ОБЛАЧНАЯ ПЛАТФОРМА ВТБ

Описание продукта

Содержание

1 Введение

Документ содержит описание назначения, архитектуры и функциональных возможностей программного продукта «Облачная платформа ВТБ».

Содержание разделов документа структурировано с учетом основных модулей, входящих в продукт.

1.1 Термины и определения

Раздел содержит определения основных терминов, используемых в настоящем документе.

Термин	Определение
HAProxy	Серверное программное обеспечение для обеспечения высокой доступности и балансировки нагрузки для TCP и HTTP-приложений посредством распределения входящих запросов на несколько обслуживающих серверов
Keycloak	Продукт с открытым кодом для поддержки технологии единого входа (single sign-on) с возможностью управления доступом
Kong	Продукт с открытым исходным кодом, реализующий шлюз API для мультиоблачных и гибридных систем
REST API	Архитектурный стиль взаимодействия компонентов распределенного приложения в сети. REST представляет собой согласованный набор ограничений, учитываемых при проектировании распределенной гипермедиа-системы
Гипервизор	Диспетчер виртуальных машин; процесс, обеспечивающий создание и выполнение виртуальных машин. С помощью гипервизора один компьютер узла может поддерживать несколько гостевых виртуальных машин за счет предоставления виртуального общего доступа к своим ресурсам, например к памяти и процессорам
Микросервис	Микросервис – сервис, входящий в состав единого приложения, состоящего из набора небольших сервисов, каждый из которых работает в собственном процессе и коммуницирует с остальными используя легковесные механизмы, как правило HTTP. Эти сервисы построены вокруг бизнес-потребностей и развертываются независимо с использованием полностью автоматизированной среды. Существует абсолютный минимум централизованного управления этими сервисами. Сами по себе эти сервисы могут быть написаны на разных языках и использовать разные технологии хранения данных
Публичное (общедоступное) облако	Модель развертывания облака, в рамках которой вычислительные службы и инфраструктура, предоставляемые по требованию сторонним поставщиком услуг, используются несколькими организациями через публичную сеть интернет

Папка/подпапка	Организационная единица (бизнес-блок, департамент, отдел и т. д.) или группа пользователей из разных подразделений, объединенная, например, в рамках одного проекта
Продукт	Функциональное инфраструктурное решение, настроенное по определенным стандартам. Является комплексом технических средств, объединенных в одну сущность по функциональному признаку
Проект	Информационная система с выбранной для нее средой
Частное облако	Модель развертывания облака по требованию, в рамках которой услуги и инфраструктура облачных вычислений размещаются в частной среде (во внутренней сети или ЦОД компании) и используют собственные ресурсы организации, в то время как другие организации не имеют к ним доступа

1.2 Сокращения

Раздел содержит расшифровку сокращений, используемых в настоящем документе.

Сокращение	Определение
AD	Active Directory (служба каталогов)
DMZ	Demilitarized Zone (демилитаризованная зона)
DNS	Domain Name System (служба доменных имен)
HTTP	Hypertext Transfer Protocol (протокол передачи данных)
HTTPS	Hypertext Transfer Protocol Secure (защищенный протокол передачи данных)
IaaS	Infrastructure as a Service (инфраструктура как услуга)
IAM	Identity and Access Management (система управления доступом)
IPAM	IP Address Management (система управления адресным пространством)
PaaS	Platform as a Service (платформа как услуга)
SDN	Software-defined network (программно-определяемая сеть)
SaaS	Software as a Service (программное обеспечение как услуга)
SSO	Single Sign-On (технология единого входа)

TCO	Total Cost of Ownership (совокупная стоимость владения)
VM	Virtual machine (виртуальная машина)
БД	База данных
ИС	Информационная система
ОС	Операционная система
СУБД	Система управления базами данных
ЦОД	Центр обработки данных

2 Назначение

Программный продукт «Облачная платформа ВТБ» – единое окно управления вычислительной/виртуальной инфраструктурой, платформа для предоставления продуктов в облачной инфраструктуре с использованием моделей Iaas, PaaS и SaaS.

2.1 Функции

Основные функции:

- предоставление инфраструктурных продуктов пользователям в режиме самообслуживания;
- быстрое развертывание и масштабирование вычислительной инфраструктуры;
- аналитика данных и прогнозирование оптимального использования оборудования;
- предоставление продуктов на условно-платной основе:
  - предоставление информации о текущем денежном балансе пользователя;
  - возможность предварительного расчета стоимости планируемых ресурсов;
  - расчет стоимости использованных ресурсов, списание средств.

2.2 Принципы построения

Основные принципы построения:

- единая архитектура в каждом контуре;
- обеспечение всех потребностей в инфраструктуре;
- повышение надежности и доступности, снижение рисков;
- быстрая скорость развертывания и масштабирования ресурсов;
- повышение утилизации серверного оборудования;
- прозрачность и поминутная тарификация потребленных инфраструктурных продуктов;
- снижение общей стоимости владения (TCO) IT-инфраструктуры.

2.3 Основные компоненты

Основные компоненты продукта:

- портал самообслуживания для заказа облачных продуктов с возможностью автоматизации развертывания инфраструктуры с использованием REST API:
  - административный и пользовательский веб-интерфейс;
  - REST API;

  - биллинг;
  - аналитика;
- оркестратор, обеспечивающий автоматизацию развертывания инфраструктурных продуктов на ресурсах под управлением портала:
  - продуктовый каталог;
  - ядро оркестратора;
- инфраструктура для поддержки различных групп ресурсов, предоставления виртуальных и физических серверов:
  - OpenStack VM (Проект с открытым исходным кодом по разработке платформы, позволяющей строить частные и публичные Облака (cloud computing));
  - Bare metal (Услуга предоставления вычислительных ресурсов в виде отдельных серверов x86 без установленного системного программного обеспечения);
  - Kubernetes (Платформа оркестрации контейнеров с открытым исходным кодом).

2.4 Группы продуктов

Группы продуктов в облачной инфраструктуре:

- Группа 1 – виртуальные машины x86 (IaaS/IaaS+): виртуальные машины на общем оборудовании с операционной системой и стандартным технологическим стеком, необходимым для приложений;
- Группа 2 – контейнеры (PaaS): небольшие виртуальные среды на операционной системе с использованием платформы Kubernetes;
- Группа 3 – СУБД (PaaS): готовая инфраструктура кластеров и серверов СУБД для различных задач;
- Группа 4 – физические серверы: выделенные серверы в облачной среде с собственной копией операционной системы.

3 Функциональные возможности

Облачная платформа ВТБ состоит из нескольких модулей, реализующих определенную функциональность программного продукта. Раздел содержит описание функциональности модулей и сервисов, входящих в состав продукта.

Продукт «Облачная платформа ВТБ» включает в себя модули, перечисленные в Табл. 1

Табл. 1. Компоненты продукта «Облачная платформа ВТБ»

Название модуля	Описание модуля
Портал облачной платформы ВТБ (1482)	Подсистема заказа облачных продуктов с возможностью автоматизации развертывания инфраструктуры с использованием REST API: • административный и пользовательский веб-интерфейс; • REST API; • биллинг; • аналитика
Универсальная реляционная СУБД (1985)	Модуль управления прикладными базами данных с использованием реляционной модели хранения данных
Кэширующая СУБД (2003)	Модуль организации и управления резидентными базами данных, используемыми для кэширования «горячих», не требующих резервирования

Название модуля	Описание модуля
Распределенная колоночная СУБД (1997)	Модуль управления прикладными базами данных с использованием нереляционной, колоночной модели хранения данных для транзакционных нагрузок (OLTP)
Колоночная аналитическая СУБД (1992)	Модуль управления прикладными базами данных с использованием нереляционной, колоночной модели хранения данных для аналитических нагрузок (OLAP)
СУБД полнотекстового поиска (1991)	Модуль управления прикладными ИТ-системами для сбора и хранения данных с возможностью полнотекстового поиска и анализа данных

3.1 Портал Облачная платформа ВТБ (1482)

Модуль «Портал Облачная платформа ВТБ» (далее – Портал) является основным компонентом программного продукта «Облачная платформа ВТБ», реализующим функции управления вычислительной/виртуальной инфраструктурой.

Перечень основных функций, доступных пользователю Портала:

- просмотр каталога компонентов (сервисов, библиотек, шаблонов);
- просмотр информации по отдельному компоненту;
- создание сервисов;
- импорт компонентов из BitBucket Server (GIT).

При создании или импорте компонента производится генерация уведомлений пользователей Портала о новом компоненте и его включение в поиск по каталогу компонентов.

Получение инфраструктурных продуктов

Веб-интерфейс Портала предоставляет возможность пользователям получать инфраструктурные продукты в режиме самообслуживания.

В зависимости от роли пользователю могут быть доступны следующие группы операций (см. подробное описание в документах «Руководство пользователя» и «Руководство администратора»):

- авторизация пользователя;
- выбор контекста – выбор организации папки или проекта;
- управление доступными пользователю ресурсами (продуктами);
- создание заказа на предоставление продукта;
- управление заказами, изменение конфигурации созданного заказа;
- управление SSH-ключами;
- управление организационной структурой;
- создание папок и проектов;
- управление папками и проектами;
- управление правами пользователей папки/проекта;
- управление счетами.

Управление основными сущностями

Портал предоставляет REST API-интерфейс для работы со следующими сущностями:

- Области – объединяют в себе группы и пользователей;
- Группы – объединяют в себе стенды и пользователей; в рамках иерархии принадлежат области;
- Стенды – объединяют в себе заказы, хранящиеся в сервисе создания заказов; в рамках иерархии принадлежат группе;
- Информационные системы – информационные системы из реестра; объединяют в себе среды;
- Среды и префиксы сред – определяют контекст стенда.

