

**«Demand Response Mobile»**

# **ОБЩЕЕ ОПИСАНИЕ СИСТЕМЫ**

## СОДЕРЖАНИЕ

<b>1. ОБЩИЕ ПОЛОЖЕНИЯ.....</b>	<b>3</b>
1.1. Полное наименование системы и ее условное обозначение .....	3
1.2. Сведения об организации Правообладателя и Разработчика.....	3
1.3. Основные понятия, определения и сокращения .....	3
<b>2. НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ .....</b>	<b>7</b>
2.1. Назначение Системы .....	7
2.2. Цели создания Системы .....	7
2.3. Перечень диспетчерских центров АО «СО ЕЭС», в которых предполагается использовать Систему .....	7
2.4. Перечень функций Системы .....	7
<b>3. ОПИСАНИЕ СИСТЕМЫ.....</b>	<b>9</b>
3.1. Структура Системы и назначение ее частей .....	9
<b>4. КОМПЛЕКС ТЕХНИЧЕСКИХ СРЕДСТВ .....</b>	<b>12</b>
4.1. Архитектура Системы .....	12
4.2. Информационная безопасность .....	17
4.3. Аппаратные требования для эксплуатации Системы.....	17
4.4. Программное обеспечение .....	17

## 1. ОБЩИЕ ПОЛОЖЕНИЯ

### 1.1. Полное наименование системы и ее условное обозначение

Полное наименование программы для ЭВМ: «Demand Response Mobile».

Условное обозначение – DRM, Система.

### 1.2. Сведения об организации Правообладателя и Разработчика

**Правообладатель:** Акционерное общество «Системный оператор Единой энергетической системы» (АО «СО ЕЭС») – специализированная организация, единолично осуществляющая централизованное оперативно-диспетчерское управление в Единой энергетической системе России.

**Разработчик:** Акционерное общество «Научно-технический центр Единой энергетической системы Информационные комплексы» (АО «НТЦ ЕЭС Информационные комплексы»).

### 1.3. Основные понятия, определения и сокращения

<b>API</b>	Описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой
<b>CPU</b>	Центральное процессорное устройство
<b>CSS</b>	Формальный язык описания внешнего вида документа, написанного с использованием языка разметки.
<b>DMZ</b>	Сегмент сети, содержащий общедоступные сервисы и отделяющий их от частных.

<b>Docker</b>	Программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации.
<b>DR</b>	Demand Responce
<b>DRM</b>	Demand Response Mobile
<b>HDD</b>	Постоянное запоминающее устройство
<b>HTML</b>	Стандартизированный язык гипертекстовой разметки документов для просмотра веб-страниц
<b>HTTP</b>	Протокол прикладного уровня передачи данных.
<b>HTTPS</b>	Расширение протокола HTTP для поддержки шифрования в целях повышения безопасности.
<b>IMAP</b>	Протокол доступа к электронной почте
<b>IMAPS</b>	Протокол доступа к электронной почте с поддержкой SSL
<b>Java</b>	Строго типизированный объектно-ориентированный язык программирования общего назначения
<b>JavaScript</b>	Прототипно-ориентированный сценарный язык программирования
<b>JDBC</b>	Платформенно независимый промышленный стандарт взаимодействия Java-приложений с различными СУБД
<b>JSON</b>	Текстовый формат обмена данными, основанный на JavaScript
<b>JWS</b>	JSON Web Signature — это открытый стандарт для подписи произвольных данных
<b>Nexus</b>	Менеджер репозиторий, предназначенный для проксирования репозиторий и хранения ПО.

