

Описание технической архитектуры программного обеспечения «TurbineCare AI»

Версия ПО: **1.0.0** · Правообладатель: ООО «ИНТЕБРИКС» (ИНН 9707059284) · Редакция документа: 18.09.2026

Документ для реестра российского ПО (ПП РФ № 1236): совокупность решений об организации программной системы — структурные элементы и их интерфейсы, соединение элементов, архитектурный стиль, технологическая основа, хранение данных, потоки данных, безопасность, виды экземпляров, развёртывание. В конце приведена сводная таблица технических и эксплуатационных характеристик в терминах справочников формы реестра.

1. Общие положения

1.1. Настоящий документ подготовлен ООО «ИНТЕБРИКС» (ИНН 9707059284, ОГРН 1267700207895), правообладателем программного обеспечения «TurbineCare AI» (далее — ПО, комплекс), и содержит описание технической архитектуры ПО по состоянию на выпуск версии 1.0.0 от 12 сентября 2026 года.

1.2. ПО предназначено для мониторинга технического состояния газотурбинных установок (ГТУ) и комплексных воздухоочистительных устройств (КВОУ): приёма телеметрии из автоматизированной системы управления технологическим процессом (АСУ ТП), приведения параметров к стандартным условиям, оценки деградации, обнаружения отклонений, прогнозирования сроков обслуживания и экономической оценки потерь.

1.3. ПО является серверным программным комплексом, размещаемым в вычислительной инфраструктуре заказчика (способ распространения — размещение в инфраструктуре заказчика). Облачное размещение силами правообладателя не предусмотрено; по модели SaaS ПО не предоставляется.

2. Принципы построения

2.1. Комплекс построен по сервисной схеме: функциональные части выделены в отдельные службы, каждая из которых работает в собственном контейнере и взаимодействует с остальными по внутренней сети развёртывания. Состав служб задан файлами развёртывания `docker-compose.prod.yml` и `docker-compose.db.yml`, входящими в комплект поставки.

2.2. Единственной точкой входа в комплекс извне является обратный прокси. Все прочие службы недоступны из внешней сети: их сетевые порты не публикуются на узле и доступны только внутри сети развёртывания.

2.3. Экземпляр объекта не обращается к внешним сетевым ресурсам в ходе работы, при установке и при обновлении. Отсутствие исходящих соединений подтверждено приёмочными испытаниями в сети без маршрута наружу (протоколы испытаний № 1 и №

2 от 11.09.2026, шаг 15). Единственное заявленное исходящее обращение во всё ПО — оповещения через Telegram на экземпляре центра удалённого мониторинга, включаемые заказчиком по его решению; перечень допустимых внешних адресов закреплён в исходном тексте и проверяется автоматически перед каждым выпуском.

2.4. Один и тот же исходный текст и одни и те же образы контейнеров используются для двух видов экземпляров, различающихся параметром `INSTANCE_ROLE`: экземпляр объекта (станции) и экземпляр центра удалённого мониторинга. Различие видов экземпляров описано в разделе 8.

3. Состав служб

Служба	Образ	Назначение
nginx	turbinecare-nginx	Обратный прокси, завершение TLS, ограничение частоты запросов. Единственная служба с публикуемыми портами: 80 и 443
web	turbinecare-web	Веб-приложение и программный интерфейс: разделы мониторинга, приём телеметрии по HTTP, управление настройками, отчёты
collector	turbinecare-web (отдельный процесс)	Сбор данных из АСУ ТП по OPC UA, обслуживание архива: свёртки, расчёт нормы по эталонным периодам, оценка деградации, обнаружение аномалий, чистка журнала действий
python-calc	turbinecare-python-calc	Расчётная служба: термодинамические расчёты, пороговые зоны, обнаружение аномалий, прогноз ресурса
ops	turbinecare-ops	Служба обслуживания, выполняемая по требованию: применение миграций схемы базы данных при запуске, заведение учётных записей, применение описания станции, пересчёт показателей
db-backup	postgres (клиент)	Периодическое резервное копирование базы данных средствами <code>pg_dump</code>
postgres	postgres	Встроенная СУБД. Подключается вторым файлом развёртывания и применяется для стенда и малых установок; штатный режим на объекте — СУБД заказчика

3.1. Образы `turbinecare-web`, `turbinecare-ops` и `turbinecare-nginx` построены на базовом образе Alpine Linux, образ `turbinecare-python-calc` — на образе `python:3.11-slim`. Ни один образ не зависит от операционной системы узла: среда выполнения и библиотеки находятся внутри образа.

