

robokassa

**Документация, содержащая описание
функциональных характеристик экземпляра
программного обеспечения “Робокасса процессинг”**

**г. Москва
2026г.**

| | |
|--|---|
| Оглавление | |
| 1 Предназначение ПО | 3 |
| 2 Описание функциональных характеристик..... | 3 |
| 3 Программные и аппаратные требования..... | 3 |
| 4 Состав системы | 4 |
| 5 Эксплуатация системы..... | 6 |

1 Предназначение ПО

“Робокасса процессинг” это программное обеспечение для проведения бизнес транзакций, как финансовых, например интернет-платежей, так и иных; Например: выдача фискального чека или гарантированная доставка сообщения, онбординг клиента.

2 Описание функциональных характеристик

1. Проведение распределённых транзакций, гарантирующих выполнение или отмену операции по приему денег или иной бизнес-функции.
2. Интеграция с провайдерами услуг: сервисами интернет платежей, электронными кошельками, эквайерами банковских карт, операторами фискальных данных.
3. Авторизацию доступа клиента к различным функциям системы.
4. Маршрутизацию операций в соответствии с требованиями бизнеса. например распределение трафика карточных оплат между несколькими эквайерами.
5. Выгрузку результатов транзакций в файл для последующей обработки CMS/CRM/АБС системой заказчика – сверок, учёта, отображения клиентам и менеджерам.
6. Логирование и аудит – логи в файлах или топиках kafka могут использоваться для анализа как есть, либо с использованием систем мониторинга логов типа graphana, loki и тд.
7. Подключение Zabbix для мониторинга метрик состояния сервисов комплекса.
8. Администрирование процесса прохождения транзакций: поиск, изменение статуса, возможность исправления проблемных транзакций (например, остановленных при сетевых проблемах или ошибках).

3 Программные и аппаратные требования

Программные требования:

Операционная система: CasPay TMS совместим с операционными системами, где полноценно функционирует .Net 10 и ASP .NET соответствующей версии: Ubuntu Server, Debian, Alpine.

Веб-сервер: Kestrel, как часть ASP.NET. Не требует дополнительного администрирования.

Версия: NET обновляется раз в год на актуальную LTS.

База данных: PostgreSQL 17+.

Брокер сообщений: Kafka.

TLS-сертификаты: Для обеспечения аутентификации клиентов используются клиентские сертификаты.

Аппаратные требования:

Процессор: рекомендуется использовать сервер с мощным многоядерным процессором, таким как Intel Xeon или аналогичный, с тактовой частотой не менее 2.0 ГГц. Это обеспечит быструю обработку запросов и поддержку большого числа одновременных соединений.

Оперативная память: рекомендуется от 32GiB на всех серверах.

Хранилище данных: на сервере БД по потребному объему хранения, рекомендуем начинать от 1TiB. На остальных серверах от 250GiB.

Сетевой интерфейс: необходимо наличие сетевого интерфейса с достаточной пропускной способностью. Точные требования различаются от объема ожидаемого трафика.

Резервное питание: для обеспечения непрерывной работы системы рекомендуется использовать резервное питание или иные механизмы обеспечения бесперебойного питания (UPS).

DDoS защита: в силу особенностей бизнес схем вокруг приема платежей и т.д., рекомендуем озаботиться подключением системы защиты от DDoS атак.

