

ТЕХНОЛОГИЧЕСКОЕ ВИДЕНИЕ И АРХИТЕКТУРА ПЛАТФОРМЫ МБОРД

Версия документа: 1.0

Дата: 2026-05-05

Организация-разработчик: ООО "Метрикор"

Продукт: МБорд — платформа промышленного мониторинга оборудования

Введение

Настоящий документ описывает технологическое видение, архитектурные решения и принципы построения платформы МБорд — веб-системы промышленного мониторинга оборудования, разработанной ООО "Метрикор".

Платформа МБорд предназначена для непрерывного сбора, хранения и визуализации показателей промышленного оборудования с контроллеров Segnetics и иных источников телеметрии. Система обеспечивает оперативное оповещение персонала о выходе параметров за допустимые границы, а также предоставляет инструменты для анализа исторических данных.

Документ адресован техническим специалистам, архитекторам, членам экспертного совета по включению ПО в Единый реестр российских программ для электронных вычислительных машин и баз данных.

Концепция проекта

МБорд — это специализированная платформа мониторинга промышленного оборудования, построенная как модульный монолит с чётким разделением ответственности между компонентами. Платформа решает задачу централизованного контроля параметров технологических процессов в реальном времени.

Ключевые бизнес-задачи:

- Непрерывный мониторинг показателей промышленного оборудования (температура, давление, расход, уровень и пр.)
- Автоматическое оповещение ответственного персонала при выходе параметров за установленные пороговые значения
- Визуализация текущего состояния и исторических данных в удобном веб-интерфейсе
- Ролевое разграничение доступа для различных категорий персонала

Целевая аудитория:

- Инженеры КИПиА (контрольно-измерительные приборы и автоматика)
 - Операторы технологических процессов
 - Руководители технических служб предприятий
 - Энергетики промышленных объектов
-

1. Ключевые принципы

Архитектура платформы МБорд основана на следующих фундаментальных принципах, обеспечивающих надёжность, масштабируемость и сопровождаемость системы.

Таблица 1. Ключевые архитектурные принципы

№	Принцип	Описание	Обоснование
1	Модульная монолитная архитектура	Система построена как единое приложение с чётким разделением на модули (домены): датчики, алерты, контроллеры, пользователи	Упрощает развёртывание и отладку при сохранении чистоты кодовой базы. Каждый модуль имеет собственный router, schemas, services, models
2	API-First	Все функции системы доступны через документированный REST API (OpenAPI 3.1). Фронтенд взаимодействует с бэкендом исключительно через API	Обеспечивает возможность интеграции со сторонними системами, автоматизацию тестирования, независимую разработку фронтенда и бэкенда
3	Event-Driven	Асинхронная обработка событий через Redis Pub/Sub для оповещений и обновления состояний	Снижает связанность компонентов, обеспечивает отзывчивость системы при пиковых нагрузках на запись телеметрии
4	Cloud-Native	Контейнеризация всех компонентов (Docker), декларативная конфигурация (Docker Compose), CI/CD через GitLab	Воспроизводимость окружений, горизонтальное масштабирование, автоматизация развёртывания
5	Clean Architecture	Разделение бизнес-логики, транспортного уровня и уровня хранения. Зависимости направлены внутрь (от инфраструктуры к домену)	Тестируемость, независимость от конкретных технологий, простота замены компонентов
6	Security by Default	JWT-аутентификация в httpOnly-cookie, RBAC с 4 ролями, хеширование паролей Argon2, CORS-политики, rate limiting	Защита промышленных данных, соответствие требованиям безопасности критической инфраструктуры
7	Time-Series First	Использование TimescaleDB как специализированного хранилища временных рядов с гипертаблицами и непрерывными агрегатами	Оптимальная производительность при записи и чтении телеметрических данных с высокой частотой

2. Технологический стек

2.1 Backend

Компонент	Технология	Версия	Назначение
Язык программирования	Python	3.11+	Основной язык серверной части
Web-фреймворк	FastAPI	0.109+	Асинхронный REST API с автогенерацией OpenAPI-документации
ORM	SQLAlchemy	2.0	Объектно-реляционное отображение, поддержка async
Миграции БД	Alembic	latest	Управление схемой базы данных, версионирование миграций
Драйвер СУБД	asyncpg	latest	Асинхронный драйвер PostgreSQL
Пакетный менеджер	UV	latest	Быстрый пакетный менеджер и виртуальные окружения
Линтер	Ruff	latest	Статический анализ и форматирование кода
Тестирование	pytest	latest	Модульное и интеграционное тестирование
E2E тестирование	Playwright	latest	Сквозное тестирование веб-интерфейса

2.2 Frontend

Компонент	Технология	Версия	Назначение
Библиотека UI	React	19	Компонентный подход к построению интерфейса
Язык	TypeScript	5.9	Типобезопасность, автодополнение, рефакторинг
Сборщик	Vite	7	Быстрая сборка и HMR для разработки
CSS-фреймворк	TailwindCSS	v4	Утилитарные CSS-классы, дизайн-система
Компоненты UI	shadcn/ui	latest	Готовые доступные UI-компоненты
Управление состоянием	Zustand	latest	Легковесный state manager
Серверное состояние	TanStack React Query	latest	Кэширование и синхронизация серверных данных