Также Портал хранит информацию о пользователях. Данные о пользователях динамически загружаются и сохраняются из Keycloak при первой аутентификации. Пользователи могут состоять в различных группах/областях, создавать стенды.

Управление инфраструктурными продуктами (продуктовый каталог)

Продуктовый каталог в составе Портала обеспечивает хранение графов развертывания инфраструктурных продуктов, предоставляет данные для формирования форм заказа продуктов в веб-интерфейсе и выполняет валидацию параметров заказа.

Под продуктом понимается функциональное инфраструктурное решение, настроенное по определенным стандартам. Продукт является комплексом технических средств, объединенных в одну сущность по функциональному признаку.

Продукты могут быть простыми и составными. Простой продукт – виртуальная машина с системным программным обеспечением, проект системы запуска контейнеров, объектное хранилище. Составной продукт – кластерные решения, технологические стеки.

Создание заказов

Сервис создания заказов предоставляет REST API-интерфейс для создания заказа на получение инфраструктурного продукта, просмотр списка заказов и информации об отдельном заказе, а также получения данных о продуктах.

Помимо этого, сервис дополняет параметры заказа, которые необходимы для корректного развертывания продукта в оркестраторе. Сформированный набор параметров отправляется через сообщение в Брокер сообщений.

При создании заказа сервис выполняет необходимый набор проверок его параметров:

- доступность ресурсов;
- баланс группы/области;
- корректность параметров запроса;
- корректность заполнения в соответствии с JSON-схемой параметров продукта;
- проверка ограничений на заказ.

Ресурсное планирование

Сервис ресурсного планирования выполняет следующие функции:

- хранение информации о ресурсных пулах и их группировка по принципу однотипности и взаимозаменяемости; под ресурсным пулом понимается некоторая измеряемая емкость;
- хранение информации о существующих сетях, их свойствах и емкости;
- выдача рекомендации по размещению заказываемых продуктов в соответствии со стратегиями размещения и с учетом потребностей пользователя;
- учет ресурсов, предоставление информации для аналитики с целью прогнозирования утилизации;
- регулирование уровня выдачи ресурсов в соответствии с их фактическим объемом;
- единая точка контроля в ресурсном планировании – рекомендации формируются с учетом уже начатых процессов развертывания, а не только фактической информации о занятых объемах.

Контроль доступа

Сервис авторизации управляет контролем доступа по формуле: кто (личность) имеет какой доступ (роль) к какому ресурсу.

Ресурсами Облачной платформы могут быть, например, отдельные экземпляры виртуальных машин, базы данных, облачные диски или приложения. Также ресурсами являются организации, папки и проекты, которые используются на верхнем уровне в иерархии ресурсов.

Сервис состоит из трех компонентов:

- Resource Management – управление ресурсами каждого сервиса;
- Permission Management – управление разрешениями пользователей в отношении указанных ресурсов;
- Policy Enforcement – проверка прав доступа пользователя к указанным ресурсам.

Хранение информации о действиях и событиях в рамках заказов

Сервис состояний обеспечивает централизованное хранение информации о пользовательских объектах, выданных через Облачную платформу.

Сервис предоставляет REST API-интерфейс для сохранения информации о действиях (action) и событиях (event) в рамках заказа. Под событием понимается, например, изменение статуса ON/OFF/reboot/deleted, изменение конфигурации и т. д.

Данные по действиям и событиям рассылаются через Брокер сообщений и могут быть получены через REST API. Информация о них используется при отображении заказов, в биллинге и иных сервисах.

Оркестрация

Сервис оркестрации обеспечивает выполнение заказов на продукты в соответствии с графами развертывания, заданными для них в продуктовом каталоге.

Процесс оркестрации выполнения заказа включает следующие шаги:

Из продуктового каталога (запросы проксирует сервис создания заказов для проверки прав доступа) пользователь на портале получает список возможных для выполнения графов. При переходе на страницу продукта происходит получение формы с пользовательскими полями графа (JSON-схема). После заполнения форма отправляется в сервис создания заказов (Order Service).

Сервис создания заказов проверяет права пользователя на данный граф, резервирует место в ресурсных пулах / на хостах, заполняет системные поля графа (такие как хосты и ресурсные пулы), проверяет баланс пользователя в биллинге.

Далее начинается выполнение графа – выполняются предусмотренные им действия (action) в рамках данного заказа (order). Если это самое первое действие в заказе, заказ создается в сервисе состояний (State Service).

Для отображения хода выполнения заказа пользователю в веб-интерфейсе сервис оркестрации отслеживает изменения статуса заказа в сервисе состояний.

Данные дополняются заданными в шаблоне полями, идентификатором заказа (order_id), идентификатором действия (action_id), идентификатором графа (graph_id) и отправляются в оркестратор через RabbitMQ.

По идентификатору графа оркестратор проверяет свой кэш: если графа в нем нет, оркестратор получает его из продуктового каталога и выполняет кэширование. Выполняется получение данных по данному действию (action) из сервиса состояний (восстановление работы после сбоя оркестратора – если граф уже выполнялся, выполнение продолжается с шага, на котором произошел сбой). Затем оркестратор выполняет граф, вызывая описанные в нем RPC-модули и дополняя их ответами начальные данные. Также модули отправляют свои ответы (action) и события по сущностям платформы (event) в сервис состояний. В случае ошибки начинается операция отката графа. В ходе

выполнения оркестратор фиксирует изменения в сервисе состояний, который обеспечивает рассылку данных о них другим сервисам (обратная связь с Order Service, биллингом и т. д.). Одновременно по заказу может выполняться только один граф (для предотвращения взаимоисключающих операций).

Тарификация ресурсов

Сервис тарификации ресурсов выполняет формирование стоимости продукта.

Основные функции сервиса:

- тарификация продукта по параметрам;
- поддержка тарифных планов (в разрезе областей);
- поддержка тарифных классов (по ресурсам – например, CPU/RAM и т. д.);
- получение данных по обновлению параметров / состояния ресурса;
- отправка обновленной стоимости ресурса для списания средств.

Сервис предоставляет REST API-интерфейс для тарификации ресурсов, расчета стоимости продукта/заказа.

Также он хранит данные по тарифным планам областей, которые создаются из портала (сервис Portal). Тарифные классы создаются и редактируются через интерфейс администрирования данного сервиса.

Сервис онлайн-мероприятий (2146)

«Сервис онлайн-мероприятий» – основной компонент продукта «Облачная платформа», реализующий функции проведения онлайн-мероприятия с присутствием удаленных зрителей для организаторов внутрикорпоративных активностей, специалистов по событийному маркетингу, и продюсеров онлайн-обучения.

Основные функции сервиса

- Работа с большим количеством зрителей (до 20 тыс. человек);
- Различные форматы регистрации на мероприятие (открытое / закрытое / по списку);
- Инструменты управления вниманием (чат, реакции, ветки сообщений / вопросы спикерам, адресные вопросы / опросы / нотификации / модерация);
- Автоматическое создание лендинга мероприятия.

Архитектура сервиса

Архитектура сервиса представлена на следующем рисунке:

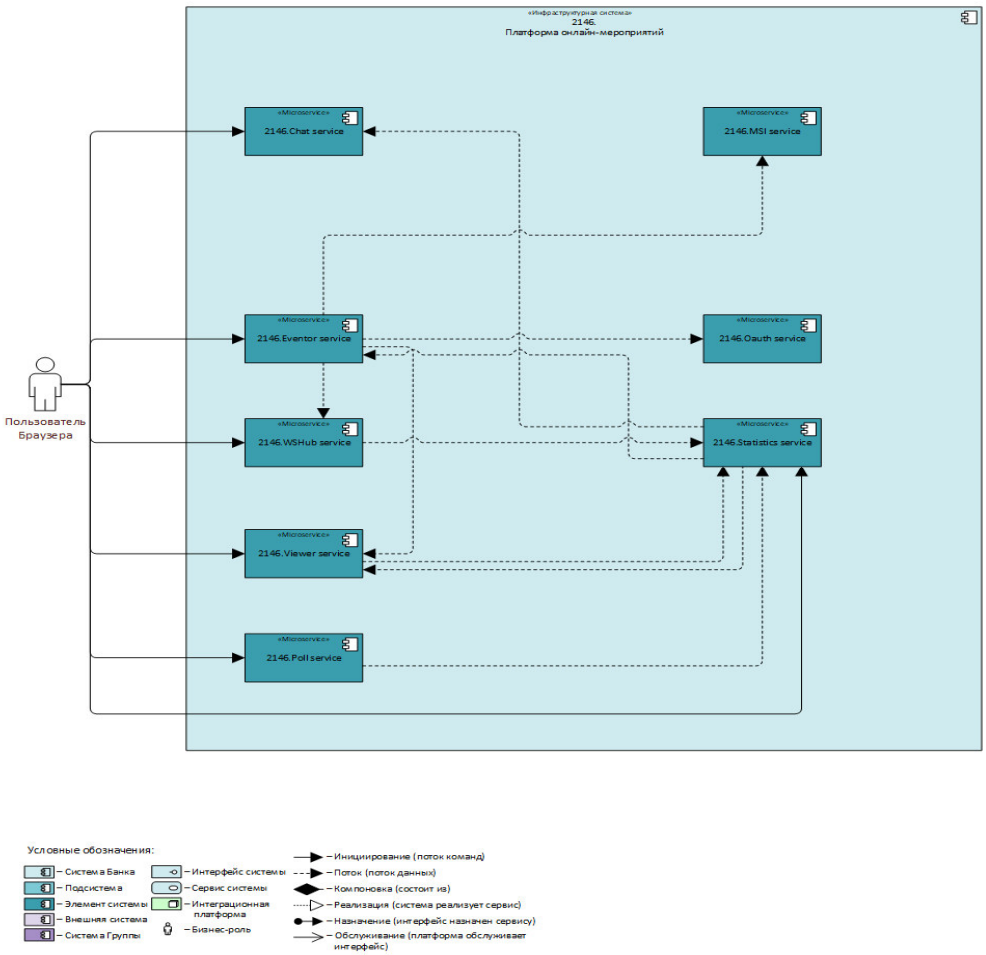

Рис. 1. Архитектура сервиса

Перечень микросервисов архитектуры представлен в Табл. 2.

