

Архитектура и технологический стек

программного комплекса FusionAI

Документ для заявки в реестр российского программного обеспечения.

Правообладатель: ООО «ФБЮЖНАИ» (ИНН 6670527516, ОГРН 1246600061388).

Дата составления: 12.08.2026.

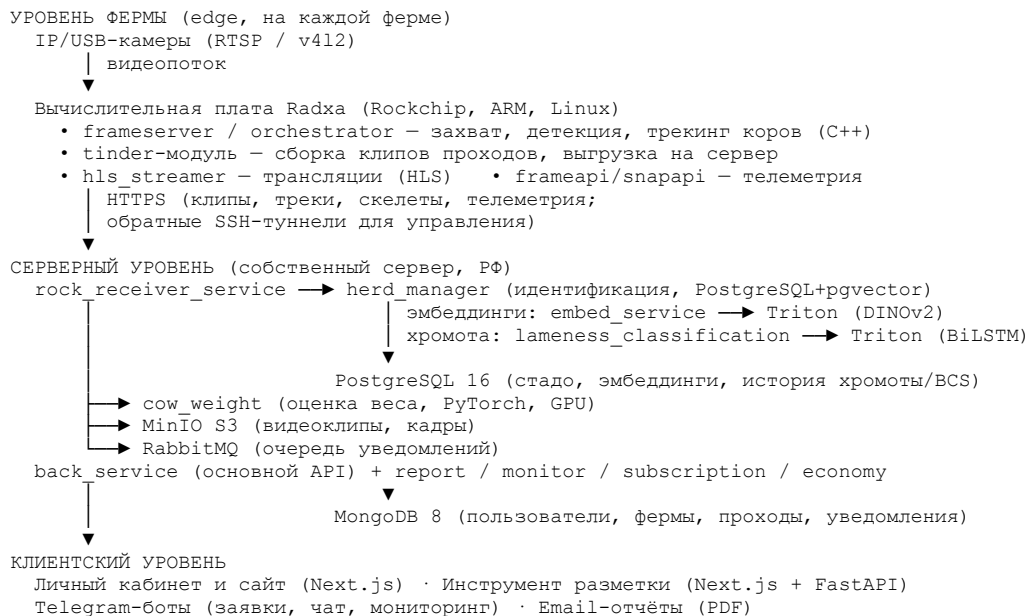
1. Общие сведения

Программный комплекс FusionAI (далее — ПК) предназначен для автоматизированного мониторинга здоровья и учёта крупного рогатого скота на молочных фермах по данным видеонаблюдения: бесконтактная идентификация животных, автоматическая оценка степени хромоты и упитанности, оценка веса, ведение электронного стада, уведомления и отчётность для зоотехников и ветеринарных специалистов.

ПК построен по трёхуровневой архитектуре:

- уровень фермы (edge) — вычислительные платы с подключёнными камерами: захват видео, детекция и трекинг животных непосредственно на ферме;
- серверный уровень — микросервисная платформа обработки, хранения и аналитики данных на собственном сервере на территории РФ;
- клиентский уровень — веб-приложения: личный кабинет фермы, публичный сайт, инструмент разметки данных, Telegram-боты.

2. Схема архитектуры и потоки данных



Основной поток данных: камера → плата (детекция/трекинг) → передача пакета (кадры, треки, скелетные точки) на сервер → идентификация животного по визуальным эмбедингам → оценка хромоты, веса и упитанности → запись в базы данных → отображение в личном кабинете, уведомления и отчёты.

3. Состав компонентов

3.1. Серверные прикладные сервисы (Python 3.11, FastAPI; единый Docker-образ, порты 9000–9500).

Компонент	Назначение
back_service	Основной API: аутентификация (JWT в HTTP-only cookies), личный кабинет (стадо, фермы, сотрудники, модули), блог и формы сайта, прокси к ML-сервисам, стримы, уведомления, чат, аудит
report_service	Формирование PDF-отчётов по стаду и рассылка аналитики на email по расписанию
monitor_service	Мониторинг парка плат: телеметрия, статусы, удалённый перезапуск (SSH)
rock_receiver_service	Приём данных с плат: проходы, оценки хромоты, видеоклипы; выгрузка в S3; интерфейс подтверждения кейсов
subscription_service	Биллинг: клиенты, подписки, счета, платежи; интеграция с эквайрингом ПАО «Сбербанк»
economy_service	Экономические расчёты: котировки цены молока, калькуляторы эффективности

3.2. Сервисы идентификации и машинного обучения.

Компонент	Назначение	Технологии
herd_manager	Идентификация животных по эмбедингам (векторный поиск), карточки, HMM-сглаживание истории хромоты, балл упитанности	Python 3.13, SQLAlchemy 2, PostgreSQL 16 + pgvector
embed_service	Извлечение визуальных эмбедингов (DINOv2, размерность 384)	Python 3.11, Triton-клиент
lameness_classification_service	Классификация степени хромоты по трекам 13 ключевых точек скелета (BiLSTM, 5 выходных голов)	Python 3.11, Triton-клиент
cow_weight	Оценка веса по изображению: сегментация U-Net + регрессор на Vision Transformer	Python 3.11, PyTorch 2.4.1, CUDA 12.1
NVIDIA Triton Inference Server	Единый инференс-сервер нейросетевых моделей (DINOv2, BiLSTM)	GPU NVIDIA

3.3. Клиентские приложения.

Компонент	Назначение	Технологии
fusionai-frontend	Публичный сайт и личный кабинет фермы (стадо, здоровье, трансляции, отчёты)	Next.js 16, React 19, TypeScript, Tailwind CSS 4
cow_labeler (web)	Инструмент разметки видеоданных для обучения моделей (хромота, походка)	Next.js 16 + FastAPI (Python 3.11)
Telegram-боты	Приём заявок с сайта, веб-чат,	Python (Telegram Bot API)

Компонент	Назначение	Технологии
	мониторинг оборудования ферм	

3.4. Программное обеспечение уровня фермы (edge).

