

**ПЛАТФОРМА ДЛЯ СОЗДАНИЯ, РАЗВИТИЯ И ЭКСПЛУАТАЦИИ
ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ЕДИНОЙ
ЦИФРОВОЙ ПЛАТФОРМЫ РОССИЙСКОЙ ФЕДЕРАЦИИ «ГОСТЕХ»**

ОБЩЕЕ ОПИСАНИЕ ПЛАТФОРМЫ

на 121 листе

2021 г

АННОТАЦИЯ

Настоящий документ является общим описанием Платформы для создания, развития и эксплуатации прикладного программного обеспечения для единой цифровой платформы Российской Федерации «ГосТех» (далее - Платформа).

Настоящий документ описывает Платформу в целом.

СОДЕРЖАНИЕ

Перечень терминов, сокращений и обозначений	4
1 Назначение Системы	8
1.1 Вид деятельности, для автоматизации которой предназначена система	8
1.2 Перечень объектов автоматизации, на которых используется система	9
1.3 Перечень функций, реализуемых системой	9
2 ОПИСАНИЕ СИСТЕМЫ	10
2.1 Структура Платформы и назначение ее частей	10
3 ОПИСАНИЕ ВЗАИМОСВЯЗИ АС С ДРУГИМИ СИСТЕМАМИ	25
3.1 Перечень систем, с которыми связана Платформа	25
3.2 Описание связей между системами	25
3.3 Описание регламента связи	26
3.4 Описание взаимосвязей Платформы с подразделениями объекта автоматизации	27
4 ОПИСАНИЕ ПОДСИСТЕМ	28
4.1 Описание функционирования подсистем и их частей	28
4.1.1 Облачная платформа Сбербанка	28
4.1.2 Инструменты управления контейнерами	36
4.1.3 Инструменты управления производственным процессом «SberWorks»	37
4.1.4 Компоненты управления данными	52
4.1.5 Компоненты аналитики данных	77
4.1.6 Компоненты интеграции	81
4.1.7 Инструменты и компоненты безопасности Platform V IAM	100
4.1.8 Интерфейсные компоненты	104

Перечень терминов, сокращений и обозначений

Используемые в настоящем документе термины и основные понятия области автоматизированных систем определены в ГОСТ 34.003-90. Также в тексте настоящего документа введены специальные термины:

Обозначение	Описание
Компонент Платформы	Структурный элемент Платформы, обеспечивающий реализацию части функционала для разработки Приложений на Платформе
Платформа	Платформа для создания, развития и эксплуатации прикладного программного обеспечения для единой цифровой платформы Российской Федерации «ГосТех»
Пользователь Платформы	Участник команды разработки Ведомства (инженер, разработчик, тестировщик, аналитик и др.), использующий Платформу в целях создания государственных информационных систем и их компонентов
Пользователь приложений на Платформе (далее по тексту Конечный Пользователь).	Лицо, использующее функции государственных информационных систем и их компонентов, созданных и эксплуатирующихся на Платформе, допускается определение по тексту Конечный Пользователь
Экземпляр	Установленный экземпляр ПО, обладающий собственным идентификатором и набором данных.
Кластер	Группа серверов, объединённых логически, способных обрабатывать идентичные запросы и использующихся как единый ресурс. Объединение серверов в один ресурс происходит на уровне программных протоколов.

В настоящем документе введены следующие термины и специальные сокращения на русском и английском языках:

Сокращение	Полное наименование
Веб-браузер	Прикладное программное обеспечение для просмотра страниц, содержания веб-документов, компьютерных файлов и их каталогов; управления веб-приложениями; а также для решения других задач

Сокращение	Полное наименование
Гб	Гигабайт (сокр. Гбайт; Гб, международное сокр.: Gbyte, GB) — единица измерения количества информации
ГОСТ	Государственный стандарт
ЕСИА	Информационная система в Российской Федерации, обеспечивающая санкционированный доступ участников информационного взаимодействия (граждан-заявителей и должностных лиц органов исполнительной власти) к информации, содержащейся в государственных информационных системах и иных информационных системах
ПО	Программное обеспечение
СМЭВ	Единая система межведомственного электронного взаимодействия (СМЭВ) - федеральная государственная информационная система, предназначенная для организации информационного взаимодействия между информационными системами
ФСТЭК России	Федеральная служба по техническому и экспортному контролю (ФСТЭК России) — федеральный орган исполнительной власти России, осуществляющий реализацию государственной политики, организацию межведомственной координации и взаимодействия, специальные и контрольные функции в области государственной безопасности
ЭП	Электронная подпись
API	Application Programming Interface — программный интерфейс приложения) — это набор способов и правил, по которым различные программы общаются между собой и обмениваются данными
CPU	Центральный процессор (ЦП; также центральное процессорное устройство — ЦПУ; англ. central processing unit, CPU, дословно — центральное обрабатывающее устройство, часто просто процессор) — электронный блок либо интегральная схема, исполняющая машинные инструкции (код программ), главная часть аппаратного обеспечения компьютера или программируемого логического контроллера

Сокращение	Полное наименование
Ghz	Гигагерц (англ. gigahertz) ГГц — это единица измерения тактовой частоты ЦП
HTTP	HTTP (англ. HyperText Transfer Protocol — «протокол передачи гипертекста») — протокол прикладного уровня передачи данных, изначально — в виде гипертекстовых документов в формате HTML, в настоящее время используется для передачи произвольных данных
IOPS	IOPS (аббревиатура от англ. input/output operations persecond — количество операций ввода-вывода в секунду; произносится как «ай-опс») — количество операций ввода-вывода, выполняемых системой хранения данных, за одну секунду
RAM	Оперативная память (англ. Random Access Memory, RAM, память с произвольным доступом) или оперативное запоминающее устройство (ОЗУ) — энергозависимая часть системы компьютерной памяти, в которой во время работы компьютера хранится выполняемый машинный код (программы), а также входные, выходные и промежуточные данные, обрабатываемые процессором
REST	REST (Representational state transfer) – это стиль архитектуры программного обеспечения для распределенных систем, таких как World Wide Web, который, как правило, используется для построения веб-служб. Термин REST был введен в 2000 году Роем Филдингом, одним из авторов HTTP-протокола. Системы, поддерживающие REST, называются RESTful-системами
SATA	SATA (англ. Serial ATA) — последовательный интерфейс обмена данными с накопителями информации. SATA является развитием параллельного интерфейса ATA (IDE), который после появления SATA был переименован в PATA (Parallel ATA)
SAS	Serial Attached SCSI (SAS) — последовательный компьютерный интерфейс, разработанный для подключения различных устройств хранения данных, например, жёстких дисков и ленточных накопителей

Сокращение	Полное наименование
SSD	SSD (solid state drive, твердотельный накопитель) — это энергонезависимое запоминающее устройство, которое использует флэш-память для хранения информации
ssh	SSH (англ. Secure Shell — «безопасная оболочка») — сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений (например, для передачи файлов)
VM	Виртуальная машина, сервер с установленной операционной системой
vCPU	Виртуальное ядро, часть ресурсов физического процессора предоставленная виртуальной машине

1 Назначение Системы

Платформа для создания, развития и эксплуатации прикладного программного обеспечения для единой цифровой платформы Российской Федерации «ГосТех» предназначена для организации производственного и технологического процесса разработки и обслуживания программного обеспечения при создании и развитии информационных систем и их компонентов на единой цифровой платформе Российской Федерации «ГосТех», в том числе для разработки и «беспроводного» подключения новых каналов обслуживания граждан, организаций и органов власти с использованием облачных технологий и обеспечением безопасности информации и масштабируемости исполняемых на платформе приложений. Основным назначением Платформы является автоматизация технологических аспектов жизненного цикла приложений: непрерывное внедрение изменений без приостановки обслуживания, защиту от сбоев в следствии человеческого фактора; мониторинг состояния системы, локализацию корневых причин проблем и инцидентов; корректную работу и целостность данных при отказе инфраструктурных элементов, отказе интеграций, аномальных всплесках нагрузки; линейное масштабирование приложений; изоляцию разных потребителей по взаимовлиянию; централизованное управление поведением приложений; набор инструментов автоматизирующих штатные операции разработчиков приложений на платформе; моделирование и выполнение пользовательских сценариев и бизнес-процессов.

1.1 Вид деятельности, для автоматизации которой предназначена система

Платформа предназначена для автоматизации процессов разработки, тестирования, эксплуатации и развития государственных информационных систем и их компонентов и является безопасной, надежной и масштабируемой облачной средой исполнения приложений, создаваемых на платформе.

1.2 Перечень объектов автоматизации, на которых используется система

Объектами автоматизации являются процессы разработки, тестирования, эксплуатации и развития государственных информационных систем и их компонентов.

1.3 Перечень функций, реализуемых системой

Платформа обеспечивает предоставление набора инструментов и компонентов для организации производственного процесса и ведения разработки приложений:

- набор инструментов, позволяющих автоматизировать подготовку инфраструктуры для приложений;
- набор инструментов, поддерживающих производственный процесс создания приложений;
- набор вспомогательных компонентов и инструментов, облегчающих и ускоряющих реализацию бизнес-логики приложений, реализацию различных типов интеграций, организацию хранения и обработку данных, машинного обучения.

2 ОПИСАНИЕ СИСТЕМЫ

2.1 Структура Платформы и назначение ее частей

Платформа предназначена для организации производственного и технологического процесса разработки программного обеспечения при создании, переводе и развитии государственных информационных систем и их компонентов на единой цифровой платформе Российской Федерации «ГосТех», в том числе для разработки и «бесшовного» подключения новых каналов обслуживания граждан, организаций и органов власти с использованием облачных технологий и обеспечением безопасности информации и масштабируемости.

Целями использования Платформы для Пользователей Платформы является:

- организация производственного и технологического процесса разработки программного обеспечения для создания государственных информационных систем и их компонентов, в том числе с использованием облачных технологий;

- апробация подходов к применению аппаратно-программных комплексов в целях создания государственно-информационных систем и их компонентов;

- обеспечение единой среды создания и исполнения приложений с требуемым уровнем надежности, безопасности и масштабируемости;

- создание условий для сокращения сроков прохождения аттестации по требованиям безопасности информации во ФСТЭК России государственных информационных систем и их компонентов, разработанных с помощью Платформы, в том числе за счет использования готовых компонентов: идентификации, аутентификации, авторизации и аудита (в том числе регистрация событий безопасности), обеспечивающих соблюдение требований ФСТЭК России по информационной безопасности.

- Целями использования Платформы для Конечных Пользователей является:

- обеспечение комплексного взаимодействия между гражданами, государственными органами и коммерческими организациями;

- обеспечение доступа к качественным и быстро создаваемым функциям государственных информационных систем и их компонентов.

Технологическая платформа Platform V – набор программных продуктов, предоставляющих совокупность функциональных возможностей и позволяющий обеспечить быстрое конструирование информационных систем из множества готовых компонентов. Решает задачи масштабирования и безопасности. Максимально снимает задачи технического уровня с разработчиков прикладной логики.

Технологическая платформа Platform V позволяет решить следующие задачи:

- ускорить выпуск новых продуктов;
- обеспечить надежность и производительность;
- обеспечить целостность и качество данных;
- повысить утилизацию оборудования;
- обеспечить независимость от иностранных вендоров;
- устранить дублирование затрат между подразделениями и непрозрачность расходов.

Компоненты Технологической платформы Platform V обеспечивают:

- исполнение типовых задач эксплуатации приложений, включая: развертывание, конфигурирование, мониторинг, журналирование, трассировка;
- среду интеграции приложений в парадигме поставщик/потребитель API по моделям запрос/ответ и публикация/подписка, а также инструменты интеграции этой среды со внешним миром;
- возможности визуального проектирования, исполнения, и мониторинга оркестрируемых технологических и бизнес-сценариев;
- эффективное достижение требуемых нефункциональных характеристик работы интегрированной системы приложений, включая:
 - производительность;
 - надежность;
 - доступность;
 - безопасность.

На рисунке ниже (см. Рисунок 1) представлена структуры Платформы.

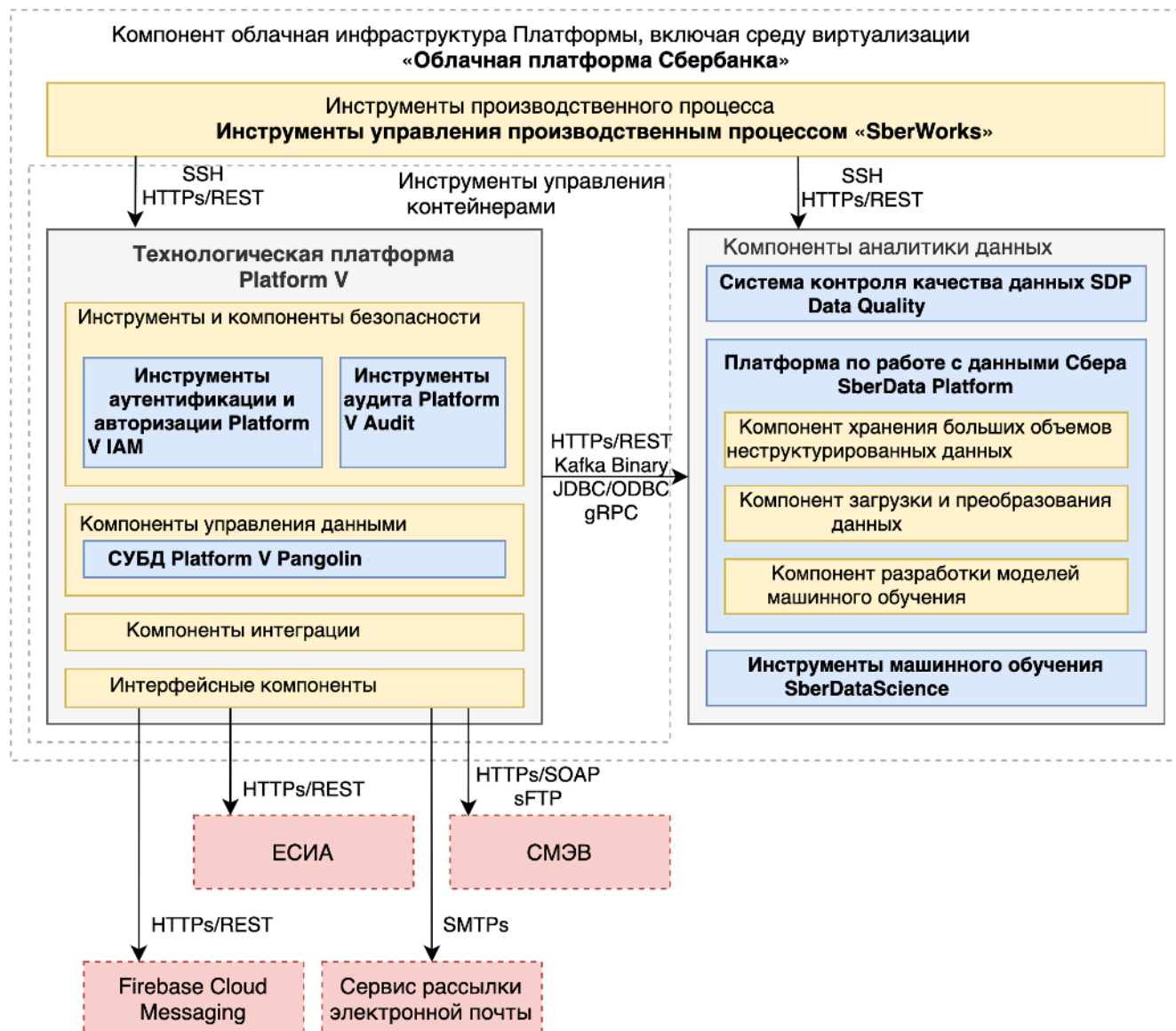


Рисунок 1 - Структура Платформы

Платформа состоит из следующих основных сегментов:

– облачная инфраструктура Платформы, включая среду виртуализации – «Облачная платформа Сбербанка», обеспечивающая предоставление функционирующим на её базе компонентам, инструментам и сервисам Платформы услуги по модели обслуживания IaaS;

– инструменты управления контейнерами – свободно распространяемое программное обеспечение (Open Source Software), отвечающие за автоматизацию и управление жизненным циклом контейнеров и сервисов. В жизненный цикл контейнера входит планирование ресурсов, управление масштабируемостью, правила балансировки нагрузки и контроль доступности, а также организация виртуальных сетей;

– инструменты управления производственным процессом – инструменты «SberWorks», обеспечивающие организацию единой среды разработки для инженеров и разработчиков, с возможностью управления всеми этапами производственного процесса: от разработки идей до внедрения;

– компоненты управления данными – компоненты работы с данными для решения таких задач как: работа со структурированными и слабоструктурированными данными, хранение данных в оперативной памяти, сбор данных из разных источников, управление качеством данных, распределенное хранение данных;

– компоненты аналитики данных – платформенные компоненты, предназначенные для обработки больших массивов структурированных и неструктурированных данных: преобразование «сырых» данных, пересчитанных данных (витрин данных) в информацию, пригодную для принятия решений;

– компоненты интеграции – платформенные компоненты, обеспечивающие реализацию интеграционной логики приложений в архитектуре SOA, которые призваны ускорить и упростить разработку различных типов интеграций приложений;

– инструменты и компоненты безопасности – инструмент, функциональные компоненты которого обеспечивают идентификацию, аутентификацию, авторизацию пользователей, а также регистрацию событий аудита (в том числе событий безопасности) в журнале аудита;

– интерфейсные компоненты – набор инструментов, реализующих функциональность пользовательского взаимодействия с системой, моделирования бизнес-процессов, интеграциями с сервисами регистрации/авторизации пользователей через системы ЕСИА, СМЭВ, ОТТ. Данный сегмент Платформы включает также инструменты для обеспечения функциональности по отправке/получению push-уведомлений, заказу отчётности, журналированию событий приложения и хранению справочно-конфигурационных данных.

