Правообладатель: Махмудов Рустам Гаджимагомедович
Свидетельство № 2023667275 от 14.08.2023

МЕДИЦИНСКАЯ ИНФОРМАЦИОННАЯ СИСТЕМА ДЛЯ ЧАСТНЫХ МЕДИЦИНСКИХ ЦЕНТРОВ «ELCRM»

Описание технической архитектуры

ELCRM.AP

Из каких подсистем состоит ELCRM, как они связаны, где хранятся данные, как устроены безопасность, развёртывание и масштабирование.

Версия программы
2026.11

Редакция документа
2026.11.16 от 16.11.2026

Реализация лицензий
ИП Махмудова Ф. М., ИНН 054604349442

Поддержка
+7 989 457-40-40, info@elcrm.ru

Содержание

1 Архитектурный стиль
2 Структурные элементы
3 Хранение данных
4 Интерфейсы и соединение элементов
5 Развёртывание
6 Безопасность
7 Производительность и масштабирование

Для кого: ИТ-служба клиники, архитекторы, эксперты Реестра российского ПО

1 Архитектурный стиль

Медицинская информационная система для частных медицинских центров «ELCRM» версии 2026.11 — клиент-серверная веб-система с выделенным экземпляром для каждой клиники (single-tenant). Ядро — модульный монолит основной подсистемы (CRM), вокруг него — клиентские приложения и специализированные службы, которые работают с данными только через программные интерфейсы ядра.

- **Бизнес-логика — на сервере.** Интерфейсы в браузере и мобильном приложении отображают данные и отправляют команды; правила расчётов, статусов визита и кассы исполняются в ядре.
- **Одна схема данных.** Таблицы общей базы создаются только миграциями ядра; остальные подсистемы читают и пишут их через свои API.
- **Границы клиентов.** Рабочее место врача обращается только к своему API; сайт и мобильное приложение — к публичному API ядра; лабораторный интерфейс — к лабораторной службе.
- **Вертикали — модулями.** Стоматология, косметология и другие направления включаются флагами модулей, без отдельной копии кода.
- **Решения фиксируются.** Каждое архитектурное решение описано в журнале решений (ADR) с контекстом и последствиями.

2 Структурные элементы

Таблица 1. Подсистемы ELCRM

Подсистема	Назначение	Технологии
Ядро (CRM)	Регистратура, расписание, визиты, касса, склад, аптека, контакт-центр, задачи, отчёты, администрирование; публичный API для сайта и приложения	PHP 8.4, Laravel 13, Horizon, Reverb (PHP-FPM или Octane на FrankenPHP); интерфейс — React 17, TypeScript, Ant Design
API рабочего места врача	Приём пациента, медицинская документация, задачи и доход врача	PHP 8.4, Laravel 13, Horizon, Reverb, Sanctum; режим Octane (FrankenPHP) для высокой нагрузки
Рабочее место врача (веб)	Интерфейс врача в браузере	React 18, TypeScript, Vite, Ant Design
Лабораторная информационная система	Заказы, штрихкоды, преаналитика, результаты, обмен с лабораториями	Служба на Go (Gin, GORM); интерфейс — React 19, Vite
Сервер лицензий	Выпуск и проверка лицензионных ключей экземпляра	PHP, Laravel; интерфейс администратора — React

Подсистема	Назначение	Технологии
Сайт клиники	Публичный сайт, онлайн-запись, кабинет пациента	Next.js 15, React 19, TypeScript
Мобильное приложение пациента	Запись, визиты, анализы, чат, оплата, бонусы	Capacitor 8, React 19, TypeScript; сборки Android и iOS
Carbon	Наблюдение за сетью клиники, угрозы, инциденты, КИИ	Go: один исполняемый файл в режимах «сенсор» и «ядро»; раздел интерфейса — в ядре

ИИ-узел клиники (HUB, шлюз моделей, голосовые агенты) поставляется отдельно на оборудовании клиники с графическими ускорителями и в объём основной поставки не входит. Его описание — на странице модуля «Искусственный интеллект и ИИ-узел».

3 Хранение данных

- **MariaDB 10.11** — основная реляционная база экземпляра: пациенты, расписание, визиты, финансовые операции, склад, лаборатория, лицензии, данные Carbon (`carbon_*`). Миграции схемы — только дополняющие.
- **Redis 7** (лицензия BSD) — очереди фоновых задач, кэш, сессии, масштабирование WebSocket.
- **Файловое хранилище сервера** — сканы, протоколы, результаты анализов, печатные формы. Внешние облачные хранилища не используются.
- Резервные копии базы и файлов создаются на сервере клиники по регламенту резервного копирования.