<b>RAM</b>	Оперативное запоминающее устройство
<b>REST</b>	Архитектурный стиль взаимодействия компонентов распределённого приложения в сети. REST представляет собой согласованный набор ограничений, учитываемых при проектировании распределённой гипермедиа-системы
<b>TLS</b>	Протокол защиты транспортного уровня
<b>SMTP</b>	Сетевой протокол, предназначенный для передачи электронной почты
<b>SMTPS</b>	Сетевой протокол, предназначенный для передачи электронной почты с поддержкой SSL
<b>SSH</b>	Сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой
<b>SSL</b>	Криптографический протокол, обеспечивающий защищённую передачу данных между узлами в сети.
<b>SPA</b>	Веб-приложение или веб-сайт, использующий единственный HTML-документ как оболочку для всех веб-страниц и организующий взаимодействие с пользователем через динамически подгружаемые HTML, CSS, JavaScript
<b>WAF</b>	Межсетевой экран для веб-приложений, который выявляет разнообразные информационные атаки
<b>Агрегатор</b>	или субъект электроэнергетики – это организации, которые выявляют и координируют способность группы конечных потребителей управлять своей нагрузкой и конвертируют ее в товары и услуги на рынках электроэнергии, мощности и системных услуг.

<b>АО «НТЦ ЕЭС Информационные комплексы»</b>	Акционерное общество «Научно-технический центр Единой энергетической системы Информационные комплексы»
<b>АО «СО ЕЭС»</b>	Акционерное общество «Системный оператор Единой энергетической системы»
<b>БД</b>	База данных
<b>ИА</b>	Исполнительный аппарат АО «СО ЕЭС»
<b>НСИ</b>	Нормативно-справочная информация
<b>ПАК ЕСМ</b>	Программно-аппаратный комплекс «Единая система мониторинга» АО «СО ЕЭС»
<b>Объект управления, ОУ</b>	Совокупность энергопринимающих устройств потребителей, участвующих в групповом управлении изменением нагрузки.
<b>ОС</b>	Операционная система
<b>ПО</b>	Программное обеспечение
<b>СУБД</b>	Система управления базами данных
<b>ФПА</b>	Фонд программ и алгоритмов АО «СО ЕЭС»
<b>Энергопринимающее устройство, ЭУ</b>	Совокупность всех энергопринимающих устройств, объектов электросетевого хозяйства и (или) объектов по производству электрической энергии, находящихся в границах балансовой принадлежности указанного энергопринимающего устройства.

## **2. НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ**

### **2.1. Назначение Системы**

Система предназначена для оперативного отображения субъектам электроэнергетики (Агрегаторам), заключившим договор на оказание услуг по управлению спросом с АО «СО ЕЭС», информации, связанной с деловыми процессами в области управления спросом на электроэнергию.

### **2.2. Цели создания Системы**

Целью создания Системы является улучшение качества процессов в области управления спросом на электроэнергию за счет:

- консолидации информации в единое информационное пространство;
- сокращения времени и трудоемкости задач по доведению результатов расчета до Агрегаторов.
- повышения степени оперативного информирования Агрегаторов о ключевых событиях.

### **2.3. Перечень диспетчерских центров АО «СО ЕЭС», в которых предполагается использовать Систему**

Система разработана для использования в ИА.

### **2.4. Перечень функций Системы**

Основные функции Системы сгруппированы в следующие функциональные разделы:

- 1) Отображение НСИ для объектов управления и энергопринимающих устройств Агрегаторов.

- 2) Отображение ежедневного статуса готовности к снижению электропотребления для объектов управления и энергопринимающих устройств Агрегаторов.
- 3) Отображение статуса наличия события по управлению спросом в любой день расчетного периода.
- 4) Отображение и графическая визуализация результатов расчета событий по управлению спросом для Агрегаторов.
- 5) Формирование и доведение до Агрегаторов информационных сообщений или оповещений.

