценарии — scripts/maintenance/strict_input_mutations.py (148 подмен).
- **Не закрыто.** Слои короче стенки в /api/calculate/fem принимаются без замечания (решение владельца). Отказ политики Блюма (класс B) длиннее 20 слов. Отказы, которые находит только сам расчёт (сходимость, геометрия, область формулы давления), у фоновой задачи приходят состоянием failed с кодом 422. Результаты завершённых задач /jobs хранятся час без лимита числа и объёма. Английские тексты входа, проектов, ассистента и реестра. Проверка общей устойчивости идёт без семафора и дедлайна.

POST /api/calculate/fem: статус, дедлайн, семафор, задачи ([0293-C3])

Эндпоинт — `async def calculate_fem_endpoint`: математика в синхронной `calculate_fem(req, ctl=None)` (её зовут тесты, сочетание нагрузок и фоновые задачи), она выполняется в потоке (`run_in_threadpool`) под RunControl. До счёта занимается «место» общего семафора (`api/calc_runtime.try_acquire_slot`): HYDROWEB_MAX_CONCURRENT_CALCS мест на МАШИНУ (по умолчанию 2), реализовано файловыми блокировками flock на N файлах-«местах» — счётчик в памяти процесса при двух воркерах не ограничил бы сумму, а ядро ОС снимает блокировку при смерти процесса (утёкшего места не бывает; где fcntl нет — откат на счётчик в процессе). Нет места — 429 с текстом «Сервер сейчас считает другие расчёты (одновременно допускается не более N). Повторите через 10 с.» и заголовком Retry-After: 10. `watch_disconnect` раз в 0,5 с спрашивает request.is_disconnected() и ставит флаг отмены — клиент ушёл, счёт прекращается (не проверено на реальном разрыве в uvicorn: тестировались наблюдатель на двойнике и ветка отмены).

Дедлайн: query-параметр `deadline_s` (ge=5, le=600; по умолчанию 240 с или HYDROWEB_CALC_DEADLINE_S, меньше 300 с у nginx). По исчерпании — HTTP 200 со статусом timeout, решёнными этапами и текстом причины вместо 504.

Поле ответа	Смысл
`status`	ok — сошёлся строго, признаков предела нет; soft — «мягкий приём» (строгий допуск не достигнут, остаток ниже инженерного порога; НЕ признак предела: бывает и при Kз 4–6); near_limit — сошёлся, но Kз ⩽ 1,10 / предохранитель этапа вернул решение в обход нефизичной

Поле ответа	Смысл
	ветви / узлы связей залочены / прогиб > 10 % длины стенки; mechanism — детектор механизма (lambda_limit < 1) либо несходимость при Кз ⩽ 1,10; not_converged — не сошёлся без признаков механизма; timeout — дедлайн исчерпан. Берётся первое сработавшее правило в порядке timeout → mechanism → not_converged → near_limit → soft → ok (api/calc_status.classify_fem_result).
`reliable`	True для ok и soft: числа можно использовать без дополнительной проверки.
`reason`	Русский текст причины без сырых WARNING_*; для mechanism — полный текст детектора (этап, «нагрузка выше предельной на X %», что делать).
`completed_stages`	Сколько этапов решено (при timeout меньше числа этапов).
`elapsed_s`	Длительность расчёта, с. НЕ детерминировано: сравнивая ответы целиком, исключайте (поправлены test_api_seismic_notes и test_load_combinations).
`stages[].mechanism_detected / lambda_limit / mechanism_message`	Данные детектора механизма (C2); null, пока этап не перепроверялся.
`stages[].stage_diagrams`	Эпюры этапа по узлам стенки для интерфейса ([0453-интерфейс-прототип]); null, если решатель этапа эпюр не дал. Поле только добавлено: расчёт не менялся, Mmax, Qmax, wmax, Кз, Км на запросах 1648.1, 1656, 2001.1, 1949.3 в ответе до и после правки равны. Состав — ниже.
`stages[].shear_elements`	Поэлементная эпюра Q этапа: по две точки на элемент ([0447]); максимум равен Qmax этапа.

Поэтапные эпюры: `stages[].stage_diagrams` (модель StageDiagramsOut в hydroweb/api/main.py, заполняется _stage_diagrams_of из полей StageResult в core/fem/staging.py). Массивы одной длины, по узлам стенки сверху вниз: `z` — глубина от верха стенки, м; `moment` — M, кН·м/м; `deflection_mm` — w, мм; `p_net` — p_back − p_front, кПа; `p_back` и `p_front` — расчётное давление грунта слева (тыл) и справа (котлован) после отсечки пределами, кПа; `lim_back_active` и `lim_back_passive` — предельные активное и пассивное давления тыла; `lim_front_passive` — предельное пассивное давление лица, null у узлов выше дна котлована (активный предел лица равен 0 и в массив не выводится). Источник — уже вычисленные решателем массивы и законы грунтовых связей (StageResult.soil_pressure_back, soil_pressure_front, limit_back_active, limit_back_passive, limit_front_passive; пределы берутся из SoilSpringLaw каждого узла). Новой физики нет; тест — hydroweb/tests/test_api_stage_diagrams.py.

Пороги — допущения проектировщика ядра, не норматив (СП 381 коэффициента запаса в заделке как критерия не знает); меняются в api/calc_status.py, не трогая решатель. Тест test_markers_exist_in_core_sources проверяет, что маркеры, по которым классифицируются тексты ядра («принято по инженерной точности», «работает у предела несущей», «ушёл на нефизичную»), по-прежнему есть в исходниках, — иначе классификация молча «ослепнет». Старые поля ответа не менялись.

Фоновые задачи (`api/jobs.py`, `api/routers/calc_jobs.py`). POST …/jobs валидирует тело той же схемой FemCalculateRequest (422 того же вида) и отвечает 202 {job_id, state: "queued"} с заголовком Location; расчёт идёт в фоновом потоке воркера-приёмника. Состояние — таблица calc_jobs в общей SQLite (воркеров два, опрос может прийти к другому): тикер раз в 0,5 с пишет progress и heartbeat_at и читает флаг отмены, поставленный DELETE у любого воркера. progress = {stage, stages_total, stages_done, iter, elapsed_s, est_total_s, est_basis}, где est_basis — measured (по времени решённых этапов) либо prior (априорная оценка cost_estimate, upper_s). Задача без свободного места ждёт в состоянии queued до 900 с, затем failed 503 «Очередь расчётов переполнена — повторите позже.» Лимиты: 20 активных всего, 3 на владельца (иначе 429 «В очереди слишком много расчётов…» с Retry-After: 10). Протухший heartbeat (> 60 с) — failed при ближайшем опросе, результат живёт 1 час, дедлайн по умолчанию 600 с (до 1800). result — ровно тот JSON, который вернул бы синхронный эндпоинт. Слежения за закрытием вкладки у задач нет (наблюдатель разрыва — только у синхронного эндпоинта): задача считает до конца или до дедлайна, пока клиент не пришлёт DELETE; клиент шлёт его при «Отмена» и при размонтировании App (выход из редактора). Права как у /api/calculate/fem (он не требует входа): неугадываемый id (128 бит); с действующей сессией задача записывается на пользователя, чужая — 404.

POST /api/calculate/fem: новые входы и выходы

- Входы (все — opt-in, дефолт не меняет прежний результат; решатель контакта один — Ньютон с [0297], раздел 6.13): `mesh_snap`; `aquifers[]` + `aquifer_mode`; `free_standing_mode`; `residual_strength_active` + `layers[].phi_res_deg/cohesion_res`; `load_combination` + `emergency_water_depth`, `water_gamma_f_back/front`; `stages[].backfill_depth` и `aquiclude_depth` (теперь доходят и из интерфейса). Описание — §6.13, §6.15.
- Выходы: `status, reliable, reason, completed_stages, elapsed_s`; `stages[].mechanism_*`; `support_checks` — по каждой связи осевое усилие на 1 элемент (как force_axial_kN), N_пред из запроса, K = N_пред/N_расч, определяющий этап, состояние (упруго / предел / не нагружена) и разбивка по этапам; связь, упёршаяся в предел (K = 1), проверку не проходит (api/fem_support_checks.py; ядро считало K_anchor_material/K_strut_material, API не отдавало); `load_combination`; `pore_pressure`; `coefficients` (таблица Ka/Kp/K0 из ядра).

Проверки вне МКЭ-расчёта: /api/checks/*

Три роутера подключены одной парой строк в конце main.py; датумы как у /api/calculate/fem (глубины котлована, уровней воды и водоупора — от натуральной поверхности; верх стенки — wall_top_depth = бровка пионерного среза); границы полей — Field(ge/le), NaN → 422, 500 — общий текст без деталей. Ни одна проверка не меняет расчёт стенки и не входит в вердикт интерфейса, DOCX и XML (решение владельца открыто).

- `POST /api/checks/hydraulic` (core/hydraulic_stability.py, 28 аналитических + 14 API-тестов). Взмыв дна напором нижележащего горизонта (ф. «γf,w·γw·(z_кр − z_п) ⩽ (γd/γn)·(γf,g·Σγ_i·h_i + γf,w,бл·γw·h_котл)»), фильтрационный выпор перед стенкой (столб грунта от дна до носка против u_н), градиент против критического i_кр = γ′/γw. Расчётные нагрузки: γf воды 1,1 / 0,9, грунта 1,0; особое сочетание п. 6.2.25 — 1,0 и аварийный

уровень. СП 381 формул проверок НЕ содержит: проверки построены на допущениях А1–А7 по аналогии — требуемое γn/γd взято из чисел п. 6.3.19 и ф. (6.13), относящихся в своде к сдвигу и опрокидыванию стен (А1); виды проверок UPL/HYD — возможный источник СП 248.1325800.2023 п. 7.2.2 (по тексту редакции 2023 относит UPL и HYD к предельным состояниям I группы; условия и коэффициенты в доступном тексте не найдены, не сверены); прочие допущения (баланс сил столба без трения по стенке; линейная потеря напора по кратчайшему контуру обтекания и консервативная граница head_loss_model="front_only"; i_доп задаёт проектировщик; не заданный уровень в котловане — осушён; γ_sat по умолчанию γ; вода над дном — γf 0,9) возвращаются в ответе. Вход: layers, wall_length, wall_top_depth, stages, water_table_*, aquiclude_depth, confined_aquifer {top_depth, piezometric_depth}, geotechnical_category, load_combination, emergency_water_depth, head_loss_model, allowable_gradient. Ответ: verdict, gamma_n, gamma_d, gamma_f, checks[] (key, status, factor, required, formula, norm_refs, assumptions, stages[]).

- `POST /api/checks/stability` (core/slope_stability.py; test_slope_stability.py — 45, test_api_checks_stability.py — 20, test_stability_ku_ladder.py — 35: сверка десяти опытов лестницы с печатным по фикстуре ku_ladder_cases.json). Круглоцилиндрические поверхности; Феллениус (N′ = W·cosα − u·l ⩾ 0) и упрощённый Бишоп (ширина отсека l·cosα, mα ⩾ 0,2 в зоне выхода); векторный счёт numpy; поиск — сетка центров и радиусов, шаблонное уточнение из 6 стартов, доводка на 64 отсеках, итог на 120; со стенкой — только поверхности через её нижнюю точку и ниже (п. 6.4.5); для открытого откоса — ещё семейство окружностей через подошву. Учёт по этапам: слои, вода своей стороны, водоупор, зоны напора, пригрузки, бермы, пионерный срез, подсыпка. [k_st] = γn·ψ/γd по СП 116.13330.2012 п. 5.2.2–5.2.3: γn по классу сооружения из structure_class.py (1,25 / 1,20 / 1,15 / 1,10), ψ = 1 и γd = 1 — допущение (для временного котлована ψ = 0,95 даёт меньшее значение; k_st_required в запросе задаёт значение вручную, в панели такого поля нет); в ответе k_st_source и k_st_norm. Вердикт (редакция 02.10) — по упрощённому Бишопу: verdict_method = bishop (по умолчанию) | min (меньший из двух, «в запас») | fellenius; норма метод не выбирает — СП 381 п. 6.1.22 требует минимума по поверхностям, а не по методам. classify_verdict: «проходит» — Бишоп ⩾ [k_st]; «не проходит» — Бишоп < [k_st]; «у предела» (ok = null) — Бишоп ⩾ [k_st], но на этапе Феллениус < [k_st], у критической поверхности Бишопа есть отсеки с mα < 0,2 и методы расходятся более чем на 30 % (порог ненормативный). Расхождение методов — не предупреждение, а справочная строка method_note; ответ несёт verdict, verdict_title, verdict_stage, verdict_note, method_note, k_st_norm. Opt-in: include_supports (связи как внешние силы, у анкера — доля корня за поверхностью скольжения) и wall_shear_capacity (срезающая сила стенки); выключено — результат побитово прежний. Режим воды ([0391-ку-режим-воды], решение N2 владельца 15.10.2026): поле water_mode = weighted (по умолчанию) | pore_pressure. «Взвешенные веса» — ниже уровня воды своей стороны γ_sat − γw, поровое давление не прикладывается (так считает эталонная программа; на лестнице опытов 31 строка с водой: −12,5…−42,6 % → −0,6…−3,6 % от печатного Ку); это менее консервативно, чем «поровое давление» (вес γ_sat и u·l): в сдвигающем моменте вес облегчён, в нормальной силе разность u·l·sin²α, в связном грунте Ку при воде может быть выше, чем без воды. При воде выше поверхности грунта (затопленный котлован, вода на откосе среза) этап считается как pore_pressure с замечанием; зона напора в weighted добавляет только превышение над гидростатикой, водоупор — облегчение веса лишь выше него. Прежнее: water_mode = pore_pressure в

запросе; в режиме pore_pressure числа прежние до последнего знака, без воды режимы дают одинаковые числа. Справочно рядом с вердиктом: Янбу упрощённый без поправки f0 (ΣFx = 0, не определён со связями, срезающей силой стенки и водой выше поверхности; в СП 116/СП 381 его нет, формулы эталона не опубликованы — подобран по печатным числам: −0,6…−1,0 % к печатному Янбу на лестнице) и reference — «Ку как у эталона» = меньшее из Феллениуса (у эталона без внешних сил печатается как «касательных сил», число то же, отдельного метода нет) и Янбу, с числом Бишопа и comparable. Ответ несёт water_mode, water_mode_by_stage, method_labels, reference, у этапов k_st_janbu, k_st_reference; поля methods, k_st_by_method, arcs прежние. Вердикт остаётся по Бишопу (основу вердикта выбирает владелец): при воде в режиме по умолчанию он менее консервативен прежнего, у эталона вердикт даёт reference, ниже Бишопа на 13–35 %. Валидация: бесконечный откос и плоский клин — 1e-9…1e-12; окружность против многоугольника — 1e-6, против независимых тонких полос — ⩽ 0,3 %; таблица Тейлора при φ = 0 (N = 0,2610 / 0,2191 / 0,1906 против 0,261 / 0,219 / 0,191); c = 0 — +0,03 % к tgφ/tgβ; поиск против перебора 113 тыс. окружностей — ⩽ 0,05 %; сверка 0,985 с «≈1,00» для откоса 2:1 — значение взято по памяти (гипотеза). Время: 3 этапа ≈ 2 с. Основание вердикта по умолчанию — замер на 64 расчётах (53 эталонных + 11 проектировщика, [k_st] = 1,20): Феллениус ниже Бишопа во всех 64 (медиана отношения 0,80; у 16 проектов с водой в грунте 0,50–0,70), Бишоп проходит все 64 (минимум 1,39), прежний вердикт «по меньшему» расходился с «по Бишопу» у 10 проектов, предупреждение о разрыве > 15 % горело у 49. Переклассификация по сохранённым ответам: 63 «проходит», 1 «у предела» (этап 3, Феллениус 1,11, Бишоп 1,62, разрыв 31 %), 0 «не проходит». Открыто: ψ и γd в [k_st] (приняты 1); порог 30 %; расчёты без вскрытия ниже бровки возвращают 422. Не реализовано: произвольная форма, Моргенштерн–Прайс, Шахунянц (в отчётах эталона не печатались), связи в Янбу и в справочном значении, трещины отрыва, сейсмика; не учитываются нагрузки на стенку, наклон засыпки, грунтовые распорки, слои воды с напором.

- `POST /api/checks/anchors` (core/anchor_capacity.py; test_anchor_capacity.py — 36 с эталонами ручного пересчёта, test_api_checks_anchors.py — 22, в том числе связка с реальным /api/calculate/fem). R_a;k: п. Л.2 ф. (Л.1)–(Л.5) (инъекционные: σ0g, A1, θ, K_p, γc 0,72 / 0,64, ξ0 0,43 / 0,55 / 0,72, Dc/Dк 0,3–0,9); п. Л.3 ф. (Л.6) (с давлением инъецирования: k 0,6, m_p); п. Л.4 ф. (Л.7)–(Л.8), табл. Л.1 (самозабуриваемые); либо по испытаниям (п. 7.3.19). Забивных анкеров прил. Л не даёт. Проверки: P_d ⩽ R_a;k/γa (п. 7.3.20, ф. (7.1)–(7.2), табл. 7.1: 1,50 / 1,75 / 1,05); тяга P_d ⩽ R_t;d/γa,t (п. 7.3.21, при заданном R_t;d); корень за границей призмы ⩾ 1 м по нормали (п. 7.3.8, 7.3.14 и), рис. 7.11); корень на глубине ⩾ 4 м (п. 7.3.14 б)); не в слабом грунте (п. 7.3.2; насыпной — предупреждение); рекомендации п. 7.3.14 в), м) — предупреждения. φ, c, E, ν — средневзвешенные по длине корня, γ_l·h_к — с взвешиванием. Допущения (печатаются по каждому анкеру): граница призмы — ломаная от низа стенки (опция — от дна котлована) под 45° + φ/2 в каждом слое; поверхность над корнем — верх стенки (грунт выше пионерного среза не учитывается, в запас); q в ф. (Л.2) = 0; P_d — огибающая осевого усилия МКЭ без γf (в API множитель load_factor; без него P_d = force_axial_kN ответа /api/calculate/fem либо нормальная составляющая ÷ cos α·cos β); УГВ — самый высокий «УВ слева»; m_p глинистых 0,2; γsat без задания — γ + 2. Состояния, которых проверка не выполняет (not_checked: оголовок, выдёргивание тяги из корня, общая устойчивость с анкерами, ослабление натяжения, разрушение конструкций анкерными

усилиями, деформации групп анкеров, отказ любого анкера, пп. 7.3.9–7.3.10), перечислены в ответе с причиной («методики в своде нет» или «в другой проверке»); статус каждого из восьми состояний п. 7.3.9 по данным проекта собирает клиент (core/anchorLimitStates.ts). Ответ: R_a;k, γa, R_a;d, P_d, запас, минимальная свободная длина (по призме и по 4 м), требуемая длина корня, промежуточные величины формул.

- `POST /api/checks/water` ([0460-вода-приток-водопонижение]; `routers/checks_water.py`, ядро — `core/water_inflow.py`, `core/fragment_method.py`). Приток воды в котлован. Вход: слои с `k_f` (м/сут; `k_f` задан — слой водоносный), УГВ, дно котлована, размеры в плане, длина стенки (ниже дна — завеса), «понижение ниже дна» (по умолчанию 0), радиус влияния (необязательный). Водоносная толща — подряд идущие помеченные слои, содержащие УГВ; T — от УГВ до подошвы; k — $\Sigma(k_i \cdot m_i)/\Sigma m_i$ ниже УГВ (пункт источника не указан — замечание `water.k_mean`). Схема A (без завесы): $Q = \pi \cdot k \cdot (H^2 - h^2)/\ln(R_k/r_0)$, $r_0 = \sqrt{(F/\pi)}$, $R_k = R + r_0$; схема C (шпунт ниже дна, метод фрагментов): $Q = k \cdot (H^2 - h^2)/(2 \cdot (\Phi_1+\Phi_2+\Phi_3))$, $\Phi_1 = \ln(R_k/r_0)/(2\pi)$, $\Phi_3 = d/(\pi \cdot r_0)$; радиус влияния, если не задан, — Кусакин $R = 2 \cdot S \cdot \sqrt{(k \cdot H)}$. Ответ: Q (м³/сут, м³/ч, л/с), S, R, r_0, T, k, Q без завесы и снижение, схема, формула, `source`, `notes` {code, line, explanation}. Источники по докстрингам модулей: СП 22.13330.2016 разд. 11, СП 45.13330.2017, Пособие к СНиП 2.06.14-85, Справочник гидрогеолога (Гавич, 1985); номера пунктов в модулях не указаны и не выдуманы — ответ несёт эту оговорку. Не выведено: схема B (нет источника), напорный горизонт (`compute_gap` ставит кровлю на уровень УГВ — для напорного неверно; УГВ вне водоносных слоёв — 422), `safety_factor` 1,3 и «+0,5 м» (API передаёт 1 и 0, без пункта нормы), подбор насосов `dewatering.py` (границы 10/50/200 м³/ч — практика). Числа модулей не менялись.
- **Проверки сечения распорки и обвязочного пояса ([0530-проверки-распорки-и-пояса-в-api]).** `POST /api/checks/struts` и `POST /api/checks/wales` (роутер `routers/checks_supports.py`, ядро `core/strut_check.py`, `core/wale_check.py`, `core/js_compat.py`). Перенос проверок по СП 16.13330.2017 с клиента на сервер один в один: формулы, коэффициенты, тексты блоков и порядок действий прежние, расхождения с СП 16 не исправлялись. Каждый маршрут принимает список позиций (1–200) и возвращает результаты в том же порядке: проект с несколькими распорками и подбор трубы в «Подборе решений» идут одним запросом. Позиция распорки: класс сечения, A, I, W, i, масса, E, нормативное `Ryn` (`Ry = Ryn/γm` считает сервер), высота двутавра по сортаменту (необязательно), длина, осевое усилие из МКЭ, момент (не задан — от собственного веса $q \cdot \cos\alpha \cdot L^2/8$), угол наклона, γc, γn, γm, μ, λ_max, L/n (умолчания 0,9; 1,0; 1,05; 1,0; 120; 250). Позиция пояса: A, W, `Ryn`, реакция F, шаг s, γc, γn, γm. Блоки распорки: центральное сжатие (прочность п. 7.1.1, устойчивость п. 7.1.3 с φ по табл. Д.1 для типа сечения по табл. 7; высота двутавра свыше 500 мм — тип a, примечание 1), внецентренное сжатие (прочность п. 9.1.1; устойчивость упрощённо $\varphi e = \varphi/(1 + m)$ вместо табл. Д.2, Д.3 п. 9.2.2), предельная гибкость (п. 10.4.1, табл. 32), прогиб от собственного веса. Блоки пояса: изгиб, срез, приведённое напряжение (схема одна: однопролётная балка с силой F посередине, $M = F \cdot s/4$, $Q = F/2$, $A_w = 0{,}5 \cdot A$). Ответ: блоки (имя, описание, формула с подстановкой, demand, capacity, использование, запас, passed), вердикт, наименьший запас, наибольшее использование, тип сечения и основание, подсказка к φ, λ, $\bar{\lambda}$, φ, Ry, N, M, σ, предел, предельное осевое усилие по устойчивости (`limit_force_stability_kN`) и по прочности A·Ry (`limit_force_strength_kN`, [0569-релиз-6-20-w-марка-ary]: Ry = Ryn/γm, тот же, что в блоке 2; от N, M, угла и длины не зависит; формулы проверки не менялись, поле считает

`check_strut` в `core/strut_check.py` и несёт `StrutCheckResult`, роутер его пробрасывает; сервер прежней версии поля не присылает, клиент тогда считает A·Ry сам); число, которое в JSON не записывается (бесконечность), отдаётся `null`. `js_compat.py` держит три правила JavaScript, которыми Python отличается: `toFixed` округляет точное двоичное значение вверх по модулю при равенстве, число в шаблонной строке без «.0», деление на ноль даёт ±Infinity или NaN; без них тексты блоков и округлённые числа не совпали бы с прежними. Вход строгий (`StrictBody`), отказ 422 одной русской строкой («Распорка № 2, площадь сечения: значение должно быть не меньше 0,1.»); границы шире любых значений окон: γc 0,3–1,5, γn 0,5–2, γm 1–1,5, μ 0,1–10, λ_max 10–1000, L/n 10–10 000, A не меньше 0,1 см², E 10^3–10^6 МПа, `Ryn` 10–2000 МПа, N по модулю не меньше 0,001 кН либо ровно 0 (при N порядка 10^{-300} эксцентриситет M/N бесконечен). Это единственное поведенческое различие с прежним клиентом: раньше считалось любое число, вплоть до E = 0. Названия полей для отказов — `hydroweb/api/refusal.py`, допустимые запросы для обхода «хулиганскими» значениями — `hydroweb/tests/_strict_bases.py` (`struts_main`, `wales_main`); у позиции пояса нет ни одного строкового значения, поэтому в обходе суррогатов (`test_strict_input_r2.py`) для неё проверяется только имя лишнего поля. `/api/checks/` — расчётный префикс: после пробного срока закрыт так же, как остальные проверки.

- **Клиент и сверка «клиент = сервер» ([0530-проверки-распорки-и-пояса-в-api]).** `services/sectionChecks.ts` собирает запрос из сечения проекта и усилий МКЭ (`planSectionChecks`), кеширует ответ по телу позиции, отправляет один запрос на список, при отказе на одной позиции запрашивает остальные по одной (ошибка остаётся у своей распорки) и переводит ответ в вид интерфейса (null → Infinity или NaN); `hooks/useSectionChecks.ts` — `useSupportSectionChecks` (карточка связей, вердикт, готовность к экспертизе) и `useServerCheck` (окна «Расчёт распорки» и «Расчёт обвязочного пояса»). Строки «Распорка N: прочность и устойчивость сечения» и «Обвязочный пояс распорки N» (`core/supportChecks.ts`) показывают ответ из `SupportContext.sectionChecks`; пока ответа нет — «Проверка не выполнялась: ответ сервера ещё не получен.», отказ — «Проверка не выполнялась: сервер отказал (причина).». Вкладка «Результаты», готовность к экспертизе, DOCX (`buildFemDocxReportDocument` запрашивает ответ сам) и XML отчёта МКЭ получают проверки с сервера; `pickStrutSection` в «Подборе решений» стала асинхронной и подбирает трубу одним запросом на весь список, при отказе сервера вариант получает причину. Прежние `calc/strutCheck.ts`, `calc/waleCheck.ts` остались опорой тестов. Общий файл `hydroweb/tests/data/strut_check_parity.json` (пересоздаётся `UPDATE_SECTION_PARITY=1 npx vitest run src/calc/sectionChecksParity.test.ts`): 23 случая распорки (9 труб, 7 двутавров, 3 швеллера, сдвоенный двутавр, коробчатое, пользовательское, малая площадь; длины 3,1–18 м, усилия 60–4200 кН, три наклона) и 10 случаев пояса; клиентский тест требует, чтобы прежний счёт давал файл, тест сервера — чтобы то же давал сервер, без допуска. В [0569-релиз-6-20-w-марка-ary] в «ожидаемое» каждого случая распорки добавлено только поле limit_force_strength_kN (записано прежним клиентским счётом, localSectionChecks.ts); остальное в файле побитно прежнее. Результат: 22 из 23 случаев распорки и все 10 случаев пояса побитно; в одном (сдвоенный двутавр, наклон 59°) момент от собственного веса различается в последнем разряде (2,928361788841043 против 2,9283617888410434), потому что `Math.cos` движка V8 и `cos` библиотеки C различаются в последнем разряде у 36 из 899 углов сетки 0,1°…89,9°. Вторая

причина различий — `Math.pow(L, 4)` в блоке прогиба: заменена цепочкой умножений ((L·L)·L)·L и на сервере, и в прежнем клиентском модуле. Тесты: `test_strut_check.py` (41), `test_wale_check.py` (5), `test_api_checks_supports.py` (29), `test_strut_check_parity.py` (12), `sectionChecksParity.test.ts`.

- **Проверки распорки и пояса: что оставлено как было, открыто.** Не исправлялось, решения владельца: (а) снимки сечений в проектах, сохранённых до 22.10.2026, несут прежнее расчётное Ry каталога (С235 — 230, С245 — 240, С255 — 250, С345 — 335 МПа, сталь 10 — 196 МПа вместо нормативных 235, 245, 255, 345 и 205), а сервер, как прежний клиент, делит его на γm ещё раз: сопротивление ниже на 2,0–2,9 % (сталь 10 — на 4,4 %), в запас; миграции в коде нет, сколько таких проектов на проде, не проверялось. (б) γn не входит в проверку распорки и делит Ry в проверке пояса. (в) Текст «γm = 1,025» в строке «Распорка N: прочность и устойчивость сечения» (экран, DOCX, XML) печатается при расчёте с γm = 1,05. (г) Снято в [0544-fm-распорки-по-устойчивости]: прежнее предельное усилие Fm = A·Ryn (`strutMaxForce`, поле `maxForce`, предел пружины в МКЭ) бралось без γm и γc, а проверка сечения шла с Ry = Ryn/γm и γc = 0,9, поэтому на одном сечении было два предела и Fm был выше предела проверки примерно в 1,17 раза. Теперь при выбранном сечении Fm — меньшее из A·Ry и φ·A·Ry·γc по ответу сервера (пункт «Fm распорки по устойчивости» ниже; A·Ry — поле limit_force_strength_kN ответа, с [0569-релиз-6-20-w-марка-ary]); собственный счёт A·Ry остаётся запасным значением, когда сервер поля не прислал или не ответил, и значением для строки без сечения. (д) γc = 0,9 — допущение программы, не значение из таблицы СП 16. (е) Обозначение «Ry» в блоке 1 (нормативное; предел печатается как Ry·γc/γm) и в блоках 2–4 (расчётное) — одно имя для двух величин. (ж) Пояс: знак реакции не учитывается, отрицательная F даёт отрицательные σ и τ и «проходит» (закреплено тестом `test_negative_reaction_passes_silently_known_gap`); σ_max (середина пролёта) и τ_max (опора) складываются в одном приведённом напряжении (в запас); A_w = 0,5·A; ссылки на пункты и формулы СП 16 в блоках взяты из клиентского кода и по тексту нормы в этой задаче не сверялись. (з) Блок 4: φe = φ/(1 + m), η = 1 — упрощение вместо табл. Д.2, Д.3; тип сечения коробчатого — b, пользовательского — c; φ при $\bar{\lambda}$ меньше 0,4 у типа c — 0,984; устойчивость только по сильной оси. (и) λ_max вводится, позиция табл. 32 не подбирается; прогиб — от собственного веса без влияния N. Формулировка «двойной γm в `strutCheck.ts`» из прежних записей журнала относится к состоянию до [0417-единое-правило-Ry] и для сечений, выбранных после 22.10.2026, устарела.

Узлы крепления: POST /api/joints/check ([0462-узлы-пилот])

Пилот «расчёт и чертёж узла крепления распорки-трубы к обвязочному поясу через фасонку» (на прод не идёт без решения владельца). Ядро — `hydroweb/core/joints.py`, роутер — `hydroweb/api/routers/joints.py` (строгий вход, 422 одной русской строкой, названия полей и значений — в `refusal.py`).

- **Вход.** Усилие — осевое из ответа МКЭ (`strut_forces[].force_axial_kN`, наибольшее по этапам; клиент берёт из расчёта и указывает этап), сечение — труба D×t из каталога, сталь — марка справочника `steel_grades` (R_yn, R_un по толщине), соединение — сварка или болты. Единственный тип узла — `strut_pipe_gusset`.

- **Швы (СП 16.13330.2017).** П. 14.1.16, формулы (176), (177): по металлу шва и по границе сплавления с переключением по β_f·R_wf/(β_z·R_wz); β_f, β_z — табл. 39; R_wf — табл. Г.2; R_wz = 0,45·R_un (табл. 4); конструктивные требования п. 14.1.7 а)–г), табл. 38; условие электрода п. 14.1.8; односторонние швы п. 14.1.9; требуемый катет — целые мм от табл. 38 до 1,2·t_min; требуемая расчётная длина.
- **Болты.** П. 14.2.9: формулы (186) срез, (187) смятие, (188) растяжение; γ_b — табл. 41 с примечаниями 1 и 2; R_bs, R_bt — табл. Г.5; R_bp — табл. Г.6; площади — табл. Г.9; число болтов — формула (189), коэффициент β при l > 16d, +10 % по п. 14.2.14; расстояния — табл. 40 (минимальные). Значения таблиц и виды формул сняты с текста и изображений страниц СП 16.13330.2017 (с. 10–11, 75–80, 82–84, 124–126; текстовый слой PDF формул не несёт).
- **Где таблица значения не даёт — отказ 422 или пометка «уточнить», значение не подбирается:** катет меньше 3 мм, положение шва вне табл. 39, a/d и s/d вне напечатанных диапазонов табл. 41, R_bt у классов 5.8 и 12.9, R_un < 360 при R_bp, R_yn > 440 для R_bp. Расхождения самого СП (R_bs класса 5.8: Г.5 даёт 210, табл. 5 — 205; R_bt класса 8.8: 451 и 448) оставлены по Г.5 с замечанием. Консервативные отказы (п. 14.1.7 б при гарантированном проваре; прим. 2 к табл. 40 для одноболтовых соединений t ⩽ 6 мм) оставлены, записаны как открытое.
- **Ответ.** K = предел/усилие, `ok`, требуемые катет или число болтов, проверки и замечания `{code, line, explanation}`, `not_checked`.
- **Клиент.** Кнопка «Узел» в таблице распорок (у распорок-труб; `AnchorsTab.tsx`), диалог `JointCheckDialog.tsx`, `core/jointCheck.ts`, сохранение в `HydrowebProject.jointChecks` (`services/jointChecksProject.ts`, не входит в отпечаток расчёта), раздел «Проверка узлов крепления» в записке (`modules/documentation/core/jointChecksSection.ts`, только если проверка сохранена). Чертёж — `modules/drawing/joints/jointDrawing.ts`: лист A3, SVG и DXF, масштаб из ряда ГОСТ 2.302, размеры по ГОСТ 2.307, обозначение шва по ГОСТ 2.312 (текста стандарта в проекте нет — вид знака принят по практике, уточнить).
- **Проверка независимым чтением.** Табл. 39 для автоматической и механизированной сварки d = 1,4–2 мм читалась неверно (β_f 0,9/0,9/0,8/0,7 вместо 0,9/0,8/0,7/0,7; β_z 1,05/1,05/1,05/1,0 вместо 1,05/1,0/1,0/1,0) — предел шва завышался до 14 % при катете 9–16 мм; исправлено, тест закрепляет значения по тексту страницы.
- **Не сделано / уточнить.** Фланцевый стык трубы (методики в СП 16 не найдено), смятие бетона (СП 63 в проекте нет), момент и поперечная сила (ф. (178)–(183), п. 14.2.11–14.2.13), стыковые и фрикционные соединения, максимальные расстояния табл. 40, сечение пояса на чертеже (условно), типы «пояс к шпунту» и «анкерная голова», правка чертежа пользователем.

Кабинет поставщика и каталоги поставщиков ([0461-кабинет-поставщика])

Роль `supplier` → свои позиции (восемь категорий, [0471]) → модерация администратором → публичный каталог → выбор у проектировщика. Расчёт и прежние каталоги не менялись.

- **Сервер.** Таблица `supplier_items` (`db/connection.py`, только добавляющая схема); роль `supplier` допустима в `admin_set_role`, приглашении и `InviteRequest`; `api/supplier_specs.py` — таблицы полей по типам (единый источник для проверки и формы: `GET /api/supplier/schema`), границы и согласованность;

`api/routers/supplier.py`: `/api/supplier/items` (CRUD, `submit`, `withdraw`; чужая позиция — 404, роль проверяется по БД через `require_role`), `/api/admin/supplier-items` (очередь, `approve`, `reject` с запиской; журнал `supplier_item_approve|reject`), публичный `/api/supplier-catalog`.

- **Слияние в выдачи.** Одобренные позиции действующих поставщиков (учётная запись активна и роль `supplier`) добавляются в конец `/api/profiles`, `/api/anchor-tendons`, `/api/rrsk/sections` (`pipes`) с `source: «поставщик: <организация>»` и `supplier_item: true`; без одобренных позиций выдачи прежние (тест сравнивает три выдачи и расчёты `/api/calculate`, `/api/calculate/fem` без и с позициями). Марки стали поставщиков (одобренные позиции прежнего вида «марка стали») отдаёт `/api/steel-grades` (глава 8). Правка позиции на модерации или в каталоге — только после «Отозвать»; не более 200 позиций на поставщика; удаление учётной записи удаляет позиции. Отказы — русские строки, без `str(e)`; названия полей добавлены в `refusal._FIELD_RU`.
- **Клиент.** `SupplierCabinetPage` (кнопка «Кабинет поставщика» на странице проектов; внизу страницы уведомление о принятии условий при первой отправке позиции), вкладка «Модерация каталога» в администрировании, `data/supplierCatalog.ts` (один запрос `/api/supplier-catalog` на сеанс; сбой оставляет встроенные списки), `findSheetPile` вместо `ECO_PROFILES.find` в 11 потребителях.

Кабинет v2 ([0471-кабинет-поставщика-v2]). Категорий восемь: шпунт, трубы, двутавры, швеллеры, тяги анкеров — в каталоги расчёта; винтовые и грунтовые анкеры, распорные системы, болты и крепёж — только в публичный каталог (`calc_used: false`). Схема полей с единицами, подсказками и признаком расчётного поля — `api/supplier_specs.py` (единый источник для проверки, формы и шаблона); проверки согласованности: Wx ≈ 2·Ix/h и масса ≈ 0,785·A у прокатных профилей, жёсткость или EA у анкеров, длина «от ⩽ до» у распорных систем.

- **Excel.** `api/supplier_xlsx.py`: шаблон (`GET /api/supplier/template.xlsx?kind=`), импорт (`POST /api/supplier/import.xlsx?kind=`), экспорт (`GET /api/supplier/export.xlsx`). Файл только .xlsx ⩽ 2 МБ (предел в `BodyGuardMiddleware`, `UPLOAD_PATH_LIMITS`, и повторно в обработчике), читается в памяти; до разбора проверяется устройство архива (число частей, распакованный объём, степень сжатия, макросы, внешние связи); формулы не принимаются; не более 200 строк. Ответ — `{created, rows, errors}`: верные строки становятся черновиками, ошибочные перечисляются построчно, отказ целого файла — 422; повторная загрузка отклоняет уже созданные позиции как дубликаты по наименованию. Ошибки — русской строкой, без `str(e)`.
- **Профиль, история, спрос.** Профиль — `GET/PUT /api/supplier/profile` (таблица `supplier_profiles`, ИНН 10 или 12 цифр с контрольными цифрами). Лента статусов — `supplier_items.history_json`; цена с датой и сроком, срок поставки, минимальная партия, длины шпунта — `detail_json`. Спрос — `supplier_demand (item_id, project_id)`: пересчитывается при сохранении проекта по ссылке `supplier:<id>` в любом месте данных проекта (так в спрос попадает и анкерная тяга из ячейки «Тяга»: её снимок `anchors.rows[].tendon` хранит id позиции) или по имени в полях выбора сечения стенки; повторное сохранение не накручивает, смена выбора и удаление проекта уменьшают.

- **Модерация и каталоги.** Фильтры по статусу, категории и поиск; массовое решение `POST /api/admin/supplier-items/bulk` (до 100 позиций, журнал — запись на позицию). Без одобренных позиций ответы `/api/profiles`, `/api/anchor-tendons`, `/api/rrsk/sections` и расчёта прежние (тесты, в том числе с одобренными позициями всех расчётных категорий: числа расчёта те же). Двутавры поставщика — в `ibeams`, швеллеры — отдельный ключ `channels` только при наличии одобренных.
- **Профили поставщиков в классическом расчёте и автоподборе** ([0541-поставщик-марки-и-блюм], 29.10.2026; формулы ядра, политика Блюма `blum_policy.py` и границы Pydantic не менялись, правился только источник данных каталогов). `calculate` и `auto_select_profile` стали async и считают в потоке (`run_in_threadpool`, как расчёт МКЭ); `_calculate_impl(req, suppliers)` ищет профиль запроса сначала во встроенном каталоге (прежний цикл, прежние числа; для встроенного профиля запрос в БД не делается вовсе), затем среди одобренных профилей поставщиков. Профиль поставщика строится в `SheetPileProfile` ядра из его позиции без пересчёта чисел (`supplier_specs.sheet_pile_from_row`) и считается тем же `check_profile` (Ry — по steel_grade_default и толщине профиля, как у встроенных). Ответ `/api/calculate` новых полей не получил; `/api/calculate-staged` вызывает `_calculate_impl` напрямую и видит только встроенные профили, как прежде.
- **Чтение профилей поставщиков: предел 1,5 с, кэш, откат.** `hydroweb/api/supplier_piles.py`: `PILES_TIMEOUT_S = 1.5` — отдельная задача и `asyncio.wait`, а не `wait_for` (отмена ждала бы очереди соединения до BUSY_TIMEOUT_S = 30 с); просроченное чтение дочитывается в фоне и закрывает соединение само; по истечении срока расчёт идёт без профилей поставщиков, в ответе автоподбора `supplier_profiles_skipped: true` и `supplier_profiles_note` («Профили поставщиков не учтены: их каталог сейчас не отвечает, подбор шёл по встроенным.»). Кэш в памяти процесса — `PILES_TTL_S = 60` с (на проде два воркера, у каждого свой кэш: одобренный или снятый профиль виден расчёту не позже чем через 60 с, а воркеру, принявшему действие, сразу — `reset_piles_cache` вызывают отзыв, удаление, модерация, смена роли и активности, удаление учётной записи, одобрение заявки поставщика); неудачное чтение запоминается на `PILES_FAIL_TTL_S = 10` с. Расчёт по профилю не из встроенного при недоступном каталоге — 503 «Каталог профилей поставщиков сейчас не отвечает: повторите расчёт или выберите встроенный профиль.», а не «не найден в каталоге». Проверка схемы (`_blum_precheck`: слои и политика Блюма) идёт до чтения БД и в calculate, и в auto-profile.
- **Автоподбор по профилям поставщиков.** Перебор идёт по объединённому списку [*встроенные, *поставщики] по возрастанию W (при равном W встроенный раньше: сортировка устойчива); подобран профиль поставщика — в ответ добавлены `source: «поставщик: <организация>»` и `supplier_item: true` (у встроенных этих полей нет, ответ прежний). Профиль без данных в подбор не входит (запись в журнал, один раз на пару id и updated_at). Подбор — синхронная `_auto_profile_impl` через `run_in_threadpool`; Ryn для публичной строки берётся из интервалов справочника напрямую (`steel_interval`), без `warnings.catch_warnings` в пути запроса; тест сверяет `steel_interval` с `get_steel_strength` на 14 толщинах × 7 марок.

- **Согласованность позиции: ввод, одобрение, расчёт.** Одно условие для трёх мест — `sheet_pile_issue` (толщины меньше половины высоты, W ≈ 2·I/h, W ⩾ 100 как в конструкторе `SheetPileProfile`, масса, A): его вызывают `_cross_sheet_pile` (при вводе, отправке и одобрении — `normalize_spec`) и `sheet_pile_gaps` (отбор в расчёт); отказ конструктора ядра — тоже причина («расчёт не принимает такое сечение»), а не «не найден». Профиль, которому не хватает A, W, I (в интерфейсе J), габаритов, толщин, массы, типа или известной марки стали, отвечает 422 одной строкой не длиннее 20 слов: «Профиль «X»: у поставщика нет A, W, J. Выберите другой профиль или обратитесь к поставщику.» (названия — A, W, J, ширина, высота, полка, стенка, масса, тип, марка; до пяти) или «Профиль «X»: данные не годятся: W меньше 100 см³/м. Выберите другой профиль или обратитесь к поставщику.»; проверка идёт по spec позиции как есть, поэтому одобренная строка, записанная до уточнения схемы, не роняет выдачу. Неизвестное имя — прежний текст «не найден в каталоге». `approved_public` стал построчным (одна позиция, которую нельзя показать, не прячет все позиции всех поставщиков; `json.dumps(item, allow_nan=False)` внутри построчного try), чтение одобренных строк вынесено в `_approved_rows` (общее для `approved_public` и `approved_sheet_piles`); в списках позиций (владелец, администратор) поля `calc_ready` и `calc_issue`.
- **Марка стали и толщина проката.** Марка профиля и толщина должны лежать в таблице справочника (`steel_interval`): при вводе и отправке 422 «Марка S235 при толщине 12 мм вне таблицы справочника (таблица до 4 мм): выберите другую марку.»; у уже одобренных позиций — `steel_table_note` в публичной строке и строка в `warnings` ответа расчёта (только у профилей поставщиков: Ry взято для ближайшего интервала, сопротивление нужно уточнить у поставщика). Имена списка интерфейса (`UI_ONLY_STEEL_GRADES`: S240GP, S270GP, S390GP, C245) зарезервированы в `builtin_names("steel")`: тест разбирает `KNOWN_STEEL_GRADES` в constants.ts и требует «список интерфейса ⊆ зарезервированные» и равенства «только интерфейсных» разности списков. Спрос: выбор марки поставщика в проекте (`wall.steelGrade`) считается в «выбрано в проектах» наравне с профилем и трубой.
- **Клиент ([0541]).** `data/supplierCatalog.ts`: `findSteelGrade(id, profileName?)` — список интерфейса, затем марки поставщиков, затем собственная марка выбранного профиля поставщика (первой берётся `steelRyn` профиля: Ryn по толщине проката, то же число у классического пути и МКЭ), `supplierSteelGrades()`, `supplierSheetPileByName()`; профиль поставщика несёт `steelGrade` и `steelRyn`. `RetainingWallTab.tsx`: после встроенных марок — группа «Марки поставщиков» со строкой «<марка> (Ry=… МПа) — поставщик: <организация>», собственная марка профиля поставщика, которой нет в списке интерфейса (S235, S355, S390 справочника ядра), — строкой «<марка> (по профилю поставщика) (Ry=… МПа)» с Ryn по толщине профиля; автоподбор: профиль не найден у клиента или сервер назвал поставщика — каталог перечитывается, не нашёлся и после — проект не меняется (прежние J и A не остаются), в тексте итога «(поставщик: …)» и «Профили поставщиков не учтены.». `core/uiDefaults.ts`: смена профиля на профиль поставщика берёт его марку (`catalogueSteelGrade` читает позицию; профили Л5-УМ и ECO — как раньше), обратно — каталожную. Нормативное Ryn марки поставщика уходит в запрос МКЭ как `sigma_yield_MPa` (`wallRyn` в `services/projectToRequest.ts`), его же берут замечания о марке, `core/expertiseReadiness.ts` и таблицы DOCX и XLSX (`reports/generateDocx.ts`, `generateXlsx.ts`, `steelResistance.ts`: `findSteelGrade`

вместо `KNOWN_STEEL_GRADES.find`, иначе печаталось бы «355» вместо Ry поставщика). Модуль упругости E марки поставщика в E стенки не переносится.

- **Проверка и тесты ([0541]).** Числа встроенных каталогов и расчётов не менялись: снимок до и после (`/api/profiles`, `/api/anchor-tendons`, `/api/rrsk/sections`, `/api/supplier-catalog`; `/api/calculate` и `/api/auto-profile` на восьми запросах; `/api/calculate/fem` на сохранённых запросах 1, 2, 3, 4, 5, 11 набора `scripts/gw_compare/refs_requests`) — 26 ответов, файлы равны побайтно (без полей elapsed_*); смоук `deploy/smoke_check.py` — 7/7. `hydroweb/tests/test_supplier_calc_catalog.py` (расчёт профиля поставщика равен расчёту встроенного профиля с той же геометрией, неполный профиль и неизвестная марка — 422, при наличии профилей поставщиков встроенные расчёт и подбор дают прежние ответы, автоподбор берёт профиль поставщика с меньшим W, пропускает неполный, предел 1,5 с и откат, кэш, 503, спрос на марку); vitest: `supplierCatalog.test.ts`, `uiDefaults.test.ts`, `expertiseReadiness.test.ts`, `projectToRequest.supplier.test.ts`, `RetainingWallTab.supplier.test.tsx`. Фикстуры `PUBLIC_STEEL`, `PUBLIC_CATALOG_STEEL` в `testing/supplierFixtures.ts`.
- **Открыто.** Расчёт МКЭ с профилем поставщика не видит его в каталоге на сервере: износ по СП 28 к нему не применяется (в замечаниях при заданном износе стоит «Коррозионный износ НЕ учтён: …»), сечение и Ryn берутся из запроса; проверки Pydantic `FemCalculateRequest` синхронны и в БД не ходят. «Подбор решений» профили поставщиков не предлагает. `steel_grade_default` профиля поставщика выбирается из справочника ядра (S235, S255, S345, S355, S355GP, S390, S430GP), собственная марка поставщика в профиль не подставляется; список интерфейса и справочник ядра расходятся (интерфейс — S240GP, S270GP, S390GP, C245; ядро — S235, S355, S390), поэтому для марок, которых нет в списке интерфейса, интерфейс берёт Ryn сервера. Классический Блюм и автоподбор по профилю поставщика наследуют политику [0390] (класс B — 422 «рассчитайте МКЭ»). Использованное `sigma_yield_MPa` вместе с результатом не хранится.
- **Открыто (кабинет поставщика).** Условия кабинета поставщика приняты действием (первая отправка позиции или кнопка «Принимаю условия» в разделе «Моя продукция в проектах») и записываются в consents, действует редакция документов 2026-10-29 (разделы «Регистрация по ролям» и «Правовые документы» ниже); организация указывается в позиции, не привязана к учётной записи; нет загрузки файлов документа; выбора швеллера и информационных позиций (анкеры, распорные системы, крепёж) у проектировщика и в спецификации ТЭО пока нет.

Регистрация по ролям и заявки поставщиков ([0485-регистрация-ролей])

При регистрации по почте и после первого входа через Яндекс человек выбирает, кто он: проектировщик или поставщик. Проектировщик работает как прежде. Поставщик прикладывает карточку организации и до одобрения заявки администратором остаётся «на проверке»: вход возможен, платформа закрыта. Расчётное ядро не менялось.

- **Вид учётной записи.** `users.account_kind` (designer | supplier | NULL, глава 3); `auth.account_kind_of` и `auth.supplier_pending` (account_kind = supplier при роли ниже supplier и admin); `/api/auth/me` и ответ входа несут `accountKind`. Клиент выбирает экран в `AppWrapper.tsx`: first-login, supplier-status («Заявка на проверке»), account-kind (выбор

роли после первого входа через Яндекс), затем вопрос о профессии. Приглашённый администратором получает вид при создании, выбора ему не предлагают.

- **Закрытая платформа.** Пока заявка не одобрена, `require_role` (кабинет поставщика, запросы цены, администрирование), `_get_user_id` (все маршруты проектов), `projects/import-file` и ассистент отвечают 403 «Раздел закрыт: заявка поставщика ещё не одобрена администратором.» (`SUPPLIER_PENDING_MESSAGE`); профиль, выход и собственная заявка доступны. Роль читается из базы, а не из токена: одобрение действует в уже открытой сессии. Смена роли администратором на любую, кроме supplier и admin, снимает «на проверке» (`admin_set_role`). Расчётные маршруты входа не требовали и не затронуты.
- **Заявка.** `clean_fields`: наименование 2–120 знаков; ИНН проверяет `inn_valid` кабинета поставщика (10 или 12 цифр, контрольные цифры); контактное лицо 2–120; телефон 10–15 цифр при длине до 30 (допустимы пробелы, «+», скобки и дефис); сайт необязателен (`clean_website`: без схемы дописывается https://, логин и пароль в адресе, пробелы и схема не http и не https дают отказ). Одна активная заявка на пользователя (частичный уникальный индекс), `APPLICATIONS_PER_DAY` = 5 в сутки; подача и «Продолжить как проектировщик»: 30 за 10 минут (корзина supplier_application); замена файла: 10 в час с адреса; регистрация: прежние 5 за 10 минут.
- **Файл карточки.** PDF, JPG, PNG или DOCX не больше `MAX_CARD_BYTES` (10 МБ). Тип определяет `detect_card_type` по сигнатуре (`%PDF-`, JPEG, PNG; DOCX: ZIP с `[Content_Types].xml` и `word/document.xml`, иной архив не принимается), расширение обязано ей соответствовать (`check_card`). Файл хранится в `supplier_cards/` рядом с базой (том данных контейнера, не `frontend/public`) под случайным именем `<32 hex>.<тип>`, права 0600, каталог 0700; исходное имя, очищенное от пути и служебных символов, хранится только для показа. Путь из запроса не принимается: имя берётся из базы и проверяется по шаблону и по каталогу (`card_path`). Отдаёт файл только администратор, всегда вложением (`Content-Disposition: attachment`, nosniff, no-store).
- **Тело запроса.** Три маршрута загрузки названы в `refusal.UPLOAD_PATH_LIMITS` (предел 10 МБ + 256 КБ служебных частей) с текстом «Файл больше 10 МБ.»; `_read_capped` читает тело не дальше предела; часть data не больше 16 КБ; форма разбирается в памяти (`parse_form` в `routers/project_import.py`, тот же разбор у импорта проектов). nginx: `client_max_body_size 11m` на уровне server и в `location /api/` (`deploy/nginx-site.conf`; тест `test_nginx_site_conf.py` держит предел не меньше самого большого из `UPLOAD_PATH_LIMITS`) и 11M в `deploy/nginx-host.conf` (на сервере файл правят вручную и перезагружают nginx: тело формы с файлом ровно 10 МБ больше 10M).
- **Решение.** `_decide`: один раз (`UPDATE … WHERE status = 'new'`, повторное решение даёт 409); отклонение требует причину 3–500 знаков (`NOTE_MAX`), одобрение принимает записку по желанию; одобрение пишет роль supplier, account_kind = supplier, onboarding_completed и профиль организации (`INSERT … ON CONFLICT(user_id) DO UPDATE`: название, ИНН, контактное лицо и телефон; регионы, основание и почта профиля не меняются); администратору роль supplier не назначается (409). Действия пишутся в журнал: `supplier_application_approve`, `supplier_application_reject`, `supplier_application_designer` (исполнитель последнего: сам пользователь). Письмо о решении уходит владельцу заявки фоновой задачей после ответа

(`supplier_notify.notify_application_decided`; текст — `email_templates.application_decided`; сбой почты ответ не меняет); о самой заявке администратору письма нет. Решение видно и на экране, счётчик новых заявок отдаёт `GET /api/admin/supplier-applications/summary` (заголовок администрирования, вкладка «Заявки поставщиков» и кнопка «Администрирование» в списке проектов).

- **Удаление и копии.** `auth.delete_account` удаляет заявки пользователя (`forget_user_applications`) и после фиксации базы стирает файлы (`remove_card_files`). Строки таблицы входят в `backup_db.sh` и возвращаются `restore_db.sh`; файлы карточек лежат в каталоге тома, который копирует `backup_data.sh`. Восстановление строк из копии без тома оставляет заявки без файла (администратор увидит «Файл карточки не найден на сервере.»).
- **Тесты.** `hydroweb/tests/test_supplier_applications.py` (76: регистрация по ролям, подача и замена файла, тип по сигнатуре и размер ровно 10 МБ и 10 МБ + 1 байт, права и 403 до одобрения, одобрение и отклонение, «Продолжить как проектировщик», лимиты, удаление, миграция старой базы); vitest: `AccountKindScreen`, `SupplierApplicationStatus`, `AdminApplicationsTab`, `supplierApplication.test.ts`, `AppWrapper.roles.test.tsx`, `AuthPage.consents` (первый шаг выбора роли).
- **Открыто.** Письма о новой заявке администратору нет (счётчик на экране); ссылка в письме о решении ведёт на главную страницу платформы; карточка только скачивается (показ в браузере потребовал бы ослабить защиту ответа); заявки неподтверждённых адресов не чистятся, пока их не удалит администратор; в `supplier_profiles` нет поля сайта (он остаётся в заявке); поставщик «на проверке» в таблице пользователей показан как «Стандартный».

Правовые документы: редакция 2026-10-29 и журнал согласий ([0484-правовые-тексты-цены], [0499-правовые-тексты-сводка-поставщика], [0520-обращения-с-вложением-и-редакция-2026-10-27], [0531-обращения-до-пяти-вложений], [0552-правовая-редакция-2026-10-29])

Версия редакции стоит в трёх местах, они равны друг другу: `LEGAL_DOC_VERSION` в `hydroweb/api/auth.py`, `LEGAL_VERSION` в `frontend/src/legal.ts` и «версия 2026-10-29» на пяти страницах `frontend/public/docs/` (`privacy`, `consent-pd`, `consent-mailing`, `consent-suppliers`, `supplier-terms`; сверяет `test_consents.py`). Источник страниц: `scripts/legal/build_legal_pages.py` (правка текста — в генераторе, затем `python3 scripts/legal/build_legal_pages.py`); лицензионное соглашение `license.html` и сертификат `certificate.html` по тексту не менялись. Изменился состав данных или получатели: обновите тексты и версию.

- **Согласие на передачу данных поставщикам.** `consent-suppliers.html`; формулировка отметки в диалоге «Запрос цены» `SUPPLIER_CONSENT_LABEL` дословно равна тексту на странице (в `legal.ts` и в генераторе; сверяет `test_consents.py`), под отметкой `PriceRequestDialog.tsx` показывает адрес, который будет передан, и ссылку на страницу. Согласие даётся на каждый запрос, пишется в `consents` (`kind = price_request_contacts`, `source = price_request:<id>`) в одной транзакции с запросом; отказанный запрос (422) записи не оставляет. Проектировщику поставщик отдаётся только названием организации:

ключи ответов `price-requests` зафиксированы тестом `test_designer_sees_only_the_organization_name_of_a_supplier`.

- **Условия кабинета поставщика.** `supplier-terms.html`; принятие происходит действием: `submit` в `routers/supplier.py` вызывает `_record_supplier_terms` и пишет `kind = supplier_terms`, `source = supplier-cabinet` не чаще одного раза на редакцию документов; черновики, импорт из Excel и отказанная отправка записи не оставляют. Ту же запись пишет `POST /api/supplier/accept-terms` (кнопка «Принимаю условия» в разделе «Моя продукция в проектах»). В кабинете внизу страницы стоит уведомление «Отправляя позицию на модерацию, вы принимаете условия использования кабинета поставщика» со ссылкой.
- **Профиль и выгрузка.** Записи видны в блоке «Согласия» профиля с подписями «Передача данных поставщикам (запрос цены)» и «Условия использования кабинета поставщика» (`ProfilePage.tsx`, `HISTORY_TITLES`), входят в выгрузку «Мои данные» и удаляются вместе с учётной записью. Уже записанные согласия остаются с прежней версией; повторное согласие у действующих пользователей не запрашивается.
- **Заявка поставщика в текстах.** `privacy.html` и `consent-pd.html` называют сведения заявки на доступ поставщика: название организации, ИНН, контактное лицо, телефон, сайт и файл карточки организации (политика называет ещё решение администратора и причину отказа). Пункт 2.2 в `supplier-terms.html` описывает заявку: поставщик выбирает роль при регистрации и прикладывает карточку организации (PDF, JPG, PNG или DOCX до 10 МБ), администратор одобряет заявку или отклоняет её с причиной, до одобрения кабинет и расчётные разделы недоступны. Срок `hw_session` в таблице cookie политики — до 14 суток без активности с продлением при работе, как в коде (`TOKEN_EXPIRE_HOURS` = 336).
- **Редакция 2026-10-27 ([0520-обращения-с-вложением-и-редакция-2026-10-27]).** Правка по решениям владельца 27.10: пробный доступ по метке ссылки и обращения «Написать нам». `privacy.html`: понятия «Метка ссылки» и «Пробный период» (раздел 2), новая строка таблицы целей «Пробный доступ по ссылке с меткой» (метка, время перехода, срок пробного периода) и переписанная строка об обращениях (раздел, текст, вложение, отметка администратора); пункт 8.4.3 (обращения видят только администраторы), новые пункты 8.4.5 (обращения «Написать нам») и 8.4.6 (метка, запись hydroweb_from в браузере, кука hw_oauth, продление); пункт 8.6 (срок хранения обращений и вложений: пока существует учётная запись, отметка «обработано» срок не меняет; метка, время и дата окончания пробного периода: так же); раздел 9 (вводный абзац больше не говорит «только технически необходимые», новая строка локального хранилища hydroweb_from, расширено назначение hw_oauth с меткой на 10 минут, последний абзац); раздел 10 (мера о вложениях). `consent-pd.html`: новый пункт данных (метка, время перехода, дата окончания пробного периода), расширен пункт «иные сведения» (обращения «Написать нам» и приложенный файл), две цели (пробный доступ; обработка обращений с файлами). Остальные три страницы: только номер версии и дата. Страницы перегенерированы `scripts/legal/build_legal_pages.py`; тексты и «было → стало» лежат в `документы/правовые_документы/редакция_2026-10-27/`.
- **Редакция 2026-10-28 ([0531-обращения-до-пяти-вложений]).** Правка по решению владельца 28.10: форма «Написать нам» принимает до пяти файлов вместо одного. Текст правится только в генераторе `scripts/legal/build_legal_pages.py` (`LEGAL_VERSION`, `DATE_RU` «28 октября 2026 г.»): `privacy.html`, раздел 6 (строка об обращениях: до пяти

приложенных файлов, каждый до 10 МБ, имя и размер каждого; «Текст и файл» заменено на «Текст и файлы»), пункт 8.4.5 («до пяти файлов»; администраторы читают обращение и скачивают файлы; файлы хранятся под случайными именами, тип каждого файла проверяется по содержимому), `consent-pd.html` («приложенные мной файлы»). Остальные страницы изменились только номером и датой. `LEGAL_DOC_VERSION` (`auth.py`) = `LEGAL_VERSION` (`legal.ts`) = «версия 2026-10-28» на пяти страницах; `scripts/books/docs_capture_common.py` — та же версия. Срок хранения не менялся. Текстовые копии и перечень «было → стало» одной строкой — `документы/правовые_документы/редакция_2026-10-28/` (проект до утверждения юристом).

- **Редакция 2026-10-29 ([0552-правовая-редакция-2026-10-29]).** Правка по решению владельца 29.10: в политику внесена строка о журнале событий, который платформа ведёт в период тестирования ([0543-журнал-событий-беты]). Слов «бета» и «beta» в правовых текстах нет. Текст правится только в генераторе `scripts/legal/build_legal_pages.py` (`LEGAL_VERSION` «2026-10-29», `DATE_RU` «29 октября 2026 г.»): `privacy.html`, пункт 8.4.7 (одна фраза: идентификатор пользователя, время и вид действия, служебные признаки без содержимого проектов и файлов; цель — оценка работы платформы и проверка критериев выхода из тестирования; хранится, пока существует учётная запись, удаляется вместе с ней, виден только администраторам платформы, не передаётся поставщикам и иным третьим лицам), раздел 6 (новая строка таблицы целей: состав, основание «осуществление прав и законных интересов Оператора», виды обработки; основание выбрано без указаний владельца — вопрос юристу), пункт 8.6 (срок хранения журнала). Остальные страницы изменились только номером и датой; согласие на обработку персональных данных не менялось. Дополнение 29.10 вечером ([0563-письма-политика-тексты-книги], решение владельца): в ту же редакцию, без смены версии, в строку раздела 6 «Направление служебных сообщений» добавлены уведомления по событиям учётной записи (запросы цены, ответы на них, решения по позициям и заявкам поставщика); формулировки «отписаться нельзя» нет, строка в письма не включена (`POLICY_LINE` = `None`); согласия по тексту не менялись, вопросы юристу — в README каталога редакции, разделы 5 и 6. `LEGAL_DOC_VERSION` (`auth.py`) = `LEGAL_VERSION` (`legal.ts`) = «версия 2026-10-29» на пяти страницах; `scripts/books/docs_capture_common.py` — та же версия; уже записанные согласия остаются с прежней версией, повторное согласие не запрашивается. Текстовые копии, «было → стало» и вопросы юристу (основание, срок цели относительно периода тестирования, перечень журнала в согласии, идентификатор пользователя как персональные данные) — `документы/правовые_документы/редакция_2026-10-29/` (проект до утверждения юристом). Тесты: `test_consents.py` (редакция в генераторе, на пяти страницах и в текстовых копиях; строка раздела 6, пункт 8.4.7 одной фразой, пункт 8.6).
- **Сводка спроса в текстах.** С редакции 2026-10-26 тексты описывают обезличенную сводку спроса для поставщиков. В `privacy.html`: понятие в разделе 2, строка таблицы целей в разделе 6 (что записывается и что нет, основание), пункт 8.4.4 (кто что видит; обезличивание, но не анонимизация: оператор связывает запись с проектом по идентификатору проекта; при одном проекте в месяце или регионе поставщик видит числа этого проекта, порога нет), срок в пункте 8.6 и мера в разделе 10. В `supplier-terms.html`: раздел 10 (сводка ориентировочная, запрет устанавливать по ней проект или заказчика, использовать её вне оценки спроса и предложения своей продукции, публиковать и

передавать), заключительные положения стали разделом 11; пункты, на которые ссылается книга поставщика, не сдвинулись. `consent-suppliers.html` уточняет, что сводка этим согласием не охватывается.

