

# Описание технической архитектуры программного обеспечения

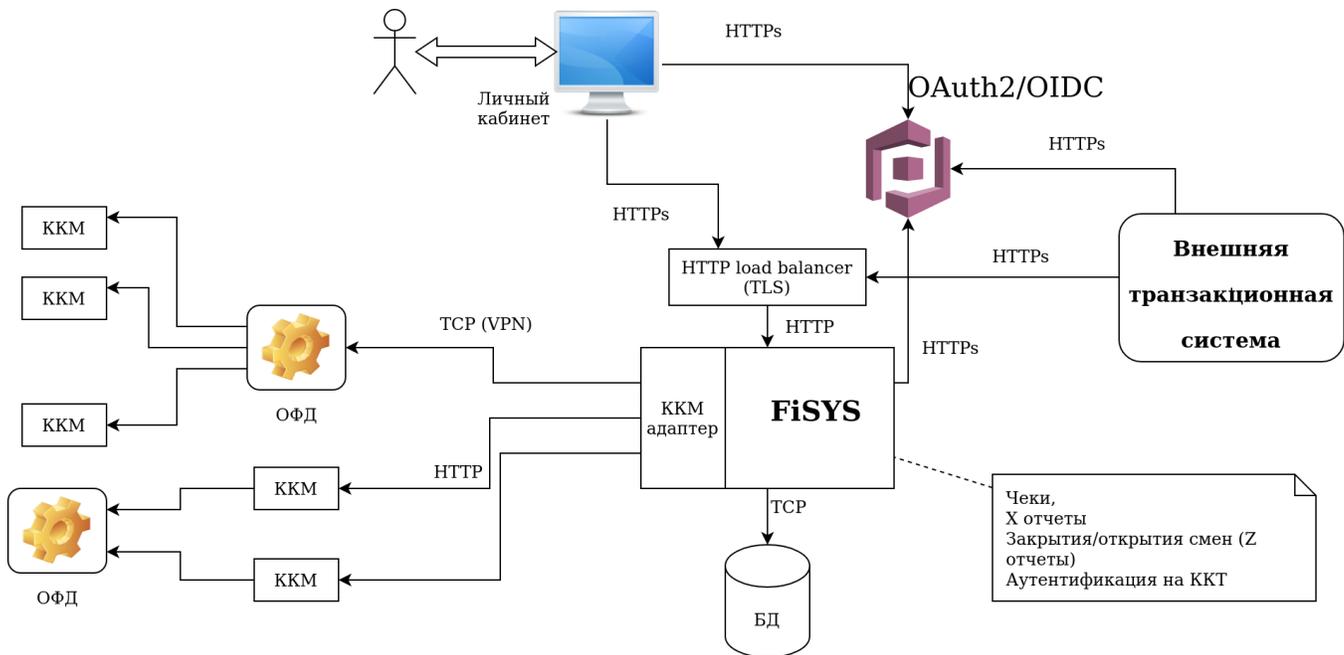
Система построена в виде сервис-ориентированной архитектуры программного обеспечения, используется взаимодействие слабо связанных и легко изменяемых модулей — микросервисов. Данная архитектура подразумевает использование широко известных стандартов взаимодействия между компонентами, что обеспечит возможность инкапсулировать требуемую логику работы внутри проектируемого модуля, а в качестве вспомогательных модулей/сервисов использовать готовые решения.

Сервис FiSYS представляет собой Java-based приложение, написанное с использованием платформы [Micronaut framework](#) на языке Java (версия 17). В основе платформы лежит [Netty network application framework](#) обеспечивающий реактивную (не блокирующую) реализацию обмена данными по сети. В качестве реализации реактивного подхода в JVM выбрана [Project Reactor](#) библиотека, имеющая [интеграционные модули для Micronaut](#). Для хранения справочников, настроек, зарегистрированных клиентов, компаний, внешних систем, ККТ и прочих данных используется База Данных. В качестве БД может использоваться [Oracle MySQL Community Edition](#) (версии не ниже 8.0) или [MariaDB](#) (версии не ниже 10.8): в сервис уже включены соответствующие JDBC драйвера. Создание схемы БД, инициализации необходимых записей и последующие обновление схемы реализовано за счет встроенных SQL скриптов и их исполнения через специализированную библиотеку версионности изменений в БД [Flyway Community edition](#).