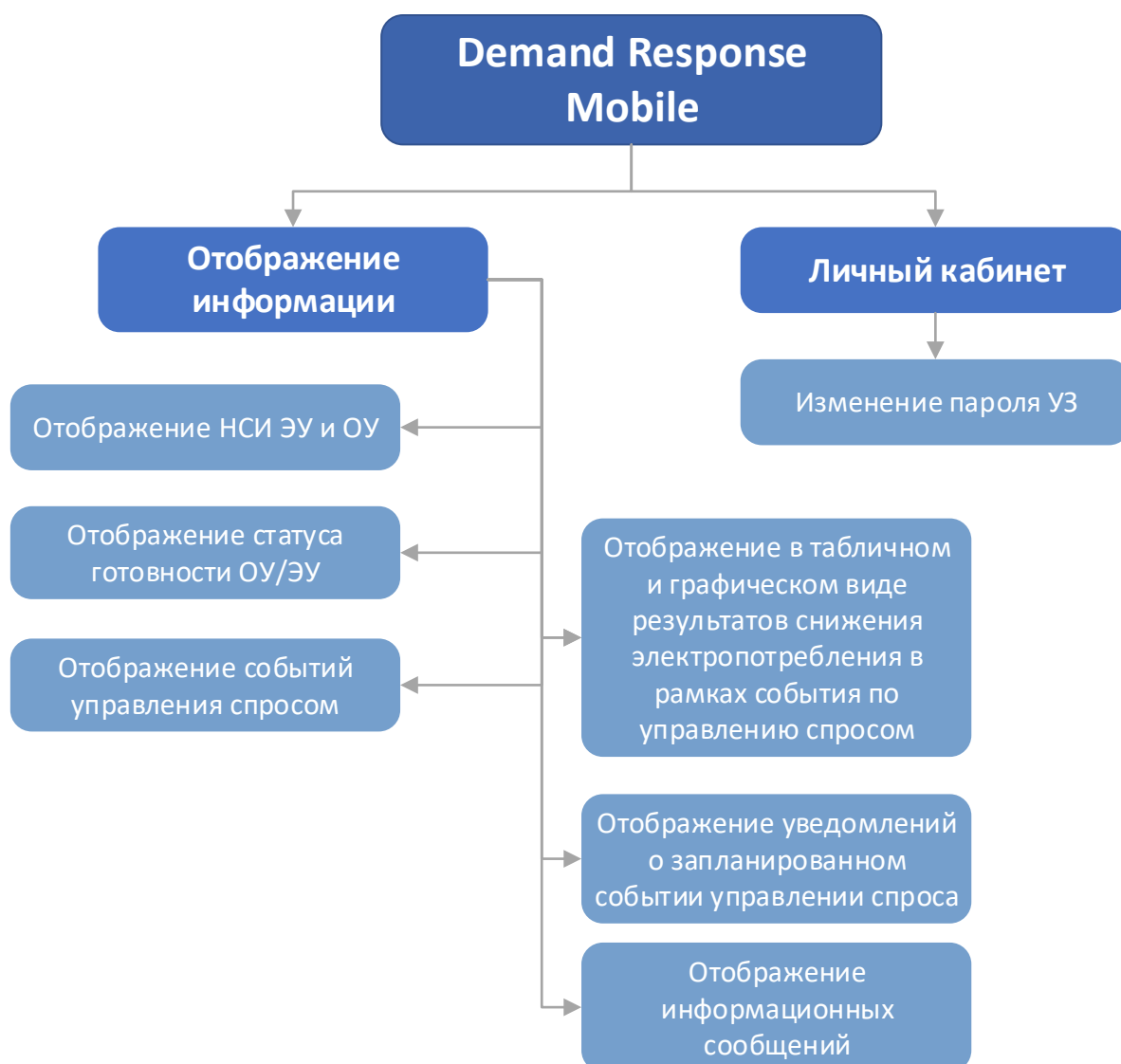


Рисунок 1 – Функциональная структура Системы



### **3. ОПИСАНИЕ СИСТЕМЫ**

#### **3.1. Структура Системы и назначение ее частей**

DRM является подсистемой программного обеспечения «Demand Response».

«Demand Response» имеет централизованную структуру – компоненты обработки и хранения данных размещены на серверном оборудовании ИА.