- **Открыто.** Редакции 2026-10-24, 2026-10-26, 2026-10-27, 2026-10-28 и 2026-10-29 остаются проектами до утверждения юристом; вопросы (форма согласия, публикация названия организации поставщика-предпринимателя в открытых каталогах, статус предложения, принятие условий действием вместо флажка, хранение записей о согласиях после удаления учётной записи, срок исполнения отзыва) собраны в `документы/правовые_документы/редакция_2026-10-24/README.md`, раздел 8, вопросы по сводке спроса (основание обработки для проектировщика, порог малых групп, принятие новой редакции условий до показа сводки) в `документы/правовые_документы/редакция_2026-10-26/README.md`, раздел 8. Вопросы к редакции 2026-10-27 (Ю-1…Ю-5 и два вопроса владельцу: убрать ли метку из учётной записи, вход через VK ID в политике не описан с прежних редакций) собраны в `документы/правовые_документы/редакция_2026-10-27/README.md`. Отзыв согласия исполняется вручную по письму оператору.

Импорт файлов проектов в продукт: POST /api/projects/import-file ([0459-импорт-проектов-в-продукт])

Импорт файлов `.gwx` (ZIP с `project.xml`) и `.gw4` перенесён из исследовательского харнеса в продукт; расчётное ядро не менялось. В текстах продукта название программы-источника не употребляется — «файл проекта .gwx/.gw4». Формат читается только для переноса данных пользователя: программа-источник не запускается, защита не обходится.

- **Модули.** `hydroweb/importers/gwx.py` (читатель и разбор в словарь входов; файл из байтов, пути к диску нет), `request_builder.py` (`build_request`, `sanitize_request`, `apply_overrides`; сечение за отчёт не подставляется — E, A, R из файла, сторож `test_steel_constants`), `project_model.py` (запрос → модель проекта интерфейса, обратное к `buildFemRequest`), `messages.py` (тексты для пользователя ⩽ 20 слов), `file_import.py`. Второй копии нет: `scripts/gw_compare/import_gwx.py` — обёртка, `prod_sweep` импортирует сборщик из продукта.
- **Эндпоинт.** `routers/project_import.py`: `multipart/form-data`, часть `file`, `.gwx`/`.gw4`, ⩽ 5 МБ; ответ `{project, warnings, unsupported, kind}`; `project = null` — файл распознан, но содержит непереносимое; ошибка — 422 одной русской строкой; вход только вошедшим, 30 запросов за 10 минут. На проде `client_max_body_size` не меньше 6 МБ (сейчас 10 МБ в `deploy/nginx-host.conf`).
- **Безопасность входа.** Тело проверяет `BodyGuardMiddleware` до разбора (свой предел и только `multipart`, список `UPLOAD_PATH_LIMITS`); архив — записей ⩽ 64, сумма распакованных размеров ⩽ 16 МБ и `project.xml` ⩽ 8 МБ по оглавлению ДО распаковки, чтение ограничено пределом; XML — только UTF-8, `DOCTYPE`/`ENTITY` — отказ; форма разбирается в памяти, на диск ничего не пишется, имена и пути из файла не используются; повреждённый `.gw4` — отказ одной строкой.
- **Что переносится и что нет.** Переносятся слои, этапы с водой слева/справа, подсыпка тыла и лица, срез (датум глубин: в проекте от бровки), водоносные слои, водоупор, пригрузки, нагрузки на стенку, анкеры, распорки и раскосы, бермы, доля минимального активного

давления. Ограждение — профиль списка интерфейса при совпадении J и W с каталогом в пределах 0,1 % и марке стали из списка, иначе сечение числами как «стена в грунте» с предупреждением. Не переносятся, файл не открывается (причина печатается пользователю): пионерный котлован с постоянным углом, грунт подсыпки бермы, ось распорки не в котлован, буровые сваи, трапеция пригрузки в .gw4, срок действия распорки до этапа в .gw4, слой воды нулевой толщины в пределах стенки или с напором. Читаются, строка в списке «Принято с допущением» ([0574-импорт-пять-проектов]): стена из труб (как «Другое»: один элемент, A, Jxc, Wx на элемент шириной Step, деление на Step; строка с d×t и шагом), трапеция пригрузки .gwx (QZ1 у края X1, QZ2 у X2; в запросе intensity — край у стенки, intensity_far — дальний; строка с обоими давлениями), ограждение «Другое...» в .gw4 (поля Main.Ograda* на 1 п.м при шаге Ogradab = 1 м; одна строка вместо общего предупреждения о сечении числами), срок действия пригрузки .gw4 (StageOut — последний этап включительно, remove_stage = StageOut + 1), слой воды нулевой толщины на 100 м и глубже или ниже низа стенки (пропускается, строка), строка пригрузки .gw4 без ширины (при q ≠ 0 — строка, при q = 0 — молча). В «Не перенесено» при открытом файле: поле Fs анкера (смысл не определён) и поле a распорки .gw4 («угол установки»; значение называется, распорка идёт горизонтальной; то же для расхождения a в таблицах Con и Strut и a не числом).

- **Покрытие и инвариант.** Из 51 уникального файла проектов проектировщика в продукте открываются 49 (46 прежних запросов не изменились; не открываются 2194.1 и DemoExample). Запрос импортёра и запрос интерфейса (`buildFemRequest(parseProjectFile(модель))`) на `/api/calculate/fem` дают числа M, Q, w, Кз, Км с отклонением не более 0,535 % (у 22 из 49 файлов — ровно 0); причина объяснена: три различия запроса, которые вносит сам интерфейс (δ слоя без округления до 0,1°, дальний край полосовой пригрузки, нулевая подсыпка тыла); при приведении запроса импортёра к виду интерфейса — 0,0000 % по всем пяти на всех 49 файлах (`scripts/gw_compare/research/import_invariant_1024.py`). Кз здесь калибровочная величина, не нормируемый коэффициент.
- **Клиент.** Меню «Импорт» на странице проектов (`ProjectsPage.tsx`: «Проект HydroWeb (.hwproj.json, ZIP)» и «Файл проекта .gwx / .gw4»), `services/projectFileImport.ts`, диалог `ProjectFileImportDialog.tsx` («Перенесено» / «Принято с допущением» / «Не перенесено»; «Создать проект» — только если `project` не `null`), `projectExport.ts`; образцы — `services/__fixtures__/importedProjects.json`.

Запрос цены поставщикам: /api/price-requests/* ([0478-запрос-цены])

Проектировщик из спецификации запрашивает цену у поставщиков, поставщик отвечает в кабинете ([0471-кабинет-поставщика-v2] — основа). Письма по событиям добавлены записью [0562-письма-кабинета-поставщика] (подраздел ниже): новый запрос уходит поставщикам, предложение и отказ проектировщику; закрытие запроса письма не вызывает. Расчёт и прежние каталоги не менялись.

- **Сервер.** `api/routers/price_requests.py` (подключён в `main.py`), таблицы `price_requests`, `price_request_targets`, `price_offers` (глава 3, таблица БД). Права проверяются по базе через `require_role`: проектировщик — admin, professional, standard; поставщик — supplier; администратор — admin. Маршруты — в таблице выше (одиннадцать:

восемь прежних и три маршрута экрана администратора, [0528-экран-запросов-цены-администратора]).

- **Экран администратора ([0528-экран-запросов-цены-администратора]).** `GET /api/admin/price-requests` принимает `status`, `category`, `supplier` (идентификатор поставщика с дефисами или без), `seen` (`new` | `seen`), `limit` (до 1000, по умолчанию 100) и `offset` (до 100000); неверное значение — 422 одной русской строкой. Ответ: `total`, `counts` (новые и просмотренные по всем запросам, без учёта фильтров), `suppliers` (варианты фильтра) и `requests`; в строке списка предложения не раскрыты (`offers_count`, `valid_until` — наибольший срок «действует до»). `GET /api/admin/price-requests/{id}` отдаёт запрос с адресатами (состояние, причина отказа), предложениями и `history` (отправка, предложения, отказы, закрытие по времени); `GET /api/admin/price-requests/summary` — число непросмотренных; `POST /api/admin/price-requests/{id}/seen` с телом `{seen: true | false}`. Все четыре маршрута только для роли «администратор» (401 без сессии, 403 прочим ролям). Администратор только смотрит: пометка не меняет `status` и `updated_at` запроса, проектировщик и поставщик её не видят. Переписки между сторонами в продукте нет (сообщений платформа не передаёт; письма-сигналы о событиях в базе не хранятся и здесь не показаны), поэтому хранятся и показаны комментарий запроса, ответы с комментариями и причины отказов. Почта проектировщика отдаётся администратору только при `contact_consent = 1` (раньше отдавалась всегда); организация хранится только при согласии; проект-источник — только идентификатор из самого запроса. Просмотр списка и карточки пишется в журнал (`admin_view_price_requests`, без самих данных, в карточке `request_id`), пометка — действием `price_request_seen` (подпись и описание в `auditFormat.ts`).
- **Экран администратора: клиент и тесты.** `components/admin/AdminPriceRequestsTab.tsx` — вкладка «Запросы цены»: таблица, фильтры «Статус», «Поставщик» и «Просмотр», страницы по 100 строк, «Обновить», строка «Новых: N, просмотрено: M», разворот строки с предложениями и ходом запроса, кнопки «Просмотрено» и «Вернуть в новые»; вызовы — `services/priceRequestApi.ts` (`adminPriceRequestApi`); число непросмотренных стоит на вкладке и читается при открытии страницы. Тесты: `hydroweb/tests/test_price_requests_admin.py` (33 собранных: 401 и 403 по четырём маршрутам и трём ролям, строка списка, почта только при согласии, фильтры и страницы, отказы 422, карточка, пометка не меняет сделку, журнал, миграция старой таблицы), `AdminPriceRequestsTab.test.tsx` (7), `AdminUsersPage.tabs.test.tsx`, `auditFormat.test.ts`.
- **Экран администратора: открыто.** Политика (пункт 8.4.3) уже даёт администраторам доступ к запросам цены, предложениям, профилям и позициям поставщиков для модерации каталога, технической поддержки и рассмотрения обращений, поэтому просмотр покрыт; но цель «наблюдение за работой запросов цены» названа не прямо, а служебная пометка «просмотрено» с именем администратора и журнал просмотров в политике не описаны (для обращений есть пункт об отметке «обработано»). Решение за владельцем и юристом. Срок в таблице — наибольшая дата «действует до» среди предложений, срока ответа у самого запроса нет; фильтра по категории в окне нет, хотя маршрут его принимает. Интерфейс проверен тестами компонентов, в браузере не просматривался.

- **Адресация и видимость.** Адресаты записываются при создании запроса: выбранные поставщики или все поставщики, у которых на этот момент есть одобренная позиция категории; поставщик, одобренный позже, прежний запрос не получает. Поставщик видит только адресованные ему запросы, проектировщик только свои, чужой или несуществующий запрос — 404 (существование не раскрывается); администратор читает все, создавать запросы и отвечать за поставщика не может. Организация и почта проектировщика отдаются поставщику только при `contact_consent = 1`: почта берётся из учётной записи на сервере, организация без согласия в базе не сохраняется; в ответе поставщику нет ни `user_id`, ни `project_id`.
- **Статусы и жизненный цикл.** `new` (ответов нет), `answered` (есть хотя бы одно предложение), `declined` (отказались все адресаты), `closed` (закрыт проектировщиком; предложения и отказы после закрытия — 409). Повторное предложение заменяет прежнее, отказ после предложения — 409. Состояние адресата: new, answered, declined.
- **Пределы и ввод.** Цена и количество больше нуля и не выше 1e9; единицы количества шт, м.п., кг, т; единицы цены `rub_per_t`, `rub_per_m`, `rub_per_pc`, `rub_per_kg` (валюта RUB); даты ГГГГ-ММ-ДД, дата цены не позже завтрашнего дня (UTC), срок действия не раньше даты цены; не более 500 запросов на пользователя (409); создание и закрытие — 60 запросов за 10 минут, ответы поставщика — 300 за 10 минут (rate_limit). Отказы — русские строки без `str(e)`; названия полей добавлены в `refusal.py` (контейнер `supplier_ids` — сторож подписей полей).
- **Клиент.** `services/priceRequestApi.ts` (типы, подписи, `offerTotal`: сумма считается только при совпадении единиц количества и цены); `modules/documentation/core/priceRequestSeed.ts` — предзаполнение по обозначению позиции спецификации (Ш шпунт, Р распорки из труб, двутавров и швеллеров, П пояса, А тяги анкеров; строка без категории запроса не предлагает); `PriceRequestDialog.tsx` и `PriceRequestsPanel.tsx` (раздел «Запросы цен» под спецификацией, кнопки «Обновить» и «Закрыть»); колонка «Запрос цены» — в `SpecificationPreview.tsx`; `DocumentsTab` получил необязательное свойство `projectId` (передаёт `App.tsx`); у поставщика — `components/supplier/SupplierRequestsList.tsx`, вкладка «Запросы цены» и счётчик новых в `SupplierCabinetPage`.
- **Удаление учётной записи** `auth.delete_account` удаляет свои запросы, адресации и предложения; статусы затронутых запросов пересчитываются.
- **Тесты.** `hydroweb/tests/test_price_requests.py` (права по ролям и владельцу, адресация, контакты при согласии и без, границы ввода, жизненный цикл, удаление учётных записей); vitest: `priceRequestSeed.test.ts`, `PriceRequestDialog.test.tsx`, `PriceRequestsPanel.test.tsx`, `DocumentsTab.priceRequest.test.tsx`, `SupplierCabinetPage.requests.test.tsx`.
- **Открыто (решения владельца).** Передача организации и почты проектировщика поставщикам меняет состав получателей персональных данных: формулировка согласия, правка политики и согласий и новая версия `LEGAL_DOC_VERSION` (`auth.py`, `legal.ts`, страницы `/docs`) не сделаны; согласие в таблицу `consents` не записывается, признак хранится только в запросе. Нет экрана администратора для запросов (только `GET`), удаления и очистки запросов (при 500 запросах новые не принимаются), счётчика на кнопке «Кабинет поставщика», отдельного поля «цена с НДС» и передачи контактов поставщика

проектировщику. «Всем поставщикам категории» означает поставщиков на момент отправки.

Письма по событиям кабинета поставщика ([0562-письма-кабинета-поставщика], [0563-письма-политика-тексты-книги])

Платформа отправляет письма по пяти событиям кабинета поставщика. Каждое письмо уходит фоновой задачей (`BackgroundTasks`) после ответа маршрута клиенту; ответ маршрута и расчёт не меняются.

Событие и маршрут	Получатель	Состав письма
`POST /api/price-requests`	каждый поставщик-адресат	Тема «Запрос цены: <наименование>». Наименование и марка, категория, количество и единица, раздел кабинета «Запросы цены», ссылка. Контактов проектировщика, его комментария и идентификаторов нет даже при contact_consent = 1.
`POST /api/supplier/price-requests/{id}/offer`	проектировщик	Тема «Предложение цены: <наименование>» («Обновлено предложение: …» при замене). Организация, цена с единицей, дата цены, срок действия или «не указан», название проекта (если проект есть в базе), путь «проект, вкладка «Документы», раздел «Запросы цен»». Комментарий поставщика не входит. Письма нет, если повторное предложение совпало по цене, единице цены и сроку действия.
`POST /api/supplier/price-requests/{id}/decline`	проектировщик	Одна строка: организация поставщика, позиция и причина отказа.
`POST /api/admin/supplier-items/{id}/approve` и `…/reject`	владелец позиции	Тема «Позиция одобрена: <наименование>» или «Позиция отклонена: <наименование>». Решение, категория, записка модератора; у отклонённой позиции путь «Мои позиции».
`POST /api/admin/supplier-items/bulk`	поставщик, одно письмо на поставщика	Тема «Позиций одобрено: N · HydroWeb» или «Позиций отклонено: N · HydroWeb». Список не больше 20 строк (LIST_MAX), остальное строкой «… и ещё N»; позиции с отказом (failed) в письмо не входят; одна решённая позиция поставщика даёт письмо о позиции.
`POST /api/admin/supplier-applications/{id}/approve` и `…/reject`	владелец заявки	Тема «Заявка поставщика одобрена · HydroWeb» или «Заявка поставщика отклонена · HydroWeb». Записка администратора или причина отказа; при отказе подсказка: подать заявку повторно или продолжить как проектировщик.

- **Модули.** `hydroweb/api/email_templates.py` — чистые функции (тема, тело) без базы и почты; `hydroweb/api/supplier_notify.py` — пять задач `notify_new_price_request`, `notify_price_offer`, `notify_price_decline`, `notify_items_decided`, `notify_application_decided`; письма уходят по одному через `email_send.send_email` в потоке (SMTP блокирующий). Маршруты `price_requests.py`, `supplier.py`, `supplier_applications.py` получили параметр `background: BackgroundTasks`.
- **Получатель.** Действующая учётная запись (`users.is_active = 1`) с непустым `users.email`; адрес профиля организации поставщика (`supplier_profiles.email`) не используется. Закрытие запроса, отправка позиции на модерацию, её отзыв и удаление и новая заявка

писем не вызывают. Согласие на рассылку не проверяется: письма по событиям от него не зависят.

- **Гарантии.** Задача не бросает исключений (`_never_raises`): сбой базы, шаблона или почты пишется в журнал `hydroweb.notify` с трассировкой, адреса только в маске (`mask_email`), ответ клиенту не меняется и не содержит текста исключения; сбой одного письма не останавливает остальные. Повторной отправки при сбое нет.
- **Почта и ссылка.** Без `HYDROWEB_SMTP_HOST` функция `send_email` печатает тему и тело в журнал (адрес в маске) и возвращает False, как для писем подтверждения и восстановления доступа; число принятых писем пишется строкой `NOTIFY <событие> letters=N sent=M`. Ссылка — `HYDROWEB_FRONTEND_URL` (по умолчанию `http://localhost:5173`), на главную страницу: у интерфейса нет адресов проекта, вкладки «Документы» и разделов кабинета, письмо называет раздел словами.
- **Оформление.** Простой текст, по-русски; тема не длиннее 60 знаков (`SUBJECT_MAX`, текст из запроса сокращается многоточием); подпись «— Платформа HydroWeb» и строка «Письмо отправлено автоматически по событию в вашей учётной записи HydroWeb.». Строка о том, что письма нельзя отключить, не включена (`POLICY_LINE` = `None`): владелец её не просил. Политика называет эти уведомления среди служебных сообщений (строка раздела 6 «Направление служебных сообщений», дополнение к редакции 2026-10-29 без смены версии).
- **Не сделано.** В письмах нет срока ответа и региона запроса: в `price_requests` таких полей нет (регион есть только у позиции поставщика). Ссылки ведут на главную страницу. Письма в базе не хранятся: виден только журнал. Отдельного переключателя этих писем в профиле нет.
- **Тесты.** `hydroweb/tests/test_supplier_emails.py` (29: письмо на каждое событие и адресата, отсутствие контактов и комментариев в письмах, явный список адресатов, неактивная учётная запись, повторное предложение, отказ одной строкой, одиночное и массовое решение, решение по заявке, без SMTP только журнал в маске, сбой отправки и сбой задачи не меняют ответы, сбой одного письма не останавливает остальные, шаблоны); `hydroweb/tests/test_consents.py` (строка служебных сообщений политики называет события кабинета; страницы, текстовая копия и README каталога редакции равны); `test_books_619_1029.py` и `bookRelease619Claims.test.ts` (тексты книг и справок интерфейса о письмах равны коду).

Продукция поставщиков в проектах: сводка спроса ([0498-продукция-поставщика-в-проектах])

Раздел кабинета «Моя продукция в проектах» и вкладка администрирования «Продукция поставщиков в проектах»: обезличенная сводка того, какие позиции заложены в сохранённых проектах, в каком объёме и по какой цене. Расчёт и прежние ключи ответов не менялись; новые поля проекта в расчёт не идут.

- **Поля проекта.** Четыре необязательных поля `HydrowebProject` (глава 3, строка «projects.data_json»). `services/specSummary.ts` строит `specSummary` по `docPackage.quantities`: у шпунта штуки, погонные метры (число шпунтин на длину шпунтины) и тонны (масса из спецификации, с запасом на материал, если он задан), у труб, двутавров и швеллеров распорок и поясов то же. Позицию поставщика клиент опознаёт по

загруженному каталогу (`supplier:<id>`), встроенную по ключу каталога (`catalog:<ключ>`). В сводку не идут анкер, коробчатое и нетиповое сечение, стенка не из шпунта и железобетонная стена (профиль, оставшийся в проекте от прежнего выбора, спроса не создаёт). Отпечаток `specInputKey` берёт всё, что читает подсчёт объёмов (стенка, распорки с поясами, анкеры, размеры котлована в плане): изменился любой из них, а спецификацию не пересобирали, в проект сводка не пишется, и строки проекта на сервере удаляются. Расценки, запас и цену шпунта из сметы держит редактор, а не вкладка «Документы»: раньше они терялись при уходе со вкладки и при перезагрузке.

- **Сбор на сервере.** `record_project_usage` вызывается после `POST/PUT /api/projects`: каждая строка `specSummary` проверяется схемой `SpecLine` (строгие числа, лишние ключи отбрасываются, не больше 200 строк, количество больше нуля и не больше 1 000 000, цена больше нуля и не больше 1 000 000 ₽ за тонну); неверная строка пропускается целиком. Позиция поставщика сверяется с базой: одобрена, категория совпала, наименование берётся из базы. Регион даёт `regions.normalize_region` из поля проекта «Регион»: засчитывается полное название одного из 89 субъектов (в падежах и с сокращениями «обл.», «респ.») или общепринятое короткое («Москва», «Санкт-Петербург», «Подмосковье», «Татарстан», «ХМАО»); одно прилагательное («Московская») субъекта не даёт, слова после «ул.», «пр.», «р-н» и подобных не засчитываются, два разных субъекта в тексте дают пустой регион, а не первый попавшийся. Строки проекта пересобираются при каждом сохранении (нет `specSummary` — удаляются), для несуществующего проекта не пишутся; сбой пишется в журнал и сохранение проекта не затрагивает. Копия проекта — отдельный проект.
- **Цена за тонну.** Цена сметы РИМ у шпунта (материал «шпунт», единица «т») берётся всегда, когда она в смете есть: введённая проектировщиком вместо цены ФГИС ЦС — источник `rim`, цена ФГИС ЦС — `fgis`; клиент передаёт её с происхождением (`rimPriceSource`: `user` или `fgis`), цена без происхождения не принимается. Расценка с вкладки «Документы» (`manual`: «Шпунт — металл», «Распорки — металл», «Обвязочные пояса — металл», ₽/т) берётся, только когда цены в смете нет; у распорок и поясов цена всегда расценка. Расценки на работы (погружение, монтаж, бурение) в сводку не идут, у анкерных тяг цены нет. Цена у позиции в проекте остаётся, только если она есть у всех строк позиции и источник один; цены нескольких строк усредняются по количеству.
- **Маршруты.** `GET /api/supplier/usage` (роль `supplier`): `{items, projects_total, recent_days: 90}`; у позиции `projects`, `projects_90d`, `volumes` (`{unit, quantity}` по шт, м.п., т), `prices` (`{unit, min, median, max, projects, manual, rim, fgis}`), `last_used` (ГГГГ-ММ), `months` (не больше 36, новые сверху) и `regions` (по числу проектов; ключ `null` — «не указан»). Одобренные позиции шпунта и труб без строк показываются с нулями. `GET /api/admin/supplier-usage` (admin): те же агрегаты по всем позициям с полями `source` и `org`; условий поставщика не требует. Ответы не содержат идентификаторов проектов и пользователей. Проекты считаются по `project_id`: проект один раз, сколько бы раз его ни сохраняли.
- **Принятие условий.** `GET /api/supplier/usage` отвечает 403, если у поставщика нет согласия `supplier_terms` версии `LEGAL_DOC_VERSION` (запись прежней редакции, отозванное и другое согласие не в счёт). Согласие то же, что пишет `submit` (`_record_supplier_terms`, `source = supplier-cabinet`, одна запись на редакцию); без отправки позиции его пишет `POST /api/supplier/accept-terms` (предел `supplier_write`:

300 действий за 10 минут), статус отдаёт `GET /api/supplier/terms`. Клиент `SupplierCabinetPage` перед сводкой читает статус; не принято — `SupplierUsageSection` показывает приглашение со ссылкой на условия, датой редакции из `legal.ts` и кнопкой «Принимаю условия», после неё раздел перезагружается. Вкладка администрирования условий поставщика не требует.

- **Клиент.** `UsageTable` рисует строки позиций и разворот «Подробнее» (по месяцам, по регионам, строка источников цены с тремя числами, в том числе нулевыми); `AdminUsageTab` добавляет столбец «Поставщик» (организация или «каталог»), фильтр по категории (`kind`), вывод по 50 строк (`limit` и `offset`, «Показано N–M из K», «Назад» и «Вперёд»). Строка под таблицей расценок на вкладке «Документы» (`CostEstimatePanel`, `data-testid=supplier-summary-notice`) называет проектировщику, что уходит поставщику; формулировка согласована и приведена дословно, справка «?» поясняет состав.
- **Тесты.** `hydroweb/tests/test_supplier_usage.py` (118 собранных: сбор строк и цен, запись при сохранении, права, удаление вместе с проектом, позицией и учётными записями, агрегаты и границы 90 суток, обезличенность ответов и таблицы, регион, принятие условий, реквизиты проекта в данных проекта: в сводку не попадают и регион не задают); vitest: `specSummary.test.ts`, `supplierUsageView.test.ts`, `SupplierUsageSection.test.tsx`, `SupplierCabinetPage.usage.test.tsx`, `App.usage.test.tsx` (запись полей в проект и устаревание на настоящем хранилище), `DocumentsTab.specSummary.test.tsx`, `documentsProject.test.ts`, `AdminUsageTab.test.tsx` и добавления в `AdminUsersPage.tabs.test.tsx` (шесть вкладок, обход стрелками), `EstimatePanel.test.tsx`. Книги сверяют `hydroweb/tests/test_books_usage_1026.py` и `frontend/src/reports/bookUsage1026Claims.test.ts`.
- **Открыто.** Порога малых групп нет (решение владельца 26.10.2026): при одном проекте в месяце или регионе поставщик видит числа этого проекта. Это обезличивание в смысле «нельзя определить без дополнительной информации», а не анонимизация: оператор по `project_id` находит проект и его владельца. Данные приходят от клиента и не проверяются сверх границ: вошедший пользователь может записать строки на любую одобренную позицию (как он может исказить счётчик «В проектах»); наименьшая и наибольшая цена уязвимы к одной искажённой, но правдоподобной строке, медиана устойчива. Старый клиент при сохранении не присылает поля, и строки проекта удаляются до следующего сохранения новой версией. Со сводкой связываются шпунт и трубы распорок: двутавры и швеллеры поставщика в распорках и поясах выбрать нельзя, анкерная тяга поставщика с 26.10.2026 несёт в проекте номер позиции (anchors.rows[].tendon.id = supplier:<id>, ячейка «Тяга») и учитывается в supplier_demand («В проектах»), но buildSpecSummary тяги не собирает, поэтому в сводке supplier_usage их нет.

Сравнение вариантов проекта: поле variants и ключ utilization_wall_metal ([0475-сравнение-вариантов])

Результат МКЭ запоминается как вариант А или Б вместе с вводом проекта и стоимостью; два варианта стоят рядом в таблице на вкладке «Результаты», выгрузка — DOCX. Расчёт и прежние ключи ответов не менялись.

- **Поле проекта.** `HydrowebProject.variants?: { a?, b? }` (`types.ts`): вариант — `savedAt`, `input` (`VariantInput` = проект без `variants`), `result` (сжатый снимок ответа МКЭ: Mmax, Qmax, wmax, Км, η, Кз, ориентир [Кз] (ориентир, не норматив), длина стенки, число этапов, статус), `cost` (`VariantCost | null`). Отдельной таблицы нет: поле лежит в `data_json` проекта, сервер его не знает; старые проекты без поля открываются как прежде, поле не пишется, пока ничего не запомнено; вариант весит около 6 КБ. В реестре `projectFieldFlow.ts` — строка `variants.*` со статусом «отчёт» (в расчёт не идёт).
- **Новый ключ ответа.** `POST /api/calculate/fem` возвращает `utilization_wall_metal` = 1/Км: наибольшее по этапам и подэтапам использование сечения стенки, округление до 4 знаков; `null`, если металл не проверялся (`FemCalculateResponse` в `api/main.py`; `types.ts`: `utilization_wall_metal?: number | null`). Раньше η давал только метод Блюма (Ks = 1/η); считать 1/Км на клиенте нельзя (числа только из API). Числа расчёта и прежние ключи не менялись (`test_api_fem_utilization_metal.py`); у сервера без ключа в таблице стоит прочерк (`femResults.ts`: `utilizationMetal`).
- **Клиент.** `services/variantComparison.ts` — чистые функции: `snapshotInput`, `snapshotResult`, `makeVariant`, `normalizeVariants` (чтение из сохранённого проекта), `buildComparison` (группы Профиль, Схема, Металл, Запас в заделке Кз, Стоимость; разность Б − А и её доля от А — единственные вычисления на клиенте, заделка d = длина − глубина; лучшее значение по направлению строки (у строк Кз «лучшего» нет: подсветка читалась бы как соответствие норме, а Кз — ориентир, не норматив; разность Б − А у них показывается); при ненадёжном расчёте подсветки нет; при разных способах сметы разность не считается); `services/variantInputDiff.ts` — `diffInputs` («что отличается во вводе»: `SCALAR_FIELDS` и `TABLE_FIELDS` с подписями и единицами как на вкладках; две строки о размерах котлована в плане, planSize.width и planSize.length, нет поля — 20 × 30 м; строки таблиц сопоставляются по коду строки, поля, значимые не всегда, сравниваются, когда значимы; тест не даёт добавить в реестр поле «доходит» без подписи); `services/variantCost.ts` — `summarizeManualCost` и `summarizeRimCost` (стоимость из уже посчитанных сумм; costMaterialRow задаёт подпись строки: «Материалы, ₽» у сметы РИМ — материалы ЛСР целиком, «Металл, ₽» у ручного ввода — позиции металла ведомости, «Металл или материалы, ₽» у разных способов, где разность не считается; id строки costMetal прежний); `components/VariantComparison.tsx` — раздел во вкладке «Результаты» (не девятая вкладка: восемь уже прокручиваются в панели шириной 340–520 px); `reports/generateComparisonDocx.ts` — `buildVariantComparisonDocument` и `generateVariantComparisonDocx` (общий `gostDoc.ts`; модуль грузится по нажатию кнопки, `docx` остаётся отдельным куском сборки; образец добавлен в `exportedDocs.guard.test.ts`). `EstimatePanel` и `DocumentsTab` сообщают редактору показанную стоимость (`onCostChange`), `App.tsx` держит её с ключом ввода: ввод изменён после расчёта сметы — в вариант она не входит.
- **Найдено тестом.** Сообщение вкладки «Документы» о стоимости шло по объекту, а не по значению, и зацикливалось: редактор перерисовывает вкладку с новым объектом проекта на каждом проходе. Теперь сообщение идёт по значению (JSON).
- **Тесты.** `variantComparison.test.ts`, `variantInputDiff.test.ts`, `variantCost.test.ts`, `VariantComparison.test.tsx`, `ResultsTab.variants.test.tsx`, `App.variants.test.tsx` (настоящий редактор: запомнить А, изменить, запомнить Б, сохранение в проекте,

повторное открытие), `variantComparisonDocx.test.ts`, `DocumentsTab.cost.test.tsx`, `femResults.test.ts`; pytest `test_api_fem_utilization_metal.py`.

- **Открыто.** Подсчёт спроса кабинета поставщика (`routers/supplier.py`) находит ссылки `supplier:<код>` на любой глубине данных проекта, поэтому позиция поставщика в запомненном варианте считается выбранной в проекте; оставить так или исключать `variants` из обхода — не решено. Не сделано: «открыть вариант» (вернуть ввод и эпюры в редактор), больше двух вариантов, имена вариантов, выгрузка в XLSX; у Блюма варианты не запоминаются (прогиб и Kз как результат он не выдаёт).

Откуда это число: реестр оснований ([0476-почему-это-число])

У ключевых результатов расчёта стоит кнопка «?», раскрывающая карточку: метод, формула, пункты норм, источник данных, исходные величины и замечания расчёта. Расчёт, числа ядра и значения ключей ответов не менялись; клиент ничего не считает.

- **Реестр.** `frontend/src/data/resultBasis.ts` — типизированный, без `any`: `BASIS_IDS` (13 величин: Mmax, Qmax, wmax, Kmetal, Kz, embedmentDepth, wallLength, supportForce, toeReaction, profileW, profileA, steelRy, steelGrade); у каждой варианты `fem`, `femAxial`, `femRigid`, `blum` с текстом «как получено», формулой, расшифровкой обозначений и пунктами норм (`NormRef`: источник, пункт, чему служит); фраза «пункта нормы нет» (`NO_NORM_DEFAULT`); у Kз пометка «Ориентир, не норматив» (поле `guide`). Пункты — только те, что использует ядро (комментарии и заголовки модулей), сверены с текстом сводов; ссылки — только из `data/openSources.ts` (реестр хранит идентификатор источника, не адрес). У метода Блюма указан «СП 381, разд. 6» по заголовку `core/embedment.py`, без номера пункта.
- **Таблица «код замечания → величины».** `NOTE_BASIS` в том же файле: 22 кода — 12 прежних именованных правил `core/notes.py` (`deflection_exceeded`, `steel_grade_default`, `kz_below_one` и другие), восемь именованных кодов [0479] (`blum_upper_bound`, `embedment_margin_low`, `delta_default`, `gamma_sat_default`, `phi_out_of_range`, `corrosion_allowance`, `mmax_anomalous`, `length_not_above_excavation`), один код-хэш замечания об учёте износа в МКЭ и один именованный код [0525-замечание-бермы-с-полосой-и-чистка-замечаний] (`berm_strip_over_slope`: берма с отпором, полоса пригрузки на поверхности не дальше 4,5 м от стенки и срез откоса на одном этапе; `berm_strip_over_slope_notes` в `core/input_notes.py`; граница 4,5 м взята из наблюдений, а не выведена). Остальные замечания несут код-хэш от текста с названиями слоёв и связей, он нестабилен, и к величинам они не привязаны. Менять существующие `code` нельзя: это значения ключей ответа. Коды сверяет `test_note_codes_basis.py`.
- **Сборка карточки.** `core/resultBasisView.ts`: `buildBasisView(key, ctx)` берёт числа из ответа расчёта и проекта (этап наименьшего запаса — выбор записи из `stages`); `components/BasisCard.tsx` — кнопка и карточка (один компонент на оба места: таблица и панель этапов), `useBasisCard.ts` — состояние (одна открытая карточка на экран; Esc, «×» и повторное нажатие закрывают, фокус возвращается на кнопку); правки `ResultsTab.tsx`, `ResultsVerdict.tsx`, `StagePanel.tsx`. Источник профиля — `SHEET_PILE_SOURCES` в генерируемом `data/sheetPileSources.generated.ts` (`scripts/catalog/build_ui_profiles.py`): документ и строка `source` каталога для каждого профиля списка; у Л4, Л5, Л5-УМ и Л7 строки `source` в каталоге ядра нет, карточка

называет документ без страницы. `/api/profiles` не менялся: интерфейс его не читает, а точное равенство ответа без одобренных позиций закреплено тестами кабинета.

- **Тесты.** `resultBasis.test.ts` (полнота реестра, ссылки только из `openSources.ts`, запретные названия, стиль текстов), `resultBasisView.test.ts` (настоящие ответы сервера), `BasisCard.test.tsx`, `ResultsTab.basis.test.tsx`, `StagePanel.basis.test.tsx`; pytest `test_note_codes_basis.py` и `test_ui_profile_sources.py`.
- **Открыто.** Подстановка чисел в формулу с итогом («Км = … = 6,40») не печатается: ответ не несёт W и A после износа и номера этапа наименьшего запаса; нужен новый ключ ответа с использованными величинами (не решено). Карточек нет у таблицы «По этапам строительства» и на вкладке «Этапы». Ссылка на СП 381 в `openSources.ts` ведёт на страницу, на которой автоматическое чтение пересказало примечание к п. 6.1.7 иначе, чем текст свода (пересказ мог быть неточным): страницу нужно открыть и сверить.

Готовность к экспертизе ([0477-готовность-к-экспертизе])

Список полноты данных и документов проекта стоит первым блоком вкладки «Документы». В четырёх первых группах нормативных требований, новых вычислений и ссылок на документы нет; две группы с нормативным основанием добавлены в [0505-чек-лист-пп87-ецпэ] (отдельный пункт ниже). Сервер не менялся.

- **Пункты.** `core/expertiseReadiness.ts` — чистая функция `buildReadiness` (вход: проект, настройки, результаты, реквизиты, состояние сохранения, отметки о выгрузках; выход: пункты, группы и счётчики; без `any`). Двадцать пять идентификаторов `ReadinessId` в четырёх прежних группах (`input`, `calc`, `docs`, `project`; их перечисляет `BASE_GROUP_IDS`) и ещё пятнадцать в двух группах с основанием (`pp87` — 11, `ecpe` — 4; всего сорок), состав зависит от проекта; статусы `ready`, `notReady`, `manual`. Каждый пункт берёт готовую проверку: `core/schemeCheck.ts` (окно «Проверка схемы»), `describeCalcStatus`, `core/limitStates.ts` и `core/resultsVerdict.ts` (предельные состояния), `validateForDocumentation`; состояние сохранения `ExpertiseReadiness.tsx` читает из хранилища проектов (`getSyncStatus`, `getProjectConflict`). Критичные замечания — `CRITICAL_NOTE_CODES` (`kz_below_one`, `strength_not_met`, `bisection_not_converged`) и тексты, которые `splitWarnings` считает критичными.
- **Группы с основанием ([0505-чек-лист-пп87-ецпэ], 26.10.2026).** `ReadinessGroupId` дополнен значениями `pp87` («По Положению № 87», 11 пунктов; «Мероприятия по защите конструкций» — только у стального ограждения) и `ecpe` («Для подачи через ЕЦПЭ», 4 пункта, все всегда manual). У пункта этих групп есть поле `source` (`{ doc: 'pp87' | 'minstroy783', point }`): `sourceLabel` строит «ПП № 87, п. 14 «в»» (подписи документов — `SOURCE_DOC_LABEL`), `sourceUrl` берёт адрес официального текста из реестра `data/openSources.ts` (записи `pp87`, `minstroy783` и `minstroyXml`, только с verified: true; нет записи — ссылки нет). Пересказ пункта документа стоит первым предложением в help пункта и начинается с обозначения основания. У групп есть собственный `ReadinessGroup.help` с общим примечанием `NORM_GROUPS_NOTE`: «Это подсказка, не замена требований экспертизы: состав раздела «Конструктивные решения» определяют задание на проектирование и Положение № 87». Новых вычислений нет: `pp87Items(inp, base)` берёт состояние проекта, реквизиты и уже собранные пункты прежних групп (`pp87Groundwater` читает `groundwater`, `pp87Strength` — `metal`, `embedment` и `supports`); результатные пункты

проходят `normResultGate`: нет расчёта или он устарел — notReady. Где программа сведений не ведёт (топография, метеорология и климат, химический состав вод, подпись, имена файлов при подаче), пункт manual. Все пятнадцать входят в общий итог `summarize`.

- **Источники групп (прочитаны 26.10.2026).** Положение о составе разделов проектной документации и требованиях к их содержанию, утверждённое постановлением Правительства РФ от 16.02.2008 № 87, редакция от 21.10.2025 (срок действия документа ограничен 1 сентября 2028 года; п. 14 — раздел 4 «Конструктивные решения» в редакции постановления от 27.05.2022 № 963, п. 10 — раздел 1, п. 23 — раздел 7, п. 3 — общий состав). Требования к формату электронных документов, утверждённые приказом Минстроя России от 12.05.2017 № 783/пр (официальное опубликование, изменяющих и отменяющих актов на портале опубликования не найдено). Страница «XML-схемы» сайта Минстроя: среди действующих есть схема «Раздел №1 проектной документации «Пояснительная записка»», схемы для раздела «Конструктивные решения» нет. Документы Главгосэкспертизы об XML-схемах и ЕЦПЭ не читались (сайт не открылся), пунктов по ним в списке нет.
- **Отрисовка и переходы.** `modules/documentation/components/ExpertiseReadiness.tsx`: первый блок `DocumentsTab`, три колонки при ширине блока от 640 px (её измеряет `ResizeObserver`), ниже — две. По умолчанию блок свёрнут: видны название, строка «Готово N из M · не готово K · вручную L» (части с нулём не пишутся) и цветная полоса. Разворот запоминается в localStorage под ключом hydroweb_readiness_open (`readinessPref.ts`: одна отметка интерфейса, данных проекта в ней нет; чтение и запись в try/catch, без хранилища выбор действует до закрытия вкладки); флажок «Только невыполненное» у развёрнутого списка. Переходы идут через `setActiveTab` в `App.tsx`, окно «Проверка схемы» и панель реквизитов. Строка «Основание: …» у пунктов с источником — `SourceLine` (data-testid readiness-source): обозначение — ссылка с target=_blank и rel=noopener noreferrer, если адрес есть в реестре; в строке группы с help стоит `HelpTip`, примечание `NORM_GROUPS_NOTE` повторено и в «?» всего блока.
- **Правила.** Результат устарел, если после расчёта изменён ввод (`shownResultsStale` в `App.tsx` считается по показанному результату, а не по методу в шапке): пункты по результатам и выгрузки по ним не засчитываются. Выгрузки (`ExportLog`: package, report, xml, drawing) помечаются отпечатком данных проекта на момент выгрузки (`packageStamp` для пакета включает реквизиты комплекта), отметка действует, пока отпечаток совпадает; журнал живёт в `App.tsx` и сбрасывается при импорте проекта, в проект и на сервер ничего не пишется. Кз и перемещение метод Блюма как результат не выдаёт — пункты идут «вручную». Отметка «Просмотрено» (`notesReviewed` в `App.tsx`): `notesStamp` — отпечаток списка замечаний (код, строка и пояснение каждого, порядок не важен); у пункта «Замечания расчёта просмотрены» в статусе manual есть `reviewStamp`, кнопка вызывает `onNotesReviewed`, совпадение отпечатка даёт ready с подписью «Замечаний: N, просмотрены»; у критичных замечаний (`CRITICAL_NOTE_CODES` и тексты, которые `splitWarnings` считает критичными) кнопки нет; отметка сбрасывается при импорте проекта и в проект не пишется. Перемещение больше рекомендуемого (при заданном допустимом смещении [w] — больше заданного вами [w], слова «заданного вами» в строке пункта, название пункта прежнее) — notReady (как «не проходит» в `buildFemVerdict`; тест

сверяет статус пункта со светофором на десяти значениях перемещения), у предела — manual, в запасе — ready.

- **Тесты.** `expertiseReadiness.test.ts`, `ExpertiseReadiness.test.tsx`, `DocumentsTab.readiness.test.tsx`, `DocumentsTab.layout.test.tsx` (раскладка вкладки не менялась), `App.readiness.test.tsx` (настоящий редактор: расчёт, правка ввода, четыре выгрузки); фикстуры — `testing/readinessFixtures.ts`. Длину строк проверяет тест списка: название до 10 слов, найденное до 16, «что сделать» до 14. С [0505] в `expertiseReadiness.test.ts` (84 теста) проверены группы `pp87` и `ecpe`: состав и порядок идентификаторов, основание у каждого пункта, статусы пунктов по результатам и по слоям грунта, пункт защиты конструкций только у стального ограждения; в `ExpertiseReadiness.test.tsx` (28) — строка «Основание» и «?» групп; `bookLinesClaims.test.ts` сверяет книги с кодом: сорок идентификаторов, из них 25 прежних, 11 и 4 в новых группах, шесть групп названы в обеих книгах.
- **Открыто.** Хранить ли отметки выгрузок в проекте (сейчас они только в сеансе); нужны ли разные списки по стадиям; включать ли в список состояние `ChecksSection` (устойчивость, анкеры). Решения владельца по [0505]: считать ли группы с основанием в общем итоге (сейчас считаются) или показывать отдельной строкой; нужны ли пункты по проекту организации строительства шире п. 23 «ф» (не добавлялись); векторный PDF чертежа (сделан в [0526-pdf-чертежа-с-текстовым-слоем]: `modules/drawing/export/pdfExporter.ts`, текстовый слой есть, описание — в разделе о чертеже ниже; пункт списка готовности переписан) и имена файлов «Раздел ПД № N» при выгрузке (остаются открытыми); уточнение текста прежнего пункта «XML для ЕЦПЭ сформирован» (он говорит «схема ЕЦПЭ официально не утверждена», а утверждённые схемы Минстроя есть для раздела 1; текст прежнего пункта не менялся). Размеры котлована в плане с [0490] лежат в проекте (раздел ниже). На стенде `dev-app.html` расчёт, запущенный через `startup.autoCalculate`, до экрана не доходил (на прод-сборку не влияет).

Размеры котлована в плане: поле проекта planSize ([0490-доработки-интерфейса])

Размеры вводились в трёх местах отдельно и при смене вкладки возвращались к 20 × 30 м. Теперь это одно необязательное поле проекта; расчётное ядро, числа ответов и ключи полей API не менялись, сервер не менялся.

- **Поле.** `HydrowebProject.planSize?: { width, length }` (м, `types.ts`). `services/planSizeProject.ts`: `normalizePlanSize` (не объект или не положительные числа читаются как «размеры не вводились»), `withPlanSize` (пока размеры не вводились, ключа в проекте нет совсем), `planSizeOf` (введённые либо `DEFAULT_PLAN_SIZE` = 20 × 30). Старые проекты и файлы без поля читаются как 20 × 30 м. В запросы к расчёту поле не входит: реестр `projectFieldFlow.ts` относит поле к виду «отчёт», сторож проверяет, что тело запроса МКЭ не меняется.
- **Один источник.** `App.tsx` держит поле и передаёт вниз `hooks/usePlanSize.ts` (чтение из проекта, запись через `onPlanSizeChange`); три ввода читают и пишут его же: вкладка «Документы» (спецификация, длина распорок, выгрузка), вкладка «Чертёж» и карточка «Вода: приток и понижение» (`WaterInflowPanel`). Вкладки, собранные отдельно (без редактора, в тестах), держат значение у себя.

- **Следствия.** Размеры входят в отпечаток данных проекта (`projectStamp`): их смена снимает отметки о выгрузках, как любая правка данных; в ключ ввода стоимости варианта: от них зависит длина распорок; `waterSnapshotCurrent` сверяет размеры запроса притока воды с общими, и результат для других размеров в раздел записки не идёт; `variantInputDiff.ts` печатает две строки («Ширина котлована в плане, м», «Длина котлована в плане, м»; нет поля: 20 × 30); пункт списка «Размеры котлована в плане заданы» считается по наличию поля (введённые 20 × 30 дают «готово», размеры в подписи без лишних нулей: «25 × 40 м», «12,5 × 44 м»); чертёж и раньше считал размеры введёнными по факту ввода («условные» снимается любым вводом).
- **Тесты.** `planSizeProject.test.ts`, `App.planSize.test.tsx` (настоящий редактор: размеры введены в одном месте и видны в двух других), `DrawingTab.test.tsx`, `WaterInflowPanel.test.tsx`, `variantInputDiff.test.ts`, `projectFieldFlow.test.ts`.

Реквизиты проекта и карточка тома ТЭО в проекте: поля projectInfo и teoCard ([0504-тэо-по-ревизии])

Реквизиты вкладки «Документы» и ввод диалога «Том ТЭО» жили в состоянии вкладки и после перезагрузки страницы и повторного открытия проекта вводились заново. Теперь это два необязательных поля проекта; расчётное ядро, числа ответов, ключи полей API и сервер не менялись (поля идут вместе с проектом общим путём data_json).

- **Поля.** `HydrowebProject.projectInfo?: Partial<ProjectInfo>` (шифр code, марка markChoice и customMark, название, организация, заказчик clientName, местоположение location, стадия, исполнители developer, checker, approver, normControl, основание designBasis, отчёт об изысканиях geologyReportRef, дата) и `HydrowebProject.teoCard?: TeoCard` (types.ts). `services/projectInfo.ts`: `normalizeProjectInfo` (текстовые поля с предельной длиной, stage — П, Р или ППР, markChoice — из списка; неверное отбрасывается по полям, а не роняет открытие), `resolveProjectInfo` (умолчания `DEFAULT_PROJECT_INFO` с названием проекта, поверх них сохранённое), `storedProjectInfo` (в проект идёт только отличное от умолчаний; пустая строка равна отсутствию), `withProjectInfo` (правок нет — ключа в проекте нет). `teo/teoCard.ts`: `normalizeTeoCard` (пределы длины текстов 20 000 и 1 000 знаков, не больше 200 строк, пустое не хранится, участок по умолчанию 10 п.м не пишется) и `withTeoCard`.
- **Один источник.** `App.tsx` держит `docInfo` (читается из saved.projectInfo сразу при открытии, до вкладки «Документы»; при импорте — из файла) и `teoCard` (`handleTeoCard`); ключ `infoKey` — содержимое реквизитов, поэтому открытие вкладки без правок проект не пишет; название в реквизитах, равное прежнему названию проекта, следует за переименованием (`prevProjectName`). `DocumentsTab` получает `teoCard` и `onTeoCardChange`; `TeoDialog` читает карточку один раз при открытии и сообщает о правках через `onCardChange` (первое состояние правкой не считается). Автосохранение и экспорт .hwproj.json несут оба поля, импорт читает их.
- **Кто читает реквизиты.** Основная надпись чертежа, параметры выгрузки отчётов, список готовности (пункты реквизитов, основания, pp87Norms), пояснительная записка и том ТЭО (через вкладку «Документы»). Регион для сводки поставщиков из реквизитов не берётся: сервер берёт его из карточки проекта (meta.region, routers/supplier_usage.py), а projectInfo.location — свободный текст местоположения для документов (тест

test_supplier_usage.py: реквизиты в данных проекта в сводку не попадают и регион не задают).

- **Известное ограничение.** В панели «Реквизиты проекта» нет полей «Заказчик» (`clientName`) и «Местоположение» (`location`), а карточка тома их больше не вводит: диалог показывает их из реквизитов («не задано, в томе «[заполнить]»») со ссылкой «Изменить в реквизитах проекта». Пока значения не заданы иначе (например, в файле проекта), эти два сведения в томе остаются «[заполнить]». Решение за владельцем.
- **Реестр и тесты.** В `projectFieldFlow.ts` записи `projectInfo.*` и `teoCard.*` — вид «отчёт» (в расчёт не идут), `anchors.rows[].tendon.*` — «ui»; сторож проверяет, что тело запроса МКЭ не меняется (projectFieldFlow.test.ts, 4). `projectInfo.test.ts` (12), `teoCard.test.ts` (6), `App.projectInfo.test.tsx` (11, настоящий редактор: ввод пишется в проект и переживает повторное открытие, мусор в поле открытие не ломает, переименование проекта, основная надпись, список готовности, записка и том ТЭО, импорт из файла и старый файл без реквизитов).

Сквозной сборщик запроса и сторож «поле проекта → запрос» ([0293-C4])

Аудит прохождения параметров 01.10.2026 нашёл: из 160 полей проекта на МКЭ-путь доходили 105, 24 терялись на фронте, 6 были только в отчёте, ещё одно (`pioneer.mode`) принималось и игнорировалось; подсыпка по этапам, водоупор, «Корректировка давлений», W/Ry ж/б стены, настройки расчёта (на МКЭ-путь не доходил ни один из 107), углы связей > 85° (422), «Автоподбор профиля» (запрос без слоёв — 422, кнопка молча мертва), таблица Ka/Kp/K0 (считалась в браузере: Ренкин без δ, K0 = 1 − sin φ). Решение:

- `services/projectToRequest.ts` — единственное место, где решается, какие вводы доходят до ядра: `buildFemRequest` (тело POST /api/calculate/fem), `buildBlumRequest` (тело /api/calculate и /api/auto-profile), `blumIgnoredInputs` (что из проекта метод Блюма не учитывает; уходит как ignored_inputs ⩽ 40 строк по 200 символов, сервер возвращает одной строкой в warnings), `buildCoefficientsRequest`. Подсыпка по этапам, водоупор (AQUICLUDE_NONE = 100 — «нет», не шлётся), «P0 → Pp», шаг сетки и доля минимального активного из настроек, W/Ry ж/б стены, глубина пионерного среза в режиме «Полный откос» (равна глубине котлована, общая со схемой), водоносные слои (buildAquifersPayload, «Zабс.1» — глубина кровли от верха стенки, как водоупор и бермы), free_standing_mode, остаточная прочность. Углы связей — SUPPORT_ANGLE_MAX = 85. Метод Блюма при supports > 1 предупреждает «учтена только первая опора» (числа те же, что при одной опоре).
- `services/projectFieldFlow.ts` + `projectFieldFlow.test.ts` — реестр: каждое листовое поле HydrowebProject классифицировано sent (доходит; note — поле API) / notSent (сознательно не доходит; в интерфейсе помечено «Пока не учитывается в расчёте») / report (только отчёты) / ui (служебное). Сторож: полный проект задан с типами Required<…> — новое поле в типах ломает typecheck теста, пока его не добавят; затем каждое поле изменяется по одному, и тело запроса сравнивается: sent обязано измениться, остальные — нет. Поле интерфейса без записи в реестре — красный тест. Инвариант: при дефолтных новых полях тело МКЭ побайтно прежнее (projectToRequest.test.ts сравнивает JSON с дословной копией старой сборки, legacyFemRequest.fixture.ts; тест на настоящем App).

- Прочее из C4: настройки проекта сохраняются (автосохранение, экспорт/импорт); диалог «Настройки» оставил только работающие органы (шаг сетки, доля минимального активного, 10 флагов вкладки «Отчет», которые читает DOCX); схема рисует УГВ, взятый в расчёт; отчёты, чертёж и записка печатают уровни воды по таблице этапов (а не «Глубину УГВ по умолчанию»), подсыпку и водоупор, если заданы; тесты — test_api_param_flow.py (в том числе полный запрос интерфейса для /api/auto-profile).

Клиент: статус, ход, отмена, проверки ([0293-C3, C5b, C6, C7])

- `services/femJobs.ts` — fetchFemCalculation(run, init) возвращает обычный Response: POST …/jobs → опрос GET раз в 700 мс (первый через 300 мс; до 4 подряд сетевых сбоев) → 200 с JSON результата либо код и {detail}; при 404/405/501 на старте — откат на синхронный эндпоинт (отмена тогда обрывает соединение через AbortController, сервер замечает разрыв). createFemRun().cancel() прерывает опрос и шлёт DELETE.
- `hooks/useCalcProgress.ts` — только факты («этап N из M · прошло X с · оценка ~Y с (предварительная)», «в очереди на расчёт»); у синхронного пути — предварительная оценка по модели, зеркалу cost_estimate (upper_s). Кривая «95 %» и баннер «дольше обычного» убраны: факт обгонял оценку в медиане в 3,8 раза, баннер срабатывал на 52 расчётах из 56. `core/calcStatus.ts` — тексты шести статусов, перевод WARNING_* (humanizeSolverWarning), `components/CalcStatusBlock.tsx` (data-testid="calc-status"), `App.tsx` — только doCalculateFem и баннер хода с кнопкой «Отмена»; 429 и 422 показываются без префикса «Ошибка МКЭ-расчёта».
- `core/resultsVerdict.ts` — вердикт «проходит» требует reliable !== false и учитывает крепления: buildSupportLimitStates (строки «N ⩽ N_пред» из support_checks, core/supportChecks.ts), проверки сечения распорок и поясов (СП 16; N/F из огибающей МКЭ, M распорки от собственного веса $q \cdot L^2/8$); сошёлся, но unreliable → не выше «у предела»; не сошёлся — заголовок статуса вместо совета про длину стенки.
- `components/ChecksSection.tsx` — общий контейнер дополнительных проверок на «Результатах»: запись SECTIONS {id, title, description, isAvailable, emptyText, render}; регистрация — одной записью (hydraulic, stability, anchors). `services/checksInput.ts` — buildChecksInput() собирает вход из того же тела buildFemRequest (датум, вода, водоупор, подсыпка, «полный откос» совпадают с расчётом); панели HydraulicChecksPanel, StabilityPanel (SVG-схема дуг), AnchorCapacityPanel (усилия МКЭ добавляет по id); параметры корня анкера (rootMethod, rootSoil, rootDiameterMm, boreholeDiameterMm, injectionPressureKPa, rootRakKN, tendonCapacityKN, anchorDuty) — AnchorRootInputs, в МКЭ не идут (реестр: report). Тесты: HydraulicChecksPanel.test.tsx, StabilityPanel.test.tsx, AnchorCapacityPanel.test.tsx (9), checksInput.test.ts, checksInput.anchors.test.ts (7), App.anchorRoot.test.tsx.
- `core/applicability.ts` — реестр «Границы применения расчёта и невыполненные проверки»: запись {id, title, norm, note, defaultStatus, applies, evaluate}; статус done | partial | not_done (выполнена / выполнена частично / не выполнялась) определяется по фактам расчёта (collectApplicabilityFacts: путь МКЭ или Блюма, наличие результатов, крепления, сейсмика, коррозия, режим остаточной прочности). Печатается в ПЗ (раздел 10), расчётных отчётах DOCX МКЭ и Блюма (таблица перед выводами) и XML (элемент ГраницыПрименения). Строка «metal» у метода Блюма — partial: в отчёте показана одна

проверка (σ = M/W), результаты касательных и приведённых напряжений и местной устойчивости расчёт в отчёт не передаёт ([0400-отчёты-блюм-xml]). registerCheck(def) заменяет запись по id на месте — заготовки general_stability, filtration_stability и load_factors остаются «не выполнялась», потому что ни одна панель их не регистрирует: результаты проверок в отчёты не попадают. Анкеры — не заготовка: вместо одной строки anchor_ground реестр содержит восемь строк по состояниям п. 7.3.9 СП 381 (anchor_head, anchor_ground — код прежний, anchor_pullout, anchor_tendon, anchor_stability, anchor_prestress, anchor_structures, anchor_groups; применяются, когда есть включённые анкеры; статус и пояснение, которое печатается при любом статусе, даёт core/anchorLimitStates.ts, [0549-применимость-анкеры-статусы]).

- **Строка «subgrade_k» — коэффициент постели** (05.10.2026): «Коэффициент постели основания в контактной модели (модель Винклера)», СП 381.1325800.2018 п. 6.4.14; применяется только на пути МКЭ (метод Блюма контактной модели не использует). Свод допускает описывать основание моделью Винклера с коэффициентом постели, растущим линейно с глубиной и различным для участков стены выше и ниже уровня экскавации; в расчёте он постоянный в пределах слоя грунта (задан в таблице грунтов либо выведен из модуля деформации). Статус всегда «выполнена частично»; факт subgradeProportional (задан скалярный коэффициент пропорциональности K, wall.subgradeK > 0; поля kProp и subgradeZOrigin в факт не входят — они не отправляются в запрос) только уточняет примечание: ниже уровня экскавации коэффициент постели растёт линейно от дна котлована каждого этапа, C = K·(z − H_exc) по СП 24.13330, прил. Б, выше уровня экскавации у тыла — остаётся постоянным в пределах слоя; начало отсчёта глубины и значения K свод не называет — их выбирает проектировщик. Формулировка в отчёте — без оценок: что допускает свод и что принято в расчёте. Тест — core/applicability.test.ts.

Сессия и безопасность

Сессия — JWT в cookie `hw_session` (HttpOnly, SameSite=Lax, Secure везде, кроме локальной разработки по http), модуль `api/session_cookie.py`. Токен запроса берётся из действительной cookie, иначе из заголовка Authorization: Bearer (совместимость со старыми вкладками и внешними клиентами). Порядок важен: заголовок в браузере несёт только остаток старой сессии из localStorage, и при «заголовок сильнее» он перебивал свежий вход ([0179]). Новые токены в localStorage не пишутся; фронт стирает старый при входе.

Ограничение частоты: `api/rate_limit.py` — регистрация 5/10 мин, вход 10/5 мин, смена пароля 10/10 мин (эти три настраиваются переменными окружения, см. раздел 2), повторное письмо 3/10 мин, по IP из X-Forwarded-For; счёт — на один процесс uvicorn. Проверка и хеширование пароля (bcrypt) — вне цикла событий (hash_password_async / verify_password_async в api/auth.py). Ответы API не содержат str(e): текст исключения — только в серверный лог. Это первый слой защиты входа; второй — счётчик по учётной записи (ниже).

Отзыв сессий. Токен несёт два поля, по которым `get_user_by_token` проверяет его при каждом запросе: `pwd_fp` (отпечаток текущего хеша пароля — смена пароля гасит все прежние токены) и `ep` (эпоха сессий пользователя). `POST /api/admin/users/{id}/logout` увеличивает `users.session_epoch` — все выданные токены перестают проходить без смены пароля; токен без поля ep считается эпохой 0. Вход через Яндекс/VK тоже выдаёт токен текущей эпохи

(`OAuthUser.session_epoch`). Смена пароля пользователем (`POST /api/auth/change-password`) выдаёт ему новую куку и accessToken в теле, а остальные устройства выбрасывает.

Проверка Origin (второй слой против CSRF). `api/origin_guard.py` (`OriginGuardMiddleware`, чистый ASGI): изменяющий запрос (POST, PUT, PATCH, DELETE) под /api/ с кукой `hw_session` отклоняется кодом 403 с текстом «Запрос с этого сайта не допускается.», если Sec-Fetch-Site равен cross-site, Origin равен null или Origin не входит в белый список CORS (`HYDROWEB_CORS_ORIGINS`) и не совпадает с Host запроса (разработка по 127.0.0.1 и стенд под другим именем работают без правки списка). Не блокируются: запросы без куки (Bearer, curl), запросы без Origin (старые клиенты, same-origin GET), GET, HEAD и OPTIONS. Первый слой остаётся прежним: SameSite=Lax у куки и требование application/json для тел под /api/ (`refusal.BodyGuardMiddleware`); слой Origin закрывает два маршрута загрузки с multipart/form-data, куда кросс-сайтовая форма дошла бы без предзапроса CORS. В журнал пишется строка CSRF_BLOCK с путём и причиной. `parse_cors_origins` в `api/main.py` отвергает на старте «*», запись без схемы и с путём. Тесты: `test_origin_guard.py`, `test_cors_origins.py`.

Срок и продление сессии. `TOKEN_EXPIRE_HOURS` (api/auth.py): 336 ч (14 суток), было 720. Срок скользящий: `SessionRenewalMiddleware` (`api/session_renewal.py`, чистый ASGI, внутри OriginGuardMiddleware, поэтому отклонённый им запрос сессию не продлевает) для запроса с кукой `hw_session` проверяет возраст токена — `session_needs_renewal(payload)`: поля iat нет, набор полей токена не менялся, возраст = срок минус остаток до exp. Если он не меньше `SESSION_RENEW_AFTER_S` (3600 с) или токен выпущен с прежним, более длинным сроком, то в момент ответа токен проверяется по базе так же, как защищённым маршрутом (`get_user_by_token`), и в ответ добавляется Set-Cookie со свежим токеном. Не продлеваются: Bearer без куки, ответы, сами ставящие или удаляющие куку (вход, выход, смена пароля, OAuth), отозванные сессии, токены purpose=verify и reset.

