

**Программное обеспечение «Базис Документы»**

**Общее описание функциональных характеристик программного обеспечения и информации, необходимой для установки и эксплуатации программного обеспечения**

Листов 9

1. Концепция	3
1.1. Роли пользователей	3
2. Решения по архитектуре	4
2.1. Функциональная архитектура	5
2.1.1. Описание взаимодействия	6
2.1.2. Функциональные характеристики программного обеспечения	8
2.1.3. Требования к программному и аппаратному обеспечению компьютера пользователя	8
2.1.3.1. Требования к программному обеспечению ПК пользователя:	8
2.1.3.2. Требования к сети	8
2.1.3.3. Требования к аппаратному обеспечению	8
2.1.3.4. Требования к персоналу (пользователю)	8
2.1.4. Загрузка, запуск Системы и завершение работы Системы	9
2.1.4.1. Загрузка Системы	9
2.1.4.2. Запуск Системы	9
2.1.4.3. Завершение работы Системы	9

## 1. Концепция

Базис Документы (далее – Система) – это цифровой B2B-сервис, позволяющий построить совершенно уникальный пользовательский опыт взаимодействия с документами. Представляет из себя комплексное использование технологии нейронных сетей и продвинутой системы ручного распознавания для типизации, валидации и распознавания данных. Сервис пользуется спросом у компаний из разнообразных сфер: от банковского сектора до ЖКХ. Система решает комплекс задач по автоматизации обработки клиентских документов:

- извлечение и распознавание данных из документа с использованием нейронных сетей и системы ручного распознавания;
- передача извлечённых данных в ИС пользователя сервиса.

Результатом обработки является распознанные данные для каждого документа, который передается в информационную систему пользователя сервиса для дальнейшей обработки согласно бизнес-процессу.

### 1.1. Роли пользователей

Для промышленного контура и контуров тестирования используются различные учетные записи администраторов системы.

Таблица 3. Роли пользователей

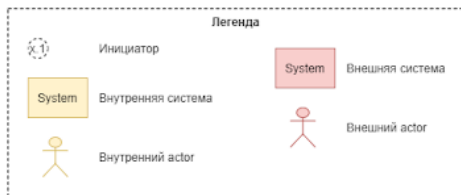
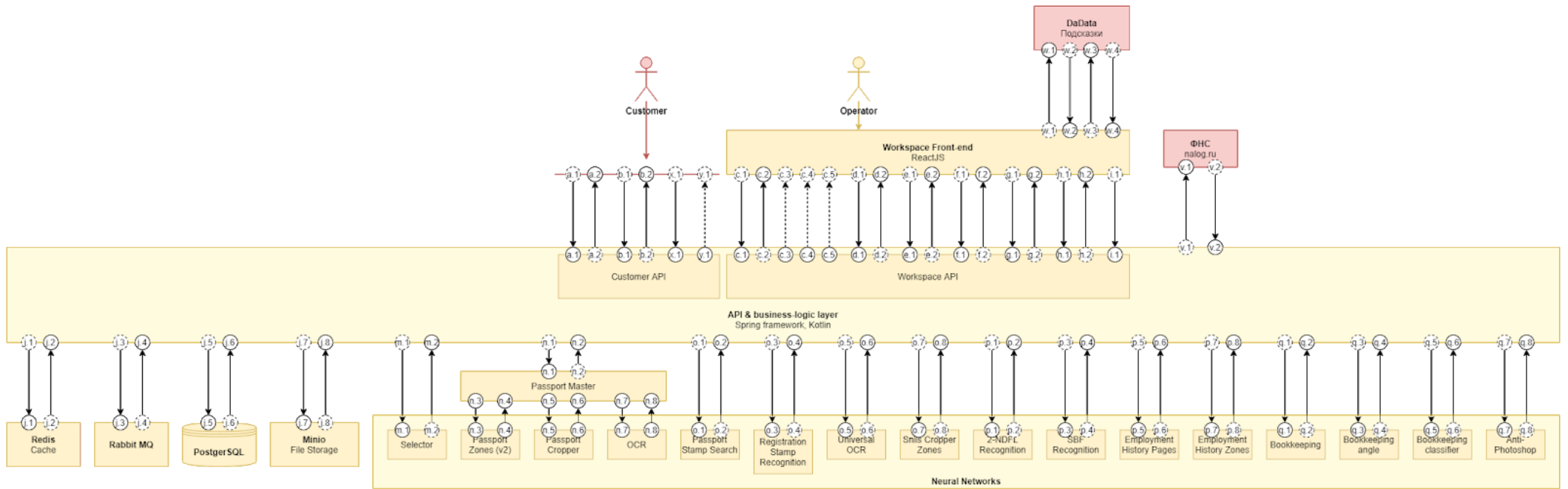
Роль	Описание доступных действий и полномочий
Администратор баз данных Postgres	- обеспечивает настройку параметров работы базы данных; - обеспечивает работоспособность базы данных; - обеспечивает выявление ошибок в работе базы данных и их устранение.
Прикладной администратор	- обеспечивает настройку параметров работы прикладного программного обеспечения; - обеспечивает работоспособность прикладного программного обеспечения; - обеспечивает выявление ошибок в работе прикладного программного обеспечения и их устранение; - обеспечивает выгрузку данных из базы данных прикладного программного обеспечения.
Специалист внедрения	- обеспечивает установку ПО; - обеспечивает настройку ПО в процессе внедрения/отмены внедрения новых версий ПО;