В составе основных сегментов Платформы выделяются следующие компоненты:

– облачная инфраструктура Платформы, включая среду виртуализации:

- компонент «Облачная платформа Сбербанка. Подсистема виртуализации вычислительных ресурсов», предназначенный для предоставления набора вычислительных ресурсов или их логического объединения, абстрагированного от аппаратной реализации, и обеспечивающего при этом логическую изоляцию друг от друга вычислительных процессов, выполняемых на одном физическом ресурсе;
- компонент «Облачная платформа Сбербанка. Подсистема вычислительной сети», предназначенный для создания и управления программно-определяемой сетью (SDN);
- компонент «Облачная платформа Сбербанка. Подсистема хранения данных», предназначенный для создания и управления программно-определяемой системой хранения (SDS);
- компонент «Облачная платформа Сбербанка. Подсистема мониторинга», предназначенный для сбора, хранения и визуализации метрик утилизации ресурсов;
- компонент «Облачная платформа Сбербанка. Подсистема управления», предназначенный для управления инфраструктурой Платформы;
- компонент «Облачная платформа Сбербанка. Библиотека шаблонов и образов», предназначенный для создания, хранения и запуска шаблонов и образов элементов инфраструктуры;
- компонент «Облачная платформа Сбербанка. Подсистема оркестрации», предназначенный для автоматизации управления жизненным циклом элементов инфраструктуры и прикладных приложений;
- компонент «Облачная платформа Сбербанка. Подсистема резервного копирования», предназначенный для создания, хранения и восстановления из резервных копий;
- инструменты управления контейнерами, свободно распространяемое ПО (Open Source Software), предназначенное для обеспечения автоматизации и управления жизненным циклом контейнеров и сервисов;
- инструменты управления производственным процессом «SberWorks»:

- инструмент управления планированием, предназначенный для управления жизненным циклом целей и задач;
- инструмент управления требованиями, предназначенный для управления жизненным циклом требований;
- инструмент управления релизами, предназначенный для учёта и ведения релизной деятельности, и управления жизненным циклом релизов;
- инструмент управления дефектами, предназначенный для управления жизненным циклом дефектов;
- инструмент управления тестированием, предназначенный для автоматизации процессов тестирования и ведения отчётности по проведённым испытаниям;
- инструмент управления репозиториями дистрибутивов, предназначенный для обеспечения организации систем хранения артефактов программных продуктов: дистрибутивов и пакетов для установки ПО, библиотек для разработки;
- инструмент версионного контроля исходного кода и конфигураций, предназначенный для версионного контроля исходного кода и конфигураций;
- инструмент анализа качества кода, предназначенный для анализа кода и визуализации результатов;
- инструмент проектирования API и архитектурного контроля «МЕТА», предназначенный для проектирования интеграционных взаимодействий и управления их жизненным циклом;
- инструменты управления поставкой ПО, предназначенные для обеспечения непрерывности цикла сборки, поставки и развертывания ПО;
 - инструмент управления сборкой ПО, предназначенный для организации непрерывного процесса сборки и проверки качества исходного кода ПО;
 - инструмент управления поставками дистрибутивов, предназначенный для контроля и управления конвейерами поставок;

- инструмент управления развертыванием ПО, предназначенный для организации непрерывного процесса развертывания и автоматического тестирования приложений;
- инструмент аналитики и мониторинга производственного процесса, предназначенный для сбора, хранения, обработку и визуализацию метрик производственного процесса разработки приложений;
- компоненты управления данными:
 - компонент «СУБД Platform V Pangolin», предназначенный для обеспечения управления высоко-доступными, отказоустойчивыми объектно-реляционными базами данных в приложениях любого класса критичности;
 - компонент «Распределенное хранение и обработка данных в памяти Platform V Ignite Sber Edition», предназначенный для обеспечения надежного распределенного хранения данных в оперативной памяти и распределенную обработку этих данных;
 - компонент хранения больших объемов неструктурированных данных «Платформа по работе с данными Сбера SberData Platform», предназначенный для обеспечения хранения данных в объемах, исчисляемых петабайтами, а также высокую скорость обработки данных с учетом распределенной модели исполнения запросов;
 - компонент загрузки и преобразования данных «Платформа по работе с данными Сбера SberData Platform», предназначенный для обеспечения агрегации данных из различных источников для наполнения хранилищ данных в соответствии с потребностью бизнес-модели;
 - компонент «Прикладной журнал Platform V Data Tools», предназначенный для обеспечения функционала приёма, хранения и передачи векторов изменений данных на уровне приложения;
 - компонент «Архивирование Platform V Data Tools», предназначенный для обеспечения инкрементальной загрузки данных из прикладных приложений в компоненты аналитики данных;

- компонент «Виртуализация слоя хранения данных Hibernate Platform V Data Tools», предназначенный для обеспечения унифицированного доступа к данным в реляционных базах данных посредством подхода object-to-relational mapping (ORM);
- компонент генерации синтетических тестовых данных «Synteta», предназначенный для создания набора синтетических тестовых данных для прохождения сквозных тестовых сценариев по нескольким интегрированным приложениям, через API, предоставляемые этими приложениями;
- компоненты автонумерации и генерации уникальных идентификаторов, предназначенные для обеспечения функциональности централизованной выдачи отформатированных номеров для сквозной нумерации бизнес-сущностей и генерации идентификаторов сущностей на множестве узлов приложения:
 - компонент «Генерация уникальных ID Platform V Generators», предназначенный для обеспечения функциональности централизованной выдачи отформатированных номеров для сквозной нумерации бизнес-сущностей;
 - компонент «Автонумерация Platform V Generators», предназначенный для генерации идентификаторов сущностей на множестве узлов приложения;
- компонент «Управление справочной информацией Platform V Dictionaries», предназначенный для работы со справочной информацией плоской структуры из прикладных приложений, реализованных на Платформе;
- компоненты аналитики данных:
 - компонент управления качеством данных «Система контроля качества данных SDP Data Quality», предназначенный для управления качеством данных;

- компонент разработки моделей машинного обучения «Платформа по работе с данными Сбера SberData Platform», предназначенный для обеспечения поставки и подготовки данных, исследования данных и проверки гипотез, а также построения моделей;
- компонент управления жизненным циклом моделей машинного обучения «SberDataScience», предназначенный для загрузки данных, обучения моделей машинного обучения, валидации моделей машинного обучения, исполнения моделей машинного обучения, мониторинга моделей машинного обучения и формирования отчетности.
- компоненты интеграции:
 - компонент «HTTP-клиент Platform V Synapse Service Mesh», предназначенный для использования в составе прикладного приложения при работе с HTTP-запросами;
 - компонент «REST-запросы Platform V Synapse Service Mesh», предназначенный для использования в составе прикладных приложений при создании их REST-интерфейсов;
 - компоненты проксирования запросов:
 - компонент «Сервисный прокси Platform V Synapse Service Mesh», предназначенный для управления трафиком интеграционных сценариев-сервисов;
 - компонент «Граничный прокси Platform V Synapse Service Mesh», предназначенный для обеспечения управляемого вызова интеграционных компонентов прикладных приложений;
 - компонент «Шлюзы AC Platform V Synapse Enterprise Integration», предназначенный для организации интеграции микросервисных прикладных приложений, размещённых в облаке, с приложениями вне облака;
 - компонент внутреннего и внешнего шлюза «Внутренний шлюз Platform V API Management», предназначенный для организации балансировки трафика на узлы прикладных приложений;

- компонент «Доступ к единице топологии Платформы Platform V Application Sharding», предназначенный для маршрутизации запросов к экземплярам прикладных приложений в геораспределенной топологии с учетом возможности одновременной эксплуатации нескольких контуров/инсталляций Платформы;
- компоненты событийной передачи и обработки событий:
 - компонент «Передача событий Platform V Synapse Event Processing», предназначенный для передачи публикуемых поставщиками событий потребителям в рамках одного событийного домена;
 - компонент «Потоковая обработка событий Platform V Synapse Event Processing», предназначенный для потоковой обработки событий, поступающих из одного или нескольких событийных доменов;
- компонент «Обмен сообщениями Platform V Kafka Sber Edition», предназначенный для обмена сообщениями между компонентами Платформы и прикладными приложениями;
- компоненты межкластерной индексации и маршрутизации:
 - компонент «Межкластерная индексация Platform V Application Sharding», предназначенный для управления данными, необходимыми для реализации горизонтального масштабирования;
 - компонент «Маршрутизация Platform V Application Sharding», предназначенный для управления данными маршрутизации конечных пользователей Платформы на целевые контуры;
- компонент «Ведение метаинформации программных сервисов APIM Platform V API Management», предназначенный для хранения и управления метаинформацией программных сервисов, осуществления передачи конфигураций программных сервисов на шлюзы, предоставления средств управления шлюзами API;

- компоненты моделирования композитных сервисов, предназначенные для обеспечения разработки и выполнения композитных сервисов в контейнерной среде:
 - компонент «Моделирование композитных сервисов Ядро КБТ Platform V Functions», предназначенный для выполнения композитных сервисов в контейнерной среде;
 - компонент «Моделирование композитных сервисов Визард КБТ Platform V Functions», предназначенный для разработки композитных сервисов;
- компонент «Файловая передача Platform V Synapse File Exchange», предназначенный для передачи файлов между компонентами Платформы и прикладными приложениями;
- компоненты асинхронных запусков пакетных вычислительных задач, предназначенные для обеспечения функциональности асинхронного запуска пакетных вычислительных задач по расписанию и в порядке очереди:
 - компонент «Планировщик заданий Platform V Batch», предназначенный для обеспечения функциональности асинхронного запуска пакетных вычислительных задач по расписанию;
 - компонент «Пакетная обработка задач Platform V Batch», предназначенный для обеспечения функциональности асинхронного запуска пакетных вычислительных задач в порядке очереди;
- инструменты и компоненты безопасности:
 - компонент «Аутентификация Platform V IAM («Модули IAM. Технологическая платформа Сбербанка»)), предназначенный для обеспечить идентификацию, аутентификацию конечных пользователей Платформы;
 - компоненты авторизации, предназначенные для авторизации доступа конечных пользователей Платформы к функционалу прикладных приложений на основе проверки ролей и прав доступа, а также на основе атрибутов субъектов и объектов доступа;

- компонент «Авторизация ЕФС Platform V IAM («Модули IAM. Технологическая платформа Сбербанка»)), предназначенный для авторизации доступа конечных пользователей Платформы к функционалу прикладных приложений на основе проверки ролей и прав доступа, а также на основе атрибутов субъектов и объектов доступа;
- компонент «Авторизация ППРБ Platform V IAM («Модули IAM. Технологическая платформа Сбербанка»)), предназначенный для авторизации доступа конечных пользователей Платформы к функционалу прикладных приложений на основе проверки прав пользователя;
- компонент «Аудит Platform V Audit («Модуль Аудит. Технологическая платформа Сбербанка»)), предназначенный для регистрации, сбора и хранения событий аудита;
- интерфейсные компоненты:
 - компоненты моделирования и исполнения бизнес-сценариев:
 - компонент «Моделирование и исполнение бизнес-сценариев Engine Platform V Flow», предназначенный для исполнения экземпляров бизнес-процессов;
 - компонент «Моделирование и исполнение бизнес-сценариев Designer Platform V Flow», предназначенный для визуального проектирования бизнес-процессов в нотации BPMN;
 - компонент «Моделирование и исполнение бизнес-сценариев Tasklist Platform V Flow», предназначенный для работы с пользовательскими задачами в рамках исполнения бизнес-процессов;
 - компонент «Управление статическим и динамическим контентом Platform V Content Management», предназначенный для управления статическими данными справочников Ошибки и Подсказки, а также маскирования данных;
 - компонент «Формирование печатных форм Platform V Print», предназначенный для генерации электронных печатных форм документов по размеченным шаблонам;

- компоненты управление сессиями:
 - компонент «Стартовый менеджер Platform V Starting Manager», предназначенный для инициализации сессионного контекста аутентифицированного конечного пользователя Платформы и вычисление конфигурации фронтального приложения в соответствии с ролевой моделью;
 - компонент «Сессионные данные Platform V User Sessions», предназначенный для временного хранения оперативных и справочно-конфигурационных данных по конечному пользователю Платформы;
- компонент «Центр уведомлений Platform V Notification Center», предназначенный для доставки конечным пользователям Платформы уведомлений и иных видов сообщений;
- компонент «Интеграция с ЕСИА Platform V ESIA Gateway», предназначенный для получения данных из ЕСИА прикладными приложениями, создаваемыми на базе Платформы;
- компонент «Интеграция со СМЭВ Platform V SMEV Gateway», предназначенный для интеграции прикладных приложений с единой системой межведомственного электронного взаимодействия (СМЭВ);
- компонент «Журналирование Platform V Monitor», предназначенный для сбора, хранения, обработки и визуализации цепочек событий;
- компонент «Platform V One-Time-Token», предназначенный для аутентификации и авторизации межсервисных вызовов для компонентов Платформы с использованием механизма токенов;
- компонент «Электронная подпись Platform V Secret Management», предназначенный для формирования и проверки электронной подписи;
- компонент «Управление настройками приложений Platform V Configuration», предназначенный для хранения, управления и предоставления по запросу параметров конфигурации компонентов Платформы и прикладных приложений, разработанных на Платформе;
- компоненты прикладного мониторинга:

- компонент «Прикладной мониторинг Platform V Monitor», предназначенный для сбора, обработки, агрегирования и визуализации метрик о прикладных приложениях или компонентах Платформы;
- компонент «Хранение метрик Abyss Platform V Monitor», предназначенный для хранения, обработки и агрегирования о прикладных приложениях или компонентах Платформы;
- компонент «Управление данными Platform V DataSpace», предназначенный для проектирования, расширения модели данных микросервисного приложения на платформе и генерации базовых сервисов для работы с данными на основе модели;
- компонент «Обеспечение надёжности Failover Platform V Monitor», предназначенный для обеспечения обнаружения и устранения неисправностей в работе серверов приложений RedHat Wildfly Application Server и прикладных приложений под его управлением;
- компонент «Ведение электронного документооборота Platform V Documents», предназначенный для работы с объектами документа и выполнения базовых операций документооборота.

Платформа функционирует на базе общесистемного программного обеспечения, которое включает:

- операционные системы серверов (ОС);
- среды оркестрации контейнеров;
- сервер приложений (RedHat Wildfly Application Server);
- веб-серверы (Nginx);
- служба мониторинга (Prometheus);
- службы репликации данных (Apache Ni-Fi);
- службы полнотекстового индексирования и поиска (ElasticSearch);
- службы визуализации данных ElasticSearch (Kibana);
- Zookeeper;

- Docker-nexus3;
- Jenkins;
- Curator;
- GIT;
- KeyCloak;
- Python;
- Node.js;
- Grafana;
- веб-браузер.

В качестве операционных систем используются:

- Свободно распространяемые операционные системы;
- Операционная система специального назначения «Astra Linux Special Edition».

Для обеспечения функционирования Платформы и прикладных приложений, разработанных на её основе, при проведении эксперимента в рамках Государственного контракта от 24.12.2020 № 0173100007520000032_144316 на оказание услуг по передаче неисключительных прав и обеспечению функционирования, администрирования и бесперебойной работы программного обеспечения в целях проведения эксперимента по созданию, переводу и развитию государственных информационных систем и их компонентов на единой цифровой платформе Российской Федерации «ГосТех» Исполнитель предоставляет лицензии на коммерческое ПО третьих лиц. Количество лицензий и вид лицензирования выбраны на основе политики лицензирования от поставщика ПО.

Для обеспечения заданного уровня сервиса Исполнителем осуществляется закупка технической поддержки от поставщиков коммерческого ПО третьих лиц.

3 ОПИСАНИЕ ВЗАИМОСВЯЗИ АС С ДРУГИМИ СИСТЕМАМИ

3.1 Перечень систем, с которыми связана Платформа

Внешними по отношению к Платформе являются:

– система межведомственного электронного взаимодействия (СМЭВ), предназначенная для обмена данными, необходимыми для оказания государственных услуг гражданам и организациям, в электронном виде, взаимодействие с которой осуществляется через компонент «Интеграция со СМЭВ Platform V SMEV Gateway» по протоколам SOAP и FTP;

– единая система идентификации и аутентификации (ЕСИА), предназначенная для обеспечения санкционированного доступа участников информационного взаимодействия к информации, содержащейся в государственных информационных системах и иных информационных системах, взаимодействие с которой осуществляется через компоненты «Аутентификация Platform V IAM («Модули IAM. Технологическая платформа Сбербанка»)» и «Интеграция с ЕСИА Platform V ESIA Gateway» по протоколам OpenID Connect и RESTful;

– система Firebase Cloud Messaging, предназначенная для доставки Push-уведомлений до Android и iOS устройств, взаимодействие с которой осуществляется через компонент «Центр уведомлений Platform V Notification Center» по протоколу RESTful;

– сервис рассылки электронной почты, предназначенный для доставки электронной почты, взаимодействие с которым осуществляется через компонент «Центр уведомлений Platform V Notification Center» по протоколу SMTP.

3.2 Описание связей между системами

Взаимодействие с внешними системами осуществляется:

– взаимодействие между прикладными приложениями, разработанными на базе Платформы и системой межведомственного электронного взаимодействия (СМЭВ) осуществляется через компонент «Интеграция со СМЭВ Platform V SMEV Gateway» по протоколам SOAP и FTP в XML и файловом форматах;

– взаимодействие между прикладными приложениями, разработанными на базе Платформы и единой системой идентификации и аутентификации (ЕСИА) осуществляется через компонент «Аутентификация Platform V IAM («Модули IAM. Технологическая платформа Сбербанка»)» по протоколу OpenID Connect в формате JSON;

– взаимодействие между прикладными приложениями, разработанными на базе Платформы и единой системой идентификации и аутентификации (ЕСИА) осуществляется через компонент «Интеграция с ЕСИА Platform V ESIA Gateway» по протоколу RESTful в формате JSON;

– взаимодействие между прикладными приложениями, разработанными на базе Платформы и системой Firebase Cloud Messaging осуществляется через компонент «Центр уведомлений Platform V Notification Center» по протоколу RESTful в формате JSON;

– взаимодействие между прикладными приложениями, разработанными на базе Платформы и сервисом рассылки электронной почты осуществляется через компонент «Центр уведомлений Platform V Notification Center» по протоколу SMTP в формате почтового сообщения (Internet Message Format).

3.3 Описание регламента связи

Взаимодействие прикладных приложений, разработанных на базе Платформы, с другими системами производится в режимах «запрос/ответ» и «передача/прием» в согласованных между системами форматах взаимодействия по мере необходимости или по инициативе одной из сторон взаимодействия согласно установленным нормативно-правовыми актами срокам.

Обмен данными между прикладными приложениями, разработанными на базе Платформы, и системой межведомственного электронного взаимодействия (СМЭВ) осуществляется как в режиме реального времени, так и по расписанию по инициативе прикладного приложения в соответствии с внутренними бизнес-процессами.

Обмен данными между прикладными приложениями, разработанными на базе Платформы, и единой системой идентификации и аутентификации (ЕСИА)

осуществляется в режиме реального времени по инициативе прикладного приложения в соответствии с внутренними бизнес-процессами.

Обмен данными между прикладными приложениями, разработанными на базе Платформы, и системой Firebase Cloud Messaging осуществляется как в режиме реального времени, так и по расписанию по инициативе прикладного приложения в соответствии с внутренними бизнес-процессами.

Обмен данными между прикладными приложениями, разработанными на базе Платформы, и сервисом рассылки электронной почты осуществляется как в режиме реального времени, так и по расписанию по инициативе прикладного приложения в соответствии с внутренними бизнес-процессами.

3.4 Описание взаимосвязей Платформы с подразделениями объекта автоматизации

Порядок взаимодействия Платформы с подразделениями объекта автоматизации согласуется с пользователями Платформы в индивидуальном порядке.

4 ОПИСАНИЕ ПОДСИСТЕМ

4.1 Описание функционирования подсистем и их частей

4.1.1 Облачная платформа Сбербанка

4.1.1.1 Компонент «Облачная платформа Сбербанка. Подсистема виртуализации вычислительных ресурсов»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Облачная платформа Сбербанка. Подсистема виртуализации вычислительных ресурсов».

Подсистема виртуализации вычислительных ресурсов обеспечивает реализацию следующих требований:

- поддержка виртуализации серверов на аппаратной платформе Intel 64 или AMD64 или amd64 или EM64T или x86_64 или x86-64 или x64 или аналогов;
- автоматическое восстановление работы виртуальных машин в случае выхода из строя одного или нескольких серверов виртуализации (гипервизоров);
- поддержка одновременной работы нескольких кластеров (кластеров виртуализации, групп гипервизоров);
- возможность создания кластеров, состоящих не менее чем из 32 узлов;
- выравнивание инструкций процессора на всех серверах виртуализации кластера под минимально совместимый набор инструкций;
- поддержка добавления серверов виртуализации в кластер без остановки и перебоев в работе подсистемы виртуализации;
- возможность перевода серверов виртуализации в режим обслуживания (запрет на размещение и перемещение на них виртуальных машин).

Подсистема виртуализации вычислительных ресурсов обеспечивает следующий функционал в части автоматического управления размещением ресурсов между кластерами:

- поддержка встроенных механизмов, дающих возможность выделить виртуальным машинам больше ресурсов, чем есть на физических вычислительных узлах на основе данных об использовании ресурсов серверов кластера;
- распределение виртуальных машин между несколькими кластерами;
- поддержка шлюза API, маскирующего внутреннее устройство подсистемы виртуализации вычислительных ресурсов.

