

Архитектура программного обеспечения

Документ описывает архитектуру программного продукта «Почтовая платежная система NPP.LTT» (далее Система)

Система использует трехуровневую архитектуру. Уровень представления состоит из нескольких клиентских программ для различных операционных систем, в том числе мобильных. Уровень приложения представляет собой серверное web приложение. Уровень данных — реляционная система управления базами данных (СУБД).

Сервер

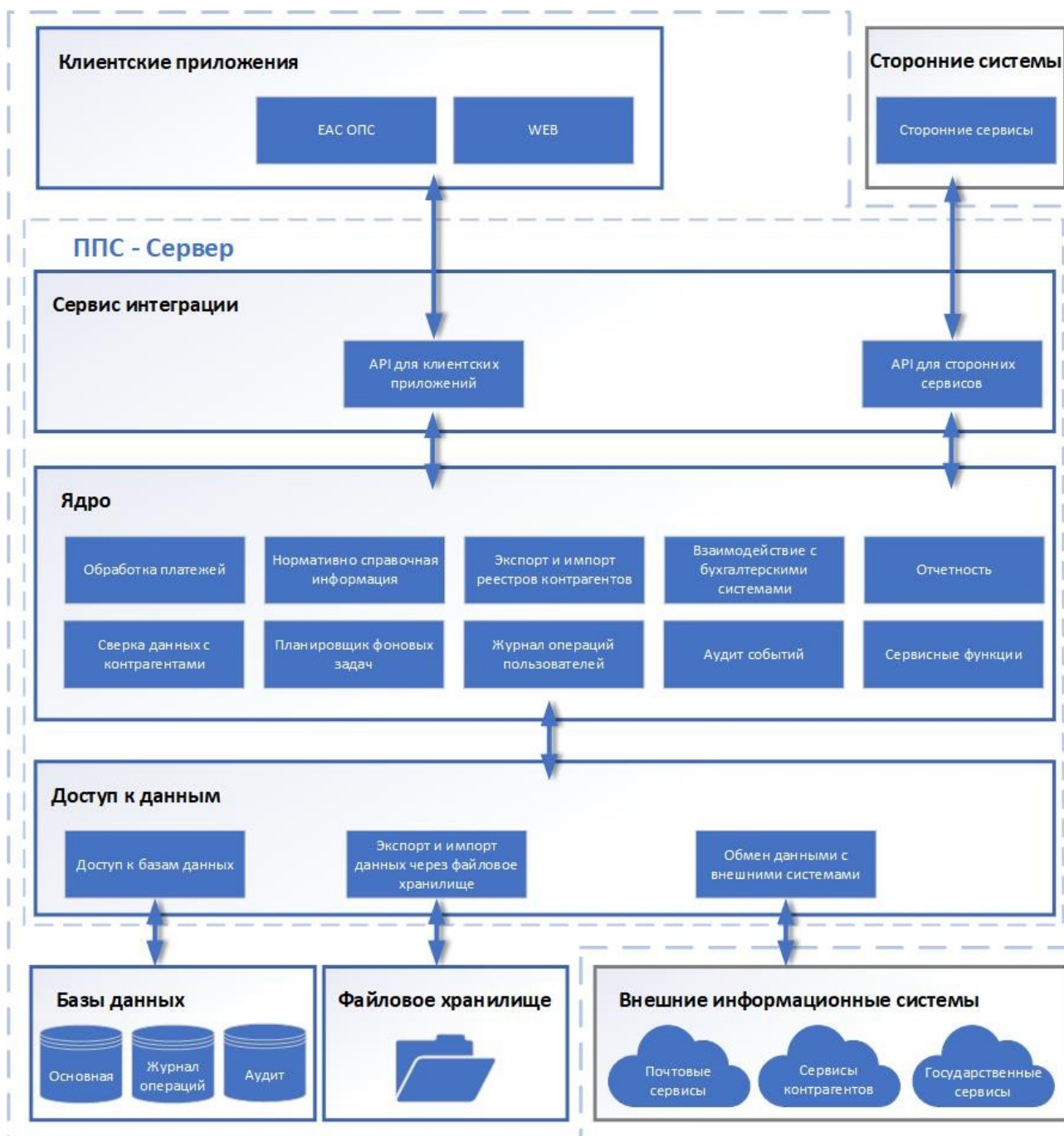
Серверное приложение представляет собой web приложение, реализованное с помощью языка C# на платформе .NET Framework 4.8.