Аудитор информационной безопасности	<ul style="list-style-type: none"> <li>- осуществляет просмотр и анализ событий и действий пользователей, зарегистрированных в журналах аудита;</li> <li>- осуществляет просмотр локальных групп пользователей.</li> </ul>
Аудитор	<ul style="list-style-type: none"> <li>- определяет события, которые должны регистрироваться в журналах безопасности операционных систем и СУБД;</li> <li>- осуществляет настройку локальной политики аудита ОС и СУБД, если политика не определена для всего домена Microsoft Active Directory;</li> <li>- осуществляет настройку локальных журналов аудита;</li> <li>- обеспечивает настройку аудита доступа к файлам и папкам;</li> </ul> <p>(указанные выше полномочия могут быть отнесены к компетенции других административных ролей);</p> <ul style="list-style-type: none"> <li>- осуществляет просмотр и анализ событий и действий пользователей, зарегистрированных в журналах безопасности ОС и СУБД.</li> </ul>
Аналитик	<ul style="list-style-type: none"> <li>- осуществляет просмотр и анализ программных логов и объектов базы данных с целью анализа расхождений с задокументированной функциональностью системы.</li> </ul>
Верификатор	<ul style="list-style-type: none"> <li>- осуществляет ручное распознавание документов с помощью специального пользовательского интерфейса.</li> </ul>

## 2. Решения по архитектуре

Архитектура системы основана на построении и развертывании микросервисов на предварительно подготовленной инфраструктуре. Система строится на основе программного обеспечения для автоматизации развертывания и управления в средах с поддержкой контейнеризации Docker, с оркестровкой и управлением, разворачиваемая на базе OS Linux.

## 2.1. Функциональная архитектура



### 2.1.1. Описание взаимодействия

Взаимодействие внутренних систем пользователя с сервисом осуществляется через микросервис Customer API. Протокол взаимодействия с внешними системами-источниками – REST.