Табл. 2. Микросервисы архитектуры

№	Наименование	Назначение	Хранимые данные (опционально)
1.	2146. Chat service	Микросервис чата отвечает за взаимодействие зрителей в коммуникационном блоке зрительского интерфейса и в интерфейсе модератора административного интерфейса.	Вкладка “Чат” зрительского интерфейса Вкладка “Вопросы” зрительского интерфейса Таб “Модерация” административного интерфейса Страница “Вопросы спикеру”
2.	2146. Poll Service	Микросервис опросов отвечает за управление блоком опросов зрительского интерфейса и интерфейса управления опросами в административном интерфейсе.	Вкладка “Опросы” зрительского интерфейса Таб “Опросы” административного интерфейса

№	Наименование	Назначение	Хранимые данные (опционально)
			Страница “Результаты опросов”
3.	2146. Eventor Service	Основной микросервис, отвечает за управление мероприятиями и расписанием.	Фрейм зрительского интерфейса Блок расписания зрительского интерфейса Блок настройки мероприятия административного интерфейса Блок настройки зала административного интерфейса Таб “расписание” административного интерфейса Таб “мониторинг” административного интерфейса
4.	2146. WSHub Service	Микросервис отвечает за рассылку зрителям широковещательных сообщений, связанных с изменением состояния мероприятия (старт, завершение, отмена запуска, изменение настроек).	
5.	2146. MSI Service	Микросервис отвечает за интеграцию микросервисной части с медиаподсистемой.	
6.	2146. Oauth Service	Микросервис отвечает за управление доступом организаторов к административному интерфейсу.	
7.	2146. Viewer Service	Микросервсис отвечает за регистрацию зрителей в зрительском интерфейсе.	Регистрационная форма зрительского интерфейса Блок рассылки регистрационных писем зрителям
8.	2146. Statistic Service	Микросервис отвечает за сбор статистики по взаимодействию зрителей и зрительского интерфейса, а также за подготовку статистических	Модуль сбора статистики по действиям зрителя

№	Наименование	Назначение	Хранимые данные (опционально)
		отчетов по мероприятию для организатора.	Таб “Статистика” административного интерфейса

3.2 Универсальная реляционная СУБД (1985)

Модуль «Универсальная реляционная СУБД» (1985) (далее - Универсальная реляционная СУБД) разработан на основе PostgreSQL и предоставляет пользователю возможность управления прикладными базами данных с использованием реляционной модели хранения данных. Универсальная реляционная СУБД предоставляется пользователю в виде готовой услуги в режиме самообслуживания на Портале.

Описание предоставляемых модулем функциональных возможностей приведено в Табл. 3

Табл. 3. Описание функциональных возможностей

Функция	Описание
Управление БД	Обеспечивает хранение и обработку данных в таблицах и связей между ними, обеспечивает универсальные способы обработки как транзакционных (OLTP), так и аналитических (OLAP) запросов
Автоматическое развертывание и масштабирование БД	Автоматическое развертывание и масштабирование СУБД осуществляется средствами «Облачной платформы ВТБ» с помощью входящих в ее состав специализированных скриптов
Мониторинг	Предоставляет возможности мониторинга стандартных и расширенных метрик состояния оборудования и ПО, входящего в состав СУБД, и передачу их в систему мониторинга

Архитектура «Универсальной реляционной СУБД» представлена на следующем рисунке.

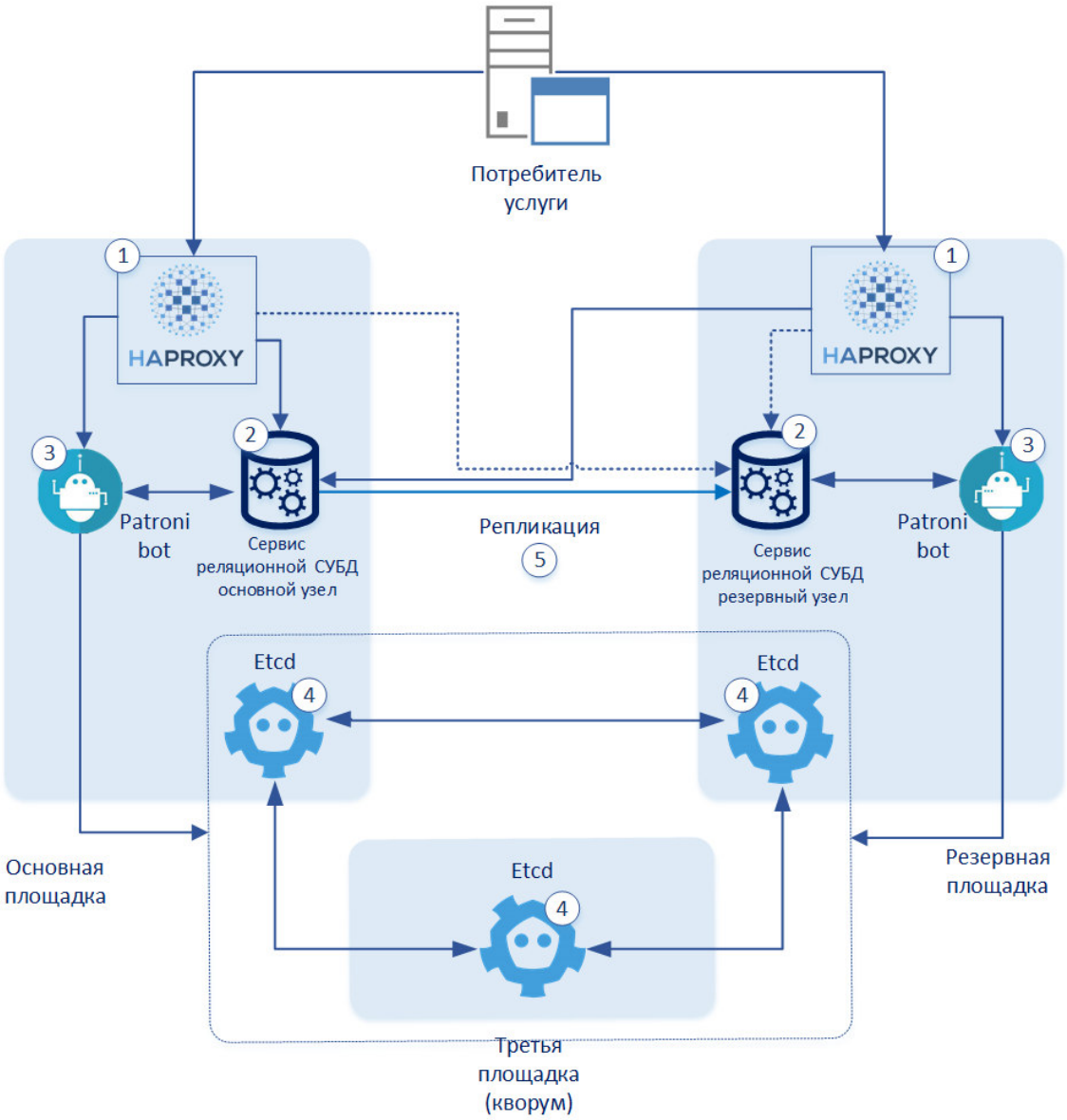

Рис. 2. Архитектура «Универсальной реляционной СУБД»

Архитектура СУБД состоит из следующих основных компонентов:

1. Haproxy – балансировщик сетевой нагрузки, обеспечивает балансировку сетевых подключений потребителей услуги между основным и резервным узлом кластера реляционной СУБД по заданным правилам;
2. Сервис реляционной СУБД – непосредственно сервисы СУБД для управления базами данных, предоставляются в виде преднастроенного кластера, состоящего из двух узлов: основного и резервного. Кластер развертывается автоматически в соответствии с запросом потребителя услуги и заданным им параметрами;
3. Patroni – инструмент управления кластером реляционной СУБД, обеспечивает автоматизацию процессов обслуживания баз данных и переключения ролей кластера;
4. кластер Etcd - представляет собой распределенной отказоустойчивое хранилище данных для Patroni, обеспечивает хранение конфигурационной баз данных и кластера реляционной СУБД;
5. репликация баз данных между основной и резервной площадкой, обеспечивает защиту данных в случае выхода одного из узлов кластера из строя.

3.3 Кэширующая СУБД (2003)

Модуль «Кэширующая СУБД» (2003) (далее - Кэширующая СУБД) разработан на основе Redis и предоставляет пользователю возможность организации и управления резидентными БД, используемыми для кэширования «горячих» данных, не требующих резервирования. Кэширующая СУБД предоставляется пользователю в виде готовой услуги в режиме самообслуживания на Портале.

Описание предоставляемых модулем функциональных возможностей приведено в Табл. 4.

Табл. 4. Описание функциональных возможностей

Функция	Описание
Управление БД	Организация и управление резидентными БД, используемыми для кэширования «горячих» данных прикладных ИТ-систем, не требующих резервирования
Транзакционная нагрузка (OLTP)	Поддерживает любые типы информации, которые могут быть обработаны транзакционной системой (OLTP) в реальном времени, при которой система работает с небольшими по размеру транзакциями, идущими большим потоком
Автоматическое развертывание и масштабирование БД	Автоматическое развертывание и масштабирование СУБД осуществляется средствами «Облачной платформы ВТБ» с помощью входящих в ее состав специализированных скриптов
Мониторинг	Предоставляет возможности мониторинга стандартных и расширенных метрик состояния оборудования и ПО, входящего в состав СУБД, и передачу их в систему мониторинга

Верхнеуровневая архитектура «Кэширующей СУБД» представлена на следующем рисунке.