Выход. `POST /api/auth/logout`: если сессия действует, вызывает `revoke_user_sessions(user_id)` (её же зовёт admin_force_logout): users.session_epoch +1 в БД, поэтому отзыв виден всем процессам uvicorn; затем кука удаляется (без действующей сессии — только удаляется). Скопированные до выхода кука и токен перестают проходить `get_user_by_token`. Побочное следствие по решению владельца (24.10.2026): выход завершает все сессии пользователя, в том числе на других устройствах.

Блокировка входа по учётной записи. `api/login_guard.py` и таблица `login_attempts` (acct_key — SHA-256 нормализованной почты, failures, window_start, locked_until; LOGIN_ATTEMPTS_SCHEMA в db/connection.py; в копию deploy/backup_db.sh не входит: запись живёт не дольше 15 минут). Одна атомарная инструкция INSERT … ON CONFLICT DO UPDATE, счёт общий для процессов uvicorn. Проверка идёт в `login_user` ДО поиска пользователя и проверки пароля, поэтому заблокированная запись получает отказ и с верным паролем. Неудача — неверный пароль или неизвестный адрес (текст отказа один, по нему не узнать, есть ли такая запись); «деактивирован» и «адрес не подтверждён» не считаются. `HYDROWEB_LOGIN_LOCK_FAILURES` (20) неудач за `HYDROWEB_LOGIN_LOCK_WINDOW_S` (900 с) дают блокировку на `HYDROWEB_LOGIN_LOCK_S` (900 с): ответ 429 с Retry-After и текстом «Слишком много неудачных попыток входа. Повторите через N мин или восстановите пароль.» (N

округляется вверх до минут). Счёт снимают успешный вход до порога, сброс пароля по ссылке из письма и установка пароля администратором. Лимит по адресу (`rate_limit.py`) остаётся первым слоем и по-прежнему считается в памяти одного процесса; общие для процессов только счётчики по учётной записи. Цена решения: посторонний может запереть чужой адрес на срок блокировки, отправив 20 неверных паролей; сброс пароля по почте при этом работает.

Content-Security-Policy. Заголовок в `deploy/nginx-site.conf` блокирующий (до 24.10.2026 стоял Report-Only): `default-src 'self'`; `script-src 'self'` плюс хеш sha256 одного встроенного скрипта оглавления в книгах /docs/user.html, dev.html и supplier.html; `style-src 'self' 'unsafe-inline'` (SVG чертежей со style=, <style> в книгах); `img-src 'self' data: blob:`; `font-src` и `connect-src 'self'`; `frame-ancestors` и `object-src 'none'`; `base-uri` и `form-action 'self'`. Изменился скрипт оглавления — изменится хеш, и test_nginx_site_conf.py падает, пока хеш в конфиге не обновлён (глава 10, «Сборка и публикация руководств»).

Приглашённые пользователи. Ответы `/api/auth/login` и `/api/auth/me` несут флаги `mustChangePassword` и `consentsPending` (в `AuthResponse` по умолчанию false). Клиент по ним показывает экран первого входа `FirstLoginScreen` (ветка first-login в AppWrapper перед выбором роли): документы, затем смена временного пароля через `/api/auth/change-password`. Это клиентский барьер: на сервере доступ приглашённого до смены пароля и принятия документов не ограничен. Кабинет и приглашение не должны менять пароль мимо `set_user_password` или `clear_must_change_password` — иначе флаг останется и пользователь зациклится на экране смены.

ИИ-ассистент: API, база знаний и источники

`POST /api/assistant/chat` (`hydroweb/api/assistant.py`) отдаёт SSE-поток строк `data: {...}` с объектами `{text}`, `{sources}`, `{done}` и `{error}`. Доступ — роли standard, professional и admin (зависимость _require_assistant_user; 401 без действительной сессии, 403 для неизвестной роли); токен берётся из cookie сессии, иначе из заголовка Authorization (token_from_request). Ограничение частоты — 30 запросов за 10 минут с одного IP (rate_limit, бакет assistant); проверяется первым и суточный лимит не расходует. Лимиты запроса (ChatRequest): messages — не более 40 сообщений, первое от user, каждое до 8000 символов; project_context — до 24 000 символов. Контекст проекта и выдержки базы знаний подаются в системный промпт как ДАННЫЕ, а не инструкции (в промпте на этот счёт отдельное правило). Провайдер LLM задаёт HYDROWEB_AI_PROVIDER (по умолчанию yandexgpt).

Поиск по базе знаний. `_retrieve_sources(query)` берёт последний вопрос пользователя, получает эмбеддинг (Yandex, 256 измерений) и ищет по `kb.db` (`ai/knowledge_base.py`: SQLite + numpy, косинус как скалярное произведение нормированных векторов). В промпт попадают до 6 выдержек с косинусом не ниже 0,33 и суммарным объёмом до 9000 символов; список источников уходит клиенту первым событием `sources`. Любая ошибка поиска не роняет ответ: ассистент работает без источников.

Элемент `sources[]` содержит `label` (подпись: документ, страница, раздел) и `score` (релевантность), а для документов с известным открытым адресом — ещё `title` (полное название) и `url`. Ссылки берёт `ai/source_links.py`: словарь `SOURCE_LINKS` сопоставляет имя файла в базе (`chunks.doc`) записи {title, url}, а `link_for(doc)` возвращает её или None. Сейчас в нём СП 381.1325800.2018 (открытый текст), книга Цытовича и Тер-Мартиросяна «Основы

прикладной геомеханики в строительстве» (карточка Национальной электронной библиотеки: полного текста в открытом доступе нет) и два руководства платформы (`/docs/user.html`, `/docs/dev.html`). Все адреса проверены вручную 01.10.2026; документ без записи показывается без ссылки — адрес не выдумывается. Адрес СП 381 совпадает с `frontend/src/data/openSources.ts`: это проверяет тест `test_sp381_link_matches_frontend_open_sources`.

ИИ-ассистент: суточный лимит и /api/assistant/quota

Лимит считает и проверяет сервер (`hydroweb/api/assistant.py`); интерфейс только показывает остаток. Таблица счётчика — `assistant_usage` (см. «Схема БД: что добавил релиз» в главе 3).

- **Кому и сколько.** `_ALLOWED_ROLES` = standard, professional, admin. Суточный лимит только у standard: `HYDROWEB_ASSISTANT_DAILY_LIMIT_STANDARD`, по умолчанию 3; нечисловое значение — предупреждение в журнале и умолчание 3; ноль или отрицательное — ассистент для роли закрыт (в БД не пишется). У professional и admin суточного лимита нет, но счётчик ведётся: used_today в /quota и статистика использования во вкладке «Состояние» админки точные.
- **Сутки.** Календарные, по Europe/Moscow (если в образе нет tzdata — фиксированное UTC+3: в Москве смещение постоянное). Часы подменяются через `assistant._now()` — на этом построены тесты границ суток.
- **Атомарное списание.** Одна инструкция `INSERT … ON CONFLICT(user_id, day) DO UPDATE SET count = count + 1 WHERE count < ?`: rowcount = 1 — вопрос списан, 0 — лимит исчерпан. Гонки между воркерами нет (SQLite сериализует запись). Чистка `DELETE … WHERE day < (сегодня − 40 суток)` выполняется в той же транзакции.
- **Порядок проверок.** Авторизация и роль → частота (429 «слишком частые запросы») → провайдер (503, если не настроен; вопрос не списывается) → суточный лимит (429) → поток. Если провайдер вернул ошибку до первого куска текста, вопрос возвращается в счётчик (`_refund_question_sync`).
- **429 суточного лимита** отдаётся JSON-ответом: `detail` (строка — старый клиент покажет её как есть), `code: "assistant_daily_limit"`, `limit_per_day`, `used_today`, `remaining: 0`, `reset_at` (ISO со смещением +03:00) и заголовок Retry-After. Так клиент отличает его от 429 по частоте запросов.
- **GET /api/assistant/quota** — `{role, limit_per_day | null, used_today, remaining | null, reset_at}`; у ролей без лимита limit_per_day и remaining равны null. Ответ с Cache-Control: no-store. При включённых проверках по уровню ([0570-уровни-доступа-шаг-2a]) добавляется `level_quota` (уровень, период, лимит, использовано, остаток, дата обновления), а у уровня без ассистента ответ — 403 level_required.
- **Лимит уровня ([0570-уровни-доступа-шаг-2a]).** Только при включённом выключателе HYDROWEB_LEVELS_ENFORCED; выключено — прежний путь без изменений. Лимит берётся из матрицы уровня: «Бесплатный» — 5 вопросов в календарный месяц по Москве, «Старт» — 30, прочие уровни — 300. Тот же счётчик assistant_usage: месячная сумма — SUM(count) по суткам месяца (месяц до включения выключателя считается ретроактивно). Списание — одна инструкция: вставить или увеличить, пока за период меньше лимита уровня и за

сегодня меньше суточного лимита роли; между воркерами гонки нет. Если исчерпан суточный лимит роли standard (3 в сутки), отвечает прежний 429 assistant_daily_limit, иначе исчерпанный лимит уровня даёт 403 level_required с датой обновления: «Лимит уровня «Бесплатный» исчерпан: 5 вопросов в месяц, обновится ДД.ММ.ГГГГ. Нужен уровень «Старт».»

- **Клиент.** `assistantAccess.ts` — roleCanUseAssistant(role), единственное место условия доступа (раньше оно дублировалось в App.tsx и ProjectsPage.tsx). `services/assistantQuota.ts`: fetchAssistantQuota (любой сбой даёт null — панель работает без строки остатка), parseQuota (проверка формы чужого ответа), quotaLine («Осталось N из 3 вопросов сегодня»), limitReachedText, PRO_MODE_HINT, msUntilReset. `assistantService.ts` получил обработчик onQuotaExceeded (429 с кодом assistant_daily_limit). `AssistantPanel.tsx`: остаток читается при каждом открытии и после ответа, при исчерпании блокируются поле ввода, отправка и подсказки, вопрос возвращается в поле ввода, таймер перечитывает остаток к полуночи по Москве (не чаще раза в 30 с). `AssistantFab` принимает raised (на телефоне выше нижней панели). `solution-finder/SolutionFinderAssistant.tsx` + `solutionFinderAssistantContext.ts` подключают ассистента к экранам «Подбор решений» (область диалога solution-finder, контекст — условия, грунты, три решения, предупреждения и допущения как ДАННЫЕ).

Тесты: `hydroweb/tests/test_assistant_quota.py` (лимит, сутки по Москве, атомарность при параллельных запросах, чистка 40 суток, возврат вопроса, 401/403/429), `test_api_assistant_auth.py`; на клиенте — `AssistantPanel.test.tsx`, `assistantQuota.test.ts`, `assistantAccess.test.ts`, `SolutionFinderAssistant.test.tsx`. Политика обработки ПДн (/docs/privacy.html, генератор scripts/legal/build_legal_pages.py) описывает счётчик: идентификатор, дата, число вопросов, без текста вопросов, хранение 40 суток; LEGAL_DOC_VERSION при этом не менялась.

База знаний: очистка от материалов третьих лиц

Выдержки базы подставляются в промпт и показываются пользователю в списке источников, поэтому в ней не должно быть материалов и названий сторонних коммерческих расчётных программ и компаний-поставщиков: ссылаться следует на нормативы, открытую литературу и документацию платформы. Правила едины для сборки базы и для очистки готового файла — `hydroweb/ai/sanitize.py`:

- `EXCLUDED_NAME_MARKERS` — документы, которые не индексируются независимо от содержимого (проверка по имени файла, `is_excluded_name`): справка и руководство сторонней расчётной программы (на них стоит запрет воспроизведения), вопросы-ответы поставщика шпунта (текст от первого лица: склад, цены, обратный выкуп — платформа такого утверждать не может) и внутренние отчёты передачи этапов разработки (stage_3_summary, stage_4_handoff).
- `MAX_BANNED_SHARE = 0,30` — если запрещённые названия встречаются в большей доле чанков документа, он считается сравнением или рекламой и исключается целиком (`exclusion_reason`).
- В остальных документах `scrub_text` вырезает предложения, пункты списков, заголовки и строки таблиц, в которых встречаются запрещённые названия (регулярное выражение

BANNED_RE); scrub_chunk отбрасывает чанк, от которого осталось меньше 60 символов или в котором запрещённое всё же уцелело, а scrub_label чистит подписи источников.

Защита стоит на трёх уровнях, чтобы не зависеть от того, кто и с какими аргументами запустит сборку:

- **Сборка.** ingest.collect_files пропускает файлы по имени, ingest.clean_chunks применяет к чанкам правила sanitize до получения эмбеддингов, _label_for чистит подпись. В базу подают только нужные документы, а не каталог целиком:

```
python scripts/knowledge_base/build_kb.py --src база_знаний/источники \
    документы/нормативы/SP-381.1325800.2018.pdf --out hydroweb/data/kb.db
```

- **Готовая база.** scripts/knowledge_base/kb_scrub.py — разовая очистка kb.db без пересчёта эмбеддингов (ключ Yandex не нужен; у исправленных чанков остаются прежние векторы — текст отличается на отдельные предложения, на поиск это не влияет). Ключ --dry-run только печатает отчёт; перед записью делается копия в .backups/kb-ГГГГММДД-ЧЧММ.db, в таблицу meta пишется scrubbed_at, в конце проверяется, что упоминаний не осталось (иначе код возврата 3). Запуск: python scripts/knowledge_base/kb_scrub.py --db hydroweb/data/kb.db.
- **Рантайм.** assistant._retrieve_sources применяет scrub_chunk и scrub_label к каждой выдержке до промпта и до клиента — страховка на случай базы, собранной из неочищенных материалов. Системный промпт дополнительно запрещает называть сторонние программы и компании и сравнивать с ними результаты. Тесты — hydroweb/tests/test_kb_sanitize.py, включая проверку, что поставляемая kb.db чиста.
- **Размер чанка.** hydroweb/ai/chunking.py: целевой размер 2800 символов, перекрытие 400, жёсткий потолок HARD_MAX_CHARS = 3700. До 08.10.2026 потолок был 6000: чанк из 5908 символов (с идентификаторами кода) дал 2275 токенов и 400 Bad Request, потому что лимит модели эмбеддингов Yandex — 2048 токенов на текст, а число токенов на символ зависит от текста. Потолок подобран по замеру всех 607 чанков штатной сборки (три руководства, СП 381 и учебник): медиана 0,28 токена на символ, руководства — до 0,41, СП 381 и учебник — до 0,55 (разреженный текст PDF, но короткие чанки до 1,9 тыс. символов); по худшему замеру 3700 × 0,548 ≈ 2028 ⩽ 2048. На обычные чанки (до целевого размера плюс перекрытие, около 3200 символов) потолок не влияет — режутся только одиночные абзацы длиннее 3300 символов. Символьный потолок гарантирует токенный лимит лишь для замеренных видов текста: для нового источника сборку нужно проверять. Разделитель абзацев (двойной перенос строки, 2 символа) теперь входит в бюджет перекрытия (прежде чанк мог выйти на 2 символа длиннее потолка); _MAX_TEXT_CHARS = 8000 в embeddings.py — страховка от запредельно длинного текста, на практике не срабатывает. Тесты — test_ai_rag.py (18).
- **Системный промпт и запас в заделке.** В расшифровке сокращений промпта (api/assistant.py) запас в заделке назван «Кз» — калибровочной величиной программы, не нормируемым коэффициентом; допустимое значение — ориентир, не норматив; в данных API поле называется kst или K_заделки. Пример записи чисел в промпте — «Кз=1,92». В источниках базы (база_знаний/источники) то же сказано у K_заделки в developer_guide.md и fem_theory.md; тест test_kb_sanitize.py проверяет и промпт, и эти источники.

Документы, которые попадают в базу

> Руководства база_знаний/источники/user_guide.md, developer_guide.md и supplier_guide.md индексируются как источники ассистента, поэтому в них (как и в текстах книг-методичек, собираемых из scripts/books/book_*.js) не должно быть названий третьих лиц. Проверяйте это при каждой правке документации поиском по запрещённым названиям из BANNED_RE.

ИИ-ассистент: клиентская часть

- `components/AssistantPanel.tsx` — выдвижная панель. Свойства: getContext (сериализованный контекст), userId, scope, scopeLabel, suggestions. Родитель создаёт панель с key=scope, чтобы при смене области состояние пересоздавалось. Ответ рисуется через RichText (`linkifyNorms`); блок «Источники из базы знаний» (SourceList) сворачивает повторяющиеся подписи и делает строку ссылкой «открыть ↗», только если url начинается с http(s). Стриминг — `services/assistantService.ts` (streamAssistant; 401 сначала проверяется запросом /api/auth/me, чтобы не разлогинить действующую сессию).
- `components/AssistantFab.tsx` — плавающая кнопка «Ассистент», общая для редактора проекта и страницы проектов. `UserMenu` ассистента не открывает: единственная точка входа — эта кнопка (на странице проектов, в редакторе и на экранах «Подбор решений»). Условие показа — roleCanUseAssistant из assistantAccess.ts: любая из ролей standard, professional, admin; суточный лимит standard проверяет сервер.
- `assistantContext.ts` — buildProjectsContext(entries, selectedIds): сводка списка проектов для страницы проектов (до 40 проектов — MAX_PROJECTS_IN_CONTEXT; название, статус, стадия, режим, адрес, заказчик, дата изменения; выбранные галочками помечены). В начале явно сказано, что расчётных данных у ассистента нет. Подписи статусов и стадий вынесены в `projectLabels.ts`.
- `services/assistantHistory.ts` — диалог в localStorage. Ключ `hw:assistant:v1:<userId>:<scope>`, где scope — `general` (страница проектов, GENERAL_SCOPE) или `project:<id>` (projectScope). Значение — {messages, draft, updatedAt}; диалог старше 30 суток (DIALOG_TTL_MS) удаляется при загрузке. Для хранилища — не более 60 последних сообщений, 20 000 символов на сообщение и 150 000 на диалог (trimForStorage); прочитанное нормализуют cleanMessages и cleanSources (в ссылки пропускаются только http/https). Каждое обращение к хранилищу обёрнуто в try/catch: приватный режим или переполнение не ломают ассистента. Сохранение — после ответа, для черновика с задержкой 300 мс, а также при pagehide и размонтировании панели (идущий стрим прерывается, частично полученный ответ сохраняется). На сервер диалог не отправляется и там не хранится.
- `buildApiHistory(messages)` — история для запроса по лимитам сервера: не более 40 последних сообщений (API_MAX_MESSAGES), каждое не длиннее 8000 символов (API_MAX_CHARS, хвост обрезается «…»), первое — от пользователя. Без этого длинный ответ ассистента или 20-й вопрос подряд получали 422 и диалог «ломался». Если ответа не пришло ни куска, вопрос возвращается в поле ввода для повторной отправки по Enter.
- `normLinks.ts` + `data/openSources.ts` — ссылки на нормы в тексте ответа. OPEN_SOURCES — список {id, designation, title, url, cntdUrl, verified, note, fullText}: СП 381, 22, 16, 248, 24, 28, 47, 20, 45, 14, ГОСТ 25100, 20522, 27751 и книга Цытовича и Тер-Мартиросяна. Каждая ссылка проверена вручную, в индекс попадают только записи с verified: true; не добавляйте

адрес, не открыв страницу. Основной url — meganorm.ru (полный текст без входа); cntdUrl — тот же документ на docs.cntd.ru (часто более свежая редакция, но полный текст без входа открыт только вечером по Москве; в интерфейс пока не выводится). linkifyNorms(text) находит обозначения «СП|ГОСТ + номер [+ год]», строит ключ без года (normKey: «СП 22.13330.2016» → «СП 22.13330», «ГОСТ 25100-2020» → «ГОСТ 25100») и превращает их в ссылки, если в индексе есть запись; сам текст обозначения не меняется, оговорка о редакции (note) попадает в подсказку.

Тесты: `assistantContext.test.ts`, `services/assistantHistory.test.ts`, `components/AssistantPanel.test.tsx`.

Экспорт и импорт проектов (projectExport.ts)

Формат файла — тот же `.hwproj.json`, что и в редакторе (поля HydrowebProject: projectName, geology, wall, loads, anchors, struts, groundStruts, berms, excavationStages, aquifers), плюс необязательные служебные поля `hw*`: hwFormat (EXPORT_FORMAT = hydroweb-project), hwVersion (EXPORT_VERSION = 1), hwMeta (карточка ProjectMeta без id, createdBy, createdAt, updatedAt и enrichmentScore) и hwSolutionFinderResult, если проект создан мастером. Редактор лишние поля игнорирует, а список проектов по ним восстанавливает карточку. Результаты расчёта в файл не входят.

- **Выгрузка.** `exportProjects(entries)`: один проект — JSON (Blob application/json), несколько — ZIP по файлу на проект (`buildArchive`, библиотека jszip подключается динамическим `import('jszip')`, как и остальные тяжёлые библиотеки, DEFLATE). Имена файлов — `uniqueProjectFileName` (safeFileName плюс суффикс -2, -3 при совпадении), архив — `hydroweb-projects-ГГГГ-ММ-ДД.zip` (`archiveFileName`).
- **Загрузка.** `importProjectFiles(files, userId)` возвращает ImportReport {imported, failed[{file, reason}], lastId}. Принимает .json и .zip (из ZIP берутся только файлы .json, служебная папка __MACOSX пропускается; в отчёте имя вида «архив › файл»). Лимиты: MAX_IMPORT_FILE_BYTES = 10 МБ на проект, MAX_IMPORT_FILES = 100 за раз. `parseProjectFile` бросает Error с русским текстом (не JSON; нет geology.layers или wall) и дополняет недостающие поля значениями по умолчанию (withDefaults(DEFAULT_*, значение), рекурсивно; массивы берутся из файла как есть): файл мог выгрузить старая версия или собрать вручную, а без полей вроде seismics вкладка геологии падала белым экраном. Чужой файл не вправе подставить идентификатор, владельца и даты (META_OMIT): проект создаётся через createProject как новый, владелец — загрузивший; существующие проекты не затираются. Ошибка одного файла остальные не останавливает.
- **Интерфейс.** `ProjectsPage.tsx`: панель выбранных проектов (Экспортировать / Удалить / Снять выделение), кнопка «Импорт» (скрытый input[type=file] multiple, accept .json и .zip), пункт «Экспортировать» контекстного меню строки; плашка с итогом — успех гаснет через 8 секунд, ошибка остаётся, причины по файлам показываются списком (первые пять и «…и ещё N»). Тесты: `projectExport.test.ts`, `ProjectsPage.transfer.test.tsx`.

Администрирование пользователей (admin API)

Все маршруты закрыты `require_role(["admin"])`. Отказ бизнес-логики — `AdminActionError(status, message)` с заранее написанным русским текстом (он отдаётся

клиенту; str(e) чужих исключений — нет). В журнал `hydroweb.auth` пишутся строки `ADMIN_ACTION <действие> actor=… user_id=…` — без паролей, хешей, ссылок сброса и email.

Маршрут	Что делает
`GET /api/admin/users`	Список пользователей. Помимо id, email, first_name, access_role, profession, is_active, created_at, last_login отдаёт provider (email | yandex | vk), projects_count (число проектов на сервере), email_verified (bool), а с [0517-пробный-доступ-по-метке] и [0520-обращения-с-вложением-и-редакция-2026-10-27] ещё signup_source, signup_source_at, trial_until и feedback_count (поля calc_count нет), а с [0548-уровни-доступа-шаг-1] — access_level, access_level_until, effective_level, effective_level_label, level_source, level_expired.
`POST …/{id}/password`	Тело {new_password}. Не короче 8 символов (422 с русским текстом), не длиннее предела bcrypt в 72 байта (400). Меняет хеш, и все выданные пользователю сессии перестают действовать (сверка pwd_fp в get_user_by_token). Работает и для учётных записей Яндекс/VK — после этого возможен вход по email и паролю. Пароль и хеш не логируются и не возвращаются.
`POST …/{id}/reset-link`	Выпускает тот же токен, что и «Забыли пароль» (create_reset_token, действует 1 час, привязан к хешу пароля), и отправляет письмо (send_password_reset_email). Ответ {ok, emailed, reset_url}: reset_url заполнен только при emailed = false (почта не настроена или отправка не удалась) — администратор передаёт ссылку вручную. 400 — для учётных записей не с provider = email и для деактивированных (ссылка заведомо не сработала бы).
`POST …/{id}/verify-email`	email_verified = 1; идемпотентно (повторный вызов безвреден). Доказательств подтверждения система не требует — ответственность на администраторе.
`POST …/{id}/delete`	Тело {confirm_email}. Одной транзакцией (BEGIN IMMEDIATE) удаляет projects, project_tombstones, oauth_identities, consents и users; ответ {ok, deleted_projects}. 400 — самоудаление, email не совпал (без учёта регистра и пробелов по краям), удаление последнего активного администратора; 404 — пользователя нет.

Клиент — `components/AdminUsersPage.tsx`: поиск, фильтры по роли и статусу, сортировка, постраничный вывод по 50 (PAGE_SIZE), сводка summarize (по всему списку), меню строки («Действия») с диалогами пароля, ссылки сброса и удаления. Локальное состояние (роль, активность, строка) меняется только после успешного ответа сервера, ошибки показываются плашкой или внутри диалога. «Скачать CSV» выгружает весь отфильтрованный и отсортированный список: разделитель «;», UTF-8 с BOM, значения, начинающиеся с = + - @, табуляции или возврата каретки, получают апостроф (защита от CSV-инъекции — имя задаёт сам пользователь). Генератор пароля — на crypto.getRandomValues без смещения (отбраковка хвоста), 14 символов, минимум по одному из каждого класса, без похожих символов. Тесты: `hydroweb/tests/test_admin_user_management.py`, `AdminUsersPage.test.tsx`.

Администрирование: журнал, принудительный выход, приглашение, состояние

Новые маршруты живут в `hydroweb/api/routers/admin_extra.py` (подключён из main.py; права — `require_admin`, тонкая обёртка над require_role). Прежние админские маршруты остаются в main.py и auth.py, но пишут в общий журнал.

Маршрут	Что делает
`GET /api/admin/audit`	Параметры limit (1…500, по умолчанию 50), offset, actor, action, target. Ответ {items, total, limit, offset}, новые сверху. actor и target — точный id ЛИБО подстрока имени или адреса (регистронезависимо и для кириллицы: встроенный LOWER у SQLite работает только с ASCII, поэтому подключена собственная SQL-функция; символы %, _ и обратная косая черта в фильтре экранируются). В выдаче есть подписи actor_email / actor_name / target_email / target_name из текущей таблицы users (в журнале адресов нет; у удалённого пользователя подписи null).
`POST /api/admin/users/{id}/logout`	«Завершить все сессии»: увеличивает users.session_epoch (revoke_user_sessions, как выход) и пишет force_logout. Администратор может применить действие и к себе (текущая сессия тоже завершается). 401/403/404 — как у соседних маршрутов.
`POST /api/admin/users`	Тело {email, first_name, role}. Создаёт учётную запись (provider email, email_verified = 1 — за адрес ручается администратор; must_change_password = 1, consents_pending = 1) и возвращает временный пароль (14 символов, secrets, без похожих символов, по символу каждого класса) ОДИН раз; Cache-Control: no-store. 409 на занятый адрес (регистронезависимо, включая гонку по UNIQUE), 422 на неверный адрес, имя или роль. Письмо (send_invitation_email) уходит только при настроенном SMTP и содержит ссылку установки пароля — токен сброса на 72 часа, пароля в письме нет; без почты или при сбое отправки ссылка возвращается в ответе. При ненастроенном SMTP тело письма в журнал не печатается (там была бы рабочая ссылка).
`POST /api/auth/invite/accept-consents`	Тело {consents:[{kind, version}]}. Любой вошедший пользователь; обязательны terms и pd, marketing необязателен. Пишет в consents с source = 'invite' и снимает consents_pending; повтор идемпотентен. При создании приглашения согласия НЕ записываются.
`GET /api/admin/health`	Состояние сервера, каждая секция независима: версия приложения и штамп выкатки (HYDROWEB_DEPLOYED_COMMIT / DEPLOYED_COMMIT из окружения, затем /app/DEPLOYED_COMMIT; на проде без правки compose штамп может быть неизвестен); БД (имя файла без пути, размер, WAL, число пользователей и проектов, свободное место на томе); kb.db (наличие, число чанков, meta.scrubbed_at, размер; нечитаемый файл — error: read_failed); ЕГРЗ — SELECT MAX(id), помечено approx; флаги SMTP, Яндекс, VK, ассистента (+имя провайдера) без значений секретов; использование ассистента за 7 и 30 суток по ролям из assistant_usage (нет таблицы или иная схема — {available: false}); топ-10 пользователей по объёму проектов (число и байты, без содержимого).

Журнал: `record_admin_action(action, *, actor_id, target_id=None, details=None)` в `api/admin_audit.py` — единая точка записи: строка в таблице admin_audit И строка в файловом журнале (логгер hydroweb.auth, прежний формат `ADMIN_ACTION <действие> actor=… user_id=… <поле>=<значение>`). Действия: set_role, set_active, set_password, reset_link, verify_email, delete_user, force_logout, invite_user, trial_extend, trial_clear, set_access_level, feedback_handled, price_request_seen, admin_view_price_requests, egrz_reimport, egrz_reimport_blocked (попытка пути вне data/ пишется без самого пути). _clean_details отсеивает секреты по имени поля (pass | token | hash | secret | url | link | email | mail) и по виду значения (@, http://, reset_token=), вложенные структуры не пишутся, строки не длиннее 200 символов. Сбой записи не ломает уже выполненное действие — он логируется с трассировкой. Хранение — 400 суток, чистка при каждой записи.

Клиент: `components/admin/` — AdminTabs (WAI-ARIA tabs: tablist/tab/tabpanel, стрелки, Home/End), AdminAuditTab (фильтры, страницы по 50, CSV до 10 000 строк постранично по 500 с пометкой об усечении; защита от CSV-инъекций — тот же csvCell, что у списка пользователей), AdminHealthTab, AdminApplicationsTab (вкладка «Заявки поставщиков»: фильтр по статусу и поиск, раскрытая заявка, скачивание карточки, «Одобрить» и «Отклонить» с причиной; число новых заявок в заголовке, на вкладке и на кнопке «Администрирование» в списке проектов), AdminFeedbackTab (вкладка «Обращения»), AdminCatalogTab (модерация каталога), AdminUsageTab (вкладка «Продукция поставщиков в проектах»: таблица по всем позициям, фильтр по категории, вывод по 50), AdminEventsTab (вкладка «События», [0543-журнал-событий-беты]: период «С» и «По», блок критериев, таблицы «По пользователям» и «По дням», без графиков), AccessLevelDialog (окно «Уровень доступа» в списке пользователей, [0548-уровни-доступа-шаг-1]), InviteUserDialog, ForceLogoutDialog, adminApi.ts и AdminModal.tsx (общие помощники и диалог, вынесенные из AdminUsersPage без изменения поведения), auditFormat.ts (подписи действий и описание подробностей). Вкладок девять: «Пользователи», «Заявки поставщиков», «Обращения», «Модерация каталога», «Продукция поставщиков в проектах», «Запросы цены», «Журнал», «Состояние», «События» (ADMIN_TABS в adminApi.ts). Вкладка «Пользователи» остаётся в DOM под hidden, чтобы поиск и страница не терялись при переключении. `FirstLoginScreen.tsx` — экран первого входа приглашённого. Тесты: test_admin_audit.py, test_admin_session_epoch.py, test_admin_invite.py, test_admin_health.py; AdminUsersPage.tabs.test.tsx, admin/*.test.*, FirstLoginScreen.test.tsx, auth.invite.test.ts.

Кабинет пользователя (routers/profile.py)

Роутер `hydroweb/api/routers/profile.py` подключён в main.py сразу после роутера ассистента. Доступен любому вошедшему пользователю; у каждого маршрута свой бакет rate_limit.

Маршрут	Что делает
`GET /api/profile`	id, email, имя, роль, provider, loginMethods (email + привязанные Яндекс/VK), requiresOldPassword, emailVerified, дата регистрации, число проектов, accessLevel (уровень доступа с матрицей, [0548-уровни-доступа-шаг-1]); без хеша и токенов.
`PUT /api/profile`	Смена имени (strip, схлопывание пробелов, 1–100 символов, без управляющих). Email и роль не меняются. PUT, а не PATCH: в CORS проекта PATCH не разрешён. 30 запросов / 10 мин.
`POST /api/auth/change-password`	Тело {old_password, new_password}. Минимум 8 символов (422), максимум по password_too_long (72 байта, 400), совпадение с текущим — 400. У provider = email проверяется текущий пароль (400 «Неверный текущий пароль.»; не 401, чтобы клиент не принял это за истёкшую сессию), у Яндекс/VK — нет (так пользователь «задаёт пароль для входа по email»). Запись хеша с проверкой старого (UPDATE … WHERE password_hash = ? → 409 при гонке). Меняется pwd_fp: все токены гаснут, но в ответ ставится новая сессионная кука и отдаётся accessToken (для клиентов с заголовком). 10 запросов / 10 мин.
`GET/POST /api/profile/consents`	GET: состояние по видам и вся история (тип, версия, дата, источник). POST {kind, granted}: для marketing отзыв — строка granted = 0 с source = 'profile', повторное согласие — granted = 1; идемпотентно (changed = false, строка не добавляется). terms и pd отзыву не подлежат (400 — только удалением учётной записи).

Маршрут	Что делает
`GET /api/profile/export`	JSON-вложение (version 3, EXPORT_VERSION): профиль (в том числе signupSource, signupSourceAt, trialUntil и accessLevel — действующий уровень), accessLevelChanges (смены уровня: changedAt, from, to, fromUntil, toUntil, reason; без идентификатора администратора), events ([{kind, createdAt}] — вид и время событий журнала; meta_json не выгружается), согласия, обращения «Написать нам» (feedback[]; файлов вложений нет), проекты (карточка, данные, результат Solution Finder), привязки провайдеров (включая providerUserId), надгробия; без хеша пароля, токенов и ключей. NaN/Infinity из старых данных → null. 10 запросов / 10 мин.
`POST /api/profile/delete`	Тело {confirm_email, password?}. Делегирует auth.delete_account(user_id, confirm_email, *, actor_id=None, self_service=False): адрес (без регистра, с обрезкой) → пароль (только при provider = email, проверяется ДО транзакции, чтобы bcrypt не держал блокировку записи) → одна транзакция BEGIN IMMEDIATE (projects, project_tombstones, oauth_identities, consents, assistant_usage, feedback, users; файлы вложений обращений стираются после фиксации базы) → последнего активного администратора удалить нельзя (400) → кука гасится. admin_delete_user — тонкая обёртка с прежними сообщениями и строкой журнала. 5 запросов / 10 мин.

Клиент: `components/ProfilePage.tsx` (оверлей поверх экрана, редактор не размонтируется; страница-подтверждение AccountDeletedPage), `profileApi.ts` (типизированный клиент, ProfileApiError с русским текстом; 401 → «Сессия истекла»), `profileLocalData.ts` с `clearAllDialogs` (`services/assistantHistory.ts`, ключи hw:assistant:v1:<userId>:*) и `purgeLocalProjectsOf` (`projectStore.ts`) — при удалении учётной записи стираются профиль входа, диалоги ассистента и локальные копии проектов. В AppWrapper — состояния profileOpen и accountDeleted, пункт «Профиль» в UserMenu (onOpenProfile). Тесты: test_profile_account/_password/_consents/_export/_delete.py; ProfilePage.test.tsx, UserMenu.profile.test.tsx, AppWrapper.profile.test.tsx, profileLocalData.test.ts.

Пробный доступ по метке ссылки и обращения «Написать нам» ([0517-пробный-доступ-по-метке], [0520-обращения-с-вложением-и-редакция-2026-10-27], [0531-обращения-до-пяти-вложений])

Ссылка `https://hydroweb.online/?from=pto` заводит пробную запись: 15 суток от подтверждения почты, ассистент и «Подбор решений» закрыты, после срока закрыты расчёт и выгрузки. Администратор продлевает срок на 15 суток, снимает режим и читает обращения. Ядро и числа расчёта не менялись. Счётчик расчётов по пользователям не ведётся (решение владельца 27.10.2026): сервер не записывает успешные расчёты, поле calc_count и столбец «Расчётов» убраны, условие продления подтверждает обращение с приложенным отчётом.

- **Метка в браузере и на сервере.** `frontend/src/trial.ts`: `captureArrivalSource` читает `?from=` (`normalizeSource`: [a-z0-9_-]{1,32} без учёта регистра), запоминает первую метку и время перехода в localStorage (`hydroweb_from`; без секретов; при недоступном хранилище живёт в памяти вкладки). Регистрация шлёт поля `source` и `source_at` (`registrationSourceFields`), вход через Яндекс ID передаёт `?from=&from_at=` (`oauthSourceQuery`), сервер кладёт их в подписанную куку `hw_oauth` (10 минут) и записывает метку только новому пользователю (`oauth_common.py`, `oauth_yandex.py`, `oauth_vk.py`). Метка не по форме отбрасывается молча, регистрация идёт как обычная; время перехода из будущего (больше 10 минут), старше года или не разбираемое

отбрасывается, метка остаётся. После регистрации и после возврата от провайдера метка в браузере стирается (`clearArrivalSource`).

- **Срок.** `hydroweb/api/trial.py`: `TRIAL_DAYS = 15`. При регистрации с меткой `trial_until` ставится предварительно (регистрация плюс 15 суток), подтверждение почты (ссылка из письма или «Подтвердить email» администратора, `mark_email_verified`) пересчитывает его от момента подтверждения один раз: повторный переход по ссылке срок не двигает, снятый администратором режим подтверждение не возвращает. Вход через провайдера: срок от первого входа. Администратор и поставщик пробного периода не имеют (`trial_state`).
- **Сервер: TrialGuardMiddleware.** Чистый ASGI-мидлвар (`hydroweb/api/trial.py`). Запрос вошедшего пользователя с пробным периодом к `/api/assistant/*` (любой метод) получает 403 «В пробном периоде ассистент и подбор решений недоступны. Напишите нам, чтобы открыть доступ.» весь срок и после него. После конца срока POST к `/api/calculate`, `/api/calculate-staged`, `/api/calculate/fem` (и `/jobs`), `/api/auto-profile`, `/api/estimate/calc`, `/api/checks/*` и `/api/joints/*` получает 403 «Пробный период завершён ДД.ММ.ГГГГ. Расчёт и выгрузки недоступны, проекты можно просматривать. Напишите нам, чтобы продлить доступ.» (дата по Москве). Запросы без сессии мидлвар не затрагивает: расчётные маршруты, как и прежде, доступны без входа, пока выключатель HYDROWEB_LEVELS_ENFORCED выключен (при включённом расчёт требует вход — подраздел «Уровни доступа, шаг 2а»). Вход, `/api/projects` и `/api/auth/me` остаются; `/api/auth/login` и `/api/auth/me` несут `trialUntil` (только у записи в пробном периоде).
- **«Подбор решений».** Классический расчёт и проверку анкеров мастер делит с профессиональным режимом: `/api/calculate` и `/api/checks/anchors` общие. Проверку МКЭ с [0570-уровни-доступа-шаг-2а] он считает на своём маршруте `/api/calculate/fem/finder` (подраздел «Уровни доступа, шаг 2а»). Закрыть общие маршруты значило бы закрыть расчёт профессионального режима, поэтому мастер закрыт в интерфейсе (кнопка на экране выбора режима видна и открывает сообщение; сохранённый проект подбора у пробной записи открывается в редакторе, а не в мастере), а ассистент закрыт и в интерфейсе, и на сервере. Закрыть мастер на сервере отдельным решением теперь возможно (шаг 2б плана уровней); пока этого нет, после срока пробного периода маршрут подбора отвечает 403 «Пробный период завершён…» вместе с остальными `/api/calculate/*`.
- **Выгрузки.** Отчёты (DOCX, XLSX, XML) и чертежи строит браузер, сервер их не видит. `setTrialGuard` и `exportBlockReason` в `trial.ts` держат дату конца срока на уровне модуля (`AppWrapper.tsx` ставит её при каждой смене пользователя). После срока `handleGenerateReport`, сравнение вариантов (DOCX), чертёж узла и чертёж (SVG, DXF, PDF) показывают причину и не выгружают, вкладка «Документы» заменена `TrialExpiredPanel`; выгрузка проекта в файл `.hwproj.json` (копия собственных данных) не закрыта.
- **Администратор: пробный период.** `POST /api/admin/users/{id}/trial/extend` (от большего из конца срока и текущего момента, плюс 15 суток; 409, если пробного периода нет), `POST /api/admin/users/{id}/trial/clear` (`trial_until` пуст, метка остаётся); в журнал пишутся `trial_extend` и `trial_clear` (`AUDIT_ACTIONS`, подписи в `auditFormat.ts`). `GET /api/admin/users` отдаёт `signup_source`, `signup_source_at`, `trial_until` и

`feedback_count`. Условие продления администратор проверяет по обращению с приложенным отчётом сам, автоматически срок не продлевается.

- **Обращения: приём.** Новый модуль `hydroweb/api/routers/feedback.py`. `POST /api/profile/feedback` принимает только `multipart/form-data`: часть `data` (JSON {kind, text}) и повторяющуюся часть `files[]` (до пяти файлов; имена `files` и прежнее `file` тоже читаются, поэтому открытая в браузере старая вкладка не ломается). Разделы `review`, `calc_issue`, `question`, `proposal`; текст до 2000 знаков (переводы строк приводятся к одному виду, управляющие символы убираются); 5 обращений за 10 минут (`FEEDBACK_PER_WINDOW`, `FEEDBACK_WINDOW_S`). Тело JSON на этом пути отклоняется («Отправьте файл формой multipart/form-data.»), предел тела знает `BodyGuard` (`UPLOAD_PATH_LIMITS` в `refusal.py`: 50 МБ плюс 256 КБ, текст отказа «Общий размер файлов больше 50 МБ.»; константы `MAX_FILES = 5`, `MAX_FILE_BYTES` 10 МБ, `MAX_TOTAL_BYTES` 50 МБ, `MAX_BODY_BYTES` = 50 МБ + 256 КБ полей формы). Чтение тела — `_read_capped` (из `supplier_applications`) с необязательными пределом и текстом отказа; `parse_form` (из `project_import`) — с необязательным `max_total` (предел всего тела отдельно от предела одной части). Форма доступна любой вошедшей записи, в том числе после конца пробного периода и поставщику на проверке.
- **Обращения: вложения.** До пяти файлов на обращение: DOCX, PDF, XLSX, .hwproj.json, PNG, JPG, каждый до 10 МБ («Файл больше 10 МБ.»), вместе до 50 МБ («Общий размер файлов больше 50 МБ.»); шестой файл, а также форма с девятью и больше частями отклоняются («Приложите не больше пяти файлов.»). `check_attachments` проверяет весь набор ДО записи любого файла (число, затем каждый файл `check_attachment`, затем сумма): негодный файл отклоняет всё обращение, на диске и в базе ничего не остаётся, а при нескольких файлах отказ называет файл. Тип определяется по содержимому и обязан совпасть с расширением (`detect_attachment_type`, `check_attachment`): PDF, JPEG и PNG по началу файла; DOCX и XLSX как ZIP с [Content_Types].xml и word/document.xml либо xl/workbook.xml (чужой архив отказ); файл проекта как JSON-объект с полями projectName и geology (BOM допускается; битый, вложенный глубже предела разбора и пустой файл отказ). Файл лежит в `feedback_files/` рядом с базой (том данных, не frontend/public) под именем <32 hex>.<тип>, права 0600, каталог 0700; исходное имя очищается от пути и служебных символов и нужно только для показа; путь из запроса не принимается (`feedback_file_path` проверяет имя из базы по шаблону и каталогу). Сведения о файлах лежат в таблице `feedback_files` (глава 3).
- **Обращения: администратор.** `GET /api/admin/feedback?status=new|handled&kind=&limit=&offset=` (дата, пользователь, раздел, текст, вложения `files: [{id, name, size}]` в порядке отправки, отметка; counts по всем обращениям; имя файла на диске не отдаётся), `GET /api/admin/feedback/summary` (число необработанных, счётчик на вкладке), `GET /api/admin/feedback/{id}/file/{id}` (скачивание одного файла; прежнего `/{id}/file` больше нет; файл выбирается по паре «обращение + идентификатор файла»; Content-Disposition attachment, nosniff, no-store; не-администратору 403, в том числе автору обращения; незнакомый id, идентификатор файла другого обращения и подделанный путь в базе дают 404), `POST /api/admin/feedback/{id}/handled` (`{handled: bool}`; в журнал действие `feedback_handled` с целью «над кем» и details.handled, без

текста обращения), `GET /api/admin/users/{id}/feedback` (карточка пользователя: раздел, вложения, отметка).

- **Обращения: хранение, миграция, nginx ([0531-обращения-до-пяти-вложений]).** Таблица `feedback_files` (глава 3) создаётся в `FEEDBACK_SCHEMA`; столбцы `feedback.file_path`, `file_name`, `file_size` остаются, не удаляются и больше не заполняются. Миграция в `_DATA_MIGRATIONS` (`db/connection.py`): запись с `feedback.file_path` переносится в `feedback_files` одним `INSERT … SELECT … WHERE NOT EXISTS (… file_path)` при каждом старте и при возврате из копии прежнего состава; идентификаторы переносимых строк — `lower(hex(randomblob(16)))`. Общий предел тела 11m контейнерного nginx не пропускал бы 50 МБ: в `deploy/nginx-site.conf` добавлен `location = /api/profile/feedback { client_max_body_size 52m; … }`, в `deploy/nginx-host.conf` — `52M`, остальной /api/ остаётся на 11m. Хостовый nginx на сервере ставится вручную (`deploy/nginx-host.conf` в `/etc/nginx/sites-available/hydroweb.online`, затем `nginx -t && systemctl reload nginx`): без этого шага форма с файлами больше примерно 11 МБ получит 413 от хостового nginx; контейнерный nginx обновляется выкаткой образа.
- **Обращения: выгрузка, удаление, копии.** `GET /api/profile/export` отдаёт `feedback[]` (kind, text, createdAt, `files: [{name, size}]` вместо прежних fileName и fileSize, handled; самих файлов нет; версия формата выгрузки 2 не менялась до [0543-журнал-событий-беты], с которым стала 3: добавлен раздел events) и в профиле `signupSource`, `signupSourceAt`, `trialUntil`. `auth.delete_account` удаляет строки feedback и feedback_files в своей транзакции (`forget_user_feedback` собирает имена файлов из `feedback_files` и, на случай ещё не перенесённой записи, из `feedback.file_path`), файлы вложений стирает после фиксации базы (`remove_feedback_files`; отсутствующий файл не мешает). Таблицы feedback и feedback_files входят в `deploy/backup_db.sh` (feedback_files — шестнадцатая), `restore_db.sh` и `restore_check.sh`; `restore_db.sh` после заливки удаляет строки feedback_files, у которых в восстановленной feedback нет обращения (копия прежнего состава). Файлы вложений в дамп не входят, об этом сказано в шапках backup_db.sh и backup_data.sh, полная копия тома (backup_data.sh) их содержит, как и supplier_cards/. В .gitignore добавлены hydroweb/data/supplier_cards/ и hydroweb/data/feedback_files/.
- **Клиент.** `services/feedback.ts` (разделы, `MAX_ATTACHMENTS = 5`, `MAX_TOTAL_ATTACHMENT_BYTES`, `checkAttachment` и `checkAttachments` — проверка по имени и размеру до отправки, части `files[]`, `FeedbackItem.files`, `fileUrl(feedbackId, fileId)`, подсказка «До 5 файлов, до 10 МБ каждый: DOCX, PDF, XLSX, .hwproj.json, PNG или JPG.», запросы администратора), `components/FeedbackForm.tsx` (раздел, текст со счётчиком до 2000, поле выбора с `multiple` и подпись «Вложения (необязательно)»; выбор дополняет список, а не заменяет, тот же файл дважды не добавляется, негодный файл отклоняется по имени с причиной, остальные остаются; список выбранных с кнопкой «Убрать» у каждого, счётчик «Приложено N из 5», при пяти файлах поле выбора неактивно; в плашке пробного периода добавлена строка о продлении), `FeedbackDialog.tsx` и пункт «Написать нам» в `UserMenu.tsx` (одним местом для редактора и списка проектов, всем вошедшим), `TrialBanner.tsx`, `TrialLockedDialog.tsx`, `TrialExpiredPanel.tsx`. Администрирование: `components/admin/AdminFeedbackTab.tsx` (вкладка «Обращения»: фильтры «Статус» и «Раздел», по умолчанию новые, текст свёрнут до трёх строк, столбец «Вложения» с

отдельной ссылкой на защищённый маршрут у каждого файла, кнопки «Обработано» и «Вернуть в новые», число новых на вкладке), `UserFeedbackDialog.tsx` (окно «Обращения пользователя»); в `AdminUsersPage.tsx` столбцы «Откуда», «Пробный до», «Обращения», пункты меню «Продлить на 15 дней», «Снять пробный режим» и «Показать обращения», в CSV столбцы «Откуда», «Пробный до (UTC)» и «Обращений».

- **Тесты.** `hydroweb/tests/test_trial_access.py` (61 собранный: регистрация с меткой и без, плохие метки и время, пересчёт срока при подтверждении, поставщик, trialUntil при входе и в /me, 403 ассистента весь срок, 403 расчёта после срока по маршрутам, срок точно до секунды, продление, снятие, список администратора, журнал, права, вход через провайдера с меткой), `hydroweb/tests/test_feedback.py` (97 собранных: разделы, пределы текста, лимит 5 за 10 минут, семь типов вложений и подделки под них, пять файлов по порядку и имена части `files[]` / `files` / `file`, шестой отклоняется, форма с 11 частями, негодный файл среди нескольких отклоняет всё и называется, 2 × 10 МБ и пять по 10 МБ принимаются, сумма сверх предела, тело больше предела отклоняется до разбора, ровно 10 МБ и на байт больше, скачивание каждого файла по своему id, идентификатор файла другого обращения 404, чужой файл 403, «обработано» с журналом, миграция прежнего вложения (переносится один раз, повторный запуск не дублирует), выгрузка «моих данных», удаление учётной записи стирает строки и файлы). Клиент: `trial.test.ts`, `components/Trial.test.tsx`, `App.trial.test.tsx`, `AppWrapper.trial.test.tsx`, `services/feedback.test.ts`, `components/FeedbackForm.test.tsx`, `components/admin/AdminFeedbackTab.test.tsx`, `components/AdminUsersPage.trial.test.tsx`.
- **Открыто.** Отдельного удаления одного обращения администратором нет: обращение и вложения живут до удаления учётной записи, как записано в политике. Метку ссылки из учётной записи администратор убрать не может, снять можно только пробный режим. Формулировки политики о метке и обращениях ждут юриста (документы/правовые_документы/редакция_2026-10-27/README.md, вопросы Ю-1…Ю-5; изменения редакции 2026-10-28 — документы/правовые_документы/редакция_2026-10-28/).

Журнал событий беты ([0543-журнал-событий-беты], 29.10.2026)

Простой журнал для администратора (план развития, месяц 1, п. 5; критерии выхода из беты — стратегия, раздел 1.4). Решение владельца 29.10: журнал нужен и остаётся простым, счётчик пробного не делается: **лимитов, квот, блокировок и показа счётчиков пользователю нет**; журнал виден только администратору, на работу пользователя не влияет (с шага 2а при включённом выключателе HYDROWEB_LEVELS_ENFORCED событие calc_fem или calc_blum «продлевает» проект в счёте активных проектов уровня — подраздел «Уровни доступа, шаг 2а»), пробный доступ по метке ([0517-пробный-доступ-по-метке]) не тронут. Ядро расчёта, числа и ключи ответов API расчёта и сверка не менялись.

- **Таблица и виды.** `events` (глава 3, схема `EVENTS_SCHEMA`); виды `KINDS` в `hydroweb/api/routers/events.py`: calc_fem, calc_blum, import_file, project_save, export_docx, export_xlsx, export_pdf, export_dxf, export_xml и export_svg (отступ от списка владельца: у чертежа три кнопки — SVG, DXF, PDF, и без SVG выгрузка чертежа в этом виде осталась бы невидимой для критерия «полный пакет»). `clean_meta` оставляет в строке

только project_id, part, engine, stages, status и new (ключи и значения проверяются `re.fullmatch`: завершающий перевод строки не проходит); `record_event` никогда не бросает, сбой пишется в лог. Запись — `INSERT … SELECT … WHERE EXISTS (SELECT 1 FROM users …)`.

- **Серверные события.** В тех же маршрутах, только для вошедшего пользователя и после успешного ответа, не на горячем пути (BackgroundTasks после ответа): `POST /api/calculate` → calc_blum (внутренний вызов из calculate-staged событий не пишет); `POST /api/calculate/fem` → calc_fem (число этапов и итоговый статус); фоновая задача `/api/calculate/fem/jobs` → calc_fem по завершению (`jobs.run_job`, запись после освобождения места расчёта; отказ и отмена событий не пишут); `POST /api/projects/import-file` → import_file (только если проект получен; вид файла и содержимое не пишутся); `POST/PUT /api/projects` → project_save: создание — всегда (new: true), повторные сохранения того же проекта — не чаще раза в `PROJECT_SAVE_EVERY_S = 600` (автосохранение иначе засорило бы журнал; пауза работает и при идентификаторе не по форме: окно по паре «пользователь, проект не назван»), 409 и 410 событий не пишут.
- **Проект расчёта.** Клиент называет проект заголовком `X-Hydroweb-Project` (случайный идентификатор проекта, не название; не по форме — игнорируется; заголовок, а не строка запроса, чтобы не менялись адреса, по которым тесты редактора сопоставляют запросы; добавлен в `allow_headers` CORS в `main.py`, предзапрос с чужого сайта — 403). Расчёт без проекта (подбор решений, прямой вызов API) пишется с `meta.src = no_project`: в сводке это отдельная колонка «Вне проекта», в колонки «МКЭ» и «Блюм» он не входит, в «Всего» входит, в критерии не идёт. Подбор решений от редактора клиент не отличает: признак один, «нет проекта».
- **Выгрузки браузера.** `POST /api/events` (глава 8) и `frontend/src/services/betaEvents.ts`: `reportEvent` — без ожидания ответа, `keepalive`, любые сбои молча, project_id не по форме не отправляется. Вызовы стоят после успешной выгрузки: `App.tsx` (расчётный отчёт DOCX, XLSX, XML — part report и движок; он же передаёт `project_id` расчётам), `DocumentsTab.tsx` (пакет: XLSX spec и DOCX note; «Спецификация DOCX» — spec), `DrawingTab.tsx` (SVG, DXF, PDF — drawing; новое поле projectId), `TeoDialog.tsx` (teo), `EstimatePanel.tsx` (estimate), `VariantComparison.tsx` и `ResultsView.tsx` (comparison), `JointCheckDialog.tsx` (узел SVG и DXF — joint); `femJobs.createFemRun(onProgress, projectId)`.
- **Сводка и критерии.** `GET /api/admin/events/summary` (глава 8): константы `ACTIVE_MIN_CALCS = 3`, `ACTIVE_MIN_PROJECTS = 2`, `ACTIVE_TARGET = 30`, `ORGANIZATIONS_TARGET = 10`, `FULL_PACKAGE_TARGET = 40`, `CALC_ISSUES_PER_10_ACTIVE = 1.0`, `WINDOW_DAYS = 30`. Активные проектировщики — не меньше трёх расчётов МКЭ в двух и более проектах за 30 суток до последнего дня периода; организации — по домену почты, публичные почтовые сервисы (`PUBLIC_MAIL_DOMAINS`) не считаются; полный пакет — по паре (пользователь, проект) выгружены note, spec и drawing, нарастающим итогом за всё время журнала по последний день периода (поле `full_package.until`); обращения calc_issue — за 30 суток и на 10 активных (не больше 1). Критерии берут только действующие записи (is_active = 1), без администраторов и поставщиков (в том числе ждущих проверки) и без адресов и доменов из переменной окружения

`HYDROWEB_EVENTS_EXCLUDE` (через запятую, по умолчанию пусто; проброс в docker-compose.yml, строка в .env.example; в таблицах такие записи остаются, на вкладке стоит пометка), и только события, чей project_id — существующий проект ЭТОГО пользователя (JOIN projects p ON p.id = pid AND p.user_id = e.user_id; связь проверяется при подсчёте, а не при записи: проект может попасть на сервер позже первого расчёта). Не считается журналом: условие «в Готовности к экспертизе нет ошибок», подтверждения экспертизы, сверка «все пять», оплаты — об этом сказано в notes ответа.

- **Вкладка «События».** `components/admin/AdminEventsTab.tsx`, `adminApi.ts` (ADMIN_TABS — девятая, «События»), `AdminUsersPage.tsx`, типы в `types.ts`: фильтр дат, блок критериев, таблицы «По пользователям» (шапка по видам, колонка «Вне проекта», строка итогов) и «По дням», без графиков. Тест `AdminEventsTab.test.tsx` (10); `AdminUsersPage.tabs.test.tsx` — девять вкладок.
- **Копии, выгрузка данных, удаление.** Таблица — семнадцатая в `deploy/backup_db.sh`, `restore_db.sh` (после заливки не чистится; пользователей, которых нет в восстановленной users, снимает приложение при старте) и `restore_check.sh`; `EXPORT_VERSION` выгрузки «Мои данные» 2 → 3, раздел events (вид и время; служебные поля не выгружаются); `delete_account` удаляет события пользователя. Политика обработки персональных данных описывает журнал в редакции 2026-10-29 ([0552]) — см. «Правовые документы» выше.
- **Тесты.** `hydroweb/tests/test_events.py` (57: таблица и индексы, белый список, POST /api/events — 401, 422 и запись по всем видам, расчёты Блюма и МКЭ и фоновая задача, импорт, сохранения с паузой 10 минут, сбой журнала не ломает маршрут, сводка — 401 и 403, по пользователям и дням, границы периода по Москве, критерии, чужой и придуманный проект не накручивают их, предзапрос CORS, запись после ответа, сироты, удаление учётной записи, выгрузка данных); vitest: `betaEvents.test.ts` (6), `AdminEventsTab.test.tsx` (10), `App.betaEvents.test.tsx` (2), дополнены `femJobs`, `DrawingTab`, `DocumentsTab.readiness`, `AdminUsersPage.tabs`; `test_every_query_parameter_of_every_route_has_a_russian_name` проверяет и названия-алиасы параметров (`from` и `to` в `refusal.py`).

Уровни доступа, шаг 1 ([0548-уровни-доступа-шаг-1], 29.10.2026)

Этап Г плана развития, шаг 1: уровни «Пробный», «Бесплатный», «Старт», «Профи», «Команда», «ВУЗ» хранятся, показываются и меняются администратором, **ничего по ним не закрывается**. Ядро, API расчёта, выгрузки и каталоги не менялись. Матрица уровней, вопросы владельцу и точки закрытия шага 2 — в плане развития (раздел 3.1, «Этап Г, шаг 1»). Проверки по уровню появились в шаге 2а (следующий подраздел) и включаются выключателем, по умолчанию выключенным: пока он выключен, всё сказанное здесь верно.

- **Модуль.** `hydroweb/api/access_levels.py`: LEVELS и LEVEL_LABELS; `LEVEL_MATRIX` — машинная запись таблицы стратегии (раздел 2.1) без цен, где стратегия не решила, поле open называет вопрос владельцу; `level_state` по порядку: поставщик — уровней нет; назначен и не истёк — он (source assigned); срок вышел — FALLBACK_LEVEL «Бесплатный» (source expired, прежний уровень в expiredLevel, хранимое значение не меняется и журнал не пишется); не назначен, но пробный период по метке — «Пробный» со сроком trial_until (source trial); иначе BETA_DEFAULT_LEVEL = pro (source default, beta = true, «Профи (бета)»). `LEVELS_ENFORCED` — флаг enforced в ответе (в шаге 1 был константой False; с шага 2а

читается из переменной окружения HYDROWEB_LEVELS_ENFORCED, по умолчанию False). Пробный режим по метке и уровень — два независимых механизма (trialActive в ответе); единый механизм — шаг 2. Администратор уровнями не ограничивается никогда (unrestricted). `admin_set_access_level`: одна транзакция (users + строка access_level_changes), причина обязательна и не длиннее REASON_MAX_CHARS = 300, срок «ГГГГ-ММ-ДД» — конец суток по Москве включительно, в будущем и не дальше UNTIL_MAX_YEARS = 5 лет, у «Пробного» обязателен, у «Бесплатного» и уровня по умолчанию невозможен.

- **API и данные.** Маршруты и таблицы — главы 8 и 3. Названия полей запроса «уровень доступа» и «срок уровня» добавлены в _FIELD_RU (`api/refusal.py`): тест test_every_field_of_every_request_schema_has_a_russian_name требует название у каждого поля тел запросов. В GET /api/admin/users и ответе приглашения — access_level, access_level_until, effective_level, effective_level_label, level_source, level_expired; в выгрузке «Мои данные» — profile.accessLevel и accessLevelChanges. Журнал смен удаляется вместе с учётной записью; в суточную копию БД не входит (глава 3).
- **Клиент.** `frontend/src/accessLevels.ts` (названия, плашка «Ваш уровень: Профи (бета)», срок «до ДД.ММ.ГГГГ»; сверка состава и порядка уровней с сервером — test_levels_of_the_server_equal_the_levels_of_the_client); `components/admin/AccessLevelDialog.tsx` (уровень, срок, причина, журнал смен; открывается щелчком по столбцу «Уровень» и пунктом меню «Уровень доступа…»; строка списка меняется только после ответа сервера; говорит, что функции по уровню не закрываются (при enforced — «Ограничения по уровню действуют.»), и предупреждает о действующем пробном режиме по метке); столбец «Уровень» в `AdminUsersPage.tsx` и закреплённый у правого края (sticky) столбец «⊠»: таблица пользователей и до правки была шире окна (1176 px при 1150 видимых на экране 1280), столбец «Уровень» добавил около 130 px, и меню строки уходило за край; `ProfilePage.tsx` — плашка без счётчиков и перечня лимитов; `auditFormat.ts` — «Смена уровня доступа».
- **Тесты.** `hydroweb/tests/test_access_levels.py` (49: состав уровней и матрица; какой уровень действует; 401, 403, 404; назначение, журнал, аудит без причины; срок и ошибки ввода; атомарность; список пользователей; выгрузка данных и удаление учётной записи; миграция старой базы; откат на прежний образ; журнал не в копии БД); vitest: `accessLevels.test.ts` (11), `AdminUsersPage.level.test.tsx` (11), дополнения `ProfilePage.test.tsx` и `auditFormat.test.ts`.
- **Открыто (вопросы владельцу, записаны в план, не решены этим шагом).** Пробный: стратегия (7 дней, весь «Профи») против кода [0517] (15 суток по метке, ассистент и подбор закрыты); «Подбор решений» входит в «Бесплатный», но считает МКЭ, а МКЭ — «Профи» (решено 30.10.2026: у подбора свой маршрут, шаг 2а); выгрузки строит браузер, и серверная проверка уровня для них невозможна без подписанного разрешения или переноса сборки; «общие проекты» у «Команды» в коде нет; уровень и журнал смен — новый состав данных, политика конфиденциальности его не называет (правовая редакция до платных уровней).

Уровни доступа, шаг 2а ([0570-уровни-доступа-шаг-2а], 30.10.2026)

Этап Г плана развития, шаг 2а: серверные проверки действий по уровню доступа за выключателем `HYDROWEB_LEVELS_ENFORCED`. **По умолчанию выключено**: ответы прежних маршрутов не меняются, пользователь ничего нового не видит. Ядро расчёта не менялось. Решения владельца от 29.10 и 30.10.2026 записаны в плане развития («Этап Г, шаг 2а») и взяты как заданные.

- **Выключатель.** `access_levels.LEVELS_ENFORCED` читает переменную HYDROWEB_LEVELS_ENFORCED один раз при запуске процесса: «1» — включено, всё прочее (пусто, «0», «true») — выключено, нестандартное значение пишет предупреждение в журнал. Проброс в контейнер — docker-compose.yml, строка — .env.example (глава 2); смена значения — перезапуск контейнера. Флаг enforced в accessLevel (/api/auth/me, /api/profile, ответ входа) следует за выключателем. Выключено — каждая проверка возвращает «разрешено» до обращения к базе и ничего не считает.
- **Модуль.** `hydroweb/api/access_gate.py`: действия assistant_question, solution_finder, fem_calculate_pro, project_create и вход для расчёта; функции check_feature (возможность из LEVEL_MATRIX[...]["features"]), guard_project_create и count_active_projects, assistant_plan и assistant_exhausted, guard_login и guard_fem. Администратор (unrestricted) и поставщик (level_state пуст) проверкой по уровню не затрагиваются; вход для расчёта нужен всем. Отказ по уровню — LevelRequiredError, приложение отвечает 403 одной формы `{detail, code: "level_required", level, required, action}`: detail — одна русская строка не длиннее 20 слов, required — младший уровень лестницы «Бесплатный → Старт → Профи» выше текущего, на котором действие доступно (null — предложить нечего). Без действующей сессии при включённом выключателе — 401 «Для расчёта войдите в учётную запись.»