Инстанс приложения представляет собой настраиваемый WEB сервис с REST API интерфейсом для Внешних транзакционных систем (система реализующая бизнес логику, требующей регистрации платежных транзакций в ККМ), а также WEB интерфейсом с Личным кабинетом для сотрудников поддержки, инспекторов и администратора. Для реализации защищенного канала соединения по HTTP должен использоваться TLS, который реализуется сторонними сервисами (и/или балансерами нагрузки) такими как [Netty/Caddy2/AWS ELB/YndexCloud ALB](#) и т.п.

REST API реализуется на базе HTTP(s) протокола и имеет описание функциональности через [OpenAPI](#) спецификацию с возможностью предоставить WEB UI интерфейс ([Swagger UI](#)) для демонстрации API.

Для утентификации клиентов Личного кабинета или Внешней системы используется внешний Сервис аутентификации и авторизации, реализующий стандарты [OAuth2](#) и [OpenID Connect](#). В качестве реализации может быть выбран продукт [Ory Hydra](#) с [Ory Kratos](#), [Okta](#), [Keycloak](#), [AWS Cognito](#).



Для взаимодействия с ККТ/ККМ (контрольно-кассовая техника/контрольно-кассовая машина) в системе FiSYS реализованы адаптеры с общим интерфейсом. На данный момент имеется адаптер KZ\_1 для работы с ККМ через "Казахтелеком" ОФД (оператор фискальных данных АО «Казахтелеком») и адаптер STARRUS\_1 для работы с СтарРус ККМ по протоколу взаимодействия версии 3.5.40. Есть возможность интеграции с другими ОФД или ККМ через реализацию новых адаптеров (Пейкюиск, АТОЛ). Для общения с ОФД от Казахтелеком используется [gRPC](#) протокол с передачей информации по выделенному каналу связи (VPN). Для работы с ККМ СтарРус используется HTTP протокол и связь также должна обеспечиваться по закрытым каналам соединения. В системе реализована поддержка off-line режима работы ККМ: когда касса становится не доступна, но при этом определенный набор операций доступен с занесением всей истории в специальную структуру хранения в БД, в строгой последовательности и с защитой от возможной несакционированной модификации данных. Когда связь с ККМ восстанавливается, происходит синхронизация данных собранных в off-line режиме. Для проверки работы FiSYS в тестовых среда, а также для интеграционных тестирований реализован тестовый режим работы ККМ, который эмулирует основные функции.

Система может быть масштабирована как вертикально так и горизонтально, за счет увеличения количества инстансов FiSYS приложения. Для повышения отказоустойчивости требуется использовать не менее чем по 2 экземпляра каждого жизненно важного элемента системы, работающего в кластере или за балансером. Нагрузка на БД предполагается не высокой, но также требуется кластеризация/дубликация с периодическим сохранением дампов. Выбор количества инстансов каждого сервиса в инфраструктуре определяется текущими требованиями по пропускной способности, выдерживаемой нагрузке и требованиями по отказоустойчивости. Текущая реализация позволяет использовать системы оркестровки такие как [Kubernetes \(K8s\)](#), [Kubernetes Rancher](#) и соответствующие облачные сервисы [Yandex Instance Groups](#), [Yandex Managed Service for Kubernetes](#), [Amazon Elastic Kubernetes Service](#), [Amazon EC2](#), [Amazon Elastic Container Service](#).

В FiSYS для логирования внутренних событий используется библиотека [Logback](#), позволяющая гибко настраивать адаптеры и уровни логирования.

Минимальные требования к инстансу сервиса FiSYS:

- 512 Мб ОЗУ (без учета потребления от ОС и других смежных сервисов)
- 1000 MHz одноядерное CPU
- В качестве ОС рекомендуется выбирать Unix/Linux-based системы: [Alphine](#), [Debian](#).
- JDK 17