

УТВЕРЖДАЮ



Генеральный директор ООО  
«Алнисофт»

А.С. Азаров

М.П.

« 28 »

декабря

2024 года

Эхотрекер

Описание технической архитектуры программного обеспечения

На 10 листах

## **Аннотация**

Настоящий документ содержит описание технической архитектуры программного обеспечения Эхотрекер (далее – ПО, Эхотрекер, Система), предоставленного для проведения экспертной проверки.

Документ содержит сведения о назначении ПО, схему технической архитектуры ПО, описание компонентов (сервисов и модулей) системы и технологий и средств разработки ПО.

# Содержание

<b>Перечень сокращений.....</b>	<b>4</b>
<b>1 Общие сведения.....</b>	<b>5</b>
1.1 Наименование программного обеспечения.....	5
1.2 Краткое описание возможностей средства автоматизации .....	5
1.1 Характеристика объекта автоматизации .....	5
1.1.1 Задача.....	5
1.1.2 Потенциальные пользователи Решения .....	6
1.1.3 Варианты использования .....	6
<b>2 Описание технической архитектуры системы .....</b>	<b>7</b>
2.1 Схема технической архитектуры .....	7
2.2 Описание компонентов (сервисов и модулей) системы .....	8
2.3 Технологии и средства разработки .....	9
2.3.1 Внешнее окружение и технологии в части бэк-энда.....	9
2.3.2 Внешнее окружение и технологии в части фронт-энда.....	10
2.3.3 Внешнее окружение и технологии в части искусственного интеллекта .....	10

## Перечень сокращений

Термин/ сокращение	Определение
DICOM	(англ. Digital Imaging and Communications in Medicine) – стандарт обработки, хранения, передачи, печати и визуализации медицинских данных.
HTML	(англ. HyperText Markup Language) – стандартизированный язык разметки документов для создания структуры веб-страниц
HTTP	(англ. HyperText Transfer Protocol) – сетевой протокол передачи данных, предназначенный для отправки запросов веб-серверу и получения ответов в виде HTML-страниц или файлов
HTTPS(s)	(англ. HyperText Transfer Protocol Secure) – безопасный протокол передачи данных, который поддерживает шифрование посредством криптографических протоколов, и является расширенной версией протокола HTTP
PACS	(анг. picture archiving and communication system) – система (сервер) хранения медицинских изображений, предназначенный для получения, передачи и архивации DICOM-данных
ИИ	Искусственный интеллект
ПО	Программное обеспечение
РАСУДМ	Российская ассоциация специалистов ультразвуковой диагностики в медицине
ССЗ	Сердечно-сосудистые заболевания
СЭМД	Структурированный электронный медицинский документ, является стандартом Министерство здравоохранения Российской Федерации для обмена информацией в рамках государственных систем в здравоохранении и обеспечения электронного медицинского документооборота.
УЗ	Ультразвук, ультразвуковой

## **1 Общие сведения**

### **1.1 Наименование программного обеспечения**

Программное обеспечение Эхотрекер, Система.

### **1.2 Краткое описание возможностей средства автоматизации**

Программное обеспечение Эхотрекер представляет собой рабочую станцию, предназначенную для хранения, просмотра, полуавтоматической аннотации и количественной обработки эхокардиографических изображений с цифровых УЗ-аппаратов и составления протокола эхокардиографического исследования. Аннотация изображений выполняется с использованием технологий искусственного интеллекта (компьютерное зрение).

Использование Эхотрекер позволяет обеспечить лучшее качество проведения и описания исследования за счет большей воспроизводимости измерений, сокращения времени на проведение измерений вручную и заполнение протокола, возможности использования экспертных функций (оценка глобальной и сегментарной сократимости) вне зависимости от класса УЗ-аппарата. Как следствие, врач и пациент получают более достоверное заключение, что способствует выбору корректной тактики ведения заболевания.

### **1.1 Характеристика объекта автоматизации**

#### **1.1.1 Задача**

Одним из ведущих методов диагностики сердечно-сосудистых заболеваний является эхокардиография – самый широко используемый и доступный метод кардиовизуализации, необходимый и достаточный для диагностики и ведения различных сердечно-сосудистых заболеваний. При статистике ССЗ в Российской Федерации, составляющей путь меньше 25тыс. случаев на 100тыс. человек, в год проводится более 12млн. эхокардиографических исследований. При этом проведение одного исследования по рекомендациям, опубликованным РАСУДМ, занимает 45-60 минут.