Действие	Маршрут	Что проверяется
`assistant_question`	`POST /api/assistant/chat`	Лимит ассистента из матрицы уровня за календарный период по Москве: «Бесплатный» — 5 вопросов в месяц (решение владельца 30.10.2026; было 5 в сутки), «Старт» — 30 в месяц, прочие уровни — 300 в месяц. Счёт — таблица assistant_usage
`solution_finder`	`POST /api/calculate/fem/finder`	Возможность solution_finder: есть на всех уровнях. Границы запроса — ниже
`fem_calculate_pro`	`POST /api/calculate/fem`, `/jobs`	Возможность fem — с уровня «Профи»; ниже отказ «Недоступно на уровне «...»: нужен уровень «Профи».»
`project_create`	`POST /api/projects`	Лимит projects.max_active: «Бесплатный» — 1 активный проект, «Старт» — 5, остальные без лимита. Проверяется только создание (ветка «записи ещё нет», после проверок 409 и 410)

- **Подбор решений — отдельный маршрут.** Различать подбор и профессиональный режим по заявлению клиента (полю запроса) нельзя: любой запрос можно объявить подбором. Поэтому источник расчёта определяет маршрут, а не поле запроса, и у подбора заведён

`POST /api/calculate/fem/finder`: тот же расчёт (общее тело _run_fem), но запрос ограничен тем, что строит мастер (`hydroweb/api/finder_limits.py`). Границы двух видов. Белый список присланных полей: ALLOWED_FIELDS (31 поле верхнего уровня) и ALLOWED_NESTED_FIELDS (поля слоёв, этапов, связей и пригрузок) равны ключам снимка запросов самого мастера — `hydroweb/tests/fixtures/finder_fem_requests.json` (78 запросов, семь сценариев); проверяются поля, присланные клиентом (model_fields_set), умолчания модели не в счёт; лишнее — 422 «Поле «…» в подборе не используется. Такой расчёт — в профессиональном режиме.» Состав расчёта (finder_problem), выведенный из services/solutionEngine (quickInputToProject.ts, supportSizing.ts): не больше двух этапов, не больше одной связи (распорка или ярус анкеров), без грунтовых распорок, берм, среза, подсыпки, нагрузок на стенку и водоносных слоёв, не больше одной пригрузки; иначе 422 одной русской строкой «Подбор решений не считает … Такой расчёт — в профессиональном режиме.» Границы действуют при любом положении выключателя: маршрут новый, его единственный клиент — мастер. Это ограничение состава запроса, а не доказательство, что считает именно мастер: пользователь, шлющий запросы сам, может посчитать на этом маршруте любую схему из разрешённых. Клиент: runFemCheck получил опцию finder (адрес FEM_FINDER_URL), её ставит solutionGenerator; автоподбор профиля во вкладке «Ограждение» и проверка вариантов ТЭО остались на профессиональных маршрутах (при включённых проверках «Бесплатному» закрыты — решение владельца 30.10.2026). Классический расчёт и проверка анкеров мастера — прежние общие маршруты. Фоновых задач у подбора нет (мастер зовёт runFemCheck синхронно), маршрута /finder/jobs нет.

- **Вход для расчёта.** При включённом выключателе `POST /api/calculate`, `/api/calculate/fem`, `/api/calculate/fem/jobs` и `/api/calculate/fem/finder` без действующей сессии отвечают 401; выключено — как раньше. Остальные расчётные маршруты вход пока не требуют — шаг 2б. Пробный период по метке ([0517]) остаётся отдельным механизмом: его мидлвар отвечает раньше проверок по уровню.
- **Ассистент.** Лимит уровня и порядок отказов — в разделе «ИИ-ассистент: суточный лимит и /api/assistant/quota» (глава 8). Месячный счёт ретроактивный: при включении выключателя считаются вопросы всего текущего месяца.
- **Проекты.** Активный проект — тот, что правился или считался за ACTIVE_PROJECT_DAYS = 90 суток: правка — projects.updated_at (её обновляет каждая запись проекта), расчёт — событие calc_fem или calc_blum журнала событий с этим проектом (events.meta_json.project_id, заголовок X-Hydroweb-Project). Журнал событий стал основанием лимита; запись события — «по возможности», поэтому точность лимита ограничена ею. Миграции данных нет. Гонка лимита закрыта: guard_project_create сначала считает без блокировки и отказывает сразу, если лимит исчерпан; иначе перед вставкой захватывает запись (BEGIN IMMEDIATE), считает заново, а вставка и commit идут в той же транзакции (8 параллельных созданий у «Бесплатного» дают ровно одно, 12 у «Старта» — ровно пять). Лимит из матрицы читается первым: у «Профи», «Команды», «ВУЗа» и у записей без назначенного уровня подсчёт не выполняется и журнал событий не читается.
- **Клиент.** `accessLevels.ts`: LEVEL_REQUIRED_CODE, isLevelRequired (по коду), levelRequiredLine (показывает строку detail сервера, при исчерпанном лимите она называет и сам лимит; нет её — собирает «Недоступно на уровне «…»: нужен уровень «…».» по

подписям уровней клиента), ENFORCED_LINE («Ограничения по уровню действуют.»); `services/apiDetail.ts`: apiErrorText; типы LevelRequiredBody и LevelGateAction. Отказ по уровню и 401 показываются строкой сервера: в расчёте МКЭ профессионального режима (App.tsx, без префикса «Ошибка МКЭ-расчёта»), в runFemCheck, в ассистенте (assistantService.ts) и при синхронизации проектов (projectStore.ts: 403 level_required → «… Проект сохранён только в этом браузере.», локальная копия цела, повторов нет). При enforced плашка профиля и окно «Уровень доступа» администратора показывают ENFORCED_LINE вместо «ограничений нет»; у администратора в плашке пояснения нет, а в окне уровня записи администратора добавлено «Администратора они не касаются.» Новых компонентов нет.

- **Сверка и смоук.** При включённом выключателе расчёт требует вход, поэтому `scripts/gw_compare/prod_sweep.py` (`make_caller`: по адресу --url и локально) и `deploy/smoke_check.py` входят тестовой записью: логин и пароль в переменных HYDROWEB_SWEEP_LOGIN и HYDROWEB_SWEEP_PASSWORD, токен берётся из ответа POST /api/auth/login и уходит заголовком Authorization (scripts/gw_compare/README.md, раздел «Запуск»). В репозитории паролей нет; не заданы — запросы идут без входа. Локально (--local) вход выполняется, только если выключатель включён: иначе вход не нужен, а записи в локальной базе нет. epure_prof кэширует вызывающего на адрес (вход один, а не на каждый запрос: лимит входа 10 за 5 минут). Для смоука из контейнера переменные передаются ключами: `docker exec -i -e HYDROWEB_SWEEP_LOGIN -e HYDROWEB_SWEEP_PASSWORD hydroweb python - http://127.0.0.1:8000 < deploy/smoke_check.py`. Тестовая запись обычная, с уровнем «Профи» (расчёт идёт маршрутом /api/calculate/fem); единственная запись, которую смоук делает на сервере, — отметка последнего входа (last_login). Маршрут подбора ни сверка, ни смоук не трогают (закреплено тестом).
- **Откат.** Клиент мастера и маршрут подбора едут одним образом, откат возвращает оба (deploy/rollback.md). Открытая вкладка с новым клиентом против старого сервера получит на /api/calculate/fem/finder ответ 404 «Не найдено.», и мастер напишет «Проверка МКЭ не выполнена: Не найдено.»: расчёт не искажается, вкладку достаточно обновить.
- **Копия БД.** assistant_usage в суточную копию не входит намеренно (комментарии deploy/backup_db.sh и db/connection.py). Раньше таблица была суточным счётчиком, теперь это месячная квота: восстановление в работающую базу счётчик не трогает, потеря тома и восстановление в новую базу посреди месяца обнуляют израсходованные вопросы — в пользу пользователя.
- **Тесты.** `hydroweb/tests/test_access_gate.py` (63: выключатель и его проброс в compose; матрица «уровень × действие»; администратор и поставщик; МКЭ-маршруты — профессиональный режим закрыт ниже «Профи», подбор на своём маршруте открыт; вход для расчёта — 401 без сессии на четырёх маршрутах при включённом выключателе, с сессией (заголовок и кука) и при выключенном — как прежде; смоук и харнес сверки входят тестовой записью, пароль не печатается; ассистент — месячный и суточный периоды, граница месяца по Москве, лимит роли рядом с лимитом уровня, параллельные запросы не проходят лимит; проекты — 1 и 5 активных, граница 90 суток, расчёт из журнала событий, параллельные создания; пробная запись при включённых проверках; форма отказа) и `hydroweb/tests/test_finder_route.py` (83: 78 настоящих запросов мастера из снимка —

допустимое тело и в границах подбора; предел этапов и связей равен максимуму в снимке; белый список равен ключам снимка; запросы профессионального режима — этапы, связи, берма, срез, подсыпка, нагрузки, пригрузки, водоносный слой — на маршруте подбора дают 422 одной русской строкой; полный ответ /finder равен ответу /fem, кроме полей elapsed_*). Снимок запросов мастера переснимается командой UPDATE_FINDER_FIXTURE=1 (тест finderFemRequests.test.ts сверяет его с тем, что мастер шлёт сейчас). vitest: accessLevels.test.ts, femCheck.test.ts (маршрут подбора по опции finder, строка отказа), assistantService.test.ts, projectStore.sync.test.ts, ProfilePage.test.tsx и AdminUsersPage.level.test.tsx (подсказки при enforced), finderTransfer.test.ts (мастер считает на маршруте подбора).

- **Не сделано и открыто.** Оставлено на шаги 2б и 2в: объединение пробного режима и уровня; роли standard и professional → уровни и снятие лимита роли; остальные расчётные маршруты по функциям и вход для них; просмотр сверх лимита после понижения уровня; запрос цены и импорт файлов; показ закрытого и остатка ассистента за месяц в панели; выгрузки (интерфейс и пометки; серверная защита — после месяца 3); письмо за 7 дней; режим «ВУЗ». Вопрос владельцу: у «Старта» в матрице есть auto_profile, но нет fem, а автоподбор профиля (класс Б) и варианты ТЭО проверяют МКЭ по маршруту профессионального режима; при включённых проверках «Старт» получит «Проверка МКЭ подобранного профиля не выполнена: Недоступно на уровне «Старт»…» (код не менялся). Перед включением выключателя: решение о назначении уровней, правовая редакция, тестовая запись для сверки и смоука.

Интерфейс редактора: границы ошибок, вкладки, вердикт, шаблон проекта

- `ErrorBoundary.tsx` + `errorReport.ts` — три уровня: всё приложение (AppWrapper экспортирует обёртку, прежний компонент — AppWrapperInner), редактор проекта (внутри AuthProvider; «Вернуться к проектам» = handleBackToProjects) и содержимое вкладки в App.tsx (variant="panel", key={activeTab}: сбой вкладки не роняет шапку и соседние вкладки). Текст сведений об ошибке — имя, сообщение, стек, область, время, браузер и ТОЛЬКО путь страницы без query и hash; без пользователя и данных проектов; при недоступном буфере показывается textarea.
- `TabNavigation.tsx` + `editorTabs.ts` (EDITOR_TABS: geology, wall, loads, stages, anchors, results, drawing, docs; подписи — TAB_LABELS в constants.ts) — прокручиваемая полоса со стрелками поверх неё (иначе на краю прокрутки вкладки «дрожали»), role=tab/aria-selected, scrollIntoView активной вкладки при открытии и программной смене.
- **Вкладка «Этапы»** ([0453-интерфейс-прототип], 24.10.2026). `StagesTab.tsx` — ввод этапов (глубина, УВ слева и справа, поверхность тыла, ИГЭ подсыпки), УГВ, водоупор, бермы, водоносные слои (AquiferSection.tsx) и таблица «Итоги по этапам» (Макс и этапы 1…N: H, Км, Кз, Mmax, Qmax, wmax, усилия связей и запас К, признак сходимости); вкладка при открытии ничего не пишет в проект (тест). Анкеры, распорки и грунтовые распорки остались в AnchorsTab.tsx тремя таблицами подряд, подвкладок нет; геология, бермы и водоносные слои — таблицы той же формы (GeologyTab.tsx, AquiferSection.tsx). Общие части — DenseTable.tsx (таблица, RowTools: ▲ ▼ ⎘ × и «+ Добавить»), числа — одна точность на столбец через digits в useNumericField.ts (показ вне фокуса; хранимое значение не меняется). Пояснения вынесены в HelpTip.tsx («?»); на экране остаются подпись, единица, значение и красная строка ошибки.

- **Таблицы связей ([0503-прототип-таблица-распорок], 26.10.2026).** `AnchorsTab.tsx`: у анкера столбцы Вкл, Тяга, Эт.вкл, Эт.выкл, h, Угол, Гор.угол, Шаг, Ls, Lk, C, F, Fm; у распорки Вкл, Сечение, Пояс, Эт.вкл, Эт.выкл, h, Угол, Гор.угол, Шаг, L, C, Fm; у грунтовой распорки Вкл, Эт.вкл, Эт.выкл, h, Шаг, C, Fm (поле `spacing`, у новой строки 3 м). Ячейки «Тяга», «Сечение» и «Пояс» — кнопки класса dt-pick (обозначение и сталь, у пары двутавров «×2»; пусто — «выбрать» или «нет»; A и стандарт — во всплывающей подсказке): сечение и пояс открывают прежний `StrutDetailDialog` (поле «Длина» из него убрано, окно читает L из строки; группа названа «Сечение распорки»), тяга — новый `AnchorTendonDialog.tsx` («Тяга анкера N»: каталог `ANCHOR_TENDONS` и тяги поставщиков, число тяг n, «ОК» пишет тягу и C = E·A·n/Ls, «не выбрана» + «ОК» снимает тягу, C остаётся). Кнопка «L», строки «Ярус N: …» и «Тяга из каталога» под таблицами и компонент `AnchorTendonPicker.tsx` удалены; кнопка с галочкой, «Узел» и «W» остались в конце строки. `core/supportStiffness.ts` — одни формулы для окон и строки: `strutStiffness` (E·A·100/(L·1e6)), `strutMaxForce` (A·Ry: запасное значение Fm, когда ответа сервера нет или он без поля limit_force_strength_kN, и значение для строки без сечения; число то же побитно, что у сервера, — тест), `strutStiffnessForLength` и `anchorStiffnessForLength` (C при новой длине, округление 0,01 кН/мм; null — сечение или тяга не выбраны, пересчитывать нечего): правка L или Ls в строке меняет C, только если сечение или тяга выбраны; Fm у анкера от длины не зависит, а у распорки с сечением по устойчивости пересчитывается вместе с C (пункт ниже); ручная правка C остаётся до следующей правки длины. Тест сверяет, что пересчёт в строке и «ОК» окна дают одно число.
- **Fm распорки по устойчивости ([0544-fm-распорки-по-устойчивости], 29.10.2026).** Fm = floor(min(A·Ry; φ·A·Ry·γc)), где φ·A·Ry·γc — поле `limit_force_stability_kN` ответа `POST /api/checks/struts` (блок «Центральное сжатие — Устойчивость»; от N, M и угла не зависит — защитный тест `test_limit_force_stability_does_not_depend_on_axial_force_or_moment`), Ry = Ryn/1,05. Прочность A·Ry — тоже поле ответа (`limit_force_strength_kN`, [0569-релиз-6-20-w-марка-ary]: Ry = Ryn/γm, тот же, что в блоке устойчивости; от N, M, угла и длины не зависит; считает `check_strut` в `core/strut_check.py` рядом с устойчивостью, поле результата `StrutCheckResult`, роутер его пробрасывает): `strutFmFromCheck` берёт его из ответа, а если поля нет (сервер прежней версии) или оно не число — считает прежним клиентским `strutMaxForce`; число то же побитно (тесты — в следующем пункте). Запрос уходит с N = 0, длиной из строки и принятыми по умолчанию μ = 1,0, γc = 0,9; целые кН округляются вниз и не меньше 1 кН (`fmToKN`: у своего сечения с малой A число по устойчивости доходило до нуля, а `max_force` сервера расчёта не меньше 0,001 кН, иначе 422). Следствие: K = Fm/|F| распорки равен запасу Ks блока «Устойчивость» окна «Расчёт распорки» (с точностью до округления Fm). Основание хранится в `StrutRow.maxForceBasis` ({kind: stability | manual, governing, strengthKN, stabilityKN, phi, slenderness, curve, mu, gammaC, inputs}); состояние строки — none, stability, manual, strength (записан по прочности до [0544]) или stale (сечение, сталь или длина изменились). Файлы: `core/strutFm.ts` (выбор меньшего, состояние строки, подписи, FM_HELP_TEXT, fmLimitsForceNote), `services/strutFm.ts` (запрос с N = 0), `hooks/useStrutFm.ts` (автопересчёт строк, у которых основание «по устойчивости», а сечение, сталь или длина изменились, и кнопка «Пересчитать Fm» для записанных по прочности), `components/StrutDetailDialog.tsx` и `AnchorsTab.tsx` (подпись под Fm и строка «Fm

распорки 1, 3 рассчитан по прочности; пересчитать по устойчивости»), `types.ts` (`StrutSectionCheck.limitForceStabilityKN` и `limitForceStrengthKN`), `services/sectionChecks.ts` (поле клиентского вида; границы входа `STRUT_CHECK_LIMITS` и `strutCheckable` рядом с запросом), `services/projectFieldFlow.ts` (struts[].maxForceBasis.* — вид «отчёт»), `core/supportChecks.ts`, `reports/modelAssumptions.ts`, `generateXml.ts` (необязательный атрибут связи `ОснованиеПредельногоУсилия`), `generateXlsx.ts` (лист «Крепление»: графа «Основание макс. силы» с ф и λ; на её месте раньше начинался «Эт.вкл», графа «Учёт в расчёте» сужена с трёх столбцов до двух), `generateDocx.ts` (оговорка «N_пред задаёт проектировщик или принимается по сечению распорки — меньшее из прочности A·Ry и устойчивости ф·A·Ry·γc»). Предел ожидания ответа сервера проверок — 15 с (`postJson`: AbortController и гонка с таймером, работает и при подмене fetch; `setSectionChecksTimeout` для тестов): по истечении окно отпускает «ОК», Fm по прочности, подпись «Устойчивость не рассчитана: сервер не ответил», строка таблицы потом получает замечание с кнопкой. Расчёт стенки: Fm по-прежнему уходит в запрос числом `max_force`; в [0544] ядро, формулы `strut_check.py` и API не менялись (в `test_strut_check.py` один защитный тест добавлен); в [0569-релиз-6-20-w-марка-ary] в `StrutCheckResult` добавлено одно поле (формулы СП 16 прежние), а в ответ `POST /api/checks/struts` — один ключ.

- **Fm по устойчивости: что не входит, замер, открыто.** Внецентренное сжатие (блок 4, capacity4) в Fm не входит: эксцентриситета и момента в проекте нет, а при M от собственного веса блок зависит от N из расчёта, и Fm стал бы зависеть от результата, которым сам задаётся; появится поле эксцентриситета — добавляется одной строкой в `services/strutFm.ts`. γc = 0,9 — допущение программы, не значение из таблицы СП 16. μ — не из проекта: закрепление концов задаётся только в окне «Расчёт распорки» на время сессии. Замер (`scripts/gw_compare/research/strut_fm_stability.py`): 17 распорок из 15 запросов набора из 56, у которых печать эталонного отчёта называет трубчатое сечение, сопоставленное с запросом по жёсткости EA/L (⩽ 10 %); сталь 10, Ryn 205; F — осевое усилие из расчёта МКЭ на определяющем этапе. Fm по устойчивости ниже A·Ry у всех 17: отношение 0,53…0,90 (медиана 0,77 = 0,9·ф), ф 0,591…0,997, λ 16…114; K падает на 10…47 % (наименьшее после — 1,64, запрос 1729.6: Д219×5, L = 5 м, F = 313,5 кН); вердикт «N ⩽ N_пред» не меняется ни у одной, числа стенки в режиме по умолчанию совпадают побитно. Это верно, пока Fm — только база запаса: в `main.py` force_limited = not (strut_tension and not strut_force_limit and inclination_angle == 0), то есть у раскоса, при выключенном «Распорка воспринимает растяжение» и при strut_force_limit Fm — предел усилия, и при F > Fm по устойчивости числа расчёта меняются (1729.6 с Fm = 200 кН < F = 313,5 кН при выключенном растяжении: M 139,3 → 84,4, w 16,96 → 5591 мм, Kз заделки — ориентир, не норматив — 1,43 → 1,0: модель считает отказ распорки); в окне сечения для этих случаев стоит янтарная строка. Решения владельца ждут: γc в Fm, μ и эксцентриситет в проект, оговорка в интерфейсе для раскоса и выключенного растяжения, пересчёт Fm при переносе из «Подбора решений» (там Fm по прочности: перенесённая распорка получает замечание и кнопку). Тесты: `core/strutFm.test.ts` (40: среди них по одному тесту на каждое сечение файла сверки — поле сервера берётся, поля нет или NaN — тот же Fm и то же основание), `StrutDetailDialog.strength.test.tsx` (2: число сервера берётся, а не пересчитывается; сервер вернул null или поля нет — тот же Fm по счёту клиента), `StrutDetailDialog.test.tsx` (34), `AnchorsTab.strutSection.test.tsx` (42),

`reports/strutFmPrint.test.ts` (6); `hydroweb/tests/test_strut_strength_limit.py` (4: ядро `check_strut`; маршрут; тот же Ry при смене γm; на сечениях труб, двутавров, швеллеров и сдвоенных двутавров файла сверки число сервера равно прежнему клиентскому A·Ry побитно, а Fm = floor(min(...)) тот же) — вынесены в отдельный файл, потому что книга называет число тестов в `test_strut_check.py` и `test_api_checks_supports.py` (в последнем ключ ответа добавлен в проверку состава).

- **Тяги по ТУ в окне «Тяга анкера» ([0524-каталог-анкеров-по-ту]).** `AnchorTendonDialog.tsx` получил группу «Анкеры по ТУ 25.11.23-001-22252558-2024» (23 штанги, подпись «ECO40*12, Ø40/16, F разр. 660 кН»); данные окна — `frontend/src/data/groundAnchorsTu.generated.ts` (только штанги; комплектующие остаются в `ground_anchors_tu.json` и в ответе `GET /api/ground-anchors-tu`, тип `GroundAnchorsTuCatalog` в `types.ts`). Жёсткость C = E·A·n/Ls. В ТУ нет площади сечения штанги и модуля упругости: поле `area_mm2` пусто, для жёсткости берётся оценка `area_from_mass_mm2` = масса 1 п. м / 7850 кг/м³ (плотность стали, СП 16.13330.2017, табл. Б.1), E = 206000 МПа (`hydroweb/core/steel_constants.py`, СП 16.13330.2017, п. 5.1, табл. Б.1), а не 200000 МПа арматуры, как у стержней ГОСТ 34028 в `anchor_tendons.json`; погрешность оценки площади относительно паспортной не установлена (у ECO-A масса точно равна массе трубы D/d, у ECO ниже геометрии на 4–25 %, поэтому расчёт по D и d завысил бы A и C до 33 %). «ОК» пишет в строку снимок штанги и C, `Fm` строки окно не меняет. Графа ТУ «несущая способность по материалу» не расшифрована (нормативное это значение или расчётное, коэффициенты надёжности не заданы) и в R_t;d проверки анкеров не подставляется. Чего в ТУ нет, записано в поле `gaps` каталога: площади сечения, предела текучести и временного сопротивления в МПа, модуля упругости, длин штанг, длины корня, шага, несущей способности закрепления анкера в грунте, нормирования антикоррозионной защиты, соответствия видов долот типам грунта. Аномалии самого ТУ оставлены как напечатаны, помечены `note` и закреплены тестами: ECO-A103*23 (внутренний диаметр 77 мм не сходится с обозначением, вероятна опечатка), равные усилия у ECO103*26 и ECO-A103*26 при разной массе, масса плиты ECO73 200×200×40 и ECO-A103 200×200×50 больше массы сплошной плиты таких размеров, материал группы ECO-A слитно в текстовом слое, материал 42CrMo у долот ECO103/175 EXX. Разрывное усилие ECO200*53,5 в ТУ 14320 кН (оба извлекателя дали это значение); в справочном каталоге другого изготовителя для той же модели 14230 кН, вероятно перестановка цифр в одном из двух документов, внесено значение ТУ. Текстового слоя нет у с. 1 и 32–33 основного файла и у файлов титула и каталожного листа, поэтому даты утверждения ТУ в каталоге нет; файл ТУ лежит в `рабочие_материалы/` (в git его нет). Тесты: `TestGroundAnchorsFromTU` в `test_data_integrity.py` (число записей и таблиц, источник и страница у каждой записи, значения 23 штанг как напечатано, d = D − 2t с одним помеченным исключением, оценка по массе с основанием, E равен константе свода правил, условия и `gaps`), три теста в `test_anchor_tendons.py` (эндпоинт и побитно прежний ответ `/api/anchor-tendons`, актуальность генерации, второй проход по PDF — пропуск, если нет PDF или `pdftotext`), `AnchorTendonDialog.tu.test.tsx`. Не сделано: привязка анкера по ТУ к проверке корня по грунту (ТУ закрепления в грунте не нормирует), сопоставление позиций поставщиков категории «Анкерные тяги» с моделями ТУ.
- **Тяга в проекте.** Единственное новое поле проекта — `anchors.rows[].tendon` (`AnchorTendonChoice` в types.ts: id, designation, standard, count, area_mm2, E_MPa — снимок

позиции каталога, как `StrutDetail` у распорки). В запрос оно не идёт (реестр `projectFieldFlow.ts`, вид «ui»; в расчёт идёт записанная жёсткость C), без него ячейка не показала бы выбранную тягу. Побочное следствие: id позиции поставщика (supplier:<id>) теперь лежит в данных проекта, и спрос поставщика (`supplier._selected_refs`) считает такую тягу в столбце «В проектах». Сторож `AnchorsTab.strutSection.test.tsx` (32 проверки): открытие вкладки ничего не пишет в проект; тела запросов МКЭ и Блюма байт в байт прежние на десяти проектах (демо-проект с трубой и поясом, он же с тягой анкера и грунтовой распоркой, 2182.1-p84, семь SCHEME_CASES); каждая из девяти ячеек строки распорки меняет в запросе ровно своё поле; «ОК» окна тяги меняет только stiffness; «Шаг» грунтовой распорки — только spacing. Остальное: `AnchorTendonDialog.supplier.test.tsx` (5), `supportStiffness.test.ts` (5), `AnchorsTab.tables.test.tsx` (5).

- **Цифры в таблицах ввода.** Решение [0453] (14 px, шрифт интерфейса, табличные цифры) записано в классе `.dt-in`, но общее правило input[type="number"] (JetBrains Mono 13 px) было сильнее одного класса, и с релиза 6.5 числовые ячейки всех таблиц ввода и числовые поля вкладки «Ограждение» шли моноширинным шрифтом 13 px. В `index.css` добавлена переменная `--font-num` (системный шрифт) и правило `input.dt-in, .dt input[type='number'], .num-14 input[type='number']` после общего: 14 px, `font-variant-numeric: tabular-nums`; текстовые ячейки таблиц (`.dt tbody td`) — тем же шрифтом, класс `num-14` стоит у вкладки «Ограждение». В DM Sans из комплекта сайта нет табличных цифр и кириллицы, поэтому шрифт интерфейса для цифр не выбран. Сторож — `DenseTable.digits.test.ts` (2). В таблице «Корень анкера: проверка по грунту» единицы и «?» в заголовках, цифры целые.
- **Правая панель** `StagePanel.tsx` (видна с любой вкладки; заменила статичный разрез и плашку «Эпюра: нет / σ / M / Q / w»). Плашка этапов «1 … N» и плашка эпюр «Давление, M, Q, w» с множественным выбором; выбор этапа общий для схемы и эпюр, по умолчанию — определяющий этап расчёта. Эпюры берутся из stages[].stage_diagrams и shear_elements (stageCharts.ts) и накладываются на схему этапа вдоль оси стенки: геометрию и подписи считает schemeModel.ts (SchemeEpure, buildEpureShapes), рисует drawScheme в schemeRenderer.ts; нулевая линия эпюры — ось стенки, давление — тыл влево и котлован вправо с пределами пунктиром, M, Q, w — по знаку; масштаб на эпюру (максимум — 35 % ширины половины схемы). Подписи, попавшие под линию эпюры, переносятся в ближайшее свободное место с выноской (LabelLayout.relocate); тест schemeLabels.test.ts — 240 схем, пересечений и выходов за кадр нет. Метод Блюма — общие эпюры результата без прогиба. Чип включённой эпюры показывает цвет и максимум выбранного этапа. Подэтап «k.F0» показан отдельной вкладкой плашки (следующий пункт). Эпюра p_net в ответе есть, на экране не рисуется.
- **Подэтап «k.F0» на плашке** ([0480-подэтап-натяжения-плашка], 24.10.2026; ядро, числа и ключи ответа не менялись). `core/substage.ts`: `substageRecord` (запись `stages[]` с `substage: 'F0'` и номером этапа k), `stagesWithSubstage`, `f0AnchorIds` (включённые анкеры этапа k с преднатягом больше нуля), `substageCaption`, `substageFacts` (итоги: усилие анкера, M, Q, w, Км, Кз; у подэтапа до откопки Кз (ориентир, не норматив) назван «справочно», нет числа — «не определяется»), `SUBSTAGE_HELP`. `StagePanel.tsx`: вкладка «k.F0 Натяжение анкера» между этапами k−1 и k (цвет анкера на схеме; не помещается плашка — оценка ширины по числу этапов и подэтапов — подпись «k.F0», полное название

в подсказке); схема — `schemeModel.ts` (поле `substage`: состояние этапа k−1, новый анкер как сила, `SchemeAnchorShape.f0`; для k = 1 исходное состояние до откопки, как в `staging._f0_plan_stage`), `schemeRenderer.ts`; эпюры — из записи подэтапа (`stageCharts.ts`). Вкладка видна только при свежем ответе МКЭ и не у метода Блюма; число этапов, нумерация и определяющий этап не меняются. Сервер не менялся: запись подэтапа уже несёт `stage_diagrams`, `shear_elements`, `H_exc` этапа k−1 (для k = 1 — 0), усилие анкера, равное F0, Км и Кз (`test_api_substage_diagrams.py`). Стенд `dev-prototype.html`: `?f0=3`, `?sample=2023.1`, `?sub=F0`, `?openhelp=1`. Открыто: что делать с Кз подэтапа «1.F0» (оставить справочным, писать «не определяется» или убрать; число ядра по печатям эталона в допуске у одного подэтапа из пяти, [0468]); показывать ли вкладку по вводу, до расчёта; чертёж и выгрузки подэтап не показывают.

- **Снятые с экрана органы.** Сняты способ уплотнения, метод бокового давления, тип давления воды, Sr, w, e, n, степень разложения торфа, значения II группы (подпись «в расчёт идёт набор I»), Fs анкера, «Расчёт 0 этапа» и переключатель жёсткости по коэффициенту постели / пропорциональности: для них не было метода с источником либо поле дублировало другое. Поля остались в модели проекта (открытие старых проектов без потерь), реестр projectFieldFlow.ts обновлён (поле без записи — красный тест); kProp и subgradeZOrigin в запрос не шли и не идут. Проект со сохранённым wall.subgradeK считается по нему, показана красная строка SubgradeKNotice.tsx с кнопкой «Вернуть k_s» (тест subgradeSwitchRemoved.test.tsx).
- **Стороны коррозии.** `wall.corrosionSides` (1 | 2, types.ts) → `corrosion_sides` в обоих путях запроса (projectToRequest.ts, реестр projectFieldFlow: «не задано — умолчание API: 2»); значение API — default 2, ge 1, le 2 (hydroweb/api/main.py, FemCalculateRequest и CalculateRequest; steel_check.check_profile — тоже 2). Не заданное значение в запрос не уходит.
- `CreateProjectModal.tsx` — быстрое создание: одно окно «Название» + «Создать»; путь через ЕГРЗ сохранён необязательной ссылкой, возврат из ЕГРЗ к быстрому созданию сбрасывает подставленные данные. `newProjectTemplate.ts` (createNewProjectData; projectStore.createDefaultProjectData вызывает его) — шаблон нового проекта: стенка 11 м (заделка 6 м), песок 9 м, суммарно 12 м геологии, остальное как в DEFAULT_*. Сами DEFAULT_* в constants.ts НЕ менялись: ими пользуются Подбор решений, досев полей при импорте и открытие старых проектов. Тесты дрейфа: hydroweb/tests/test_new_project_template_fem.py берёт тело запроса из tests/fixtures/new_project_fem_payload.json (сходимость, время < 15 с, Кз > [K], прогиб, Км, отсутствие предупреждений о механизме), а newProjectTemplate.test.ts сверяет данные шаблона с тем же файлом.
- `core/resultsVerdict.ts` + `components/ResultsVerdict.tsx` — плашка «Результат: проходит / у предела / не проходит» по тем же критериям, что core/limitStates.ts (K_металл ⩾ 1,0; Кз ⩾ [Кз] — ориентир по классу сооружения, не норматив; прогиб ⩽ min(H/100; 100 мм), а при заданном допустимом смещении [w] (wall.deflectionLimitMm) — ⩽ [w]; сходимость); для Блюма — η ⩽ 1, длина стенки не меньше рекомендуемой, сходимость. «У предела» — допущение интерфейса: NEAR_LIMIT_FRACTION = 0,9. Критичные предупреждения (regexp не сошл|механизм|исчерпан|нефизич|строгий допуск) остаются на виду, остальные — в свёрнутом «Замечания (N)». Над плашкой — пустой контейнер data-

testid="calc-status" (CalcStatusSlot) под бейдж статуса расчёта. Формулировки новых предупреждений ядра, не подходящие под regexp, попадут в «Замечания»: список в resultsVerdict.ts при необходимости расширяют.

- **Сталь расчёта хранится с результатом ([0569-релиз-6-20-w-марка-ary], долг [0541]).** Расчёт берёт нормативное Ryn (`sigma_yield_MPa`) по выбранной марке, в том числе по марке поставщика, которая со временем может измениться или исчезнуть из каталога; раньше результат этого не помнил, и после правки марки документы печатали марку и Ry текущего выбора проекта. Теперь `FemCalculationResults` несёт `steelGrade` (марка проекта в момент запроса), `sigmaYieldMPa` (Ryn; нет, если запрос его не нёс и сервер принял Ryn по каталогу профиля), `sigmaYieldCatalogMPa` (этот случай: число из ответа сервера) и `profileName` (профиль в момент запроса: пара «профиль + сталь» печатается из одного источника). Записывает их `mapFemResponse(d, n, steel)` в `services/femResults.ts` (вызов в `App.tsx`): Ryn — из ключа ответа `sigma_yield_used_MPa`, у сервера прежней версии — число запроса; у метода Блюма сервер присылает марку, Ryn, Ry и γc сам (`steelGrade`, `RynCheckMPa`, `RyCheckMPa`), там менять нечего. Читатель один — `core/femSteelUsed.ts`: `femSteelUsed(fem, wall)` отдаёт марку, профиль и Ryn результата, а если полей нет (результат посчитан до этой правки) — текущий выбор проекта (тест сверяет с `wallRyn`). Через неё идут: карточка результатов (строка «Марка стали» «S355GP, Ryn = 355 МПа» и отдельная строка «Ry, расчётное»; в строке стоит Ryn, а не Ry, потому что `sigma_yield_MPa` — нормативное сопротивление, расчётное Ry = Ryn/γm), окно «откуда число» (Км и «Ry = Ryn / γm»), проверка металла в предельных состояниях, DOCX отчёта МКЭ (строка «Марка стали», Ryn и Ry таблицы параметров, подстановка в формуле Км), XML и пояснительная записка (п. 3.1, Ryn и Ry, подстановка Км). Ключ запроса `femRequestKey` (`services/resultsFreshness.ts`) включает метку марки: переименование марки при том же Ryn (S355 → S355GP) делает результат устаревшим, потому что результат печатает имя марки, а в запрос уходит только её Ryn. Вне записи остаются разделы, где марка — свойство проекта, а не результата (спецификация, ведомости, ТЭО, чертёж); J и A таблицы параметров и строки «Профиль», W, A карточки на экране читаются по текущему выбору проекта, поэтому при устаревшем результате (он помечен) могут расходиться с парой. Тесты: `core/femSteelUsed.test.tsx` (11), `core/femSteelUsed.server.test.tsx` (6), `App.steelInResult.test.tsx` (настоящий App: после расчёта марку в проекте меняют на S255, строка результата остаётся «S355GP, Ryn = 355 МПа», в запросе sigma_yield_MPa = 355).
- `useNumericField.ts` — formatNumericDisplay: 6 значащих цифр (не меньше 2 знаков после запятой у больших чисел), округляется только показываемая строка, onChange без правки не вызывается. `SoftPromptBanner.tsx` + `softPromptPref.ts` — подсказка сворачивается или закрывается, выбор — в localStorage (ключ hydroweb_soft_prompt, все обращения в try/catch) для всех проектов сразу. `AuthPage.tsx` — форма email видна сразу, провайдеры ниже, VK «в разработке» компактной серой строкой.

Сметный движок: модуль hydroweb/estimate и /api/estimate/*

Сметный расчёт стоимости ограждения — ресурсно-индексным методом: Методика 421/пр (пп. 10в, 46, 50, 93–98; форма ЛСР — Приложение 3), нормы ГЭСН-2022 сборников 05 и 09, цены ФГИС ЦС, НР и СП по приказам 812/пр и 774/пр. Это отдельный модуль, расчётного ядра (hydroweb/core) он не касается. Принцип модуля: ничего не придумывать, не подгонять, не

маскировать — нет цены, нормы или норматива, и позиция остаётся без суммы, попадает в «не посчитано» с причиной, а итог помечается неполным (complete: false).

Показатель	Формула	Пункт
ОТ	Σ ЗТ_i · СЦ_ЗТ(тек) · V_i (код нормы 1-100-NN)	п. 79а, 93, ф. (1)
ОТм	Σ ЗТ_k · СЦ_k · V, ЗТ_k = T_k · LabourMach_k (ФСБЦ_Маш), цена по DriverCode (4-100-NNN)	п. 80, 93, ф. (1.2)
ЭМ	Σ T_m · СЦэм(тек), цена машины без оплаты труда машиниста	п. 81, 94, ф. (2)
М	Σ P_j · См(тек), P = P_it · V	п. 85, 86, 96, ф. (4), (5)
Цена ресурса	цена текущего уровня ФГИС ЦС, иначе цена 01.01.2022 × индекс группы; иначе вводит пользователь (конъюнктурный анализ)	п. 95–98, 90
ФОТ, НР, СП	ФОТ = ОТ + ОТм; НР = ФОТ·Н_НР/100 (812/пр); СП = ФОТ·Н_СП/100 (774/пр); итог по позиции = прямые затраты + НР + СП	812/пр п. 18–22; 774/пр п. 11–14

- **Книга норм и данные.** gesn.py читает ГЭСН.xml выбранных сборников (по умолчанию 05 и 09), машины и затраты труда машиниста на 1 маш.-ч из ФСБЦ_Маш.xml и технологические группы; компактный дамп — hydroweb/data/fgis/gesn2022_20260709.json.gz: 1 745 норм, 316 машин, 519 технологических групп. Версия данных (имя базы, доп. № ГЭСН, приказ об утверждении ФСБЦ, sha256 архива) фиксируется в NormBook.meta и попадает в каждый ответ отдельно от версии цен: версии норм и цен не обязаны совпадать.
- **НР и СП.** nr_sp.py + data/estimate/nr_sp_812_774.json: шифр вида работ берётся из самой нормы (Пр/812-005.1, Пр/774-005.1), 136 строк таблиц, 134 подтверждены текстом обоих приказов (104.1 и 104.2 — раздел «Прочие работы», в сборниках 05 и 09 не встречаются); цитата строки возвращается в выходе. Если у нормы два вида работ, норматив не выбирается — позиция без суммы. Графа «Территория / МПРКС / РКС» не привязана к ценовым зонам (источник такого соответствия не найден): параметр options.territory, по умолчанию «Территория» — допущение.
- **Двойной счёт металла.** Норма погружения (05-01-011, 05-01-012, 05-01-013, 05-01-094) уже содержит металл 1,01 т на 1 т (абстрактный ресурс 08.3.10.02, технологическая группа 15.12.011). Позиция-ресурс с кодом из этой группы, совпадающая с ресурсом нормы, из семейств стального шпунта (08.3.10.02-, 08.3.10.04-, 59.1.08.03-) или с флагом is_sheet_pile получает статус rejected_double_count и в итог не входит. Современные U/Z-профили в технологическую группу не входят: цена задаётся как цена обобщённого ресурса нормы (price_overrides, основание conjuncture_analysis), без неё позиция остаётся без суммы.
- **Распорки, пояса, анкеры: подбор нормы по условиям.** support_selection.py (select_support_norm, validate_selection); параметр selection позиции с kind anchor, strut или belt в POST /calc либо GET /support-select с теми же параметрами. Без norm норма подбирается по условиям; вместе с norm — должна совпасть с подобранной, иначе 422; параметры не того вида — 422; чего недостаточно — status: not_selected (200) с причиной. Результат (norm_selection позиции) несёт conditions[] (имя, значение, ok: true / false / null — «не проверено») и reservations[]; оговорки дублируются структурными warnings (support_norm_scope, support_norm_reservations). Анкеры: ГЭСН 05-03-007 (стержневой инъекционный АИ, предел «до 42 тс») и 05-03-008 (четырёхпрядевой, «до 61,7 тс»), суффикс -01…-04 = группа грунта 1–4; тип анкера и группу грунта задаёт пользователь (в проекте

типа нет, определений групп грунта в открытых данных нет); предел читается из наименования таблицы нормы (1 тс = 9,80665 кН), выше предела норма не подбирается (422 с точным пределом: 42 тс = 411,8793 кН, 61,7 тс = 605,070305 кН); явная norm из 05-03-007/008 без selection и без force_kn возвращает предупреждение support_force_limit_unchecked (предел не проверен); с force_kn позиции (NormPosition.force_kn в calc.py, поле запроса в api/routers/estimate.py) предел проверяется в support_selection.explicit_norm_force_warnings: усилие выше предела — предупреждение support_force_limit_exceeded (positions[], source — таблица ГЭСН), норма применяется по выбору пользователя, расчёт не блокируется; force_kn вместе с selection, у позиции не-анкера или у нормы вне 05-03-007/008 — 422 (validate_explicit_force); клиент: buildEstimate.manualAnchorForceKn выбирает поле по таблице нормы (05-03-007 — расчётное сопротивление тяги Rt;d, tendonCapacityKN; 05-03-008 — напрягаемое усилие, prestressForce), normPositionRequest кладёт force_kn только при normSource «manual», тип NormPositionIn.force_kn — в services/estimateTypes.ts; бурение, установка и натяжение на проектное усилие уже в составе норм (отдельных норм нет — двойного счёта нет); единица «м анкера» трактуется как полная длина по оси (допущение), наклон нормой не учитывается, характеристики буровой машины нормы (ФСБЦ_Маш) сверяются оговоркой. Распорки и пояса из проката: 05-01-014-02 (т) при selection.material=steel — с оговорками: наименование таблицы ограничено котлованами под опоры мостов (scope_restricted), суффикс «стального» относится к шпунтовому ряду (условие «Материал» ok: null, допущение support_norm_steel_suffix), единица «т» и масса без запаса — допущение (support_norm_unit_t). Нормы вне подобранных сборников — сб. 29 «Тоннели и метрополитены» (29-02-002, 29-02-006, 29-02-011, 29-02-012) и 39-01-021 — возвращаются как outside_scope_candidates; в книгу норм (сборники 05 и 09) они не загружены, подбор по ним — вручную. Не подобрано: контрольные испытания анкеров (нормы нет), анкеры выше 61,7 тс, монтаж и демонтаж распорок по сб. 09. У ресурсов с расходом «П» норма расход не задаёт — его и цену вводит пользователь. Позиция без norm и без selection получает статус norm_not_selected и список candidates.

- **Запрос POST /api/estimate/calc.** zone_id; positions[] (до 100; поле type обязательно): type=norm (id, kind, norm — шифр вида 05-01-012-10, volume в единицах нормы, abstract_choices, price_overrides, quantity_overrides) или type=resource (code, unit, quantity, price, is_sheet_pile); options (territory, rounding line|none, unit_price_rounding); positions[].selection (подбор нормы опор); ssr (chapter 1–12, object_class, contingency_percent ⩽ 10, vat_rate_percent, temporary_buildings и winter — {percent 0…100, basis}: ВЗиС и зимнее удорожание, ниже). Числа ограничены Field(gt/ge, le), строки — длиной и шаблоном, id позиций уникальны, лимит 30 запросов в минуту на адрес. Ошибки — только авторские сообщения без str(e): 404 (зона), 422 (вход), 503 (нет снимка или книги норм), 500 — общее сообщение, подробности в логе.
- **Ответ.** source и attribution (ФГИС ЦС, обладатель информации, период, версии; ПП РФ № 959, п. 10), method, prices и norms (версии отдельно), zone, options, lsr {positions[], totals, complete, not_calculated[], notes}, osr, ssr (по параметрам; НДС считается только при заданной ставке, иначе vat_status «не рассчитан»), complete, not_accounted[] и assumptions[] — в КАЖДОМ ответе, чтобы итог не принимали за полную смету. Статусы позиции: calculated, incomplete, norm_not_found, norm_not_selected, rejected_double_count, invalid_input.

- **Допущения и «не учтено».** Допущения (возвращаются в assumptions): округление «половина вверх» до копеек, графа НР, цена машины без оплаты труда машиниста, код «2» нормы как суммарные затраты труда машинистов (подтверждено на всех 1 745 нормах: Σ T_m · LabourMach совпадает в пределах 0,005), цена «база × индекс» не округляется, цена пользователя — сметная цена текущего уровня без НДС с транспортом и заготовительно-складскими расходами, ставка НДС, дата вступления в силу Методики 181/пр. Не учтено (not_accounted): ВЗиС и зимнее удорожание — только по параметрам ssr.temporary_buildings и ssr.winter, процент по умолчанию не подставляется, прочие затраты главы 9, служба заказчика, проектные работы, оборачиваемость шпунта (п. 67–69, 71), перевозка сверх учтённой сметными ценами, коэффициенты на условия производства (Приложение 10), оборудование, несерийные машины, индексы п. 11 а–в.
- **Приёмка.** test_estimate_acceptance.py: независимая реализация на Decimal читает JSON-фикстуры напрямую (код движка не использует) и совпадает с движком до копейки на всех 51 норме фикстуры, считаемых без пользовательского ввода, в обеих зонах; иллюстративный расчёт 05-01-012-10 для Москвы — 133 203,47 ₽ за 1 т. Для Свердловской области (фикстура) движок даёт 101 543,48 против 101 543,45 в исходном отчёте: расхождение 0,03 вызвано построчным округлением графы 12 (п. 42, Приложение 3, прим. 13), варианты округления зафиксированы в тестах. Подробности и риски — документы/разработка/проектные_записки/сметный_движок_ресурсно_индексный.md.
- **ВЗиС (глава 8) и зимнее удорожание (глава 9).** summary.build_ssr; параметры ssr.temporary_buildings и ssr.winter — {percent, basis}: основание обязательно (источник значения: норматив, пункт, таблица), без процента основание не принимается, без ssr.chapter — 422. Формулы (тысячи рублей, два знака, каскад по п. 42): ВЗиС = П_взис · ЛСР / 100; Зимн = П_ндз · (ЛСР + ВЗиС) / 100; итог глав = ЛСР + ВЗиС + Зимн, резерв (п. 179) и НДС (п. 181) считаются от итога глав. Принимаются только при главе ЛСР 1–7 (иначе 422 с пояснением): по открытым копиям Методики 332/пр п. 23 база ВЗиС — итоги СМР и монтажных работ глав 1–7 ССР, по п. 18 Методики 325/пр зимнее считается от итогов СМР объектных смет глав 1–7; для резерва и НДС главы 1–12 допустимы. Откуда процент: ВЗиС — Методика 332/пр, п. 21 лишь выбирает метод (нормативный или расчётный), процент при нормативном методе — Приложение N 1 (N 2 — капремонт) по п. 23–25; текст 332/пр в источниках платформы не загружен, пункты даны по открытым копиям (редакция копии не указана — нумерацию сверить с официальным текстом). Зимнее — 325/пр (копия ФГИС ЦС с изменениями 170/пр от 21.03.2025 и 192/пр от 30.03.2026): НДЗ — проценты от сметной стоимости по таблице Приложения N 1 (вид объекта × температурная зона), коэффициент графы 6 Приложения N 4 по п. 13. Допущения помечены: база ВЗиС «итог ЛСР выбранной главы» (assumptions: vzis_winter_base; по 332/пр база — итоги глав 1–7 в целом) и база зимнего «ЛСР + ВЗиС» — прочтение п. 18 автором, не буквальный текст. Подсказки (hints[] ответа /calc и GET /ssr-hints; у каждой пункт или таблица и дословная цитата ⩽ 15 слов, все цитаты проверены тестом по текстам приказов); температурная зона III заведена только для 4 зон снимка из 147 (Москва 191, Московская область 127, Санкт-Петербург 206, Ленинградская область 162), для остальных 143 подсказка пишет «данные не заведены — определите по таблице». Ответ: строки глав 8 и 9 в ssr.chapters[] (kind, percent, base_thousand_rub, basis, refs), ssr.temporary_buildings, ssr.winter. XLSX, лист «ОСР и ССР»: строки глав 8 и 9 с процентом, базой, основанием пользователя и пунктами приказов. Не учтено: прочие затраты главы 9, зимние дороги и очистка от снега (325/пр п. 26),

перебазировка машин, коэффициенты ветра, расчётный метод ВЗиС. Тесты: test_estimate_support_selection (74), test_estimate_ssr_extras (51): регрессия ЛСР 05-01-012-10, 1 т, при заданных ВЗиС и зимнем не меняется.

Снимок цен ФГИС ЦС и книга норм: формат и пересборка

Снимок — hydroweb/data/fgis/snapshot_2026Q3.json.gz (2 634 524 байт, 2,51 МБ; в разжатом виде JSON 31,2 МБ; 147 зон × 5 212 кодов = 766 164 записи цен), собирается scripts/fgis/build_snapshot.py детерминированно: ключи отсортированы, gzip без метки времени, одинаковые входы дают побайтно одинаковый файл. Формат (kind — material, machine, labor или transport):

```
{"meta":  {"period_label", "period_id", "fetched_at", "fsnb_version",
           "ksr_version", "attribution", "coverage": {"complete": false},
           "zones_not_included", "files": [...]},
 "codes": {"<код>": {"name", "unit", "kind", "opt_2022", "smeta_2022",
                     "group_no", "group_name"}},
 "zones": {"<zone_id>": {"subject_id", "name", "index_letter",
                         "base_price_act", "labor_pay_act",
                         "prices": {"<код>": {"p": <цена>|null,
                                              "i": <индекс>|null}}}}}
```

- **Покрытие — 147 из 148 ценовых зон ФГИС ЦС (все существующие, кроме одной):** 3 квартал 2026 (период 427). В снимке 147 зон 86 субъектов: полностью (все зоны субъекта) — 85 субъектов, частично — Ямало-Ненецкий АО, 4 зоны из 5 (241, 242, 243, 244). Не вошла зона 245 (ЯНАО, 5 зона): во ФГИС ЦС для неё нет периода 427 (последний — 425, 1 квартал 2026; ShowExportSplitForm отвечает show:false), чужой период не подставлялся; смета по ней невозможна — запрос отклоняется (404 для зоны вне снимка), а не подменяется ценами другой зоны. meta.coverage.complete = false (с 07.10.2026 вычисляется в build_snapshot.py: истинно только при переданном индексе всех зон и отсутствии недостающих), not_downloaded_zone_ids = [245], перечень — meta.zones_not_included (зона, название, причина) и zones_index. Остальные 120 зон скачаны 07.10.2026 scripts/fgis/download_splitforms.py тем же способом, что первые 27 (публичный GET открытого API, без входа и капчи; последовательно, пауза не меньше 2,5 с); размер и sha256 всех 147 файлов равны манифесту. Исключённых зон среди скачанных нет: базовые цены 2022 совпали у всех 147 по всем 5 212 кодам, период у всех «3 квартал 2026». Ограничение применения «за счёт средств федерального бюджета» найдено только в примечаниях зоны 191 (estimate/zone_notes_2026Q3.json; запись обязательна для каждой зоны снимка); у остальных 146 такого текста нет (проверено по примечаниям всех 147 файлов); движок не решает, подпадает ли объект под условие, и показывает примечание в ответе (zone.applicability_notes и предупреждение zone_applicability). Контроль по зонам (иллюстративный расчёт 05-01-012-10, 1 т, шпунт 08.3.10.02-0011): Москва 133 203,47 ₽, Московская область 128 454,56 ₽, Санкт-Петербург 129 096,23 ₽, Ленинградская область 129 776,64 ₽ — не менялись при переходе с 27 зон на 147 (до копейки); для всех 147 зон POST /calc отвечает 200, на контрольной позиции итог полный у 145, у зон 251 (Иркутская область, 6 зона) и 261 (Красноярский край, 6 зона) complete = false: в данных ФГИС ЦС у двух материалов нет ни цены, ни индекса (зона 251 — 01.3.02.03-0001 ацетилен и 01.3.02.08-0001 кислород; зона 261 — 08.1.02.11-0001 поковки и 11.1.03.01-0067 брус обрезной; в исходных листах в колонках 8 и 9 стоит «-» и «-»), итог 0,00, движок перечисляет коды в unresolved, цену по конъюнктурному анализу вводит пользователь (пп.

90 и 98 — по записке о снимке цен) — это свойство источника, не снимка; состав фиксирует тест test_control_position_is_priced_in_every_snapshot_zone. Графа НР «Территория» по зонам в источниках не найдена ни для одной из 147 — допущение движка по умолчанию; для отдалённых зон индексы цен заметно выше (шпунт 08.3.10.02-0011: 1,22…2,06 против 1,26 в Москве), проверка графы пользователем там особенно важна. Область снимка (data/fgis/scope.json): ресурсы всех норм сборников 05 и 09, кандидаты абстрактных ресурсов по технологическим группам, оплата труда машинистов по коду водителя машин; в области 5 247 кодов, строки есть у 5 212 (68 труд, 316 машин, 4 828 материалов); 35 кодов без цены ни в одной зоне — движок обязан трактовать их как «нет цены». Индексы вне [1; 3] (137 834 записи в 139 группах: 130 352 меньше 1, 7 482 больше 3; максимум 34,79 у группы № 30 в зоне 261, минимум 0,28 у группы № 595 в зоне 129; индексы выше 3 есть в 76 зонах из 147) — предупреждение сборки, движок их не отвергает. Подробности — документы/разработка/проектные_записки/снимок_цен_фгис_цс_3кв_2026.md. Проверка 29.10.2026 ([0542-сметы-фгис-2026-08-12]): паспорта наборов ФСНБ-2022, КСР и технологических групп в открытых данных ФГИС ЦС байт в байт равны копиям от 02.10; последние версии наборов в списках версий — те, что в снимке (ФСНБ-2022 — data-20260709, КСР — data-20260812, технологические группы — data-20260515); цен 4 квартала 2026 у зоны 191 нет. Цены сметы не менялись: «12.08.2026» — это версия КСР и дата акта базовых цен 527/пр (оба уже в снимке) и отметка в паспорте ФСНБ-2022, за которой файла данных нет. Пересборка build_norm_book.py и build_snapshot.py на тех же входах даёт файлы, побайтно равные закоммиченным.

- **Проверки сборки** (нарушение — ошибка, файл не создаётся, код выхода 2): базовые цены 2022 одинаковы во всех зонах; у строки не более одного из p и i; период один и совпадает с периодом API; sha256 файлов совпадает с манифестом; раскладка листа проверяется по тексту (строка заголовка «Код ресурса…» ищется, а не берётся по номеру: у Москвы она 19-я, у Свердловской области 17-я, по 147 зонам — с 16-й по 19-ю; примечания ищутся между реквизитами и этой строкой). Предупреждения в отчёт: индексы вне [1; 3] (в Москве индексы лежат в диапазоне 0,53…2,33, в Московской области 0,52…1,96, предупреждение дают значения ниже 1; в дальних зонах бывают до 34,79, это не ошибка), расхождения имени и единицы между зонами, коды области без строки, коды материалов вне КСР.
- **Порядок цены ресурса** (pricebook.py): p из ФГИС ЦС (графа 8) → иначе smeta_2022 × i (графы 5 и 9, не округляется) → иначе цены нет, и цену вводит пользователь. Одна известная аномалия источника: у кода 14.4.01.01-0003 в листе единица «кг», в ФСБЦ «т» — для расчёта берётся единица и цена листа.

```
# сплит-формы зон: последовательно, пауза >= 2,5 с, останов при 403/429
python3 scripts/fgis/download_splitforms.py \
    --out рабочие_материалы/фгис_цс/splitform_all \
    --subjects рабочие_материалы/фгис_цс/meta_api/_subjects.json
# снимок цен (действующий снимок собран с --zones all — все скачанные зоны, 147;
можно список --zones 191,127,115)
python scripts/fgis/build_snapshot.py \
    --splitforms рабочие_материалы/фгис_цс/splitform_all/427 \
    --manifest рабочие_материалы/фгис_цс/splitform_all/manifest_427.json \
    --gesn-zip <data-...-structure-....zip> \
    --techgroups <techgroups_....xml> --ksr-csv <ksr_....csv> \
    --zones all --zones-index
рабочие_материалы/фгис_цс/splitform_all/zones_index.json
# примечания об ограничениях применения по зонам (запись для каждой зоны снимка
```

```
обязательна)
python scripts/estimate/build_zone_notes.py --period-id 427 --period-tag 2026Q3 \
    --dir рабочие_материалы/фгис_цс/splitform_all/427 --out
hydroweb/data/estimate/zone_notes_2026Q3.json
# книга норм ГЭСН-2022 (старый gesn2022_*.json.gz удалить)
python scripts/estimate/build_norm_book.py \
    --zip <data-...-structure-....zip> --techgroups <techgroups_....xml> \
    --out hydroweb/data/fgis/gesn2022_<дата>.json.gz
# нормативы НР/СП — только при новой привязке или новых приказах
python scripts/estimate/build_nr_sp_table.py --xlsx ... --t812 ... \
    --t774 ... --out hydroweb/data/estimate/nr_sp_812_774.json
# тесты сметы и снимка
pytest hydroweb/tests/test_estimate_*.py \
    hydroweb/tests/test_fgis_snapshot.py \
    hydroweb/tests/test_data_integrity.py -q
```

Что учесть при обновлении данных

Источник данных указывается всегда (ПП РФ № 959, п. 10: ссылка на ФГИС ЦС и обладателей информации); значения не изменяются. Сырые файлы (xlsx сплит-форм, архив ФСНБ-2022, КСР) в git не лежат — путь к ним задаёт HYDROWEB_FGIS_RAW (по умолчанию рабочие_материалы/фгис_цс); тесты на реальных файлах пропускаются, если их нет.

Эталонные числа приёмки относятся к фикстуре 3 квартала 2026 (scripts/estimate/build_test_fixture.py) и меняются вместе с ней; тест «движок = независимая реализация» остаётся верным всегда.

Снимок читается целиком из JSON (SnapshotPriceBook.load): при росте до десятков мегабайт его стоит разделить по зонам — интерфейс PriceBook это допускает без изменения движка. Фактический размер на 147 зонах — 2,51 МБ (на 27 зонах было 0,57 МБ; JSON в разжатом виде 31,2 МБ). Узкое место — память, а не размер файла: загруженный снимок добавляет к процессу ≈ 315 МБ (на 27 зонах — ≈ 64 МБ; измерено по ps rss), загрузка ленивая, 0,4 с; воркеров по умолчанию 2, то есть до +0,6 ГБ памяти контейнера — перед выкаткой проверить запас памяти сервера либо хранить цены по зонам.

Привязка шпунтового каталога к КСР

scripts/catalog/build_ksr_binding.py сопоставляет каждый профиль каталога (sheetPiles.generated.ts, вывод build_sheet_pile_catalog.py) с позицией классификатора строительных ресурсов ФГИС ЦС (ksr_data-20260812-structure-20221117.csv, открытые данные). Правило явного основания: exact — тот же тип профиля (U или Z) и совпадение четырёх размеров (b, h, t_web, t_flange, допуск 0,05 мм) с наименованием позиции КСР; code_successor — код с тем же названием в ФССЦ-2020, присутствующий в КСР; type_conflict — размеры совпали, тип расходится (кода нет); none — иначе. Масса 1 м в КСР отсутствует и основанием не служит. Результат — 29 записей: 17 exact, 1 code_successor, 11 none (18 профилей с кодом); у остальных кода нет, «по похожести» он не подбирается.

- **Файлы:** `frontend/src/data/ksrBinding.json` — источник правды (читает интерфейс через modules/catalog/core/ksrBinding.ts: ksrBindingByProfileId, ksrBindingByCatalogId, ksrCodeOf — код возвращается только при exact и code_successor); `hydroweb/data/ksr_binding.json` — побайтная копия для бэкенда; `hydroweb/data/ksr_excerpt.json` — выписка из КСР по привязанным кодам (код, наименование, единица), чтобы проверка не требовала 28-мегабайтного файла.
- **Использование.** Карточка каталога (SheetPileCard.ksrCode), строка «Код КСР (ФГИС ЦС)» в справочных свойствах сечения на вкладке «Ограждение» и смета: buildEstimate подставляет код профиля как выбор цены металла только если он входит в

технологическую группу нормы погружения; иначе сообщает, что цены во ФГИС ЦС нет, и требует цену пользователя. Тип профилей ECO 616/618/622T/627/632 в hydroweb/data/eco_profiles.json исправлен на U (по подтверждению владельца; в каталоге производителя и KCP они U-образные 600 мм) — тип используется только в печатной форме и в расчёте износа, числа расчётов не меняются.

- **Тесты:** test_ksr_binding.py (16; размеры разбираются из наименований выписки независимо от скрипта сборки; полная перепроверка по KCP-файлу — если задан HYDROWEB_KSR_CSV) и ksrBinding.test.ts (10), RetainingWallTab.ksr.test.tsx (3).