В качестве архитектуры «Demand Response» был выбран сервис-ориентированный подход. «Demand Response» состоит из набора взаимосвязанных сервисов, которые общаются между собой посредством REST API, а в качестве формата данных используют JSON. Для хранения данных выбрана СУБД PostgreSQL, а в качестве веб и обратного прокси сервера Nginx.

На рисунке 2 представлена схема взаимодействия сервисов «Demand Response». Красными прямоугольниками выделены сервисы, относящиеся к DRM и сервисы, работа которых необходима для функционирования Системы.

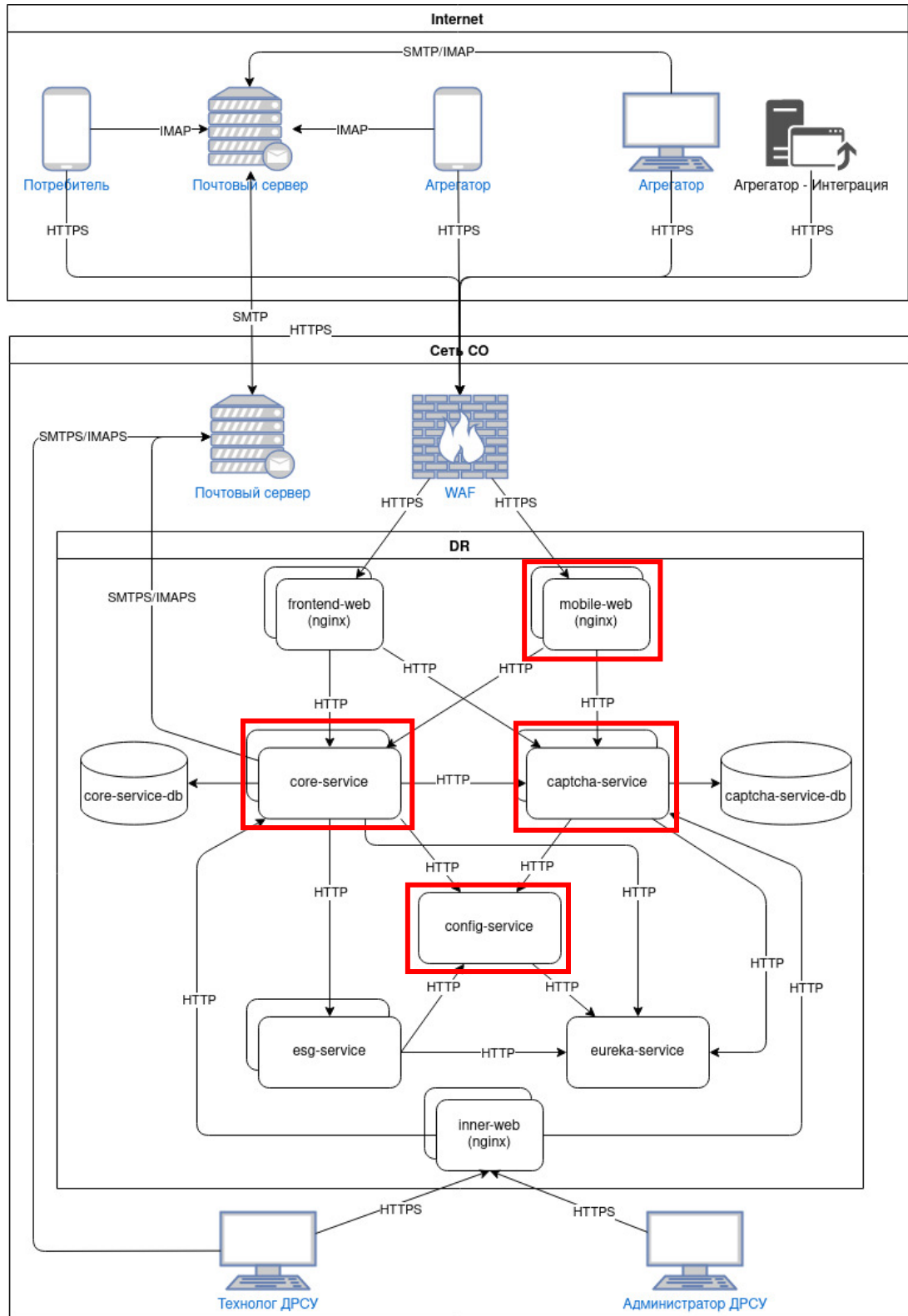


Рисунок 2 – Схема взаимодействия сервисов «Demand Response»