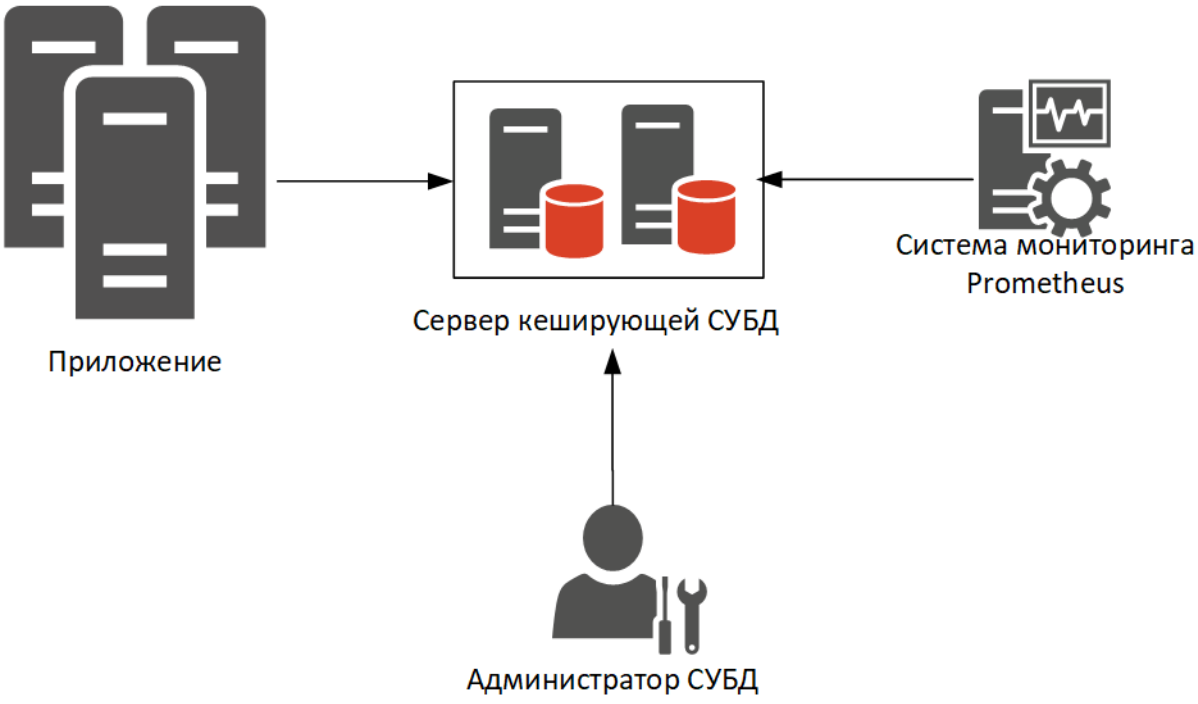

Рис. 3. Архитектура «Кэширующей СУБД»

Описание взаимодействие компонентов СУБД приведено в Табл.5.

Табл.5. Взаимодействие компонентов СУБД

№	Инициатор	Получатель	Порт	Описание
1	Приложение	Сервер кэширующей СУБД	6379 (TCP)	Приложение читает/записывает данные в БД
2	Администратор СУБД	Сервер кэширующей СУБД	22 (TCP)	Администратор СУБД производит первичную настройку сервера через SSH и осуществляет дальнейшее его администрирование
3	Администратор СУБД	Сервер кэширующей СУБД	6379 (TCP)	Администратор СУБД производит проверку на доступность приложения СУБД
4	Администратор СУБД	Сервер кэширующей СУБД	9121 (TCP)	Администратор СУБД производит проверку на доступность метрик СУБД
5	Prometheus	Сервер кэширующей СУБД	9121 (TCP)	Мониторинг Prometheus собирает метрики СУБД для осуществления контроля СУБД и дальнейшего оповещения о различных событиях в базе данных

3.4 Распределенная колоночная СУБД (1997)

Модуль «Распределенная колоночная СУБД» (1997) (далее - Распределенная колоночная СУБД) разработан на основе ScyllaDB и предоставляет пользователю возможность управления прикладными БД с использованием нереляционной, колоночной модели хранения данных для транзакционных нагрузок (OLTP). Распределенная колоночная СУБД предоставляется пользователю в виде готовой услуги в режиме самообслуживания на Портале.

Описание предоставляемых модулем функциональных возможностей приведено в Табл. 6

Табл. 6. Описание функциональных возможностей

Функция	Описание
Управление БД	Обеспечивает хранение и обработку данных в нереляционной, колоночной модели хранения данных для транзакционных нагрузок (OLTP)
Автоматическое развертывание и масштабирование БД	Автоматическое развертывание и масштабирование СУБД осуществляется средствами «Облачной платформы ВТБ» с помощью входящих в ее состав специализированных скриптов

Функция	Описание
Мониторинг	Предоставляет возможности мониторинга стандартных и расширенных метрик состояния оборудования и ПО, входящего в состав СУБД, и передачу их в систему мониторинга

Архитектура «Распределённой колоночной СУБД» представлена на следующем рисунке.

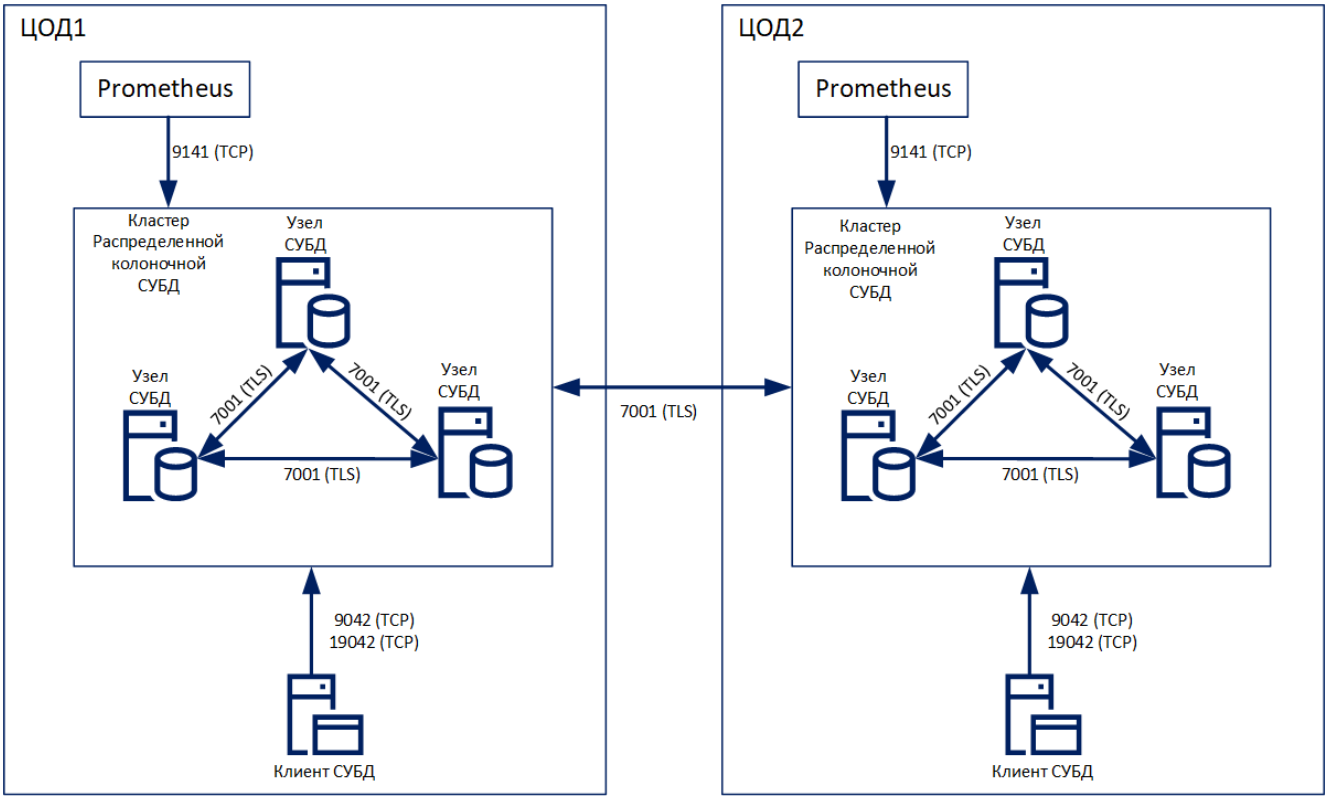

Рис. 4. Архитектура «Распределённой колоночной СУБД»

Описание взаимодействие компонентов СУБД приведено в Табл.7

Табл. 7. Взаимодействие компонентов СУБД

№	Инициатор	Приемник	Протоколы взаимодействия	Описание	Примечание
1	Клиент СУБД	Узел кластера СУБД	9042 (TCP), 19042 (TCP)	Подключение к сервисам базы данных для выполнения запросов чтения/добавления/ удаления/ изменения/ управления	Клиент инициирует подключение к любой доступному узлу. После получения информации о конфигурации кластера, клиент может подключиться ко всем узлам кластера, но должен отправлять запросы только на узлы, в том же ЦОД, к которому

				объектами и данными в СУБД	было инициировано подключение или на узлы ЦОД, который указан в конфигурации клиента до тех пор пока доступен хотя бы одни узел с данными в выбранном ЦОД
2	Prometheus	Узел кластера СУБД	9141 (TCP)	Подключение для сбора метрик работы ПО кластера	Метрики собираются с каждого узла кластера, каждый узел содержит метрики только о своей работе в кластере
3	Узел кластера СУБД	Узел кластера СУБД	7001 (TLS)	Взаимодействие между узлами кластера	Шифрованное подключение. Каждый узел кластера подключается ко всем другим узлам кластера. Текущая топология и состояние узлов кластера распространяется между узлами по протоколу GOSSIP. Так же по этому порту происходит всё остальное взаимодействие между узлами: чтение и запись данных, изменения системных данных, потоковая передача данных на новые узлы, восстановление данных между узлами, сообщения, используемые для реализации LWT (облегчённых транзакций) и другое

3.5 Колоночная аналитическая СУБД (1992)

Модуль «Колоночная аналитическая СУБД» (1992) (далее - Колоночная аналитическая СУБД) разработан на основе ClickHouse и предоставляет пользователю возможность управления прикладными БД с использованием нереляционной, колоночной модели хранения данных для аналитических нагрузок (OLAP). Колоночная аналитическая СУБД предоставляется пользователю в виде готовой услуги в режиме самообслуживания на Портале.