Подсистема виртуализации вычислительных реализует следующие функции:

- поддержка «горячего» перемещения виртуальных машин с одного сервера виртуализации (гипервизора) на другой без прерывания их работы;
- поддержка «холодного» перемещения виртуальных машин с одного сервера виртуализации (гипервизора) на другой;
- управление перемещением и размещением данных, обрабатываемых с использованием виртуальных машин (дисковых томов) и файлов-образов виртуальных машин между системами хранения данных;
- обработка отказов перемещения виртуальных машин и обрабатываемых на них данных;
- доступ к консоли виртуальной машины из системы управления платформой виртуализации и базовое управление виртуальной машиной через данную консоль;
- возможность работы вложенной виртуализации при поддержке гостевой операционной системой виртуальной машины;
- создание виртуальных машин из шаблонов библиотеки шаблонов и образов;
- автоматическое развертывание операционных систем с требуемой конфигурацией и программного обеспечения на виртуальных машинах;
- управление жизненным циклом виртуальных машин (создание, размещение, удаление);
- удаление виртуальных машин;
- поддержка добавления и изъятия вычислительных ресурсов (процессоры, память);
- поддержка динамического добавления блочных устройств (дисковых томов) без остановки виртуальной машины;

- возможность автоматической регистрации созданной виртуальной машины в подсистемах мониторинга, сбора статистики и журналирования;
- получение списка поддерживаемых продуктов от подсистемы оркестрации;
- получение запросов на доступные к созданию шаблоны и образы виртуальных машин через пользовательский интерфейс Подсистемы управления;
- получение запросов на создание, редактирование, удаление объектов через пользовательский интерфейс Подсистемы управления;
- работа виртуальных машин под управлением свободно распространяемых операционных систем или операционной системы специального назначения «Astra Linux Special Edition»;
- отправка запросов на создание, редактирование, удаление объектов на платформе виртуализации через подсистему оркестрации;
- полный запрет перемещения виртуальных машин;
- ограничение перемещения виртуальных машин в пределах зоны безопасности;
- ограничение перемещения виртуальных машин между зонами безопасности;
- ограничение по перемещению виртуальных машин и обрабатываемых на них данных (дисковых томов) только на заданные серверы виртуализации (гипервизоры) или системы хранения данных.

4.1.1.2 Компонент «Облачная платформа Сбербанка. Подсистема вычислительной сети»

Соответствие Платформы предъявляемым требованиям обеспечено компонентом «Облачная платформа Сбербанка. Подсистема вычислительной сети».

Подсистема вычислительной сети обеспечивает реализацию следующих функций:

- подключение виртуальных машин к локальной сети виртуальной инфраструктуры на скорости не менее 1 Гбит/сек;
- доступность виртуальных серверов по сети для пользователей;
- трансляция сетевых адресов;

- поддержка логических маршрутизаторов, позволяющих организовать маршрутизацию пакетов без привязки к нижележащей топологии сети и физическому расположению;

- изоляция потоков данных, передаваемых и обрабатываемых серверами виртуализации и сетевых потоков виртуальной вычислительной сети;

- высокая доступность (кластеризация) контроллеров;

- поддержка Jumbo-кадров (кадр сети Ethernet, в котором, можно передать данные) не менее чем 9000 байт;

- сетевая подсистема поддерживает программный интерфейс API (REST) для мониторинга и конфигурирования сетевого взаимодействия виртуальных машин облачной платформы.

4.1.1.3 Компонент «Облачная платформа Сбербанка. Подсистема хранения данных»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Облачная платформа Сбербанка. Подсистема хранения данных».

Подсистема хранения данных построена в соответствии с архитектурой программно-определяемой системы хранения (SDS) и обеспечивает реализацию следующих требований:

- функционал одновременного расширения пула хранения, пропускной способности и вычислительных мощностей системы хранения данных путем добавления дополнительных серверов SDS;

- поддержка выделения ресурсов по мере необходимости для дисковых томов виртуальных машин;

- поддержка ограничения и приоритезации операций ввода-вывода индивидуально для каждого диска виртуальной машины (QoS);

- создание моментальных копий виртуальных дисков виртуальных машин;

- поддержка кэширования операций ввода-вывода на чтение и запись;

- поддержка кластеризации контроллера SDS;

- наличие открытого механизма (API) для подключения Системы хранения данных (далее -СХД) сторонних производителей;

- подсистема хранения данных поддерживает программный интерфейс API (REST) для использования в облачной платформе.

4.1.1.4 Компонент «Облачная платформа Сбербанка. Подсистема мониторинга»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Облачная платформа Сбербанка. Подсистема мониторинга».

Подсистема мониторинга обеспечивет реализацию следующих требований:

- поддержка следующих базовых метрик производительности:

- CPU Utilization;
- RAM Utilization;
- Storage Utilization;
- Network Utilization;

- отслеживание метрик производительности и доступности серверов виртуализации (потребление CPU, RAM, IOPS, Throughput, Latency);

- поддержку сбора базовых метрик подсистемы хранения данных (IOPS, Throughput, Latency, информация по объемам пространства).

Подсистема мониторинга обеспечивает выполнение следующих требований к конфигурированию и сбору метрик мониторинга:

- возможность расширения базового набора метрик путем изменения правил мониторинга, без необходимости установки дополнительных модулей на целевые серверы;

- дополнительные модули-агенты мониторинга должны управляться централизованно;

- изменение правил мониторинга должно поддерживать распространение сразу на все или на определенное количество агентов, в зависимости от выбора администратора;

- возможность получения исторических значений метрик за выбранный период.

Подсистема мониторинга поддерживает программный интерфейс API (REST) для облачной платформы.

4.1.1.5 Компонент «Облачная платформа Сбербанка. Подсистема управления»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Облачная платформа Сбербанка. Подсистема управления».

Подсистема управления предоставляет разный уровень доступа для администраторов инфраструктуры и администраторов ВЦОД.

Для администраторов инфраструктуры предоставлена возможность полного доступа к управлению инфраструктурой Платформы.

Для администраторов ВЦОД подсистема управления обеспечивает следующие функции:

- управление виртуальными машинами:
 - отображение, добавление и удаление виртуальных машин, изменение отображаемого названия;
 - отображение сетевых адресов виртуальной машины;
 - поддержка иерархической ресурсной структуры;
 - формирование заказов виртуальных машин;
 - заказ нескольких виртуальных машин одностипной конфигурации;
 - просмотр профиля виртуальной машины с отображением количества запрошенных ресурсов;
 - отображение процесса загрузки виртуальной машины, работы с интерфейсом виртуальной машины в режиме чтения и записи, выполнения операций запуска, перезагрузки и выключения виртуальной машины;
 - обновление статусов работы виртуальной машины в реальном времени без необходимости обновления страниц через пользовательский интерфейс Подсистемы управления;
 - управление жизненным циклом ключевых пар (SSH key pair) доступа к виртуальным машинам;

- изменение конфигурации (характеристик, количества потребляемых ресурсов) виртуальной машины;
- подключение дисковых томов виртуальных машин без прерывания работы;
- маркировка виртуальных машин метками;
- создание мгновенного снимка (snapshot) виртуальной машины без прерывания ее работы, возврат виртуальной машины к моменту создания мгновенного снимка;
- журналирование действий с виртуальной машиной, осуществляемых через пользовательский интерфейс подсистемы управления;
- управления вычислительной сетью:
 - управление жизненным циклом виртуальных частных сетей;
 - управление жизненным циклом внешних сетевых адресов для виртуальных машин через пользовательский интерфейс Подсистемы управления.

Пользовательский интерфейс подсистемы управления поддерживает:

- протокол HTTPS;
- автоматическое отключение сессии пользователя после определяемого администратором времени бездействия;
- актуальные версии браузеров, основанных на технологии WebKit (Google Chrome, Яндекс.Браузер).

Пользовательский интерфейс подсистемы управления предоставляет пользователям справочную информацию (скриншоты интерфейса, видео) с описаниями функционала пользовательского интерфейса для следующих действий пользователя:

- первая авторизация;
- выделение виртуальных машин;
- управление виртуальными машинами.

4.1.1.6 Компонент «Облачная платформа Сбербанка. Библиотека шаблонов и образов»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Облачная платформа Сбербанка. Подсистема библиотеки шаблонов и образов».

Библиотека шаблонов и образов реализовывает следующие требования:

- поддержка запуска множества виртуальных машин на базе одного образа (copy-on-write);
- возможность хранения мгновенных снимков, шаблонов, клонов виртуальных машин;
- возможность хранения библиотеки образов, используя ресурсы подсистемы хранения данных;
- возможность настройки видимости образов для организаций и групп;
- поддержка форматов образов raw, qcow2.

Библиотека шаблонов и образов поддерживает программный интерфейс API (REST) для платформы виртуализации.

4.1.1.7 Компонент «Облачная платформа Сбербанка. Подсистема оркестрации»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Облачная платформа Сбербанка. Подсистема оркестрации».

Подсистема оркестрации обладает следующей функциональностью:

- возможность кластеризации оркестратора;
- возможность автоматизированного управления жизненным циклом инфраструктуры и приложений в информационной системе;
- возможность запуска множества комплексных облачных приложений на основе шаблонов;
- возможность интеграции с системами управления конфигурациями.

Подсистема оркестрации поддерживает программный интерфейс API (REST) для платформы виртуализации.

4.1.1.8 Компонент «Облачная платформа Сбербанка. Подсистема резервного копирования»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Облачная платформа Сбербанка. Подсистема резервного копирования».

Подсистема резервного копирования обладает следующей функциональностью:

- определение мест хранения резервных копий виртуальных машин и данных, обрабатываемых в облачной платформе;
- возможность резервного копирования данных виртуальных машин, размещенных на серверах виртуализации;
- управление подсистемой резервного копирования с помощью API.

4.1.2 Инструменты управления контейнерами

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью свободно распространяемым программным обеспечением (Open Source Software).

Инструмент управления контейнерами обеспечивает автоматизацию и управление жизненным циклом контейнеров и сервисов. В жизненный цикл контейнера входит планирование ресурсов, управление масштабируемостью, правила балансировки нагрузки и контроль доступности, а также организация виртуальных сетей. Программное обеспечение обладает следующей функциональностью:

- наличие инструментов для создания, развертывания, запуска и управления приложениями в контейнерной среде;
- поддержку процессов автоматического развертывания docker контейнеров с собранными внутри приложениями (как платформенными, так и прикладными);
- автоматическое масштабирование приложений путем мониторинга утилизации ресурсов приложениями и автоматический запуск дополнительных экземпляров приложений в случае превышения пороговых значений;

– управление кластером, развертываниями, подами (pods) и другими ресурсами через графический интерфейс, командную строку или Rest API;

– встроенные средства планирования ресурсов кластера и балансировки нагрузки.

4.1.3 Инструменты управления производственным процессом

«SberWorks»

4.1.3.1 Компонент «Инструмент управления планированием»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Инструмент управления планированием».

Инструмент управления планированием обеспечивает следующую функциональность:

- создание (постановка) целей исполнителям, включая следующие атрибуты:
 - наименование цели;
 - описание;
 - постановщик;
 - исполнитель;
 - плановый срок исполнения;
 - приоритет;
 - статус;
 - изменение атрибутов, отмена или удаление созданных целей;
 - управление жизненным циклом в части изменения статусов цели;
 - связь целей с объектами в инструментах управления требованиями и релизами
- задачами бэклога и релизами, направленными на достижение целей;
- отслеживание текущего статуса целей и связанных с ними задач в бэклоге;
 - многопользовательский режим работы в части одновременной работы нескольких пользователей с одной целью.

4.1.3.2 Компонент «Инструмент управления требованиями»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Инструмент управления требованиями».

Инструмент управления требованиями обладает следующей функциональностью:

- ведение требований по принципам гибкой методологии в формате бэклога упорядоченных по приоритетам перечня требований к продукту, известных на текущий момент времени, с возможностью инкрементного уточнения или приоритизации по мере получения обратной связи от потребителей продукта;
- ведение бэклогов требований в составе следующих элементов:
 - фича - группировка изменений (историй) в законченную бизнес-функциональность;
 - история - атомарные изменения функциональности;
- возможность включения в бэклог объектов "дефект" инструмента управления дефектами;
- управление этапами жизненного цикла требований через изменение статусов элементов бэклога;
- ведение версионности описания функциональных и нефункциональных требований, с возможностью просмотра истории изменений;
- возможность добавления в текстовые описания изображений, графиков, различных видов диаграмм, вложений офисного формата;
- многопользовательский режим работы в части одновременной работы нескольких пользователей с одним требованием.

4.1.3.3 Компонент «Инструмент управления релизами»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Инструмент управления релизами».

Инструмент управления релизами обеспечивает следующую функциональность:

- учет и ведение релизов, в части:

- создания релиза;
- редактирования релиза;
- отмены релиза;
- закрытия релиза;
- формирование состава релиза:
 - включение объектов в состав релиза:
 - объектов «история» инструмента управления требованиями;
 - объектов «дефект» инструмента управления дефектами;
 - исключение объектов из состава релиза;
- управление этапами жизненного цикла релиза, в части изменения статусов релиза;
- учет прохождения приемо-сдаточных испытаний и внедрения релиза, в части:
 - заполнения атрибутов для начала приемо-сдаточных испытаний;
 - приглашения заинтересованных лиц для участия в приемо-сдаточных испытаниях: автоматически и/или вручную;
 - принятия решений об участии в приемо-сдаточных испытаниях;
 - фиксации результатов проведения приемо-сдаточных испытаний и внедрения релиза;
 - формирования, согласования и утверждения протокола приемо-сдаточных испытаний;
 - формирование, согласование, утверждение распоряжения о внедрении;
- многопользовательский режим работы, в части одновременной работы нескольких пользователей с одним релизом.

4.1.3.4 Компонент «Инструмент управления дефектами»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Инструмент управления дефектами».

Инструмент управления дефектами обеспечивает следующую функциональность:

- управление дефектами, в части:

- заведения дефектов;
- изменения дефектов;
- закрытия дефектов;
- возможность построения отчетности по дефектам в разрезе:
 - количества дефектов в зависимости от приоритета;
 - количества дефектов в зависимости от этапа обнаружения;
- связь с объектами управления требованиями, управления тестированием, управления релизами путем указания идентификатора связанного объекта, зарегистрированного в соответствующем инструменте;
- возможность настройки атрибутивного состава дефектов:
 - возможность добавления новых пользовательских полей дефекта;
 - возможность удаления пользовательских полей дефекта;
- возможность связи дефектов с изменениями в коде путем указания идентификатора запроса на слияния веток инструмента версионного контроля исходного кода и конфигураций;
- многопользовательский режим работы в части одновременной работы нескольких пользователей с одним дефектом.

4.1.3.5 Компонент «Инструмент управления тестированием»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Инструмент управления тестированием».

Инструмент управления тестированием обеспечивает следующую функциональность:

- ведение планов тестирования и сценариев тестов:
 - создание и изменение планов тестирования;
 - создание и изменение сценариев тестирования;
- фиксирование дефектов тестирования в инструменте управления дефектами;
- обеспечение связи с объектами управления дефектами, управления требованиями, управления релизами путем указания идентификатора связанного объекта, зарегистрированного в соответствующем инструменте;

- настраиваемый атрибутный состав тестовых сценариев и плана тестирования:
 - добавление новых полей тестовых сценариев;
 - добавление новых полей плана тестирования;
 - удаление добавленных полей плана тестирования;
 - удаление добавленных полей тестового сценария;
- сохранение результатов выполнения ручных тестов;
- многопользовательский режим работы в части одновременной работы нескольких пользователей с одним тестовым сценарием.

4.1.3.6 Компонент «Инструмент управления репозиториями дистрибутивов»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Инструмент управления репозиториями дистрибутивов».

Инструмент обладает следующей функциональностью:

- получение, хранение и предоставление необходимых библиотек для сборки ПО из репозитория библиотек, расположенных в сети интернет;
- хранение и предоставление библиотек ПО собственной разработки, в том числе посредством API, предоставляемого для инструмента управления сборкой ПО;
- хранение и предоставление дистрибутивов промежуточных сборок ПО, в том числе посредством API, предоставляемого для инструментов управления сборкой ПО;
- хранение и предоставление дистрибутивов релизных сборок ПО, в том числе посредством API, предоставляемого для инструментов управления сборкой или развертыванием ПО;
- хранение и предоставление пакетов установок ПО, в том числе посредством API, предоставляемого для инструментов управления сборкой или развертыванием ПО;
- хранение и предоставление базовых, промежуточных и релизных образов контейнеров ПО;
- многопользовательский режим работы с репозиторием дистрибутивов.

4.1.3.7 Компонент «Инструмент версионного контроля исходного кода и конфигураций»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Инструмент версионного контроля исходного кода и конфигураций».

Инструмент версионного контроля исходного кода и конфигураций предоставляет следующие функции:

- создание и ведение репозиториев исходного кода, конфигураций сред и конвейера DevOps;
- версионный контроль исходного кода и конфигураций с возможностью восстановления при использовании утилиты git;
- управление правилами сохранения исходного кода для каждой команды за счет запрета на удаление веток;
- создание запроса на внесение изменений в исходный код и веб-интерфейс для обсуждения предлагаемых изменений до их включения в официальный репозиторий проекта;
- управление запросами на совместную верификацию и согласование изменений программного кода за счет использования запросов на слияние веток;
- настройка правил и требований к запросам на изменение и слияние веток в репозитории (Pull Request):
 - минимальное количество успешных сборок;
 - минимальное количество согласующих запрос на слияние;
- аудит изменений в репозитории исходного кода;
- указание связей запросов на слияние веток с объектами "история" и "дефект" за счет указания идентификатора объекта в инструментах управления требованиями и управления дефектами соответственно;
- разграничение прав доступа и аудит изменений;
- связь с объектами управления требованиями, управления дефектами;

– многопользовательский режим работы с репозиторием исходного кода или конфигураций.

4.1.3.8 Компонент «Инструмент анализа качества кода»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Инструмент анализа качества кода».

Инструмент обеспечивает следующую функциональность:

- непрерывный анализ и измерение качества кода, поддержку разработчиков при обнаружении недостатков созданного или повторно использованного кода за счет:
 - возможности интеграции с инструментом управления сборкой ПО для встраивания анализа качества кода в конвейер сборки дистрибутива;
 - наличия встроенных правил и стандартов кодирования, обеспечивающих надежность и безопасность кода, правил оформления и документирования исходного кода, как минимум на языке Java;
- выявление показателей и предоставление отчетности по метрикам кода:
 - степень отклонений от стандартов кодирования, обеспечивающих надежность и безопасность кода;
 - отклонения от стандартов оформления и документирования кода;
- выявление показателей и предоставление отчетности по метрикам кода:
 - степень покрытия кода модульными тестами;
 - наличие и уровень дефектов, выявленных тестами;
- выявление показателей и предоставление отчетности по метрике дублирования кода;
- выявления показателей и предоставление отчетности по метрике поддерживаемости кода и объему технического долга (показатель исправления ранее выявленных дефектов);
- выявление показателей и предоставление отчетности по метрике структурного качества и сложности программного кода (показатель количества возможных путей исполнения кода);

- установка контрольных точек просмотра мест возникновения выявленных дефектов в исходном коде, предоставление описания причины возникновения дефекта;
- настройка допустимого уровня дефектов по метрикам покрытия модульными тестами и критичности дефектов, выявленных модульными тестами, степени дублирования кода, размеру технического долга;
- многопользовательский режим работы с проектом.

4.1.3.9 Компонент «Инструмент проектирования API и архитектурного контроля «МЕТА»

Соответствие Платформы предъявляемым требованиям к инструменту проектирования API и архитектурного контроля обеспечено функциональностью компонента «Инструмент проектирования API и архитектурного контроля «МЕТА».