- **core-service** – основной сервис «Demand Response». Данный сервис обеспечивает работу всех бизнес-функций, в т.ч. функций DRM. В качестве языка разработки данного сервиса используется Java. Дистрибутив поставляется в виде Docker образа.
- **captcha-service** – вспомогательный сервис «Demand Response». Основная задача данного сервиса – защита от подбора паролей путем запроса ввода капчи после неудачной аутентификации пользователя. Сервис осуществляет генерацию капчи в виде изображения и ее проверку после ввода на форме авторизации. В качестве языка разработки используется Java. Дистрибутив поставляется в виде Docker образа.
- **config-service** – вспомогательный сервис «Demand Response». Данный сервис хранит распределенную конфигурацию для всех остальных сервисов и служит единой точкой для конфигурации. В качестве языка разработки используется Java. Дистрибутив поставляется в виде Docker образа.
- **mobile-web** – представляет собой SPA приложение (сайт), которое выполняется в браузере клиента и размещается на веб-сервере Nginx. Данное приложение предназначено для внешних пользователей – Агрегаторов, подключенных с **мобильных устройств**. В качестве языка разработки используется Java Script с использованием библиотеки React Native. Дистрибутив поставляется в виде zip архива, содержащего набор html, js, css файлов.

Описание остальных сервисов приводится «Demand Response» приводится в документе «Общее описание системы «Demand Response».

## 4. КОМПЛЕКС ТЕХНИЧЕСКИХ СРЕДСТВ

### 4.1. Архитектура Системы

«Demand Response» включает в себя серверы СУБД, серверы приложений и веб-серверы. Все сервера размещаются на уровне ИА. Веб-серверы размещаются в сегменте DMZ, сервера приложений и БД в корпоративном сегменте. Для функционирования «Demand Response» необходим доступ к ФПА Nexus для обновления приложений, обновления Docker engine с официального репозитория, обновления пакетов ОС с официального репозитория.

Приложения «Demand Response» запускаются в Docker-контейнерах. Под серверы допускается выделение виртуальных машин.

Эксплуатация Системы должна выполняться в инфраструктуре вычислительного центра АО «СО ЕЭС» с учетом обеспечения технической и физической защиты аппаратных компонентов Системы, носителей данных, бесперебойного энергоснабжения, текущего сервисного обслуживания.

Предлагаемый подход предусматривает возможность масштабирования Системы по производительности.

**Требования к сети:** каналы связи не менее 100 МВ между узлами Системы. Рекомендованная пропускная способность между сайтами Системы и пользователем – не менее 10МВ.

Отказоустойчивость «Demand Response» реализуется посредством двойного резервирования узлов сети, и создания резервных копий данных, отказ отдельных зарезервированных узлов не приводит к выходу из строя Системы, время на восстановление узла сети из резервной копии 24 часа. Отказоустойчивость Системы достигается за счет дублирования сервисов и

распределения запросов между каждым экземпляром сервиса по алгоритму round-robbin (перебор по кругу).

Резервная копия узлов Системы выполняется 1 раз в месяц с хранением копии до 3 месяцев. Резервная копия СУБД выполняется ежедневно в полном объеме с хранением резервных копий за последние 14 дней.

Контроль работоспособности Системы осуществляется ПАК ЕСМ посредством мониторинга показателей ОС и доступности сетевых портов сервисов Системы.