3.2. Служба `web` и служба `collector` собраны из одного образа и одного исходного текста. Процесс сбора запускается отдельно, чтобы приём и обслуживание данных не зависели от нагрузки на веб-приложение и не прерывались при его перезапуске.

4. Технологическая основа

Уровень	Средство
Веб-приложение и программный интерфейс	TypeScript, Next.js (React), среда выполнения Node.js
Процесс сбора данных	TypeScript, Node.js; клиент OPC UA
Расчётная служба	Python, FastAPI; библиотеки численных расчётов NumPy, pandas, scikit-learn; библиотека термодинамических свойств CoolProp
Доступ к базе данных	Prisma ORM с адаптером PostgreSQL
СУБД	PostgreSQL версии 14 и новее (в том числе сборки из Единого реестра российского ПО)
Обратный прокси	nginx
Контейнеризация	Docker Engine версии 24 и новее, Docker Compose версии 2

4.1. Объём исходного текста версии 1.0.0 (по метке v1.0.0): 443 файла на языке TypeScript (из них 429 в каталоге приложения src/), 23 файла на языке Python в расчётной службе. Все компоненты сторонних правообладателей распространяются на условиях свободных лицензий; их перечень приведён в файле THIRD-PARTY-LICENSES.md, входящем в образ приложения (экран «О программе»), и опубликован на сайте правообладателя.

5. Модель данных

5.1. Данные комплекса хранятся в реляционной базе данных PostgreSQL. Основные сущности:

Сущность	Содержание
Organization, Asset, Turbine	Организация, объект (площадка) и машина. Каждая машина принадлежит одной организации; доступ к данным машины разрешён только учётным записям её организации
TurbineConfig, SignalDefinition, TagMapping	Описание машины: нормативные параметры, перечень принимаемых сигналов, соответствие тегов АСУ ТП внутренним именам сигналов
TelemetryPoint	Принятые точки телеметрии с отметкой времени и признаком качества сигнала
ComputedMetric	Расчётные показатели, вычисленные по каждой принятой точке
KpiSnapshot, DegradationSnapshot	Свёртки показателей по часам и суткам; оценки деградации
Setpoint, ReferencePeriod	Уставки сигнализации с двусторонними коридорами (нижние и верхние предупредительный и аварийный пороги, зона нечувствительности, гистерезис) и эталонный период, по которому

Сущность	Содержание
	фиксируется статистическая норма
Alarm	Журнал сигнализации: возникновение, квитирование, закрытие
MaintenanceEvent, ResourceCounter	События технического обслуживания и ресурсные счётчики
ForecastResult, ForecastQuality	Результаты прогноза и оценка их качества по факту
IngestionSource, CollectorState	Источники данных и состояние процесса сбора
User, Session, AuditEntry	Учётные записи, сессии и журнал действий

5.2. Данные хранятся по срокам п. 5.3: детальные строки после истечения срока удаляются администратором по команде (п. 6.4), свёртки хранятся не менее пяти лет; удалению всегда предшествует свёртка. Схема данных рассчитана на переход к секционированию таблиц телеметрии и расчётных показателей по времени в последующих выпусках; готовность СУБД заказчика к этому (PostgreSQL версии 14 и новее, создание, отсоединение и удаление секции) проверяется приёмочными испытаниями (шаги 13 и 14).

5.3. Сроки хранения данных задаются параметрами развёртывания. Значения по умолчанию: точки телеметрии — 90 суток, расчётные показатели — 30 суток, журнал сигнализации — 365 суток, свёртки — 1825 суток (5 лет). Сроки согласованы во всех документах и проверяются автоматически перед выпуском.

6. Поток данных

6.1. Приём. Телеметрия поступает одним из пяти способов: опросом сервера OPC UA процессом сбора; передачей точек по HTTP на программный интерфейс комплекса (с ключом доступа); загрузкой файла истории; опросом внешней базы данных; из имитационного источника для проверки установки (воспроизведение ранее загруженных точек). Каждая точка проходит проверку соответствия перечню принимаемых сигналов; при передаче по HTTP теги, не входящие в перечень, возвращаются в ответе как неприятые.

6.2. Расчёт. По каждой принятой точке вычисляются расчётные показатели: приведение мощности, расхода топлива и температур к стандартным условиям, перепады давления на ступенях КВОУ с приведением к плотности воздуха, теплотехнические характеристики, интегральный индекс состояния. Термодинамические расчёты выполняет расчётная служба; при её недоступности применяется встроенный расчёт на TypeScript той же формулы.

6.3. Сигнализация. Расчётные показатели сверяются с уставками. Уставки двусторонние: срабатывание происходит при выходе как за верхние, так и за нижние пороги. Переход в худшее состояние фиксируется немедленно, возврат — с запасом (гистерезис), что исключает дребезг. Событие записывается в журнал сигнализации.

