

**ОПИСАНИЕ ТЕХНИЧЕСКОЙ АРХИТЕКТУРЫ
ПРОГРАММНОГО КОМПЛЕКСА УПРАВЛЕНИЯ ЖИЗНЕННЫМ
ЦИКЛОМ ЗАЯВОК НА ОБСЛУЖИВАНИЕ “SERVICE DESK”**

ОГЛАВЛЕНИЕ

1.	ОБЩИЕ СВЕДЕНИЯ.....	3
2.	АРХИТЕКТУРА	3
2.1	Взаимодействие компонентов	3
2.2	Уровни архитектуры.....	4
2.3	Описание компонентов.....	4
4.	ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ	9

1. ОБЩИЕ СВЕДЕНИЯ

Документ содержит описание технической архитектуры программного комплекса управления жизненным циклом заявок на обслуживание «Service Desk» (далее – SD). Данный программный комплекс позволяет:

- управлять жизненным циклом заявок: получение от интеграций, создание, управление, выполнение и завершение;
- просматривать и перемещать остатки оборудования;
- управлять учетными записями пользователей;
- получать и настраивать отчёты;
- управлять номенклатурой;
- управлять и просматривать все этапы ремонта оборудования (терминалы самообслуживания, банкоматы, POS-терминальное и т.д.).

2. АРХИТЕКТУРА

2.1 Взаимодействие компонентов

SD имеет трехзвенную архитектуру с web-клиентом, реализующим пользовательский интерфейс с помощью любого современного web-браузера, что обеспечивает кроссплатформенность на любых рабочих станциях. Мобильное приложение инженера для работы с заявками доступно для операционных систем iOS и Android.

2.2 Уровни архитектуры

Архитектура состоит из 3 уровней:

1. Клиентская часть (web-браузер или мобильное приложение);
2. Сервер приложений;
3. Сервер баз данных.

2.3 Описание компонентов

SD состоит из следующих компонентов:

- Подсистема администрирования "SD: Администратор". Управление правами доступа и ролями, логирование событий безопасности. Настройка интеграций и бизнес-логики, автоматизация процессов, включая назначение и закрытие заявок, настройка графиков работы;
- Подсистема учета складов и оборудования "SD: Склад". Управление неограниченным количеством складов, учет оборудования по серийным номерам, мониторинг запасов и автоматизация процессов для предотвращения дефицита;
- Подсистема «SD: Интеграции». Гибкая интеграция с внешними системами через API и почтовый протокол. Подсистема подключена к сервису Vroom для сортировки полученных заявок инженера по маршруту и передает адреса заявок для прямого геокодирования в сервис Яндекса;
- Подсистема отчетности "SD: Отчеты". Формирование и кастомизация отчетов, мониторинг в реальном времени через дашборды. Контроль SLA, автоматический расчет и прогнозирование нагрузки для оптимизации ресурсов;
- Подсистема сервисного центра "SD: Ремонт". Управление всеми стадиями ремонта, контроль процессов для поддержания качества и

минимизации простоев. Эффективное управление ресурсами ремонтных центров;

- Подсистема работы с заявками "SD: Заявки". Обеспечивает полный контроль жизненного цикла заявок и детальный анализ. Автоматическое распределение и уведомления ускоряют обработку и предотвращают просрочки;

- Справочная подсистема "SD: Номенклатура". Централизованный учет моделей, модулей, партномеров и расходников. Номенклатура содержит актуальные справочники, связанные с оборудованием, передаёт все обновления справочников в подсистему "SD: Склад" через брокер очередей RabbitMQ;

- Сервис "Идентификация пользователей". Управление аутентификацией и авторизацией пользователей. Единый вход (SSO), управление ролями и правами доступа, интеграция с различными системами и протоколами безопасности, такими как OAuth2 и SAML. Централизованное управление учетными записями пользователей, обеспечение высокого уровня безопасности и контроля доступа к подсистемам;

- Мобильное приложение инженера для работы с заявками. Работа с заявками и отчетами в любое время, поддержка бизнес-процессов, геопозиция, уведомления, фотофиксация, прикрепление чеков, электронные акты, новости и обновления ПО, расчет сроков SLA.

Подсистемы "SD: Заявки", "SD: Интеграции", "SD: Склад", "SD: Номенклатура" и API для мобильного приложения для работы с заявками подключены к бакетам объектного хранилища S3 от Яндекс cloud.

На рисунке 2.1 показана схема взаимодействия компонентов SD.