Актуальность автоматизации в сфере эхокардиографии диктуется необходимостью повышения доступности исследования, стандартизации процессов и качества изображений, снижения вариабельности результатов, оптимизации времени врачей (больше времени пациенту, меньше на рутинные операции) и концентрации его усилий на решении более сложных клинических задач.

### **1.1.2 Потенциальные пользователи Решения**

Целевой аудиторией проекта являются врачи функциональной диагностики, кардиологи, реаниматологи, выполняющие диагностическую эхокардиографию на цифровых УЗ-аппаратах различного уровня

### **1.1.3 Варианты использования**

Система Эхотрекер предполагает два основных варианта использования при работе с эхокардиографическим исследованием:

1) Проведение измерений структур миокарда, оценки гемодинамики и описание исследования для создания протокола с показателями из перечня СЭМД «Протокол трансторакальной эхокардиографии стандартный»;

2) Расчет показателей деформации левого желудочка – определение глобальной и сегментарной сократимости для оценки состояния миокарда и выявления патологий на ранней стадии.

## 2 Описание технической архитектуры системы

Система построена с использованием сервис-ориентированной структуры, в основе которой лежит применение микросервисов – слабо-связанных модулей (компонент) ПО для инкапсулирования ожидаемой логики работы. Микросервисная архитектура подразумевает использование широко известных стандартов взаимодействия, таких как HTTP(s), что позволяет включать в систему в качестве вспомогательных модулей готовые компоненты и решения.