Описание предоставляемых модулем функциональных возможностей приведено в Табл. 8

Табл. 8. Описание функциональных возможностей

Функция	Описание
Управление БД	Обеспечивает хранение и обработку данных в нереляционной, колоночной модели хранения данных для аналитических нагрузок (OLAP)
Автоматическое развертывание и масштабирование БД	Автоматическое развертывание и масштабирование СУБД осуществляется средствами «Облачной платформы ВТБ» с помощью входящих в ее состав специализированных скриптов
Мониторинг	Предоставляет возможности мониторинга стандартных и расширенных метрик состояния оборудования и ПО, входящего в состав СУБД, и передачу их в систему мониторинга

Архитектура «Колоночной аналитической СУБД» представлена на следующем рисунке

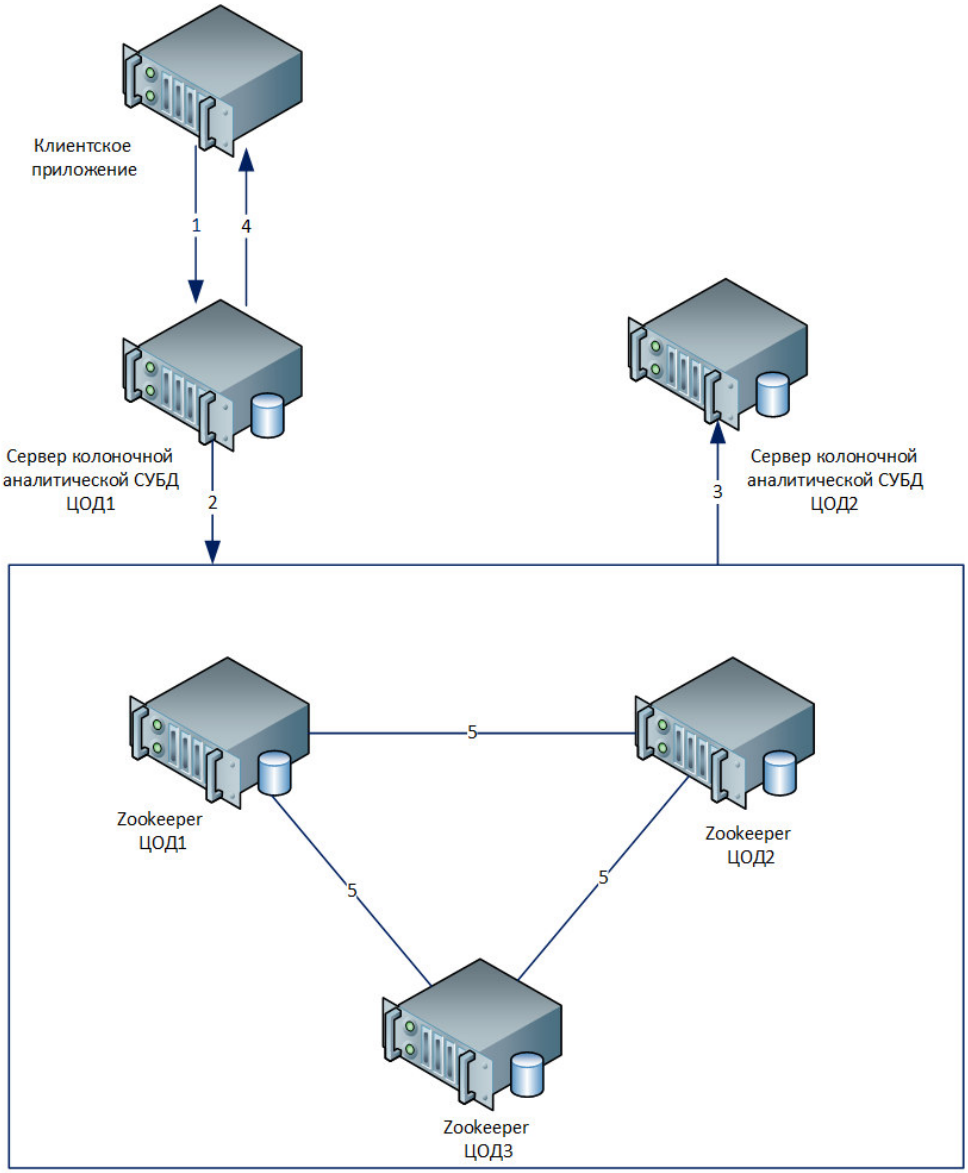

Рис. 5. Архитектура «Колоночной аналитической СУБД»

Описание взаимодействие компонентов СУБД приведено в Табл.9.

Табл. 9. Взаимодействие компонентов СУБД

№	Инициатор	Приемник	Передаваемые данные/ управляющие воздействия	Форматы/ протоколы взаимодействия	Примечание описывающее взаимодействие
1	Клиентское приложений	Сервер колоночной аналитической СУБД ЦОД1	SQL запрос на запись или на чтение	HTTPs\TCP secure	Пользователь или приложение ИС обращается к серверу СУБД
2	Сервер колоночной аналитической СУБД ЦОД1	Кластер Zookeeper	Реплицируемые данные	TCP secure	Сервер СУБД на ЦОД1 делает записи изменений в кластер Zookeeper
3	Кластер Zookeeper	Сервер колоночной аналитической СУБД ЦОД2	Реплицируемые данные	TCP secure	Сервер СУБД ЦОД2 делает записи изменений из кластера Zookeeper
4	Сервер колоночной аналитической СУБД ЦОД1	Клиентское приложений	Ответ на SQL запрос	HTTPs\TCP secure	Сервер СУБД отвечает клиенту на SQL запрос или об успешно записанных данных
5	Zookeeper	Zookeeper	Поддержание кворума, обмен данных внутри ансамбля, выборы лидера	TCP	Ноды Zookeeper голосуют для определения ведущего\активного мастера

3.6 СУБД полнотекстового поиска (1991)

Модуль «СУБД полнотекстового поиска» (1991) (далее - СУБД полнотекстового поиска) разработан на основе OpenSearch и предоставляет пользователю возможность управления прикладными ИТ-системами для сбора и хранения данных с возможностью полнотекстового поиска и анализа данных. СУБД полнотекстового поиска предоставляется пользователю в виде готовой услуги в режиме самообслуживания на Портале.

Описание предоставляемых модулем функциональных возможностей приведено в Табл. 10

Табл. 10. Описание функциональных возможностей

Функция	Описание
Управление БД	Обеспечивает сбор, хранение и обработку данных с возможностью полнотекстового поиска и анализа данных

Функция	Описание
Автоматическое развертывание и масштабирование БД	Автоматическое развертывание и масштабирование СУБД осуществляется средствами «Облачной платформы ВТБ» с помощью входящих в ее состав специализированных скриптов
Мониторинг	Предоставляет возможности мониторинга стандартных и расширенных метрик состояния оборудования и ПО, входящего в состав СУБД, и передачу их в систему мониторинга

Функциональная архитектура «СУБД полнотекстового поиска» представлена на следующем рисунке.