Инструмент обеспечивает следующую функциональность:

- проектирование API приложений в концепции API First и их публикации в едином реестре:
 - предоставление инструмента проектирования API-спецификаций в формате OpenAPI Specification 3.0 (OAS 3.0);
 - поддержка шагов создания, согласования и публикации спецификаций OAS 3.0 в едином реестре;
 - предоставление средств контроля интеграционных взаимодействий на этапе проектирования за счет:
 - инструмента ведения плановых межкомпонентных взаимодействий (связей);
 - инструмента по разметке архитектурным статусом межкомпонентных связей;
 - инструмента обратного проектирования, выявляющего фактические связи из кода приложений-потребителей API, реализованных в виде интерфейсов Java;

- инструмента конфигурирования списков контроля доступа (ACL) к API с последующей их передачей в конвейер DevOps для применения изменений правил доступа;
- предоставление информации о наборе API, входящих в состав Платформы;
- ведение единого репозитория информации о наборе API, входящих в состав компонент и бизнес-областей, предоставление информации о наборе API, входящих в состав Платформы за счет:
 - структуры хранения API в привязке к определенному компоненту;
 - структуры хранения компонент с учетом их принадлежности к определенной функциональной области;
 - функционала маркировки компонент признаком принадлежности к платформе;
- управление этапом архитектурного контроля API;
- контроль качества API-спецификаций, размещенных в едином реестре, на предмет соответствия архитектурным требованиям и управление этапом архитектурного контроля за счет:
 - функционала обратной связи о качестве API со стороны команд разработки;
 - функционала автоматической мультикритериальной оценки качества спецификаций API, выявленных в Java-коде, методами обратного проектирования;
 - функционала разметки API архитектурным статусом;
 - функционала массовой разметки API архитектурным статусом;
 - функционала формирования отчета по методам API в разрезе их архитектурных статусов;
- поддержку версионирования API за счет:
 - загрузки и хранения API в разрезе версий, согласно правилам семантического версионирования semver;
 - отображения версий в агрегации по MAJOR.MINOR-номерам semver;

- возможность выбора и просмотра определенной загруженной версии спецификации API;
- управление нефункциональными требованиями к API с помощью:
 - инструмента описания нефункциональных требований (НФТ) в разрезе каждого метода API с возможностью наследования НФТ от API и компонента;
- поддержка принципа обратного проектирования (формирование метаданных API по исходному коду объекта):
 - предоставление инструмента обратного проектирования (plugin) для включения в жизненный цикл сборщиков maven, gradle, обеспечивающего выявление и последующую публикацию в едином реестре спецификаций API, определенных:
 - Java-интерфейсами, размеченными аннотацией API- библиотек;
 - Java-интерфейсами, размеченными аннотациями jsonrpc4j;
 - OpenAPI спецификациями (OAS 3.0) в формате yaml, включенных в сборки кода;
 - генерация и ведение реестра интеграционных заглушек API, для ускорения разработки приложений и повышения качества кода за счет раннего тестирования интеграционных взаимодействий за счет:
 - функционала создания проектной области для работы с эмуляциями;
 - функционала создания эмуляции работы REST API, с возможностью задания описаний значений для заголовков, cookies, кодов ответа, метода и содержимого ответа;
 - инструмента создания эмуляций с сохранением состояния между обращениями к ним;
 - возможности просмотра истории вызовов эмуляций;
 - функционала работы с эмуляциями (публикация и потребление) в едином публичном реестре.

4.1.3.10 Инструменты управления поставкой ПО

4.1.3.10.1 Компонент «Инструмент управления сборкой ПО»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Инструмент управления сборкой ПО».

Инструмент обеспечивает следующую функциональность:

- организация непрерывного процесса сборки и проверки качества исходного кода ПО за счет:
 - непрерывного выполнения заданий сборки и проверки качества по событиям изменения кода в инструменте версионного контроля исходного кода и конфигураций;
 - настройка сборки промежуточных и релизных дистрибутивов из исходного кода в инструменте версионного контроля исходного кода и конфигураций;
 - настройка проверки качества исходного кода в инструменте анализа качества кода;
 - настройка проверки корректности сборки дистрибутива приложения за счет выполнения smoke-тестов;
 - настройка заданий по выполнению модульных тестов;
 - выполнение заданий по сборке и проверке качества исходного кода и дистрибутива;
 - сохранение промежуточных и релизных дистрибутивов в инструменте управления репозиториями дистрибутивов.

4.1.3.10.2 Компонент «Инструмент управления поставками дистрибутивов»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Инструмент управления поставками дистрибутивов».

Инструмент имеет следующую функциональность:

– управление потоками развертывания и тестирования, автоматизацию процессов поставки и контроль критериев качества дистрибутивов за счет:

- настройки состава этапов конвейера;
- настройки последовательности действий внутри этапа по сборке, развертыванию, настройке и проверке ПО;
- указания условий перехода между этапами и действиями конвейера на основе критериев качества ПО;
- контроль и управление конвейером поставок в едином окне:
 - просмотр состояния и состава дистрибутива;
 - запуск развертывания дистрибутива;
 - перезапуск заданий, отработавших с ошибкой;
 - настройка автоматического запуска заданий;
- настройка нескольких конвейеров, с индивидуальными параметрами заданий в зависимости от версии дистрибутива;
- просмотр истории дистрибутива или релиза:
 - статусы прохождения этапов конвейера;
 - длительность каждого шага от момента появления дистрибутива до внедрения в промышленную эксплуатацию;
- централизованная настройка контроля качества на основе результатов проверок в инструментах управления сборкой и развертыванием ПО:
 - добавление в конвейер этапов контроля качества дистрибутива;
 - настройка логики этапов конвейера в зависимости от результатов контроля качества на предыдущих шагах;
 - отправка оповещений о прохождении этапов контроля качества.

4.1.3.10.3 Компонент «Инструмент управления развертыванием ПО»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Инструмент управления развертыванием ПО».

Инструмент управления развертыванием ПО обеспечивает следующую функциональность:

- организация непрерывного процесса развертывания и автоматического тестирования приложений за счет:

- непрерывного выполнения заданий по развертыванию и тестированию приложений по событию сборки релизной версии дистрибутива;

- настройка заданий по развертыванию, конфигурированию и тестированию приложений:

- настройка развертывания приложений из дистрибутивов и конфигурирования параметров исполнения приложений с использованием скриптов в инструменте версионного контроля исходного кода и конфигураций;

- настройка заданий автоматического тестирования приложений с использованием скриптов и кода автотестов в инструменте версионного контроля исходного кода и конфигураций;

- выполнение заданий по конфигурированию параметров исполнения приложений, развертыванию приложений из дистрибутивов и автоматическому тестированию приложений.

4.1.3.11 Компонент «Инструмент аналитики и мониторинга производственного процесса»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Инструмент аналитики и мониторинга производственного процесса».

Инструмент обеспечивает следующую функциональность:

- сбор, обработку и хранение данных, обеспечивающих производственный процесс:

- функциональная структура;

- справочник информационных систем;

- информацию о запусках заданий, выполняемых инструментами DevOps;

- release notes в привязке к дистрибутивам;

- pull-requests;

- дефекты;
- релизы;
- распоряжения о внедрении;
- требования (фичи, истории, задачи, CR, дефекты);
- предоставление отчетности по соблюдению обязательных требований к процессу разработки приложений по заданным метрикам:
 - M1. Доля команд, которые ведут бэклог в системе управления требованиями, состоящий из фич или историй;
 - M4. Доля фич со стандартной статусной моделью. Статусная модель считается стандартной, если в ее названии указано "APProcess";
 - M29. Доля фич, связанных с описанием требований. Фича считается связанной с описанием требований, если у нее есть ссылка на страницу в системе описания требований;
 - M28. Доля декомпозированных фич. Фича считается декомпозированной, если она связана с историями;
 - M 6. Доля историй, связанных с релизом;
 - M 9. Доля Change Request со стандартной статусной моделью. Статусная модель считается стандартной, если в ее названии указано "APProcess";
 - M 30. Доля Change Request, связанных с автоматизированной системой;
 - M 10. Доля Change Request, связанных с элементом бэклога смежной команды. Метрика показывает долю CR в статусах "To Do" и "In Progress", связанных с фичей, историей или задачей;
 - M 11. Доля Change Request, содержащих связь с родительским объектом. В качестве родительского объекта для CR должна выступать фича или история;
 - M14. Доля дефектов ПРОМ, связанных с объектом "Релиз";
- предоставление отчетности по производственному процессу разработки приложений по заданным метрикам:

- Deployment Frequency - частота успешных установок дистрибутивов в ПРОМ;
- Change Failure Rate - доля попыток установок дистрибутивов в ПРОМ, требующих исправления. Установка требует исправления в случае, если джоб выполнен неуспешно и/или указан неуспешный результат внедрения в распоряжении о внедрении;
- Lead Time - длительность задачи с момента начала работы над задачей до момента успешной установки дистрибутива в ПРОМ. Дата установки в ПРОМ определяется следующим образом - для каждого дистрибутива, в составе релизов которых задача выходила в ПРОМ, определяется самая ранняя дата установки. Далее из полученных дат выбирается самая поздняя. Дата начала работы над задачей определяется следующим образом: для задачи, связанных с ней подзадач/задач, а также связанных с ней pull requests выбирается самая ранняя дата перевода в категорию статуса "В работе" или дата вливания PR в мастер ветку;
- многопользовательский режим работы:
 - в функциональной оргструктуре один пользователь может быть привязан только к одному юр. лицу;
 - в ролевой модели для отчетности один пользователь может быть привязан к нескольким юр. лицам; пользователь может просматривать данные только в рамках тех юр. лиц, к которым он привязан к ролевой модели;
 - в функциональной оргструктуре один пользователь может быть привязан к нескольким проектам, но в рамках одного юр. лица;
 - дистрибутив связывается со всеми командами, с которыми связаны задачи, вошедшие в состав релиза по этому дистрибутиву;
 - связь дистрибутива с юр. лицом осуществляется через маппинг на информационную систему; при этом у одной информационной системы может быть указано только одно ответственное юр. лицо.

4.1.4 Компоненты управления данными

4.1.4.1 Компонент «СУБД Platform V Pangolin»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «СУБД Platform V Pangolin».

Компонент «СУБД Platform V Pangolin» обеспечивает управление высоко-доступными, отказоустойчивыми объектно-реляционными базами данных в приложениях любого класса критичности, включая:

– поддержку транзакций с использованием версионности данных (MVCC), включающую:

- неблокирующее выполнение чтений записей таблицы в параллельно выполняемых транзакциях;
- неблокирующее выполнение вставки записей в таблице параллельно с чтением записей из той же таблицы;
- неблокирующее выполнение изменения записей в таблице параллельно с чтением записей из той же таблицы;
- неблокирующее выполнение удаления записей в таблице параллельно с чтением записей из той же таблицы;
- неблокирующее выполнение вставок записей таблицы в параллельно выполняемых транзакциях, при условии отсутствия на таблице уникальных или первичных ключей, либо при отсутствии параллельных вставок записей с одинаковыми значениями полей ключей;
- неблокирующее выполнение вставки записей в таблице параллельно с изменением записей в той же таблице, при условии отсутствия на таблице уникальных или первичных ключей, либо при отсутствии параллельных операций вставки и изменения записей с одинаковыми значениями полей ключей;
- неблокирующее выполнение вставки записей в таблице параллельно с удалением записей в той же таблице, при условии отсутствия на таблице уникальных или первичных ключей, либо при отсутствии параллельных

операций вставки и удаления записей с одинаковыми значениями полей ключей;

– неблокирующее выполнение изменения записей в таблице параллельно с изменением непересекающегося набора записей в той же таблице. При пересечении наборов изменяемых записей для операций изменения будет происходить блокировка до применения изменений блокирующей транзакции;

– неблокирующее выполнение удаление записей в таблице параллельно с изменением непересекающегося набора записей в той же таблице. При пересечении наборов удаляемых и изменяемых записей для операций удаления будет происходить блокировка до применения изменений блокирующей транзакции;

– неблокирующее выполнение удаление записей в таблице параллельно с удалением непересекающегося набора записей в той же таблице. При пересечении наборов удаляемых записей для операций удаления будет происходить блокировка до применения изменений блокирующей транзакции;

– изоляцию изменений данных в рамках транзакции уровня READ COMMITTED;

– атомарность применения всех изменений транзакции;

– ссылочную целостность данных, включающую:

– поддержку синтаксиса запросов создания внешнего ключа с поля(ей) таблицы на поле(я) другой таблицы;

– функциональность запрета создания или изменения записи в таблице с внешним ключом при отсутствии записи, соответствующей значениям в ссылающихся полях добавляемой или изменяемой записи, в таблице, на которую ссылается внешний ключ;

– функциональность каскадного удаления зависимых записей в ссылающейся таблице, при удалении соответствующих записей,

- адресуемых внешним ключом, в таблице, на которую ссылается внешний ключ;
- функциональность запрета удаления записей в таблице, на которую ссылается внешний ключ, при наличии зависимых записей в другой таблице, ссылающейся через внешний ключ на удаляемые записи;
 - функциональность каскадного изменения значений полей зависимых записей в ссылающейся таблице, при изменении соответствующих полей записей, адресуемых внешним ключом, в таблице, на которую ссылается внешний ключ;
 - функциональность запрета изменения значений полей записей, на которые ссылается внешний ключ, в таблице, при наличии записей в другой таблице, ссылающейся через внешний ключ на изменяемые поля записей;
- поддержку реализации пользовательских типов данных, включающую:
- поддержку синтаксиса запросов создания пользовательского типа данных, таких как:
 - существующих типов данных со связанными ограничениями (доменов);
 - типов данных - пустышек, резервирующих имя типа;
 - составных типов данных, представляющих собой структуру полей с заданными типами и, опционально, правилами сортировки;
 - типов перечислений, представляющих собой список именованных меток;
 - диапазонных типов данных, позволяющих оперировать значениями заданного подтипа как частью диапазонов значений с заданными правилами;
 - базовых скалярных типов;
 - возможность использования пользовательских типов как типов, используемых для:
 - определения типов полей таблиц, кроме типов - пустышек;
 - определения типов параметров функций, кроме типов - пустышек;

- оперирование значениями пользовательских типов при:
 - чтении из таблицы значения поля пользовательского типа, кроме типов - пустышек;
 - изменении в таблице значения поля пользовательского типа, кроме типов - пустышек;
 - индексации в таблице поля пользовательского типа, кроме типов - пустышек;
 - передачи в функцию значения параметра пользовательского типа, кроме типов - пустышек;
 - приема из функции значения параметра пользовательского типа, кроме типов - пустышек;
- защиту данных от привилегированных пользователей (администраторов баз данных), которая включает:
- защиту настроечных параметров СУБД компонента УРБД, позволяющих отключить или ослабить защитные меры, препятствующие доступу к данным со стороны привилегированных пользователей. К таким параметрам относятся параметры, регулирующие:
 - шифрование данных при передаче по сети, сложность и способы шифрования;
 - способ шифрования аутентификационной информации учетных записей пользователей, хранящейся в СУБД;
 - использование прозрачного шифрования данных;
 - состав и размещение подключаемых библиотек расширения функциональности СУБД;
 - состав внешних серверов аутентификации (ldap, radius), разрешенных к использованию для аутентификации пользователей;
 - правила аутентификации пользователей; за счет передачи управления и/или контроля (за счет требования по совместному изменению значений части параметров как администраторами БД, так и администраторами безопасности) для значений таких настроечных параметров,

администраторам безопасности, путем использования внешнего защищенного хранилища, управляемого администраторами безопасности, для управления значениями параметров; защищенное хранилище не должно входить в состав компонента УРБД и должно поставляться, устанавливаться, настраиваться и обслуживаться отдельно от компонента УРБД;

- реализацию защиты указанных выше настроечных параметров, с учетом:
 - предоставления возможности расширения состава поддерживаемых реализаций внешних защищенных хранилищ через реализацию плагинов интеграции;
 - использования единого защищенного хранилища, для прозрачного шифрования данных и защиты данных от привилегированных пользователей;
 - отсутствия встроенного в компонент УРБД API управления значениями параметров в защищенном хранилище и использования для такого управления собственного API и UI защищенного хранилища со стороны администраторов безопасности;
- утилиту инициализации параметров подключения к защищенному хранилищу администраторами безопасности при установке и настройке компонента УРБД;
- отключение слабых типов аутентификации компонента УРБД, позволяющих получить доступ от имени другого аккаунта без предъявления соответствующего секрета, либо небезопасные с точки зрения возможности утечки секрета;
- защиту данных объектов баз данных под управлением компонента УРБД, путем введения дополнительного механизма контроля доступа к объектам баз данных, управляемым администраторами безопасности и функционирующим параллельно со стандартной системой прав СУБД компонента УРБД, и защищающим доступ к:

- таблицам - на действия DML чтения, вставки, изменения и удаления, DDL - изменения и удаления;
- материализованным представлениям - на действия DML чтения, DDL - изменения и удаления;
- представлениям - на действия DML чтения, вставки, изменения и удаления, DDL - изменения и удаления, создания или изменения триггера по представлению, создания или изменения правила по представлению;
- функциям - на вызов, изменение, удаление;
- ролям - на действия удаления, изменения, выдачу роли-объекту, отзыв у роли-объекта, выдачу в качестве роли-объекта, отзыв в качестве роли-объекта, смену пароля, назначения текущей роли сессии;
- API управления:
 - защищаемыми объектами;
 - разрешенными для политик действиями над защищаемыми объектами;
 - составом политик;
 - назначенными пользователям политиками механизма защиты данных;
 - составом учетных записей администраторов безопасности;
- защиту и разрешение доступа только для администраторов безопасности, назначенных через соответствующие функции API, для:
 - функций API-управления механизмом защиты данных;
 - объектов, хранящих данные механизма защиты данных;
- утилиту инициализации механизма защиты данных администраторами безопасности при создании базы данных под контролем компонента УРБД;
- прозрачное шифрование данных, включая:
 - интеграцию с защищенным хранилищем, для обеспечения хранения мастер-ключей прозрачного шифрования данных, с учетом:
 - использования внешнего защищенного хранилища, управляемого администраторами безопасности; защищенное хранилище не входит в

состав компонента УРБД и должно поставляться, устанавливаться, настраиваться и обслуживаться отдельно от компонента УРБД;

– предоставление возможности расширения состава поддерживаемых реализаций внешних защищенных хранилищ через реализацию плагинов интеграции;

– использования единого, для прозрачного шифрования данных и защиты данных от привилегированных пользователей, защищенного хранилища;

– утилиту инициализации параметров подключения к системе хранения и управления ключами шифрования администраторами безопасности при установке и настройке компонента УРБД;

– API управления ключами шифрования прозрачного шифрования данных, включая функции:

– задания мастер-ключа;

– ротации мастер-ключа;

– перешифрования локальных ключей шифрования текущим мастер-ключом;

– восстановления локальных ключей шифрования при сбое смены мастер-ключа;

– API управления составом шифруемых объектов под управлением компонента УРБД через добавление новой опции для табличных пространств, указывающей необходимость шифрования данных объектов, хранящихся в помеченном такой опцией табличном пространстве;

– шифрование с применением алгоритмов AES-256 файлов данных объектов баз данных под управлением компонента УРБД, хранящихся в табличном пространстве, помеченном как шифруемое;

– шифрование с применением алгоритмов AES-256 файлов данных временных таблиц, размещаемых в шифруемом табличном пространстве;

– шифрование с применением алгоритмов AES-256 временных файлов, используемых для сортировки или слияния данных в операциях над данными компонента УРБД;

- шифрование с применением алгоритмов AES-256 файлов журнала операций (write ahead log);
- невозможность использования данных из зашифрованных файлов данных или зашифрованных файлов журнала операций при их переносе на узел компонента УРБД, относящийся к другому отказоустойчивому кластеру компонента УРБД, либо на узел с отключенным прозрачным шифрованием данных;
- шифрование с применением алгоритмов AES-256 данных, передаваемых между узлами отказоустойчивого кластера компонента УРБД при потоковой репликации;
- снятие резервной копии, содержащей зашифрованные объекты баз данных под управлением компонента УРБД;
- восстановление резервной копии, содержащей зашифрованные объекты баз данных под управлением компонента УРБД, на узле того же отказоустойчивого кластера, с которого была снята резервная копия. Узел отказоустойчивого кластера должен быть подключён к защищенному хранилищу, хранящему мастер-ключ, который использовался при создании резервной копии, содержащей зашифрованные объекты баз данных. А также быть настроенным на работу с включенным прозрачным шифрованием данных;
- невозможность восстановления резервной копии, содержащей зашифрованные объекты баз данных под управлением компонента УРБД, на узле другого отказоустойчивого кластера, либо на узле с отключенным прозрачным шифрованием данных;
- функции для получения следующих показателей, связанных с функциональностью:
 - признак активности прозрачного шифрования данных;
 - парольные политики пользователей, включая:
 - API управления парольными политиками, включающий функции:
 - вывод парольной политики, примененной для пользователя или роли;