### 2.1 Схема технической архитектуры

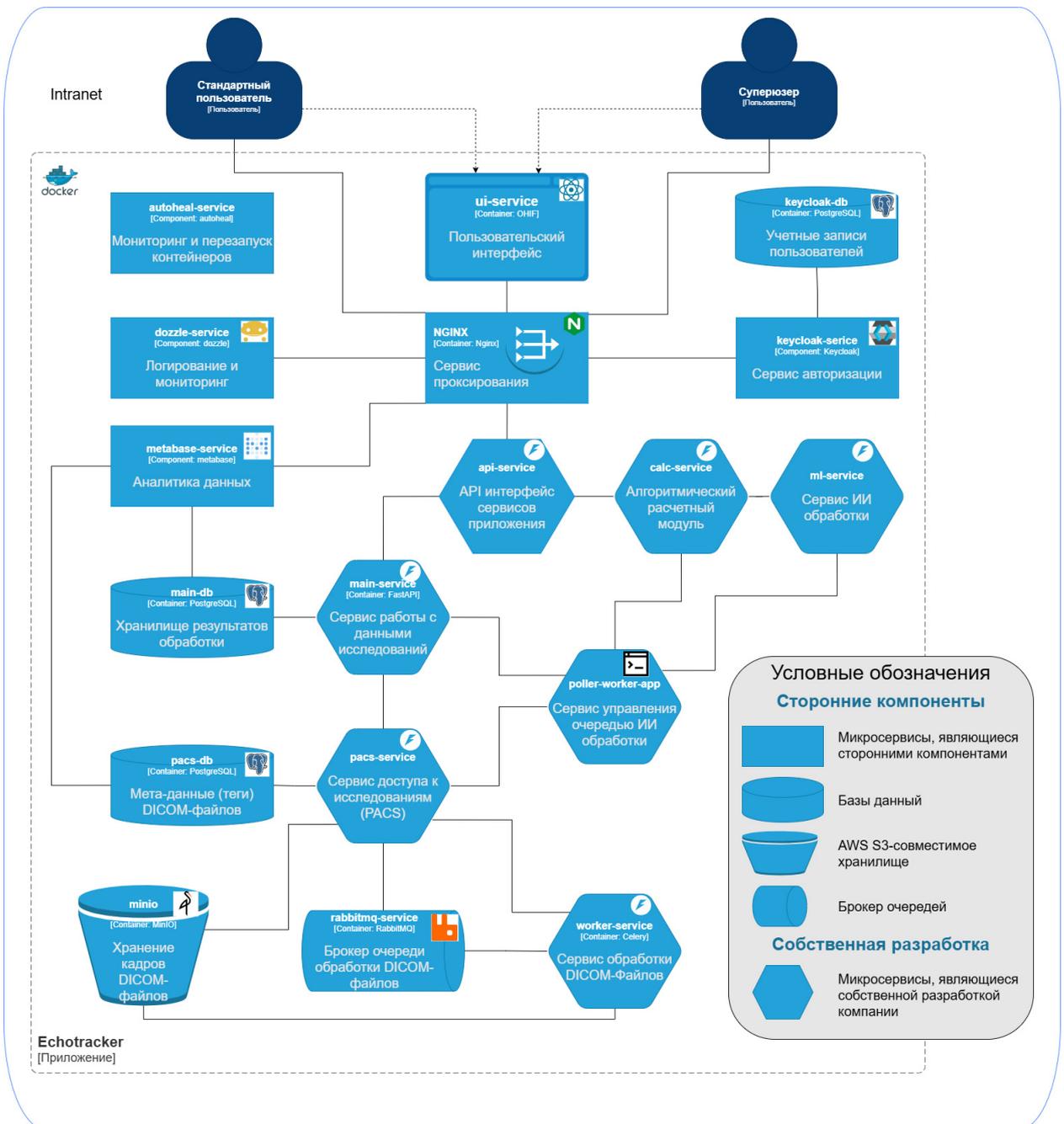


Рисунок 1. Схема технической архитектуры решения

Схема технической архитектуры представлена на рис. 1.

## 2.2 Описание компонентов (сервисов и модулей) системы

Система состоит из следующих ключевых компонентов:

1) **Пользовательский web-интерфейс (ui-service)** – клиентская часть приложения, предоставляющего пользователю графический интерфейс с помощью web-браузера (Яндекс.Браузер, Chrome). Является основным способом взаимодействия пользователей с сервисом Эхотрекер.

2) **API интерфейс сервисов приложения (api-service)** отвечает за перенаправление пользовательских запросов в другие модули системы в соответствии с логикой работы ПО, а также выполняет операции по простым преобразованиям формата данных.

3) **Алгоритмический расчетный модуль (calc-service)** содержит все необходимые функции для проведения аналитических расчетов по совершенным измерениям и отвечает за вызов функций из сервиса ИИ обработки.

4) **Сервис ИИ обработки (ml-service)** выполняет автоматическую предварительную обработку кадров исследования с помощью нейросетевых алгоритмов: генерация контуров эндокарда левого желудочка на апикальных проекциях, генерация линейных измерений на продольной парастернальной проекции и др.

5) **Сервис работы с данными исследований (main-service)** предназначен для работы с данными приложения (сбор сведений об исследованиях, генерация отчетов и др.).

6) **Модуль доступа к исследованиям (pacs-service)** предназначен для доступа к исходным dicom файлам, данные которых содержатся в двух хранилищах для быстрого доступа: AWS S3-совместимое хранилище MinIO для изображений (кадров) и база данных PostgreSQL для метаданных dicom-файлов. Совместное использование в Эхотрекере этих хранилищ и модуля доступа выполняет роль PACS сервиса.

7) **Сервис управления очередью ИИ-обработки (poller-worker-app)** автоматически запускает обработку исследований ИИ-сервисом в зависимости от их статуса в Хранилище результатов обработки.

8) **Сервис обработки DICOM-файлов (worker-service)** выполняет затратные по времени операции обработки DICOM-файлов.

9) **Брокер очереди обработки DICOM-файлов RabbitMQ (rabbitmq-service).**

10) **Сервис проксирования Nginx** – позволяет обращаться к разным компонентам системы по одному адресу и ограничивает прохождение запросов к системе по размеру загружаемых данных, частоте запросов и другим параметрам.

11) **Сервис авторизации Keycloak** (keycloak-service) выполняет аутентификацию и разграничение уровней доступа пользователей. Информация об учетных записях пользователей содержится в сервисной базе данных PostgreSQL.

12) **Сервис логирование и мониторинг Dozzle** (dozzle-service) позволяет собирать и просматривать отчеты о работе приложения (логи).