Архитектуру серверного приложения можно условно разделить на слои: ядро, сервис интеграций и доступ к данным.

Слой ядра реализует бизнес-функции, обеспечивает взаимодействие и обмен данными со слоями сервиса интеграций и доступа к данным. Бизнес-функции разделены на самостоятельные, но связанные между собой функциональные блоки: обработка платежей, нормативно справочная информация, экспорт и импорт реестров контрагентов, взаимодействие с бухгалтерскими системами, отчетность, сверка данных с контрагентами, планирование фоновых задач, журнал операций пользователей, аудит событий, сервисные функции. Каждый функциональный блок реализован в виде отдельного подключаемого модуля или библиотеки классов. Состав модулей, их реализация, а также набор связей между ними могут быть изменены без изменения остальных модулей. Модули обрабатывают информацию, которая поступает как от сторонних источников, так и от внутренних подсистем.

Для управления доступом к данным в серверном приложении выделен отдельный функциональный слой доступа к данным. Данный слой служит для двунаправленной передачи данных между ядром и внешними источниками. Внешние источники представляют собой базы данных приложения, файловую систему и внешние информационные системы (ИС).

С базами данных приложение интегрируется с помощью провайдера, поставляемого разработчиком СУБД. Используется экземпляр свободной кроссплатформенной реляционной СУБД Firebird 3.0, работающей на ОС Linux, Microsoft Windows и некоторых Unix-платформах, что обеспечивает безопасность, надёжность хранения и целостность данных.



1 Серверное приложение

С внешними ИС обмен информацией реализован с помощью передачи сообщений по протоколу HTTP/HTTPS 1.1. Взаимодействие с внешними ИС выполняется по защищенному каналу связи. Поддерживаются современные протоколы шифрования TLS 1.3, 1.2, использующие ключи следующих алгоритмов шифрования RSA, ГОСТ 34.12-2018, ГОСТ Р 34.12-2015, ГОСТ 28147-89.

Данные могут поступать в серверное приложение от клиентских приложений (уровень представления). Для этого реализован слой сервиса интеграции, через который приложение получает данные от пользователей системы и передает их ядру на обработку. После обработки одним или несколькими модулями ядра

приложение сохраняет результат операций через слой доступа к данным и по необходимости уведомляет о результате выполненной работы оператора через сервис интеграции. Сервис интеграции разделен на два модуля — клиентский и публичный. Клиентский обеспечивает взаимодействие с клиентскими приложениями входящими в состав Системы. Публичный предоставляет возможность сторонним сервисам использовать часть бизнес-функций или иметь доступ к результатам выполненной работы системы.

Плагин для платформы ЕАС ОПС

Клиентское приложение Системы для настольных компьютеров представляет собой подключаемый плагин для программной платформы ЕАС ОПС. Использование клиентского приложения отдельно от платформы не допускается. Приложение написано на языке С# на базе технологии WPF. Архитектура приложения выполнена на основе шаблона проектирования MVVM.

Устанавливается приложение непосредственно на компьютерах пользователей и позволяет им выполнять бизнес-функции, определенные служебными обязанностями и правами доступа.

Для взаимодействия с серверным приложением используется протокол SOAP и WSDL 2.0. Поддерживаются протоколы безопасности SSL 3.0, TLS 1.3, 1.2.

Модуль плагина для взаимодействия с серверным приложением обеспечивает авторизацию пользователя, загрузку актуальной НСИ, а также передачу на сервер данных, внесенных пользователем.

Приложение может работать с различными СУБД. Доступ к базе реализован с использованием фреймворка Entity-Framework 6.

Модуль формирования отчетности приложения использует отечественный генератор отчетов FastReport .NET.