4 Состав системы

Описание технической инфраструктуры

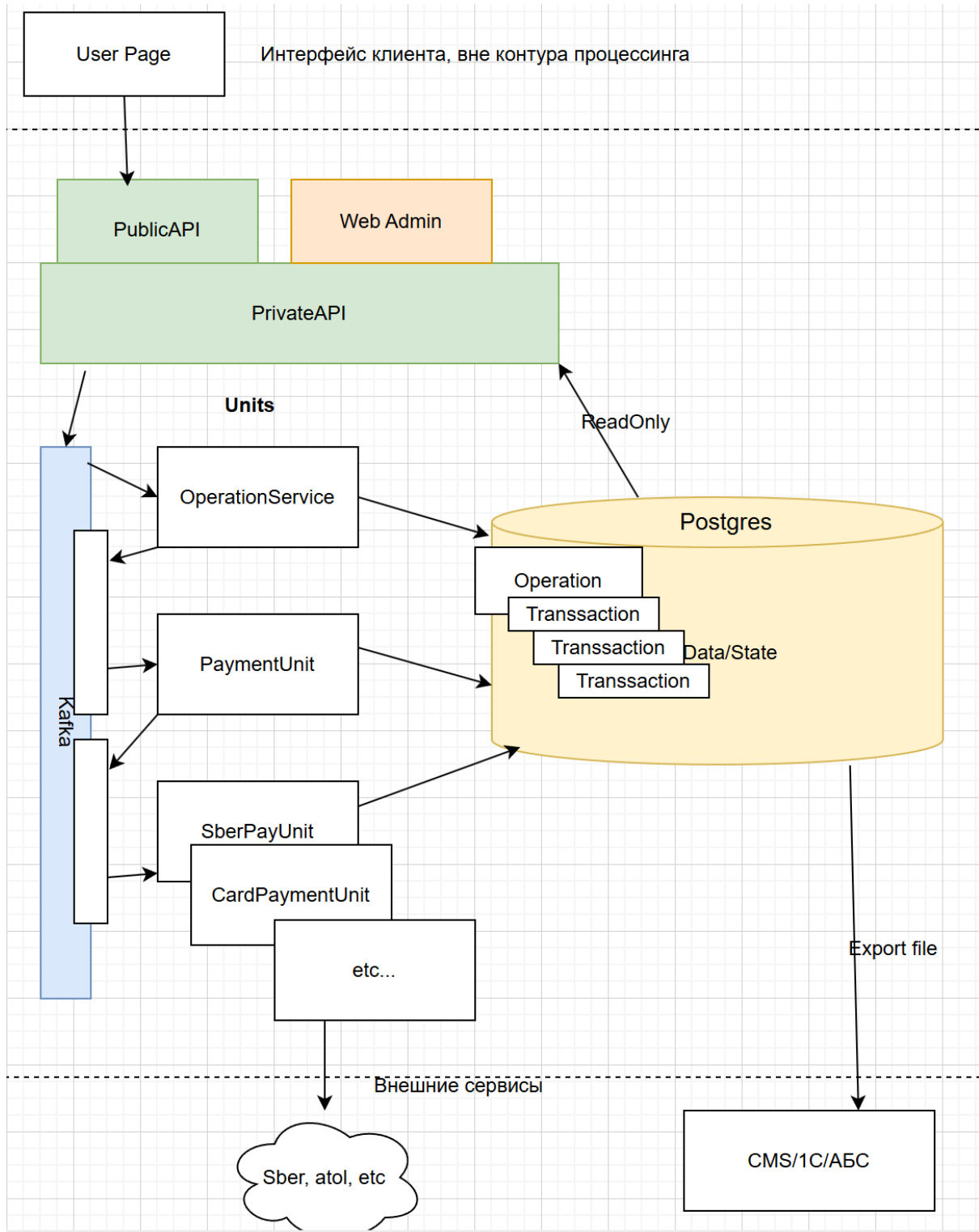
Язык программирования — C#, sql, js, ts

Используемая платформа разработки — .Net, ASP.NET

Используемые сторонние компоненты ПО – Фреймворк - Vue.js,

Библиотеки: Microsoft.Extensions.Hosting, NLog, Swashbuckle.AspNetCore, OpenTelemetry.Instrumentation.AspNetCore, OpenTelemetry.Exporter.OpenTelemetryProtocol, OpenTelemetry.Instrumentation.Runtime, Microsoft.Extensions.Http.Resilience, Microsoft.Extensions.ServiceDiscovery, OpenTelemetry.Extensions.Hosting, OpenTelemetry.Instrumentation.Http, Confluent.Kafka, Libkafka.redis, dotnet-etc, Aspire.Hosting.Orchestration.win-x64, Aspire.Dashboard.Sdk.win-x64, Aspire.Hosting.AppHost, jose-jwt, NUnit.Analyzers, Microsoft.NET.Test.Sdk, NUnit, coverlet.collector, NUnit3TestAdapter, Npgsql, Moq,

Общая структура системы



- User Page – сайт для взаимодействия с пользователем: ввода данных для оплаты и тд. В некоторых интеграциях, например для работы с фискальными чеками, он не требуется.

- PublicAPI – api для взаимодействия бекенда клиентов с системой. Позволяет создавать операции, получать статусы и передавать данные и команды в процессе исполнения.
- PrivateAPI – внутреннее api для эксплуатации системы. Предоставляем методы для низкоуровневого управления процессингом операций. Используется инструментами администрирования.
- Web Admin – приложения для администрирования системы.
- Units – приложения выполняющие либо интеграции с внешними сервисами либо реализующие внутренние функциональные единицы (сервис оплат, сервис фискального чека, сервис маршрутизации банковских карт и тд). Хостятся в docker контейнерах.
- Kafka – брокер сообщений. Все задачи к юнитам (Units) поступают через топика kafka.
- Postgres – БД, в ней хранятся операционные данные процессинга.
- Export file – периодически экспорт данных о завершенных операций в файл для обработки и помещения в OLAP хранилище.

5 Эксплуатация системы

Программный комплекс предназначен для непрерывной работы. Любая остановка в работе является внештатной ситуацией, однако при серьезных обновлениях возможны кратковременные отказы в обслуживании клиентов. Для гарантированной надежности работы заказчик должен обеспечить:

- Развертывание программного комплекса в соответствии с инструкцией.
- Настройку и администрирование БД PostgreSQL в соответствии с рекомендациями производителя.
- Своевременное (не реже раза в сутки) создание бекапа данных.
- Разворачивание и администрирование кластера kafka в соответствии с промышленными стандартами и целевой нагрузкой.
- Организации наблюдения за средствами мониторинга и своевременное реагирование на инциденты.
- Поддержание собственной сетевой инфраструктуры: настройка firewall и DDoS защиты, и прочие настройки в сетевой инфраструктуре заказчика.

Для операционного администрирования следует использовать сайт Web Admin, и наборы сервисных скриптов (описание см в документе про эксплуатацию системы).