Компонент	Назначение
frameserver / orchestrator	Захват видео (RTSP/v4l2), детекция и трекинг животных на плате (C++)
tinder-модуль	Формирование клипов проходов животных и их выгрузка на сервер
hls_streamer	Трансляции с камер (HLS-сегменты в RAM-диске)
frameapi / snaparapi	HTTP-интерфейсы телеметрии, снимков и управления платой

Аппаратная платформа: одноплатные компьютеры Radxa (SoC Rockchip, ARM64, ОС Linux); связь с сервером — HTTPS и обратные SSH-туннели.

3.5. Инфраструктурные компоненты.

Компонент	Роль	Версия
MongoDB	Основная документная СУБД платформы	8.0
PostgreSQL + pgvector	Реляционная СУБД идентификации и истории здоровья, векторный поиск	16
RabbitMQ	Брокер сообщений (очередь уведомлений)	3.13
MinIO	S3-совместимое объектное хранилище медиаданных	актуальная
Milvus	Векторная СУБД вспомогательного сервиса поиска	2.x
Nginx	Обратный прокси, TLS-терминация	актуальная
Docker / Docker Compose	Контейнеризация и оркестрация всех сервисов	v2

4. Технологический стек (сводно)

Категория	Технологии (лицензии)
Языки программирования	Python 3.11/3.13 (PSF), TypeScript (Apache-2.0), C++ (edge-компоненты), SQL, Bash
Backend-фреймворки	FastAPI (MIT), Uvicorn (BSD-3), Pydantic (MIT), SQLAlchemy (MIT), Alembic (MIT), Motor (Apache-2.0), aio-pika (Apache-2.0)
Frontend	Next.js (MIT), React (MIT), TanStack Query (MIT), Zustand (MIT), Tailwind CSS (MIT), Radix UI (MIT), next-intl (MIT), hls.js (Apache-2.0)
Машинное обучение	PyTorch (BSD-3), NVIDIA Triton Inference Server (BSD-3), timm (Apache-2.0), segmentation-models-pytorch (MIT), OpenCV (Apache-2.0), NumPy/Pandas (BSD-3). Модели: DINOv2-эмбединги, BiLSTM-классификатор хромоты, U-Net + ViT оценки веса, YOLO-детекция на платах
Хранение данных	MongoDB 8 (SSPL), PostgreSQL 16 (PostgreSQL License), pgvector (PostgreSQL License), MinIO (AGPL-3.0), Milvus (Apache-2.0)
Инфраструктура	Docker/Compose (Apache-2.0), Nginx (BSD-2), RabbitMQ (MPL-2.0), ОС Linux (GPL-2.0)

Категория	Технологии (лицензии)
Интеграции	Эквайринг ПАО «Сбербанк», SMTP, Telegram Bot API, S3-хранилище российского провайдера (АО «Таймвэб»)

Всё стороннее программное обеспечение в составе комплекса — свободное ПО с открытыми лицензиями (MIT, Apache-2.0, BSD, PostgreSQL, MPL, LGPL, GPL/AGPL; компоненты с ограничительными лицензиями используются как самостоятельные процессы и сервисы без линковки с собственным кодом). Проприетарные компоненты иностранных правообладателей, требующие лицензионных отчислений, в составе ПК отсутствуют. Обновление ПК выполняется правообладателем с собственной инфраструктуры; механизмы принудительного обновления или управления из-за рубежа отсутствуют.

5. Хранение и обработка данных

- все серверные компоненты и базы данных размещены на собственном сервере правообладателя на территории Российской Федерации (см. письмо о технических средствах хранения данных);
- MongoDB — учётные записи, фермы, стадо, проходы, уведомления, биллинг;
- PostgreSQL — карточки животных, визуальные эмбединги (pgvector), история хромоты и упитанности;
- MinIO (S3) — видеоклипы и кадры; часть медиа публичного контура — в S3-хранилище российского провайдера АО «Таймвэб»;
- на платах ферм хранятся только временные буферы (кадры, HLS-сегменты в оперативной памяти).

6. Взаимодействие компонентов

Внутреннее взаимодействие — HTTP REST в изолированной docker-сети; асинхронные события — RabbitMQ (AMQP). Внешние интерфейсы: HTTPS (личный кабинет, API, приём данных с ферм), RTSP/v4l2 (камеры → платы), HLS (трансляции), SMTP (отчёты), Telegram Bot API (боты), SSH (обслуживание плат через обратные туннели). Все внешние подключения терминируются обратным прокси (Nginx) с TLS.

7. Разработка, сборка и развёртывание

Исходный код ведётся в системе контроля версий Git. Сборка и развёртывание — Docker-образы, оркестрация Docker Compose; конвейер непрерывного развёртывания — self-hosted раннеры на инфраструктуре правообладателя (развёртывание по фиксации изменений в основной ветке, отдельный тестовый контур). Автоматизированное тестирование — pytest (интеграционные контуры с изолированными БД в контейнерах), Jest.

8. Требования к среде функционирования

Уровень	Требования
Сервер	x86-64, ОС Linux, Docker; GPU NVIDIA (CUDA 12) для инференса моделей; SSD/NVMe для СУБД, HDD для видеoarхива
Платы ферм	Radxa/Rockchip ARM64, ОС Linux, Docker; локальная сеть с камерами

Уровень	Требования
	(RTSP/USB)
Клиенты	Современный веб-браузер; установка клиентского ПО не требуется