6.4. Обслуживание архива. Процесс сбора по расписанию формирует часовые и суточные свёртки показателей, рассчитывает норму по закрывшимся эталонным периодам, вычисляет оценки деградации, выполняет обнаружение аномалий средствами расчётной службы и укорачивает журнал действий. Обработанный объём отмечается водяным знаком, поэтому повторный запуск обрабатывает только новые данные. Удаление детальных строк, вышедших за сроки хранения (п. 5.3), в версии 1.0.0 в расписание не включено и выполняется администратором по команде (обработчик `POST /api/cron/retention` под ключом `CRON_SECRET`); порядок обязателен: сначала свёртка, затем удаление.

6.5. Представление. Веб-приложение строит разделы мониторинга по свёрткам и расчётным показателям за выбранный период. Период выбирается пользователем и закреплён в адресе страницы; чтение из базы ограничено окном периода и перечнем показателей, что исключает неограниченные запросы. Отчёты формируются в формате PDF.

7. Безопасность

7.1. Аутентификация: учётная запись с паролем, а также вход через каталог предприятия по протоколу LDAP и через провайдера единого входа по протоколу OpenID Connect. Пароли хранятся в виде хешей bcrypt. Учётная запись, заведённая при установке, получает временный пароль и обязана сменить его при первом входе; до смены разделы недоступны.

7.2. Авторизация: ролевая модель (администратор, инженер, наблюдатель) с изоляцией данных по организациям. Проверка прав выполняется на стороне сервера для каждого запроса; программный интерфейс отвечает гостю отказом с кодом 401, а не перенаправлением.

7.3. Защита от подбора: ограничение частоты попыток входа по адресу источника; нарастающая задержка после серии неудачных попыток по учётной записи (только для неверного пароля — верный пароль принимается всегда, чтобы подбор от чужого имени не запирал оператора); выравнивание времени ответа для существующих и несуществующих учётных записей.

7.4. Журнал действий: изменяющие операции записываются в журнал с цепочкой хешей SHA-256, где каждая запись включает хеш предыдущей; подмена или удаление записи в обход приложения обнаруживаются при проверке цепочки. Записи старше срока хранения (по умолчанию 730 суток) удаляются обслуживанием штатно, с отдельной записью о выполненной чистке.

7.5. Транспорт: соединения с комплексом защищены TLS на обратном прокси; при отсутствии сертификата заказчика при первом запуске создаётся самоподписанный сертификат. Учётные данные подключений к источникам хранятся в базе данных в зашифрованном виде.

7.6. Секреты развёртывания (ключ сессий, общий секрет расчётной службы, ключ шифрования учётных данных) создаются при установке случайными и хранятся в файле

настроек с ограниченными правами доступа. В образах контейнеров и артефактах сборки секреты отсутствуют, что проверяется автоматически перед выпуском.

7.7. Известные уязвимости компонентов проверяются перед выпуском; выпуск при наличии критических уязвимостей в поставляемых компонентах не производится.

8. Виды экземпляров

8.1. Экземпляр объекта (INSTANCE_ROLE=object) устанавливается на станции. Он принимает телеметрию, выполняет все расчёты и хранит данные. Лицензионных ограничений на экземпляре объекта нет: комплекс работает бессрочно, разделы управления лицензиями и подпиской на нём отсутствуют.

8.2. Экземпляр центра удалённого мониторинга (INSTANCE_ROLE=operator) предназначен для обслуживающей организации, ведущей несколько объектов: по замыслу он не принимает телеметрию и не выполняет расчётов, а получает от экземпляров объектов только сводные показатели по решению заказчика. В версии 1.0.0 реализованы роль экземпляра и разбор её значения, перечень возможностей и заслоны (разделы, не относящиеся к роли, отвечают кодом 404), роли пользователей OPERATOR_ANALYST и OPERATOR_ADMIN, модели данных центра (профили и пакеты выгрузки, площадки, снимки, запросы, акты и реестр передач) и подписка с файлом лицензии; сборщик и отправитель пакетов, приёмник на стороне центра и экраны центра в состав версии 1.0.0 не входят и поставляются в последующих выпусках. Оповещения через Telegram подключаются только к экземпляру центра; на экземпляре объекта оповещения выдаются в веб-интерфейсе и по электронной почте. Мобильное приложение правообладателя — отдельный продукт, в состав ПО не входит.

8.3. Разделение реализовано в одном исходном тексте и проверяется приёмочными испытаниями: разделы, не относящиеся к виду экземпляра, отвечают кодом 404.