- вывод детализированной информации по примененной парольной политике для пользователя или роли;
- создание или изменение парольной политики;
- активация парольной политики;
- деактивация парольной политики;
- вывод всех политик;
- разблокировка пользователя или роли;
- проверку пароля пользователя на соответствие эффективной парольной политике пользователя при изменении пароля пользователя, на:
 - повторение пароля, использованного ранее, в соответствии с настройками парольной политики;
 - минимальное время, которое должно пройти между изменением пароля;
 - соответствие критериям сложности пароля, заданным настройками парольной политики, включая – минимальную длину пароля; минимальное количество цифр в пароле; минимальное количество букв в пароле; минимальное количество символов, не являющихся буквой или цифрой, в пароле; минимальное количество строчных букв; максимальное количество повторяющихся символов; использование утилиты zxcvbn для проверки пароля; проверку сложности пароля пользовательской функцией; использование библиотеки cracklib при проверке пароля;
- блокировку пользователя при:
 - истечении времени действия пароля в соответствии с настройками парольной политики;
 - превышении попыток аутентификации с неверным паролем за интервал времени в соответствии с настройками парольной политики;
 - превышении интервала времени с момента последнего логина пользователя (неактивности) в соответствии с настройками парольной политики;
- разблокировку пользователя при:

- истечении времени блокировки для случая блокировки при превышении попыток ввода неверного пароля при аутентификации, в соответствии с настройками парольной политики;
- через вызов функции API;
- оповещение пользователей о приближении окончания времени действия пароля при аутентификации;
- функции для получения следующих показателей:
 - признак активности парольных политик пользователей;
- аудит действий пользователей, включающий:
 - настройку правил формирования записей аудита в виде:
 - конфигурационного файла экземпляра компонента УРБД;
 - установки параметров базы данных, сессии или роли через команды СУБД;
 - настройку правил формирования записей аудита только пользователя с правами администратора СУБД;
 - фиксацию в лог аудита действий пользователя в рамках конкретной сессии для запросов, успешно прошедших проверку прав пользователя, в соответствии с настройками:
 - по запросам чтения данных;
 - по запросам изменения данных;
 - по запросам вызова функций;
 - по запросам управления ролями и их привилегиями в рамках стандартной модели прав СУБД;
 - по запросам DDL (языка описания данных);
 - по служебным командам;
 - по командам установки параметров;
 - фиксацию в лог аудита действий пользователей над конкретным отношением (таблицей, представлением):
 - команда SELECT;

- команда INSERT;
- команда UPDATE;
- команда DELETE;
- фиксацию в лог аудита действий пользователей в виде записей, включающих информацию по (в зависимости от настроек и типа команды):
 - типу аудита;
 - идентификатору запроса;
 - классу команды;
 - команде;
 - типу объекта, адресованного командой;
 - имени объекта, адресованного командой;
 - тексту команды;
 - значения параметров команды;
- функции для получения следующих показателей:
 - признак использования библиотеки аудита;
- функционирование в режиме отказоустойчивого кластера, а именно:
 - предоставление двух экземпляров СУБД как узлов отказоустойчивого кластера, в том числе одного узла кластера как ведущего узла, принимающего запросы на изменение;
 - синхронную репликацию данных между ведущим и ведомым узлами кластера, обеспечивающую сохранность примененных изменений данных;
 - автоматическое переключение роли ведомого узла на роль ведущего узла при сбое функционирования ведущего узла;
 - автоматическое восстановление сервиса после одиночного сбоя функционирования отказоустойчивого кластера компонента УРБД при условии соблюдения рекомендаций по развертыванию и настройке кластера компонента УРБД;
 - API (REST и утилита командной строки) контроля и управления отказоустойчивым кластером, включающее следующие функции:

- определение ведущего и ведомого узла;
 - определение типа репликации для ведомого узла - синхронный или асинхронный;
 - проверка жизнеспособности СУБД компонента УРБД;
 - проверка жизнеспособности и готовности координатора кластера компонента УРБД;
 - получение информации по топологии кластера компонента УРБД;
 - получение истории переключения ролей узлов кластера компонента УРБД;
 - получение текущей конфигурации кластера компонента УРБД;
 - изменение конфигурации кластера компонента УРБД;
 - переключение ролей узлов кластера компонента УРБД;
 - перезапуск СУБД узла кластера компонента УРБД;
 - чтение и применение конфигурации кластера компонента УРБД;
- возможность снятия резервной копии с ведомого узла кластера компонента УРБД;
- функциональность снятия резервной копии аналогична таковой при снятии с ведущего узла.

Компонент выполняет следующие функции:

- поддержка транзакций с использованием версионности данных (MVCC);
- поддержка ссылочной целостности данных;
- поддержка реализации пользовательских типов данных;
- функционирование в режиме отказоустойчивого кластера: синхронная репликация данных;
- механизмы переключения роли узла при аварии на ведущем узле;
- управление отказоустойчивым кластером;
- автоматическое восстановление сервиса после одиночного сбоя;
- снятие резервной копии с ведомого узла;
- поддержка прозрачного шифрования данных;

- шифрование временных файлов, используемых для сортировки или слияния данных в операциях над данными;
- восстановление из резервной копии на ноде, относящейся к другому кластеру без шифрования;
- исключение переноса файлов отношений на ноду, относящуюся к другому кластеру без шифрования;
- обеспечение защиты параметров от привилегированных пользователей;
- обеспечение защиты данных от привилегированных пользователей;
- создание парольной политики пользователей;
- обеспечение аудита действий пользователей;
- получение показателей, связанных с функциональностью;
- отключение слабых типов аутентификации;
- инициализация механизма защиты данных с помощью вспомогательной утилиты;
- восстановление резервной копии, содержащей зашифрованные объекты баз данных под управлением компонента УРБД, на узле того же отказоустойчивого кластера, с которого была снята резервная копия;
- инициализация параметров подключения к защищенному хранилищу с помощью вспомогательной утилиты;
- интеграция с защищенным хранилищем.

4.1.4.2 Компонент «Распределенное хранение и обработка данных в памяти Platform V Ignite Sber Edition»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Распределенного хранения и обработки данных в памяти Platform V Ignite Sber Edition».

Данный компонент Платформы обеспечивает надежное распределенное хранение данных в оперативной памяти и распределенную обработку этих данных, включая:

- хранение данных в оперативной памяти с возможностью сохранения данных на диск, включающее в себя:
 - распределённое хранение данных вида «ключ-значение» в оперативной памяти:
 - в режиме, когда на каждом узле кластера хранится полная копия данных;
 - в режиме, когда на узле кластера хранится только часть данных (партиция);
 - возможность сохранения данных из оперативной памяти на жёсткий диск узла кластера
 - отсутствие потери данных при выходе из строя одного узла кластера;
 - отсутствие потери данных при полной аварийной остановке кластера, при включённом сохранении данных на диск и достаточном размере диска для сохранения всех данных из оперативной памяти;
 - возможность доступа к данным при помощи Java Key Value API и SQL DML, соответствующий стандарту ANSI 99, включающее в себя:
 - операции записи данных в оперативную память с использованием Java Key Value API;
 - операции чтения данных из оперативной памяти с использованием Java Key Value API;
 - операции записи данных в оперативную память с использованием JDBC SQL;
 - операции чтения данных из оперативной памяти с использованием JDBC SQL;
- возможность распределенной обработки данных, включающее в себя:
 - запуск на кластере произвольной вычислительной задачи с помощью толстого клиента;
 - разделение задачи на подзадачи;
 - отправка подзадач на узлы кластера для одновременной параллельной обработки;
 - сбор результатов выполнения задачи;

- обобщение результатов;
- передача обобщённого результата выполнения задачи на толстый клиент;
- шифрование данных на диске, включающее:
 - локальное хранение ключей шифрования на узле кластера;
 - хранение зашифрованного кэш-ключа в Meta Store;
 - хранение зашифрованного мастер-ключа в Java Keystore;
 - возможность управления ключами шифрования прозрачного шифрования данных, включая функции:
 - смена мастер-ключа;
 - шифрование файлов журнала операций (WAL);
 - шифрование файлов данных, хранящихся на жёстком диске на узле кластера ;
 - Резервное копирование данных кластера, включающее:
 - создание целостной резервной копии всех нешифрованных данных кластера, хранящихся на жёстких дисках узлов кластера;
 - возможность восстановления данных на всех узлах кластера из резервной копии на ту же топологию.

4.1.4.3 Компонент «Прикладной журнал Platform V Data Tools»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента прикладной репликации «Прикладной журнал Platform V Data Tools».

Компонент «Прикладной журнал Platform V Data Tools» предназначен для прикладной репликации данных, т. е. логической репликации непосредственно с уровня сервера приложения и выбора БД, с которой работает приложение – основная (NORMAL) или резервная (STANDIN) (вторая БД при этом должна функционировать в режиме логической реплики).

Компонент предоставляет следующие возможности:

- приложению (с использованием клиентской библиотеки standin-client):

- отправить вектор изменений (т. е. строку, содержащую первичный ключ и набор неключевых атрибутов) в прикладной журнал;
- получить отправленный вектор изменений;
- получить информацию, какая БД в настоящее время является основной (NORMAL или STANDIN);
- администратору (с использованием АРМ администратора, входящего в комплект поставки):
 - понять статус репликации по каждому приложению, использующему standin-client;
 - управление блокировками объектов данных (установка признака блокировки объекта, доставка признака потребителю);
 - переключить приложение, использующее standin-client, с базы данных NORMAL на STANDIN и обратно;
 - повторно отправить вектор изменений, если он применился с ошибкой;
 - отключить и включить поток векторов изменений.

4.1.4.4 Компонент «Архивирование Platform V Data Tools»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонент «Архивирование Platform V Data Tools».

Компонент «Архивирование Platform V Data Tools» имеет следующую функциональность:

- возможность потоковой (инкрементальной) загрузки данных из прикладного приложения в Фабрику данных:
 - инкрементальная (потоковая) выгрузка реализована путем обработки векторов изменений (Change Vector), генерируемых прикладным приложением;
 - получение векторов изменений реализовано путем интеграции с компонентом «Прикладной журнал Platform V Data Tools»;
 - инкрементальная выгрузка работает в фоне, постоянно (непрерывно);

- управление потоком векторов изменений выполняется из АРМ компонента «Прикладной журнал Platform V Data Tools» (включение, отключение репликации, повторная репликация векторов изменений);
- данные при потоковой (инкрементальной) загрузке отфильтровываются согласно правилам фильтрации (Белого списка), задаваемых в компоненте «Архивирование Platform V Data Tools»;
- интеграционное взаимодействие между Платформой по работе с данными Сбера и компонентом «Архивирование Platform V Data Tools» организовано с использованием интеграционной шины с использованием компонента обмена сообщениями;
- для оперирования моделью данных Источника компонент «Архивирование Platform V Data Tools» интегрирует в себе (путем прямого подключения зависимости в виде MAVEN-артефакта) представления хранения модели системы-Источника. Для поддержки изменения модели данных Источника компонент «Архивирование Platform V Data Tools» также поддерживает обновление модели путем обновления зависимости посредством пересборки (перекомпиляции) артефакта адаптера соответствующего источника с указанием обновленной версии MAVEN- зависимости (зависимостей), содержащих описание модели;
- для поддержки специфичных (нестандартных) нотаций первичных ключей, ссылок, трансформации специфичных типов данных в строковое (читаемое) представление, компонент «Архивирование Platform V Data Tools» реализует внутри себя специализированный код и формирует данные преобразования на этапе компиляции (сборки) адаптера Источника. Для этого в компоненте «Архивирование Platform V Data Tools» реализован механизм для однократной (по требованию) генерации описания схемы данных приложения источника-данных, на основе которой на стороне Платформы по работе с данными Сбера выполняется генерация таблиц в хранилище, в которой описывается структура хранения данных в хранилище данных. Формат передачи данных между компонентом «Архивирование Platform V Data Tools» и Платформой по работе с данными Сбера соответствует следующим принципам:

- объекты передаются преобразованными из специфичной модели хранения Источника в плоскую реляционную модель по принципу "одна сущность - один самостоятельный объект";
- вложенные (дочерние) объекты передаются как отдельные сущности, родительский объект содержит уникальный идентификатор (ссылку) на дочерний объект;
- при отсутствии уникальной идентификации дочерних объектов их естественным ключом - ключ формируется как композитный, путем соединения ключа родительского и дочернего объекта через символ точки;
- объекты передаются сериализованными, соответствуют плоскому реляционному представлению структуры каждого объекта, передаваемого в Фабрику данных, и соответствующую спецификации Avro не ниже 1.8;
- конфигурирование компонента «Архивирование Platform V Data Tools», включая параметры подключения к компоненту обмена сообщениями и настройки правил фильтрации (Белых Списков), выполняются посредством конфигурирования соответствующих артефактов в компоненте «Управление настройками приложений Platform V Configuration». При этом формат белого списка должен соответствовать списку по следующей нотации: <имя типа>:< имя атрибута>\n;
- загрузку данных в Платформу по работе с данными Сбера.

4.1.4.5 Компонент «Виртуализация слоя хранения данных Hibernate Platform V Data Tools»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Виртуализация слоя хранения данных Hibernate Platform V Data Tools».

Компонент представляет собой библиотеку, обеспечивающую работу с данными в реляционной БД как с объектами Java (Object/Relational Mapping).

Компонент обеспечивает интеграцию с сервисами Платформы:

- интеграцию с компонентом «Межкластерная индексация Platform V Application Sharding»:

- при сохранении новой сущности, имеющей глобально проиндексированное поле, значение этого поля и идентификатор шарда передаётся в API сервиса межкластерной индексации;
- при изменении глобально проиндексированного поля новое значение передаётся в API сервиса межкластерной индексации;
- интеграцию с компонентом «Генерация уникальных ID Platform V Generators»:
 - при создании нового объекта и его записи в БД идентификатор записи генерируется с использованием API компонента генерации уникальных идентификаторов сущностей;
- интеграцию с компонентом прикладной репликации «Прикладной журнал Platform V Data Tools»:
 - все изменения данных, выполненные в транзакции, передаются в компонент прикладной репликации;
 - обеспечивается применение изменений, полученных из компонента прикладной репликации, в резервную СУБД, если эти изменения сгенерированы компонентом виртуализации;
 - переключение между основной и резервной СУБД осуществляется по команде из ARМ администратора компонента прикладной репликации;
 - компонент может запрещать пакетные изменения в БД.

4.1.4.6 Компонент генерации синтетических тестовых данных «Synteta»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью «Компонента генерации синтетических тестовых данных «Synteta».

Компонент обеспечивает создание набора синтетических тестовых данных в тестируемых приложениях через их API за счет предоставления следующих функций:

- настройка модели тестовых данных;
- создание сущностей на ER-диаграмме;
- добавление атрибутов сущностей;

- редактирование параметров атрибутов, таких как:
 - идентификатор;
 - название;
 - значение по умолчанию;
 - регулярное выражение;
 - пример параметра;
 - генерируемое значение параметра;
 - признак обязательности заполнения параметра;
- настройка сценариев записи тестовых данных в приложения в виде связанной совокупности объектов:
 - настройка последовательности шагов по получению синтетических тестовых данных на основе модели данных;
 - настройка последовательности шагов по записи данных в приложения;
- подключение API приложений для записи и получения данных путем указания API и реквизитов для подключения (endpoint, учетная запись/пароль);
- запуск генерации тестовых данных через UI:
 - выбор модели данных;
 - выбор сущности для генерации;
 - редактирование параметров атрибутов сущностей;
 - запуск генерации тестовых данных в формате асинхронного задания;
- возможность скачивания тестовых данных по успешно выполненным заданиям в структурированном формате.

4.1.4.7 Компоненты автонумерации и генерации уникальных идентификаторов

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Генерация уникальных ID Platform V Generators» и компонента «Автонумерация Platform V Generators».

4.1.4.7.1 Компонент «Автономная генерация Platform V Generators»

Компонент «Автономная генерация Platform V Generators» обеспечивает функциональность централизованной выдачи отформатированных номеров с помощью счетчиков для сквозной нумерации бизнес-сущностей (например, номеров заявок).

Компонент предоставляет следующие возможности с использованием API по протоколу REST:

- регистрация счетчика с заданным именем;
- установка формата выдаваемых номеров для счетчика;
- выдача счетчиком отформатированных номеров.

Компонент «Генерация уникальных ID Platform V Generators» обеспечивает функциональность по генерации идентификаторов сущностей на множестве узлов приложения.

4.1.4.7.2 Компонент «Генерация уникальных ID Platform V Generators»

Компонент «Генерация уникальных ID Platform V Generators» предоставляет возможности с использованием API по протоколу REST:

- регистрация прикладной сущности с заданным именем;
- координация узлов прикладного сервиса-потребителя;
- формирование уникальных целочисленных 64-битных идентификаторов сущностей.

4.1.4.8 Компонент «Управление справочной информацией Platform V Dictionaries»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Управление справочной информацией Platform V Dictionaries».

Компонент «Управление справочной информацией Platform V Dictionaries» обеспечивает работу со справочной информацией плоской структуры из приложений, реализованных на Платформе, а именно, обеспечивает:

- возможность, используя АРМ администратора справочников:
 - найти необходимый справочник по его коду;
 - перейти к просмотру данных версии справочника;
 - создать новую версию внутреннего справочника;
 - создать новую версию внутреннего справочника путём копирования существующей версии справочника;
 - отредактировать произвольную строку созданной версии внутреннего справочника;
 - выполнить загрузку данных в созданную версию внутреннего справочника из файла CSV определенного формата;
 - выполнить активацию (сделать доступной для использования) наполненной данными версии внутреннего справочника;
- возможность, используя API:
 - запросить данные активной версии справочника;
 - выполнить создание новой версии внешнего справочника;
 - выполнить загрузку данных, переданных в теле запроса, в созданную версию внешнего справочника;
 - выполнить активацию (сделать доступной для использования) наполненной данными версии внешнего справочника.

4.1.4.9 Компонент «Управление данными Platform V DataSpace»

Соответствие Платформы предъявляемым требованиям обеспечено реализацией следующего функционала в компоненте «Управление данными Platform V DataSpace»:

- возможность проектирования и расширения модели данных бизнес-области
 - данные каждой бизнес-области (агрегата) изолированы друг от друга (возможность разделения в разные физические БД). Модель должна быть описана в формате XML согласно руководству разработчика.
- инструмент генерации базовых сервисов для работы с данными на основе модели (п.1) в рамках конкретной бизнес-области (агрегата):

- создание/изменение данных с поддержкой идемпотентности и возможностей оптимистичного/пессимистичного блокирования ресурсов (элементов модели):
 - предоставляет API с возможностью создания объекта (в т.ч. с возможностью установки ключа идемпотентности)
 - предоставляет API с возможностью изменения объекта (в т.ч. с возможностью установки ключа идемпотентности)
 - предоставляет API с возможностью блокировки/разблокировки объекта(пессимистичного/оптимистичного)
- иерархическое чтение данных:
 - предоставляет API для чтения с возможностью задать спецификацию, включающую:
 - примитивные атрибуты
 - ссылки на другие объекты, с запросом атрибутов у этих объектов
 - коллекции других объектов, с запросом атрибутов у этих объектов
- поиск данных:
 - предоставляет API для поиска объектов с возможностью задать спецификацию, включающую условие с фильтрами EQ, NOT_EQ, IN, заданными по атрибутам объекта.
 - предоставляет API для поиска объектов с возможностью задать спецификацию, включающую условие с фильтрами LESS, GREATER, BETWEEN, заданными по датам, числовым и строковым атрибутам объекта.
 - предоставляет API для поиска объектов с возможностью задать спецификацию, включающую условие с фильтрами LIKE заданными по строковым атрибутам объекта.
 - предоставляет API для поиска объектов с возможностью задать спецификацию, включающую условие с логическими операторами AND и OR для фильтров.