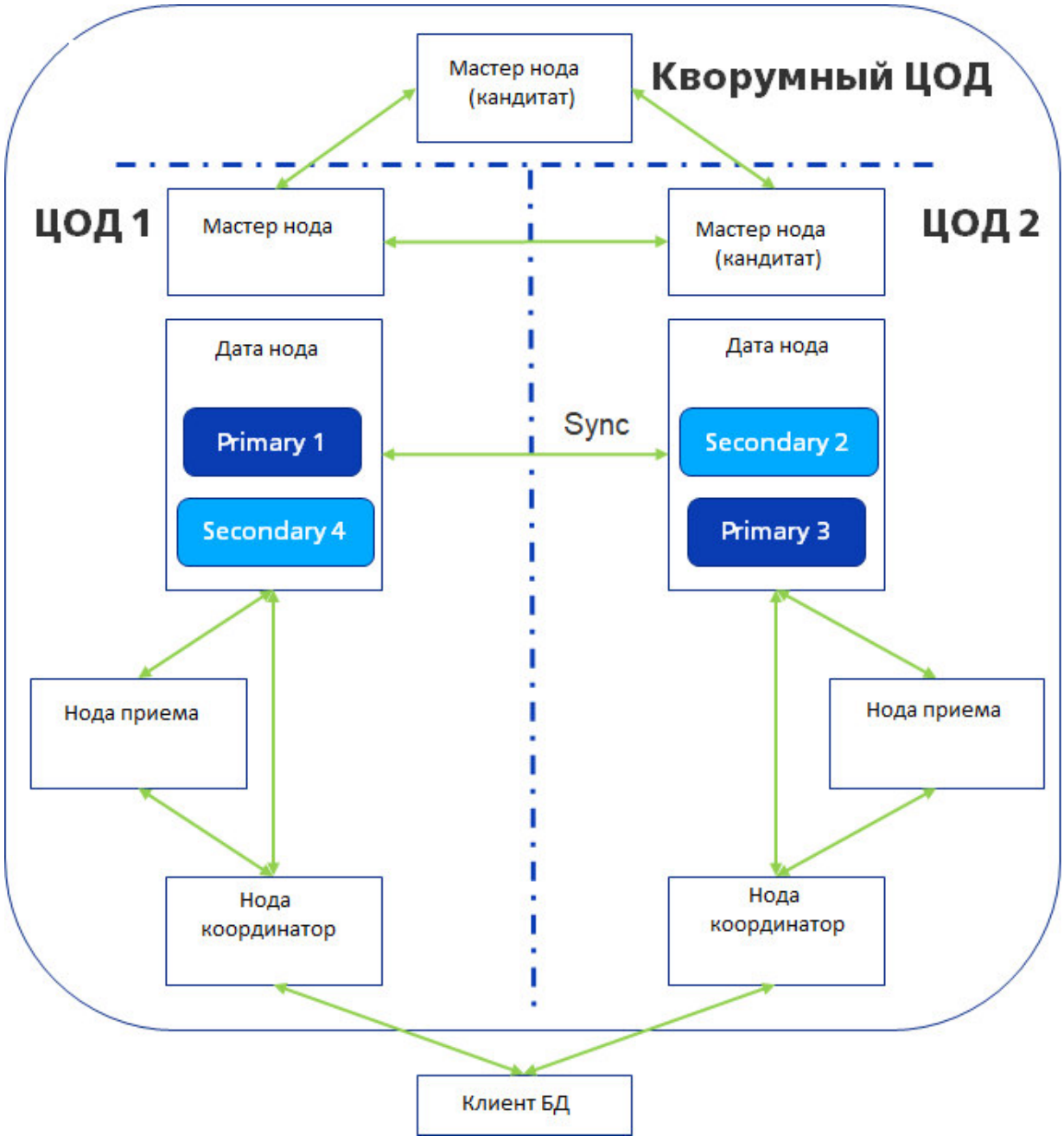

Рис.6. Архитектура «СУБД полнотекстового поиска»

Описание взаимодействие компонентов СУБД приведено в Табл.11

Табл.11. Взаимодействие компонентов СУБД

№	Инициатор	Приемник	Передаваемые данные/ управляющие воздействия	Протоколы взаимодействия	Примечание описывающее взаимодействие
1	Клиент базы данных	Нода координатор	Все запросы приходящие в кластер СУБД	REST API / HTTPS	Способы взаимодействия с кластером СУБД выбирает владелец ИС
2	Нода координатор	Дата нода	Все запросы на поиск, удаление или обновление данных в индексах	REST API / HTTPS	Если на ноду координатор поступил запрос на поиск, удаление или обновление данных запрос отправляется сразу на дата ноды минуя ноды координаторы
3	Дата нода	Нода координатор	Возвращает ответ на поиск, удаление или обновление данных в индексах	REST API / HTTPS	После выполнения запроса, дата ноды возвращают искомую информацию или ответ что изменения и удаления применены.
4	Нода координатор	Клиент базы данных	Возвращает ответ на поиск, удаление или обновление данных в индексах	REST API / HTTPS	Кластер СУБД выдает искомую информацию из индексов предварительно объединив ее в один ответ или ответ что изменения и удаления применены.
5	Нода координатор	Нода приема	Запросы на запись с данными	REST API / HTTPS	Все данные на запись приходят на ноды приема для преобзаровния данных
6	Нода приема	Дата нода	Запросы на запись с данными	REST API / HTTPS	После преобразования данных, нода приема отправляет данные на дата нода для их записи

					для хранения и последующей работы с данными
7	Дата нода	Нода координатор	Ответ на успешную запись данных на дата ноде	REST API / HTTPS	После успешной записи на дата ноде, дата нода возвращает ответ что данные записаны
9	Нода координатор	Клиент базы данных	Ответ на успешную запись данных на дата ноде	REST API / HTTPS	После того как дата нода возвращает ответ что данные записаны координатор ноде, координатор нода возвращает ответ клиенту
10	Мастер нода	все ноды	Опрашивает все ноды кластера о их состоянии	REST API / HTTPS	В случае проблем на какой-либо ноде, запускает алгоритмы минимизации ущерба. Например в случае высокой утилизации дискового пространства на дата ноде дает указания на релокацию шардов

4 Среды виртуализации («Облачные решения»)

Для развертывания частного облака в рамках продукта «Облачная платформа» используются среды виртуализации на основе проекта с открытым исходным кодом OpenStack.

Общие сведения об OpenStack

OpenStack – проект с открытым исходным кодом по разработке платформы, позволяющей строить частные и публичные Облака (cloud computing). Цель проекта – предоставление простых и удобных широкомасштабных и многофункциональных решений для любого типа облака. Технология включает в себя серию взаимосвязанных проектов, обеспечивающих разработку многочисленных составляющих инфраструктурного решения для Облака.

OpenStack обеспечивает массовый запуск однотипных виртуальных серверов для хостинга приложений с собственными средствами обеспечения отказоустойчивости. Сама платформа не предлагает высокой доступности отдельно взятого виртуального сервера.

OpenStack предоставляет дополнительные услуги, такие как управление идентификацией, оркестрация, учет потребляемых ресурсов в той же программной основе через API. OpenStack также

создает базу для реализации практик DevOps, обеспечивающих непрерывную интеграцию и методологию непрерывного развертывания.

OpenStack не является гипервизором, но он поддерживает несколько гипервизоров (коммерческие и с открытым исходным кодом) через слой абстракции.

Функциональные подсистемы и слои, серверные роли

Система состоит из следующих функциональных подсистем:

- подсистема управления (ПУ);
- подсистема управления приложениями (ПУП);
- подсистема вычислительных ресурсов (ПВР);
- подсистема вычислительной сети (ПВС);
- подсистема хранения данных (ПХД);
- подсистема мониторинга и журналирования (ПМ);
- подсистема вспомогательных сервисов (ПВИС).

Функциональная схема системы представлена на Рис. 7.

Для описания архитектуры системы используется понятие «серверная роль». В системе выделены следующие серверные роли:

1. Контроллер (control node) – это сервер, содержащий управляющие компоненты различных подсистем, а именно:

Компоненты подсистемы вычислительных ресурсов	Компоненты подсистемы вычислительной сети	Компоненты подсистемы хранения данных	Компоненты подсистемы мониторинга и журналирования	Компоненты подсистемы вспомогательных сервисов
• Nova; • Ironic	• Neutron	• Glance; • Cinder; • Ceph-mon; • Swift	• Prometheus; • Loki; • Grafana	• Galera (кластер MySQL); • HAProxy

2. Сервер виртуализации (compute node, сервер виртуализации) – сервер ПВР со следующими компонентами:
   - Nova-Compute;
   - Neutron-OVS agent.
3. Сервер хранения (storage node) – сервер ПХД со следующими компонентами:
   - Cinder-volume;
   - Ceph-OSD.
4. Сетевой сервер (network node) – сервер ПВС со следующими компонентами:
   - Neutron agents.

Состав программных компонентов, установленных на серверах с различными ролями, отличается в зависимости от региона. Также один физический сервер или виртуальная машина могут одновременно иметь несколько серверных ролей.

Помимо функциональных подсистем и серверных ролей архитектуру системы можно представить в виде функциональных слоев, каждый из которых представляет собой набор серверов с идентичными конфигурациями и установленными компонентами:

- слой управления (control plane) – состоит из серверов с ролью контроллер;

- слой обработки данных (data plane) – состоит из серверов с ролью compute node (серверы виртуализации) и серверов с ролью network node (сетевые серверы);
- слой хранения (storage plane) – состоит из серверов с ролью storage node (серверы хранения).

Примечание. Подсистема мониторинга и журналирования является обособленной и не относится к слоям, описанных выше.

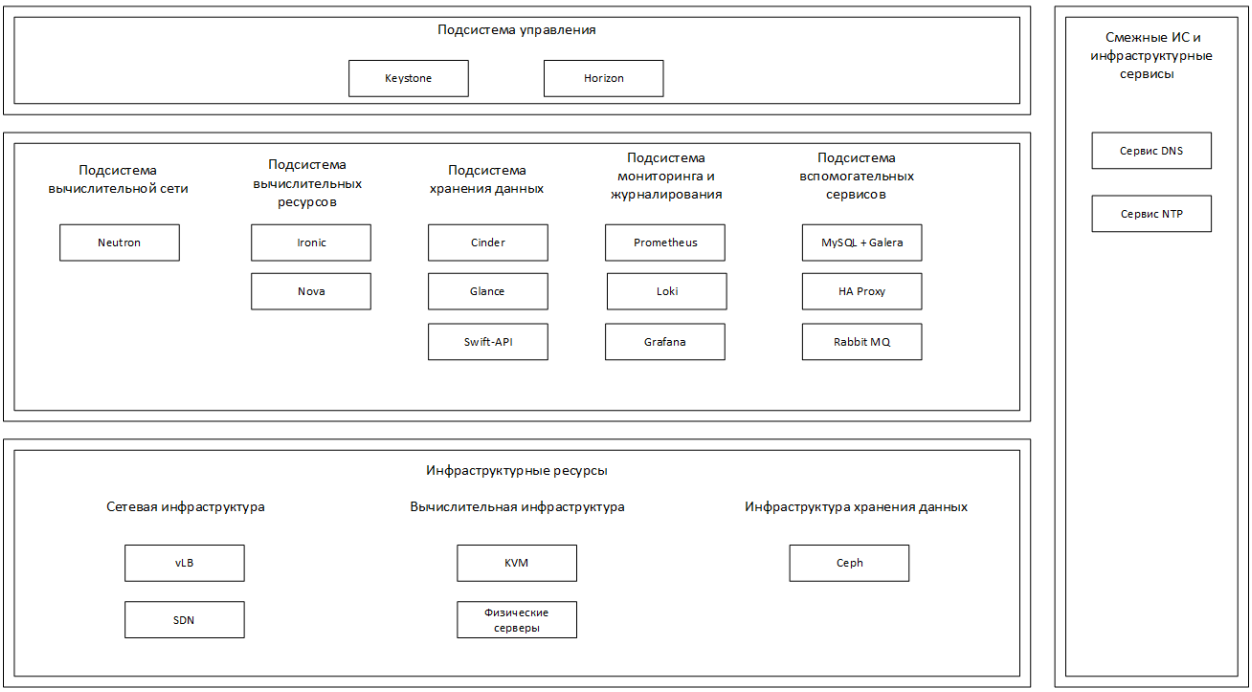

Рис. 7. Функциональная схема платформы виртуализации на основе OpenStack

5 Платформы контейнеризации («Сервисы платформ контейнеризации»)

В качестве платформ контейнеризации используются OpenShift и Kubernetes.