Площадь сечения A каталога шпунта: на 1 м стенки ([0325-каталог-A])

- **Смысл поля.** W, I и A каталога интерфейса (constants.ts: ECO_PROFILES = data/sheetPileProfiles.generated.ts, производный от hydroweb/data/eco_profiles.json скриптом scripts/catalog/build_ui_profiles.py; 41 профиль из 63 записей каталога ядра, без A марки в интерфейс не выводятся) заданы на 1 м стенки: см³/м, см⁴/м, см²/м. A ≈ g / 0,785, где g — масса 1 м² стенки (сталь 7850 кг/м³: 1 см² на 1 м = 0,785 кг). Поле уходит в запрос как section_A (wall.sectionA; осевая жёсткость EA и собственный вес при axial_model ≠ none), печатается в DOCX («Площадь сечения A», generateDocx.ts) и XLSX («A», generateXlsx.ts), показано на вкладке «Ограждение».
- **Что было неверно.** У Л4, Л5, Л7 (экранный каталог) и у Л4, Л5, Л7, «Л5 УМ» (второй каталог — modules/catalog/data/sheetPiles.generated.ts, на экране пока не используется) в A стояла площадь одной сваи: 94,26 / 127,4 / 183,8 / 145,0 см². Причина — в файле-источнике: колонка озаглавлена «Площадь сечения 1 п.м. стены» (см²/м), но у серии Л в ней площадь сваи. Диагноз сделан по трём признакам, не по названию профиля: (1) A на 1 м ≈ масса 1 м² / 0,785 и ≈ масса сваи / 0,785 / b (у 26 из 29 профилей оба признака в допуске 5 %, у Л4, Л5, Л7 — минус 57,5…60,0 %, ровно множитель b); (2) радиус инерции √(I/A) не может превышать половину высоты профиля (со старой A у Л4 / Л5 / Л7 он был 1,14 / 1,16 / 1,06 от H/2); (3) внешняя точка: в отчёте эталонной программы для Л5 напечатано «Площадь сечения A (1 п.м.), мм² = 30333» (303,33 см²/м) при тех же J = 50943 и W = 2962. Исправлено: Л4 235,65, Л5 303,33, Л7 432,47 (экранный каталог), «Л5 УМ» 286,56 (второй каталог; в экранном каталоге у Л5 УМ значение 286,8 не менялось — оно в допуске). Таблицы внешних источников с характеристиками Л4 / Л7 не найдено: источник правки — вкладка каталога, подтверждённая отчётом (Л5) и признаками массы и геометрии.
- **Генератор.** `scripts/catalog/build_sheet_pile_catalog.py`: функция area_per_metre() принимает A строки как «на 1 м» или «на сваю» (A / b) по признакам массы с допуском 5 %; если не сходится ни так, ни так или данных нет — ValueError, молчаливых правок и перечня профилей в коде нет. Баннеры сгенерированных файлов перечисляют пересчитанные строки; повторный запуск файлы не меняет.
- **Что не менялось и почему.** `hydroweb/data/eco_profiles.json` (бэкенд) — A там на сваю, и у четырёх записей есть собственные расхождения: Л4 / Л5 / Л7 — 37,7 / 53,5 / 78,1 (площадь сваи, ошибочно умноженная на b), L5-UM — 290,1 при b = 500 (значение на 1 м, не на сваю). Абсолютное значение этой A в расчёт не идёт (corrosion.reduce_section масштабирует его отношением; /api/profiles его не отдаёт), поэтому по слову владельца исправлялось только поле интерфейса; расхождения закреплены списком KNOWN_A_DEVIATIONS в test_sheet_pile_area.py (тест падает и при новом расхождении, и

при исправлении одного из известных). В docstring SheetPileProfile.A (core/models.py) смысл записан как «на одну сваю».

- **Сохранённые проекты.** wall.sectionA хранится по значению на момент выбора профиля: в проекте с Л4 / Л5 / Л7 остаётся прежнее число, пока профиль не выбран заново (список «Профиль ECO», автоподбор) или не нажата ссылка «Заполнить E/J/A из профиля …» (она подставляет J и A; E не меняет). Миграции содержимого проектов нет — решение за владельцем; в DOCX / XLSX и в тексте проекта для ассистента (App.tsx) такой проект покажет старое число.
- **Влияние на расчёт (запросы к /api/calculate/fem; код бэкенда не менялся).** При axial_model = none (по умолчанию; в интерфейсе модель продольной силы не задаётся, projectFieldFlow) ответ побитно одинаков (без elapsed_s) при старом и новом A — на запросе проекта 2139, на запросе 2130.3 (Л5) и на нём же с сечениями Л7 и Л4 (J / W / A подставлены, синтетика). При axial_model ≠ none M, Q, w и Кз прежние, меняются продольная сила N и (режим full с вертикальными составляющими связей и нагрузок) Км, потому что σ_N = N / A: на синтетическом запросе с анкером 30° и вертикальной нагрузкой Л5 Км 8,70 → 9,59, Л7 12,57 → 13,89, Л4 6,77 → 7,53 — старая A занижала Км в режиме full на 9…10 %.
- **Тесты:** frontend/src/data/sheetPileArea.test.ts (7: A интерфейса по g и по G·b, модуль каталога, геометрия, расхождение копий, точные значения Л4 / Л5 / Л7) и hydroweb/tests/test_sheet_pile_area.py (13: генератор — пересчёт и отказ при противоречии; закреплённые расхождения бэкенда; фронт ↔ бэкенд; сквозной прогон /api/calculate/fem).

Сортамент труб для распорок и стоек РРСК

Сортамент стальных труб генерируется scripts/catalog/build_pipe_catalog.py из сырых таблиц стандартов (scripts/catalog/pipe_sources/): таблица 1 ГОСТ 10704-91 в редакции Изменения № 3 (1293 типоразмера, Ø10–1420 мм, толщина 0,8–50 мм; колонка 50 мм есть только у Ø1020, Ø1220, Ø1420) и таблица 1 ГОСТ 8732-2025 (1793 типоразмера, Ø20–550 мм, толщина 2,5–75 мм; ГОСТ 8732-78 заменён с 01.03.2026). Источник и проверки — документы/разработка/сортамент_труб_для_распорок_2026-10-07.md.

- **Что берётся из стандарта, что считается.** Из таблицы — масса 1 м; A, I, W, i считаются по наружному диаметру D и толщине t (кольцо, d = D − 2t; формулы те же, что в core/section_properties.ring_section, расхождение каталога с ней не более 0,02 %, проверяет тест). Допуски на диаметр и толщину стенки, поправка массы для труб по ГОСТ 10706 (+1 % и +1,5 % по прим. 2 к таблице 1 ГОСТ 10704-91), марка стали и условия поставки в каталог не вносились. Каждая масса сверяется с формулой стандарта: для ГОСТ 10704-91 — M = 0,0246615·(D − S)·S с допуском в единицу последнего печатного разряда (ячейка вне допуска без пометки в сырых данных останавливает генератор; единственное исключение — Ø920×14: напечатано 312,78, по формуле 312,81, −0,009 %, в редакции 2015 г. было 312,81; принята как напечатано и помечена «marginal»); для ГОСТ 8732-2025 — M = ρ·π·S·(D − S)/1000, ρ = 7,85 г/см³, все 1793 числа в пределах шага округления 0,01. Масса таблицы отличается от геометрической (по A) не более чем на 0,451 %, при расхождении более 1 % генератор останавливается. Таблица ГОСТ 10704-91 разобрана из текстового слоя PDF (1290 ячеек) и дополнена тремя ячейками колонки 50 мм (Ø1020, Ø1220, Ø1420), которую обрезал край страницы в копии КонсультантПлюс: значения перечитаны по скану

Изменения № 3 и сверены с формулой стандарта; второго независимого полного текста редакции 2021 г. не найдено; таблица ГОСТ 8732-2025 — скан, прочитана по изображениям страниц; второе, независимое чтение дал частично распознанный текстовый слой (1428 из 1553 ячеек на страницах 7–13 PDF, 92 %): все распознанные числа совпали с расшифровкой, на последней странице таблицы (D 325–550, t 32–75) текстовый слой непригоден и там сверка идёт только по формуле стандарта. Остаточный риск: формула не ловит ошибку «прочерк ↔ число» (какие размеры существуют).

- **Файлы.** `hydroweb/data/pipe_catalog.json` (бэкенд: core/pile_catalog.py → GET /api/rrsk/sections; эндпоинт отдаёт 1293 трубы ГОСТ 10704-91 вместо прежних 20 — ответ вместе с двутаврами вырос с ≈ 8 до ≈ 278 КБ; параметр standard есть у pile_catalog.pipe_catalog(), но в эндпоинт не выведен, поэтому труб ГОСТ 8732-2025 у стоек нет; двутавры остались в hydroweb/data/pile_sections.json), `frontend/src/data/pipeCatalog.generated.ts` (интерфейс: data/strutCatalog.ts → PIPE_CATALOG, PIPE_STANDARDS; модуль в основном бандле, ≈ 225 КБ исходника, ≈ 71 КБ в gzip, отложенная загрузка не делалась; идентификаторы карточек каталога modules/catalog/core/strutSectionCatalog.ts у труб ГОСТ 10704-91 прежние, у остальных стандартов — с суффиксом стандарта, например «@8732-2025»). Руками не править: генератор с ключом --check проверяет, что файлы равны результату генерации (тест test_pipe_catalog.py).
- **Выбор в интерфейсе.** components/PipeSectionPicker.tsx — «стандарт → наружный диаметр → толщина» (три связанных выпадающих списка; список «Стандарт» показывается, только если стандартов больше одного: в окне распорки их два, у стоек РРСК — один); чистые функции — core/pipeSelect.ts (MAIN_DIAMETER_FROM_MM = 159: диаметры меньше — отдельная группа «Малые диаметры»). При смене диаметра остаётся та же толщина, если она есть; при смене стандарта — тот же диаметр и толщина. Сохранённая распорка хранит стандарт и обозначение («Д273,0 х 8,0»): ключ поиска — пара «стандарт + обозначение» (findPipe); пустой или неизвестный стандарт даёт первую строку с таким обозначением — ГОСТ 10704-91, как в прежнем каталоге; обозначения прежних 20 труб ГОСТ 10704-91 не менялись. Сечение, которого нет в каталоге (ручной ввод, «Труба 325×8»), открывается со своими сохранёнными характеристиками и подсказкой, «ОК» не блокируется (раньше блокировалась). Числа в сохранённой распорке не меняются, пока в окне не нажата «ОК»: проверки распорки (calc/strutCheck.ts, core/supportChecks.ts) работают по снимку в проекте, а не по каталогу.
- **Прежние значения.** В старой таблице из 20 труб момент инерции и момент сопротивления у труб Ø377 и крупнее были занижены до 2,2 % (наибольшее — Д530×10: I −2,24 %, W −2,23 %), радиус инерции — до 1,1 % (Д630×10); закономерность по диаметру, а не опечатки (таблицы ГОСТ A, I, W не содержат, так что это были не «табличные» значения, как объясняло прежнее описание теста); в новом каталоге это точная геометрия, допуск теста test_pile_catalog.py вместо 3 % — 0,1 %. Площадь и масса совпадали (A до 0,19 %; масса у Д426×8 и Д530×8 отличалась на 0,06 и 0,04 %). При повторном «ОК» у старой распорки I и W вырастут у 17 труб из 20 и немного уменьшатся у Д159,0 х 4,5, Д159,0 х 6,0 и Д219,0 х 6,0 (I на 0,19–0,45 %, W на 0,31–0,64 %). Влияние на окно «Расчёт распорки» (СП 16.13330.2017; сравнение старых 20 строк и нового каталога прогоном runPipeStrutCheck; измерено 07.10.2026 на прежней таблице Ф — до перехода на таблицу Д.1 [0340-ф-

распорки-0708], с таблицей Д.1 не повторялось): в четырёх примерах (Д325×8, Д426×10, Д530×10, Д219×8) вердикт и число не пройденных проверок не изменились, наибольшая загрузка менялась на 0,2–1,8 п. п. и везде уменьшалась; на сетке из 1080 сочетаний (20 прежних труб × шесть пролётов L = 6…20 м × девять значений N = 10…90 % от предельной силы) общий вердикт не изменился ни в одном сочетании, а загрузка росла лишь у Д159,0 х 4,5, Д159,0 х 6,0 и Д219,0 х 6,0 (не более чем на 0,2 п. п. при загрузке до 160 %).

- **Не включено.** ГОСТ 8734-75 (холоднодеформированные, точные трубы машиностроения), ГОСТ 10705-80 и ГОСТ 10706-76 (технические условия, размеры — по 10704), ГОСТ 20295-85 (магистральные трубопроводы), ГОСТ 8696-74 (электросварные спиральношовные общего назначения: для распорок применим, но таблица не получена в надёжном виде — включить можно, когда она будет получена). Диапазоны Ø5–250 мм (8734) и Ø159–2520 мм (8696) взяты из сводок страниц-источников, полные тексты этих стандартов не читались. Таблица ГОСТ 8732-2025 с прежним ГОСТ 8732-78 не сравнивалась (независимых полных текстов 1978 г. не получено).
- **Тесты:** hydroweb/tests/test_pipe_catalog.py (15), test_pile_catalog.py (5); frontend: data/pipeCatalog.test.ts, core/pipeSelect.test.ts, components/PipeSectionPicker.test.tsx, StrutDetailDialog.test.tsx, RetainingWallTab.pipePicker.test.tsx.

Сортамент двутавров и швеллеров для распорок, поясов и стоек РРСК

Сортамент генерируется scripts/catalog/build_beam_catalog.py из выписок таблиц стандартов (scripts/catalog/beam_sources/*.csv, протоколы извлечения и проверок рядом): ГОСТ 35087-2024 (357 типоразмеров), ГОСТ 8239-89 (17), ГОСТ 26020-83 (90 из 91 строки выписки), СТО АСЧМ 20-93 (73 из 76) — двутавры; ГОСТ 8240-97 (77 из 78) — швеллеры; всего 614. Строки, в которых напечатанное в копиях стандарта несовместимо с остальными числами в разы (явная опечатка), в каталог не входят, число «по формуле» в строку не подставляется; небольшие расхождения напечатанного с формулами (до нескольких процентов) остаются как напечатано с пометкой к строке в интерфейсе. ГОСТ Р 57837-2017 (340 строк) отдельной позицией не входит: он построчно совпадает с ГОСТ 35087-2024 и служит проверкой распознавания; в каталоге это псевдоним. Источники, проценты ячеек, подтверждённых независимо, и всё, что не взято — документы/разработка/сортамент_швеллеров_и_двутавров_2026-10-08.md.

- **Что берётся и что проверяется.** Все числа — как напечатано в стандарте (радиусы инерции приводятся в сантиметры). Генератор проверяет: массу против 0,785·A, W = 2I/h и i = √(I/A) в пределах точности печати, геометрию без галтелей и с галтелями (для двутавров с параллельными гранями полок — A, Ix, Iy в пределах 0,5 %), повторы обозначений и три перекрёстные сверки между стандартами. Непредусмотренное расхождение останавливает генерацию; известные расхождения самих стандартов перечислены в ANOMALIES с основанием и попадают в интерфейс пометкой к строке.
- **Файлы.** `hydroweb/data/beam_catalog.json` (полный каталог для тестов и справки; код бэкенда его не читает), `hydroweb/data/pile_sections.json` (14 прежних обозначений СТО АСЧМ 20-93 в прежней форме полей для стоек РРСК: core/pile_catalog.py → /api/rrsk/sections; генерируется тем же скриптом, числа — из таблицы стандарта), `frontend/src/data/beamCatalog.generated.ts` (интерфейс: data/strutCatalog.ts → BEAM_CATALOG, BEAM_STANDARDS). Руками не править: генератор с ключом --check проверяет, что файлы равны результату генерации (тест test_beam_catalog.py). В прежних

именах полей «Y» — сильная ось сечения (X–X в стандартах), «Z» — слабая. Карточки справочника сечений (modules/catalog/core/strutSectionCatalog.ts) строятся из тех же массивов: добавлена форма «channel»; идентификаторы 14 прежних двутавров СТО АСЧМ 20-93 не изменились («beam-20B1»), у строк других стандартов к идентификатору и названию добавляется стандарт («beam-20B1@35087-2024», «channel-12P@8240-97»), потому что «20Б1» разных стандартов — разные профили; меньший радиус инерции карточки у двутавра и швеллера — iy слабой оси.

- **Выбор в интерфейсе.** components/BeamSectionPicker.tsx — «стандарт → серия (тип) → профиль»; чистые функции — core/beamSelect.ts. Ключ поиска — пара «стандарт + обозначение»: одно обозначение («20Б1») в разных стандартах — разные профили. Сохранённая распорка хранит стандарт, обозначение и сами значения: прежние 14 строк «СТО АСЧМ 20-93» (20 Б1 … 40 Ш1) сохранены под теми же обозначениями, но числа в них теперь из таблицы СТО; стандарт, названный в проекте и отсутствующий в каталоге, по обозначению не угадывается. Обвязочный пояс (WaleDetail) устроен так же: тип «Двутавр» / «Двутавр х 2» открывает двутавры, «Швеллер» / «Швеллер х 2» — швеллеры (раньше список был один, двутавровый); если профиль пояса из проекта в каталоге не найден, его характеристики сохраняются как были.
- **Стойки РРСК.** Список стоек в components/RetainingWallTab.tsx остался плоским — 14 обозначений «СТО АСЧМ 20-93» из /api/rrsk/sections (выбор «стандарт → серия → профиль» здесь не вводился); числа в нём теперь из таблицы стандарта. Проект хранит значения сечения (beamJ, beamA, beamW), а не ссылку на таблицу, поэтому при отличии сохранённых значений от таблицы больше чем на 0,5 % вкладка показывает под списком строку с отличием и кнопкой «Обновить по таблице»; молча числа не меняются (тест RetainingWallTab.beamPicker.test.tsx).
- **Тесты:** hydroweb/tests/test_beam_catalog.py (21), test_pile_catalog.py (5); frontend: data/beamCatalog.test.ts (11), core/beamSelect.test.ts (12), components/BeamSectionPicker.test.tsx (9), StrutDetailDialog.test.tsx (24), RetainingWallTab.beamPicker.test.tsx (3), calc/beamCatalogImpact.test.ts (2).

Выписки сортаментов и норм, у которых нет потребителя в продукте ([0338-каталоги-b-0708])

В scripts/catalog/ лежат ещё три выписки, собранные тем же методом (число из первоисточника с указанием таблицы и страницы → проверка формулами → протокол рядом с данными). Ядро, интерфейс и данные продукта ими не менялись, JSON-каталогов и генераторов для них нет — только CSV, скрипты извлечения и проверки, тесты и документы.

- **Что выписано.** (1) Шпунт зарубежного изготовителя — 243 строки: 150 текущих (каталог 2027 года) и 93 старые (только сайт изготовителя). (2) Бетон и арматура по СП 63.13330.2018 и ГОСТ 34028-2016: классы бетона, модули Eb, классы арматуры (действующая редакция с Изм. № 1 и № 2 и, отдельным файлом, исходная редакция 2018 г.), диаметры арматуры, коэффициенты; scripts/catalog/concrete_sources/. (3) Арматурные канаты — 111 строк: ГОСТ Р 53772-2010 в двух редакциях, ГОСТ 13840-68 и таблица prEN 10138-3 по листку изготовителя (текст самого prEN платный и не читался); scripts/catalog/strand_sources/. Копии источников в репозиторий не входят: воспроизведение — по адресам и sha256 из протоколов.

- **Что проверено.** scripts/catalog/confirmation_stats.py считает долю ячеек, напечатанных так же во втором, отдельно набранном издании (другая публикация изготовителя или другой издатель текста стандарта); числа закреплены тестом test_catalog_confirmation_stats.py. Шпунт, текущие строки — 69,5 %; бетон и арматура — 78,5 %; канаты — 62,5 %. Это проверка переписи, а не истинности стандарта: если ошибся сам официальный текст, копии повторят его. Ячейки, у которых второго издания нет (например, классы сечения шпунта, ГОСТ 13840-68, 21 строка prEN), остаются подтверждёнными только самой выпиской. Тесты: hydroweb/tests/test_*_sheet_piles_source.py (19), test_concrete_sources.py (17), test_strand_sources.py (17).
- **Что из этого следует для продукта.** Для арматуры класса A400 по действующей редакции СП 63 Rs = Rsc = 340 МПа, Rs,n = 390 МПа; 350 МПа — исходная редакция 2018 г., действовавшая до 22.05.2020. Расхождения данных продукта с каталогом изготовителя (запись одного профиля в каталоге аналогов, значения Ru двух марок шпунтовой стали) в документе пакета перечислены и не правились — решение за владельцем. Разбор, источники, замечания независимой проверки и предложение по интеграции — документы/разработка/каталоги_пакет_2_*_2026-10-08.md, сортамент_арматурных_канатов_2026-10-08.md, справочные_данные_железобетон_2026-10-08.md.

Проверка распорки по СП 16: коэффициент φ (core/strut_check.py, calc/strutCheck.ts)

Окно «Расчёт распорки» (components/PipeStrutCheckDialog.tsx) и строки «Распорка N: прочность и устойчивость сечения» таблицы предельных состояний (core/supportChecks.ts → buildStrutSectionChecks) показывают ответ сервера: с [0530-проверки-распорки-и-пояса-в-api] шесть блоков (прочность и устойчивость при центральном и внецентренном сжатии, предельная гибкость, прогиб от собственного веса) считает hydroweb/core/strut_check.py по POST /api/checks/struts (глава 8, раздел «Проверки вне МКЭ-расчёта»). Клиентский calc/strutCheck.ts (runPipeStrutCheck(detail, L, input, curve)) остался опорой тестов: сверка клиент = сервер знак в знак — hydroweb/tests/data/strut_check_parity.json. Описанный ниже алгоритм один для сервера и прежнего клиентского модуля.

- **φ при центральном сжатии.** phiCentralCompression($\bar{\lambda}$, curve) — таблица Д.1 СП 16.13330.2017 (PHI_TABLE_D1, 43 строки $\bar{\lambda}$ = 0,4…10,0, столбцы a, b, c, значения ×1000 как напечатано; объединённые ячейки при $\bar{\lambda}$ = 4,6…5,6 (a и b) и 5,8…10,0 (все три) повторены по столбцам), линейная интерполяция; $\bar{\lambda} < 0{,}4$ — a и b: 1, c: 0,984 (первая строка); $\bar{\lambda} > 10$ — $7{,}6/\bar{\lambda}^2$. Условная гибкость $\bar{\lambda} = \lambda\cdot\sqrt{(Ry/E)}$ (п. 7.1.3), Ry — графа марки (compressionStrength). Перенос таблицы сверен с формулами (8), (9) и α, β таблицы 7: 126 из 129 ячеек в пределах половины единицы последнего знака; три узла, где напечатано иное, программа берёт напечатанное — $\bar{\lambda}$ = 3,8 тип a (напечатано 526 = $7{,}6/3{,}8^2$, формула (8) даёт 530,6) и $\bar{\lambda}$ = 5,8 тип c (напечатано 226 ≈ $7{,}6/5{,}8^2$, формула (8) даёт 226,9): в обоих ограничение $7{,}6/\bar{\lambda}^2$ напечатано уже в узле, хотя в тексте — «свыше»; $\bar{\lambda}$ = 4,4 тип b (392 против 392,54 по формуле — усечение при печати). До [0340-φ-распорки-0708] здесь стояла таблица неизвестного происхождения со ссылкой на несуществующую «таблицу Б.3», и гибкость делилась ещё на π.
- **Тип сечения.** bucklingCurveFor(detail, rolledBeamHeightMm?) — таблица 7 для оси, по которой проверяет окно (сильная): pipe — a; beam и doubleBeam — b, при высоте свыше 500

мм — a (примечание 1); channel — c; box — b; custom — c. Высоту двутавра из сортамента берёт calc/strutCurve.ts (strutBucklingCurve: findBeam по стандарту и обозначению; без стандарта — не угадывается, тип b); оба вызывающих места передают тип четвёртым аргументом. В calc/strutCheck.ts только import type: его транспилируют по одному файлу scripts/gw_compare/analog_2139_strut_ts.mjs и scripts/experiments/strut_phi_sp16_check.py. Тип и основание выводятся в тексте блока устойчивости, в подсказке окна (bucklingCurveNote / beamBucklingHint) и в пояснении строки таблицы предельных состояний (оно же в DOCX).

- **Что оставлено как было (описано, не менялось).** Блок 4: φe = φ/(1 + m), η = 1 — упрощение программы вместо таблиц Д.2, Д.3 (п. 9.2.2); устойчивость из плоскости (слабая ось) не проверяется; Ry_d = Ry (двойное деление на γm снято в [0417-единое-правило-Ry]: Ry = Ryn/γm, γm = 1,05, берётся один раз); γn в расчёт не входит (в окне «Расчёт распорки» рядом с полем γn стоит подпись «не входит в проверку: вводите усилие расчётным, с учётом γn», в пояснении к строке таблицы предельных состояний — «γn в проверку не входит»); блок прочности 1 при A = 0 или Ry = 0 и блок 3 при A = 0 не учитывают продольную силу (загрузка 0 или только от изгиба) — не менялись; блоки устойчивости 2 и 4 с [0340] при нулевой несущей способности и сжатии не пройдены. Разбор — документы/разработка/ф_распорки_д1_2026-10-08.md.
- **Предельная гибкость.** Блок 5 ссылается на п. 10.4.1 и таблицу 32 СП 16.13330.2017 (предельные гибкости сжатых элементов; таблица 33 — растянутых). До [0340-ф-распорки-0708] стояла ссылка на таблицу 31 — это коэффициенты расчётной длины колонн рам (п. 10.3.4), к предельной гибкости она не относится; ссылка исправлена и в строке источника таблицы предельных состояний (core/supportChecks.ts). Значение λ_max задаёт пользователь (по умолчанию 120); позицию таблицы 32 и коэффициент α (180 − 60α, 210 − 60α, 220 − 40α) программа не подбирает.
- **Проверки.** calc/strutCheck.test.ts (все 129 ячеек Д.1 как напечатано, согласие с формулами (8), (9), интерполяция, границы, тип сечения, контрольные сечения «было → стало», страж ссылок на «таблицу Б.3» — в исходниках кода любой падеж, включая винительный, в исходниках книг прежний шаблон: несуществующую «таблицу Б.3» книга разработчика упоминает сознательно, как прежнюю ссылку программы), calc/strutCurve.test.ts (тип по высоте для каждого двутавра каталога), calc/beamCatalogImpact.test.ts; независимый пересчёт и сетка «до/после» на коде двух коммитов — scripts/experiments/strut_phi_sp16_check.py (режимы table, control, predict, random, grid).

Клиент: смета (РИМ) на вкладке «Документы»

- **Состав.** `modules/documentation/estimate/EstimatePanel.tsx` — панель (создаётся при первом открытии подвкладки «Смета (РИМ)» и остаётся смонтированной скрытой, чтобы выбор и результат не терялись); `buildEstimate.ts` — позиции из ProjectQuantities и запрос к /calc (шпунт и связи — тонны БЕЗ запаса на материал, анкеры — метры); `normSelection.ts` — подбор нормы шпунта по классам из НАИМЕНОВАНИЙ норм, полученных от API (масса 1 м, глубина или длина, группа грунта), нет совпадения — норма не подбирается; `xlsxEstimate.ts` — выгрузка; `services/estimateApi.ts` + `estimateTypes.ts` — клиент с типизированными ошибками (unavailable, not_found, invalid, rate_limited, server, network, bad_response, aborted) и отменой запроса.

- **Без молчаливых умолчаний.** Ценовая зона — только из ответа /zones; способ погружения, схема использования шпунта (остаётся / извлекается), группа грунта и НДС — явный выбор пользователя (EMPTY_DRAFT: всё не выбрано, НДС «не учитывается»); пока выбора нет, позиция уходит на сервер без нормы и возвращается «не посчитано» с причиной. Границы классов норм («до N» включает N) — допущение, показывается в панели и в выгрузке. Результат помечается устаревшим по ключу JSON-запроса (stale), выгрузка блокируется до нового расчёта.
- **Выгрузка XLSX** (exceljs): листы «ЛСР» (12 граф Приложения 3 к 421/пр; суммы и произведения — формулы Excel), «Ведомость ресурсов», «ОСР и ССР», «Источники и допущения»; в «ОСР и ССР» — строки глав 8 и 9 с процентом, базой, основанием и пометкой «допущение». Правило: пустое остаётся пустым — не посчитано, значит нет суммы, в итог позиция не входит, итог помечается «НЕПОЛНЫЙ»; источник данных — на каждом листе и в нижнем колонтитуле. Графы 8–9 (базисная цена и индекс) API не отдаёт, поэтому не заполняются. Прежний ручной ввод расценок сохранён подвкладкой «Расчёт стоимости (ручной ввод)» (CostEstimatePanel, core/costEstimate.ts); та же ведомость выгружается листом «Расчёт стоимости» в XLSX-таблицах документации. Том ТЭО — отдельный документ (подраздел «Клиент: том ТЭО» этой главы).
- **Опоры и главы 8–9.** EstimatePanel: блок «Грунтовые анкеры» (тип анкера и группа грунта 1–4 — без умолчания; ANCHOR_TYPES, anchorSelectionParams в buildEstimate.ts: усилие для проверки предела — расчётное сопротивление тяги Rt;d проекта у стержневого, преднапряжение F у прядевого, не задано — «не проверено»); норма подбирается запросом GET /support-select (selectionRequests, кэш по selectionKey), в карточке позиции — условия, оговорки, строка «Область подбора — сборники ГЭСН-2022 05 и 09; …» с кандидатами вне области; в запрос /calc уходят и norm, и selection (сервер перепроверяет). Объёмы в единицах нормы: анкеры — метры (число × длина по оси), распорки и пояса — тонны без запаса на материал. Блок «Сводный сметный расчёт (ССР): ВЗиС, зимнее удорожание, НДС»: включение, процент и основание (подсказки из API с цитатами), глава ССР по умолчанию не выбрана (ssrChapter: null); при включённых ВЗиС или зимнем главы 8–12 дают предупреждение (ssrChapterConflict) и ССР в запрос не идёт.
- **Тесты:** buildEstimate.test.ts, normSelection.test.ts, EstimatePanel.test.tsx, xlsxEstimate.test.ts, estimateApi.test.ts, DocumentsTab.estimate.test.tsx (сумма по папке estimate и сервису — см. главу 9); панель проверяет в том числе список из 147 зон (выбор по числовому id), конфликт главы 10 и 11, карточку 05-01-014-02 («не проверено», «шпунтовому ряду», кандидаты вне области); фикстуры __fixtures__ — настоящие ответы бэкенда, обновляются одноразово сервером, не руками.
- **Ручной «Расчёт стоимости»: возвратный шпунт ([0306-выгрузки-источники], 07.10.2026).** modules/documentation/core/costEstimate.ts. Прежняя «доля невозврата» металла 0,15 (lossFraction) удалена: в тексте нормы такого числа нет. Вместо неё — turnovers (число оборотов, целое не менее 2, значения по умолчанию нет) и sheetPileSteelPerTon(n): таблица ГЭСН 81-02-05-2022, общие положения сборника 5, п. 1.5.6 (SHEET_PILE_TURNOVER_SOURCE) — 0,65 т на 1 т погружаемого шпунта при 2 оборотах, 0,40 при 3, 0,25 при 4–5, 0,22 при более чем 5; норма учитывает износ, потери и восстановление. Металл возвратного шпунта = масса × норма, в примечании строки — число оборотов, норма и масса. Пока числа нет или оно не по норме (1, нецелое), позиция «Шпунт

— металл» не считается: она попадает в новый список CostEstimate.pending с причиной (TURNOVERS_HINT), complete = false. Панель (CostEstimatePanel.tsx): поле «Число оборотов шпунта» (только при схеме «извлекается», пустое, min 2), подсказка с нормой, строка «не посчитано», показ выбранного расхода. XLSX (xlsxSpecification.ts): на листах «Расчёт стоимости» и «Ведомость объёмов работ» строка без количества не пропадает — количество пустое, причина в примечании, подпись «Без количества: …», «итог неполный». Режим «остаётся в грунте» не менялся (масса по проекту). Сметный движок РИМ (hydroweb/estimate) оборачиваемость по-прежнему не учитывает. Оговорка [0336-подсказки-0708]: четыре числа таблицы введены по пересказу п. 1.5.6, текст пункта не сверен, поэтому везде, где пользователю выводятся эти числа или ссылка на пункт, стоит единая строка TURNOVER_NORM_CAVEAT («Значения введены по пересказу нормы, текст пункта не сверен — проверьте по вашей редакции ГЭСН.»): в TURNOVERS_HINT, в примечании строки «Шпунт — металл» (steelNote), в причине «не посчитано», во всплывающей подсказке (title) к расходу рядом с полем оборотов в CostEstimatePanel.tsx и — через те же строки — в листах «Расчёт стоимости» и «Ведомость объёмов работ» XLSX; xlsxSpecification.ts и CostEstimatePanel.tsx собственного текста о норме не содержат. Расчёт и числа не менялись. Сторож costEstimate.caveat.test.ts (3) просматривает исходники frontend/src: файл со ссылкой на п. 1.5.6 или числами расхода, не внесённый в его список, ломает тест; в выгрузках DOCX такие числа сейчас не выводятся, а при появлении должны идти с оговоркой (exportedDocs.guard.test.ts). Тесты: costEstimate.test.ts (16), CostEstimatePanel.test.tsx (12), xlsxSpecification.test.ts (37). Схема использования шпунта и число оборотов — часть проекта ([0317-смета-остаётся]): HydrowebProject.sheetPile { usage, turnovers } (types.ts), чтение сохранённого значения — services/sheetPileProject.ts (normalizeSheetPile: нераспознанное = «не задано»; withSheetPile: пока выбора нет, ключа в проекте нет совсем, сохранённый проект прежний), состояние держит App.tsx и передаёт во вкладку «Документы» пропсами sheetPile / onSheetPileChange; в запросы к расчёту поле не входит (projectFieldFlow.ts — «отчёт»; currentProject его не содержит). Под переключателем «остаётся в грунте» — пометка PERMANENT_USAGE_NOTICE (только экран, выгрузка прежняя): расход шпунтовой стали для этого режима в ведомость не введён, текст п. 1.5.6 в данных платформы не приведён, значение по норме не подтверждено текстом. Источник чисел таблицы: 0,65 / 0,40 / 0,25 / 0,22 введены решением [0306-выгрузки-источники] по пересказу п. 1.5.6 в рабочих материалах проекта (вне репозитория), текста нормы в репозитории и в данных платформы нет — значения требуют сверки с текстом. Расход 1,01 т на 1 т в расчёт ведомости не введён: он есть в данных расценок ФГИС (ресурс 08.3.10.02 в расценках погружения шпунта 05-01-011, 05-01-012, 05-01-176, 05-01-182), но это факт данных расценок, а не текст п. 1.5.6; так он и назван в причине «не посчитано» (costEstimate.ts). Охранный снимок прежнего поведения — costEstimate.baseline.test.ts + __fixtures__/costEstimate.baseline.json (28 случаев, снят до правки; единственная правка фикстуры — текст причины «не посчитано» в четырёх случаях, количества и итоги прежние). Тесты: sheetPileProject.test.ts, DocumentsTab.sheetPile.test.tsx, App.sheetPile.test.tsx (в том числе: тело запросов к расчёту, Блюм и МКЭ, не зависит от выбора схемы шпунта), CostEstimatePanel.test.tsx (в том числе подсказка с оговоркой у расхода рядом с полем).

Клиент: том ТЭО (DOCX)

Отдельный документ, не входящий в пакет документов: кнопка «Том ТЭО (DOCX)» на вкладке «Документы» (ExportPanel.tsx → onOpenTeo) открывает диалог TeoDialog. Код — modules/documentation/teo/; вывод собирается из данных проекта, расчёта, каталога профилей, ответа сметного движка и введённого пользователем, расчётное ядро и API не менялись. Часть разделов остаётся полями «[заполнить]» до ввода данных пользователем.

- **Состав модуля.** types.ts, teoInput.ts (организация, объект, заказчик и обозначение — из реквизитов проекта), teoFinder.ts (условия проекта → вход движка «Подбора решений»: quickInputFromProject — слои γ, γsat, ф, c, E, глубина котлована, УГВ, равномерная пригрузка у стенки, что не учтено — caveats, нет слоёв или они не доходят до дна — ok: false с причиной; runFinder — generateAllSolutions и computeVariantDefault для добавленных профилей, расчёт каждого варианта — POST /api/calculate), teoCard.ts (карточка в проекте), teoEstimateTexts.ts (словарь формулировок сметного модуля), teoVariants.ts (варианты: профиль проекта, профили каталога для сравнения, профили из «Подбора решений» (variantsFromSolutions: не прошедшие и с ошибкой расчёта остаются в списке — заключение подбора, calcProblem, длина «не определена»), прочие конструкции; объёмы условного участка ⌈L/B⌉ × G × длина; свёртка ответа движка в стоимость варианта — металл, прочее, итог; неполная смета числа не даёт; variantSupports — крепления варианта), teoEstimate.ts (смета шпунтовых вариантов теми же planPositions / buildCalcRequest и запросом /api/estimate/calc, что на вкладке «Смета (РИМ)»; без выбора зоны, способа погружения и схемы использования — «не посчитано» с причиной; цена металла вне ФГИС ЦС — только введённая пользователем с основанием), teoContent.ts (модель тома — чистые данные без docx), docxTeo.ts (вывод через reports/gostDoc.ts), TeoDialog.tsx (диалог). Выбор пользователя на «Смета (РИМ)» доходит до диалога через необязательный onDraftChange панели EstimatePanel. Диалог сам запускает подбор решений для текущего проекта («Рассчитать варианты подбора»; результат устаревает при смене условий), флажок «Включить в том» у каждого варианта (по умолчанию включены профиль проекта, каталожные и прошедшие подбор; без включённых вариантов выгрузка недоступна), стоимость вариантов — /api/estimate/calc («Рассчитать смету по вариантам»); схема подбора — консоль, при непрохождении одна распорка на 0,3H, котлован осушён, ярусы и анкеры проекта не учитываются (оговорка попадает в текст тома).
- **Структура тома.** Титул (форма 12 ГОСТ Р 21.101-2026; TitlePageSpec.changeBox + changeBoxTable() в reports/gostDoc.ts — мини-таблица «Изм. / № док. / Подп. / Дата» слева внизу над городом и годом, как в образце: шапка и три пустые строки, включена только у тома ТЭО; поле «Свидетельство СРО (как печатать)» — TeoRequisites.sroCertificate, пусто — «Свидетельство: [заполнить]»), лист состава тома («-С»), текстовая часть («-Т»): содержание, введение, 1 «Общие сведения» (местоположение, гидрометеорологические условия, таблица уровней воды из проекта, грунты, сейсмичность, примыкающие сооружения), 2 «Выбор конструкции ограждения» (подраздел и таблица на каждый вариант, сравнение стоимости, таблица «Крепления и итоговая стоимость по вариантам» (comparisonSection → supportRows), вывод отдельно без креплений и с креплениями и только по вариантам, у которых известны нужные слагаемые), 3 «Обоснование марки шпунтового ограждения», приложение А (перечень и сокращённые ЛСР по вариантам, итоги, предупреждения, допущения и неучтённое движком — формулировками тома по словарю teoEstimateTexts.ts)

и приложение Б (документы-основания цен); последним листом — таблица регистрации изменений (форма 10).

- **Оформление.** Как у пояснительной записки: Times New Roman, всё чёрное, кегль из reports/docFonts.ts (заголовки 14/13 пт, текст и таблицы 12 пт, примечания 11 пт), интервал 1,5; режим листа — рамка и формы 5/6, либо лист без основной надписи по п. 4.1.5. Номера страниц в содержании — поля PAGEREF на закладки заголовков; для режима без основной надписи в word/settings.xml дописывается updateFields, чтобы Word обновил поля при открытии.
- **Принцип.** В документе только данные проекта, расчёта, каталога, ответа движка и введённое пользователем; чего нет — видимое поле «[заполнить]» с подсказкой или «не посчитано» с причиной. Слов о верификации и названий сторонних программ и поставщиков в тексте тома нет (проверяют тесты).
- **Карточка тома в проекте ([0504-тэо-по-ревизии], 26.10.2026).** Ввод диалога — необязательное поле проекта `teoCard` (types.ts, teo/teoCard.ts): реквизиты титула, которых нет в реквизитах проекта (СРО, главный инженер, город, год, архивный и экземплярный номера), тексты «Сведения о районе и выводы», уровни воды, листы, документы-основания цен, условный участок, профили каталога, цены металла и основания, свои конструкции, флажки «Включить в том», замечания к вариантам, режим без рамки. Цены, замечания и флажки хранятся по названию профиля, а не по номеру варианта подбора (он от запуска к запуску меняется). Организация, объект, заказчик, обозначение и местоположение берутся из реквизитов проекта и показаны блоком «Из реквизитов проекта» со ссылкой «Изменить в реквизитах проекта» (onEditRequisites закрывает диалог и открывает реквизиты); результаты подбора и сметы в проект не пишутся. `variantNotes` выведен в карточку (поле «Замечания к варианту» под ценой металла) и печатается абзацами в подразделе варианта. Ничего не введено — поля нет; чтение терпимое: неверное отбрасывается, а не роняет открытие проекта.
- **Словарь формулировок сметного модуля.** `teo/teoEstimateTexts.ts`: ответ сметного модуля сведён к формулировкам для проектировщика — допущения (assumptions[].id, ASSUMPTION_PHRASES), предупреждения (warnings[].id, WARNING_PHRASES), «не учтено» (NOT_ACCOUNTED_PHRASES), причины «не посчитано» по виду причины (humanReason) — без имён полей запроса, кодов справочников, имён файлов и указаний по интерфейсу, целыми фразами, без обрезки по знакам; чего в словаре нет, не переносится дословно, а считается и печатается строкой «Сметный модуль вернул ещё сообщений: N». Словарь сверяется с перечнями движка тестом (teoEstimateTexts.test.ts, 12: идентификаторы из hydroweb/estimate); сторож exportedDocs.guard.test.ts проверяет тома ТЭО со сметами (было 6 срабатываний проверок внутренних слов и вложенных скобок в 5 абзацах, стало 0).
- **Крепления в сравнении.** `variantSupports` (teoVariants.ts) считает крепления варианта на условный участок: у профиля проекта — по проекту (`calculateQuantities`): распорки по шагу вдоль стенки (L/шаг штук, масса распорки полная), пояса по длине участка, стоимость — по расценкам `costRates` «Распорки — металл» и «Обвязочные пояса — металл»; у решений подбора известно только число уровней связей (`bracingLevels`), у профилей каталога связи не рассматривались. Чего нет — сечения, шага, расценки — результат null и причина в примечании, а не ноль; анкеры называются в примечании и в стоимость не входят. `comparisonSection` выводит таблицу «Крепления и итоговая стоимость по

вариантам» (уровни связей, масса, металл, итог «ограждение и крепления»; «нет данных» там, где слагаемого нет) и вывод «с учётом креплений» только по вариантам с обоими слагаемыми; введение, примечания и выводы не обещают точности. Тесты: teoVariants.test.ts (22), teoContent.test.ts (47).

- **Не сделано.** Результат подбора решений и смета вариантов в проект не пишутся и после перезагрузки считаются заново (карточка и ввод пользователя в проекте есть). ЛСР в приложении А — сокращённая форма (полная 12-графная — в XLSX сметы). Подбор решений для тома не учитывает ярусы, анкеры, нагрузки с отступом и на стену (перечисляется в оговорках). Крепления в сравнении — только металл распорок и поясов по расценкам проекта и только у профиля проекта; монтаж, анкеры и НДС не включены, у решений подбора известно лишь число уровней связей, у профилей каталога связи не рассматривались. В панели «Реквизиты проекта» нет полей «Заказчик» и «Местоположение», хотя том их читает (раздел ниже).
- **Реквизиты расчётов в приложении А ([0546-тэо-реквизиты-фгис], 29.10.2026).** Абзац «Реквизиты расчётов» в `appendixA()` (`teoContent.ts`) дополнен функцией `priceDataRequisites(resp)`: акт цен зоны (`resp.zone.base_price_act`, из примечания 2 сплит-формы «…цен, утверждённых приказом от … № …», собирается `scripts/fgis/build_snapshot.py`; у зоны может быть пусто), версия КСР (`resp.prices.ksr_version`) и дата получения данных (`resp.prices.fetched_at`; оба из meta снимка через `hydroweb/estimate/engine.py`); те же поля печатает XLSX сметы и блок «Источник данных» карточки «Смета (РИМ)». Вставка стоит между зоной и нормами; дата печатается как в ответе (ISO), без пересчёта; чего в ответе нет, в абзаце названо («акт о ценах зоны не указан», «версия КСР не указана», «дата получения данных не указана»), а не пропущено. Версия ФСНБ-2022 уже стояла в строке источника (`resp.source`) и не дублируется. Новых констант нет, сметный движок и API не менялись. Тест — `teoContent.test.ts` («[0546] «Реквизиты расчётов»…»: фраза из полей ответа, порядок перед нормами и источником, один раз, без запретных слов и служебных фраз, вариант с пустыми полями).
- **Тесты:** teoContent.test.ts (48: порядок разделов, нумерация таблиц и ссылки, числа только из движка и ввода, «[заполнить]» и «не посчитано», нет запрещённых слов, основание цены металла и введённая стоимость вариантов — [0502]), docxTeo.test.ts (23: шрифт, цвет, кегль по шкале docFonts, ширины таблиц, подписи, формы 5/6, режим без надписи, содержание и PAGEREF, updateFields), teoVariants.test.ts, teoEstimate.test.ts, teoEstimateTexts.test.ts, teoCard.test.ts, teoFinder.test.ts, TeoDialog.test.tsx (22; карточка из проекта, блок «Из реквизитов проекта», замечания к варианту), DocumentsTab.teo.test.tsx — проверяют в том числе мини-таблицу на титуле, поле СРО и подбор в диалоге; пример-тест teoSample.test.ts пропускается без HYDROWEB_TEO_SAMPLE_DIR.

Оформление выгружаемых документов (ГОСТ Р 21.101-2026)

Единственное место, где заданы шрифт, цвет, поля и вид таблиц DOCX, — reports/gostDoc.ts, кегль — reports/docFonts.ts (SZ, полупункты: body, table и pageHeader 24, note 22, h1 и title 28, h2 26, changeBox 20 — мини-таблица «Изм.» на титуле тома ТЭО; TEXT_SIZES = 22, 24, 26, 28 — шкала для сторожа выгрузок); рамка и основная надпись — gostStamp.ts; режим «без штампа» — gostPlain.ts; сквозная нумерация таблиц и ссылки на них — reportLayout.ts. Пояснительная

записка, отчёты Блюма и МКЭ, сравнительный анализ и блок «Границы применения» берут оформление отсюда, а не держат своих констант.

- **Правила оформления:** Times New Roman везде (стили заголовков и гиперссылок переопределены — у docx-библиотеки они синие), весь текст 000000 без заливок (статус словом: «выполняется» / «не выполняется» / «не проверено»; у запаса в заделке — «выше ориентира» / «ниже ориентира», см. kzLabel.ts), кегль по docFonts.ts — раздел 14 пт, подраздел 13 пт (п. 6.6.2: подраздел мельче раздела), текст и таблицы 12 пт, примечания 11 пт (п. 6.12.2: мельче текста) — решение владельца 26.10.2026 [0502] по образцу тома ТЭО, п. 5.1.1 ГОСТ Р 2.105-2019 (14 пт для основного текста, допускает 13 и 11 пт) этим не выполняется; интервал 1,5, отступ 12,5 мм (пп. 5.1.2, 5.1.4); подпись «Таблица N — …» над таблицей, шапка с tblHeader, строки cantSplit и не ниже 8 мм, ширина не больше текстовой области; титул — отдельная секция без рамки (поля 30/15/20/20 мм).
- **Запас в заделке в документах: reports/kzLabel.ts.** Экспорты KZ («Кз») и KZ_DEFINITION («Кз — запас в заделке»), KZ_GUIDE_MARK («ориентир, не норматив»), KZ_GUIDE_NOTE — единственная оговорка о Кз в документе («Кз — калибровочная величина программы; [Кз] — ориентир, не норматив (по классу сооружения)»: строка «Обозначений» DOCX (KZ_NOTATION_MEANING), примечание листа «Результаты» XLSX Блюма, атрибут «Пункт» строки Кз в XML, примечание тома ТЭО, строка допущений сравнения вариантов), KZ_SOURCE_CELL («ориентир программы» — графа «Источник» таблицы проверок), kzGuideStatus(ok) («выше ориентира» / «ниже ориентира» / «не определён») и kzGuideCondition(min). Допустимое значение [Кз] по классу ответственности (core/limitStates.kEmbMinFor: 1,25 / 1,20 / 1,15 / 1,10) в выгрузках — ориентир, не норматив; ссылка на СП 116.13330.2012 у запаса в заделке снята и остаётся только у общей устойчивости [k_st] (core/slope_stability.py, /api/checks/stability). Единый список проверок core/limitStates.buildFemLimitStates общий для экрана и документов, поэтому с 06.10.2026 экран показывает те же слова (core/limitStates.ts берёт их из reports/kzLabel.ts; LimitStatesPanel выводит statusWord, «Ориентир» и «Характер значения»; плашка «Результат» — «выше / ниже ориентира»); выгрузка (generateDocx.ts) пропускает проверки через exportedCheck(), который у проверки id «embedment» подменяет condition, source (KZ_SOURCE_CELL), sourceNote и reason и добавляет statusWord; у проверки id «deflection» (предел прогиба — рекомендация СП 381, прим. к п. 6.1.7) добавляет statusWord «не превышает рекомендуемое значение» / «превышает рекомендуемое значение»; у остальных — слово результата сопоставления (femCheckStatusWord); поля limitLabel и sourceLabel (раньше — для раздела «Обоснование проверок») удалены в [0413]. В ПЗ (docxExplanatoryNote.ts) строка «Горизонтальное перемещение» таблицы предельных состояний печатает те же слова, а под таблицей стоят «Результат сопоставления» и NO_NORM_CONCLUSION («Соответствие нормам расчёт не устанавливает») с отсылкой к разделу границ применения; в записке МКЭ (generateDocx.ts, buildFemConclusions) раздел «Выводы и рекомендации» — пункты «Сходимость», «Прочность: Км … — не меньше 1,0», «Заделка: Кз … ⩾ [Кз] … — выше ориентира», «Перемещение: wmax … — не превышает рекомендуемое значение», «Крепления», затем conclusionSentence() (NO_NORM_CONCLUSION и «Проверки, не вошедшие в расчёт, — в разделе N»); overallVerdict() служит только экрану. Те же слова — в таблице проверок и титуле ПЗ (docxExplanatoryNote.ts: responsibilityTitleLine — «Класс сооружения: N; [Кз] = …», таблица,

результат сопоставления), в XML (generateXml.ts: «Пункт» и «Статус» строки запаса в заделке; с версии схемы 0.2-beta имя атрибута Кз у обоих методов — КЗаделки, прежний Kst убран, у метода Блюма допустимого значения Кз нет — «не сопоставляется»), в сравнительном анализе (generateComparisonDocx.ts) и в строке «embedment» границ применения (core/applicability.ts). Сторож exportedDocs.guard.test.ts («[Кз] — ориентир, не норматив» и блок «[0413, проверка волны 2] п. 1») разбирает абзацы и строки таблиц с «Кз» в образцах DOCX, ячейки XLSX Блюма, атрибуты XML и том ТЭО: если печатается число Кз, оговорка «ориентир, не норматив» стоит ровно один раз (kzCaveatProblem; число находит printsKzNumber; при политике метода Блюма числа нет и оговорки нет), «СП 116» рядом с Кз нет, слово «норматив» только после «не», статус строки Кз — «выше / ниже ориентира»; отдельно kzGuide.test.ts (exportedCheck, строка XML МКЭ) и проверка XML Блюма в blumXmlExtrema.test.ts. Числа и вычисления Кз не менялись.

- **Сведения о сертификате: reports/certificateInfo.ts.** Единственное место с реквизитами сертификата соответствия (CERTIFICATE: номер, бланк, срок, версия 1.1.0, система «PoliSoft», орган по сертификации, протокол испытаний, ГОСТ Р ИСО 9127-94 и ГОСТ Р ИСО/МЭК 12119-2000), CERTIFICATE_SECTION_TITLE («Сведения о программе и подтверждении соответствия») и certificateParagraphs() — четыре абзаца: программа и версия платформы, сертификат, что он удостоверяет (с обеими версиями), оговорка о добровольной сертификации. Используется в textTemplates.ts (подраздел 4.4 пояснительной записки) и в generateDocx.ts (programInfoParas — блок после методики в расчётных записках Блюма и МКЭ). Реквизиты сверяет с опубликованной страницей frontend/public/docs/certificate.html certificateInfo.test.ts; сторож выгрузок проверяет, что раздел есть во всех ПЗ и расчётных записках, содержит номер и оговорку «не является экспертизой проектной документации» и не называет Реестр российского ПО. Тот же сторож запрещает в выгрузках слова о поверке расчёта, слово «подтверждено» и производные от глагола «верифицировать».
- **Два режима листа** (stampMode): 'gost' — рамка (20/5/5/5 мм, границы страницы) и основная надпись: форма 5 на первом листе текста (за титулом, п. 5.2.2), форма 6 далее; в форме 5 графа 6 названа «Вид док.» (в редакции 2020 — «Стадия»); 'none' — по п. 4.1.5: список исполнителей, содержание с номерами листов (поля PAGEREF на закладки заголовков, включён updateFields), верхний колонтитул — обозначение документа, нижний — организация, наименование и «Лист N из M», поля 25/10/20/20 мм. Отчёты вкладки «Результаты» по умолчанию — 'none' (это отчёт программы, а не том проектной документации), документы вкладки «Документы» (ПЗ, сравнение) — 'gost'.
- **Параметры выгрузки на «Результатах»** (05.10.2026): reports/reportOptions.ts (значения и правила) + components/ReportOptionsPanel.tsx (форма). Состояние ReportOptionsState {withStamp, draft, gipCertificate?} держит App и передаёт в ResultsTab пропсами reportOptions / onReportOptionsChange / docInfo; в блоке «Формирование отчёта» (для Блюма — только при формате DOCX) стоят флажок «с основной надписью (ГОСТ)» (по умолчанию выключен: withStamp = false → stampMode 'none') и раскрывающийся блок «Реквизиты документа»: шифр (базовое обозначение), организация, разработал, проверил, нормоконтроль, утвердил, стадия (не указана / П / Р). Реквизиты вкладки «Документы» поднимаются в App (DocumentsTab получает sharedInfo и сообщает о каждом изменении через onInfoChange — вкладка размонтируется при смене вкладки и не теряет ввод; с

[0504] реквизиты ещё и часть проекта — поле projectInfo, оно читается при открытии проекта, не дожидаясь вкладки «Документы») и предзаполняют блок (resolveRequisites); введённое в отчёте главнее. Правило «пустое остаётся пустым» — requisitesPresent в generateDocx.ts: стадия на титуле печатается, только если задана; обозначение «<шифр>-<марка>.РР» — только при заданных шифре и стадии (марка выводится из стадии), без этого титул и надпись реквизитов комплекта не выдумывают. У кнопки выгрузки — пояснение (docxReportNote): в режиме без надписи в файле есть содержание с номерами листов (поля PAGEREF), Word при открытии спрашивает «Обновить поля?», ответ «Да» нужен, чтобы номера проставились (LibreOffice считает их сам). Флажок «справка ГИП после содержания» хранится в ReportOptionsState.gipCertificate (не менялся — undefined): gipCertificateOn(doc, opts) = opts.gipCertificate ?? (стадия реквизитов === 'П'); exportParams() передаёт значение в generateDocx как gipCertificate, а gipCertificateEnabled() в generateDocx.ts при undefined берёт то же правило по info.stage. Выбор пользователя главнее стадии.

- **Метод бокового давления в отчёте Блюма.** generateDocx.ts: lateralMethodText, lateralCoefficientCaptions и приложение с формулами берут метод, ФАКТИЧЕСКИ применённый в расчёте (wall.calcMethod уходит на сервер как method; δ = (δ/φ)·φ передаётся только при Кулоне, projectToRequest.ts), вместо прежней фиксированной фразы «Кулона-Ренкина»: Ренкин — Ka = tg²(45° − φ/2), Kp = tg²(45° + φ/2), δ не учитывается; Кулон — коэффициенты приведены к нормальной составляющей давления на стенку, для пассива при φ > 20° δ = 0 (СП 22.13330.2016, п. 9.23). Тест — reports/reportOptions.test.ts (13).
- **Обозначения.** `modules/documentation/core/projectMark.ts`: марка комплекта — настройка (auto: стадия П → КР, Р → КМ; КМ, КЖ, КМ+КЖ, своя до трёх букв), предзаполненной марки нет; обозначение «<база>-<марка>.<шифр>» (шифры ПЗ, РР, СМ — «СО» для КМ и КЖ не применяется, ГОСТ 21.110 п. 4.1); ПЗ раздела ПД при авто-марке — «<база>-КР-ТЧ» (п. 4.1.3.3 ГОСТ Р 21.101-2026), раздел на титуле — «Конструктивные решения» (ПП РФ № 87, п. 3(1) «г»). Версия платформы в документах — reports/platformVersion.ts (сверяется с файлом VERSION тестом). Номер части раздела КР (07.10.2026, [0311]): cleanBaseCode срезает марку в конце базы вместе с номером части (регулярное выражение TRAILING_MARK: КМ, КЖ, КМД, ПЗ, РР, КР, КР1, КР4.3), baseCodeSectionPart(code) возвращает номер («154-0920-П-КР4.3» → «4.3»), а marksForCodes() подставляет его в марку КР в обозначениях documentCodes, volumeCode, volumeContentsCode: «154-0920-П-КР4.3», «…-КР4.3-ТЧ», «…-КР4.3-С»; pdCompositionCode — по базе без марки («…-СП»); при марке КМ, КЖ или своей номер не подставляется. База в XLSX (makeCtx) тоже проходит cleanBaseCode. Отдельного поля части раздела в ProjectInfo нет; предупреждение validationRules.ts называет номер части на стадии П.
- **XLSX.** reports/xlsxStyle.ts + xlsxTable.ts — слой оформления на exceljs (динамический импорт, отдельный чанк ≈ 930 КБ): Times New Roman, чёрный, тонкие границы, двойная линия под шапкой, печать А4 альбом, fitToWidth=1, область печати, повтор шапки, колонтитул «Лист N из M», пересчёт формул при открытии; exceljs жёстко ставит Calibri 11, поэтому styles.xml правится после записи (serializeWorkbook). Формы — на сетке 5 мм (54 столбца = 270 мм): modules/documentation/export/gostForm.ts (основная надпись — упрощённая форма 3 ГОСТ Р 21.101-2026, 185 × 55 мм; форма 6 для последующих листов не реализована — оговорка об этом стоит один раз, в «Примечаниях» листа «Общие

сведения» (addGeneralSheet; STAMP_FORM6_NOTE с каждого листа-формы удалена в [0413]): надпись выполнена по форме 3 (п. 5.2.3), форма 6 не воспроизводится, надпись одна, в конце листа; примечания ведомости элементов называют причину пустых граф «Эскиз» и «Усилия для прикрепления» (п. Е.2 ГОСТ 21.502-2016)), xlsxSpecification.ts (листы: «Общие сведения», «Ведомость элементов» — форма 1, «Спецификация металлопроката» — форма 2 ГОСТ 21.502-2016, «Анкеры и распорки», «Ведомость объёмов работ», «Расход материалов», «Расчёт стоимости»), metalSpec.ts (строки металла, итоги формулами, масса в т). Спецификация металлопроката с 05.10.2026: шифр «<база>-КМ.СМ1» (ГОСТ 21.502-2016, п. 7.6: шифр и порядковый номер спецификации; выгрузка даёт одну СМ на комплект — номер 1; SM_SUFFIX) и масса позиции округлена до 0,1 т (прил. Л, п. Л.1; lineT/roundT), итоги суммируют округлённые значения (формат ячеек «0,0»), примечание на листе поясняет, что позиция легче 0,05 т показана как 0,0.

- **Документы тома в ПЗ** (05.10.2026, ГОСТ Р 21.101-2026, раздел 8.1): modules/documentation/export/volumeSheets.ts, обозначения — projectMark.ts (volumeCode, volumeContentsCode «<том>-С», pdCompositionCode «<база>-СП»). Включаются флажком в панели выгрузки «Документов» (состояние volumeSheets в DocumentsTab, по умолчанию включён; ExportPanel): содержание тома (п. 8.1.5, форма 2) со своей основной надписью — заполнены само содержание, на стадии П состав ПД и пояснительная записка этого файла, остальные строки и «Общее количество листов» пустые; состав ПД (п. 8.1.6, форма 13) — только стадия П, строки пустые (допускается комплектовать отдельным томом — тогда флажок снимают); таблица регистрации изменений (прил. М, форма 10) — отдельный последний лист, строки пустые (М.3 допускает не приводить её до первого изменения). Каждый документ — в своей секции с надписью (форма 5 на первом листе, форма 6 далее); Листы документов тома нумеруются с 1 («Листов 1»), текст записки — тоже с 1: «Листов» — PAGEREF на закладку HW_TEXT_END в конце текста (gostDoc.createGostDocument, GostStampProps.totalSheetsBookmark; NUMPAGES считает страницы всего файла, SECTIONPAGES LibreOffice не вычисляет); без листов тома нумерация сквозная. Поля пустые намеренно: другие тома, листы и изменения — не данные программы.
- **Справка ГИП** (07.10.2026, [0311]): export/volumeSheets.ts — GIP_CERTIFICATE_TITLE, gipCertificateText(objectName), buildGipCertificateBody(info, cols), buildGipCertificate(info) (лист пояснительной записки: название листа и содержимое) и buildVolumeContents(input, withComposition, withGipCertificate) (строка «Справка ГИП» без обозначения в содержании тома). С [0401-отчёты-мкэ-пз-тэо-чертёж] это ПУСТОЙ БЛАНК: наименование объекта (info.name, не введено — PLACEHOLDER «[заполнить]»), заголовок «Справка главного инженера проекта», поле «Заключение … [заполнить]» с оговоркой, что программа заключений о соответствии не формирует (gipCertificateText), таблица подписи «Главный инженер проекта» с полями «[заполнить]» для фамилии, подписи и даты; поля ГИП в ProjectInfo не заведено, от имени пользователя ничего не заполняется. ПЗ (docxExplanatoryNote.ts): справка — первый лист текста, сразу после документов тома, первый раздел начинается с нового листа; параметр gipCertificate в NoteDocInput и exportExplanatoryNoteDocx, не задан — включена при info.stage === 'П'; флажок — в ExportPanel («Справка ГИП после содержания тома …»), состояние gipChoice в DocumentsTab (undefined — по стадии, выбор главнее), независим от флажка volumeSheets. Расчётная записка (generateDocx.ts; по умолчанию бланк НЕ вкладывается — gipCertificateEnabled(): не задан → false; gipCertificateOn() в reportOptions.ts):

buildGipCertificateSection() — первым разделом после содержания, breakBeforeNextHeading() из reportLayout.ts переносит следующий заголовок на новый лист; параметр gipCertificate обоих построителей (Блюм и МКЭ). Ссылка на постановление Правительства РФ № 985 из текста убрана: по замечанию рецензента оно утратило силу (реквизиты не подтверждены; проверка — за владельцем продукта).