#### Схема процесса распознавания:

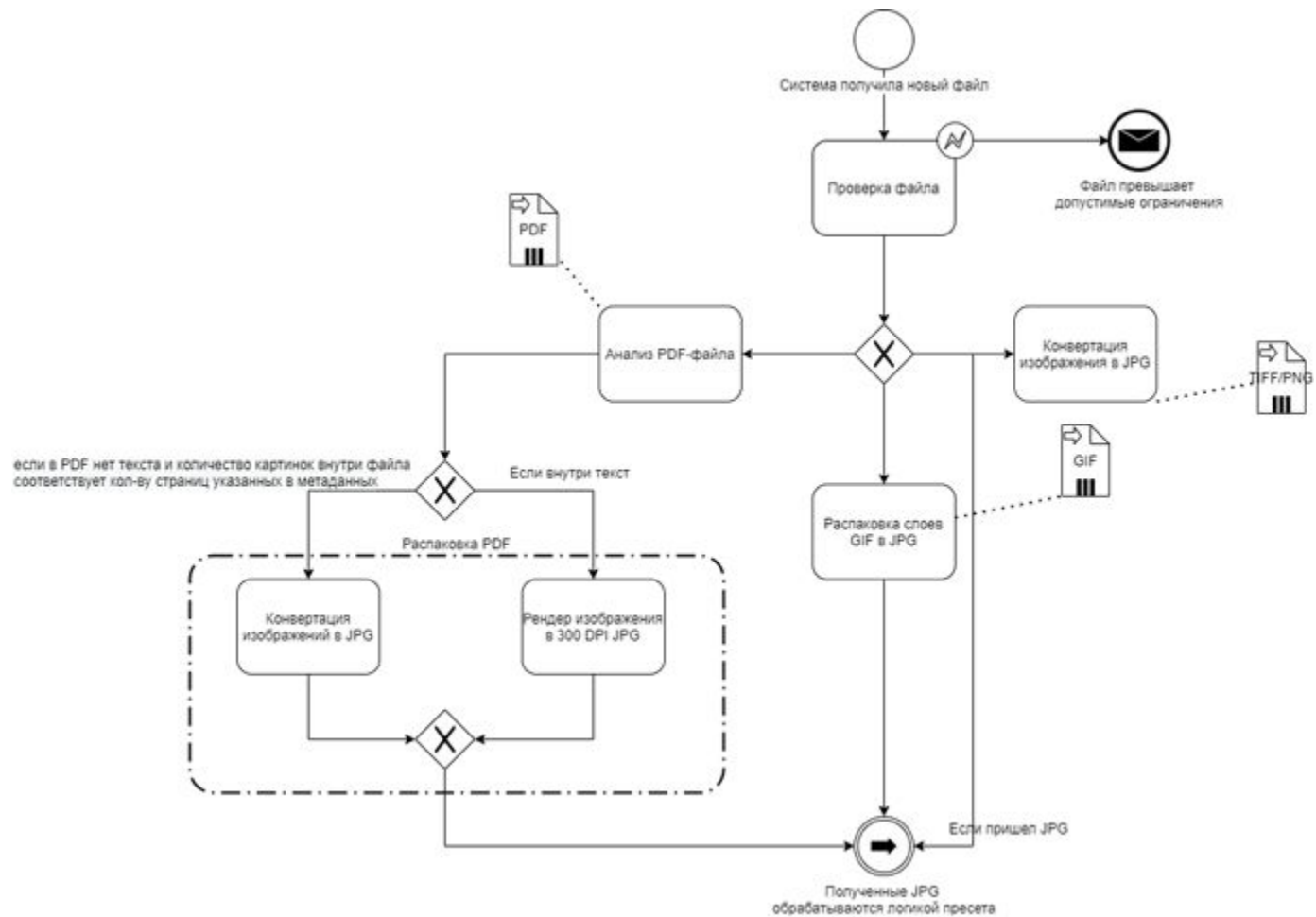
- Подключение к базе данных всех микросервисов производится с использованием стандартных средств dataSource на уровне конфигураций микросервисов, где указывается параметры для подключения к базе данных, а именно: host, port и schema.
- Логин и пароль от базы данных хранится в secrets и передаются в микросервис через переменные окружения. При подключении к базе данных, микросервис, передает логин и пароль в базу данных.
- Подключение к очереди RabbitMQ всех микросервисов производится аналогичным образом, как и описано выше. В конфигурациях каждого микросервиса, использующего в работе RabbitMQ указываются параметры для подключения к менеджеру очередей: host, port.
- Логин и пароль, аналогичным образом размещаются в secrets и передаются в микросервис через переменные окружения.

#### Описание процесса распознавания:

- В микросервис /access-tokens поступает запрос на аутентификацию, в котором указывается логин и пароль системы.
- В теле ответа возвращается токен аутентификации, с которым выполняются последующие запросы.
- В микросервис /tasks поступает запрос на создание задачи на распознавание документов, в котором указываются имена файлов на сетевом каталоге или сами файлы, которые необходимо обработать.
- В теле ответа возвращается квиток с ответом, что процесс распознавания запущен и идентификатор задачи распознавания.
- Система-источник запроса периодически опрашивает микросервис /tasks/{id} для получения статуса распознавания.
- Если распознавание завершено, то в результатах распознавания будет соответствующий статус и все распознанные параметры.

В рамках задачи по распознаванию документов в решении предусмотрены следующие функциональные операции:

- Аутентификация (синхронная операция)
- Отправка запроса на распознавание (Асинхронная операция).
- Получение результата распознавания (Синхронная операция).