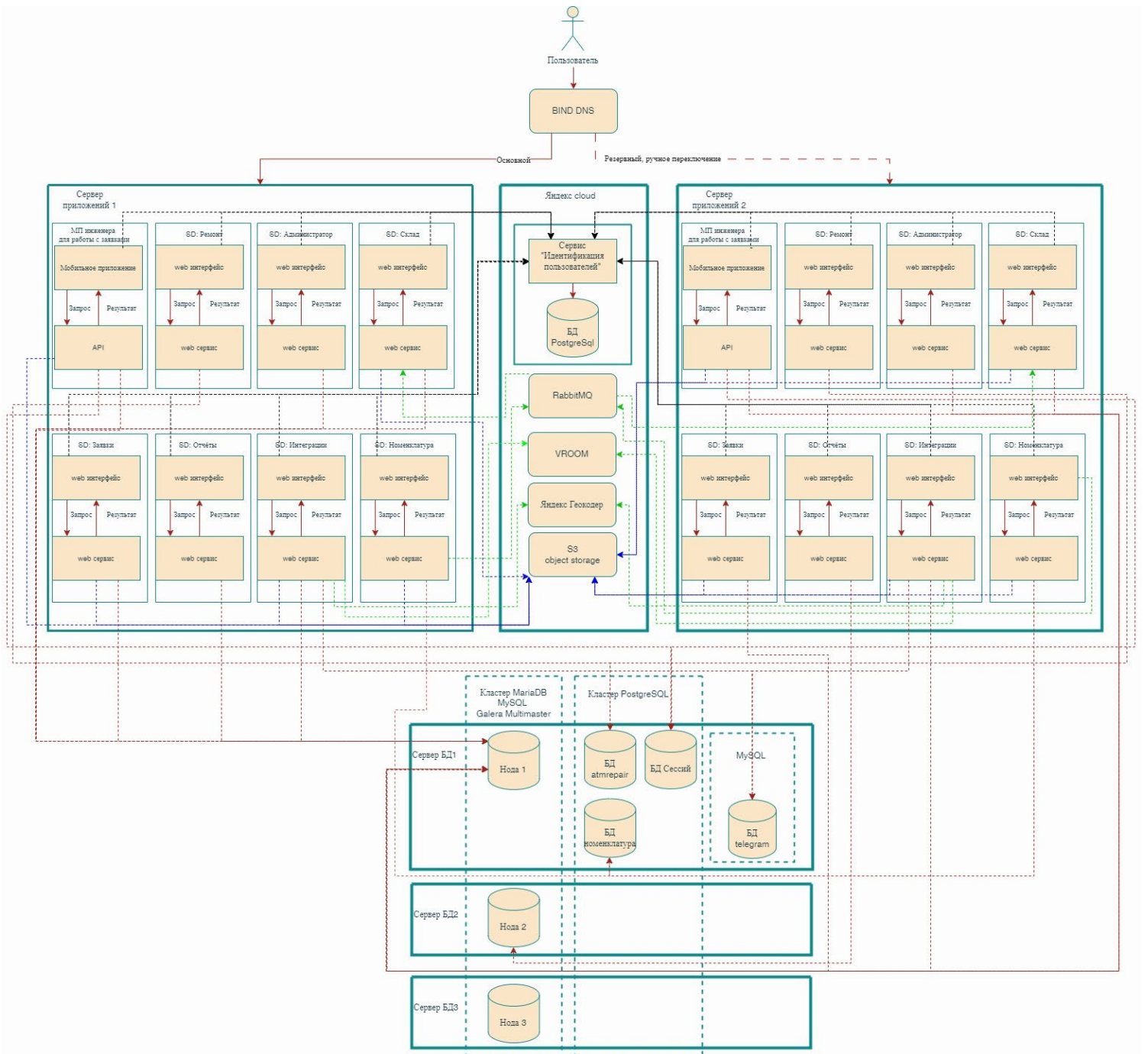


Рисунок 2.1 – Схема взаимодействия компонентов SD

4. ТЕХНОЛОГИЧЕСКИЙ СТЕК

3.1 Серверная часть

- Языки и фреймворки: PHP (Yii2, Laravel), JavaScript (React, React Native), Golang;
- Веб-сервер: Apache HTTP Server, Nginx;
- Ос: Linux / Unix;
- Контейнеризация: Debian (Docker), Kubernetes (k3s);
- Кэширование: Redis.

3.2 Базы данных

- СУБД: MariaDB 11, PostgreSQL 14.

3.3 Инфраструктура и интеграции

- Брокер очередей: RabbitMQ, Kafka;
- Объектное хранилище: S3 (Yandex Cloud);
- Протоколы интеграции: REST API, SMTP;
- Геокодирование: DaData, Яндекс.Геокодер;
- Маршрутизация: VROOM, OpenRouteService;
- Картография: OpenStreetMap, Яндекс.Карты.

3.4 Безопасность и аутентификация

- SSO-сервер: Keycloak;
- Протоколы: OAuth2, SAML;
- Функции: единый вход (SSO), управление ролями и правами доступа.

3.5 Мобильное приложение

- Платформы: iOS (от версии 14), Android (от версии 10);
- Магазины: App Store, RuStore.

5. ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ

Ниже представлены рекомендуемые требования для полноценной работы для 2000 пользователей со средней нагрузкой в 5000 заявок в сутки и для взаимодействия с 70 интеграциями.

1) Требования к серверу приложений:

- Операционная система: Linux/Unix;
- Процессор: 48 ядер, 2,3 ГГц;
- Оперативная память: 128 Гб;
- SSD: 2 x 1 Тб (Raid 1);
- Сеть: 500 Мбит;

2) Требования к серверу баз данных:

- Операционная система: Linux/Unix;
- Процессор: 24 ядер, 2,3 ГГц;
- Оперативная память: 128 Гб;
- SSD: 2 x 1 Тб (Raid 1);
- Сеть: 500 Мбит;

3) Требования к клиентским машинам:

- Операционная система: Linux/Unix/Windows (не младше Windows 7);
- Оперативная память: 8 Гб;
- Процессор: 4 ядра;
- Web-браузер: Chrome/Yandex Browser/Microsoft Edge;
- Дополнительное ПО: офисный пакет LibreOffice/OpenOffice.

4) Требования к смартфонам:

- Операционная система: android от 9 версии/ios от 14 версии;

- Наличие камеры;
- Включенный модуль GPS;
- Не менее 1Гб свободного места.