- **Числа.** quantityCalculator.ts: масса шпунтины — G × L (G — кг/м.п. ОДНОЙ шпунтины), а не g × L (g — кг/м² стенки), как было: сталь, спецификация, расход материалов, ТЭО и ПЗ завышались в 1/B раз (контроль: 17 шпунтин по 10 м ECO 622T — 14 773 кг, ECO 632 — 19 380 кг, в тесте); у распорки без сечения и анкера без массы тяги масса «не определена» (massKnown = false, предупреждение в пакет и ТЭО), вместо выдуманных «труба Д200» и 50 кг/м.
- **Десятичная запятая: reports/numFormat.ts и testing/decimalDots.ts ([0311], 07.10.2026).** numFormat.ts — единственное место, где задан способ записи числа для текста Word: fmtNum(v, digits) (фиксированное число знаков: «1042,8», «0,30»), fmtTrim(v, digits) (без хвостовых нулей, не больше digits знаков), fmtRaw(v) (String(v) с запятой); для null, undefined и нечисла — прочерк «—». commaDecimals(text) ставит запятую в десятичных числах готовой строки, которую написала программа (значения проверок, формулы, текст шаблонов), и НЕ трогает: ссылки на нормы и пункты (регулярное выражение NORM_REF: «п.», «ф.», «табл.», «прил.», «рис.», «разд.», «§», «№», СП, ГОСТ, СНиП, ТР, ТУ, ISO, EN, DIN, ГЭСН, ФЕР, ТЕР, МДС, РДС, РД, СТО, СТУ, ОСТ с номером, включая перечисления «ф. 9.8 и 9.12»), даты, версии и обозначения («05.10.2026», «1.1.0», «КР4.3») и номера пунктов или разделов в начале ЛЮБОЙ строки текста (LEADING_ITEM_NO; строки обрабатываются порознь — первая версия защищала только начало всей строки, и «6.2.», «9.4.» в многострочном тексте шаблонов превращались в «6,2.»). Применено в generateDocx.ts (num() — через fmtNum, para() — через commaDecimals; расчётные записки Блюма и МКЭ), explanatoryNoteGenerator.ts (переменные — fmtNum / fmtRaw / fmtTrim; текст шаблонов — commaDecimals ДО подстановки переменных, поэтому строки, введённые пользователем — название объекта, основание, названия грунтов, обозначения сечений, — не затрагиваются; textTemplates.ts по запятой не правился), docxExplanatoryNote.ts (спецификация, таблица предельных состояний, расход материалов, титул), generateComparisonDocx.ts, applicabilityDocx.ts. Том ТЭО печатал запятую сам (teoContent.fmt, Intl ru-RU). Не меняются: XLSX (числа — значения ячеек), XML, JSON, запросы к серверу и любые числа в вычислениях. testing/decimalDots.ts: decimalDotContexts(text) — сторож, независимый от форматтера (свой список допустимых мест ALLOWED: ссылки на нормы, даты и номера из трёх и более частей, обозначения с буквами и цифрами, номер пункта в начале абзаца или ячейки): ищет в тексте DOCX любое «цифра.цифра» вне этих мест и возвращает контекст. Подключён к образцам exportedDocs.guard.test.ts и к тому ТЭО (docxTeo.test.ts); тесты форматтера — numFormat.test.ts (13).
- **Что запрещает exportedDocs.guard.test.ts** (выгрузки генерируются на демо-проекте testing/demoFixtures.ts и распаковываются; основные образцы: ПЗ — стадия Р, стадия П, марка КЖ явно, проект без сечения распорки, стадия П с шифром КР4.3; расчётные записки Блюма и МКЭ — без надписи, с надписью, МКЭ без реквизитов, МКЭ с распорками из двутавра 30Ш1 (ГОСТ 35087-2024), МКЭ стадии П со справкой ГИП; сравнительный анализ; затем МКЭ с заметками о пороге пригрузки и с оговорками бермы, отчёты Блюма на входах

тестовой выгрузки 15.10.2026 — П1, П2, П3б, — отчёт МКЭ П3б (обычный лист и с рамкой) и П3 П3б; отдельно, только на внутренние формулировки, — тома ТЭО, отчёты МКЭ и П3 П1, П2). Запрещено: любой шрифт, кроме Times New Roman; любой цвет, кроме 000000, заливки и тема; названия изготовителей и сторонних программ, «КЖ» без явного выбора марки; таблица шире текстовой области своей секции; таблица без повторяемой шапки и со строками, которые рвутся; неполная нумерация «Таблица N — …» и таблица без ссылки в тексте; кегль вне шкалы docFonts.TEXT_SIZES (11 / 12 / 13 / 14 пт; [0502] проверяет и стили Heading1 = 14 пт, Heading2 = 13 пт, текст по умолчанию и ячейки таблиц 12 пт, примечания под таблицами 11 пт); пустые абзацы для отступа; заголовки не полужирные или с точкой в конце; отсутствие формы 5 / 6 в режиме 'gost' или штампа в режиме 'none'; внутренние идентификаторы связей в тексте; [0404] метка связи без кавычек рядом со словами «анкер» / «распорка» / «учтена» («учтена только первая — S1»), слово «ярус» вне ссылки на СП 381 прил. И п. И.4, указания «выберите / нажмите / перейдите / откройте», вложенные скобки в тексте для людей, слова «прочность … обеспечена» и «Условие … выполняется»; [0405] слова «выполняются», «требуется» (кроме «не требуется»), «не выполнено», «укажите / используйте / задайте / отключите» (шаблон метки связи не срабатывает после слова «формата»: «формата А3, связи» — размер листа) и вложенные скобки в записке и томе ТЭО при реквизитах со скобками (образцы «П3, реквизиты со скобками» и «том ТЭО, реквизиты со скобками» — только внутренние формулировки). По содержанию текстов: нет чисел производства работ и охраны труда без адреса нормы («не менее 15 м», «1,1 м», «±50 мм», «0,5 м», СНиП 12-03 и 12-04), нет «Допустимое перемещение (H/100)» и «Предельно допустимое горизонтальное перемещение верха ограждения» — предел назван рекомендацией СП 381 (прим. к п. 6.1.7); Ry = Ryn/γm с γm = 1,05 (табл. 3 СП 16), γc названо допущением программы, не нормой; запас в заделке не печатается как «kst» / «k_st», расшифровка Kз — не более одного раза, рядом с Kз нет «СП 116», а слово «норматив» стоит только с «не»; нет слов о поверке расчёта, слова «подтверждено» и производных от глагола «верифицировать»; нет заявления «удовлетворяет условиям / требованиям … деформативности / устойчивости / жёсткости», строки «Деформативность … ВЫПОЛНЯЕТСЯ», фразы «по принятым проверкам условия ВЫПОЛНЯЮТСЯ» и «условие деформативности ВЫПОЛНЯЕТСЯ» (порогов в нормах нет, [0313]; для прогиба exportedCheck() в kzLabel.ts подставляет statusWord «не превышает / превышает рекомендуемое значение», вывод перечисляет результат сопоставления и оговаривает «Соответствие нормам расчёт не устанавливает»); в тексте нет числа с десятичной точкой вне допустимых мест, а номера пунктов («6.2.»), подразделов («3.1»), ссылки на нормы и шифры остаются с точкой; у шифра с номером части («154-0920-П-КР4.3») нет «КР4.3-КР»; в списке «Формулы» раздела «Проверки предельных состояний» у распорки из двутавра или швеллера стоит подсказка про сильную ось, таблицу Д.1 СП 16 и «слабая ось не проверяется», у трубы подсказки про ось нет и назван тип сечения а; числа расхода шпунта по п. 1.5.6 ГЭСН в тексте выгрузки допустимы только вместе с оговоркой «значения введены по пересказу нормы, текст пункта не сверен» ([0336-подсказки-0708]). Сторож проверяет и сам себя: числа с точкой он находит, номера и шифры пропускает. Для отчётов метода Блюма и XML сторож ([0400-отчёты-блюм-xml]) дополнительно требует: разделы нумеруются 1…N подряд, у «Замечаний и неучтённых данных» и «Границ применения» есть номер; нет слов «ВЫПОЛНЯЕТСЯ», «ПРОХОДИТ», «условие устойчивости / прочности выполняется»; раздел о заделке называется «Глубина заделки и длина шпунта»; методика

не заявляет проверку касательных напряжений и перемещений, а строка «Прочность ограждения по металлу» в границах применения — «выполнена частично»; в XML нет атрибутов «Kst», «Утилизация», «MmaxКНмм», «QmaxКНм», «ЕМПа» и статусов «ВЫПОЛНЯЕТСЯ» заглавными, версия схемы «0.4-beta» ([0512]; 0.3-beta — [0401]; 0.2-beta — [0400]); в тексте ячеек XLSX отчёта Блюма нет чисел с десятичной точкой. Блок [0401-отчёты-мкэ-пз-тэо-чертёж] требует во всех образцах и томах ТЭО: нет «ярус» как номера в общем списке («Опора ярус», «Усилие в опоре, ярус»), нет «228,7 кН/м» у анкера, нет готовой фразы «разработана в соответствии с действующими нормами» и «№ 985», нет строки «Расстояние от дна котлована до крепления», нет «расчет» без «ё»; в отчётах МКЭ есть раздел «Параметры расчёта» с k_s и графой «Основание»; у суглинка без модуля деформации в таблице грунтов — «—», ф — «20,4»; у отчётов вкладки «Результаты» с рамкой — «Содержание» и updateFields.

- **Спецификация DOCX ([0527-спецификация-docx]).** Причина: приказ Минстроя № 783/пр допускает XLSX только для сметных документов (пункт списка готовности «Форматы и имена файлов по требованиям к подаче», [0505-чек-лист-пп87-ецпэ]), спецификация подаётся текстовым документом; книга XLSX осталась без изменений в составе и числах. `modules/documentation/export/docxSpecification.ts` (`buildSpecificationDocument`, `exportSpecificationDocx`; файл `<проект>_спецификация.docx`): лист в рамке с основной надписью по формам 5 и 6 (общий `reports/gostDoc.ts`, как у записки), без титульного листа, кегль из `reports/docFonts.ts`; обозначение `<база>-КМ.СМ1` то же, что на листе «Спецификация металлопроката» книги. Два раздела: «Спецификация элементов ограждения» (таблица записки, позиции и количества те же, что на листе «Ведомость элементов») и «Спецификация металлопроката» (форма 2 ГОСТ 21.502-2016, как лист книги; округление до 0,1 т по прил. Л, п. Л.1 — номер пункта взят из кода книги). Форма 7 и ГОСТ 21.110 в документе не названы: в коде (`projectMark.ts`, ссылка на п. 4.1 ГОСТ 21.110) спецификацию «СО» для комплектов КМ и КЖ не применяют. Общие строки формы 2 вынесены в `export/metalSpec.ts`: `buildMetalTable` (строки и суммы в порядке печати), округление `lineT`, `roundT`, `sum` и шифр `SM_SUFFIX`; `markColumns(lines, max)` — на листе DOCX не больше пяти столбцов массы по элементам (`DOCX_MAX_MARK_COLUMNS`), при большем числе марок столбец на группу с общей буквой; лист XLSX берёт отсюда округление и шифр, суммы в книге по-прежнему формулы таблицы. Кнопка «Спецификация DOCX» в `ExportPanel` стоит рядом с «Скачать пакет документов» и «Том ТЭО (DOCX)»; в пакет она не входит; отметка для списка «Готовность к экспертизе» та же, что у пакета (вид `package`, отпечаток `packageStamp`); после конца пробного периода выгрузка закрыта общим `exportBlockReason()`. Тесты: `docxSpecification.test.ts` (13: равенство DOCX и книги XLSX на демо-проекте — позиции, единицы, количества и форма 2 построчно по результатам формул книги; рамка и обозначение; запятая и запретные слова; нет металла; больше пяти марок; имя файла), `reports/exportedDocs.guard.test.ts` (два новых образца: спецификация DOCX и она же для проекта без сечения распорки; образцов 32), `ExportPanel.test.tsx` (+4), `DocumentsTab.readiness.test.tsx` (+4).
- **Тесты-сторожа и счётчики:** exportedDocs.guard.test.ts (185), gostStamp.test.ts (7), xlsxSpecification.test.ts (37), metalSpec.test.ts (8), docxSpecification.test.ts (13), projectMark.test.ts (18), specificationGenerator.test.ts (10, итог по шпунту), gipCertificate.test.ts (13), stageResultsTable.test.ts (8, таблицы «по этапам»), certificateInfo.test.ts (8), textTemplates.test.ts (19), titleBlockBuilder.test.ts (15), volumeSheets.test.ts (7),

reportOptions.test.ts (13), ResultsTab.reportOptions.test.tsx (10), bookBlumClaims.test.ts (12) и bookFemClaims.test.ts (7) — цитаты книги пользователя против выгрузок, bookDevClaims.test.ts (6) — утверждения этой книги о выгрузках против кода, bookRelease613Claims.test.ts (12) — утверждения книг о таблицах связей (столбцы и ячейки из AnchorsTab.tsx, формулы жёсткости из supportStiffness.ts), группах готовности «По Положению № 87» и «Для подачи через ЕЦПЭ», томе ТЭО и реквизитах проекта против кода, bookSync.test.ts (5) — фразы исходников трёх книг есть в опубликованных HTML, зеркало книги поставщика равно тому, что пишет книга, сторож не слеп, printedArithmetic.test.ts (6), disabledTests.test.ts (2), gostDoc.keepNext.test.ts (3). Новая выгрузка с цветом, чужим шрифтом, таблицей шире поля или числом с точкой ломает сторож.

- **Чистка выгрузок ([0413-чистка-выгрузок], 20.10.2026; расчётное ядро и ответы API не менялись; числа результатов — в таблицах DOCX, ячейках XLSX и числовых атрибутах XML — совпадают с версией до чистки (499df4bc), отличаются только числа внутри пояснительных текстов).** Замечания ядра: reports/clientWarnings.ts — ENGINEERING_REWRITES (замечания не длиннее 20 слов: распорка растянута — одно на распорку, срез «по силам» при ф = 0, диапазон ф, марка стали шпунта и стоек, длина распорки при чертеже, откат бермы с причиной и числами — shortFallbackReason), BERM_ASSUMPTIONS_TEXT (условие применимости модели с отпором), BERM_FALLBACK_EFFECT_TEXT (последствие отката на «пригрузку на дне» — одна строка на документ), KS_FROM_E, clientFemWarnings(list, ctx, {xml}); для XML — xmlFemWarnings(fem, ctx) и clientXmlWarningText: одно замечание ядра — один элемент «Предупреждение» с номером (исходный список — FemCalculationResults.rawWarnings, services/femResults.ts), документы Word получают слитую редакцию. Границы применения: core/applicability.ts — поле showNote (пояснение печатается и при статусе «выполнена» / «не выполнялась» у строк deflection, zakoly, time_effects), applicabilityStatusText(); reports/applicabilityDocx.ts — три столбца «Проверка — норма — статус»; в ПЗ — только невыполненные проверки. Параметры расчёта: reports/modelAssumptions.ts — LOAD_FACTOR_ROW_LABEL / LOAD_FACTOR_ROW_VALUE (γf = 1,0), STRIP_LOAD_RULE_TEXT, STRIP_LOAD_ANNEX_E_TEXT, assumptionBasis() (графа «Основание»: пункт нормы, «задано проектом», «параметр программы», «допущение программы»; пустых нет), reports/blumAssumptions.ts — γf в строках «Принято в расчёте». Записка МКЭ (reports/generateDocx.ts): FEM_METHOD_TEXT (методика в три предложения), buildFemLimitStatesTable (таблица проверок; «Формулы:» с подстановкой чисел — metalMarginSubstitution из steelResistance.ts, Кз, wmax ⩽ min(H/100; 100 мм); «Исходные данные проверок распорок и поясов» и определения F, s, L, q, μ, λ_max, ℓ, η), buildFemConclusions и conclusionSentence (одна оговорка вывода и номер раздела границ применения), «Обозначения» только по использованным символам. Усилия связей: reports/supportForces.ts — wallNormalText, femSupportForceLines (нормальная составляющая и «на 1 м стенки нормально к стенке … при шаге …»). Метод Блюма: blumDocxParts.ts (раздел «Оценка перемещения w»: оценка, рекомендуемый предел min(H/100; 100 мм), «с пределом не сопоставляется»), blumDeflection.ts (blumDeflectionSchemeText — атрибут Оценка@РасчётнаяСхема в XML), оговорка п. 6.2.7 СП 381 в методике, предел перемещения — в «Основных результатах», приложении «Расчётные формулы», XLSX и XML. ПЗ: modules/documentation/core/explanatoryNoteGenerator.ts — renumberSections (сквозные

номера 1…N по исходному номеру раздела; подразделы подряд внутри раздела; ошибка на раздел без номера, третий уровень, подраздел раньше своего раздела и повтор ключа раздела верхнего уровня — [0502]: «Приток воды в котлован» имел ключ «9», как «Замечания и ограничения», и при расчёте воды в записке было два раздела с одним номером; теперь у раздела воды свой ключ WATER_SECTION_NUMBER = '8W'; порядок разделов: расчётные 1–5, «Проверка узлов крепления» (перед первым из разделов 6–9 шаблона), типовые 6–8, «Приток воды в котлован», «Замечания и ограничения», «Границы применения»; sectionNumbering.test.ts, 35 проверок: 16 сочетаний «типовые × замечания × вода × узлы» — номера 1…N без повторов, подразделы подряд, порядок разделов; повтор ключа — ошибка; заголовки DOCX — 1…N), notationTable; noteSettings.includeTypicalSections и флажок «Типовые разделы 6–8» на уровне проекта (types.ts HydrowebProject.typicalSections, services/sheetPileProject.ts normalizeTypicalSections / withTypicalSections, services/projectFieldFlow.ts, App.tsx, DocumentsTab — props typicalSections / onTypicalSectionsChange; тест App.typicalSections.test.tsx). Спецификация и ТЭО: xlsxSpecification.ts (примечания к пустым графам «Эскиз» и «Усилия для прикрепления», основания отступлений от форм, правило округления массы), teoContent.ts и docxTeo.ts (требования Методики № 421/пр без искажений, «не пересчитывались» только когда это верно, сейсмика — ссылка на расчётную записку, «государственной регистрации»). Сторож служебного: testing/internalPhrases.ts — SERVICE_PATTERNS на классах Unicode («в разработке», «beta», [NNNN], пути репозитория, статистика внутренней сверки, эталон; прежние служебные слова о выкатке сохранены в перечне как защита) для DOCX, XLSX и XML; тот же перечень в scripts/experiments/reports_test_1015/check_outputs.py (самопроверка check_outputs_selftest.py). Блоки exportedDocs.guard.test.ts: «[0413] чистка примечаний» (замечания не длиннее 20 слов, в том числе образцы «все форматы замечаний сервера о бермах»; повторы фраз из 8 слов и более; один вывод на документ; служебное; ссылка «См. таблицу N.» не засчитывается таблицей СП 16), «[0413] нумерация разделов и подразделов» (разделы подряд, оглавление и ссылки «в разделе N» согласованы с заголовками, сторож не слеп) и «[0413, проверка волны 2] п. 1…п. 10»: Кз — одна оговорка на документ; усилие анкера и легенда знака; формулы раздела проверок с подстановкой (пересчёт по напечатанным числам даёт напечатанный результат); методика МКЭ и Ка, Кр; Блюм — п. 6.2.7 и предел перемещения (при политике его нет нигде); оговорки области применимости и откат бермы; обозначения, «УВ слева / справа», определения β, I, ф. Каждое восстановленное содержание проверено мутацией: удаление фрагмента из кода роняет тест. Тест reports/bookBlumClaims.test.ts сверяет цитаты книги пользователя («Итоговые выводы»: прогиб при политике и в учебном режиме; «Записка метода Блюма») с выгрузками П1, П2 и П3б.

- **Раунд 2 исправлений чистки ([0413-чистка-выгрузок], 21.10.2026; расчётное ядро и ответы API не менялись; числа результатов в таблицах DOCX, ячейках XLSX и числовых атрибутах XML равны версии до чистки (499df4bc), кроме перечисленного в журнале).** γf в XML МКЭ: core/applicability.ts — поле noteInXml у записи load_factors (LOAD_FACTOR_ROW_VALUE (γf = 1,0) стоит и в «Параметрах расчёта» Word), generateXml.ts печатает «Пояснение» у частично выполненных проверок, у строк с showNote и у noteInXml; ПредупреждениЭтапа — число замечаний ядра на этапе (FemStageResult.rawWarnings, services/femResults.ts), а не частей клиентской редакции. Формулы раздела «Проверки»: reports/steelResistance.ts —

sig3Text (три значащие цифры: η = 0,0381), marginResultText, metalMarginSubstitution (возвращает null, если запас ядра по напечатанным W, A, Ry, Mmax не воспроизводится — подстановка не подгоняется), catalogRynFromWarnings (Ryn марки вне перечня берётся из замечания сервера «Марка стали не задана … принято по каталогу … — S355GP, 355 МПа»; нет замечания — «Ry не определено»); reports/generateDocx.ts — «K = N_пред / N_расч» с числами по каждому креплению, «M = F·s/4» в исходных данных пояса; суммы Kз не печатаются: F_lim и F внутри ядра (_compute_K_embedment), в ответе API их нет, а пересчёт на клиенте запрещён — в строке сказано, что суммы считает ядро; modules/documentation/core/explanatoryNoteGenerator.ts — п. 4.3: η тремя знаками, запас подстановкой. Сторож арифметики напечатанного — testing/printedArithmetic.ts (checkMetalMarginSubstitutions, checkSupportRatios, checkWaleMoments, checkStrengthPair; интервалы округления напечатанных чисел и допуск пересчёта читателем 1,5 %; «1/0,04 при Км = 26,22» находится). Замечания не длиннее 20 слов — все форматы: reports/clientWarnings.ts — shortFallbackReason (причина отката; не умещающаяся в 20 слов — вторым замечанием), narrowBermText (перечень этапов узкой бермы: до двух этапов с числами, больше — диапазон), мера предельного пассивного давления по клину — отдельное замечание этапа, BERM_FALLBACK_EFFECT_TEXT («Откат на «пригрузку на дне»: …; на единичном замере +15 %» — один замер, а не граница), прочие правила ENGINEERING_REWRITES (механизм, перемещение больше 10 % длины стенки, вода у лица, перегрузка распорки, отметка установки анкера); clientFemWarnings делит замечание по переводу строки (XML — один элемент на замечание ядра). Фикстуры настоящих строк сервера: testing/bermNotesServer.json (19 строк: форматы и причины отката, откат трёх берм, узкая берма на четырёх этапах; их равенство с сервером проверяет test_berm_notes_server_formats.py) и testing/longServerNotes.json (механизм, перемещение, вода у лица, перегрузка распорки, отметка анкера). Харнес выгрузок: berm_oracle.py (prism_reach, fit_at, expected_narrow, printed_remark_issues, server_remark_issues — этапы и числа замечаний о бермах из геометрии запроса независимо от клиента и сервера), check_outputs.warning_length_findings (длина замечаний), SERVICE_RES; мутации раунда 2 — scripts/experiments/reports_test_1015/mutations_round2.py и mutations_round2.json (36 откатов исправлений, в том числе правок книг: выбранные тесты обязаны упасть; выживших нет). Вёрстка: reports/gostDoc.ts — compact ставит keepNext на короткие абзацы между заголовком и ссылкой «См. таблицу N.» (PARA_META), чтобы заголовок не оставался один внизу листа. Сторожа: testing/internalPhrases.ts — SERVICE_PATTERNS ловит пометку журнала с запятой, номер записи без скобок, пути без корня, имена файлов и номера опытов; testing/disabledTests.ts (findDisabledTests) — запрет it.todo, it.skip и .only в тестах; в exportedDocs.guard.test.ts — блоки «[0413, раунд 2] п. 1 … п. 10». Тесты книг: bookBlumClaims.test.ts (цитата ячейки — целиком, с кавычками-границами), bookFemClaims.test.ts (пункты раздела 6 пояснительной записки по флажку), bookDevClaims.test.ts (утверждения этой книги о выгрузках — экспорты, значения в скобках, число строк фикстуры, удалённые поля), bookSync.test.ts (длинные фразы исходников книг есть в опубликованных HTML).

- **Короткие замечания в отчёте Блюма и XLSX ([0479-выгрузки-короткие-замечания], 24.10.2026; числа и значения прежних ключей ответов API не менялись, `warnings` прежние).** Сервер: `core/notes.py` — восемь правил для замечаний классического пути (читаемый код и законченная строка вместо хэша и обрезки многоточием:

`blum_upper_bound`, `embedment_margin_low`, `delta_default`, `gamma_sat_default`, `phi_out_of_range`, `corrosion_allowance`, `mmax_anomalous`, `length_not_above_excavation`; правила общие с путём МКЭ); `/api/calculate` отдаёт `notes` тем же механизмом, что и `/api/calculate/fem` (тест `test_notes_classical.py`, в том числе совпадение снимков ответов `testing/blumApiResponses.json` с механизмом). Клиент: `reports/blumAssumptions.ts` — `blumAssumptions` сопоставляет текст `warnings[i]` с `notes[j].explanation`: строка раздела — `notes[].line`, пояснение — очищенный полный текст, код — `notes[].code` (у строк по проекту — свой код; нет `notes` — строка по-прежнему полный текст); `cleanBlumWarning` убирает «на опытах сверки …» у любой причины плашки «оценка сверху» и не теряет причины после первой; `reports/blumDocxParts.ts` — таблица «№ — Вид — Замечание — Пояснение» и приложение «Пояснения к замечаниям» (Б после «Расчётных формул», иначе А); `generateXlsx.ts` — лист «Замечания»: № — Код — Вид — Замечание — Пояснение; XML — «Текст» несёт полный текст. Сторожа: `exportedDocs.guard.test.ts` ([0479], `blumNoteProblems`), `blumNotes.test.ts`, `xlsxInspect.testutil.ts` — графа «Код» листа — идентификаторы ответа, не текст для человека, её состав проверяется отдельно. Не затронуты: пояснительная записка (печатает `warnings` Блюма через `cleanBlumWarning`), XML.

- **Что убрано и что не сделано.** Кнопка «PDF (краткий)» (jsPDF без кириллицы) и мёртвые exportPDF/exportXLSX удалены вместе с пропсами ResultsTab; полноценный PDF — конвертация DOCX на сервере (отдельная задача). Десятичная запятая введена в текстовых DOCX (выше) и в тексте ячеек XLSX (числа в ячейках XLSX и XML не менялись); рамка проверена в LibreOffice, в Word не открывалась. Шифр «РР» принят как шифр разработчика (стандарт не задаёт шифр расчётной записки для ПД).
- **XML ЕЦПЭ 0.2-beta, статусы метода Блюма, п. 1.2 ПЗ ([0400-отчёты-блюм-xml]).** reports/generateXml.ts: версия схемы «0.2-beta»; имена с единицами «MmaxКНм_на_м», «QmaxКН_на_м», «ЖёсткостьКН_на_мм»; у метода Блюма «КЗаделки» (не «Kst»), «КоэффициентИспользования» (не «Утилизация»), «WmaxОценкаМм», «ДлинаСтенкиМ» = принятая длина (wall.wallLength) и «РекомендуемаяДлинаСтенкиМ» = totalLength; элементы «Этапы», «Связи» (supportsXml; у Блюма с пометкой «УчётВРасчёте» из blumInputMarks), у МКЭ «Предупреждения» (clientFemWarnings) и усилия связей по этапам со знаком (signed_force_axial_kN или знак по виду связи, растяжение +, сжатие −; атрибут «Растяжение»); версия схемы 0.4-beta ([0512-знак-усилий-связей]: знак усилий связей поменян, в «УсилиеСвязи» наибольшие сжатие и растяжение — модули). reports/blumStatus.ts — слова сопоставления для DOCX, XLSX и XML метода Блюма: blumStrengthStatus(η) («η не превышает 1,0» / «η превышает 1,0»), blumLengthStatus(принятая, расчётная), blumShortfalls(…) для вывода при несоблюдении; «ВЫПОЛНЯЕТСЯ», «ПРОХОДИТ», «СТАТУС» в отчётах Блюма не печатаются. reports/kzLabel.ts: deflectionGuideStatus(ok) — слово статуса перемещения, общее для exportedCheck() и XML МКЭ. generateDocx.ts: разделы записки Блюма нумерует счётчик + +sec (раньше номера были заданы числами); раздел о заделке — «Глубина заделки и длина шпунта» (BLUM_EMBEDMENT_TITLE). core/applicability.ts: строка «metal» у метода Блюма — partial. modules/documentation/core/designBasis.ts — предложение п. 1.2 ПЗ из свободного поля «Основание для разработки» (шаблон textTemplates.ts берёт переменную designBasisSentence). Спецификация: тексты ведомостей и позиций (positionMapper,

specificationGenerator, elementScheduleGenerator, materialSummaryGenerator) форматируются fmtNum / fmtRaw — десятичная запятая в XLSX. Тесты: reports/generateXml.test.ts (26), блок «[0400-отчёты-блюм-xml]» в exportedDocs.guard.test.ts, blumReportP1.test.ts, core/designBasis.test.ts. Харнес тестовой выгрузки (scripts/experiments/reports_test_1015/rerun.sh) печатает PDF из XLSX под именем «<имя>_таблицы.pdf» и сверяет, что под именем документа Word нет PDF со словом «Calc» в Creator.

- **Усилия в креплениях, допущения модели, подписи чертежа ([0401-отчёты-мкэ-пз-тэо-чертёж], 16.10.2026).** Числа ядра и ответов /api/calculate не менялись; в ответ /api/calculate/fem добавлено поле model_assumptions (клиент: types.ts FemModelAssumptions, services/femResults.ts). reports/supportForces.ts — усилия связей со знаком: forceSign / signedAxialKN / signedNormalKN (из core/supportForce.ts; растяжение +, сжатие −: анкер всегда +, распорка без полей знака в ответе — сжата, −; соглашение строительной механики с [0512-знак-усилий-связей], до того сжатие +, растяжение −), femSupportForceRows (по этапам и подэтапам: наибольшее сжатие и наибольшее растяжение распорки, усилие анкера; шаг из исходных данных), femSupportForceLines (текст «на один анкер, по оси» + нормальная составляющая + «на 1 м стенки» = нормальная / шаг), stageForceValueText (ячейка таблицы по этапам: «5,1 (растяжение)», «-63,1 (сжатие)»), supportForceUnitsNote() и SUPPORT_FORCE_UNITS_XML (легенда знака: растяжение «+», сжатие «−», усилие анкера положительно); dataCollector кладёт строки в CalculationSummary.supportForceLines (у Блюма — «Опора N: F = … кН/м (реакция на 1 м стенки)»), ПЗ п. 4.2, ТЭО и XML берут их оттуда. core/supportScheme.ts — консольная / анкерная / распорная / смешанная. reports/modelAssumptions.ts — строки таблицы «Параметры расчёта» (до [0413] — раздел «Расчётные допущения и исходные данные модели»; графа «Основание» — assumptionBasis(); отчёт МКЭ — раздел, ПЗ — подраздел, ExplanatoryNoteTable.weights/leftColumns). reports/supportDepth.ts — положение дна относительно связей вместо поля bottomToTieDistance (в расчёте не участвует). reports/numFormat.ts: fmtNum не печатает «-0,00», plural(). XML 0.3-beta: «УсилияВСвязях», «ДопущенияМодели», «РасчётнаяСхема». Том ТЭО: docxTeo.ts — Packer.toArrayBuffer вместо toBuffer (nodebuffer в браузере нет; docxTeo.browser.test.ts), programFootnote() — что выполнено в томе, волны, лёд и течения — только если введены. Чертёж: поля видов в мм (coordinateSystem.ts PLAN_PAD_MM, SECTION_PAD_MM, sectionPadLeft), масштаб подбирается по размерам видов без полей (planExtent / sectionExtent), анкеры на плане — проекция L·cos α (planAnchorReach), core/labelLayout.ts — оценка занимаемого места подписей, separateLabels (разведение), findLabelOverlaps и findOutsideFrame (проверки в тестах), core/dimensionLayout.ts — геометрия размера общая для SVG и DXF (число вертикального размера повёрнуто на 90°), core/drawingChecks.ts — сверка длины распорки с шириной котлована на плане (buildDrawingSheet возвращает warnings; те же замечания — в пакете документации); titleBlockFromInfo() — стадия и исполнители основной надписи из реквизитов комплекта. Листы тома в DOCX — нумерация с 1 и «Листов 1» (gostDoc.framedSection(pageStart), GostStampProps.totalSheets); текст тома — тоже с 1, «Листов» по закладке в конце текста ([0403]); у отчёта с рамкой — «Содержание» (reportLayout.contentsBlock()). Тесты: femSupportsPrint.test.ts, drawingLayout.test.ts, docxTeo.browser.test.ts, supportDepth.test.ts, supportScheme.test.ts, test_api_fem_model_assumptions.py.

- **Третья волна правок по заключению проверки ([0403-отчёты-третья-волна], 16.10.2026).** Числа ядра, ответы /api/calculate и /api/calculate/fem, hydroweb/ не менялись. quantityCalculator.ts: масса пояса — findBeam(BEAM_CATALOG, BEAM_STANDARDS, стандарт, профиль), стандарт сравнивается через core/beamSelect.ts: standardKey / canonicalStandard (регистр, пробелы, вид тире, латинские двойники кириллицы, год из двух цифр, псевдоним ГОСТ Р 57837-2017 → ГОСТ 35087-2024); WalerQuantity.massSource ('catalog' | 'area' | 'unknown') и walerMassRemark() — «масса по каталогу не найдена, оценка по площади сечения проекта» в ведомости, спецификации и записке; elementScheduleGenerator.ts — «Распорка N» / «Анкер N» вместо «Ярус N». blumAssumptions.ts: relabelSupportInWarning (внутренняя метка «S1» → «Распорка N» / «Анкер N» по включённым строкам), допущение о воде — одна строка; testing/internalPhrases.ts — шаблоны «настройк…», dewatered, kst, метка связи «S1». Штамп текста тома: gostDoc.TEXT_END_BOOKMARK, markTextEnd(), GostStampProps.totalSheetsBookmark. Чертёж: core/labelLayout.ts — линии анкеров и распорок — препятствия для подписей (segmentHitsBox, findDimensionCrossings), core/dimText.ts (mmText — размеры в мм), core/planNotes.ts (примечания под планом: условные размеры, длина распорки по расчёту; differingStruts), DrawingInput.planSizeAssumed, buildDrawingSheet(…, planSizeAssumed), DrawingTab помнит, вводил ли пользователь размеры. Формула прочности: steelResistance.ts — metalCheckAxialN, strengthFormulaText, strengthAxialNote (ПЗ п. 4.3), metalMarginSubstitution (формулы раздела «Проверки предельных состояний» отчёта МКЭ: подстановка Ry, W, Mmax и при продольной силе N, A; константа FEM_METAL_FORMULA_AXIAL удалена в [0413]). Слова результата сопоставления МКЭ: kzLabel.ts — metalGuideStatus, supportForceGuideStatus, utilizationGuideStatus, convergenceGuideStatus, femCheckStatusWord (exportedCheck, ПЗ, XML). numFormat.ts: typographicMinus — «−» (U+2212) при записи текста в Word (gostDoc.txt / cellPara, gostPlain) и в текстовые ячейки XLSX (xlsxStyle.put). Тесты: блок «[0403]» в exportedDocs.guard.test.ts, drawingThirdWave.test.ts, DrawingTab.test.tsx, elementScheduleGenerator.test.ts, strengthFormula.test.ts, gostStamp.test.ts, beamSelect.test.ts, numFormat.test.ts.
- **Четвёртая волна правок по заключению проверки ([0404-отчёты-четвёртая-волна], 16.10.2026).** Числа ядра, ответы /api/calculate и /api/calculate/fem, hydroweb/ не менялись (сравнение ответов на запросах П1…П3б с a857f8f4: в записи четвёртой волны стояло «6659 числовых значений совпали побитно» — неточно, в сравнение входило время счёта elapsed_s; верно, по повтору в седьмой волне: из 6707 значений в общих ключах совпали 6706). Чертёж: core/labelLayout.ts — dimensionSegments(): выносные и размерная линии — препятствия для подписей (fixedObstacles, owner — размер, которому принадлежат линии: его число рядом с ними допустимо), TEXT_DESCENT (рамка текста включает 0,2 высоты под базовой линией, у чисел без знаков — нет), bubbleCoversSegment() и проверка findLabelOverlaps — кружок оси или марки разреза не закрывает связь; separateLabels() ищет место шагом 0,4 высоты шрифта и сдвигом поперёк; core/sectionGenerator.ts — dimObstacles (линии размеров как препятствия для названий грунтов и подписи УГВ), подпись УГВ ищет просвет шагом 0,4 мм, «▽ 0,000» — anchor 'start' правее выносной линии, метка q=… не заходит на выносную линию размера ширины; core/planGenerator.ts — leftOff (кружки слева за концами анкеров, BUBBLE_GAP 3 мм), anchorRowAt(), линия оси кончается у конца анкера; core/dimensionLayout.ts — verticalTextAlong(); core/svgGeometry.testutil.ts — parseSvgGeometry(), svgCrossings(): проверка по готовому SVG (подпись на линии, подпись

на подписи, кружок на линии), та же проверка независимо — check_outputs.py тестовой выгрузки. Документы: steelResistance.ts — metalMarginFormulaText() (формула запаса по металлу по наличию N; sectionFactsText() удалена в [0413]); kzLabel.ts — NO_NORM_CONCLUSION (одна оговорка вывода для МКЭ, Блюма и записки; femConclusionText() удалена в [0413] — вывод МКЭ строит buildFemConclusions, Блюма — buildConclusions в generateDocx.ts); clientWarnings.ts — relabelFemSupports(), clientFemWarnings(list, ctx) принимает SupportContext (отчёт, XML, записка); xlsxStyle.ts — minusNumFmt(), числовая ячейка получает формат с секцией отрицательных «"−"»; generateXml.ts — machineStatus() (слово статуса с точкой); quantityCalculator.ts — WalerQuantity.strutNumber (сквозной номер распорки) и catalogMismatch (фраза записки п. 3.3); testing/internalPhrases.ts — findTierWord(), findNestedParentheses(), шаблоны меток связей без кавычек и указаний «выберите / нажмите / перейдите / откройте»; check_outputs.py: R_сеч, «Q в нижней точке», слова вердикта, вложенные скобки, «ярус», метки, ASCII-минус в печати XLSX, SVG. Тесты: reports/fourthWave.test.ts, modules/drawing/core/drawingFourthWave.test.ts, ExportPanel.test.tsx, блок «[0404]» в exportedDocs.guard.test.ts, elementScheduleGenerator.test.ts, strengthVerdict.test.ts.

- **Пятая волна правок по заключению проверки ([0405-отчёты-пятая-волна], 16.10.2026).** Числа ядра, ответы /api/calculate и /api/calculate/fem, hydroweb/ не менялись (сравнение ответов на запросах П1…П3б с a857f8f4: в записи пятой волны стояло «6739 числовых значений совпали побитно» — неточно, в сравнение входили время счёта elapsed_s и значения 32 добавленных ключей; верно, по повтору в седьмой волне: из 6707 значений в общих ключах совпали 6706). Чертёж: core/planGenerator.ts — sectionCutRow() и linkRows() (линия разреза проводится по середине просвета между рядами анкеров и распорок; торцы — тоже ряды), оси 1 и 2 не идут поверх связей; core/labelLayout.ts — ширина рамки подписи по textMetrics.ts (таблица ширин знаков Arial вместо 0,55 высоты на знак), TEXT_ASCENT = 0,8 и TEXT_DESCENT = 0,2, запас от оси линии inkClear() — половина штриха и INK_GAP 0,1 мм (вместо допуска 0,15 мм внутрь), рамки подписей и чисел без допуска, препятствия — стенки (HW-WALL), плитки концов распорок, оси, стрелки пригрузки и линия разреза, labelField() — поле подписей (одни препятствия у генератора, разведения и проверки), dimensionMisfit() и placeDimension() — подбор отступа и положения числа размера, separateLabels() — поиск по сетке вокруг исходной точки; core/stripLabels.ts — placeStripLabels() (подпись воды у линии с зазором от 0,3 мм, названия слоёв по оставшемуся месту; в шестой волне заменена placeWaterLabel() и placeSoilLabels()) и stripMinWidthMm() (ширина, нужная самому длинному названию слоя; поправка шестой волны: в выбор масштаба она больше не входит — расширение полосы в раскладке листа вывело вид за рамку ещё на 10 листах); core/sectionGenerator.ts — подписи ставятся в конце по полю, профиль ограждения — в заделке или под стенкой, размеры бермы — после остальной графики; core/dimensionLayout.ts — DIM_EXT_WEIGHT и DIM_LINE_WEIGHT (одни числа у SVG и у модели), shortDimensionAlongs(), beyondDimensionAlongs(); core/svgGeometry.testutil.ts — linkCoverings() (связь закрыта параллельной линией, нарисованной позже), svgCrossings() с толщиной штриха, boxToSegmentDistance(). Документы: modules/documentation/core/userText.ts — userReference() и flattenParentheses() (ссылка пользователя на изыскания в записке — отдельным предложением «Источник данных об изысканиях: …», вложенные скобки в ней — квадратные); limitStates.ts, supportChecks.ts, applicability.ts, kzLabel.ts, generateDocx.ts — слова результатов без

«выполняются / требуется / не выполнено» (примечание к п. 6.1.7 — рекомендация); generateXml.ts — XML_NUMBER_RULE и XML_HUMAN_TEXT_ATTRS (значения, условия и статусы — точка, текст для людей — запятая; атрибут корня ФорматЧисел); testing/internalPhrases.ts — новые шаблоны сторожа. Тесты: modules/drawing/core/drawingFifthWave.test.ts (42: связи не закрыты, сетка глубин воды, Л5-УМ на А4 книжной, TEXT_DESCENT, полоса грунта, матрица 4 проекта × А4–А1 × 2 ориентации × 6 размеров — матрица по 192 листам, перечень 26 листов вне рамки, 240 листов с вариациями слоя и воды; после шестой волны перечень заменён ратчетом, а наложения названий слоёв допускаются на 9 листах сетки), reports/fifthWave.test.ts (19: слова результатов, сторож «не слеп» и «не ложен», правило чисел в XML, условие перемещения), modules/documentation/core/userText.test.ts (6), supportChecks.test.ts (17, +4). Проверка выгрузок харнеса (scripts/experiments/reports_test_1015/check_outputs.py; с шестой волны харнес лежит в git): исключения перечислены в заголовке файла, вырезание ссылки на изыскания перед проверкой скобок убрано; добавлены закрытие связей, расстояние до линии воды и независимый измеритель по контурам знаков Arial. Что проверено мутацией, что нет, и перечень известных ограничений — в записи [0405] журнала.

- **Шестая волна правок по заключению проверки ([0406-отчёты-шестая-волна], 16.10.2026).** Числа ядра, ответы /api/calculate и /api/calculate/fem, hydroweb/ не менялись (из 6707 значений ответов на запросах П1…П3б в общих ключах (6659 числовых и 48 булевых) совпали 6706, различается elapsed_s — время счёта; в записи шестой волны стояло «совпали с a857f8f4 побитно» — неточно; различаются тексты warnings[0] в шести ответах и добавлены 32 ключа). Чертёж: core/coordinateSystem.ts — calculateLayout() выбирает масштаб без ширины названий грунта, а свободную ширину листа отдаёт сначала полосе грунта (toStrip: до sectionExtent().leftMinMm), остальное делит между планом и разрезом прежним правилом; core/stripLabels.ts — widenedStripMm() (основная полоса плюс недостающее до названия, но не больше свободной ширины; в шестой волне generateSection считал её по ширине вида минус нужная — по запасу вида, а не по месту до рамки листа; в седьмой волне — по месту до рамки, см. ниже), placeWaterLabel() (приоритет перед названиями, место у линии с зазором от 0,3 мм, поиск вдоль линии шагом 2 мм, не выше поверхности — StripSpec.surfaceY, стенка — препятствие) и placeSoilLabels() (первое свободное место, иначе наименее плохое по LabelField.hits(): подписи и числа хуже линий; название длиннее полосы — стенка не препятствие и место вплотную к левому краю штриховки, признак DrawingText.overWall); core/sectionGenerator.ts — порядок подписей «вода, профиль, названия», параметр frameRight (правая граница рамки листа: отметка «▽ 0,000» слева от оси правой стенки и примечания о берме — внутри рамки, если справа не помещаются); core/labelLayout.ts — LabelField.isFree(box, asLine, overWall), hits(), признак Fixed.wall, separateLabels() обрабатывает подпись воды первой и не считает стенку препятствием для подписи с overWall; export/svgExporter.ts — рамка и основная надпись в группах g.form (renderBody()); core/svgGeometry.testutil.ts — parseSvgGeometry() без формы листа и со штриховкой, frameViolations() и isFrameViolation() (за краем листа, на рамке, за рамкой, в основной надписи), svgCrossings() без исключений для зоны надписи, linkCoverings(svg, eps, touching) — касание краями штриха (просвет 0,000 мм) — нарушение. Известные ограничения записаны и в тестах: 14 листов вне рамки (А4 при 1:1000, раскладка «рядом»), названия слоёв на стенке на 34 листах (подписи на подписях и линиях — на 9),

касание линии разреза рядов распорок на 1:1000 (20 листов), совпадение распорок на осях 1 и 2 с контуром и торцевой стенкой (4 раза на проект, в SVG и DXF распорка записана позже). Тесты: modules/drawing/core/drawingSixthWave.test.ts (38: масштаб не зависит от названий, ратчет листов вне рамки, названия на стенке, поле подписей, приоритет воды, образцы с четырьмя слоями, глубины воды 0,5…12 м, рамка листа в измерителе, касание связей, совпадения распорок, комментарии), reports/exportedDocs.guard.test.ts (+1: слова вердикта без различия регистра), core/limitStates.test.ts (+1). Харнес scripts/experiments/reports_test_1015 — в git: check_outputs.py проверяет XML и текст PDF (pdftotext) теми же словами и скобками, что DOCX и XLSX, пересечение подписи с линией своей группы, касание связей, подпись воды (нет подписи или линии — ошибка), шрифты Arial и fontTools обязательны; check_outputs_selftest.py — самопроверка на заведомо плохой выгрузке (9 мутаций харнеса ловятся); sheet_grid.test.ts и sheet_grid.py — пять показателей на сетке листов для любого коммита (распаковать frontend коммита, тот же измеритель); пути пользователя заменены переменными HW_OUT и HW_PY. Что проверено мутацией, что нет, таблицы показателей по коммитам и перечень известных ограничений — в записи [0406] журнала.

- **Седьмая волна правок по заключению проверки ([0407-отчёты-седьмая-волна], 17.10.2026).** Числа ядра, ответы /api/calculate и /api/calculate/fem, hydroweb/core не менялись (git diff a857f8f4..HEAD -- hydroweb/core пуст; в ответах на запросах П1…П3б из 6707 значений в общих ключах совпали 6706, различается elapsed_s — время счёта 0,3 против 0,2 с; различаются тексты warnings[0] в шести ответах, добавлены 32 ключа — как в шестой волне). Чертёж: core/sectionGenerator.ts — generateSection() строит разрез дважды: пробный проход buildSection(…, probe = true) с основной полосой без названий слоёв и подписи воды даёт правый край всего, что стоит правее полосы (core/labelLayout.ts, rightmostEdge(): линии с половиной штриха, рамки текстов и чисел размеров, штриховка; отметка «▽ 0,000», примечания о берме, названия слоёв и подпись воды не в счёт), и widenedStripMm() получает всё место до внутренней рамки листа за вычетом FRAME_CLEAR_MM = 1 мм (раньше — запас вида из раскладки); рамка листа неизвестна (frameRight = ∞) — прежний запас вида. core/stripLabels.ts — soilCandidates(): места названия слоя только в полосе своего слоя (рамка целиком внутри, затем положения «центр рамки в слое» для слоя тоньше рамки; слой, уходящий ниже стенки, — вся полоса штриховки, StripLayer.yNear предпочитает место до низа стенки плюс метр), solveItems() — точный выбор по цепочке слоёв динамическим программированием (цена кандидата: задетые подписи и числа по 1000, линии по 1; подписи друг на друге и на неподвижных рамках по 1000; до 220 кандидатов на слой) и уточнение парами через слой, placeStripLabels() — подпись воды (первое свободное место у линии, затем из свободных не дальше WATER_JOINT_GAP_MAX = 4 мм выбирается то, при котором названия слоёв задевают меньше всего) и названия слоёв вместе; placeWaterLabel() и placeSoilLabels() остались для отдельного вызова. core/labelLayout.ts — DrawingText.layerBand (core/types.ts): разведение подписей листа не уводит название из полосы своего слоя (центр рамки с запасом BAND_MARGIN = 0,25 мм, centreInBand()) и не сдвигает его по горизонтали и по сетке вокруг точки. Тесты: modules/drawing/core/drawingSeventhWave.test.ts (50: ширина полосы по месту до рамки, 9 листов и 25 оставшихся поимённо, инвариант «вид не выходит за рамку» на 480 значениях, названия в своих слоях на 1272 листах сеток, правила размещения, подпись воды, подписи на подписях поимённо, поправки к комментариям

шестой волны, 14 тестов текстов после независимой проверки — остатки ложных фраз в журнале, книгах и комментариях тестов, исключения харнеса, сетки oos_water и oos_thin), drawingGrids.testutil.ts (сетки измерителя), svgGeometry.testutil.ts — layerNameViolations(); тесты четвёртой, пятой, шестой волн и drawingLayout.test.ts — «наложения только с участием названия слоя»: фильтр по тексту сообщения пропускает и название слоя на подписи воды, отметке или профиле (прежний порог «наложений нет» ослаблен; после независимой проверки названия и комментарии этих тестов исправлены, логика не менялась). Харнес scripts/experiments/reports_test_1015: sheet_grid.py — шестой показатель, вода по всем подписям и линиям, сетки water_a4, layers5, longnames; grid_compare.py — несколько коммитов на всех сетках одной таблицей; check_outputs.py — тексты SVG, DXF и PDF чертежа (слова, скобки, ASCII-минус вне «▽ -N,NNN» и «▽ УГВ -N,N м»), состав выгрузки (52 файла, 32 PDF; код возврата 2 при неполном прогоне); самопроверки check_outputs_selftest.py и sheet_grid_selftest.py (18 мутаций харнеса ловятся); после независимой проверки в sheet_grid.test.ts добавлены сетки проверяющего oos_water (144 листа) и oos_thin (96 листов; grid_compare.py --sets oos_water oos_thin), в заголовке check_outputs.py — исключение: показатель «название в своём слое» на выгруженных чертежах не считается. Поправки после независимой проверки седьмой волны (подпись на подписи и вне выборки, ухудшение подписи воды на отдельных листах, перебор положений, ослабление тестов) — в записи [0407] журнала. Что проверено мутацией, таблицы показателей по коммитам и известные ограничения — в записи [0407] журнала.

- **Поправки после независимой проверки слияния ([0408-слияние-выгрузок-в-release], 17.10.2026).** Ядро hydroweb/core не менялось. API: в model_assumptions ответа /api/calculate/fem добавлено поле strut_tension (режим горизонтальной распорки, который принял расчёт: поле запроса strut_tension, без него — двусторонняя связь, _resolve_strut_tension); числовые поля ответов не менялись. Режим распорки в документах: reports/strutTensionMode.ts — порядок источников: model_assumptions.strut_tension → усилия распорок со знаком (signed_force_kN есть только у двусторонней) → wall.strutTension → по умолчанию двусторонняя связь; строка «Работа распорок» (reports/modelAssumptions.ts) берётся оттуда, наклонная распорка (раскос) названа отдельно. Способ учёта бермы (core/bermGeometry.ts, BermAccounting): явный method; иначе по ответу расчёта (замечание этапа «Этап N: берма с отпором (клин Кулона)» → с отпором, его нет → пригрузка на дне; замечание об откате с названием бермы → fallback); без расчёта — выбор в проекте, иначе «с отпором». Замечания release в документах клиента: api/main.py (_pioneer_mode_notes, срез «по силам» при слое с ф = 0) больше не печатает pioneer.accounting, разрешённые ветви названы словами; замечание ядра о марке стали (core/input_notes.py, «задайте sigma_yield_MPa …») клиент заменяет в reports/clientWarnings.ts; сторож testing/internalPhrases.ts дополнен шаблонами HW_…, pioneer…., accounting =; дословные тексты — testing/releaseWarnings.json (равенство с сервером — test_client_warning_texts.py). Политика метода Блюма в выгрузках: reports/blumPolicyDocs.ts — условие то же, что на экране (results.policy есть: прогиб и Кз Блюма как результат не печатаются, класс Б — «оценка сверху»; ответ без policy — учебный режим legacy_blum_policy, прежние числа, как на экране). Применено в generateDocx.ts (таблица результатов, ВЫВОД, таблица по этапам, раздел заделки, выводы, методика, приложение), blumDocxParts.ts, blumAssumptions.ts, generateXlsx.ts, generateXml.ts (атрибуты КЗаделки, WmaxОценкаМм и значение оценки перемещения опущены, версия

схемы не менялась; добавлены ПрогибИКзЗаделки, СтатусРезультата), CalculationSummary.blumPolicyHidden и blumUpperBound (explanatoryNoteGenerator: {deflectionLines}, {kzLine}, {upperBoundNote}; docxExplanatoryNote; teoVariants.calcFromSummary и teoContent), validationRules.ts, core/applicability.ts. Харнес scripts/experiments/reports_test_1015: проекты П4 (слой с ϕ = 0, срез «по силам») и П5 (марка стали вне перечня), 78 файлов и 48 PDF; check_outputs.py сверяет режим распорки и способ учёта бермы в документах с запросом и ответом, ищет имена кода и сервера; самопроверка и мутации — в README харнеса. Тесты: strutTensionMode.test.ts, releaseWarnings.test.ts, blumPolicyDocs.test.ts, bermGeometry.test.ts, teoFinderTexts.test.ts, ResultsTab.deformations.test.tsx, test_api_fem_model_assumptions.py, test_client_warning_texts.py, test_books_texts.py.

- **Поправки по повторной проверке ([0408-слияние-выгрузок-в-release], 17.10.2026, D8–D14).** Ядро hydroweb/core не менялось. Способ учёта бермы (core/bermGeometry.ts): bermActsAt повторяет правила alive_berms ядра (берма появилась не позже этапа, не снята до него, верх выше дна этапа, основание не нулевое); замечание «Этап N: берма с отпором (клин Кулона)» сервер пишет на этапе N, где берма действует и модель с отпором применена (core/fem/staging.py, cur_stage; на этапе, где берма действует, но модель откатилась на «пригрузку на дне», стоит замечание об откате, а при явной «пригрузке на дне» замечаний о способе нет вовсе), поэтому клиент ищет его на этапе действия, а не на этапе появления (иначе берма с этапом появления 3 и верхом на дне этапа 3 печаталась «пригрузкой на дне» при замечании «с отпором» в том же документе); строка таблицы берм несёт этап начала действия (actStage), дно и основание этого этапа, в XML — атрибут ЭтапЗадан, когда заданный этап иной. Названия берм в замечаниях расчёта (core/berm_removal.py, откат на «пригрузку на дне»: идентификатор из проекта — случайная строка) клиент заменяет номером бермы (reports/clientWarnings.ts, relabelBerms; SupportContext.berms передан в отчёт МКЭ, XML и пояснительную записку). Пояснительная записка из результатов метода Блюма очищает замечания теми же cleanBlumWarning и relabelSupportInWarning, что отчёт Блюма (modules/documentation/core/dataCollector.ts). Книга пользователя: итоговые выводы и записка метода Блюма разделены на ветви «при политике» и «учебный режим» по фактическим выгрузкам (П1 класса Б, П3б в учебном режиме). Харнес scripts/experiments/reports_test_1015: проект П3в (берма задана на этапе 3), 7 проектов, 91 файл и 56 PDF; проверка «БЕРМЫ ↔ ОТВЕТ» (check_outputs.py: berm_consistency_findings) сверяла таблицу берм, перечень «Бермы по этапам» и XML с запросом и замечаниями ответа независимо от умолчаний и кода клиента, но для ОДНОЙ бермы и с одним способом на все этапы (проект с несколькими бермами давал срабатывание «сверка не поддерживается»; с третьей проверки слияния — по этапам и для нескольких берм, см. следующий абзац); шаг 3 (chrome_docs.sh) строит документы по одному проекту и повторяет не построенный не более двух раз с явным сообщением, проверка числа файлов не ослаблена. Тесты: bermGeometry.test.ts, bermDocs.test.ts, releaseWarnings.test.ts, bookBlumClaims.test.ts (цитаты книги против выгрузок двух режимов), explanatoryNote.blumWarnings.test.ts, test_fem_berm_note_stage.py, test_books_texts.py (пороги класса А и условия классов Б и В сверяются с кодом core/blum_policy.py).
- **Поправки по третьей проверке слияния ([0408-слияние-выгрузок-в-release], 17.10.2026, D15–D21).** Ядро hydroweb/core не менялось. Клиент (frontend/src). Замечания о бермах

(reports/clientWarnings.ts, relabelBerms): список названий «Бермы «В1», «В2», этап 3:» больше не забирает запятую после последнего названия (было «Бермы 1, 2этап 3:»); тексты берутся у настоящего сервера — hydroweb/tests/test_berm_notes_server_formats.py получает по строке на каждый из 10 форматов замечаний о бермах (core/berm_removal.py — три, fallback_note в core/fem/berm_wedge.py — четыре вида названия и этапа, narrow_surcharge_berm_notes, wedge_berm_note, оговорка метода), сверяет их с frontend/src/testing/bermNotesServer.json и bermStageNotes.json и падает при новом формате; клиентские тесты — reports/bermNotesFormats.test.ts, bermStageAccounting.test.ts. Три указания по интерфейсу из core/berm_removal.py («Укажите этап снятия …», «Добавьте этап или укажите номер этапа …», «… укажите этап снятия: иначе …») заменяются на клиенте (ENGINEERING_REWRITES) утверждениями о требованиях расчёта; сторож testing/internalPhrases.ts и UI_VERB харнеса ловят «снимите» и «отметьте» (и «укажите», «задайте» в харнесе); «добавьте или усильте крепление» — инженерная рекомендация причины остановки расчёта, не указание по интерфейсу. core/bermGeometry.ts: последний этап в причине «берма не действует» не выходит за таблицу этапов (min(removeStage − 1, последний этап таблицы)); при пустой таблице этапов (запрос несёт один этап «Финал» с глубиной стенки) берма действует на этапе 1 (effectiveStages, четвёртый аргумент bermInputRows); строка берм хранит этапы действия (actingStages) и способ учёта по этапам (accountingByStage: явная «пригрузка» — она; замечание об откате этого этапа — откат; замечание «Этап N: берма с отпором» — «с отпором»; иначе «пригрузка на дне»); графа «Способ учёта» (bermAccountingText) — «на этапе 2: с отпором (клин Кулона); на этапах 3–4: пригрузка на дне (вне области применимости модели с отпором)», если способ различается по этапам, иначе одно название; перечень «Бермы по этапам» берёт способ данного этапа; в XML — элементы УчётНаЭтапе. Харнес scripts/experiments/reports_test_1015: berm_oracle.py — эталон берм (этапы действия по правилам alive_berms, способ по этапам по замечаниям ответа, ожидаемые тексты; не читает код клиента), проверка «БЕРМЫ ↔ ОТВЕТ» (check_outputs.py: berm_consistency_findings) сверяет по каждой берме и каждому этапу, в том числе при нескольких бермах (3 DOCX, XML, перечень в двух отчётах и их PDF); генеративная проверка berm_gen.sh (шаг 6а rerun.sh, отдельная команда): 16 закреплённых случаев и 300 случайных проектов с фиксированным seed, запросы тем же buildFemRequest, сервер через TestClient, документы теми же генераторами, сверка с эталоном по каждому этапу (таблицы, причина, перечень, примечание таблицы этапов, XML по этапам, номера берм, склейки, идентификаторы, указания по интерфейсу, этапы в пределах таблицы), отчёт с покрытием в берм_ген/отчёт.md; самопроверки berm_gen_check_selftest.py и check_outputs_selftest.py. Результат: до исправлений (клиент a9bfb796, тот же сервер) 2099 расхождений в 222 вариантах из 316, после — 0 (без профиля 316 вариантов, с профилем 292 ответа 200 и 24 отказа сервера 422). Не покрыто генератором: произвольные номера этапов таблицы, пионерный срез, «свободно стоящая стенка», подсыпка тыла, водоносные слои, PDF-вёрстка длинной графы способа по этапам.

Клиент: отчёт МКЭ и его приложения ([0516-отчёт-мкэ-приложения], 27.10.2026)

Отчёт МКЭ с вкладки «Результаты» (`reports/generateDocx.ts`, `buildFemDocxReportDocument`) после «Замечаний расчёта» получил три приложения: «Эпюры по этапам», «Коэффициенты

давления грунта» и «Сертификат соответствия». Порядок 14 разделов отчёта сохранён; ядро, API и интерфейс не менялись. Отчёт Блюма, пояснительная записка и том ТЭО не менялись.

- **Буквы и подписи.** `planFemAppendices` (`reports/femAppendices.ts`) раздаёт буквы подряд без пропусков: «Замечания расчёта» (есть, когда у замечаний есть пояснения длиннее строки), «Эпюры по этапам», «Коэффициенты давления грунта», «Сертификат соответствия»; при всех четырёх это А, Б, В и Г, без замечаний с пояснениями А, Б и В. Подписи «Рисунок Б.N» и «Таблица В.1», «Таблица Г.1» строят `appendixFigureCaption` и `appendixTableCaption` в `reports/gostDoc.ts` (нумерация приложения отдельная, с буквой: ГОСТ Р 2.105-2019, пп. 6.9.3 и 6.8.2); таблице предшествует абзац «См. таблицу В.1.». Ссылки в тексте: в методике («Коэффициенты давления грунта по слоям — в приложении В», «Сведения о сертификате и его скан — в приложении Г»), в разделе об эпюрах («Эпюры каждого этапа — в приложении Б»); содержание перечисляет приложения (закладки `appendixHeading`).
- **Эпюры по этапам.** `buildStageEpuresAppendix`; данные собирает `reports/femStageEpures.ts`. Для каждой записи ответа (этап и подэтап «k.F0» перед этапом k) четыре рисунка в сетке 2 × 2 на одном листе: давление (тыл слева и лицо справа с предельными значениями штрихом), M, Q по `shear_elements` и w. Под рисунками строка наибольших значений по модулю с отметками z: давление тыла и лица по узлам эпюры, Mmax, Qmax и wmax скаляры записи этапа (те же числа, что в таблице «Результаты по этапам»), глубина узел наибольшего значения эпюры того же этапа. Каждый этап с новой страницы. Рисунки строит тот же построитель, что рисунки раздела об эпюрах (`femDiagrams.ts`). Нет `stage_diagrams` у записей (результаты старого сервера): приложения нет; нет холста: остаются подписи и строка наибольших значений.
- **Коэффициенты давления грунта.** «Таблица В.1» по включённым слоям: номер, ИГЭ, φ, c, δ, метод (Ренкин или Кулон), Ka, Kp, K0. Числа берутся из поля `coefficients` ответа расчёта (его заполняет ядро, клиент их не считает); нет поля (сохранённые результаты старого сервера): тот же запрос `POST /api/coefficients` с тем же вводом, что у таблицы на вкладке «Геология» (`buildCoefficientsRequest`, `loadCoefficients`); не получены или число строк не равно числу слоёв: одна строка вместо таблицы, чисел «по догадке» нет. δ та же, что уходит в запрос (`layerDeltaDeg` в `services/projectToRequest.ts`, одно место вместо четырёх копий). Метод, δ и формулы это готовые тексты модуля отчёта (`lateralMethodText`: СП 22.13330.2016, ф. 9.8, 9.12, пп. 9.22, 9.23; `k0FormulaCaption`: п. 9.20); новых ссылок на нормы не добавлено.
- **Сертификат соответствия.** «Таблица Г.1» со строками дословно по странице `frontend/public/docs/certificate.html` (`reports/certificateInfo.ts`: `CERTIFICATE_TABLE_ROWS`, `CERTIFICATE_NATURE`), оговорка о характере сертификата (добровольная сертификация, не заменяет проверку проектных решений ответственным инженером, не государственная экспертиза) и два скана (рисунки Г.1 и Г.2), по странице на скан. Сканы это те же файлы, что публикует сайт (`public/docs/certificate/page-1.jpg`, `page-2.jpg`): отчёт берёт их запросом к сайту при формировании (`loadCertificateScans`; ответ без JPEG, например страница-заглушка, отвергается, размер читается из маркера SOF). Не загрузились: остаются таблица, оговорка и адрес страницы сертификата.
- **Тесты.** `reports/femAppendices.test.ts` (28 тестов: буквы, данные этапов на снимках сервера femStageDiagrams.json и femSubstage.json, сетки 2 × 2 и разрывы страниц, подписи

Б.N, таблицы В.1 и Г.1, запасной запрос, отсутствие холста, сканов и stage_diagrams), `reports/certificateInfo.test.ts` (12: строки таблицы и оговорка есть на странице сертификата, сканы те же файлы), сторож выгрузок `exportedDocs.guard.test.ts` (185): образцы отчёта МКЭ с приложениями и проверки букв, ссылок в тексте, подписей и содержания; правило «сертификат: две строки» относится к телу документа, приложение «Сертификат соответствия» исключено.