- предоставляет API для поиска объектов с возможностью задать спецификацию, включающую условие с инверсией фильтра (ключевое слово NOT).
- предоставляет API для поиска объектов с возможностью задать спецификацию, включающую условие с фильтром на NULL для атрибутов типа ссылка и коллекция.
- предоставляет API для поиска объектов с возможностью задать спецификацию, включающую условия с агрегирующими функциями COUNT, MIN, MAX, AVG над числовыми атрибутами и фильтром для атрибутов-коллекций.
- предоставляет API для поиска объектов с возможностью задать спецификацию, включающую условия с математическими функциями PLUS, MINUS, MUL, DIV над числовыми атрибутами.
- предоставляет API для поиска объектов с возможностью задать спецификацию, включающую условия сортировки и пейджинации результирующей выборки.
- группирование исполнения операций ((создание/изменение данных с поддержкой идемпотентности и возможностей оптимистичного/пессимистичного блокирования ресурсов (элементов модели), иерархическое чтение данных) в рамках одного транзакционного вызова по одному экземпляру агрегата (UnitOfWork));
- формирование и применение вектора изменений в рамках выполняемых транзакций (посредством интеграции с компонентом прикладной репликации «Прикладной журнал Platform V Data Tools»).

4.1.4.10 Компонент «Обеспечение надёжности Failover Platform V Monitor».

Соответствие платформы предъявляемым требованиям обеспечивается реализацией требуемой функциональности в компоненте «Обеспечение надёжности Failover Platform V Monitor».

Компонент платформы обеспечивает функционал обнаружения и устранения неисправностей в работе сервера приложений RedHat Wildfly Application Server и бизнес-приложений под его управлением.

Компонент осуществляет свои функции при помощи специализированного приложения - агента, который устанавливается на каждый узел с сервером приложений.

Компонент предоставляет следующие возможности:

- возможность, используя АРМ администратора:
 - увидеть все серверы, где установлены агенты диагностики компонента «Обеспечение надёжности Failover Platform V Monitor»;
 - увидеть все серверы приложений, которые установлены на серверах с агентами;
 - увидеть все приложения, установленные на серверах приложений с детализацией статусов работы по каждому приложению;
 - создать именованную группу серверов приложений для настройки прав доступа на основе ролевой модели;
 - добавить/удалить сервер приложений в именованную группу;
 - запросить метрики работы отдельного сервера приложений, опубликованные через jmx;
 - остановить/запустить/перезапустить сервер приложений в ручном режиме;
 - остановить прием и обработку сообщений сервером приложений;
 - возобновить прием и обработку сообщений сервером приложений;
 - выполнить настройку правил выявления неисправностей в работе сервера приложений или установленных на нём бизнес приложений на основе метрик, поставляемых приложением, с использованием логических и арифметических операций, а также условных операторов ветвления;
 - выполнить email-оповещение администраторов при возникновении события переключения на резервные узлы;
- возможность, используя правила:

- указать пороговые значения метрик, при превышении которых должны быть применены корректирующие воздействия (стоп/старт/передача управления другому серверу);
- остановить/перезапустить сервер приложений в автоматическом режиме, на основании срабатываний правил обработки метрик;
- автоматически снять нагрузку с сервера приложений в случае сбоя или нарушения функциональности самого сервера приложений или приложений, размещенных на нем;
- разграничение функциональности редактирования правил, сбора и анализа метрик согласно ролевой модели;
- агент диагностики компонента «Обеспечение надёжности Failover Platform V Monitor», запущенный на виртуальном или физическом сервере:
 - представляет собой приложение, запущенное в отдельной Java-машине;
 - читает метрики серверов приложений через JMX и REST;
 - собирает метрики работы физического или виртуального сервера используя нативные команды ОС;
 - исполняет над полученными метриками операции преобразования, агрегации и условных переходов, описанных с использованием языка groovy.

4.1.5 Компоненты аналитики данных

4.1.5.1 Компонент «Система контроля качества данных SDP Data Quality»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента управления качеством данных SDP Data Quality.

Компонент управления качеством данных имеет следующую функциональность:

- инструменты для настройки проверок и бизнес-правил, которые позволяют контролировать качество данных;

- подключение функционала контроля качества данных к процессу пакетной загрузки данных;
- создание, редактирование проверок качества данных на базе собственных бизнес-правил или параметризация готовых шаблонов проверок;
- запуск пользовательских проверок качества данных;
- визуализацию текущего состояния проверок качества данных, а также хранение версионности проверок.

4.1.5.2 Платформа по работе с данными Сбера SberData Platform

4.1.5.2.1 Компонент хранения больших объемов неструктурированных данных «Платформа по работе с данными Сбера SberData Platform»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента хранения больших объемов неструктурированных данных «Платформа по работе с данными Сбера SberData Platform».

Данный компонент Платформы обеспечивает хранение данных в объемах, исчисляемых петабайтами, а также высокую скорость обработки данных с учетом распределенной модели исполнения запросов. В том числе компонент Платформы обеспечивает:

- работу с распределенной файловой системой для хранения больших объемов данных, поблочно распределенных между узлами вычислительного кластера;
- репликацию данных на уровне узлов кластера для повышения отказоустойчивости файловой системы;
- работу с распределенными параллельными вычислениями при обработке больших объемов данных;
- доступ к хранимым данным средствами SQL и NoSQL;
- поддержку пакетной и потоковой загрузки данных;
- настройку ограничения доступа к данным в распределенной файловой системе;

- инструменты для планирования заданий и управления вычислительными ресурсами кластеров;
- инструменты для разворачивания, управления и мониторинга вычислительных кластеров;
- хранение неструктурированных данных (изображения, видео файлы, аудио файлы, скан-копии документов, интернет-статьи, электронные письма, файлы всех типов).

4.1.5.2.2 Компонент загрузки и преобразования данных «Платформа по работе с данными Сбера SberData Platform»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента загрузки и преобразования данных «Платформа по работе с данными Сбера SberData Platform».

Данный Компонент Платформы обеспечивает агрегацию данных из различных источников для наполнения хранилищ данных в соответствии с потребностью бизнес-модели, в том числе компонент обеспечивает:

- поддержку множества структурированных и неструктурированных форматов исходных данных для загрузки, а именно:
 - таблицы баз данных;
 - CSV;
- поддержку стандартных протоколов работы и доступа к данным, таких как:
 - SFTP и SSH,
 - KAFKA,
 - HTTPS и SSL,
 - HDFS,
 - SQL;
- поддержку кластеризации с возможностью работы на нескольких узлах для повышения производительности обработки данных;
- преобразование потоков данных в формат PARQUET из CSV;
- управление множеством параллельных потоков данных;

– графический пользовательский интерфейс для настройки и управления потоками загрузки и преобразования данных.

4.1.5.2.3 Компонент разработки моделей машинного обучения «Платформа по работе с данными Сбера SberData Platform»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента управления качеством данных «Платформа по работе с данными Сбера SberData Platform».

Компонент разработки моделей машинного обучения обеспечивает поставку и подготовку данных, исследование данных и проверку гипотез, а также построение моделей, в том числе:

- среду для интерактивной разработки алгоритмов машинного обучения и исследования данных;
- поддержку установки и обновления Python-библиотек машинного обучения;
- поддержку пошагового выполнения Python-инструкций;
- поддержку построения графиков для промежуточных результатов вычислений;
- возможность подключения к компонентам хранения структурированных и неструктурированных данных;
- доступ через веб-браузер к компоненту без установки дополнительного программного обеспечения на рабочие станции.

4.1.5.3 Компонент управления жизненным циклом моделей машинного обучения «SberDataScience»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента управления жизненным циклом моделей машинного обучения «SberDataScience».

Компонент управления жизненным циклом моделей машинного обучения «SberDataScience» обеспечивает:

- загрузку данных, обучение моделей машинного обучения, валидацию моделей машинного обучения, исполнение моделей машинного обучения, мониторинг моделей машинного обучения и формирование отчета;
- наличие визуального конструктора моделей машинного обучения;
- предоставление готового набора библиотек для разработки моделей машинного обучения;
- расширение функциональности Платформы за счет подключения дополнительных алгоритмов и пользовательских моделей;
- развертывание и запуск модели машинного обучения;
- возможность исполнения и мониторинга моделей.

4.1.6 Компоненты интеграции

4.1.6.1 Компонент «HTTP-клиент Platform V Synapse Service Mesh»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «HTTP-клиент Platform V Synapse Service Mesh».

Библиотека HTTP-клиент подключается как зависимость и используется в коде прикладного решения.

После подключения к прикладному решению компонент предоставляет для прикладного решения следующие возможности:

- отправка POST-запроса с body (телом) в формате application/json в заданный endpoint и получение ответов, как успешных, так и исключений;
- прокидывание полученных заголовков для обеспечения маршрутизации http-запросов;
- обогащение http-запросов необходимыми для сохранения бизнес-логики заголовками, которые используются в платформе;
- в качестве защиты от каскадных сбоев используется механизм circuitBreaker с возможностью его отключения;
- возможность настроить таймауты на соединение с сервером и на вычитывание ответов.

4.1.6.2 Компонент «REST-запросы Platform V Synapse Service Mesh»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «REST-запросы Platform V Synapse Service Mesh».

Библиотека RestApp подключается как зависимость и используется в коде прикладного решения.

После подключения к прикладному решению компонент предоставляет для прикладного решения следующие возможности:

- выставление REST-сервиса для определенной в коде точки подключения;
- расширенная обработка ошибок с возможностью формирования расширенного описания ошибки;
- настройка таймаута ответа и формат ошибки таймаута.

4.1.6.3 Компоненты проксирования запросов

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента проксирования запросов, который включает:

- компонент «Сервисный прокси Platform V Synapse Service Mesh»;
- компонент «Граничный прокси Platform V Synapse Service Mesh».

4.1.6.3.1 Компонент «Сервисный прокси Platform V Synapse Service Mesh»

Компонент «Сервисный прокси Platform V Synapse Service Mesh» выполняет следующие функции:

- балансировка запросов между несколькими экземплярами или версиями приложения на основе различных стратегий (равномерно по всем узлам (round-robin), с использованием весов, в привязке к конкретному адресу (affinity))
- маршрутизация запросов на основе различных стратегий (доступность приложения, поля заголовка, принадлежность определенной зоне);
- динамическое изменение интеграционных политик (правила маршрутизации, политики доступа, стратегия балансировки) в режиме runtime (без остановки приложения);

- применение интеграционных политик на различных уровнях транспортных протоколов (tcp, http);
- развертывание компонент платформы в виде отдельных элементов (в том числе режим sidecar) без влияния на прикладной код;
- формирование метрик мониторинга.

4.1.6.3.2 Компонент «Граничный прокси Platform V Synapse Service Mesh»

Компонент «Граничный прокси Platform V Synapse Service Mesh» обеспечивает:

- развертывание компонент платформы в виде отдельных элементов (в том числе режим sidecar) без влияния на прикладной код;
- квотирование максимального количества одновременных подключений к приложению с помощью конфигурационных файлов;
- динамическое изменение интеграционных политик (правил маршрутизации, политик доступа, стратегия балансировки) в режиме runtime (без остановки приложения);
- аутентификацию и авторизацию вызовов;
- формирование метрик трассировки запросов и мониторинга компонент.

4.1.6.4 Компонент «Шлюзы AC Platform V Synapse Enterprise Integration»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Шлюзы AC Platform V Synapse Enterprise Integration».

В состав компонента «Шлюзы AC Platform V Synapse Enterprise Integration» входят три независимых сервиса.

Сервис «Шлюз AC. JMS» обеспечивает следующие возможности при взаимодействии с ActiveMQ:

- чтение сообщений из очередей Active MQ и их отправку в соответствующие сервисы по протоколу gRPC;

- gRPC API отправки JMS сообщений в очереди ActiveMQ;
- работу в режимах:
 - асинхронном - без ожидания ответа;
 - синхронном - с ожиданием ответа и отправкой его инициатору;
- подключение к нескольким брокерам ActiveMQ;
- логирование событий приема и отправки сообщений;
- формирование трассировочной информации;
- возможность подключения к брокеру ActiveMQ с использованием аутентификации по SSL-сертификатам;
- формирование метрик мониторинга.

Сервис «Шлюз AC.Kafka» предоставляет следующие возможности:

- чтение сообщений из топиков Kafka и их отправку в соответствующие сервисы по протоколу gRPC;
- запись сообщений в топики Kafka посредством gRPC API;
- подключение к нескольким брокерам Kafka, в том числе и с помощью механизма аутентификации по SSL-сертификатам;
- логирование событий приема и отправки сообщений;
- авторизацию вызовов;
- формирование метрик мониторинга.

Сервис «Шлюз AC.Маппер» предоставляет следующие возможности:

- выполнение функции преобразование формата сообщения запроса xml на основе xsl, с последующим вызовом сервиса назначения, получения и преобразования ответа, возврат его потребителю;
- выполнение функции преобразование формата сообщения запроса json на основе jsonT, с последующим вызовом сервиса назначения, получения и преобразования ответа, возврат его потребителю.

4.1.6.5 Компонент «Внутренний шлюз Platform V API Management»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Внутренний шлюз Platform V API Management».

Компонент «Внутренний шлюз Platform V API Management» обеспечивает доступ к микросервисам приложений во внутреннем контуре сети для внутренних приложений в микросервисной архитектуре:

- reverse проху для http и https запросов (приложение выставляет свои rest end-points на шлюзе - шлюз при поступлении запросов на эти end-points проксирует на сервера с инсталляцией приложения);

- настройку системных параметров соединений:

- таймауты соединения клиента;

- таймауты соединения с upstream сервером;

- максимальный размер тела запроса, ответа.

- пассивную проверку работоспособности;

- активную проверку работоспособности:

- http и https протоколы;

- поддержку платформенного агента проверки работоспособности и его статусной модели;

- мониторинг в режиме реального времени статуса активной работоспособности по всем серверам;

- возможность проверки работоспособности и доступности приложения в кластере;

- настраиваемое добавление http заголовков в запросы (entryPoint для статики и другие):

- сквозное проксирование заголовков (для трекинга запросов);

- работа заголовков X-Forwarded-For;

- управление выбором минорной и мажорной версии микросервиса.

Настраиваемая маршрутизация запроса – выбор группы балансировки для запроса по заданной комбинации значений определенных http заголовков;

- mTLS:

- кеширование сессий;

- правила по проверке CN потребителя;

- функционал `slow start` – плавное нарастание нагрузки при включении сервера (`circuit breaker`);
- поддержку алгоритмов распределения запросов в группе балансировки:
 - `round robin`;
 - `consistent hash` по выбранному параметру запроса, например заголовок или `cookie`;
 - `sticky` по выбранному параметру запроса, например заголовок или `cookie`.
Выбор алгоритма расчета значения `sticky cookie`; `md5`, `sha`, `text`;
- квотирование запросов (`rate limits`):
 - на весь сервер;
 - к отдельному микросервису в разрезе других микросервисов;
 - к отдельному микросервису в разрезе каналов, `tenantCode`:
 - квота на сумму запросов к микросервису;
 - квоты по произвольным `http` заголовкам.
- отправку метрик работы в систему мониторинга, включая координаты источника запроса и назначения запроса (`трекинга` запросов);
- отправку лога работы шлюза в систему журналирования, настраиваемый список атрибутов записи лога;
- функционал `web` сервера – раздача статики, включая настройку параметров кеширования статики на клиенте;
- пользовательский интерфейс для мониторинга текущего состояния;
- пользовательский интерфейс для просмотра и изменения параметров кластера шлюзов, `online` изменение параметров на шлюзе.

Компонент «Внутренний шлюз `Platform V API Management`» обеспечивает единую входную точку для внешних систем к микросервисам приложений внутреннего контура в микросервисной архитектуре, в том числе:

- получение конфигураций из реестра микросервисов для прикладной бизнес-области;
- изменение конфигурации шлюза в среде исполнения;

- балансировка запросов между группами узлов микросервисов и узлами микросервисов внутри группы;
- определение доступности узла микросервисов (активная и пассивная диагностика работоспособности), при недоступности узла микросервисов узел выводится из балансировки, при этом запросы на него не направляются;
- стабильное подключение сессии приложения к узлу микросервисов (sticky session);
- квотирование максимального количества запросов/сек к группе балансировки микросервиса.

4.1.6.6 Компонент «Доступ к единице топологии Платформы Platform V Application Sharding»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Доступ к единице топологии Платформы Platform V Application Sharding».

Компонент «Доступ к единице топологии Платформы Platform V Application Sharding» обеспечивает:

- перенаправление входящих на компонент HTTP и HTTPS-запросов от клиентов в точку доступа выбранной единицы топологии. Единицы топологии и компонент должны находиться в одном контуре сети. Выбор единицы топологии происходит на основе правил маршрутизации, заданных в конфигурации компонента. Выбор единицы топологии для перенаправления запроса определяется на основании:
 - значения определённого в конфигурации компонента HTTP-заголовка в запросе;
 - ответа, полученного от компонента «Маршрутизация Platform V Application Sharding», в который компонент направляет информацию о запросе. Компонент «Маршрутизация Platform V Application Sharding» возвращает код единицы топологии, в которую компонент перенаправит запрос. Полученный код единицы топологии от компонента «Маршрутизация Platform V Application Sharding» сохраняется во

внутренней памяти компонента и используется при выборе единицы топологии при следующих аналогичных запросах;

– возможность принимать и перенаправлять запросы по протоколу TLS, путем указания в конфигурации и ее загрузки на компонент. В конфигурации должна быть возможность настроить переиспользование TLS-сессии с клиентом;

– возможность отправлять метрики по работе компонента в платформенную систему мониторинга путем указания в конфигурации и ее загрузке на компонент. Следующие метрики должны отправляться в систему мониторинга: количество установленных на текущий момент соединений, количество обработанных запросов по каждому правилу маршрутизации;

– возможность отправлять логи по работе компонента в платформенную систему журналирования путем указания в конфигурации и ее загрузки конфигурации на компонент;

– возможность посмотреть в пользовательском интерфейсе:

– конфигурационные файлы;

– версию компонента;

– общее количество обработанных компонентом запросов;

– статистику обработки запросов по каждому правилу маршрутизации от момента запуска/перезагрузки;

– возможность сквозного проксирования HTTP-заголовков для трекинга запросов и возможность работы заголовков X-Forwarded-For.

4.1.6.7 Компоненты событийной передачи и обработки событий

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью следующих компонентов:

– «Передача событий Platform V Synapse Event Processing»;

– «Потоковая обработка событий Platform V Synapse Event Processing».

4.1.6.7.1 Компонент «Передача событий Platform V Synapse Event Processing»

Компонент «Передача событий Platform V Synapse Event Processing» обеспечивает:

- возможность создания новых потоков событий со следующими параметрами:
 - срок хранения событий в транспортном слое и/или максимальный размер потока событий;
 - число копий события для обеспечения отказоустойчивости публикации событий;
 - наименование потока событий согласно правилам наименования;
 - максимальный размер события для данного потока;
 - допустимый параллелизм обработки события;
- возможность назначения прав на публикацию и подписку по каждому потоку событий на основании модели авторизации, базирующейся на полном наименовании сертификата, используемого при подключении;
- соединение с узлами передачи событий на основе протокола TLS;
- возможность публикации события (plaintext, json и xml-нотации) с сохранением его в транспортном слое;
- возможность потребления события из потока по следующим правилам:
 - потребители могут объединяться в группы для обеспечения масштабирования обработки;
 - все активные экземпляры потребителей, принадлежащих одной группе, получают суммарно весь актуальный поток событий;
 - каждая активная группа потребителей получает весь актуальный поток событий;
- удаление события по достижению заданного срока хранения/максимального размера потока.

4.1.6.7.2 Компонент «Потоковая обработка событий Platform V Synapse Event Processing»

Компонент «Потоковая обработка событий Platform V Synapse Event Processing» обеспечивает:

- возможность создания новых заданий на потоковую обработку событий с учетом текущей емкости экземпляра потоковой обработки событий;
- возможность запуска/остановки заданий на потоковую обработку событий;
- возможность запускать несколько параллельно исполняющихся идентичных задач на потоковую обработку событий;
- возможность описания преобразований (трансформация, фильтрация, агрегация) одного или нескольких входящих потоков, имеющих объектную модель и публикуемых в формате XML или JSON, на псевдоязыке, с порождением одного или нескольких результирующих потоков событий. Псевдоязык поддерживает следующие операнды:
 - математические (сложение, вычитание, умножение, деление, целочисленное деление);
 - строковые (конкатенация, получение подстроки, разделение на несколько строк, проверка префикса/суффикса, поиск по регулярным выражениям);
 - условный оператор;
 - цикл;
 - работа с датой-временем (сравнение, получение текущей);
 - работа с последовательностями (создание, расширение).