На его основе пользователям предоставляются готовые контейнеры для развёртывания приложений.

Кластеры Kubernetes развертываются на ресурсных пулах среды виртуализации OpenStack.

6 Дополнительные возможности

Продукт «Облачная платформа» предоставляет ряд дополнительных возможностей, направленных на оптимизацию стоимости владения и повышение удобства использования облачных ресурсов.

Витрина продуктов

Витрина продуктов обеспечивает сокращение времени и затрат на вывод новых продуктов в облако, а также расширение доступного каталога облачных продуктов.

В рамках витрины предоставляется открытый фреймворк для взаимодействия с внутренними потребителями в рамках работы по выводу их собственных продуктов на облачную платформу. Тем самым обеспечивается интенсивное наполнение продуктового каталога облачной платформы, а пользователи получают возможность гибко управлять составом доступных продуктов.

Основные этапы процесса вывода продукта в облако через витрину:

1. Инициация – принятие решения о разработке продукта, принятие решения об инвестировании в разработку.
2. Прототип – проверка техническое жизнеспособности решения в соответствии с видением продукта.
3. Пилот – запуск продукта в соответствии с плановыми требованиями, сроками и ресурсными ограничениями.
4. Запуск – завершение бета-тестирования и начало промышленной эксплуатации продукта; передача продукта на поддержку.
5. Мониторинг – принятие корректирующих мер, внедрение изменений.
6. Вывод – исключение из продуктовой линейки устаревших и неперспективных продуктов.

Гибридное облако -Комбинация из двух или более различных облачных инфраструктур (частных или публичных), остающихся уникальными объектами, но связанными между собой стандартизированными или частными технологиями передачи данных и приложений.

Для сокращения времени получения ресурсов во внешнем облаке и оптимизации расходов предоставляется возможность использования ресурсов публичных облачных провайдеров.

Использования внешнего облака наряду с частным обеспечивает:

максимальную гибкость и масштабируемость;

централизованное управление стоимостью;

управляемый уровень соответствия требованиям безопасности.

Планировщик инфраструктуры

Планировщик – инструмент централизованного управления циклом подготовки инфраструктуры, от планирования до развертывания. Его использование позволяет сократить расходы на инфраструктуру за счет повышения качества планирования потребностей, гибкого распределения ресурсов в зависимости от приоритета потребностей и обеспечения прозрачности процесса подготовки.

Основные этапы процесса планирования инфраструктуры с помощью данного инструмента:

1. Сбор информации о / прогнозирование потребностей для подготовки плана пополнения ресурсных пулов на основе:

   методики прогнозирования органического роста;

   заявок на крупные потребности;

   каталога ресурсных пулов и стандартных продуктов.
2. Закупка и поставка – спецификация и размещение заказа, подготовка плана поставки и монтажа на основе:

   стандарта пополнения ресурсных пулов;

   каталога оборудования в генеральных соглашениях.
3. Установка и настройка – ресурсные пулы пополнены. На данном этапе используются:

   стандарт автоматизированной разливки;

   сквозная приоритизация на основе прогнозов.

Автоматизация обслуживания (Day2-операции)

Для сокращения времени выполнения типовых операций с инфраструктурой и сокращения ошибок, связанных с человеческим фактором, большинство наиболее популярных запросов могут выполняться в режиме самообслуживания.

Примеры автоматизируемых операций:

- установка обновлений (патчей) операционных систем, системного программного обеспечения, баз данных;

- простые инфраструктурные операции: перезапуск операционной системы и системного программного обеспечения.

7 Интерфейсы

Программный продукт «Облачная платформа» предоставляет пользовательский веб-интерфейс и REST API-интерфейс.

Пользовательский веб-интерфейс

Веб-интерфейс портала облачной платформы предоставляет возможность пользователям получать инфраструктурные продукты в режиме самообслуживания.

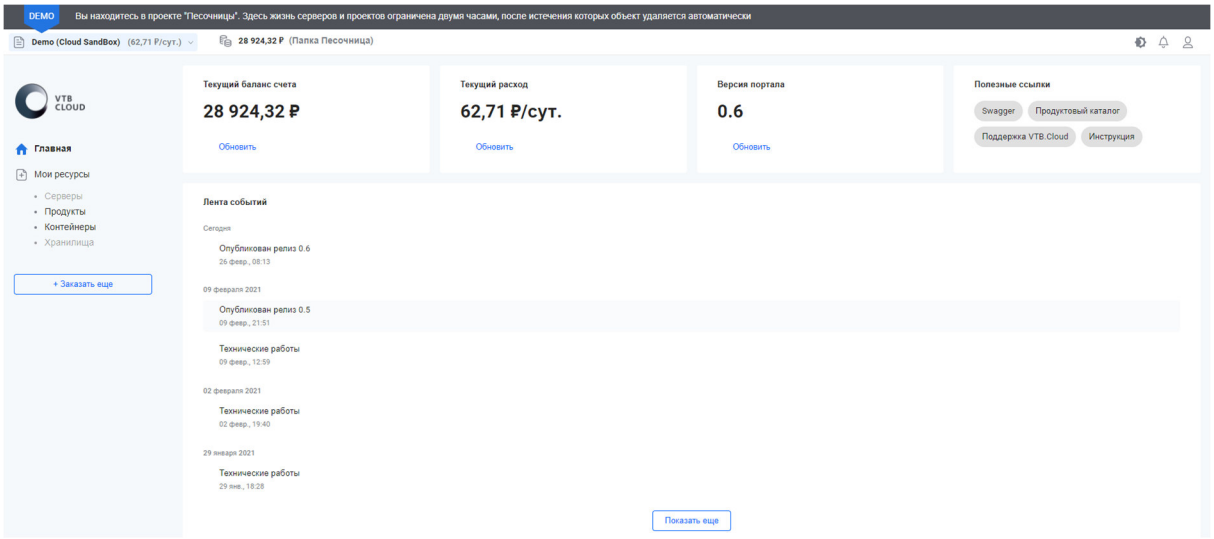

Рис. 8. Главная страница портала

REST API

Сервисы продукта «Облачная платформа» предоставляют REST API-интерфейсы, которые могут использоваться для автоматизации развертывания инфраструктуры и реализации подхода Infrastructure as Code.

Описание API предоставляется средствами Swagger. Для списков сущностей реализован постраничный вывод; с помощью настраиваемых параметров запроса могут быть получены расширенные данные.

Сервис авторизации (Authorizer) предоставляет REST API для выполнения следующих групп операций:

- операции с папками;
- операции с организациями;
- операции с сервисными учетными записями;
- операции с проектами;
- операции с разрешениями на доступ к управлению организационной структурой;
- операции с пользователями.

Бэкенд портала облачной платформы (Portal) предоставляет REST API для выполнения следующих групп операций:

- операции с группами пользователей;
- операции с группами доступа;
- операции со средами;
- операции с информационными системами;

- операции с SSH-ключами.

Сервис создания заказов (Order Service) предоставляет REST API для выполнения следующих групп операций:

- операции с ЦОД;
- операции с доменами;
- операции с сетевыми сегментами;
- операции с заказами;
- операции с платформами;
- операции с ресурсными пулами.

Сервис тарификации (Tarrificator) предоставляет REST API для выполнения следующих групп операций:

- операции со стоимостью заказа.

8 Архитектура

На Рис. 9 показана архитектура продукта «Облачная платформа».