Таблица 1 – Используемые стандартные платформы и библиотеки

Сервис	Платформа	Язык	Фреймворки/ Библиотеки	Сборщик	Формат дистрибутива
mobile-web	V8 engine (ES6)	JavaScript	react (17.0.2)	webpack (5.28.0) + npm (6.14.10)	Zip архив
core-service	JRE 1.8	Java	spring-boot(2.3.x) spring-web(5.2.x) spring-data-jpa(2.5.x) spring-cloud (Hoxton.SR8) liquibase-core(4.4.x) postgresql(42.2.x) jjwt(0.9.1) spring-ldap-core(2.3.x) handlebars(4.2.0) poi-ooxml(5.0.0) lombok(1.18.x)	Maven (3.8.2)	Docker image
captcha-service	JRE 1.8	Java	spring-boot(2.2.x) spring-web(5.2.x) spring-data-jpa(2.5.x) spring-cloud (Hoxton.SR4) liquibase-core(4.4.x) postgresql(42.2.x) simplecaptcha(1.2.1- mvn) lombok(1.18.x)	Maven (3.8.2)	Docker image

Сервис	Платформа	Язык	Фреймворки/ Библиотеки	Сборщик	Формат дистрибутива
config- service	JRE 1.8	Java	spring-boot(2.2.x) spring-web(5.2.x) spring-cloud (Hoxton.SR4) lombok(1.18.x)	Maven (3.8.2)	Docker image

На рисунке 3 представлена схема взаимодействия серверов «Demand Response».