2.3 Хранилище данных

Компонент	Технология	Версия	Назначение
СУБД	PostgreSQL	15	Реляционное хранилище
Расширение	TimescaleDB	latest	Оптимизация хранения и запросов временных рядов
Кэш/брокер	Redis	7	Кэширование, сессии, Pub/Sub оповещения

2.4 Инфраструктура

Компонент	Технология	Версия	Назначение
Контейнеризация	Docker	latest	Изоляция сервисов, воспроизводимость
Оркестрация	Docker Compose	latest	Декларативное описание многоконтейнерной среды
Реверс-прокси	Nginx	latest	TLS-терминация, балансировка, статика
CI/CD	GitLab CI/CD	latest	Автоматизация сборки, тестирования, деплоя
Реестр образов	GitLab Container Registry	latest	Хранение Docker-образов

2.5 Интеграции и безопасность

Компонент	Технология	Назначение
Аутентификация	JWT (httpOnly cookie)	Безопасная аутентификация без хранения токенов в localStorage
Авторизация	RBAC (4 роли)	Гранулярное управление доступом
Хеширование паролей	Argon2	Устойчивость к brute-force и rainbow-table атакам
Сбор данных	FileWatcher	Мониторинг CSV-файлов от контроллеров Segnetics

3. Целевые показатели эффективности

Таблица 2. KPI платформы

Показатель	Целевое значение	Метод измерения
Доступность системы (uptime)	≥ 99,5%	Мониторинг health-endpoint, расчёт за календарный месяц
Время отклика API (p95)	≤ 200 мс	Метрики FastAPI, логирование медленных запросов (>500 мс)
Время отклика API (p99)	≤ 500 мс	Метрики FastAPI
Задержка доставки алертов	≤ 30 с	От момента превышения порога до отображения в UI
Пропускная способность записи	≥ 1000 показаний/с	Нагрузочное тестирование ingestion-пайплайна
Время рендеринга дашборда	≤ 2 с	Метрики фронтенда (Largest Contentful Paint)
Глубина хранения данных	≥ 1 год (raw), ≥ 5 лет (агрегаты)	Политики retention TimescaleDB
Время восстановления (RTO)	≤ 1 час	Регламент восстановления из бэкапа
Допустимая потеря данных (RPO)	≤ 15 минут	Частота бэкапирования WAL
Время загрузки исторического графика (7 дней)	≤ 3 с	Метрики фронтенда для графиков с >10 000 точек

4. Анализ рисков

Таблица 3. Реестр рисков

№	Риск	Вероятность	Влияние	Меры митигации
1	Потеря данных телеметрии при сбое СУБД	Низкая	Критическое	WAL-архивирование, pg_basebackup, репликация. Буферизация данных на стороне FileWatcher при недоступности БД
2	Превышение допустимой задержки обработки данных при пиковой нагрузке	Средняя	Высокое	Асинхронная обработка (asyncio), батчевая вставка, непрерывные агрегаты TimescaleDB, горизонтальное масштабирование
3	Несанкционированный доступ к данным мониторинга	Низкая	Критическое	JWT в httpOnly cookie, Argon2, RBAC, rate limiting, CORS, аудит-лог действий пользователей
4	Отказ сети между контроллером и сервером	Средняя	Среднее	Локальная буферизация CSV на контроллере, автоматическая досылка при восстановлении связи, алерт на отсутствие данных
5	Исчерпание дискового пространства	Низкая	Высокое	Политики retention (компрессия старых чанков, удаление raw-данных старше retention period), мониторинг свободного места
6	Зависимость от внешних библиотек и их уязвимости	Средняя	Среднее	Фиксация версий в lock-файлах, регулярный аудит зависимостей, использование только библиотек с активной поддержкой
7	Сбой контейнерной инфраструктуры	Низкая	Высокое	Политики restart: always, health checks, автоматический мониторинг состояния контейнеров, CI/CD для быстрого переразвёртывания
8	Рост объёма данных сверх планируемого	Средняя	Среднее	Компрессия TimescaleDB (до 90% экономии), непрерывные агрегаты для длительных периодов, планирование ёмкости

5. Список сокращений

Таблица 4. Сокращения и термины

Сокращение	Расшифровка
API	Application Programming Interface — программный интерфейс приложения
CI/CD	Continuous Integration / Continuous Delivery — непрерывная интеграция и доставка
CORS	Cross-Origin Resource Sharing — совместное использование ресурсов между источниками
CSV	Comma-Separated Values — формат данных с разделителем
HMR	Hot Module Replacement — горячая замена модулей при разработке
JWT	JSON Web Token — токен аутентификации
КИПиА	Контрольно-измерительные приборы и автоматика
ORM	Object-Relational Mapping — объектно-реляционное отображение
RBAC	Role-Based Access Control — управление доступом на основе ролей
REST	Representational State Transfer — архитектурный стиль взаимодействия
RTO	Recovery Time Objective — целевое время восстановления
RPO	Recovery Point Objective — допустимая точка восстановления
СУБД	Система управления базами данных
TLS	Transport Layer Security — протокол защиты транспортного уровня
UI	User Interface — пользовательский интерфейс
WAL	Write-Ahead Log — журнал упреждающей записи