Клиент: подраздел «Анкерное крепление» записки, спецификация и отчёт МКЭ ([0529-анкеры-в-записке-и-выгрузках], 28.10.2026)

- **Подраздел записки.** `modules/documentation/core/anchorSection.ts` (`buildAnchorSection`, `ANCHOR_SECTION_TITLE`): шаблонный ключ `3.6` в `textTemplates.ts`, условие `hasAnchors`, место сразу за «Системой крепления», сквозная нумерация `renumberSections`; `ExplanatoryNoteSection.extraTables` — несколько таблиц в подразделе (DOCX-экспорт и предпросмотр). Таблицы: «Параметры анкеров» (глубина от верха стенки, α, шаг, Lсв, Lк, F0, жёсткость C, N_пред); «Тяги анкеров» (обозначение, стандарт, число тяг n, площадь всех тяг, нормативное усилие по каталогу тяг: разрывное для каната, при пределе текучести для стержня; если тяга не выбрана, таблицы нет, и текст говорит, что сечение тяги не принималось); «Расчётные усилия в анкерах по этапам» (строка на этап и подэтап «k.F0», столбец на анкер, строка «Наибольшее» с этапом; осевое усилие на один анкер, растяжение «+», из `stages[].anchor_forces`); «Проверки анкеров» (прочность тяги N ⩽ N_пред из `support_checks` ответа МКЭ, СП 381, п. 7.3.9, K из ответа, и все проверки п. 7.3 из ответа `POST /api/checks/anchors` строками как есть: название, пункт, расчётное, предельное, запас по п. 7.3.20, «условие соблюдено / не соблюдено / не выполнялась» с причиной сервера). Без результата проверки корня или с устаревшим стоит `ANCHOR_ROOT_NOT_CHECKED` («Несущая способность корня по грунту не проверялась: …»); при расчёте методом Блюма усилий по этапам и сопоставления N с N_пред нет, об этом сказано, а не выдано за выполненное. Статусов «выполняется / не выполняется» в таблицах нет (сторож [0403], п. 7).
- **Восемь предельных состояний анкеров ([0549-применимость-анкеры-статусы], 29.10.2026).** `frontend/src/core/anchorLimitStates.ts` — один источник статусов для таблицы «Границы применения» (`applicability.ts`) и подраздела записки: `ANCHOR_LIMIT_STATES` (номер в перечне п. 7.3.9, id, название, адрес нормы без выдуманных номеров, `evaluate` по фактам), `anchorLimitFacts` (включённые анкеры, способ определения R_a;k и R_t;d у анкеров, проверки N ⩽ N_пред по анкерам из ответа МКЭ, Км стенки, актуальный ответ проверки корня), `anchorLimitStateLines` («N. Название (адрес нормы) — проверено / проверено частично / не проверяется: пояснение.»). Прежняя безусловная строка «срыв корня, выдёргивание тяги и общая устойчивость с анкерами не проверялись» была неверна после [0529]: срыв корня (п. 7.3.20, прил. Л) и прочность тяги при заданном R_t;d (п. 7.3.21) проверяются и идут в записку. Статусы: 1 оголовок, 3 выдёргивание тяги, 6 ослабление натяжения — «не выполнялась: метода в СП 381 нет»; 2 срыв по контакту корня — «выполнена» при актуальном ответе проверки корня по всем анкерам, «частично» по части, иначе «не выполнялась» («нет данных» либо «результата проверки корня в документе нет»); 4 разрыв тяги — «выполнена» по R_t;d, «частично» по части анкеров либо только N ⩽ N_пред проектировщика из МКЭ; 5 общая устойчивость вместе с анкерами — всегда «не выполнялась» (результат /api/checks/stability в документы

не передаётся, как у general_stability; частичный характер проверки — анкеры как внешние силы только при включённом учёте связей — назван в пояснении; «частично» появится, когда результат устойчивости попадёт в документы); 7 разрушение конструкций анкерными усилиями — «частично» при расчёте МКЭ с Км стенки (пояс и узел головы не проверяются), иначе «не выполнялась»; 8 деформации групп анкеров — «частично» (положение корней), если в документе есть ответ проверки корня с проверками п. 7.3.14 или 7.3.8. Ответ проверки корня несёт только записка (buildApplicabilitySection(data, anchorChecks)); отчёты МКЭ и Блюма и XML его не печатают, и для них строка 2 говорит «результата проверки корня в документе нет: не запускалась или проект изменён». Пояснение — до 20 слов, у «частично» до 12; в блоке «Предельные состояния грунтовых анкеров, СП 381.1325800.2018, п. 7.3.9:» записки он заменил прежнюю строку «Не проверялись: …». Сервер: NOT_CHECKED_LIMIT_STATES в `anchor_capacity.py` описывает ЭТУ проверку, а не проект (причина у каждого состояния, состояния 7 и 8 добавлены; срыв корня и разрыв тяги не входят), поле not_checked ответа на месте (string[]), панель «Несущая способность анкеров» печатает его в подсказке. Не сделано: чек-лист 7.3.14 е), ж) и ситуации 7.3.10 в «Применимости», общая устойчивость с анкерами по двум семействам поверхностей (п. 6.4.5), результат устойчивости в документы; строка time_effects («Ползучесть, релаксация усилий в анкерах…») оставлена и частично дублирует состояние 6. Тесты: `core/applicability.test.ts` (25), `anchorSection.test.ts` (22; порядок и состав строк, срыв корня во всех вариантах, разрыв тяги, общая устойчивость, стенка и группы, длины пояснений, XML), `test_anchor_capacity.py` (список «не проверялось» с причинами), `exportedDocs.guard.test.ts` (строки состояний анкеров внесены в перечень строк «Границ применения», где пояснение печатается при любом статусе; тест восьми строк по порядку и записки с проверкой корня и без неё).

- **Проверка корня доходит до записки.** `AnchorCapacityPanel` сообщает ответ сервера вместе с телом запроса (`AnchorChecksSnapshot { key, response }`; `onResult` → `ChecksContext.onAnchorsResult` → `ResultsTab` → `App`); App держит снимок и передаёт его вкладке «Документы» только пока тело запроса, собранное сейчас из проекта и усилий МКЭ, равно ключу (`anchorChecksSnapshotCurrent` в `services/checksInput.ts`): параметры корня, длины или усилия поменялись — ответ в записку не идёт. Тот же приём, что у снимка воды. Пакет документации — `generateDocumentationPackage(project, results, info, docSettings, excWidth, excLength, femResults, water, anchorChecks)` (девятый параметр); если параметры корня заданы, а актуального результата нет, `dataCollector.ts` добавляет предупреждение.
- **Спецификация, ведомость, сводка материалов.** `quantityCalculator.ts`, `specificationGenerator.ts`, `elementScheduleGenerator.ts`, `materialSummaryGenerator.ts`: у анкера с тягой из каталога стандарт тяги стоит в графе «Стандарт», «тяга К7-9,0-1770 × 4» — в примечании; масса стали тяги = n × масса 1 м × (Lсв + Lк) по каталогу тяг (`data/anchorTendonLookup.ts`: встроенный каталог и тяги поставщиков), она входит в итог по металлу и в строку «Сталь тяг анкеров» сводки; масса без инъекционного раствора и оголовка, это названо в примечании. Тяга не выбрана или в каталоге нет массы 1 м — прежняя строка «масса тяги не задана» (или «не определена»), в итог не входит, предупреждение называет анкер. Десятичная точка обозначений каталога («А500 Ø10.0») в документах заменена запятой (`tendonDesignationText`).

- **Отчёт МКЭ.** `reports/generateDocx.ts`, `buildSupports`: рядом с таблицей анкеров (глубина, угол, шаг, длины, F0, этапы) таблица «Тяги анкеров» (обозначение, стандарт, n, A, C, N_пред) — только если тяга выбрана хотя бы у одного анкера, с пояснением обозначений и ссылкой в тексте.
- **Тесты.** `anchorSection.test.ts` (22: состав, тяга-канат и тяга-стержень, усилия по этапам, N ⩽ N_пред, проверка корня как есть, «не проверялась», расчёт Блюма, место и нумерация, предупреждение, актуальность по ключу), `quantityCalculator.test.ts` (+3: масса тяги, спецификация, ведомость, тяга вне каталога), `textTemplates.test.ts`, `AnchorCapacityPanel.test.tsx`, `exportedDocs.guard.test.ts` (ПЗ с проверкой корня и без неё, отчёт МКЭ с таблицей тяг); входные данные — `testing/anchorReportCases.ts`.

Подбор решений: клиентская часть

- **Числа карточек — из ответов сервера.** services/solutionEngine/calcClient.ts — запрос /api/calculate, разбор ответа, ошибки с текстом сервера, кэш по телу запроса, AbortSignal; в CalculateResponse есть steel_grade (тест test_api_calculate_steel_grade.py), параметры проверки металла gamma_c, Ry_check_MPa, section_modulus_cm3, moment_of_inertia_cm4 и moment_capacity_kNm (W·Ry·γc при том же γc = 0,9, что и η; поле capacity_kNm с γc = 1,15 сохранено ради «Подбора решений»; [0395-отчёты-блюм], тест test_api_calculate_report_fields.py), отчёты Блюма берут их из ответа (services/blumResults.ts, reports/blumAssumptions.ts, blumDiagrams.ts, blumDocxParts.ts). По политике [0390-политика-блюма] консоль из шпунта считает классический метод Блюма (в его вердикт входят: расчёт сошёлся, заделка не меньше расчётной, Км металла ⩾ 1), а Км, Кз заделки и прогиб решения и длину стенки из заводского ряда (FACTORY_LENGTHS) даёт проверка МКЭ (services/femCheck.ts, judgeFemResponse: расчёт сошёлся и надёжен, Км ⩾ 1, Кз не ниже ориентира по классу сооружения — ориентир, не норматив, прогиб не выше min(H/100; 100 мм)). Стойки (труба, двутавр) и схемы со связями классический метод не считает (422): их длину и запасы даёт только МКЭ; класс Б без проверки МКЭ — «оценка сверху». Клиент считает только массу и число элементов по каталогу и выбирает длину из ряда. Каталог шпунта один — ECO_PROFILES (41 профиль); таблица AUTO_PROFILES удалена. Слои, не доходящие до дна котлована, останавливаются на шаге «Грунты».
- **Шаг «Ограждение» и отбор трёх решений** ([0509-подбор-решений], 26.10.2026). Мастер из четырёх шагов (STEP_LABELS в SolutionWizard.tsx: условия, грунты, параметры, ограждение). steps/WallChoiceStep.tsx — типы ограждения (шпунт, труба-стойки, двутавр-стойки), профили, шаг стоек, схемы раскрепления; в QuickInput добавлены wallCandidates (FinderWallCandidate: вид, обозначение, сечение) и bracingOptions (FinderBracing[]); constrainedConditions теперь исключает анкеры и бермы, region переносится в проект. wallChoice.ts, wallChoiceProblem: отмеченных вариантов не меньше SHOWN_SOLUTIONS = 3 и не больше MAX_CANDIDATES = 6, отмечена хотя бы одна схема; по умолчанию defaultCandidates() — RECOMMENDED_PROFILES (ECO 622T, ECO 632, Л5-УМ). catalog.ts: WallSection — общий тип сечения стенки для шпунта и стоек; трубы ГОСТ 10704-91 (диаметры 325…1020 мм, толщины 8, 10, 12 мм; по умолчанию Д426,0 х 8,0, Д530,0 х 8,0, Д630,0 х 8,0), двутавры СТО АСЧМ 20-93 (14 обозначений: 20…60 Б1 и 20…40 Ш1; по умолчанию 40 Б1, 50 Б1, 60 Б1), сталь стоек POST_STEEL_GRADE = S345, spacingProblem — шаг стоек не меньше ширины стойки (PILE_SPACING_DEFAULT = 1,5 м). Из отмеченных вариантов на экран идут три (solutionGenerator.ts, chooseShown: лучшее по рангу каждого

типа ограждения, затем остальные по рангу; rankSolutions — прошедший расчёт, без раскрепления, короче стенка, легче 1 м², больше Км, шире профиль), остальные попадают в result.considered (ConsideredVariant: label, status, reason) — блок «Рассмотрены также». solutionWords.ts — слова по типу ограждения (шпунт или стойки: длина, ширина или шаг, число элементов), supportVolumes — число и масса связей для блока «Итого на объект» (распорки: floor(длина / шаг) + 1 на уровень, масса по сечению; анкеры: ceil(периметр / шаг)). Буросекущие сваи и стена в грунте не предлагаются.

- **Допустимое смещение [w] в стеснённых условиях** ([0545-w-стеснённые-без-умолчания], 29.10.2026). Значения по умолчанию нет: `QuickInput.deflectionLimitMm` вводит пользователь в `ConditionsStep.tsx`; пока оно пусто, не больше 0 или больше `W_LIMIT_INPUT_MAX_MM = 1000` (граница ввода от опечатки, не нормативная; с [0569] — псевдоним `DEFLECTION_LIMIT_INPUT_MAX_MM` из `core/deflectionLimit.ts`), «Далее» недоступна на всех шагах (`deflectionLimitProblem`: «Задайте допустимое смещение: в стеснённых условиях его назначает проектировщик по условиям соседней застройки.»), а `generateAllSolutions` отказывает той же строкой (сохранённый подбор с флажком без [w]). Без флажка предел — рекомендуемый min(H/100; 100 мм) (`recommendedDeflectionLimit` в `core/deflectionLimit.ts`: одна формула на мастер и проект, прежняя `deflectionLimitFor` убрана; СП 381, прим. к п. 6.1.7: рекомендация для свободной площадки, не норма). `services/solutionEngine/deflectionLimit.ts`: `userDeflectionLimit`, `deflectionLimitOf`, `deflectionLimitProblem`. Числа расчёта не менялись: предел судит клиент (`judgeFemResponse`), в запрос МКЭ проверки мастера он не входит (`buildFemCheckRequest` в `services/projectToRequest.ts` — тот же запрос без `deflection_limit_mm`: тела запросов с [w] и без совпадают побитно — тест; в запросе проекта в редакторе поле есть, см. следующий пункт). `wizardFemChecker` судит МКЭ по [w]; кэш проверок ключуется телом запроса, поэтому исход из кэша пересуживается по текущему пределу: смена [w] не запускает МКЭ заново и не оставляет старого вердикта; `femToCalcResult` берёт предел из проверки (`w_limit_mm` результата); `SolutionFemCheck.wLimitUser` и `Solution.deflectionLimitUser` (`types.ts`). Консоль, не проходящая [w], отбрасывается: мастер берёт более длинную стенку или переходит к распорке. Слова «заданного вами предела» и «задано вами до N» вместо «рекомендуемого» — в причине непрохождения и строке проверки МКЭ (`femCheck.ts`), выводах (`conclusions.ts`), карточке (`SolutionCard.tsx`), контексте ассистента, в «Принятых допущениях» (`schemeRulesOf`: «Допустимое смещение [w] = N мм задано вами…»), в обосновании рекомендации («отобрано по заданному вами [w] = N мм») и в строке «Допустимое смещение [w], заданное проектировщиком» DOCX сравнения (`generateComparisonDocx.ts`, только когда [w] задано). Тесты: `deflectionLimit.test.ts` (11), `SolutionWizard.deflectionLimit.test.tsx` (8), дополнения в `femCheck.test.ts`, `conclusions.test.ts`, `finderTransfer.test.ts` (№ 5), `comparisonDocx.test.ts`; фикстуры со стеснёнными условиями несут `deflectionLimitMm`, иначе мастер и движок отказывают.
- **Допустимое смещение [w] в проекте профрежима ([0569-релиз-6-20-w-марка-ary], 30.10.2026; долг [0545]).** `WallState.deflectionLimitMm` (мм; нет поля — предел по умолчанию min(H/100; 100 мм) как прежде; допустимы числа больше 0 и не больше 1000 мм, те же границы и та же строка ошибки, что в подборе). Новый модуль без тяжёлых импортов `core/deflectionLimit.ts`: `projectDeflectionLimit`, `userDeflectionLimitMm`,

`isDeflectionLimitUser`, `deflectionLimitBasis` («задано пользователем» / «по умолчанию min(H/100; 100 мм)»), `deflectionLimitInputProblem`, `recommendedDeflectionLimit`, `migrateWallDeflectionLimit` (неверное значение при чтении файла проекта и при открытии проекта снимается; проект без поля — прежний, объект тот же) и одна функция печати предела — `formatDeflectionLimit` и `deflectionLimitText`: [w] печатается до одного знака без хвостовых нулей («7,5»; точка в XML, на экране и в строках проверок, запятая в документах), предел по умолчанию — целым, как печатался всегда. `deflectionLimitMm(wall)` в `core/limitStates.ts` берёт [w], когда оно задано и верно, иначе прежнюю формулу. Ввод — поле «Допустимое смещение [w], мм» блока «Параметры котлована» (`RetainingWallTab.tsx`, `DeflectionLimitField`): подсказка в поле «по умолчанию N», «?», ошибка одной строкой, очистка поля убирает его из проекта; неверное [w] — ошибка «Проверки схемы» (`core/schemeCheck.ts`), расчёт не запускается. По [w] судят: `buildFemLimitStates` (условие «w_max ⩽ [w]», слово статуса «не превышает / превышает допустимое смещение»; оговорка о заколах привязана к РЕКОМЕНДУЕМОМУ значению СП 381 и остаётся, если перемещение выше него, даже когда оно ниже заданного [w]), `resultsVerdict`, `resultBasisView`, границы применения (`applicability`: факты `deflectionLimitUser` и `deflectionExceedsRecommended`), `expertiseReadiness`, проверка МКЭ автоподбора профиля во вкладке «Ограждение», ТЭО (`teoFinder`: предел — [w] проекта, ключ `finderKey` включает [w], смена [w] делает подбор несвежим) и строка различий вариантов (`variantInputDiff`). Выгрузки: там, где уже печатается предел, стоит пометка «задано пользователем»; новых разделов нет, без [w] тексты прежние слово в слово. DOCX МКЭ: строка таблицы параметров «Допустимое смещение [w], задано пользователем», формула «wmax = … ⩽ [w] = … мм (допустимое смещение задано пользователем; рекомендуемое min(H/100; 100 мм) = … мм, …)», таблица проверок (через `buildFemLimitStates`), вывод 4 («не превышает / превышает заданное пользователем допустимое смещение [w] = N мм», ссылка на СП 381 — только у предела по умолчанию); XML — «Условие» «wmax <= [w], допустимое смещение задано пользователем», «Норм» «N мм (задано пользователем)», «Статус» «не превышает / превышает допустимое смещение»; записка — таблица предельных состояний и п. 7.3 (рекомендуемое min(H/100; 100 мм) как раньше и фраза «Принято допустимое смещение [w] = N мм (задано пользователем).»). Отчёты метода Блюма (учебный режим) предел сервера не меняют. В документах (сравнение вариантов, ТЭО) слова мастера «вами» заменены безличными (`deflectionTextForDocument`). В запрос МКЭ редактора [w] уходит как `deflection_limit_mm` (раздел 6.15; реестр `projectFieldFlow`: `wall.deflectionLimitMm` — вид «sent»); смена [w] в проекте делает результат устаревшим (ключ запроса другой: замечание сервера — часть результата). Тесты: `core/deflectionLimit.test.ts` (12), `core/deflectionLimitPrint.test.ts` (20), `components/RetainingWallTab.deflectionLimit.test.tsx` (6), `modules/documentation/export/deflectionLimitNote.test.ts` (6), `reports/deflectionLimitDocs.test.ts` (8), `modules/documentation/teo/finderFresh.test.ts`, `reports/clientWarningsW.test.ts`, дополнения в `services/projectFieldFlow.test.ts`.

- **Схемы и подбор связей** (services/solutionEngine/supportSizing.ts). BRACING_ORDER = none, strut, anchor (FinderBracing); bracingOrderFor оставляет отмеченные пользователем и применимые схемы, решение варианта — первая схема, которая проходит. Стеснённые

условия: bracingNotApplicable возвращает для анкера CONSTRAINED_ANCHOR_REASON («неприменимо: стеснённые условия»), бермы мастер не создаёт (berms: [] в проекте); пересчёт варианта с анкером в настройке возвращает ту же причину. Связь — на SUPPORT_DEPTH_FACTOR = 0,3 глубины котлована, шаг вдоль стенки 3 м (FINDER_STRUT_DEFAULTS, FINDER_ANCHOR_DEFAULTS), до её установки котлован откапывается на DEFAULT_STAGES.bottomToTieDistance = 0,5 м ниже связи (preSupportDepth в quickInputToProject.ts), поэтому у схемы со связью два этапа. Распорка: труба ГОСТ 10704-91 из стали С345 (SUPPORT_STEEL) на пролёт, равный ширине котлована; сечение — лёгкая труба из strutPipeShortlist(), проходящая проверку сечения (с [0530-проверки-распорки-и-пояса-в-api] её считает сервер, POST /api/checks/struts, тем же кодом, что в окне «Расчёт распорки»; pickStrutSection асинхронна, весь список труб уходит одним запросом, при отказе сервера вариант получает причину) при усилии МКЭ; стенка пересчитывается с жёсткостью сечения (strutStiffnessForLength) не более SIZING_ROUNDS = 3 раз; если не проходит ни одна труба до Д1020 х 12, схема не проходит с этой причиной. Анкер: ANCHOR_ANGLE_DEG = 15, тяга ANCHOR_TENDON_COUNT = 3 × К7-15,2-1860 (ANCHOR_TENDON_ID), инъекционный корень (прил. Л СП 381; диаметр скважины 150 мм, корня 200 мм), начальные длины Ls = 5 м и Lk = 4 м, предел длины корня 12 м; длины берутся по POST /api/checks/anchors (buildAnchorChecksRequest из services/checksInput.ts; СП 381 п. 7.3, прил. Л) с усилием из МКЭ, при новой свободной длине стенка пересчитывается (anchorStiffnessForLength, C = E·A·n/Ls); текст карточки говорит, что прочность тяги не проверялась. variantDraft.ts (draftSolution) — заготовка решения (сечение, длина, связи), по которой собирается и проект проверки МКЭ мастера, и проект при переносе.

- **Настройка варианта** (results/SolutionTuner.tsx): профиль из списка того же типа ограждения (шпунт каталога, труба, двутавр) с отметками «Проходит / Не проходит» (profileMarks.ts: у шпунта при консоли — запрос Блюма на профиль и проверка МКЭ длины стенки, у схем со связями и у стоек — целиком МКЭ, не больше MARK_FEM_CONCURRENCY = 2 запросов одновременно, без МКЭ — группами по 6; запросы отменяются по AbortSignal при смене схемы; связи в отметках — начальные схемы мастера, отметка ориентировочная, точное решение даёт пересчёт варианта), три схемы (кнопки «Без раскрепления», «Одна распорка», «Один ярус анкеров»; анкер в стеснённых условиях недоступен), длина из ряда не меньше требуемой. Любое изменение — пересчёт на сервере (computeVariant): пауза RECALC_DEBOUNCE_MS = 300 мс, предыдущий запрос отменяется, на время пересчёта сохранение заблокировано. Марку стали выбрать нельзя: у шпунта она берётся по каталогу, у стоек — S345 (бэкенд принимает только профиль; для выбора марки нужна правка CalculateRequest и check_profile — не сделано).
- **Выводы ассистента** (AssistantConclusions.tsx, conclusions.ts::buildConclusions) — детерминированный шаблон от результата подбора и рекомендации системы: нейросеть не вызывается, суточный лимит не тратится, на одинаковом входе текст одинаков, формулировки не утверждают, что результат проверен и соответствует нормам (тест на запрещённые слова). «Уточнить у ассистента» открывает панель с готовым вопросом в поле ввода (AppWrapper → SolutionFinderAssistant.openRequest → AssistantPanel.prefill); вопрос не отправляется сам. В блоке «На чём основан расчёт» названы правила мастера: у распорки — труба по усилию расчёта, шаг и то, что пояс не подбирался; у анкера — угол, шаг, тяга и длины по проверке СП 381 (conclusions.ts; в контекст ассистента

solutionFinderAssistantContext.ts добавлены тип ограждения, шаг стоек, рассмотренные варианты и правила подбора).

- **Проект из подбора** (перенос без потерь, [0509-подбор-решений]). quickInputToProject.ts — условия и выбранное решение → HydrowebProject: слои из того же normalizeInput, по которому считается подбор; размеры котлована в плане (planSize); ограждение — шпунт (профиль и марка по каталогу ядра) или стойки (труба либо двутавр с шагом, марка S345); этапы откопки (консоль — одна стадия на полную глубину, у схемы со связью два: до отметки на 0,5 м ниже связи и до дна); вода (УГВ, если задан); пригрузка с шириной; сейсмичность; распорка (сечение, шаг, жёсткость, предел) или анкер (угол, свободная и корневая длина, тяга, параметры корня для проверки по СП 381, прил. Л); бермы не переносятся (berms: []), Ренкин и минимальное активное 0,2; регион — в projectInfo.location проекта и (finderProject.ts) в meta.region карточки. Проверка МКЭ мастера собирает запрос из этого же проекта тем же сборщиком (buildFemRequest), поэтому проект в редакторе считает то же, что карточка решения: finderTransfer.test.ts сверяет запрос МКЭ проекта с запросом, по которому мастер принял решение, на пяти сценариях (шпунт-консоль; труба-стойки с распоркой; шпунт с одним ярусом анкеров, включая запрос проверки анкера; стеснённые условия — ни анкеров, ни берм; стеснённые условия с заданным [w], №5: [w] в поле проекта и в описании, запрос МКЭ проверки мастера и числа те же). Допустимое смещение [w], заданное в стеснённых условиях, переносится в поле проекта ([0569-релиз-6-20-w-марка-agy], долг [0545]): buildWall в quickInputToProject.ts кладёт его в wall.deflectionLimitMm (userDeflectionLimit; без стеснённых условий поля нет), и редактор судит перемещение по нему — вердикт, отчёты, XML и записка читают предел из проекта (deflectionLimitMm(wall) в core/limitStates.ts); finderProject.ts дополнительно записывает его в описание проекта («…, допустимое смещение [w] = N мм (задано в подборе)»; фраза оставлена: её сверяют сторожа книг), а «?» мастера (поле [w] на шаге «Условия» и кнопка «Открыть в проф. режиме») говорит, что [w] переносится в поле «Допустимое смещение [w]» вкладки «Ограждение». Запрос МКЭ проверки мастера идёт без поля (buildFemCheckRequest в services/projectToRequest.ts), поэтому числа проверки и расчёта проекта те же. finderProject.ts — сохранение с владельцем (createdBy = user.id, способ создания solution-finder, режим professional); «Открыть в проф. режиме» создаёт проект и открывает редактор на вкладке «Результаты» с автоматическим расчётом (EditorStartup в App.tsx).
- **Удалено:** документы по вариантам (SolutionExportPanel, autoExports.ts, шаги docQuestion/docForm/docGenerating/docReady, CloneProjectDialog, services/projectStorage.ts) — остаётся «Скачать сравнительный анализ» (DOCX); пакет документации формируется в профессиональном режиме. Тесты: SolutionWizard.flow (9), .tuning (10), .numbers (6), .snapshot (3), .wallChoice (5; шаг 4, справки, регион), AssistantConclusions (9), conclusions (12), finderProject (6), AppWrapper.finder (3; путь от списка проектов через четыре шага до редактора), catalog (6), quickInputToProject (10), solutionGenerator.wallChoice (11; порядок схем, стеснённые условия, сечение распорки, корень анкера длиннее предела), finderTransfer (4).

Вкладка «Чертёж»: панорамирование и масштаб

modules/drawing/components/DrawingPreview.tsx: «рука» — перетаскивание левой (и средней) кнопкой на Pointer Events с захватом указателя (setPointerCapture: перетаскивание не

обрывается за рамкой окна; touch-action: none); колесо — масштаб нативным слушателем с { passive: false } (React вешает onWheel пассивным, и preventDefault() не работал — страница прокручивалась вместе с масштабом); диапазон 0,2…5 (20–500 %), «Fit» сбрасывает масштаб и сдвиг; разметка листа пересобирается только при смене чертежа (useMemo), сброс масштаба при смене листа — во время рендера, не в эффекте. Тесты — DrawingPreview.test.tsx (12). Основная надпись чертежа — форма 3 ГОСТ Р 21.101-2026 (прил. Е, п. 5.2.2; titleBlockBuilder.ts, 05.10.2026), прежняя упрощённая сетка по ГОСТ 2.104 заменена: 185 × 55 мм в правом нижнем углу рамки (20 / 5 / 5 / 5 мм); левый блок 65 мм — графы 14–19 (10 / 10 / 10 / 10 / 15 / 10 мм), 11 строк по 5 мм (4 строки изменений, строка заголовков, подписанты: «Разраб.», «Пров.», две свободные, «Н. контр.», «Утв.»); правый блок 120 мм — графы 1, 2, 3, 4, 6–9; на левом поле листа — графы 20–22 (поле для подшивки). Масштаб указан второй строкой графы 4, год в дате — двумя цифрами (прим. 2), организация не подменяется названием проекта; графа 7 («Лист») на документе из одного листа не заполняется, графа 8 («Листов») — только на первом листе. Не выводятся графа 27 и «Согласовано» (на форме они штриховые, прим. 3) и графа 3 (данных о здании в проекте нет). Размеры сняты с рисунка формы и проверены по размерным надписям; вид просмотрен в SVG → PDF (A3). Побочно в DXF знак угла текста приведён к SVG (подписи поля для подшивки читаются одинаково). Тесты — titleBlockBuilder.test.ts (15).

PDF чертежа: векторный вывод с текстовым слоем ([0526-pdf-чертежа-с-текстовым-слоем])

- **Что изменилось.** Приказ Минстроя № 783/пр, п. 4 «а», «в», «г» (в электронном документе работают поиск и копирование текста; пересказ — в пункте «PDF не сканированы, текст ищется, есть закладки» списка готовности). Прежняя выгрузка рисовала SVG листа в холст 150 dpi и клала в jsPDF картинкой PNG: текста в файле не было. Теперь `modules/drawing/export/pdfExporter.ts` (`sheetToPdf`, `drawSheetToPdf`) выводит `DrawingSheet` прямо векторными командами jsPDF: линия, ломаная, прямоугольник, окружность, дуга (кубические кривые Безье не более 90°), штриховка грунта (семь шаблонов SVG-листа линиями, кольцами, точками и волной, обрезка контуром области `clip`, прозрачность 0,3 как у полигона SVG), размер (выносные и размерная линии, засечки 1 мм, число текстом, у вертикального повёрнуто), выноска со стрелкой, знак разреза, текст — настоящим текстом шрифта. Растр не остался ни для одного примитива. Формат и ориентация листа (A4, A3, A2, A1) прежние; jsPDF подгружается динамически при нажатии кнопки (`import('../export/pdfExporter')` из `DrawingTab`, внутри `import('jspdf')`); файл скачивается тем же `downloadBlob`, что SVG и DXF; отметка «Чертёж выгружен» и список готовности не менялись. Закладок и оглавления в PDF чертежа нет.
- **Шрифт и лицензия.** jsPDF для кириллицы требует встроенный TTF; в проекте подходящего не было (`frontend/public/fonts/` — только woff2 без базовой кириллицы). Взят Liberation Sans 2.1.5 Regular (лицензия SIL Open Font License 1.1: встраивание в документы и распространение подмножества разрешены; fsType 0): метрики совместимы с Arial, на них построено разведение подписей листа (`core/textMetrics.ts`), положения и ширины строк в PDF и на экране совпадают. `scripts/documents/build_drawing_pdf_font.py` (fontTools) вырезает подмножество (латиница, Latin-1, кириллица, греческие, типографские и математические знаки: 373 знака, TTF 28 КБ, без подсказок отрисовки и таблиц компоновки), переименовывает шрифт («HydroWeb Drawing Sans»: у изменённой версии нет права на прежнее имя) и сохраняет записи об авторских правах и лицензии; результат —

`modules/drawing/export/drawingPdfFont.generated.ts` (base64, коммитится, руками не правится). Знака ▽ (U+25BD) нет ни в Liberation Sans, ни в Arial: контур нарисован скриптом (полый треугольник шириной 0,6 em). Знак вне шрифта в PDF заменяется на «?» и возвращается в `replacedChars` (на листах продукта таких нет — проверяет тест).

- **Отличия от SVG.** Положение текста (якоря, поворот, базовая линия `central` и `hanging`) считается в модуле по ширине строки шрифта, а не якорями jsPDF (якорь jsPDF с поворотом сдвигает строку не вдоль неё); засечки размеров — перпендикулярные штрихи 1 мм толщиной выносной линии 0,18 мм (в SVG они волосяные, 0,09 мм); стрелка выноски направлена остриём в точку выноски (в SVG маркер развёрнут вдоль линии остриём «назад»; выноска есть только на чертеже узла); замкнутая ломаная, как и в SVG, не замыкается; неизвестный шаблон штриховки не рисуется. `DASH_ARRAYS` экспортирован из `svgExporter.ts` и общий для SVG и PDF.
- **Замер и тесты.** Демо-проект (план и разрез, ЕСО и Л5-УМ, берма, две распорки): PDF A4 24 КБ, A3 28 КБ, A1 45 КБ; независимая проверка PyMuPDF: растровых объектов 0, поиск «Разрез 1-1» и шифра находит ровно по одному вхождению, 744–798 символов текста на страницу, размер страницы равен листу. Тесты: `export/pdfExporter.test.ts` (12: четыре формата и ориентации — текстовый слой PDF равен текстам листа строка в строку, растра нет, MediaBox равен листу; «▽» и греческие читаются; штриховка с `clip`; скрытый слой не выводится; все примитивы и шаблоны; замена знака вне шрифта), разбор PDF без внешних библиотек — `export/pdfText.testutil.ts`; в `components/DrawingTab.test.tsx` +2 (кнопка PDF отдаёт векторный файл с «Разрез 1-1», без обращения к холсту; пробный период закрыл выгрузку).

Глава 9. Тестирование

Backend (pytest). На дереве сборки релиза 3 со строгим входом ([0415-сборка-релиза-3-строгий-вход], 21.10.2026: к дереву [0414-сборка-релиза-3] добавлены ветка строгого входа [0412-строгий-вход], две правки ссылок на нормы в выгрузках и комментарии о γm; hydroweb/core менялся только в комментариях; в рабочей копии без kb.db и данных ФГИС/КСР, HYDROWEB_MAX_CONCURRENT_CALCS=3) — 4874 passed, 11 skipped, 12 xfailed за 755 с (4897 тестов, на 1019 больше, чем на дереве [0414-сборка-релиза-3]: тесты строгого входа); pytest -m benchmark — 27 passed, 4 skipped, 2 xfailed; scripts/gw_compare — 386 passed, 8 skipped. Мутации строгого входа (scripts/maintenance/strict_input_mutations.py) — 148 подмен, убито 148, выжило 0; мутации чистки выгрузок (scripts/experiments/reports_test_1015/mutations_round2.py) — 36, выживших 0 (оба набора на одном дереве); ответы /api/calculate/fem и /api/calculate на 429 запросах корпуса (сохранённые запросы наборов сверки, варианты проектировщика, 14 запросов харнеса выгрузок, 316 вариантов берм) против дерева [0414-сборка-релиза-3]: 288 ответов 200 побитно равны (без полей elapsed), 141 — 200 → 422, все из вариантов берм (верх не выше дна — 76, этап установки больше числа этапов — 48, нет ширины — 17). Ниже — запись предыдущей редакции: на дереве сборки релиза 3 ([0414-сборка-релиза-3] [0411-без-флагов] с визуальным пакетом V1 [0410-визуал-V1] и чистка выгружаемых документов [0413-чистка-выгрузок]; hydroweb/ слиянием не менялся — менялись два файла тестов; в рабочей копии без kb.db и данных ФГИС/КСР, HYDROWEB_MAX_CONCURRENT_CALCS=3) — 3855 passed, 11 skipped, 12 xfailed за 438–745 с в зависимости от нагрузки машины (3878 тестов, на 3 больше, чем на дереве [0411-без-флагов]: форматы замечаний сервера о бермах и фразы книг); pytest -m benchmark — 27 passed, 4 skipped, 2 xfailed; scripts/gw_compare — 386 passed, 8 skipped. Ниже — запись предыдущей редакции: на дереве [0411-без-флагов] (20.10.2026, переменные окружения, менявшие поведение, удалены; в рабочей копии без kb.db и данных ФГИС/КСР, HYDROWEB_MAX_CONCURRENT_CALCS=4) — 3852 passed, 11 skipped, 12 xfailed за ≈ 11 мин на нагруженной машине (3875 тестов, на 434 меньше, чем 4309 до удаления: удалены тесты разбора переключателей, режима выкатки с прежними умолчаниями и эндпоинта, сообщавшего умолчания, ветви «прежнего» режима в тестах умолчаний; тесты поведения сохранены и переведены на явные поля запроса); pytest -m benchmark — 27 passed, 4 skipped, 2 xfailed (как на release/v1.1); scripts/gw_compare — 374 passed, 8 skipped (было 379). Числа и ключи ответов API равны прежним (относительно release/v1.1 499df4bc), кроме полей времени счёта elapsed_s и elapsed_ms: независимая проверка на 1097 уникальных запросах корпуса (наборы эталонов и проектировщика, бермы, вода, устойчивость, мутации входа) — 1095 ответов совпали при рекурсивном сравнении без этих полей; два различия заявлены заранее: бывший эндпоинт умолчаний расчёта (ответ 200 стал 404 — эндпоинт удалён) и схема openapi (в ней нет этого эндпоинта и описаний удалённых переменных окружения); отдельно 29 файлов харнеса выгрузок и 316 вариантов генеративной проверки берм — различий 0. Ниже — запись предыдущей редакции: на дереве поправок по третьей проверке слияния [0408-слияние-выгрузок-в-release] (17.10.2026, D15–D21; в рабочей копии без kb.db и данных ФГИС/КСР, HYDROWEB_MAX_CONCURRENT_CALCS=8) — 4286 passed, 11 skipped, 12 xfailed за 340 с (4309 тестов, на 36 больше, чем в предыдущей редакции: test_berm_notes_server_formats.py — 17,

test_books_texts.py — ещё 19; среди них 30 — проверка опубликованных копий книги пользователя TestPublishedCopies); pytest -m benchmark — 27 passed, 4 skipped, 2 xfailed (как на release/v1.1). Числовые поля ответов /api/calculate и /api/calculate/fem на запросах харнеса (П1…П5, П3б, П3в; 14 запросов и 8 вариантов запросов метода Блюма — «политика» и «учебный режим»; 22 ответа) совпадают с release/v1.1 (git archive 6725a746) до последнего знака: 15675 числовых значений, различий 0 (различаются тексты замечаний — 14 строк, в том числе без «(dewatered=true)» — и новые поля model_assumptions, envelope, поля проверки металла). git diff 6725a746 -- hydroweb/core пуст. Харнес выгрузок (rerun.sh): 91 файл и 56 PDF, все семь проектов построены Chrome с первой попытки, check_outputs.py — 0 срабатываний (шаг 7: 192 листа чертежа, файл измерителя из 12027 значений совпал с файлами двух предыдущих прогонов, различий 0), генеративная проверка берм — 0 расхождений (316 вариантов). Ниже — запись предыдущей редакции (повторные поправки после проверки слияния, 17.10.2026): на дереве повторных поправок после проверки слияния [0408-слияние-выгрузок-в-release] (17.10.2026; в рабочей копии без kb.db и данных ФГИС/КСР, HYDROWEB_MAX_CONCURRENT_CALCS=8) — 4250 passed, 11 skipped, 12 xfailed за 340 с (4273 теста, из них 30 — проверка опубликованных копий книги пользователя test_books_texts.py::TestPublishedCopies: они читают копии на сайте и проходят после пересборки книги, до неё падают намеренно); pytest -m benchmark — 27 passed, 4 skipped, 2 xfailed (как на release/v1.1). Числовые поля ответов /api/calculate и /api/calculate/fem на семи проектах харнеса (П1…П5, П3б, П3в) совпадают с release/v1.1 (git archive 6725a746) до последнего знака: 13266 числовых значений, различий 0 (различаются тексты замечаний и новые поля — model_assumptions, envelope, поля проверки металла). Ниже — запись предыдущей редакции (первые поправки после проверки слияния, 17.10.2026; в рабочей копии без kb.db и данных ФГИС/КСР, HYDROWEB_MAX_CONCURRENT_CALCS=8) — 4194 passed, 11 skipped, 12 xfailed за 340 с (4217 тестов, из них 19 — проверка опубликованных копий книги пользователя test_books_texts.py::TestPublishedCopies: они читают копии на сайте и проходят после пересборки книг, до неё падают намеренно); pytest -m benchmark — 27 passed, 4 skipped, 2 xfailed (как на release/v1.1). Числовые поля ответов /api/calculate и /api/calculate/fem на шести проектах харнеса (П1…П5, П3б) совпадают с release/v1.1 до последнего знака: 11468 числовых значений, различий 0 (различаются тексты замечаний и новые поля — model_assumptions, envelope, поля проверки металла). Ниже — запись предыдущей редакции (слитое дерево до поправок): на дереве слияния (release/v1.1 `6725a746` и ветка выгрузок `work/отчёты_дефекты_1015`) — 4129 passed, 11 skipped, 12 xfailed за 339 с (4152 теста в 211 файлах); pytest -m benchmark — 27 passed, 4 skipped, 2 xfailed (как на release/v1.1). Ядро hydroweb/core при слиянии не менялось: числовые поля ответов /api/calculate и /api/calculate/fem на четырёх проектах тестовой выгрузки (П1, П2, П3, П3б) совпадают с release/v1.1 до последнего знака (различаются тексты замечаний, служебное поле elapsed_s и новые поля). Ниже — запись предыдущей редакции (ветка выгрузок до слияния): на дереве после [0406-отчёты-шестая-волна] (16.10.2026; hydroweb/ в этой правке не менялся, в рабочей копии без kb.db и данных ФГИС/КСР) — 3624 passed, 11 skipped, 12 xfailed, 3 xpassed за 293 с (то же число, что после [0405], [0404] и [0403]). Ниже — запись предыдущей редакции: после [0405-отчёты-пятая-волна] — 3624 passed, 11 skipped, 12 xfailed, 3 xpassed за 294 с. Ниже — запись предыдущей редакции: после [0404-отчёты-четвёртая-волна] — 3624 passed, 11 skipped, 12 xfailed, 3 xpassed за 321 с. Ниже — запись предыдущей редакции (release/v1.1 до слияния):

на дереве после [0376]–[0386], [0385] и [0389] (ветка внедрения, 15.10.2026, в рабочей копии без kb.db и данных ФГИС/КСР) — 3624 passed, 11 skipped, 12 xfailed, 3 xpassed за 299 с (3650 тестов в 200 файлах). Ниже — запись предыдущей редакции (ветка выгрузок и ветка внедрения до слияния): на дереве после [0376]–[0386] и [0385] (ветка внедрения, 15.10.2026, в рабочей копии без kb.db и данных ФГИС/КСР, через очередь тяжёлых прогонов) — 3591 passed, 11 skipped, 12 xfailed, 3 xpassed за 322 с (3617 тестов в 199 файлах); scripts/gw_compare — 315 passed, 6 skipped. Ниже — запись предыдущей редакции (release/v1.1 `48edf32`). Последний полный прогон 08.10.2026 на дереве сборки предыдущей редакции (release/v1.1 после [0343], hydroweb/tests, через очередь тяжёлых прогонов, в рабочей копии с копиями kb.db и egrz.db) — 2924 passed, 10 skipped, 15 xfailed, 1 xpassed за 539 с (всего 2950 тестов в 184 файлах; после ускорения [0293-C1] полный прогон занимал 5–8 мин на свободной машине, до него ~30 мин). Два из десяти пропусков — тесты сырых файлов ФГИС ЦС из рабочие_материалы, которых в рабочей копии нет: с HYDROWEB_FGIS_RAW=<каталог> оба проходят (отдельный запуск, 3 passed), то есть при полном наборе данных 2926 passed, 8 skipped; для сверки и параллельных прогонов HYDROWEB_MAX_CONCURRENT_CALCS=8. Сверочные инструменты scripts/gw_compare — 168 passed, 5 skipped (173 теста); блок контрольных задач pytest -m benchmark — 26 passed, 4 skipped, 3 xfailed. По сравнению с редакцией книг на 3e5cd8d (2880 passed при 2904 тестах) прибавилось 46 тестов [0343], все проходят (список ниже). По сравнению с прогоном утром 08.10.2026 прибавились один xfail и один skip: сравнение записи одного профиля каталога аналогов (hydroweb/data/competitors.json) с каталогом изготовителя помечено xfail — запись не принадлежит ни одной строке каталога, данные продукта не правились; тест сверки с копиями источников (test_concrete_sources.py) пропускается, пока копии не скачаны: в репозиторий они не входят. Один модуль core/X.py → один tests/test_X.py. Тесты потоков ядра: test_run_control (13), test_calc_status (24), test_api_calc_robustness (41), test_anchor_capacity (35), test_api_checks_anchors (22), test_slope_stability (33), test_api_checks_stability (15), test_hydraulic_stability (29), test_api_checks_hydraulic (14), test_aquifers_core (16), test_aquifers_api (15), test_free_standing_mode (14), test_load_combinations (25), test_mesh (11), test_api_fem_mesh_snap (5), test_ratchet_check (25), test_fem_default_newton (10), test_fem_layer_kprop (66 — K по слоям), test_gwx_projects_set (24 — набор gwx_projects), test_sheet_pile_area (13 — площадь A каталога шпунта), test_pipe_catalog (15) и test_pile_catalog (5 — сортамент труб), test_beam_catalog (21 — сортамент двутавров и швеллеров), test_*_sheet_piles_source (19), test_concrete_sources (17), test_strand_sources (17), test_catalog_confirmation_stats (4) — проверки выписок сортаментов и норм, у которых нет потребителя в продукте. Тесты надёжности и входа (07.10.2026): test_backup_scripts (39 — скрипты копий тома и пользовательских таблиц и проверки восстановления без Docker, через подмену docker в fixtures/fake_docker.sh; в том числе восстановление дампом, снятым настоящим прежним скриптом), test_remote_build (25 — шаг 4 выкатки с запасным путём при лимите Docker Hub: ssh подменяется, а также весь deploy.sh с подменёнными npm, rsync, ssh и sleep), test_prod_health (15 — монитор прода без сети), test_auth_jwt_compat (29 — переход на PyJWT: совместимость выданных токенов, отказы на неверный алгоритм, подпись, срок). Вход под нагрузкой (08.10.2026, [0343]): test_db_connection (10 — миграции на старте и ожидание блокировки: «старая» база, повторный старт без изменений, соединение запроса не держит блокировку записи, lifespan до первого запроса, сбой миграции пишется в журнал и повторяется), test_login_concurrency (4 — 25 одновременных входов дают 25 ответов 200,

/api/auth/me отвечает во время проверки пароля, bcrypt вне цикла событий во всех путях, гонка регистрации одного адреса), test_compose_env (21 — переменные проброшены в docker-compose.yml, умолчание compose равно умолчанию кода, другое значение код берёт), test_rate_limit (35, из них 11 новых — умолчания, переменные окружения, неверные значения; вход, регистрация и смена пароля на эндпоинтах). Тесты инструментов сверки лежат рядом со скриптами и запускаются отдельной командой (PYTHONPATH=. pytest scripts/gw_compare): test_import_gwx (22 — на придуманных файлах; один тест на реальных образцах пропускается без GW_SAMPLES_DIR), test_import_gwx_structures (26), test_import_gwx_projects_0708 (14 — двусторонняя откопка, пионерный котлован, вода с напором, файлы версии 7), test_import_gwx_angles (28 — округление углов при импорте), test_ry_table_v5_sweep (26 — инструмент замера подмены таблицы Ry), test_pos_sweep (19), test_manual_examples_v8 (11), test_analog_2139 (13), test_steel_modulus_sweep (9), test_gw84_parse (5). Тесты сметы и данных ФГИС ЦС: test_estimate_* — 14 файлов, 454 теста (калькулятор — 33, API — 13, приёмка — 13, подбор норм опор — 78, ВЗиС и зимнее — 52, примечания зон — 4, независимая проверка review* — 233), test_fgis_snapshot (35, часть — на реальных файлах, пропускается без них), test_ksr_binding (16).

Frontend (vitest). На дереве сборки релиза 3 со строгим входом ([0415-сборка-релиза-3-строгий-вход], 21.10.2026; --maxWorkers=2 --testTimeout=300000 --hookTimeout=300000) — 234 файла (233 запускались, 1 пропущен целиком), 2986 тестов: пройдено 2983, пропущено 3, за 309 с; npm run typecheck и npm run lint — 0 ошибок, npm run build проходит. Ниже — запись предыдущей редакции: на дереве сборки релиза 3 ([0414-сборка-релиза-3], 21.10.2026; --maxWorkers=2 --testTimeout=120000 --hookTimeout=120000) — 232 файла (231 запускался, 1 пропущен целиком), 2909 тестов: пройдено 2906, пропущено 3, за 343–632 с в зависимости от нагрузки машины (при тайм-ауте теста 120 с на нагруженной машине выходят за срок отдельные сторожа выгрузок и измеритель чертежа: в отдельном прогоне они проходят, как и весь набор с тайм-аутом 900 с) (на дереве [0411-без-флагов] с пакетом V1 было 228 файлов и 2739 тестов, на дереве чистки — 222 файла и 2781 тест); npm run typecheck и npm run lint — 0 ошибок. Ниже — запись предыдущей редакции: на дереве [0411-без-флагов] (20.10.2026, --maxWorkers=3 --testTimeout=300000 --hookTimeout=300000) — 209 файлов (208 запускались, 1 пропущен целиком), 2484 теста: пройдено 2481, пропущено 3, за 141 с (было 2582 теста в 215 файлах: удалены тесты запроса умолчаний у сервера, янтарной плашки и повтора запроса); npm run typecheck, npm run lint — 0 ошибок, npm run build проходит. Ниже — запись предыдущей редакции: на дереве поправок по третьей проверке слияния [0408-слияние-выгрузок-в-release] (17.10.2026, D15–D21; --maxWorkers=2 --testTimeout=300000 --hookTimeout=300000, одновременно шёл прогон pytest) — 215 файлов (214 запускались, 1 пропущен целиком), 2582 теста: пройдено 2579, пропущено 3, за 253 с; npm run typecheck и npm run lint — 0 ошибок. Ниже — запись предыдущей редакции (повторные поправки): на дереве повторных поправок после проверки слияния (17.10.2026; --maxWorkers=2 --testTimeout=300000 --hookTimeout=300000) — 213 файлов (212 запускались, 1 пропущен целиком), 2504 теста: пройдено 2501, пропущено 3, за 143 с; npm run typecheck и npm run lint — 0 ошибок. Ниже — запись предыдущей редакции (первые поправки): 211 файлов (210 запускались, 1 пропущен целиком), 2475 тестов: пройдено 2472, пропущено 3, за 203 с. Ниже — запись предыдущей редакции (слитое дерево до поправок): 206 файлов (205 запускались, 1 пропущен целиком),

2405 тестов: пройдено 2402, пропущено 3, за 139 с. Ниже — запись предыдущей редакции (ветка выгрузок до слияния): на дереве после [0406-отчёты-шестая-волна] (16.10.2026, лимит теста 60 с из vite.config.ts) — 191 файл (190 запускались, 1 пропущен целиком), 2152 теста: пройдено 2149, пропущено 3; npm run typecheck и npm run lint — 0 ошибок. Ниже — запись предыдущей редакции: после [0405-отчёты-пятая-волна] — 190 файлов, 2117 тестов: пройдено 2114, пропущено 3. Ниже — запись предыдущей редакции: после [0404-отчёты-четвёртая-волна] — 187 файлов, 2044 теста: пройдено 2041, пропущено 3. Ниже — запись предыдущей редакции: после [0403-отчёты-третья-волна] — 184 файла, 1970 тестов: пройдено 1967, пропущено 3. Ниже — запись предыдущей редакции: на дереве после [0376]–[0386] и [0385] (ветка внедрения, 15.10.2026) — 157 файлов, 1551 тест: пройдено 1548, пропущено 3 (на нагруженной машине один сторож выгрузок упирается в 5 с — с --testTimeout=60000 проходит). Ниже — запись предыдущей редакции (08.10.2026): npm test (npx vitest run) — 152 файла по списку (151 запускался, 1 пропущен целиком — пример тома ТЭО без HYDROWEB_TEO_SAMPLE_DIR), всего 1500 тестов: пройдено 1498, пропущено 2 (второй пропуск — выгрузка тел запросов учебных заданий без HW_DUMP_DIR и HW_DUMP_OUT) за 79 с (прогон 08.10.2026 на дереве сборки этой редакции, тест «весь конвейер» в xlsxSpecification.test.ts пропускается без демо-проекта документы/эталоны_и_примеры/примеры_файлов_продукта/проект_1.hwproj.json (в git его нет): в рабочей копии без него тот же прогон даёт 1497 passed, 3 skipped; npx vitest list показывает 1498 — без пропущенных). На загруженной машине штатные 5 с на тест иногда не хватает для сторожей выгрузок (генерация DOCX): тогда запускают с --testTimeout=120000 --hookTimeout=120000, как этот прогон (плюс --maxWorkers=4). В том числе смета (buildEstimate, normSelection, EstimatePanel, xlsxEstimate, estimateApi), том ТЭО (teo/*), сторожа выгрузок (exportedDocs.guard — глава 8, раздел «Выгрузки Word: числа, адреса норм и формулировки»; xlsxSpecification, gostStamp, projectMark, metalSpec, specificationGenerator, gipCertificate, stageResultsTable, numFormat, volumeSheets, reportOptions, ResultsTab.reportOptions, штамп чертежа titleBlockBuilder), подбор решений (SolutionWizard.*, AssistantConclusions, conclusions, finderProject), привязка КСР, DrawingPreview, проверка распорки по СП 16 (strutCheck — 34 теста, strutCurve — 4, beamCatalogImpact — 2, PipeStrutCheckDialog — 8, supportChecks — 16) и сортаменты труб, двутавров и швеллеров (pipeCatalog — 8, beamCatalog — 11, beamSelect — 9, PipeSectionPicker — 9, BeamSectionPicker — 9). Остальное: список проектов, экспорт и импорт, ассистент, кабинет и админка, меню, вкладки, результаты и вердикт, статус и ход расчёта (femJobs — 15, calcStatus — 20, useCalcProgress — 10, ResultsTab.status — 8, App.calcRun — 4), проверки, сторож «поле интерфейса → запрос» (projectFieldFlow, projectToRequest), шаблон нового проекта, границы ошибок.

Пакет 24.10.2026 (воды, узлы, поставщик, импорт). По записям журнала: [0460] — `test_api_checks_water.py` (Дюпюи вручную, равенство модулю со шпунтом, монотонность по глубине шпунта, средневзвешенный k, сухой котлован, 9 отказов 422), строгий вход 232 passed, vitest `WaterInflowPanel.test.tsx`; [0462] — `test_joints.py` (34, ручные расчёты по формулам СП), `test_api_joints.py` (29), строгие сторожа 428 passed, vitest по диалогу узла, чертежу и разделу записки; [0461] — `test_supplier_cabinet.py` (39) и pytest `hydroweb/tests` 5114 passed, 11 skipped, 13 xfailed, vitest 3091 passed (полный прогон ветки поставщика); [0459] — `test_project_import.py` (36), `test_project_import_fixtures.py`, vitest

`importedProject.test.ts` (18), `ProjectFileImportDialog.test.tsx`, `App.importedProjects.test.tsx`, pytest `hydroweb/tests` 5034 passed, 12 skipped, 13 xfailed, vitest 3113 passed, 5 skipped (256 файлов), `scripts/gw_compare` 412 passed, 8 skipped. Числа относятся к веткам этих записей до слияния.

> **Семафор расчётов и сравнение ответов целиком**
>
> Семафор С3 («не более 2 расчётов») общий на МАШИНУ: параллельные прогоны сверки и тестов получают 429. Для скриптов сверки ставить HYDROWEB_MAX_CONCURRENT_CALCS=8 (раздел 2, переменные окружения).
>
> Дедлайн 240 с долгие тесты после ускорения сборки К больше не задевает (аналог 1949.3 с распределённой нагрузкой на стенку: 519 с по часам / 288 с процессорного времени до С1, 4,2 с после — М тот же, 169,6); на медленных машинах и в CI есть HYDROWEB_CALC_DEADLINE_S.
>
> Сравнивая ответы /api/calculate/fem целиком, исключайте elapsed_s (длительность, недетерминирована): так сделаны test_api_seismic_notes и test_load_combinations.

```
pytest hydroweb/tests/ -q                    # весь набор
pytest hydroweb/tests/test_gw_references.py  # сравнение с пром. аналогом
pytest -m benchmark --benchmark-report       # реестр контрольных задач
```

Двухуровневая проверка ядра

- **Уровень 1 — замкнутые решения (жёстко):** предельные переходы ядра к замкнутым решениям (Hetényi, Ренкин, кубика Блюма на всех семи контрольных задачах Б-1…Б-7) — допуск ⩽ 0,5–2 %; у Блюма с [0551] и [0573] по скалярам не хуже 0,015 % (округление ответа), по узловым эпюрам Q и M не хуже 0,07 % от максимума, вердикт pass у всех семи (до [0551] у Б-2, Б-5, Б-6 было 1,2–3,0 % с установленной причиной — сетка внутренних усилий ядра — и отдельный статус, не pass; до [0573] — до 0,110 %); для Кз МКЭ аналитического эталона нет.
- **Уровень 2 — эталонная программа (информативно):** реалистичные сценарии, инженерные допуски ±10–25 % (разные расчётные модели; это сравнение двух численных программ, а не сопоставление с точным решением).
- **Уровень 3 — храповик сверки (регресс):** по сохранённым запросам 56 эталонов (и отдельным гейтом — набора проектировщика) проверяется, что правка кода не ухудшает сверку (см. ниже).

Храповик сверки (scripts/gw_compare/ratchet_check.py)

Статус «41 из 56» — характеристика API, а не пути через интерфейс, и до храповика ничто не мешало ему откатиться. Храповик сравнивает новый прогон с базовой линией по тем величинам, по которым ведётся сверка. Единица счёта — пара «величина × расчёт» (M, Q, w, Кз, Км на каждом расчёте); пара «в допуске», если |Δ| ⩽ 10 % от величины эталона.

- `refs_requests/` — сохранённые запросы 56 эталонов (`prod_sweep.py --dump-requests`): печатные величины эталона и запрос к API (название профиля обезличено, на счёт не влияет). Отчёты .docx в git не лежат — по этим файлам сверку повторяет любой, в том числе CI. Актуальность после правок разбора — `--verify-requests`; `--from-requests` считает по ним; в строке результата — статус, код ответа, предупреждения, поэтапные итоги, при --local время и итерации решателя по этапам.

- Гейт (аудит, приложение В, шаг 1): (1) ни одна «чистая» пара базы (в допуске) не выходит из допуска — выход в полосу 10–11 % (--warn-band) предупреждение, дальше провал; расчёт, посчитанный в базе и упавший в новом прогоне, — провал; (2) «все пять» (расчёты, у которых в допуске все пять величин) — не меньше базы; (3) сумма выхода за 10 % Σ max(0, |Δ| − 10) — не больше базы (--sum-slack — запас на шум архитектуры); (4) медиана |Δ| по каждой величине — не хуже базы более чем на 0,2 п.п. (--median-slack); (5) набор проектировщика — отдельный гейт с теми же правилами (основа linearDeformable, как у основного). Режим --solver-switch [РСТ] — для смены численного режима решателя (критерий (а) в формулировке [0297], раздел 6.13). Коды возврата: 0 — пройден (предупреждения допустимы), 1 — регресс, 2 — ошибка ввода.

```
# новый код против базовой линии (решатель по умолчанию — Ньютон; полный набор ≈ 1
мин при --jobs 4)
python scripts/gw_compare/ratchet_check.py
scripts/gw_compare/prod_sweep_local_2026-10-05_newton_base.json \
    --local --jobs 3 --designer-base
scripts/gw_compare/designer_sweep_local_lindef_2026-10-05_newton_base.json \
    --designer-local
# быстрый вариант (как в CI): подмножество ≤ 30 с на расчёт
python scripts/gw_compare/ratchet_check.py
scripts/gw_compare/prod_sweep_local_2026-10-24_v4_newton_base.json \
    --local --subset scripts/gw_compare/ratchet_lite.txt --jobs 2 --sum-slack 1.0
# два готовых прогона / отчёт о сходимости
python scripts/gw_compare/ratchet_check.py BASE.json NEW.json
python scripts/gw_compare/convergence_report.py NEW.json --base BASE.json --out
сходимость.md
# база посчитана другим режимом решателя (Пикар → Ньютон): критерий (а) [0297]
python scripts/gw_compare/ratchet_check.py picard.json newton.json --solver-switch
1.0
```

- CI: подмножество `ratchet_lite.txt` (21 расчёт, ⩽ 30 с) считается заданием «Храповик сверки (ratchet-lite)» файла `.github/workflows/ci.yml` по `refs_requests` против базы `prod_sweep_local_2026-10-24_v4_newton_base.json` с `--sum-slack 1.0` (база посчитана на arm64, x86_64 расходится на 0,1–0,5 %). Задание блокирующее, итог пишется в сводку прогона; остальные задания и условия — в разделе «Непрерывная интеграция» ниже.
- **Базовые линии с [0297] (05.10.2026, решатель по умолчанию — Ньютон, local, arm64, нескруглённые числа product_raw):** prod_sweep_local_2026-10-05_newton_base.json (основной набор, 56: все пять 41/56, Σ выхода 1692,0 п.п.), designer_sweep_local_lindef_2026-10-05_newton_base.json (восстановленные, 11: все пять 4/11, Σ 597,6) и designer_docx_sweep_local_2026-10-05_newton_base.json (настоящие отчёты, 9 строк: все пять 6, Σ 136,4; в храповик пока не подключена, берётся отдельным вызовом ratchet_check.py BASE NEW). Прежние *_2026-10-01_0293base.json — Пикар (initial_stress), оставлены как история: на ядре с дефолтом newton обычным храповиком они проваливаются (Σ +1,3 п.п., медиана Q +0,35 п.п.) — это сдвиг решателя, а не регресс; сопоставляются режимом --solver-switch. Базовые линии (*base.json) трекаются; при осознанном изменении чисел базу пересчитывают и коммитят вместе с правкой, а вышедшие и вошедшие пары записывают в журнал. Тесты — test_ratchet_check.py (25).
- **Необязательные наборы реестра ([0318-храповик-gwx], 07.10.2026).** `scripts/gw_compare/gate_sets.py` — реестр наборов сверх трёх основных: ключ → каталог запросов, база по умолчанию, включён или нет. Включён gwx_projects — два расчёта, запрос которых собран из файла проекта (2139.2, версия 8.4, и 1614.2, версия 7.x;

записи refs_requests_gwx_projects/, база gwx_projects_sweep_local_2026-10-07_newton_base.json): «все пять» 2 из 2, Σ выхода 0,0 п.п. (M +1,3 / Q −2,7 / w −7,8 / Кз +6,6 / Км +7,6 % и +0,3 / +0,6 / +0,4 / −0,3 / +0,3 %); в общий итог трёх наборов не входит (справочная строка «с наборами»: 52 из 75 при all_sweep.py --with gwx_projects). Ключ pos заведён с enabled=False: решение владельца — пробный набор, не гейт, данные вне git. Флаги ratchet_check.py --set-base КЛЮЧ[=ФАЙЛ], --set-new КЛЮЧ=ФАЙЛ и --set-local КЛЮЧ дают названному набору отдельный гейт по тем же правилам; без флагов реестр не импортируется, текст отчёта и код возврата прежние (вывод старым и новым скриптом на одних JSON побайтно равен); неизвестный, невключённый, названный дважды или неполный набор флагов — код 2. Как «проект вне выборки» для проверки гипотез эти два расчёта не годятся: файлы уже прочитаны. Шаг CI не добавлен: базы посчитаны на arm64, на x86_64 расходятся на 0,1–0,5 %, а медианы двух расчётов к этому чувствительны. Тесты: test_gwx_projects_set.py (23) и test_import_gwx.py. Реестр, запуск и контракт включения ПОС — в scripts/gw_compare/README.md.

Статус при этом даёт только prod_sweep.py через продукт: харнес collect.py зовёт ядро напрямую с исследовательскими калибровками, которых в продукте нет.

Три набора эталонов и инструменты сверки (05.10.2026, [0296])

Сверка с эталонной программой идёт по трём наборам; считаются они на одной основе — модели запаса linearDeformable (дефолт продукта), — и показываются отдельными строками. Это сравнение двух численных программ, а не сопоставление с точным решением.

Набор	Состав	«Все пять» в допуске ±10 %
Основной	56 отчётов эталонной программы (эталонные_расчёты/отчёты_эталона), входы разбираются из .docx	41 из 56
Восстановленные	11 расчётов проектировщика, вход восстановлен по скриншотам (эталонные_расчёты/расчёты_проектировщика, готовые запросы; у части два варианта входа — «как ввёл» и «с нашими поправками»)	4 из 11
Настоящие отчёты проектировщика	отчёты, присланные 05.10.2026 (8 файлов; запросы — scripts/gw_compare/refs_requests_designer_docx/, шесть с печатными числами)	5 из 6 (без дубля 1726.2, который совпадает с основным набором; 2130.3 — в отчёте нет результатов, вне счёта)

Общий итог на одной основе — 50 из 73 (прежние «45 из 67» не включали настоящие отчёты). Разрыв «4/11» против «5/6» по разбору 05.10 связан в первую очередь с тем, что вход восстановленных расчётов реконструирован по скриншотам (на 2134 по входу из отчёта расхождение не больше 2,1 %); выборка настоящих отчётов — шесть расчётов, поэтому вывод о том, насколько ядро совпадает «вне выборки», остаётся предварительным.