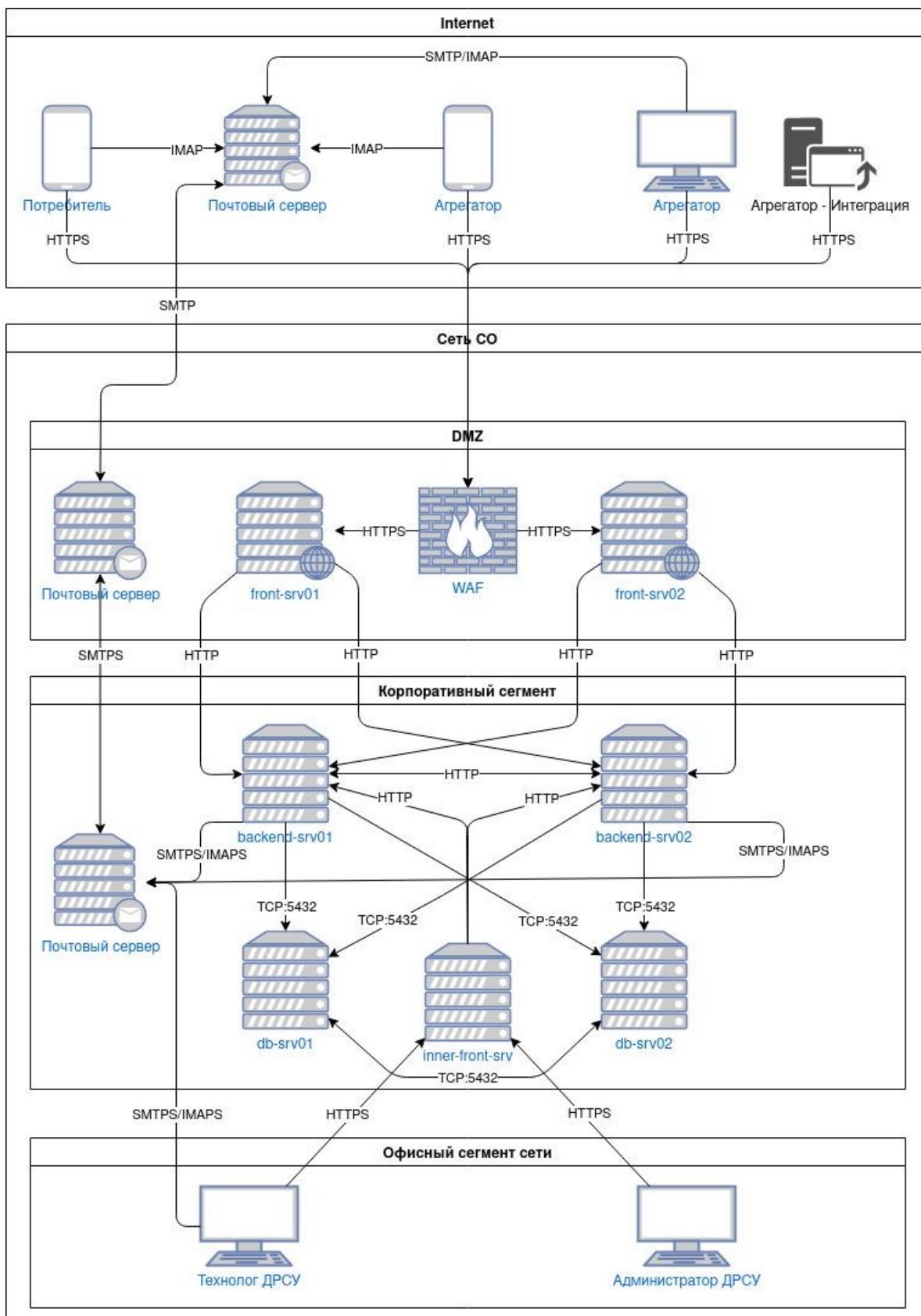


Рисунок 3 – Схема серверов «Demand Response»

## Конфигурация серверов для промышленной эксплуатации

Для запуска сервисов «Demand Response» предлагается следующее разделение по серверам.

Таблица 2 – Серверы «Demand Response»

Тип сервера	Кол-во	ОС	Запущенные сервисы
front-srv0№	2	Astra Linux	nginx frontend-web <b>mobile-web</b>
inner-front-srv	1	Astra Linux	nginx inner-web
backend-srv0№	2	Astra Linux	<b>core-service</b> <b>captcha-service</b> esg-service eureka-service <b>config-service</b>
db-srv0№	2	Astra Linux	postgresql

Таблица 3 – Технические характеристики серверов

Тип сервера	Кол-во	Характеристики сервера		
		CPU, core	RAM, Gb	HDD, Gb
front-srv0№	2	4	4	30
inner-front-srv	1	4	4	30
backend-srv0№	2	6	12	50
db-srv0№	2	4	8	600



## **4.2. Информационная безопасность**

«Demand Response» эксплуатируется в сети с ограниченным доступом. Пользователи Системы получают информацию через веб-приложения, которые доступны из сети Интернет.

Сетевой доступ к серверам Системы предоставляется исключительно Администратору системы и ПАК ЕСМ.

Доступ к серверам «Demand Response» осуществляется по протоколу SSHv2.

Авторизация сервисов в СУБД происходит при помощи JDBC использованием шифрования TLS.

Для авторизации внешних и внутренних пользователей на мобильном и основном сайтах Системы используются JWS токены со сроком жизни refresh токена в 1 неделю и привязкой к cookie браузера.

Для авторизации внутренних пользователей на внутреннем сайте используются JWS токены с бессрочным refresh токеном и привязкой к ip-адресу пользователя.

Для первичного получения access и refresh токенов внутренние пользователи должны пройти процедуру аутентификации.

## **4.3. Аппаратные требования для эксплуатации Системы**

Нагрузка на Систему не постоянна и зависит от количества Агрегаторов, обрабатываемых Системой. В таблице 3 представлены требования к аппаратным ресурсам серверов необходимых для стабильного функционирования Системы при одновременной работе 100 пользователей.

## **4.4. Программное обеспечение**

Серверы приложений:

- Операционная система – Astra Linux Common Edition;
- ПО Docker Engine версии 19.03+.

Серверы СУБД:

- Операционная система – Astra Linux Common Edition;
- СУБД – PostgreSQL версии 9.6;
- Диски в конфигурации LVM;
- ПО Keepalived.

Web-серверы:

- Операционная система – Astra Linux Common Edition;
- ПО Nginx версии 1.16.1+;