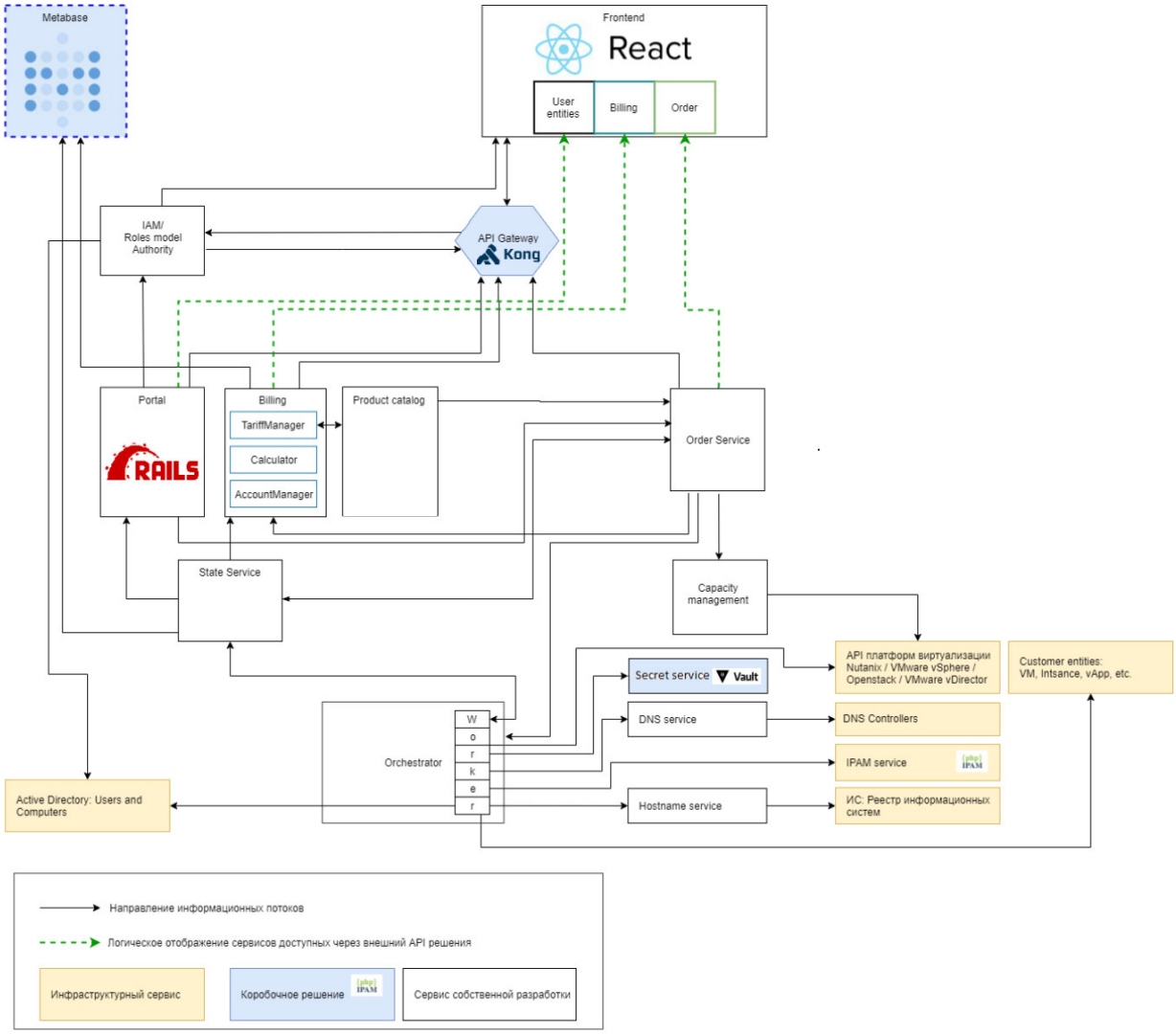

Рис. 9. Архитектура продукта «Облачная платформа»

8.1 Компоненты и их назначение

Компонент	Назначение
Frontend	Обеспечивает работу пользователя с основными бэкенд-сервисами через веб-интерфейс. Имеет адаптивный дизайн для обеспечения удобной работы на устройствах с различными разрешениями экрана
Portal	Сервис организации хранения организационных бизнес-сущностей (группы и области) и стендов
API Gateway	Шлюз API на базе продукта с открытым исходным кодом Kong
Active Directory (AD)	Службы каталогов корпорации Microsoft для операционных систем семейства Windows Server. Средство иерархического представления ресурсов, принадлежащих некоторой отдельно взятой организации, и информации об этих ресурсах.
Order Service	Сервис создания заказов
Capacity Management	Сервис ресурсного планирования
Product Service (Product Catalog)	Сервис хранения продуктов
Authorizer (IAM)	Сервис предоставления прав доступа пользователей к указанным ресурсам облачной платформы. Является частью системы IAM, обеспечивающей управление доступом (AM – access management) и делегирующий часть управления идентификацией Keycloak (IdM – identity management)
vApp	Контейнер, в котором размещаются виртуальные машины
Billing	Система управления биллингом
Metabase	Инструмент для бизнес-аналитики с открытым исходным кодом. Пользователи задают вопросы о данных, а Metabase отображает ответы в осмысленных форматах, таких как гистограмма или таблица. Вопросы к данным сохраняются и группируются в информационные панели (dashboards)
State Service	Сервис централизованного хранения информации о пользовательских объектах, выданных через облачную платформу
Orchestartor	Сервис оркестрации выполнения графов, полученных из продуктового каталога
DNS Service	Микросервис, работающий с связке с Microsoft Winodws DNS, который восполняет пробел в решении MS, предоставляя REST API для управления DNS-записями. Поддерживает управление записями с типами A, PTR, CNAME. Все имена хостов (hostname) проверяются на соответствие принятым правилам именования серверной инфраструктуры.

	Позволяет создавать записи wildcard для кластеров OpenShift
Hostname Service	Сервис именования. Позволяет зарезервировать имена виртуальной машины, кластера или проекта либо удалить резервирование
TariffManager	Сервис тарификации ресурсов
Worker	Часть сервиса оркестрации. Как правило, соответствует вершине графа развертывания продукта. Типы worker: • Worker: cloud-mate – обеспечивает асинхронные вызовы на платформах виртуализации; читает сообщения из RabbitMQ, выполняет операции на платформе; • Worker: rpm-awx – асинхронный клиент для платформы Ansible AWX/Tower; • Worker: ansible – хранение сценариев (playbook) и ролей Ansible для исполнения Ansible AWX; • Worker: dns – обеспечивает взаимодействие с DNS Service; позволяет управлять записями с типом A, PTR, CNAME во внутренних DNS-зонах; • Worker: ipam – обеспечивает взаимодействие с сервисом IPAM; позволяет резервировать IP-адреса и удалять резервирование; • Worker: hostname – обеспечивает взаимодействие с сервисом имен; позволяет резервировать имена виртуальных машин, кластеров, проектов и снимать резервирование

8.2 Информационные потоки

На Рис. 10 показаны основные информационные потоки между компонентами продукта «Облачная платформа»

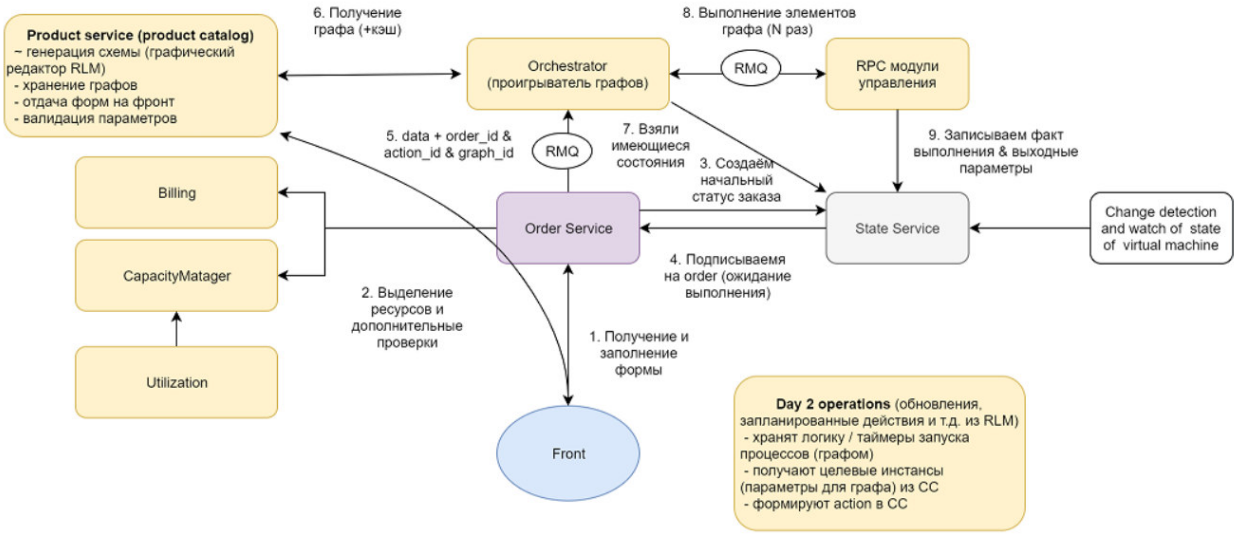

Рис. 10. Схема информационных потоков

№ на схеме	Назначение
1	Из продуктового каталога (запросы проксирует Order Service для проверки прав доступа) пользователь на портале получает список возможных для выполнения графов. При переходе на страницу продукта происходит получение формы с пользовательскими полями графа (JSON-схема). После заполнения форма отправляется в Order Service
2	Order Service: • еще раз проверяет права пользователя на данный граф; • резервирует место в ресурсных пулах / на хостах; • заполняет системные поля графа (хосты и ресурсные пулы); • проверяет баланс пользователя в биллинге
3	Выполнение графа, т. е. выполнение действия (action) в данном заказе (order). Если это первое действие в заказе, заказ создается в State Service
4	Выполняется подписка на изменение заказа в State Service для отображения хода выполнения пользователю (через сокеты)
5	Данные дополняются заданными в шаблоне полями (order_id, action_id, graph_id) и отправляются в оркестратор через RabbitMQ (предыдущие взаимодействия осуществляются через REST API). Данные (data) – системные параметры, полученные в Order Service, а также параметры, заданные в графе, и пользовательские параметры (заполнение формы по JSON-схеме)
6-9	По graph_id оркестратор проверяет свой кэш: если графа в нем нет, получает граф из продуктового каталога и кэширует его. Выполняется получение данных по данному действию (action) из State Service (восстановление работы после сбоя оркестратора – если граф уже выполнялся, выполнение продолжается с шага, на котором произошел сбой). Затем оркестратор выполняет граф, вызывая описанные в нем RPC-модули и дополняя их ответами начальные данные. Также модули отправляют свои ответы (action) и события по сущностям платформы (event) в State Service. В случае ошибки начинается операция отката графа. В ходе выполнения оркестратор записывает изменения в State Service. State Service обеспечивает рассылку данных об изменениях другим сервисам (обратная связь с Order Service, Billing и т. д.). Одновременно по заказу может выполняться только один граф (для предотвращения взаимоисключающих операций)

8.3 Используемый стек технологий

Серверы приложений	Платформы (framework)
1. Keycloak Open Source: ASL 2.0 (Apache Software Licence) 2. React Open Source: MIT License / X11 License 3. Rails Open Source: MIT License 4. Kong Open Source Software	OpenStack virtualization platform