9. Развёртывание и эксплуатация

9.1. Установка выполняется мастером первичной настройки из архива дистрибутива без обращения к внешним ресурсам: мастер записывает настройки, применяет миграции схемы базы данных, запускает состав и заводит первую учётную запись администратора. Порядок описан в документе «Инструкция по установке».

9.2. Требования к узлу: операционная система с поддержкой Docker Engine версии 24 и новее и Docker Compose версии 2; 4 ядра процессора, 8 ГБ оперативной памяти, 50 ГБ дискового пространства (минимальные требования; рекомендуемые — 8 ядер, 16 ГБ, 200 ГБ). Совместимость с операционными системами из Единого реестра российского ПО подтверждена протоколами испытаний на Astra Linux Special Edition 1.8.2 и РЕД ОС 8.0.2.

9.3. Резервное копирование выполняется службой db-backup по расписанию и сценарием scripts/backup.sh по требованию; восстановление из дампа проверено приёмочными испытаниями (шаг 12).

9.4. Обновление выполняется архивом новой версии сценарием scripts/update.sh: образы загружаются из архива, миграции применяются службой ops, данные сохраняются.

9.5. Состояние комплекса доступно по адресу /api/health: без входа — только общее состояние (для пробы живости контейнера и мониторинга), вошедшему пользователю — состояние базы данных, расчётной службы и процесса сбора, ход обслуживания архива и версия.

10. Соответствие требованиям к российскому программному обеспечению

10.1. Все компоненты ПО разработаны на территории Российской Федерации; компоненты сторонних правообладателей распространяются на условиях свободных лицензий без территориальных ограничений.

10.2. ПО не использует облачные сервисы иностранных провайдеров; экземпляр объекта не обращается к внешним ресурсам в ходе работы, при установке и при обновлении (единственное заявленное исходящее обращение — оповещения Telegram на экземпляре центра, п. 2.3).

10.3. Сборка ПО выполняется без обращения к сети Интернет из архива пакетов, входящего в комплект поставки, на технических средствах, расположенных на территории Российской Федерации.

10.4. Хранение исходного текста и объектного кода описано в документе «Описание технических средств хранения исходного текста и объектного кода».

11. Сводная таблица технических и эксплуатационных характеристик

Таблица приведена в терминах справочников формы заявления; выбранные позиции соответствуют документам «Инструкция по установке» и «Протоколы приёмочных испытаний».

Характеристика	Значение
Способ распространения	Размещение в инфраструктуре заказчика; SaaS не предоставляется
Базовая операционная система	Astra Linux Special Edition (1.8.2 — протокол № 1), РЕД ОС (8.0.2 — протокол № 2); допускаются иные дистрибутивы Linux x86_64 с Docker Engine 24 и новее и Docker Compose версии 2
Система виртуализации	Не требуется: ПО работает в контейнерах на физическом или виртуальном сервере под любым гипервизором
Балансировщик нагрузки	Не используется; входной точкой служит обратный прокси nginx
Контейнеризация	Docker (Docker Engine 24 и новее, Docker Compose версии 2)

Характеристика	Значение
Средства хранения данных	PostgreSQL (СУБД заказчика версии 14 и новее, испытано на 15 и 16; встроенная PostgreSQL 16 — для стенда и малой установки)
Облачное хранилище	Не используется
Веб-сервер / прокси-сервер	nginx 1.27
Программная платформа	Node.js 20 / Next.js 14 (React 18) — веб-приложение и процесс сбора; Python 3.11 / FastAPI — расчётная служба
Сервер приложений	Встроенный сервер Next.js (Node.js) в образе <code>turbinecare-web</code> ; Uvicorn в образе <code>turbinecare-python-calc</code> ; отдельный сервер приложений не требуется
СУБД	PostgreSQL
Компоненты	nginx (обратный прокси), web (веб-приложение и программный интерфейс), collector (сбор данных по OPC UA и обслуживание архива), python-calc (расчётная служба), ops (обслуживание по требованию), db-backup (резервное копирование), postgres (встроенная СУБД, необязательно)
Реализация (языки программирования)	JavaScript (TypeScript 5, Node.js 20); Python 3.11
Фреймворки и библиотеки	Next.js, React, Prisma, NextAuth, node-opcua, Tailwind CSS, Recharts; FastAPI, NumPy, pandas, scikit-learn, CoolProp
Средства сборки	Docker Engine и Docker Compose, Node.js 20 / npm / esbuild, pip по lock-файлу; сборка без обращения к сети из архива пакетов дистрибутива
Язык интерфейса	Русский