13) **Сервис аналитики Metabase** (metabase-service) предоставляет пользовательский интерфейс для сбора и отображения аналитики баз данных системы.

14) **Сервис мониторинга и перезапуска контейнеров** (autoheal-service) – независимый сервис, который отслеживает состояние контейнера и при необходимости автоматически перезапускает их.

15) **Хранилище результатов обработки** (main-db) представляет собой PostgreSQL базу данных и содержит информацию об исследованиях, специфичную для сервиса Эхотрекер, такую как данные о дате загрузки, статус обработки исследований, пользовательские отчеты и проведенные пользователями измерения.

16) **База данных учетных записей пользователей** (keycloak-db) представляет собой PostgreSQL базу данных и содержит информацию содержит данные пользователей, зарегистрированных в системе.

17) **Хранилище мета-данных DICOM-файлов** (pacs-db) представляет собой PostgreSQL базу данных и содержит мета-данные (теги) DICOM-файлов.

18) **Хранилище кадров DICOM-файлов** (minio) представляет собой AWS S3-совместимое хранилище MinIO и содержит изображения (кадры) DICOM-файлов.

## **2.3 Технологии и средства разработки**

Компоненты системы разработаны с использованием языков программирования/ библиотек/ фреймворков, перечисленных ниже.

### **2.3.1 Внешнее окружение и технологии в части бэк-энда**

Python, FastAPI, ASGI, Uvicorn, Pydantic, pydicom, pynetdicom, PostgreSQL, SQLAlchemy, psycopg2, alembic, MinIO, Nginx, Keycloak, Poetry.

Реализация на языке программирования Python 3.11 с использованием веб-фреймворка FastAPI и ASGI веб-сервера Uvicorn. Для автогенерации

программного интерфейса, валидации и сериализации данных применяется библиотека Pydantic совместно с внутренними инструментами типизации. Взаимодействие с dicom файлами и устройствами выполняется посредством библиотек pydicom и runetdicom. Для обеспечения высокой доступности и горизонтального масштабирования применяется Celery. В качестве основного хранилища выполненных измерений и метаданных исследований используется PostgreSQL, изображения хранятся в совместимом с интерфейсом AWS S3 хранилище MinIO. В качестве вспомогательных сервисов применяются Nginx для проксирования HTTP запросов и Keycloak, отвечающий за менеджмент учетных записей и авторизации. Для управления зависимостями и упаковки используется инструмент Poetry.

### **2.3.2 Внешнее окружение и технологии в части фронт-энда**

JavaScript, React, Redux, cornerstone, axios, yarn.

Реализация на языке программирования JavaScript с использованием фреймворка React. Для отрисовки и разметки измерений используются библиотека react-cornerstone-viewport и функционал библиотеки cornerstone-tools расширенный с помощью собственного плагина. Хранение сессионных данных и управление состоянием приложения производится с помощью библиотеки Redux. Для взаимодействия клиентской составляющей приложения с бэкендом посредством HTTP запросов используется библиотека axios. Для установки и отслеживания версий зависимости используется пакетный менеджер yarn.

### **2.3.3 Внешнее окружение и технологии в части искусственного интеллекта**

Python, jsonlines, cvat-sdk, OpenCV, ResNet34, U-net, OpenPose, pydicom, numpy, pandas, scipy, albumentations, pytorch, catalyst, hydra, clearml, poetry, nbdime, pre-commit, flake8, matplotlib, seaborn, jupyter-notebook, VSCode.

Реализация на языке программирования Python 3.9 с использованием фреймворка pytorch для обучения моделей на архитектурах ResNet34, U-net и OpenPose. Получение данных dicom-файлов с помощью библиотеки pydicom. Для организации и мониторинга экспериментов применялись фреймворк hydra платформа ClearML. Для реализации алгоритма speckle tracking использован метод вычисления оптического потока Лукаса-Канаде из библиотеки OpenCV. Для взаимодействия сервиса интеллектуальной обработки с другими сервисами приложения реализуются интерфейсы взаимодействия.

Модуль ML-обработки собран в отдельный пакет (waml-echosg-med) и при развертывании системы устанавливается в окружение python.