4.1.6.8 Компонент «Обмен сообщениями Platform V Kafka Sber Edition»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Обмен сообщениями Platform V Kafka Sber Edition».

Компонент «Обмен сообщениями Platform V Kafka Sber Edition» обеспечивает:

- запись сообщения в топик (адресное пространство);
- чтение сообщений из топика;
- доступность данных при потере одного узла;

- сохранность данных при аварийной остановке всех узлов кластера;
- возможность перераспределить нагрузку между узлами;
- отправка информации в платформенный компонент аудита, в частности:
 - создание списка управления доступом (ACL);
 - удаление ACL;
 - создание топика;
 - удаление топика;
- предоставление событий мониторинга по JMX протоколу;
- подготовку инструментов администрирования для:
 - создания/удаления топика;
 - создания/удаления ACL;
 - распределения нагрузки между узлами кластера;
 - изменения конфигурации топиков;
 - увеличения фактора репликации топиков;
 - сохранения/восстановления конфигурации ACL;
 - получения информации по смещению топиков;
 - перезапуска кластера по алгоритму RollingRestart.

4.1.6.9 Компоненты межкластерной индексации и маршрутизации

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью следующих компонентов:

- «Межкластерная индексация Platform V Application Sharding»;
- «Маршрутизация Platform V Application Sharding»;
- «Диспетчер входа Platform V API Management».

4.1.6.9.1 Компонент «Межкластерная индексация Platform V Application Sharding»

Компонент «Межкластерная индексация Platform V Application Sharding» обеспечивает управление данными, необходимыми для реализации горизонтального масштабирования Платформы и включает:

– ведение реестра индексов, путём конфигурирования через configmap Kubernetes:

– добавление нового индекса со следующими параметрами:

– имя индекса;

– тип ключа в индексе:

– String;

– Long;

– UUID;

– стратегия балансировки данных:

– выбор произвольной шарды на основе идентификатора (RoundRobin);

– отключение выбора шарды (Fail);

– изменение стратегии балансировки данных существующего индекса;

– массовое распределение ключей шардирования по прикладным шардам, с помощью утилиты командной строки, подключающейся по протоколу JDBC к: (1) базе данных источника провайдера сервиса, и (2) базе Компонента межкластерной индексации:

– массовое добавление ключей в индекс с привязкой к шарде;

– массовое изменение значения шарды по индексу и ключам;

– управление индексом распределения ключей шардирования по прикладным шардам, с помощью API:

– добавление ключа в индекс с привязкой к шарде;

– получение значения шарды по индексу и ключу:

– в теле HTTP-ответа;

– в заголовке "shard" HTTP-ответа;

– изменение значения шарды по индексу и ключу;

– удаление ключа из индекса;

– при добавлении новых данных выполняется проверка на уникальность по индексу и ключу;

- проверка типа ключа при добавлении новых данных. Тип ключа соответствует типу индекса, в который он вставляется;
- получение значения шарды по ключу, отсутствующему в индексе, по одной из следующих стратегий, с помощью API:
 - выбор произвольной шарды на основе идентификатора (RoundRobin);
 - отключение выбора шарды (Fail).

4.1.6.9.2 Компонент «Маршрутизация Platform V Application Sharding»

Компонент «Маршрутизация Platform V Application Sharding» обеспечивает следующий функционал управления данными маршрутизации конечных пользователей Платформы на целевые контуры:

- возможность, используя АРМ администратора:
 - посмотреть конфигурацию фронтальной части Платформы в виде графа с вершинами, представляющими отдельные узлы системы (уровня сегмент, сектор, кластер и контур), и ребрами, отображающими связи узлов (далее - топология), доступную Администратору;
 - посмотреть атрибуты узла топологии (код, наименование, тип, код родительского узла) и привязанные к нему коды узлов топологии, которые позволяют однозначно идентифицировать узлы, (далее - маршрут) и набор правил для вычисления маршрута (далее - стратегия);
 - привязать к узлу топологии одну из возможных стратегий:
 - шардирование по пользователю (TF_USER);
 - шардирование по компоненту (TF_SUBJECT);
 - шардирование по заголовку ufs-shard-key (TF_ANY);
 - шардирование по проценту нагрузки, направляемому на узлы топологии (Affinity Function);
 - шардирование и привязка конечного пользователя к узлу топологии (Onboarding);
 - создать привязку идентификатора конечного пользователя Платформы к маршруту;

- найти конечного пользователя Платформы, имеющего привязку к маршруту, по его идентификатору;
- посмотреть список доступных конечному пользователю Платформы маршрутов;
- изменить привязку конечного пользователя Платформы к маршруту;
- удалить привязку конечного пользователя Платформы к маршруту;
- возможность, используя API:
 - запросить маршруты конечного пользователя Платформы по его идентификатору, списку узлов топологии, для которых получается информация о дочерних узлах, и информации о приложении, осуществляющем запрос (код приложения, которое запрашивает маршрут, и узел, где оно находится);
- возможность, используя API:
 - запросить маршруты по коду узла топологии и дополнительным параметрам из заголовков клиентского запроса ("ufs-user-id", "ufs-replica-key", "ufs-shard-key", "ufs-src-node", "ufs-SubsystemCode", "ufs-SubsystemVersion", "ufs-tenant", "ufs-user-type", "x-forwarded-for"), заполняемых по правилам, определенным в документации Компонента доступа к единице топологии Платформы.

4.1.6.9.3 Компонент «Диспетчер входа Platform V API Management»

Компонент «Диспетчер входа Platform V API Management» обеспечивает следующую возможность: осуществить перенаправление конечного пользователя Платформы на контур канала, выбранный в соответствии со стратегией шардирования данного канала.

4.1.6.10 Компонент «Ведение метаинформации программных сервисов APIM Platform V API Management»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Ведение метаинформации программных сервисов APIM Platform V API Management».

Компонент «Ведение метаинформации программных сервисов APIM Platform V API Management» обеспечивает хранение и управление метаинформацией программных сервисов, осуществлять передачу конфигураций программных сервисов на шлюзы, предоставлять средства управления шлюзами API, а именно, обеспечивает:

- возможность, за счет интеграции с платформенными компонентами, реализующими функции внутреннего и внешнего шлюзов:
 - регистрировать экземпляры шлюзов в хранилище APIM;
 - осуществлять подписку шлюзов на уведомления об изменениях в параметрах программных сервисов в хранилище APIM;
 - передавать конфигурации программных сервисов на зарегистрированные шлюзы;
- возможность, используя АРМ администратора находить и видеть состояние шлюзов:
 - найти необходимый зарегистрированный экземпляр шлюза с учетом координаты кластера и типа шлюза;
 - просмотреть состояние сетевой доступности шлюза, статуса подписки, статуса наличия на шлюзе актуальных ревизий конфигураций программных сервисов;
 - отменить регистрацию экземпляра шлюза;
- возможность импортировать конфигурации программных сервисов через средства автоматизированной установки приложений, поставляемые с платформой;
- возможность, используя АРМ администратора находить и удалять конфигурации программных сервисов:

- найти необходимый программный сервис с учетом кода подсистемы, коду единицы развертывания, принадлежности к кластеру шлюзов и его типу;
- удалить программный сервис из хранилища APIM и с зарегистрированных шлюзов;
- возможность, используя АРМ администратора управлять конфигурациями программных сервисов:
 - удалить версию API программного сервиса;
 - выбрать группу балансировки программного сервиса по умолчанию;
 - изменить состав группы балансировки программного сервиса;
 - включить(отключить) режим проверки доступности узлов группы балансировки программного сервиса;
 - настроить маршрутизацию запросов на группы балансировки программного сервиса по заданным заголовкам;
 - настроить значения пропускной способности шлюза (количество запросов в секунду) для версии API программного сервиса по заголовкам, суммарно или по умолчанию;
 - настроить параметры (таймауты, максимальный размер тела запроса, буферизацию) проксирования для версии API программного сервиса;
 - просмотреть ревизии конфигурации программного сервиса;
 - откатить конфигурацию программного сервиса к конкретной ревизии;
- возможность, используя АРМ администратора видеть статус доступности на шлюзе узлов в группах балансировки программного сервиса.

4.1.6.11 Компоненты моделирования композитных сервисов

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью следующих компонентов:

- «Моделирование композитных сервисов Ядро КБТ Platform V Functions»;
- «Моделирование композитных сервисов Визард КБТ Platform V Functions».

4.1.6.11.1 Компонент «Моделирование композитных сервисов Ядро КБТ Platform V Functions»

Компонент «Моделирование композитных сервисов Ядро КБТ Platform V Functions» обеспечивает разработку и выполнение композитных сервисов в контейнерной среде (композитные сервисы представляют собой вызов API других микросервисов и используются для реализации прикладной бизнес-логики) и предоставляет следующую функциональность:

- запуск и выполнение композитных сервисов в контейнерной среде;
- вызов внешних сервисов по протоколу REST;
- вызов внешних сервисов по протоколу JSON-RPC;
- обеспечение повторного вызова внешнего сервиса в случае возникновения ошибки;
- обеспечение вызова композитного сервиса по протоколам JSON-RPC.

4.1.6.11.2 Компонент «Моделирование композитных сервисов Визард КБТ Platform V Functions»

Компонент «Моделирование композитных сервисов Визард КБТ Platform V Functions» предоставляет через APM разработчика следующую функциональность:

- создание проекта;
- редактирование настроек проекта;
- удаление проекта;
- моделирование композитных сервисов в проекте;
- сборка проекта композитных сервисов:
 - генерация Java-кода на основе BPMN-диаграммы;
 - компиляция Java-кода;
 - сборка docker-образа.

4.1.6.12 Компонент «Файловая передача Platform V Synapse File Exchange»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Файловая передача Platform V Synapse File Exchange».

Компонент «Файловая передача Platform V Synapse File Exchange» обеспечивает передачу файлов между приложениями Платформы, а именно:

- передача файлов по протоколу gRPC используя API:
 - авторизация операции передачи файлов между приложениями с помощью компонента платформы OTT;
 - шифрование файлов симметричным алгоритмом AES сгенерированным транзакционным ключом;
 - размещение зашифрованных файлов во внешнем хранилище по протоколу S3;
 - передача уведомлением приложению-адресату информации о файлах и транзакционного ключа, зашифрованного публичным ключом приложения-адресата;
- оповещение приложения о доступных для получения файлах с дешифрованным транзакционным ключом по протоколу gRPC:
 - приём уведомления от приложения-адресанта с информацией о файлах и транзакционным ключом, зашифрованным публичным ключом приложения-адресата по протоколу gRPC;
 - дешифрация транзакционного ключа приватным ключом приложения-адресата;
 - передача уведомления с информацией о файлах и дешифрованным транзакционным ключом приложению-адресату;
- оповещение приложения о доступных для получения файлах с дешифрованным транзакционным ключом по протоколу gRPC:
 - извлечение файлов из внешнего хранилища по протоколу S3;

- дешифрование файлов с помощью предоставленного приложением транзакционного ключа, и их передача приложению-адресату.

4.1.6.13 Компоненты асинхронных запусков пакетных вычислительных задач

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью следующих компонентов:

- «Планировщик заданий Platform V Batch»;
- «Пакетная обработка задач Platform V Batch».

4.1.6.13.1 Компонент «Планировщик заданий Platform V Batch»

Компонент «Планировщик заданий Platform V Batch» обеспечивает функциональность асинхронного запуска пакетных вычислительных задач по расписанию, в том числе:

- возможность, используя API по протоколу JSON-RPC:
 - создавать, получать, обновлять, останавливать и удалять задания;
 - принудительно запускать задания;
- возможность автоматически асинхронно запускать задания по расписанию;
- управление вычислительной нагрузкой для вычислений, запускаемых в порядке очереди;
- возможность запуска заданий в виде вызовов API по протоколам REST и JSON-RPC.

4.1.6.13.2 Компонент «Пакетная обработка задач Platform V Batch»

Компонент «Пакетная обработка задач Platform V Batch» обеспечивает функциональность асинхронного запуска пакетных вычислительных задач в порядке очереди, в том числе:

- возможность, используя API по протоколу JSON-RPC:
 - создавать, обновлять, удалять очереди и создавать задачи в очередях;
 - изменять параметр количества одновременно запускаемых задач в очереди;

- возможность автоматически асинхронно запускать задачи в порядке очереди, используя API по протоколу JSON-RPC;

- возможность запуска задач в виде вызовов API по протоколам REST и JSON-RPC.

4.1.7 Инструменты и компоненты безопасности Platform V IAM

Соответствие Платформы предъявленным требованиям обеспечено функциональностью следующих компонентов безопасности:

- «Модули IAM. Технологическая платформа Сбербанка»:

- компонент «Аутентификация Platform V IAM»;

- компонент «Авторизация ЕФС Platform V IAM»;

- компонент «Авторизация ППРБ Platform V IAM»;

- «Модуль Аудит. Технологическая платформа Сбербанка»:

- компонент «Аудит Platform V Audit».

Надлежащая реализация мер по защите информации на момент поставки должна быть подтверждена соответствующими сертификатами ФСТЭК России.

4.1.7.1 Компонент «Аутентификация Platform V IAM («Модули IAM. Технологическая платформа Сбербанка»)»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Аутентификация Platform V IAM («Модули IAM. Технологическая платформа Сбербанка»)».

Компонент Платформы обеспечивает идентификацию, аутентификацию пользователей при доступе к приложениям, в том числе:

- обработку запросов на аутентификацию от пользователей, при необходимости - принудительный запрос аутентификационных данных;

- реализация жизненного цикла (создание, изменение, блокировка/разблокировка) учётных записей пользователей;

- аутентификация пользователей по паролю;

- аутентификация пользователей по Open ID Connect;

- аутентификация пользователей на прокси-сервере IAM;
- поддержка HTTP(S);
- передачу идентификатора пользователя для нужд идентификации в подсистемах. Передача информации о пользователе осуществляется в формате JWT;
- формирование событий аудита в платформенную систему аудита для их последующей обработки. Передаются следующие события:
 - все действия по администрированию учётных записей пользователей;
 - изменения настроек IAM;
 - передаются события аутентификации, как успешные, так и неуспешные;
- создание аутентификационной сессии;
- ограничение доступа неаутентифицированного пользователя к компонентам Платформы за счет:
 - ограничения доступа на уровне IAM прокси, за исключением контекстов с разрешённым анонимным доступом;
 - каждая система разрешает доступ к своим ресурсам только при наличии валидного токена IAM.

4.1.7.2 Компонент «Авторизация Platform V IAM («Модули IAM. Технологическая платформа Сбербанк»))»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью следующих компонентов:

- «Авторизация ЕФС Platform V IAM («Модули IAM. Технологическая платформа Сбербанк»))»;
- «Авторизация ППРБ Platform V IAM («Модули IAM. Технологическая платформа Сбербанк»))».

4.1.7.2.1 Компонент «Авторизация ППРБ Platform V IAM («Модули IAM. Технологическая платформа Сбербанк»))»

Компонент «Авторизация ППРБ Platform V IAM («Модули IAM. Технологическая платформа Сбербанк»))» обеспечивает авторизацию доступа

пользователей к функционалу приложений на основе проверки прав пользователя, а именно:

- используя АРМ администратора, позволяет загрузить ролевую модель приложения из xml файла;
- используя АРІ получить списки прав пользователя;
- используя АРІ проверить наличие права у пользователя;

4.1.7.2.2 Компонент «Авторизация ЕФС Platform V IAM («Модули IAM. Технологическая платформа Сбербанка»)»

Компонент «Авторизация ЕФС Platform V IAM («Модули IAM. Технологическая платформа Сбербанка»)» обеспечивает авторизацию доступа пользователей к функционалу приложений на основе проверки ролей и прав доступа, а также на основе атрибутов субъектов и объектов доступа. Компонент предоставляет следующую функциональность:

- используя АРМ администратора: загрузить ролевую модель приложения из xml файла;
- экспортировать в файл модели авторизации;
- выполнить синхронизацию модели авторизации;
- отредактировать модель авторизации;
- получить список прав пользователя;
- проверить наличие права у пользователя;
- получить группу пользователей по коду категории пользователя.

4.1.7.3 Компонент «Аудит Platform V Audit» («Модуль Аудит. Технологическая платформа Сбербанка»)»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Аудит Platform V Audit («Модуль Аудит. Технологическая платформа Сбербанка»)».

Компонент «Аудит Platform V Audit («Модуль Аудит. Технологическая платформа Сбербанка»)» обеспечивает регистрацию, сбор и хранение событий аудита, а именно:

- возможность зарегистрировать событие аудита, используя API компонента аудита;
- возможность сбора и сохранения событий аудита в хранилищах компонента аудита от модулей, использующих API компонента аудита;
- временное хранение событий в транспортном модуле компонента аудита;
- хранение событий в долговременном и оперативном хранилище:
 - долговременное, надежное, защищенное хранение событий аудита сроком до пяти лет;
 - в оперативном хранилище обеспечивается возможность указать срок хранения данных (по умолчанию 90 суток). По истечении срока хранения данные автоматически удаляются из оперативного хранилища. Указание срока хранения в оперативном хранилище осуществляется при развертывании компонента аудита. Рекомендуемый срок хранения данных зависит от объема входящего потока данных и характеристик серверов, на которых размещается оперативное хранилище;
- возможность загрузки данных из долговременного хранилища в оперативное хранилище за произвольный период, по длительности не превышающий 30 суток;
- возможность потоковой передачи в сторону внешних систем событий, поступающих в компонент аудита по заданному фильтру. События аудита передаются в компонент обмена сообщениями в формате json в заданный топик. Фильтр обеспечивает возможность передачи событий аудита по точному соответствию пары значений название модуля и код события указанным в фильтре.
- поиск событий в оперативном хранилище, используя АРМ компонента аудита, за указанный пользователем период с возможностью указания дополнительных критериев поиска:
 - по названию модуля,
 - по логину учетной записи,

- по коду события,
- по значению, присутствующему в параметрах события (за исключением названий событий и дополнительных параметров)
- проверка развертывания транспортного модуля и хранилищ.

4.1.8 Интерфейсные компоненты

4.1.8.1 Компонент моделирования и исполнения бизнес-сценариев

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью следующих компонентов:

- «Моделирование и исполнение бизнес-сценариев Engine Platform V Flow»;
- «Моделирование и исполнение бизнес-сценариев Designer Platform V Flow»;
- «Моделирование и исполнение бизнес-сценариев Tasklist Platform V Flow».

Указанные компоненты Платформы обеспечивают инструментарий для проектирования и исполнения бизнес-процессов, в том числе с поддержкой пользовательских задач. Бизнес-процессы формулируются в нотации BPMN¹ и поддерживают вызовы сервисов в формате: Rest и JSON-RPC API.

4.1.8.1.1 Компонент «Моделирование и исполнение бизнес-сценариев Designer Platform V Flow»

Компонент «Моделирование и исполнение бизнес-сценариев Designer Platform V Flow» обеспечивает визуальное проектирование процессов в нотации BPMN¹, включая:

- взаимодействие спроектированных компонентов приложения в рамках сценария;
- проектирование и разработку в одной среде;

¹ список поддерживаемых элементов нотации: start event (none, message, timer, signal, error), end event (none, terminate, error), service task (JSON-RPC, Rest), user task, script task, manual task, call activity, subprocess (event, standard, multi-instance), gateway (exclusive, inclusive, parallel, event), intermediate catch event (none, message, timer, signal, error), boundary event (message, timer, signal, error), sequence flow, swim lane, receive task.

- настройки реакций на сбои при вызове внешних сервисов;
- механизм повторных вызовов.

4.1.8.1.2 Компонент «Моделирование и исполнение бизнес-сценариев Engine Platform V Flow»

Компонент «Моделирование и исполнение бизнес-сценариев Engine Platform V Flow» обеспечивает следующую функциональность:

- исполнение экземпляров бизнес-процессов, спроектированных в компоненте «Моделирование и исполнение бизнес-сценариев Designer Platform V Flow», с возможностью:
 - запуск на выполнение экземпляров бизнес-процессов;
 - обеспечения персистирования контекста экземпляра бизнес-процесса, в случаях перехода экземпляра бизнес-процесса в ожидание;
 - организации вызовов внешних сервисов, указанных в модели бизнес-процесса (API: Rest, JSON-RPC);
- управление исполнением экземпляров бизнес-процессов посредством административной панели управления: прерывание процесса, повтор сбойного шага процесса;
- управление исполнением бизнес-процессами посредством графического UI;
- версионирование моделей бизнес-процессов.