### Доступ к системе через RESTful API:

Интеграционное взаимодействие при аутентификации, создании задач на распознавание и получении результатов распознавания с системой осуществляется через HTTP с использованием RESTful API в следующем порядке:

- Для аутентификации используется микросервис /access-tokens, в который передается логин и пароль системы-источника, и в ответе возвращается аутентификационный токен в поле token тела ответа. С полученным токеном выполняются запросы на распознавание и получение результатов распознавания.

- После получения аутентификационного токена система-источник создает задание на распознавание путем вызова микросервиса /tasks с предоставлением файла для распознавания. В ответ возвращается идентификатор задания на распознавание.
- Получение результатов распознавания осуществляется путем вызова микросервиса /tasks с указанием идентификатора задания на распознавание, полученного при создании задания.

### **2.1.2. Функциональные характеристики программного обеспечения**

Система позволяет:

- извлекать данные из поступающих в обработку сканированных образов документов (а именно электронная копия документа в графическом формате). Форматы документов, принимаемые Системой: pdf, png, jpg, jpeg, pjpeg, gif, tiff или zip;
- выполнять автоматическую проверку на качество сканирования документа (возможность распознавания данных документа машинным способом);
- выполнять автоматическое распознавание данных, полученных из обработанных сканированных образов документов и ручную верификацию.

### **2.1.3. Требования к программному и аппаратному обеспечению компьютера пользователя**

2.1.3.1. Требования к программному обеспечению ПК пользователя:

- Операционная система Linux семейства Debian;
- Программное обеспечение Docker и Docker Compose

Требований к программному обеспечению конечного пользователя не установлено.

2.1.3.2. Требования к сети

Рекомендованная входящая/исходящая скорость соединения – от 256 кбит/с.

2.1.3.3. Требования к аппаратному обеспечению

Требования к пользовательскому аппаратному обеспечению, для работы с системой не предъявляются.

2.1.3.4. Требования к персоналу (пользователю)

Для эксплуатации Системы предъявляются следующие требования к квалификации конечных пользователей:



- базовые навыки работы на персональном компьютере с современными офисными приложениями и операционной системой (клавиатура, мышь, управление окнами и приложениями, файловая система);
- базовые навыки использования интернет-браузера (установка подключений, доступ к веб-сайтам, навигация, формы и другие типовые интерактивные элементы веб-интерфейса).

## 2.1.4. Загрузка, запуск Системы и завершение работы Системы

### 2.1.4.1. Загрузка Системы

Конечному пользователю не требуется установка Системы. Вся функциональность доступна конечному пользователю через HTTPS с использованием RESTful API. На компьютеры конечных пользователей выводятся лишь результаты распознавания, которые происходят на серверах.

### 2.1.4.2. Запуск Системы

Для запуска системы необходимо:

1. Направить на адрес [doc.support@basis.center](mailto:doc.support@basis.center) запрос в свободной форме на подключение к системе Базис.Документы;
2. По результатам подключения в ответном письме будут направлены данные для аутентификации (получения токена):
  - a. логин и пароль, с которыми выполняется дальнейшее взаимодействие с системой Базис.Документы;
  - b. Пресет (настройка), с которой необходимо выполнять запросы на распознавание;
  - c. Ссылка на Систему.

Интеграционное взаимодействие при аутентификации, создании задач на распознавание и получении результатов распознавания с системой осуществляется через HTTP с использованием RESTful API в следующем порядке:

- Для аутентификации используется микросервис `/access-tokens`, в который передается логин и пароль системы-источника, и в ответе возвращается аутентификационный токен в поле `token` тела ответа. С полученным токеном выполняются запросы на распознавание и получение результатов распознавания.
- После получения аутентификационного токена система-источник создает задание на распознавание путем вызова микросервиса `/tasks` с предоставлением файла для распознавания. В ответ возвращается идентификатор задания на распознавание.
- Получение результатов распознавания осуществляется путем вызова микросервиса `/tasks` с указанием идентификатора задания на распознавание, полученного при создании задания.

### 2.1.4.3. Завершение работы Системы

В Системе не предусмотрена возможность завершения работы приложения конечным пользователем. Пользовательский сеанс (сеанс распознавания) считается завершенным в тот момент, когда конечный пользователь получает результаты распознавания.