4 Интерфейсы и соединение элементов

Таблица 2. Внутренние интерфейсы

Откуда	Куда	Протокол
Браузер сотрудника	Ядро (CRM)	HTTPS, сессия; WebSocket (Reverb) для событий в реальном времени
Рабочее место врача (веб)	API рабочего места врача	HTTPS, REST/JSON, JWT

Откуда	Куда	Протокол
Сайт клиники, мобильное приложение	Публичный API ядра `/api/v1`	HTTPS, REST/JSON
Лабораторный интерфейс	Лабораторная служба (Go)	HTTPS, REST/JSON
Ядро, API врача	Сервер лицензий	HTTPS, проверка ключа экземпляра
Ядро	Фоновые обработчики (Horizon)	Очереди Redis
Сенсор Carbon	Ядро Carbon	HTTP(S) внутри сети клиники; события — в общей базе

Таблица 3. Внешние интеграции (включаются настройками клиники)

Назначение	Поставщик	Способ
Касса и ККТ (54-ФЗ)	Драйвер KKMServer на рабочем месте кассира	Локальный HTTP
Онлайн-оплата	ЮKassa	HTTPS API
SMS	SMS.ru	HTTPS API
Мессенджеры	Шлюз i2crm (MAX, Telegram, VK и другие)	HTTPS API, вебхуки
Телефония	МТС BATC, Asterisk, «Мои звонки»	HTTPS API, AMI/ARI
ЕГИСЗ и РЭМД	Medflex	HTTPS API
Лаборатории	Helix, Alfalab	HTTPS API
Аптека	ИнфоАптека	FTP/FTPS-обмен
Медицинские изображения	PACS Orthanc клиники	REST API Orthanc

Все интеграции подключаются адаптерами ядра и выключены, пока клиника не задала ключи доступа. Для работы программы в базовой конфигурации внешние сервисы не требуются.

5 Развёртывание

Схема экземпляра клиники

```
Пользователи (браузер, приложение)
        | HTTPS
        ▼
nginx ── crm.<домен>    → PHP-FPM (или Octane): ядро (CRM)
      ├─ armapi.<домен> → PHP-FPM (или Octane): API рабочего места врача
      ├─ arm.<домен>    → статика рабочего места врача
      ├─ lab.<домен>    → статика ЛИС;  labapi.<домен> → служба на Go
      ├─ lic.<домен>    → сервер лицензий
      └─ site.<домен>   → сайт клиники (Next.js)
                |
      MariaDB 10.11 · Redis 7 · файлы · Horizon · Reverb
      Carbon: ядро (systemd) + сенсор на копии трафика
```

- Базовая операционная система — Debian 12 (bookworm); поддерживаются дистрибутивы на базе Debian и Ubuntu.
- Каждая клиника получает свой сервер (физический, виртуальный или аренда в российском ЦОД) и свои доменные имена.
- TLS-сертификаты выпускаются для доменов клиники; трафик между пользователем и сервером шифруется.
- Подробная процедура — в документе «Руководство по установке».

6 Безопасность

- Аутентификация сотрудников по логину и паролю с двухфакторной аутентификацией; пациентов — по одноразовому коду.
- Ролевая модель доступа: роли (регистратура, кассир, врач, лаборант, директор, маркетолог и другие) и дополнительные разрешения по разделам.
- Журналирование действий пользователей и входов; хранение журналов на сервере клиники.
- Секреты (пароли, ключи интеграций) — только в конфигурации сервера; в интерфейс и клиентские сборки не попадают.
- Заголовки безопасности (CSP и другие), ограничение частоты запросов к входу и публичным формам.
- Регулярные аудиты безопасности кода и зависимостей.

7 Производительность и масштабирование

- Ядро и API врача работают под PHP-FPM; для высокой нагрузки — в режиме постоянно загруженного приложения (Laravel Octane на FrankenPHP). Число рабочих процессов настраивается под нагрузку.
- Тяжёлые операции (рассылки, обмен с лабораториями, построение отчётов) выполняются фоновыми обработчиками очередей; их число масштабируется независимо.
- Отчёты руководителя строятся по подготовленным моделям чтения, чтобы не нагружать рабочие таблицы в часы приёма.
- Для сети клиник с филиалами используется один экземпляр с разграничением данных по филиалам.