- **Ловушка выравнивания основы.** Харнес основного набора жёстко шлёт safety_model = "linearDeformable", а сохранённые запросы проектировщика несут "auto" (так их собрали раньше); сводить их в один счётчик без выравнивания нельзя — так завысили разрыв

между наборами в [0265]. scripts/gw_compare/all_sweep.py считает три набора с выровненной основой и печатает итог по каждому отдельно и общий.

- **Шестая величина — усилие связи** (информационно, вне гейта). Печатное «Расчетное продольное усилие» есть у 27 отчётов основного набора и у 2169.2 и 2170.3; оно хранится в reference_info.F сохранённого запроса, а строка результата получает force_info (печатное значение, значение продукта — наибольшее force_axial_kN по распоркам и анкерам, разность). Величина выводится в сводках prod_sweep.py и designer_docx_sweep.py и отдельным блоком «информационно, вне гейта» в отчётах ratchet_check.py и convergence_report.py; гейт по-прежнему читает пять величин. Информационно: в допуске у 17 из 27 отчётов, медиана |Δ| — 5,3 %.
- **Паспорт раскладки версии программы** (gw_compare_lib.layout_passport): отчёты эталонной программы различаются раскладкой в зависимости от версии — 7.x (печатная строка 7.3.26 или 7.4.0), 8.x «Глубина слева», 8.x «Zабс слева», 8.4 «одна сторона». По 66 просмотренным .docx: 7.x (7.3.26) — 47, 7.x (7.4.0) — 2, «Глубина слева» — 4, «Zабс слева» — 8, 8.4 — 3, «не определена» — 2 (файлы без результатов, вне набора 56). Паспорт хранится в поле passport запроса, в строке результата (layout) и в сводке «все пять по раскладкам»; --verify-requests сверяет теперь и паспорт, и reference_info (актуальны 56 отчётов).
- **Разбор раскладки версии 8.4** (gw_compare_lib.parse_gw84, scripts/gw_compare/test_gw84_parse.py): добавлен аддитивно — 56 прежних сохранённых запросов не изменились. В отчётах 8.x номер версии не печатается, 8.4 известна по скриншоту окна проекта 1907.
- **Пионерный срез, расчёт 2169.2** (исторический замер до 22.10.2026, когда режимы «призма» и «полосами СП 22» ещё были; scripts/gw_compare/designer_docx_pioneer_pressure.py; единственный расчёт, где известны и режим печати «по численному методу», и величины первого этапа). Выше дна выемки равнодействующая давления тыла: эталон 11,9 кН/м, призма (дефолт) 9,8 (−17 %; недобор целиком в верхних 0,5 м), «по силам» и «полосами СП 22» по 2,2 (−81 %) — при этом момент первого этапа у них выше печатного на 65–76 %: недобор давления вверху компенсируется избытком нетто-давления ниже дна, поэтому их попадание в огибающую третьего этапа — компенсация ошибок, а не подтверждение режима. Давление ниже дна на этапе 1 численно не прочитано: готовая оцифровка epure_pressure.digitize_pressure на этом рисунке не проходит собственную проверку (кривая ниже дна пилообразна). Единого режима для среза измерение не выводит; в ядро ничего не внесено. Масштаб читался по печатным подписям рисунка вручную (разброс 6,6 %), расчётная схема одна.
- **Полный храповик по умолчанию на коде ветки (до переключения на Ньютон, [0296])** (основа ядра не менялась): основной набор — «все пять» 41 → 41, Σ выхода 1690,7 → 1690,7 п.п.; набор проектировщика — 4 → 4, Σ 599,3 → 599,3: ядро считало цифра в цифру как база (решателем Пикара; после [0297] базы пересчитаны, раздел 6.13). Тесты инструментов — test_gw_compare_tooling_0296.py (12).
- **Прототип импорта файлов проектов стороннего расчётного ПО** `scripts/gw_compare/import_gwx.py` (07.10.2026, [0306-импорт-gw]). В продукт НЕ включён, решение о публикации функции — за владельцем. Читает присланные проектировщиком файлы проекта двух форматов: версия 8 — ZIP с открытым XML (служебные списки настроек, исходные данные, кэш коэффициентов давления; единицы СИ

— Па, Н/м³, Н, Н/м, радианы), версия 7 — контейнер со встроенной базой Microsoft Jet 4 и текстом в стиле INI (шифрования и пароля на данных нет; таблицы базы читает минимальный читатель, написанный по структуре страниц самой базы). Собирает словарь входов и тело запроса /api/calculate/fem тем же prod_sweep.build_request, что и разбор отчёта, поэтому сравнение «запрос из файла ↔ запрос из отчёта» сравнивает источники, а не два способа сборки. Режимы: --payload, --dump, --catalog (все поля файла), --report ОТЧЁТ.docx --compare (поле за полем), --run (локальный расчёт против печатных величин, по этапам), --mesh файл (сетка из файла), --no-snap-angles (отключает округление φ слоёв из радиан до 1e-9°, которое включено по умолчанию; флаг --snap-angles принят и равен умолчанию), --dump-record КАТАЛОГ --report ОТЧЁТ.docx (одна запись набора gwx_projects, ниже) и --verify-records КАТАЛОГ (перевывод записей из оригиналов по sha256). Версия 8 читается в объёме файлов примеров меню «Справка» и контрольных задач ([0328-примеры-gwx], 07.10.2026): шпунт и стена в грунте (железобетон — только как эквивалент: суммы E·A и E·J брутто, без трещин; Км не считается), откопка справа и слева (зеркало), анкеры (полярный угол оси, F0, C, Fm, шаг, диапазон этапов), распорки «Другое» и трубчатые, бермы Trapeze, пионерный котлован pptTrapeze, вода «Простая» (слой скважины и водопонижение) и «Сложная» (слои сторон → блок aquifers), пригрузки и нагрузки на стенку, датум Z − ZTop + ZeroPoint.Y; remove_stage = Stage2Index + 2 (прежняя + 1 была ошибкой на единицу, подтверждено таблицей анкеров Примера 3; второй анкер раньше терялся из-за списка повторяющихся тегов). По одиннадцати файлам проектов проектировщика чтение расширено ([0332-проекты-0708], 08.10.2026): файлы версии 8 — двусторонняя откопка (IsDualExcavation: сторона котлована определяется по уровням этапов, срезка и подсыпка удерживаемой стороны переходят в stages[].backfill_depth, грунт подсыпки принимается верхним слоем геологии; равные уровни сторон — отказ), пионерный котлован с SoilID = −1 (допущение названо), слой воды «Простой» с напором HAdd (блок aquifers, допущение названо); файлы версии 7 — распорки и пионерный котлован (Main.PioneerType3/H/Otstup/Alfa). Таблица Berm файла версии 7 по-прежнему отказ: из 11 файлов строятся 10, файл 1655 Расчет 1.gw4 — названный отказ. Чего прототип не умеет, он называет вслух: GwxUnsupported с названиями ВСЕХ найденных полей (.reasons) — заданные перемещения, ручные коэффициенты давления, упругая модель грунта, подсыпка и засыпка и бермы на стороне тыла при односторонней откопке, роли сторон, меняющиеся между этапами, подсыпка стороны котлована, подсыпка тыла разными грунтами или с наибольшей срезкой не на первом этапе, пионерный котлован pptInfiniteAngle и на стороне откопки, анкеры с Fs ≠ 0 или направленные вверх или в сторону котлована, пригрузка через стенку, завесы, прочие типы стен, сейсмика, связанная вода; для файлов версии 7 — таблицы Force, DistForce, StrutJet, Anchor, Wallbase и Berm с данными, связи не-распорки, виды пионерного котлована кроме Type0 и Type3, ограждение не «Шпунт»; допущения, которых образцы не подтверждают, попадают в notes с пометкой «ДОПУЩЕНИЕ» (например, режим ppmStrictForce → «по силам» принят только по названию настройки). Юридическая граница: образцы изучались только чтением присланных файлов — без дизассемблирования программы, обхода защиты и чтения её документации. Пробный результат на двух расчётах (2139.2 из файла версии 8 и 1614.2 из файла версии 7; их нет среди 56 + 11 + 8 эталонов, статистику основных наборов и базы храповика они не меняют; с [0318-храповик-gwx] они образуют необязательный набор gwx_projects): M +1,3 / Q −2,7 / w −7,8 / Кз +6,6 / Км +7,6 % и +0,3 / +0,6 / +0,4 / −0,3 / +0,3 % — все пять в допуске

±10 % в обоих. Образцы и отчёты в git не лежат, тесты — на придуманных файлах: test_import_gwx.py (22; один — на реальных образцах, пропускается без GW_SAMPLES_DIR) и test_import_gwx_structures.py (26: 25 на синтетических файлах, один на реальных образцах при GW_SAMPLES_DIR), test_import_gwx_projects_0708.py (14; один — на реальных файлах при GW_PROJECTS_0708_DIR или GW_SAMPLES_DIR) и test_import_gwx_angles.py (28; в том числе вся сетка 0,01° от 0,01° до 60° проходит «градусы → радианы → импорт» без изменения). Примеры 1–3 и контрольные задачи меню «Справка» с настоящим входом из файлов считает scripts/gw_compare/manual_examples_gwx.py (файл → запрос → локальный расчёт → печатное число руководства; не входит в храповик и базы): пар «печатное число ↔ HydroWeb» в допуске ±10 % (зачёт — вариант «файл как есть», дефолты продукта) — Пример 1: 3 из 5, Пример 2: 12 из 16 (усилия анкеров 3 из 6), Пример 3: 3 из 24 (усилия связей 0 из 7; причину разбор относит к форме бермы в продукте); вход контрольных задач из файла совпал с входом прежних аналогов по всем 61 значению, но запрос к API не собирается ни у одной из девяти. Находка о ядре: шум представления угла (20° = 0,349065850398866 рад → 20,000000000000007°) переключает порог φ > 20° (core/coefficients.py), и Кз Примера 2 меняется на 5–16 % (M, Q, w — менее 0,2 %); как программа-эталон обрабатывает ф ровно 20°, неизвестно. С 08.10.2026 ([0337-углы-0708], решение владельца) импорт файлов версии 8 по умолчанию округляет ф слоёв до 1e-9°, если отличие от сетки — шум записи радиан (не более 1e-12°; в файлах 14–15 значащих цифр, наблюдённый шум не более $2{,}84 \cdot 10^{-14}$°): 20° остаётся 20°, набранные значения (17,5; 31,1; 37,7 …) не искажаются, значение с реальными цифрами за 1e-9° не округляется и называется в notes. Запрос с --no-snap-angles (snap_angles=False) побитно прежний. Углы пионерного котлована и берм округляются всегда, файлы версии 7 не затрагиваются (ф там в градусах, без перевода и шума), угол наклона анкера — round(…, 4), как раньше. Результат изменился только там, где слой стоит на пороге 20°: Кз двух расчётов проектировщика вырос с 3,750 до 4,012 и с 5,356 до 5,583; в остальных — не более $5{,}6 \cdot 10^{-7}$ % по любой из пяти величин, «все пять» и «в допуске ±10 %» нигде не изменились (замер scripts/gw_compare/research/snap_angles_measure.py, числа — results_snap_angles_2026-10-08/). Запись 2139.2 набора gwx_projects перевыведена с умолчанием (шум ф не более $2{,}8 \cdot 10^{-14}$° убран), запись 1614.2 не менялась. Ядро не менялось: порог ф > 20° и его чувствительность к шуму остаются, меняется только вход, который подаёт импорт.

- **Пробная сверка с семью расчётами из приложения раздела ПОС** `scripts/gw_compare/pos_sweep.py` (07.10.2026, [0312-эталоны-пос]). Семь полных расчётных отчётов эталонной программы (раскладка 7.х, версия 7.4.3) из приложения «Расчёт шпунтового ограждения» локального тома ПОС: одна распорка наверху стенки, два этапа, две полосовые нагрузки. Каталог данных (запросы и печатные величины) собирается из PDF скриптом, лежащим в самом каталоге, и в git не входит (сторонний документ); в репозитории — только счёт и таблица. pos_sweep.py считает локально по записям каталога и печатает «печатное / наше / Δ %» по пяти величинам с допуском ±10 % (--stages — по этапам, --variant / --all-variants — варианты запроса, --only, --set, --out). Особенность набора: столбец «ks» напечатан в кН/м⁴ — это коэффициент пропорциональности по слоям ($k_s = K \cdot z$), а продукт принимает одно K на запрос и отсчитывает z от дна котлована; поэтому у записи несколько вариантов запроса, а основной (эмуляция подслоями по 0,5 м, реконструкция входа) объявлен ДО расчёта. Результат основного варианта на дефолтах продукта: «все пять» в допуске у 6 из 7 (вне допуска только Кз одного расчёта, +10,3 %);

формальное чтение «ks как напечатано» — 0 из 7, z от дна котлована — 0 из 7. С [0316-k-по-слоям] продукт принимает K по слоям (layers[].k_prop и subgrade_z_origin, раздел «POST /api/calculate/fem»), поэтому добавлен нативный вариант native_surface: печатное «ks, кН/м⁴» → layers[].k_prop + ground_surface без деления слоёв (строится в скрипте из formal, сверка с печатной таблицей, иначе ValueError); --all-variants печатает блок «нативный против эмуляции». Предсказания порогов записаны и закоммичены ДО первого прогона (документы/сверка_расчётов/k_по_слоям_предсказания_2026-10-07.md); итог: «все пять» 6 из 7, как у эмуляции (вне допуска только Kз одного расчёта, +10,3 %). Это согласуется с набором, но не доказывает выбор «от поверхности»: предсказания не слепые (автору были известны итоги прототипа), вход — реконструкция, а положение начала отсчёта набор определяет лишь с точностью 0,2–0,3 м (при C = K·(z + δ): δ = +0,1 м → 4 из 7, 0 → 6, −0,2 м → 6, −0,3 м → 5, −0,4 м → 2; scripts/gw_compare/research/pos_datum_shift_scan.py); пионерного среза в наборе нет. Находка post hoc (причина не изучена, ядро не менялось): расщепление однородного слоя на подслои с тем же k само меняет результат на прежнем пути (M до 1,6 %, Kз до 4,7 %, w до 4,0 %). Выборка малая, расчёты одного типа и одной площадки: набор пригоден как пробный, не как гейт. К all_sweep.py и базам храповика НЕ подключён (решение владельца), в статистику трёх наборов не входит. Тест: test_pos_sweep.py (19).

Аудит стиля кода ([0487-чистка-кода], 24.10.2026)

Код, комментарии, тесты и скрипты проверены на следы процесса разработки и небрежность: обращения к участникам процесса вместо описания сути, ссылки на несуществующие документы и пути, значки и эмодзи, английские комментарии при русской конвенции, отладочный вывод и закомментированный код, хроника «раньше было, теперь», названия тестов-рассказов, мёртвый код и неиспользуемые импорты. Два прохода по трём областям (бэкенд, фронтенд, скрипты и инфраструктура) дали 336 записей; исправлены безопасные: комментарии и докстринги переписаны по сути (329 строк с обращениями к участникам процесса), значок предупреждения убран из серверных предупреждений, английские комментарии переведены, удалены отладочный вывод, функции без ссылок и неиспользуемые импорты. Удалены серверный генератор PDF (пакет hydroweb/reports/ с шаблонами, маршрут POST /api/report/pdf; extra [reports] заменён на [tools] для скриптов книг и проверки выгрузок) и выключатель FAST_ASSEMBLY вместе с прежней сборкой через lil_matrix. Поведение, числа расчёта, ключи API и тексты выгрузок не менялись: храповик сверки на 21 расчёте (105 значений) совпал с базой побитово, сторожа выгрузок и книг не ослаблялись. Позиции, которые меняют поведение или структуру модулей, в код не внесены и перечислены в отчёте документы/аудиты/аудит_стиля_кода_2026-10-24.md.

Непрерывная интеграция на push и pull request ([0481-ci-на-push])

Файл `.github/workflows/ci.yml`: на каждый push и pull request в `release/v1.1` и по ручному запуску (`workflow_dispatch`, любая ветка) выполняются те же ворота, что перед выкаткой. Раннер `ubuntu-24.04`, Python 3.12, Node 24, `actions/*@v7`, кэши pip и npm, `permissions: contents: read`, `concurrency` с отменой устаревших прогонов; секретов и выкатки в файле нет. Прежний `ci.yml` запускался с 31.07.2026, но не прошёл ни разу: PyYAML не был объявлен в `dev` (сбор тестов падал), база храповика от 05.10.2026 устарела, а задание краснело под `continue-on-error`.

Задание	Что делает	Предел
Ядро (pytest)	Системная библиотека poppler-utils (pdftotext для теста текстов книг); `pip install -c requirements.lock -e ".[dev,ingest]"`; сначала `pytest -m benchmark`, затем `pytest hydroweb/tests -q -p no:cacheprovider -rs`.	60 мин
Сверка с эталоном (pytest scripts/gw_compare)	Те же зависимости, Node 24 и `npm ci` (без них скипаются тесты «путь интерфейс → API»); `pytest scripts/gw_compare`.	30 мин
Фронт (typecheck, lint, vitest, build)	`npm ci`, `npm run typecheck`, `npm run lint` (блокирующий: замечаний 0), `npx vitest run --maxWorkers=2 --testTimeout=300000 --hookTimeout=300000`, `npm run build`.	30 мин
Согласованность (App.tsx, копии книг)	Ровно одно вхождение `key={`${activeTab}:${epoch}`}` в `App.tsx`; `cmp` копий `frontend/public/docs/{user,dev,supplier}.{html,pdf}` с книгами в `документы/руководства/`.	10 мин
Храповик сверки (ratchet-lite)	Подмножество `ratchet_lite.txt` (21 расчёт) по `refs_requests/` против базы `prod_sweep_local_2026-10-24_v4_newton_base.json`, `--jobs 2 --sum-slack 1.0`; без `continue-on-error`; итог — в сводку прогона.	40 мин
Уязвимости зависимостей	`pip-audit -r requirements.lock --no-deps` (исключения только строкой с причиной в `.github/pip-audit-ignore.txt`, сейчас их нет; pip-audit не фильтрует по тяжести, блокируется любая найденная уязвимость) и `npm audit --omit=dev --audit-level=high`, оба шага без `continue-on-error`: найденное краснит шаг и блокирует мерж.	15 мин

- **Чего нет на раннере.** Данных вне git (kb.db, ФГИС ЦС, эталонные отчёты, `рабочие_материалы`) на раннере нет: такие тесты скипаются сами по отсутствию файла или переменной, причина печатается в журнале (`-rs`); из прогона они не исключены. Это лестница опытов 8.4 в `test_blum_policy`, тексты приказов и архивы ФГИС ЦС, CSV сторонних установок, `kb.db`, файлы проектов, образцы и отчёты вне git в сверке.
- **Прогон 24.10.2026 (GitHub-hosted, четыре подряд, шесть заданий зелёные).** Ядро — 5350 passed, 65 skipped, 14 xfailed; `-m benchmark` — 27 passed, 4 skipped, 2 xfailed; сверка — 405 passed, 25 skipped; vitest — 268 файлов, 3216 passed, 5 skipped. Время: ядро 11–15 мин (весь набор 10–13 мин), фронт 9–12 мин (vitest 7–10 мин), сверка 1–2 мин, храповик около 30 с, согласованность около 10 с, аудит около 45 с; весь прогон 12–15 мин, суммарно около 30 раннер-минут.
- **Что исправлено в коде при вводе CI.** (1) `pyproject.toml`, `dev`: добавлены `pyyaml` и `lxml`. (2) FastAPI 0.162 ставится на раннере и в боевой образ, а локальная `.venv` — 0.136.1: `include_router` стал ленивым, `app.routes` содержит `_IncludedRouter`, и шесть сторожей (`test_refusal`, `test_strict_input_generative`, `test_strict_input_r4`, `test_api_checks_stability`) видели 25 маршрутов вместо всех; общий обход

`hydroweb/tests/_strict_bases.api_routes(app)` считает оба устройства, прогнан на 0.136.1 и 0.162.0. (3) `core/fem/newton_continuation.py`: ранний выход «нет нагрузки → U = 0 точно» срабатывал и при заданном перемещении `DirichletBC` $\bar{w} \neq 0$, если `load_scale` оказывался ровно 0.0 (так было на x86_64, на arm64 — шум около 1e-5); условие дополнено `not np.any(self.fixed_vals != 0.0)`, в продукте такие перемещения не используются, обычные расчёты прежние, тест — `test_prescribed_rigid_shift_is_not_mistaken_for_no_load`. (4) `test_slope_stability`: закреплённые на arm64 числа расходились на 1 ulp на x86_64, сравнение стало `pytest.approx(rel=1e-12)` (сухой режим по-прежнему сравнивается точно).

- **Осталось (решения владельца).** Обязательные проверки «Ядро», «Сверка», «Фронт», «Согласованность» в защите ветки `release/v1.1` (настройка репозитория). Минуты Actions: репозиторий приватный, один прогон около 30 раннер-минут, при лимите бесплатного тарифа 2000 минут в месяц это около 65 прогонов; при частых push возможен `paths-ignore` для `документы/**` и `рабочие_материалы/**` (кроме `документы/руководства/**`). Версии зависимостей Python закреплены в `requirements.lock`. Числа, закреплённые на arm64 побитно, на x86_64 могут расходиться в последнем разряде (найден один такой тест).

Настоящий type-check фронта — только tsc -b

Голый npx tsc --noEmit даёт 0 ошибок (project references → 0 файлов) и НЕ является настоящим type-check.

Проверяет типы только tsc -b — он же в скрипте npm run typecheck (добавлен в [0172]) и внутри npm run build. Полный чек-лист перед коммитом: pytest hydroweb/tests/ -q && (cd frontend && npm run typecheck && npm run lint).

Глава 10. Релиз и деплой

Версионирование. Единая версия в трёх местах (синхронизированы): VERSION, pyproject.toml (version=), FastAPI(version=…) — сейчас 1.1.0-alpha.

Контейнеризация. Docker + docker-compose + nginx; домен hydroweb.online; Yandex OAuth настроен на боевой redirect URI. Скрипт deploy/deploy.sh <server-ip>:

- собирает фронт локально (npm run build → frontend/dist/);
- синхронизирует через rsync (--delete; каталог backups и файл DEPLOYED_COMMIT на сервере исключены из удаления);
- снимает архив тома данных в /opt/hydroweb/backups (имя тома берётся у работающего контейнера; права 0600, владелец deploy; меньше 1 МБ — выкатка останавливается; хранение до 14 суток и 10 копий; на первой выкатке без контейнера шаг пропускается);
- приводит том данных к владельцу app (шаг [3b/5], deploy/fix_data_volume_owner.sh; абзац «Контейнер не от root» ниже);
- пересобирает контейнер (шаг 4 — deploy/remote_build.sh): сначала прежняя команда docker compose up -d --build на сервере; если сборка упала по лимиту запросов Docker Hub (в выводе toomanyrequests, too many requests или отдельное число 429 — случай 07.10.2026), выполняется один повтор с базовым образом из зеркала (по умолчанию mirror.gcr.io/library/python:3.12-slim) и громким предупреждением: зеркало отдаёт официальный образ, но обновлённый снимок тега. Базовый образ параметризован в Dockerfile (ARG PYTHON_IMAGE), docker-compose.yml передаёт PYTHON_IMAGE (по умолчанию прежний python:3.12-slim с Docker Hub; постоянный переход на зеркало без правки настроек демона — строка PYTHON_IMAGE=… в .env рядом с docker-compose.yml). Переменная PYTHON_IMAGE_FALLBACK задаёт другое зеркало, пустая строка отключает запасной путь; любая другая ошибка сборки — остановка с кодом ssh, повтора нет;
- проверяет здоровье: до 12 попыток доступность /api/profiles и контрольную задачу Б-1 (POST /api/blum-benchmarks/B1/run, all_passed). Проверка не затрагивает базу и МКЭ-путь; откат при провале не выполняется, скрипт завершается с кодом 1;
- пишет на сервере штамп DEPLOYED_COMMIT: PENDING <коммит> до сборки, затем <коммит> (с суффиксом -dirty при незакоммиченных изменениях) или FAILED; если образ собран с зеркала, к штампу третьим словом добавляется base=mirror, а в итоговой строке выкатки стоит предупреждение.

Контейнер не от root. Dockerfile создаёт пользователя app (uid и gid 1000), `USER app`: nginx и uvicorn работают не от root. nginx внутри слушает 8080 (было 80), основной конфиг `deploy/nginx.conf` без директивы user, pid в /tmp; docker-compose.yml пробрасывает `127.0.0.1:8080:${HYDROWEB_CONTAINER_PORT:-8080}` (порт на хосте прежний). При сборке образ проверяет импорт приложения и запуск nginx под app: ошибка останавливает выкатку до замены контейнера. deploy/entrypoint.sh не делает root-операций и останавливается с понятным сообщением, если том не принадлежит app. Шаг [3b/5] deploy/deploy.sh вызывает `deploy/fix_data_volume_owner.sh`: он один раз приводит том к uid 1000 (идемпотентно, во вспомогательном контейнере alpine). `docker exec` теперь по умолчанию от app. Откат на образ до перехода контейнера на пользователя app: `HYDROWEB_CONTAINER_PORT=80 docker compose`

`up -d --no-build --force-recreate` (deploy/rollback.md); архив тома, снятый до этого перехода, возвращает владельца root, после распаковки выполняют fix_data_volume_owner.sh.

Закреплённые версии. `numpy==2.4.4`, `scipy==1.17.1` ([0172]). Поднимать — только с перепрогоном эталонного набора. Прод (x86_64) и машина разработки (arm64) на near-limit схемах расходятся на 0,1–0,5 % — это арифметика архитектур, не версии библиотек.

Файл requirements.lock. Версии всех пакетов Python (прямых и транзитивных) закреплены в requirements.lock; это constraints: версии берутся из файла, состав задаёт pyproject.toml. Снят 24.10.2026 с чистой установки extras ai, reports, dev и ingest (extra reports удалён вместе с серверным генератором PDF, его заменил tools, поэтому в файле остались пакеты, которых в образе нет; лишние строки безвредны) (fastapi 0.162.0, starlette 1.10.0, pydantic 2.14.1, uvicorn 0.54.0, anthropic 2.3.0). Dockerfile и CI ставят с `pip install -c requirements.lock …`, разработчик — `pip install -c requirements.lock -e ".[dev,ingest]"` (dev — pytest и проверки, ingest — разбор DOCX и PDF базы знаний; скрипты книг и проверки выгрузок — дополнительно extra tools: WeasyPrint и matplotlib). test_requirements_lock.py держит согласованность файла с pyproject.toml и равенство версий ключевых пакетов в окружении тестов. Обновляют отдельной задачей; задание CI «Уязвимости зависимостей» проверяет именно этот файл (глава 9).

Резервное копирование. `deploy/backup_db.sh` — ночной логический дамп семнадцати пользовательских таблиц egrz.db (cron пользователя deploy, 03:15): users (учётные записи), projects, project_tombstones (надгробия удалённых проектов), consents (согласия на обработку данных с версией редакции документов), oauth_identities (привязки Яндекс/VK), admin_audit (журнал действий администратора, хранение 400 суток), supplier_items, supplier_profiles, supplier_demand и supplier_usage (кабинет поставщика; supplier_usage — сводка продукции поставщиков в проектах, [0498-продукция-поставщика-в-проектах]), price_requests, price_request_targets и price_offers (запросы цены), supplier_applications (заявки поставщиков; файлы карточек в дамп не входят, лежат в каталоге supplier_cards/ тома, их копирует полная копия), feedback (обращения «Написать нам», [0517-пробный-доступ-по-метке]), feedback_files (вложения обращений, до пяти на обращение, [0531-обращения-до-пяти-вложений]; сами файлы в дамп не входят, лежат в каталоге feedback_files/ тома, их копирует полная копия) и events (журнал событий беты, [0543-журнал-событий-беты]). До 08.10.2026 ([0335-надёжность-0708]) в дамп входили только users, projects и project_tombstones. Не входят: реестр ЕГРЗ (восстанавливается импортом XLSX), assistant_usage (суточный счётчик вопросов), calc_jobs (фоновые расчёты живут не дольше часа), login_attempts (счётчики неудачных входов живут не дольше 15 минут) и access_level_changes (журнал смен уровня доступа, [0548-уровни-доступа-шаг-1]: уровень лежит в users и копируется, журнал с причинами — только в полной копии тома); полная копия тома, включая kb.db, — deploy/backup_data.sh (раздел «Эксплуатация» ниже). Дамп делается внутри контейнера из соединения только для чтения; каждая копия проверяется до того, как получает своё имя: она загружается во временную базу, число строк каждой таблицы сверяется с итоговыми строками дампа (-- rows <таблица> <n> после последнего COMMIT), и только затем временный файл переименовывается. Содержимое таблиц в вывод не попадает — только имена таблиц и числа строк. Хранение 30 суток в ~deploy/backups/hydroweb/; имя файла users_projects_*.sql.gz историческое, по нему идёт

ротация, состав таблиц от него не зависит. Перед каждой выкаткой — ручной запуск того же скрипта.

Восстановление. `deploy/restore_db.sh <файл.sql.gz> [--yes]` заменяет ровно те таблицы, которые есть в дампе (перед каждой в дампе стоит DROP TABLE IF EXISTS): копия, снятая после 08.10.2026, возвращает все таблицы своего списка, а копия прежнего состава не стирает таблицы, которых в ней нет (consents, oauth_identities, admin_audit, таблицы кабинета поставщика и запросов цены, feedback, events). Перед заменой скрипт делает свежую копию текущего состояния; после заливки удаляются надгробия возвращённых проектов (иначе их нельзя было бы сохранить, 410) привязки провайдеров к несуществующим пользователям (иначе повторный вход пользователя, зарегистрированного после снятия копии, упирается в занятый ключ) и строки supplier_usage проектов, которых в восстановленной projects нет (иначе они считались бы в «проектах» поставщика без владельца), и строки feedback_files, у которых в восстановленной feedback нет обращения (копия прежнего состава, [0531-обращения-до-пяти-вложений]). Согласия и журнал после заливки не чистятся: это записи о совершённых действиях. Журнал событий (events) скрипт тоже не чистит: события пользователей, которых в восстановленной users нет, снимает приложение при старте процесса (_DATA_MIGRATIONS), а до того в критерии они не идут (сводка берёт только существующих пользователей). Найденная ошибка: до 08.10.2026 restore_db.sh данные не восстанавливал — дамп шёл по конвейеру, а сам скрипт передавался heredoc-ом на тот же stdin, heredoc перекрывал конвейер, sys.stdin.read() возвращал пустую строку, и скрипт печатал счётчики текущей базы и «готово». Теперь скрипт передаётся аргументом python -c, дамп идёт по конвейеру, пустой вход — ошибка с сообщением. Дефект воспроизведён на прежнем скрипте с подменённым Docker; скрипты проверены тестами test_backup_scripts.py и test_remote_build.py, а 08.10.2026 — и вручную на настоящем Docker на сервере (врезка «Проверка на сервере 08.10.2026» ниже). Запасной путь шага 4 выкатки при лимите Docker Hub на живом сервере не проверялся.

Эксплуатация: копия тома, проверка восстановления, монитор прода, аудит зависимостей

Четыре средства надёжности добавлены 07.10.2026 ([0308-надёжность]). Расчётное ядро и расчётные числа они не затрагивают. Сами скрипты на сервере ничего не устанавливают: они приезжают с обычной выкаткой (deploy/ синхронизируется rsync, каталог backups из удаления исключён), включение по расписанию — записи в crontab пользователя deploy на сервере, которые делает владелец, после пробного запуска. Расписание на сервере (проверено 08.10.2026): 03:15 — backup_db.sh, 03:30 — backup_data.sh, по воскресеньям 04:30 — restore_check.sh.

- `deploy/backup_data.sh` — суточная копия ВСЕГО тома данных (egrz.db с пользователями и проектами, kb.db), в отличие от backup_db.sh, который выгружает семнадцать пользовательских таблиц. Том определяется у работающего контейнера hydroweb по точке монтирования /app/data (имя тома не зашито: docker compose создаёт его с префиксом проекта), архив снимается тем же способом, что в шаге [3/5] deploy.sh (docker run … alpine tar czf). Архив пишется во временный файл .hydroweb-daily-….part и становится «хорошей» копией hydroweb-daily-ГГГГММДД-ЧЧММСС.tar.gz в /opt/hydroweb/backups только после проверок: размер не меньше 1 МБ и tar tzf читает архив; иначе временный файл удаляется, код возврата 1, запись в backup.log. Права 0600. Хранение — не более 7 копий и не старше 14 суток, ротация идёт только после успешной новой копии (сломанное копирование не

стирает старые); копии deploy.sh с префиксом hydroweb-data- не затрагиваются. Одна успешная копия в сутки (повторный запуск — «ПРОПУСК», --force снимает ещё одну); от параллельного запуска защищает каталог-замок .backup_data.lock (замок умершего процесса снимается сам). Рекомендованная запись cron в шапке скрипта: 30 3 * * * (ежесуточно в 03:30 по времени сервера), вывод — в /opt/hydroweb/backups/backup.log.

- `deploy/restore_check.sh` — проверка восстановления без вмешательства в боевой том: берёт последнюю hydroweb-daily-* (или файл аргументом), проверяет размер, возраст (старше 36 часов — провал: копирование молча встало), распаковывает во временный каталог рядом с копиями (не в /tmp: там может быть tmpfs; каталог переопределяется WORK_DIR), проверяет наличие egrz.db и kb.db и выполняет PRAGMA integrity_check каждой *.db (у egrz.db ещё наличие всех семнадцати пользовательских таблиц — REQUIRED_TABLES: users, projects, project_tombstones, consents, oauth_identities, admin_audit, supplier_items, supplier_profiles, supplier_demand, supplier_usage, price_requests, price_request_targets, price_offers, supplier_applications, feedback, feedback_files, events, тот же состав и порядок, что TABLES в backup_db.sh, — и число строк в каждой; выводятся только числа, содержимое не читается, поэтому пустую, но целую копию видно по нулям); временный каталог удаляется, в том числе при ошибке. Базы проверяются python3 на хосте или образом приложения (CHECK_MODE=auto|host|docker). Код возврата 0 — копия пригодна, 1 — нет. Запускается раз в неделю, рекомендованная запись: 30 4 * * 0 (воскресенье, через час после копирования). Распаковка занимает столько места, сколько том (egrz.db около 2,4 ГБ): перед первым запуском посмотреть df -h /opt/hydroweb.

Ограничение копии тома

Том архивируется «на лету», без остановки контейнера (так же, как в deploy.sh): база SQLite, в которую в момент копирования идёт запись, может попасть в архив «рваной». Именно это и ловит restore_check.sh (PRAGMA integrity_check) — поэтому его запускают по расписанию после копирования, а не только после аварии.

Проверка скриптов без Docker — hydroweb/tests/fixtures/fake_docker.sh (подмена docker, понимает только вызовы этих скриптов) и hydroweb/tests/test_backup_scripts.py (53 теста: права 0600, малый архив отброшен, ротация 7 копий и 14 суток по счёту и по возрасту порознь, замок, «раз в сутки», битая база, нет kb.db, устаревшая копия; дамп всех семнадцати пользовательских таблиц и сверка числа строк с итоговыми строками, восстановление дампом нового и прежнего вида — в том числе снятым настоящим прежним скриптом, привязки без пользователей, согласованность списков таблиц в трёх скриптах и др.); шаг 4 выкатки с запасным путём — hydroweb/tests/test_remote_build.py (25 тестов с подменёнными ssh, а также весь deploy.sh с подменёнными npm, rsync, ssh и sleep). Подмена не заменяет настоящий Docker: те же скрипты прогнаны вручную на сервере 08.10.2026 (врезка ниже); в автоматические тесты эта проверка не входит.

Проверка на сервере 08.10.2026 ([0345])

Каталог deploy/ выкачен на сервер отдельно от приложения 08.10.2026 около 13:30 по Москве: rsync deploy/ без --delete, шесть файлов — backup_data.sh, backup_db.sh, deploy.sh, remote_build.sh (новый), restore_check.sh, restore_db.sh. Контейнер и приложение не пересоздавались, штамп DEPLOYED_COMMIT не менялся. Проверено на сервере:

1) bash -n всех скриптов — без ошибок; расписание cron прежнее (03:15, 03:30, воскресенье 04:30).

2) Ручной запуск backup_db.sh: «копия проверена … строк: users 25, projects 85, project_tombstones 1, consents 0, oauth_identities 0, admin_audit 0». consents = 0: в боевой базе один пользователь создан после даты введения согласий, остальные — через приглашения.

> 3) Проба restore_db.sh — не на боевой базе: в контейнере создана пробная база /tmp/probe.db с одним пользователем, команда DB_IN_CONTAINER=/tmp/probe.db BACKUP_DIR=/tmp/probe_backups restore_db.sh <копия> --yes заменила все шесть таблиц, пробная база стала users 25 / projects 85 / project_tombstones 1 (строка пробы заменена), боевая egrz.db не изменилась (те же 25 / 85 / 1); пробная база и каталог копий удалены. Это первый запуск restore_db.sh на настоящем Docker и подтверждение исправления ошибки с heredoc (раньше данные не заливались).
>
> 4) restore_check.sh вручную на архиве тома hydroweb-daily-20261008-033001.tar.gz (330 МБ): «OK egrz.db — integrity_check ok, users=25, projects=85, project_tombstones=1, consents=0, oauth_identities=0, admin_audit=0; OK kb.db»; временный каталог удалён.
>
> Не проверено: запасной путь шага 4 выкатки при лимите Docker Hub на живом сервере.

- `scripts/maintenance/prod_health.py` — проверка боевого сервиса без участия модели: только чтение, зависимостей нет (стандартная библиотека). Девять проверок: главная страница; /docs/user.html; /docs/dev.html; /api/estimate/zones (доступно и число ценовых зон не меньше порога --min-zones; значение по умолчанию в скрипте DEFAULT_MIN_ZONES — 147 (снимок цен 3 квартала 2026: 147 из 148 зон; до [0315-выкатка] стояло 27 — число зон прежнего снимка; теперь запуск без аргументов проверяет актуальное число зон)); четыре расчёта POST /api/estimate/calc (ГЭСН 05-01-012-10, 1 т, ресурс 08.3.10.02 → 08.3.10.02-0011) для зон Москва (191), Московская область (127), Санкт-Петербург (206), Ленинградская область (162) — итог ЛСР сверяется с ожидаемой суммой, расчёт должен быть полным; POST /api/blum-benchmarks/B1/run — all_passed. Выводит таблицу ok / FAIL, код возврата 1 при любом провале, --notify показывает уведомление macOS через osascript (текст передаётся аргументами, не вклеивается в код). Запускается вручную: python3 scripts/maintenance/prod_health.py [--url …] [--min-zones 147] [--notify]; пример расписания launchd — в докстринге скрипта, автоматически он не устанавливается. Ожидаемые суммы (таблица EXPECTED_TOTALS) привязаны к снимку цен 3 квартала 2026: с выходом нового квартала они меняются законно, и проверка покажет провал — таблицу обновляют осознанно после сверки с новым снимком, а не подгоняют. Python из Homebrew на машине разработки не видел корневых сертификатов (все запросы падали с CERTIFICATE_VERIFY_FAILED, хотя curl работал): проверку сертификата скрипт не отключает, при пустом хранилище он подгружает системный набор. Тесты — hydroweb/tests/test_prod_health.py (15, HTTP подменён).
- `scripts/maintenance/cleanup_worktrees.sh` — уборка рабочих копий git и слитых веток после релиза ([0538-уборка-защита-копий], решение владельца 29.10.2026). Необратима, поэтому запускается владельцем вручную из корня проекта (bash scripts/maintenance/cleanup_worktrees.sh). Несохранённые правки старых копий .claude/worktrees/* кладёт в _корзина/<дата>/ (diff, статус, архив неотслеживаемых файлов); ветку claude/great-curran-35cee9 сохраняет как archive/great-curran-newton-1014; удаляет копии, чья ветка слита в release/v1.1 и НЕ стоит на его вершине, и копии без ветки (detached); копии с неслитой или свежей веткой остаются и перечисляются как «пропущена»; из локальных веток удаляет полностью слитые (кроме release/v1.1, archive/* и веток на его вершине), ветки, занятые рабочей копией (знак «+» у git branch), не трогает, список лежит в _корзина/<дата>/. Правило «не на вершине» появилось после второго запуска 29.10.2026: ветка, только что созданная от release/v1.1, стоит на его вершине, и для git она «слита», хотя это идущая работа; запуск снёс копии и ветки двух только что начатых разборов (коммитов в них ещё не было, агенты пересоздали копии). Проверено пробным

прогоном с заменой удаления печатью: копия с коммитом и свежая копия пропущены, давно слитая пробная ветка выбрана.

Секреты окружения лежат только в /opt/hydroweb/.env на сервере и в git не попадают: HYDROWEB_JWT_SECRET, YANDEX_CLIENT_SECRET, ключи YandexGPT. Пустой YANDEX_CLIENT_SECRET = вход через Яндекс отвечает 500 «not configured».

> **Состояние сборки**
>
> npm run build (= tsc -b && vite build) проходит начисто: 0 ошибок TypeScript, сборка успешна (exit 0, измерено 08.10.2026). Ранее документированные 124 pre-existing ошибки устранены.
>
> npm run lint проходит начисто: 0 ошибок и 0 предупреждений (eslint . по 390 файлам, измерено 08.10.2026); npm run typecheck (tsc -b --noEmit) — 0 ошибок. Линт не входит в build, но в CI это отдельный блокирующий шаг задания «Фронт»; его запускают и перед коммитом (чек-лист выше).
>
> Настоящий type-check — только tsc -b: голый npx tsc --noEmit компилирует 0 файлов (project references) и всегда даёт 0 ошибок.

Импорт ЕГРЗ (разово, ~565K записей в SQLite + FTS5):

```
python scripts/import_egrz.py путь/к/реестр.xlsx
```

Вход под нагрузкой: блокировка БД и её устранение ([0343])

Повод — оценка ёмкости платформы для учебного класса из 25 человек ([0334]): при одновременном входе часть пользователей получала HTTP 500 «database is locked», а запрос /api/auth/me уже вошедшего пользователя ждал секунды. Расчётное ядро (hydroweb/core) правка не затрагивала.

Причина. Каждое соединение с БД (get_async_db) выполняло схему и миграции, в том числе data-миграцию UPDATE users SET email_verified = 1 WHERE email_verified = 0 AND created_at < '2026-07-28T00:00:00' (пользователи, заведённые до введения проверки адреса, считаются подтвердившими его). Перед UPDATE модуль sqlite3 открывает транзакцию, и соединение держит блокировку записи до своего commit или close; у входа между открытием соединения и commit стоял синхронный bcrypt (0,38 с на машине замера), который к тому же останавливал цикл событий процесса. Входы выстраивались в очередь за одной блокировкой с ожиданием 5 с (умолчание sqlite3): 5 / 0,38 ≈ 13 успешных, остальные — 500. Эксперимент на временной базе (второе соединение держит блокировку записи, у пробующего ожидание 0,2 с): из шагов прежнего открытия соединения блокировку берёт только этот UPDATE; схема на готовой базе (1 мс), ALTER «столбец уже есть» (сразу «duplicate column name») и SELECT её не берут.

Что изменено.

- Схема и миграции — при старте процесса, соединение запроса записи при открытии не делает (раздел «Схема БД: что добавил релиз», глава 3). Ожидание блокировки — 30 с (BUSY_TIMEOUT_S, PRAGMA busy_timeout = 30000).
- bcrypt — вне цикла событий: `hash_password_async` и `verify_password_async` в api/auth.py (asyncio.to_thread, пул asyncio по умолчанию, не пул расчётов). Места: регистрация, вход (и пустышка для несуществующего адреса), сброс и смена пароля, пароль администратором, приглашение, удаление учётной записи из кабинета, вход через Яндекс/VK. Синхронные hash_password и verify_password оставлены для скриптов и тестов. bcrypt отпускает GIL

(замер: 4 проверки по очереди — 1,53 с, в 4 потоках — 0,38 с), поэтому проверки идут параллельно на разных ядрах.

- Лимиты входа, регистрации и смены пароля — из окружения с прежними умолчаниями; переменные нагрузки и лимитов проброшены в контейнер docker-compose.yml (раздел «Переменные окружения», глава 2).
- Последовательности «проверка → запись» прежде неявно шли под блокировкой соединения, теперь нет; все такие места в обработчиках разобраны. Регистрация одного адреса двумя запросами сразу даёт прежний ответ 400 (перехват IntegrityError); пароль администратором для пользователя, удалённого за время расчёта хеша, — 404; сохранение проекта (сравнение revision и перехват IntegrityError), удаление проекта (одна транзакция), счётчик ассистента (одна инструкция) и удаление учётной записи (BEGIN IMMEDIATE) гонку уже учитывали. Не менялось и допускается: смена роли и активности администратором при одновременной правке одного пользователя (в журнал действий может попасть устаревшее «старое значение»), смена согласия на рассылку двумя одновременными нажатиями (лишняя строка в журнале согласий), первый вход одного нового пользователя через провайдера в двух окнах (второе окно получит ошибку входа через провайдера вместо привязки). Нагрузочно эти случаи не проверялись.
- WAL не включён. Дамп backup_db.sh (mode=ro) в режиме WAL видит строки из -wal (в опыте 300 из 300), но копия тома tar-ом (backup_data.sh, шаг 3 deploy.sh) читает egrz.db и egrz.db-wal в разные моменты: при контрольной точке TRUNCATE между чтениями копия проходит integrity_check, но молча не содержит зафиксированных строк (в опыте потеряны 300 из 300), при контрольной точке RESTART с новой записью — «database disk image is malformed». Условие включения WAL — копия тома через SQLite (Connection.backup() или VACUUM INTO во временный файл тома, затем tar) либо при остановленном контейнере, плюс замер времени и места на базе 2,4 ГБ; это правка backup_data.sh и deploy.sh, отдельная задача. Результат без WAL измерен (таблица ниже); для класса из 25–40 человек по этим замерам достаточен, на боевом сервере не проверялось.

Замеры до и после. «До» — код a875887, «после» — код [0343]; скрипты замера одни и те же.

Показатель	До	После
Залп из 25 входов (class_login_burst.py, три прогона): ответы 200 / 500	14 / 15 / 15 успешных, 11 / 10 / 10 ответов 500	25 / 25 / 25 успешных, 0 ответов 500
То же: задержка, медиана / максимум, с; длительность залпа, с	5,1–5,4 / 7,9–9,3; 7,9–9,4	1,1–1,3 / 1,5–1,7; 1,5–1,8
Залп из 40 входов	13 успешных, 14 ответов 500, 13 таймаутов клиента (60 с)	40 из 40, медиана 1,3 с, максимум 2,5 с, залп 2,6 с
classroom_load.py login: залп 25, три повтора, успешных	15, 15, 14 (остальные — 500)	25, 25, 25
GET /api/auth/me вошедшего пользователя во время залпа: p50, с (число ответов за залп)	2,2–9,4 (1–4)	0,005–0,006, максимум 0,05–0,13 (6–7)
Поток 3 входа/с и 5 входов/с	3 вх/с: 24 успешных и один 500, p95 3,5 с; 5 вх/с: 21 успешный и четыре 500, p95 5,9 с	25 успешных в обоих случаях, p95 0,40 с и 0,39 с

Показатель	До	После
Записи «database is locked» в журнале сервера за прогон classroom_load.py login	36	0
classroom_load.py onboarding, 25 пользователей с одного адреса: вход 200 / 429 / 500; готовы	8 / 5 / 12; 8 из 25	20 / 5 / 0; 20 из 25 (с HYDROWEB_LOGIN_RATE_LIMIT=30 и HYDROWEB_CHANGE_PASSWORD_RATE_LIMIT=30 — 25 / 0 / 0, готовы 25 из 25 за 4,5 с)
Открытие соединения: медиана, мс; соединение после открытия в транзакции	1,71; да	0,28; нет (старт init_db — 1,2 мс, изменено строк 0)

Оставшиеся 5 ответов 429 в onboarding при умолчаниях — лимитер входа (10 входов на процесс за 5 минут, процессов два), а не БД. Условия замера: macOS arm64, 8 ядер, локальный uvicorn с двумя процессами, временная база, loadavg 4–7 (на машине шли другие прогоны); открытие соединения и старт измерены на APFS-клоне базы разработчика 2,4 ГБ (1933 пользователя, миграция уже применена, журнал отката), не на боевой.

Риски и что не проверено.

- Ожидание 30 с вместо 5 с: запрос, прежде получавший 500 через 5 с, теперь ждёт до 30 с. В режиме журнала отката долгий читатель (поиск по ЕГРЗ — секунды) не даёт писателю зафиксировать изменения, а ожидающий писатель не пускает новых читателей. Не измерялось; nginx ждёт ответа 300 с, обрыва на прокси не будет.
- Потоки bcrypt: пул asyncio по умолчанию — min(32, число ЦП + 4) потоков на процесс (в контейнере os.cpu_count() показывает ЦП хоста). На сервере с 1–2 vCPU параллельно считаются 1–2 проверки, остальные ждут в очереди пула, цикл событий свободен. Оценка для 25 входов при 0,76 с на bcrypt и двух ядрах — последний вход через ≈ 9,5 с без ошибок: это расчёт, не замер.
- Миграции на старте при существующих данных проверены на синтетической «старой» базе (результат совпал с прежним путём строка в строку) и на клоне базы разработчика (старт 1,2 мс, 0 изменённых строк); на боевой базе не проверялись. Если старт не смог выполнить миграции, процесс работает и повторит их на первом соединении — с записью DB_MIGRATION / DB_INIT в журнале.
- Не проверялись: Linux/x86_64, приложение под настоящим Docker и nginx под нагрузкой (замеры — macOS arm64, uvicorn напрямую), сервер с 1–2 vCPU, 30-секундное ожидание при долгом поиске по ЕГРЗ.

Проверки при выкатке этого изменения. Перед выкаткой на сервере смотрят, не заданы ли в .env переменные нагрузки и лимитов (после пересоздания контейнера они начнут действовать); предварительная миграция схемы не нужна — схема не менялась. Выкатка обычная (deploy/deploy.sh), новые переменные попадают в контейнер при его пересоздании в шаге 4. После выкатки:

```
# до выкатки: что уже записано в .env (найденное начнёт действовать)
grep -E -e '^HYDROWEB_MAX_CONCURRENT_CALCS' -e '^HYDROWEB_MAX_ACTIVE_JOBS' \
     -e '^HYDROWEB_ASSISTANT_DAILY_LIMIT_STANDARD' \
```

```
     -e '^HYDROWEB_(LOGIN|REGISTER|CHANGE_PASSWORD)_RATE_' /opt/hydroweb/.env
# после выкатки: переменные в compose и в контейнере
docker compose config | grep -E 'RATE_|MAX_CONCURRENT|MAX_ACTIVE|ASSISTANT_DAILY'
docker compose exec hydroweb env | grep -E \
     -e 'HYDROWEB_(LOGIN|REGISTER|CHANGE_PASSWORD)_RATE' -e 'MAX_CONCURRENT' \
     -e 'MAX_ACTIVE' -e 'ASSISTANT_DAILY'
# в журнале должно быть пусто
docker compose logs hydroweb | grep -E 'DB_INIT|DB_MIGRATION|database is locked'
# код образа новый: 30.0 (читает только константу, к базе не обращается)
docker compose exec -T hydroweb python -c \
     "from hydroweb.db.connection import BUSY_TIMEOUT_S; print(BUSY_TIMEOUT_S)"
```

Затем — вход в продукт, GET /api/auth/me и GET /api/admin/health (число пользователей и проектов — как до выкатки). Откат — выкатить предыдущий коммит тем же deploy.sh: совместимо в обе стороны, схема и данные не меняются, новые переменные прежний код не читает. Нагрузочную проверку на боевом сервере не делают. Полный разбор, команды воспроизведения замеров и список непроверенного — документы/разработка/блокировка_бд_при_входе_2026-10-08.md и запись [0343-блокировка-входа-0708] в CHANGELOG_DEV.md.

Сборка и публикация руководств

Изменили то, что видит пользователь, поставщик или разработчик, — обновите руководства в той же выкатке. Скрипты — scripts/books/, готовые книги — документы/руководства/, рисунки к ним — документы/руководства/рисунки/ (экраны, схемы, формулы, фирменный_стиль), копии для сайта — frontend/public/docs/ (имена user.*, dev.* и supplier.* менять нельзя: на них ссылается интерфейс).

```
# 1) рисунки интерфейса (серверы на пустой временной БД, кадры, OCR)
.venv/bin/python scripts/books/docs_capture_all.py
.venv/bin/python scripts/books/docs_capture_supplier.py   # рисунки кабинета
поставщика (36, 38, 45, 48-50) и меню 19а
.venv/bin/python scripts/books/docs_capture_requests.py   # диалог «Запрос цены» с
согласием на передачу данных (44)
# рисунки 62-66 (сводка продукции поставщиков в проектах) сняты прототипом
scripts/prototypes/supplier_usage_capture.py на пустой временной БД
# 2) книги в DOCX (docx берётся из frontend/node_modules)
node scripts/books/book_user_guide.js
node scripts/books/book_dev_guide.js
node scripts/books/book_supplier_guide.js   # заодно пишет
база_знаний/источники/supplier_guide.md
# 3) DOCX -> PDF и HTML, по одному процессу LibreOffice за раз
cd документы/руководства
soffice --headless --convert-to pdf --outdir . <книга>.docx
soffice --headless --convert-to 'html:HTML (StarWriter)' \
        --outdir . <книга>.docx
# 4) картинки в HTML, книжная раскладка (docs_style_html.py),
#    побайтные копии в frontend/public/docs/{user,dev,supplier}.{html,pdf}
python scripts/books/docs_inline.py
# 5) убрать PNG рядом с HTML и сверить копии
rm документы/руководства/*_html_*.png
cmp документы/руководства/<книга>.html frontend/public/docs/<user|dev|
supplier>.html
```

- Размеры PNG для docx-js лежат в документы/руководства/рисунки/манифест.json; формулы и схемы пересоздают docs_formulas.py, docs_diagrams.py, docs_figures.py, логотипы — docs_brand.py.

- Сырой экспорт LibreOffice на сайт не публикуют: шаг docs_style_html.py возвращает книжную раскладку (боковое оглавление, обложка, врезки, таблицы, адаптив, печать), без внешних ресурсов. Проверять книги надо не только текстом: названия и баннеры бывают на картинках — отсюда распознавание текста на каждом PNG.
- **Встроенный скрипт и Content-Security-Policy.** docs_style_html.py вписывает в каждую книгу один и тот же скрипт оглавления, а политика в deploy/nginx-site.conf допускает его по хешу sha256. Изменили скрипт — изменится хеш, и test_nginx_site_conf.py падает, пока хеш в конфиге не обновлён. Текст книг на хеш не влияет.
- Забытая копия на сайте — типичная ошибка: с 22.07 по 21.09.2026 сайт показывал июльские руководства, потому что обновили только документы/руководства/.
- Равенство копий на сайте книгам проверяет задание «Согласованность» в CI (cmp шести файлов: html и pdf трёх книг); тест bookSync.test.ts требует, чтобы каждая длинная фраза исходника книги была в опубликованном HTML.

Глава 11. Технический долг

Полная хронология — `CHANGELOG_DEV.md` (записи [NNNN]). Текущий фокус:

Воды, узлы, кабинет поставщика, импорт (24.10.2026)

- Воды, узлы, поставщик, импорт (24.10.2026): подбор насосов, схема несовершенного котлована и напорный горизонт не выведены в продукт (нет пункта источника); в узлах открыты пункты «уточнить» (раздел о проверке узлов), в кабинете поставщика — марки стали поставщиков в списке проектировщика (согласие на передачу данных и условия кабинета оформлены редакцией 2026-10-24, она проект до утверждения юристом); импорт не открывает 2 из 51 файла проектировщика (причины печатаются пользователю).

Линии 24.10.2026 на интеграции

- Подробности и открытые вопросы — в главе 8 (разделы по записям [0475]–[0480]) и главе 9 (раздел о CI, [0481]). Главное открытое: тексты редакций 2026-10-24 и 2026-10-26 не утверждены юристом (списки вопросов — документы/правовые_документы/редакция_2026-10-24/README.md и документы/правовые_документы/редакция_2026-10-26/README.md, раздел 8 в обоих); удаления и очистки запросов цены нет (экран администратора только показывает запросы, [0528]); подсчёт спроса поставщика считает позицию в запомненном варианте выбранной; подстановка чисел в формулу карточки «Откуда это число» требует нового ключа ответа; отметки выгрузок в проекте не хранятся (размеры котлована в плане с [0490] хранятся); письма по событиям кабинета поставщика не хранятся в базе и не повторяются при сбое почты (глава 8, подраздел о письмах); обязательные проверки в защите ветки release/v1.1 и лимит минут Actions — у владельца.

Безопасность (аудит [0020])

- Закрыто: сессия в httpOnly-cookie ([0178]–[0179]), ограничение частоты входа, регистрации и смены пароля (api/rate_limit.py; пределы настраиваются окружением, счёт — на один процесс uvicorn, [0343]).
- Закрыто 24.10.2026 (аудит кибербезопасности): срок сессии 14 суток со скользящим продлением, выход отзывает сессию на сервере, блокировка входа по учётной записи, блокирующая Content-Security-Policy, контейнер не от root, requirements.lock и блокирующий аудит зависимостей в CI (глава 8, раздел «Сессия и безопасность»; главы 9 и 10).
- Открыто: локальный список проектов (localStorage) общий для всех учётных записей браузера — сервер не даёт присвоить чужой проект (409), но разделять список по пользователям нужно на клиенте.

Архитектура

- Двойное хранение проектов: клиент (localStorage) ↔ сервер (таблица projects); нужен явный источник истины и стратегия синка.
- SQLite не масштабируется на много воркеров (single-writer) → миграция на PostgreSQL (проектировать с учётом). Режим WAL не включён ([0343]): пока копия тома снимается tar-ом «на лету», он грозит копией без зафиксированных данных; условие включения — раздел «Вход под нагрузкой» главы 10.

- App.tsx крупный, состояние в useState; рефакторинг на стор — сознательно отложен.

Качество и процессы

- Юнит-тесты фронта покрывают не весь код (vitest, 2781 тест в 222 файлах на дереве раунда 2 чистки выгрузок 21.10.2026, 2688 тестов в 217 файлах — на дереве первого раунда 20.10.2026, 1551 тест в 157 файлах на дереве ветки внедрения 15.10.2026, 1500 в 152 файлах — на релизе 08.10.2026: список проектов, подбор решений, выгрузки документов и смета, ассистент, админка, кабинет, импорт и экспорт, меню, вкладки, результаты, проверки, сторож «поле интерфейса → запрос»; чертёж — точечно). CI есть: .github/workflows/ci.yml, на каждый push и pull request в release/v1.1 шесть заданий — ядро (pytest), сверка (pytest scripts/gw_compare), фронт (typecheck, lint, vitest, build), согласованность (ключ вкладок в App.tsx и копии книг), храповик сверки и аудит зависимостей; все шесть блокирующие (раздел «Непрерывная интеграция на push и pull request» главы 9).
- Замечаний eslint сейчас нет (0 ошибок и 0 предупреждений, 390 файлов, измерено 08.10.2026): ранее документированные 94 пре-существующих замечания (в основном no-explicit-any и неиспользуемые объявления) устранены. Сборку (build) новые замечания не ломают, но линт — блокирующий шаг задания «Фронт» в CI, и чек-лист перед коммитом его тоже запускает.

Документы и смета

- Снимок цен ФГИС ЦС покрывает 147 из 148 зон (нет зоны 245 — ЯНАО, 5 зона: во ФГИС ЦС нет периода 427; для неё смета отклоняется; у зон 251 и 261 в самих данных источника у двух материалов нет цены и индекса — расчёт неполный; снимок в памяти ≈ 315 МБ). ВЗиС и зимнее удорожание считаются только по параметрам с процентом и основанием пользователя (текст Методики 332/пр в источниках не загружен — пункты по открытым копиям; базы процентов помечены допущениями), температурная зона заведена только для 4 зон снимка из 147; оборачиваемость шпунта в сметном движке РИМ (в ручном «Расчёте стоимости» число оборотов вводится и считается по ГЭСН п. 1.5.6), коэффициенты на условия производства, оборудование и ряд пунктов Методики 421/пр не реализованы и перечислены в not_accounted; нормы распорок и поясов (05-01-014-02) применяются с оговорками, нормы вне сборников 05 и 09 (сб. 29, 39-01-021) в книгу не загружены; соответствие «ценовая зона — графа НР» и правило границ классов норм («до N» / «свыше N») в источниках не найдены — допущения.
- Выгрузки: десятичная запятая введена в текстовых DOCX и в тексте ячеек XLSX (числа-значения ячеек и XML остаются машинными — точка); в XLSX основная надпись одна — форма 3 в конце листа, форма 6 на последующих листах не воспроизводится (примечание на каждом листе-форме); документы тома в ПЗ (содержание, состав ПД, таблица регистрации изменений) выгружаются с пустыми полями, в справке ГИП фамилию, подпись и дату заполняет проектировщик (редакция постановления № 985 взята по образцу и не проверялась); по разбору образца тома проектной документации не сделано: «Общие данные» рабочей документации (ведомость рабочих чертежей, ведомость спецификаций, общие указания), ведомости чертежей, спецификация и примечания на листе чертежа, узлы и «Спецификация на изделие», вынос шпунта на отдельный лист спецификации и точность масс 0,01 т, строки «Часть», «Том», «Корректировка №» и роль ГИП на титуле и в основной надписи, автопроверка единства марки стали и шифров по пакету; в штампе чертежа не

выводятся графа 27 и «Согласовано»; рамка листа проверена в LibreOffice, не в Word; полноценный PDF (конвертация DOCX на сервере) не сделан. Не решено: переключатель «задание утверждено до 01.09.2022» в карточке проекта (ПП № 87; переходное правило п. 2 ПП № 963 не прочитано, без него переключатель был бы догадкой). Не прочитаны: поправка к ГОСТ Р 21.101-2026 (по сведениям исследователей, от 31.08.2026; текста нет) и Изменения № 1 и № 2 к ГОСТ Р 2.105-2019.

Расчётное ядро (МКЭ)

- Дефолт контакта — contact_strategy="newton" (с 05.10.2026); initial_stress (Пикар), penalty и active_set удалены в [0377] — поле contact_strategy принимает только newton, остальное 422 (критерии переключения и результаты — §6.13). Оценка времени до запуска не откалибрована для Ньютона.
- Бенч 1978.1 (мульти-распорка, 5 этапов) улучшился, но не закрыт. Общая устойчивость реализована отдельным модулем вне МКЭ-ядра (slope_stability, /api/checks/stability): Феллениус и упрощённый Бишоп; Янбу, Моргенштерн–Прайс, поверхности произвольной формы и сейсмика — нет; основа вердикта решена 02.10 (по умолчанию упрощённый Бишоп); открыты ψ и γd в [k_st] (приняты 1) и порог «у предела».
- Открыто после потока ядра [0293]: скрининг до счёта по estimate_solve_cost (отказ/предупреждение), подмена чисел при механизме (решение владельца), заколы с автоматическим подбором, запас по λ как SRF; данные от проектировщика для 1998.1 и пионерного среза; результаты панелей проверок не входят в вердикт, DOCX и XML; Ньютон переключён по умолчанию ([0297], §6.13): базы трёх наборов сверки (73 расчёта) пересчитаны; открыты калибровка оценки времени для Ньютона, сравнение с базой на основе safety_model="auto", подключение базы настоящих отчётов к CI, сторож near-limit для Ньютона; одно ухудшение вне диапазона крайних задач (1721.6 при заделке 0,45: near_limit → mechanism).

© HydroWeb, 2026 · hydroweb.online