4.1.8.1.3 Компонент «Моделирование и исполнение бизнес-сценариев Tasklist Platform V Flow»

Компонент «Моделирование и исполнение бизнес-сценариев Tasklist Platform V Flow» обеспечивает следующую функциональность:

- исполнение экземпляров бизнес-процессов, спроектированных в компоненте «Моделирование и исполнение бизнес-сценариев Designer Platform V Flow», с возможностью:
 - работа с пользовательскими задачами: выполнить, назначить, взять в работу, отменить, изменить приоритет.

4.1.8.2 Компонент «Управление статическим и динамическим контентом Platform V Content Management»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Управление статическим и динамическим контентом Platform V Content Management».

Компонент «Управление статическим и динамическим контентом Platform V Content Management» обеспечивает следующую функциональность:

- предоставление текста и заголовка ошибки для обработки исключительных ситуаций в приложении в соответствии со справочником (Java-библиотека). Справочник ошибок описывается в ресурсах приложения в виде JSON-файла, содержащего сопоставления кодам ошибок, соответствующих текста и заголовка ошибки;

- предоставление пакета текстовых подсказок к элементам форм для отображения на UI в соответствии со справочником (Java-библиотека). Справочник подсказок описывается в ресурсах приложения в виде в JSON-файла, содержащего тексты подсказок в привязке к иерархии <код бизнес-процесса>-<код шага бизнес-процесса>-<код элемента шага бизнес-процесса>;

- маскирование «чувствительных» данных (Java-библиотека):

- маскирование «чувствительных» текстовых данных одного из следующих типов:

- фамилия;
- дата рождения;
- адрес и составляющие его части;
- номер документа;
- номер телефона;
- номер карты;
- номер банковского счета;
- логин пользователя;
- одноразовый пароль пользователя;

- под маскированием понимается замена всех или части символов символом «*»;
- при использовании функции маскирования тип текстовых данных указывается явно;
- библиотека позволяет менять настройки существующих правил маскирования в соответствии с документацией.

4.1.8.3 Компонент «Формирование печатных форм Platform V Print»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Формирование печатных форм Platform V Print».

Компонент «Формирование печатных форм Platform V Print» обеспечивает генерацию электронных печатных форм документов по размеченным шаблонам в форматах PDF, DOC, RTF, HTML, XLS, CSV, XML:

- формирование электронных документов на основе данных, получаемых из прикладной бизнес-области и передаваемых через формат API сервиса по сетевому протоколу HTTPS, выполняя POST-запросы. Размерность документов ограничена;
- ведение реестра шаблонов в формате компонента формирования печатных форм с поддержкой версионности;
- управление шаблонами печати:
 - импорт;
 - активация\деактивация версий шаблонов;
- генерацию документов по шаблонам с использованием инструмента JasperReports;
- генерацию документов по шаблонам с использованием инструмента: FreeMarker.

4.1.8.4 Компоненты управление сессиями

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонентов:

- Компонент «Сессионные данные Platform V User Sessions»;

– Компонент «Стартовый менеджер Platform V Starting Manager».

4.1.8.4.1 Компонент «Сессионные данные Platform V User Sessions»

Компонент «Сессионных данные Platform V User Sessions» выполняет следующие функции:

- обеспечивает доступ к данным пользовательских сессий, созданных компонентом «Стартовый менеджер Platform V Starting Manager»;
- непersistентное хранение данных пользовательских сессий в оперативном кэше приложений (размер данных не более 512 КБ; ограничения, налагаемые алгоритмом сериализации Java-объектов);
- хранение мастер-копии секций сессии;
- контроль времени жизни хранимых сессионных данных и принятие решения об их удалении из кэша;
- ведение реестра активных сессий, созданных компонентом «Стартовый менеджер Platform V Starting Manager», с доступом через API.

4.1.8.4.2 Компонент «Стартовый менеджер Platform V Starting Manager»

Компонент «Стартовый менеджер Platform V Starting Manager» на уровне фронтальной логики приложений обеспечивает:

- инициализацию сессионного контекста аутентифицированного пользователя и вычисление конфигурации фронтального приложения в соответствии с ролевой моделью в формате (маппинг операции/приложения на tenant code и привилегию, полученную от компонента «Авторизация ЕФС Platform V IAM («Модули IAM. Технологическая платформа Сбербанка»)»), а именно
 - точку входа в платформу аутентифицированных пользователей на основании данных, переданных провайдером аутентификации «Аутентификация Platform V IAM («Модули IAM. Технологическая платформа Сбербанка»)» по установленному с провайдером формату.
 - возвращение списка доступных для пользователя приложений и операций в базу данных Стартового менеджера и хранящихся в нем как сущности, на

основании связанных с ними permission, переданных из сервиса «Авторизация ЕФС Platform V IAM («Модули IAM. Технологическая платформа Сбербанка»));

- добавление, удаление и изменение списка доступных для конкретного permission системы авторизации набора приложений и операций;
- инициацию открытия пользовательской сессии.

4.1.8.5 Компонент «Центр уведомлений Platform V Notification Center»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Центр уведомлений Platform V Notification Center».

Компонент «Центр уведомлений Platform V Notification Center» отвечает за доставку пользователю уведомлений и иных видов сообщений, а именно обеспечивает:

- хранение данных о каналах уведомления пользователя и их настройках:
 - добавление параметров пользователя с указанием используемых каналов, email-адреса;
 - добавление нового регистрационного токена с привязкой к пользователю;
 - удаление регистрационного токена по внешнему запросу;
 - изменение настроек отдельных каналов уведомлений пользователя;
 - удаление параметров уведомлений пользователя;
- централизованную отправку сообщений пользователю:
 - отправку Push-уведомлений по всем токенам, зарегистрированным у пользователя;
 - удаление недействительных токенов из настроек по результатам отправки Push-уведомления;
 - отправку email-сообщения на зарегистрированный в Центре уведомлений адрес электронной почты пользователя;
 - отправку уведомления одновременно на два канала: Push, Email;
 - отправку email-сообщения, сформированного на основе шаблона;
- формирования содержания сообщения на основе шаблонов;

- хранение параметров подключения к конечным серверам (SMTP, FCM, APN).

4.1.8.6 Компонент «Интеграция с ЕСИА Platform V ESIA Gateway»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью следующих компонентов:

- «Интеграция с ЕСИА Platform V ESIA Gateway»;
- «Аутентификация Platform V IAM («Модули IAM. Технологическая платформа Сбербанка»));
- «Стартовый менеджер Platform V Starting Manager».

Компонент «Интеграция с ЕСИА Platform V ESIA Gateway» обеспечивает интеграцию прикладных приложений, создаваемых на базе Платформы, с "Единой Системой Идентификации и Аутентификации" (ЕСИА) с целью получения различных данных, а именно:

- получение данных о пользователе государственных услуг из ЕСИА по авторизационным данным:
 - получение данных по ФЛ;
 - получение статуса подтвержденности учётной записи;
 - получение списка организаций пользователя;
 - получение данных из профиля организации.

Компонент «Аутентификация Platform V IAM («Модули IAM. Технологическая платформа Сбербанка»))» обеспечивает на уровне фронтальной логики приложений:

- ограничение выполнения запросов не аутентифицированного приложения пользователя к прикладным компонентам;
- восстановление сессионного контекста аутентифицированного приложения пользователя по данным ЕСИА или провайдера аутентификации OpenId Connect для выполнения запроса в прикладном компоненте:
 - поддержка мобильных устройств – восстановление аутентификационной сессии IAM по ранее выданному им маркеру;

- восстановление сессионного контекста ЕСИА по ранее полученному из ЕСИА маркеру обновления;
- обеспечение возможности обращения к ЕСИА с использованием маркера доступа ЕСИА;
- оркестрацию фронтального приложения пользователя для входа в приложение по сценарию аутентификации ЕСИА и OpenID Connect
- аутентификацию пользователя с помощью ЕСИА или иного внешнего провайдера аутентификации;
- перенаправление запроса фронтального приложения идентифицированного пользователя на вычисленную в микросервисе маршрутизации ноду запрошенного прикладного микросервиса:
 - предоставление в маркере данных пользователя, необходимых для маршрутизации в бэк-енде.

Компонент «Стартовый менеджер Platform V Starting Manager» обеспечивает:

- инициализацию сессионного контекста аутентифицированного пользователя и вычисление конфигурации фронтального приложения в соответствии с ролевой моделью, в том числе:
 - инициализацию сессионного контекста пользователя данными ЕСИА или провайдера аутентификации OpenId Connect,
 - предоставление конфигурации модулей фронтального приложения в соответствии с авторизацией пользователя,
 - добавление, удаление и изменение конфигураций модулей фронтального приложения.

4.1.8.7 Компонент «Интеграция со СМЭВ Platform V SMEV Gateway»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Интеграция со СМЭВ Platform V SMEV Gateway».

Компонент «Интеграция со СМЭВ Platform V SMEV Gateway» обеспечивает интеграцию разрабатываемых платформенных приложений с Единой Системой Межведомственного Электронного Взаимодействия (СМЭВ), и позволяет:

- отправлять запросы на получение данных в сервисы, опубликованные в СМЭВ;
- возвращать статус запроса при вызове прикладным приложением;
- получать и сохранять ответы на запросы для дальнейшей передачи в прикладное приложение;
- получать и сохранять файловые вложения к запросам при их наличии;
- отправлять данные в сервисы, опубликованные в СМЭВ;
- получать примеры запросов на получение доступа к промышленному контуру СМЭВ с использованием квалифицированного сертификата;
- проводить очистку хранимых данных по расписанию.

4.1.8.8 Компонент «Журналирование Platform V Monitor»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Журналирование Platform V Monitor».

Компонент «Журналирование Platform V Monitor» обеспечивает:

- запись логов и их хранение в централизованном хранилище:
 - централизованное хранилище поддерживает репликацию, горизонтальное масштабирование и полнотекстовое индексирование собираемых журналов;
 - транспорт обеспечивает возможности буферизирования сообщений на случай сбоев в подсистеме хранения;
 - подсистема хранения обеспечивает репликацию для отказоустойчивости;
- логическое объединение логов, инициированных приложением в рамках одной бизнес-операции:
 - записи журналов маркируются идентификатором, который позволяет сгруппировать запись в рамках запроса (requestId заполняется автоматически);
 - для категоризации журналов по типам бизнес операций обеспечивается возможность разметить журналы специфичными идентификаторами бизнес операций (businessId заполняется в контексте журналирования потребителем сервиса);

- связь логов в цепочки вызовов приложений (и(или) микросервисов):
 - каждый новый запрос системы помечается идентификатором `traceId`. В рамках запроса выделяются шаги, которые помечаются своим `spanId`;
 - сервис трассировки поддерживает спецификацию `openZipkin`;
 - для отображения цепочек вызовов, используется `APM`;
- фильтрацию логов перед отправкой в хранилище:
 - записи журналов фильтруются по уровню события и классу, порождающему запись. (например, `com.somethingpackage.ClassName:ERROR`);
 - фильтрация осуществляется в рамках элемента инсталляции подсистемы, т.е. на уровне конкретного приложения потребителя сервиса журналирования;
- обогащение записей журналов дополнительной информацией:
 - записи журналов поддерживают обогащение в двух вариантах: автоматическом и ручном режимах. Автоматический режим обогащает журналы статическими данными (`ip` адрес приложения, `hostname`, идентификатор пода, код приложения). Ручной режим обогащения поддерживает обогащение журналов дополнительной информацией (`key:value String:String`) через контекст журналирования;
 - все записи журналов содержат время события;
- инструментарий просмотра журналов и поиска в них событий:
 - административное рабочее место поддерживает поиск по телу сообщения. В фильтре поиска обязательно должно быть заполнено значение времени, за которое осуществляется поиск записей;
 - `APM` поддерживает поиск по дополнительному атрибутивному составу (поиск по дополнительным атрибутам, например, идентификатор запроса, `hostname`, идентификатор операции);
 - `APM` поддерживает возможность сортировки записей;
 - `APM` поддерживает аутентификацию/авторизацию для исключения несанкционированного доступа;

– выгрузку журналов для дальнейшего анализа в текстовый файл с учетом указанных значений в фильтре поиска используя АРМ.

4.1.8.9 Компонент «Platform V One-Time-Token»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Platform V One-Time-Token».

Компонент «Platform V One-Time-Token» обеспечивает аутентификацию и авторизацию межсервисных вызовов для компонентов Платформы с использованием механизма токенов, а именно:

– идентификацию и аутентификацию сервиса-отправителя с использованием сертификата сервиса-отправителя и аутентификационного токена, подписанного закрытым ключом сервиса-отправителя;

– управление списками контроля доступа с помощью XACML политик авторизации:

– загрузка набора XACML политик авторизации;

– обновление XACML-файлов политик авторизации без прерывания сервиса;

– авторизацию межсервисного вызова:

– проверку межсервисного вызова по спискам контроля доступа;

– выдачу авторизационного токена, используемого для авторизации межсервисного вызова;

– регистрацию событий безопасности, происходящих в процессе аутентификации и авторизации межсервисных вызовов:

– успешная выдача токена;

– ошибка аутентификации;

– ошибка авторизации.

4.1.8.10 Компонент «Электронная подпись Platform V Secret Management»

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Электронная подпись Platform V Secret Management».

Компонент «Электронная подпись Platform V Secret Management» обеспечивает формирование и проверку электронной подписи для использования в разрабатываемых приложениях:

– подписание/проверка подписи в формате СМЭВ 2 с использованием КриптоПро JCP и JTLS (2.0 и выше, сертифицированный ФСБ, для JVM версии 10 и выше) или эквивалента, обеспечивающего функционал подписания электронной подписью:

- подписание СМЭВ-2 документа без вложений подписью ЭП-СП;
- подписание СМЭВ-2 документа без вложений подписью ЭП-ОВ;
- проверка подписи сотрудника органа власти (ЭП-СП) в документе СМЭВ- 2:
 - проверка соответствия подписи сертификату;
 - проверка статуса отзыва сертификата;
 - проверка цепочки сертификатов;
- проверки подписи ИС органа власти (ЭП-ОВ) в документе СМЭВ-2:
 - проверка соответствия подписи сертификату;
 - проверка статуса отзыва сертификата;
 - проверка цепочки сертификатов;
- ГОСТ формирования подписи для СМЭВ-2

– подписание/проверка подписи в формате СМЭВ 3 с использованием КриптоПро JCP и JTLS (сертифицированный ФСБ) или эквивалента, обеспечивающего функционал подписания электронной подписью:

- подписание СМЭВ-3 документа без вложений подписью ЭП-СП;
- подписание СМЭВ-3 документа без вложений подписью ЭП-ОВ;
- проверка подписи сотрудника органа власти (ЭП-СП) в документе СМЭВ-3:

- проверка соответствия подписи сертификату;
- проверка статуса отзыва сертификата;
- проверка цепочки сертификатов;
- проверки подписи ИС органа власти (ЭП-ОВ) в документе СМЭВ-3:
 - проверка соответствия подписи сертификату;
 - проверка статуса отзыва сертификата;
 - проверка цепочки сертификатов;
- ГОСТ формирования подписи для СМЭВ-3 - ГОСТ Р 34.10-2012.

4.1.8.11 Компонент «Управление настройками приложений Platform V Configuration».

Соответствие Платформы предъявляемым требованиям обеспечено функциональностью компонента «Управление настройками приложений Platform V Configuration».

Компонент «Управление настройками приложений Platform V Configuration» обеспечивает хранение, управление и предоставление по запросу параметров конфигурации библиотек, сервисов Платформы и прикладных приложений, разработанных на Платформе, а именно:

- хранение всех настроек в едином хранилище;
- возможность, используя АРМ администратора:
 - просмотреть список параметров и конфигураций (совокупный набор параметров, определенных областью сервиса или приложения потребителя);
 - найти параметр или конфигурацию по следующим критериям: идентификатор параметра или конфигурации, идентификатор потребителя, значение, тип параметра;
 - выполнить загрузку модели конфигурации из файла формата .xml;
 - выполнить загрузку настроек конфигурации из файла *.properties;
 - поддерживать создание и хранение следующих типов параметров: Строка, Число, Дата, Логический;
 - создать новый параметр;

- просмотреть значения параметра;
- редактировать значения параметров;
- удалить значение параметра;
- просмотреть историю изменения значения параметра;
- удалить параметр;
- поддерживать несколько версий конфигурации;
- экспортировать список параметров в файл в формате json;
- экспортировать настройки конфигурации в файл *.properties и *.yaml;
- переопределить значения параметров и конфигураций;
- сравнить параметры между разными хранилищами настроек;
- реплицировать изменения параметров между хранилищами настроек;
- создать запрос на подтверждение второй рукой изменений значения параметров конфигурации;
- подтвердить второй рукой изменения значения параметров конфигурации;
- отклонить второй рукой изменения значения параметров конфигурации;
- удалить конфигурацию;
- применить изменения значений параметров и конфигурации в runtime-режиме;
- восстановить значение параметра;
- возможность загрузить параметры и конфигурации в хранилище настроек без использования UI:
 - через REST API;
 - через CLI;
 - для загрузки использовать формат файла json, xml;
- возможность, получить значения параметров и конфигураций:
 - через JAVA API получения параметров;
 - через JAVA API получения конфигураций;
 - через REST API получения параметров;
 - через REST API получения конфигураций;

– журналирование и аудит действий по изменению настроек. События аудита содержат следующую информацию:

- дату изменения;
- время изменения;
- автор изменений;
- старое значение;
- новое значение.

4.1.8.12 Компоненты прикладного мониторинга

4.1.8.12.1 Компонент «Прикладной мониторинг Platform V Monitor»

Соответствие Платформы предъявляемым требованиям обеспечено реализацией требуемого функционала «Прикладной мониторинг Platform V Monitor».

Компонент «Прикладной мониторинг Platform V Monitor» обеспечивает:

- возможность использовать любую реализацию API для установки точек мониторинга, если метрики выставляются в формате Prometheus по заданному URL в прикладном приложении;
- возможность собирать метрики в формате Prometheus по заданному URL в прикладном приложении;
- возможность интеграции с Prometheus и получения от него инфраструктурных метрик контейнеров, принадлежащие определенному проекту;
- возможность визуализации полученных метрик в виде таблиц, графиков в интерфейсе администратора системы мониторинга.

4.1.8.12.2 Компонент «Хранение метрик Abyss Platform V Monitor»

Компонент хранения метрик Платформы обеспечивает:

- хранение метрик с возможностью изменения сроков хранения данных.

4.1.8.13 Компонент «Ведение электронного документооборота Platform V Documents»

Компонент «Ведение электронного документооборота Platform V Documents» обеспечивает работу с объектами документа и выполнение базовых операций документооборота, а именно:

- формирование каталога форм документов;
- предоставление инструмента для создания, настройки и сохранения шаблонов документов (с помощью DOCX, JRXML);
- предоставление инструмента для создания документов по шаблонам форм;
- обеспечение работы с документами в различных форматах (PDF, RTF, DOC и т. д.);
- возможность просмотра шаблонов в пользовательском интерфейсе;
- передача документов по маршруту на согласование/утверждение;
- создание новых процессов согласования документов;
- запуск процесса по карточке документов для дальнейшего согласования/утверждения;
- согласование документов;
- подписание документов с помощью ЭП;
- получение (поиск) карточек документов по запросу по следующим фильтрам:
 - название, вид деятельности;
 - автор;
 - дата создания до/дата создания после;
- просмотр истории процесса, то есть информации по выполненным шагам текущего процесса:
 - наименование шага процесса;
 - ФИО пользователя, комментарий;
 - дата и время выполнения задачи;
- поддержка API для создания карточки документа и запуска процесса по ней (без использования пользовательского интерфейса).

