**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА**

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Петербургский государственный университет путей сообщения**

**Императора Александра I»**

**(ФГБОУ ВО ПГУПС)**

Факультет «Автоматизация и интеллектуальные технологии»

Кафедра «Информатика и информационная безопасность»

Лабораторная работа № 2

по дисциплине

«Защита информации в РИС и ЦОД»

«Проектирование распределенной информационной системы»

|  |  |  |
| --- | --- | --- |
| **Выполнил обучающийся**  Курс 4  Группа КИБ-012 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | Г. Е. Груздев |
|  |  |  |
| **Проверил** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись, дата | А.А. Корниенко |

Санкт-Петербург

2024

**Цель выполнения лабораторной работы:**

Изучение подходов к проектированию и построению архитектуры распределенной информационной системы.

**Описание выбранного объекта автоматизации – сервисы цифровых экосистем**

Для примера будет рассматриваться экосистема Яндекса.

Объект автоматизации представляет собой совокупность распределённых вычислительных и коммуникационных ресурсов, которые работают вместе для предоставления пользователям различных сервисов, таких как поиск, почта, музыка, логистика, облачные услуги и другие. Основная задача автоматизации в распределённой информационной системе заключается в обеспечении беспрерывной работы сервисов, эффективном распределении вычислительных ресурсов и оптимизации взаимодействия между компонентами системы.

*Иерархическая структура экосистемы Яндекса:*

* *Центральный уровень (ядро экосистемы)*

Отвечает за общую координацию работы всех сервисов, включает управление вычислительными ресурсами и разработку стратегических решений. Включает основное руководство, отделы исследований и разработок (R&D), а также команду DevOps, которая занимается поддержанием инфраструктуры.

* *Уровень сервисов*

Каждая группа сервисов, таких как поиск, логистика (например, Яндекс.Go), облачные технологии (Yandex Cloud), ведётся отдельными командами. Эти команды обеспечивают разработку и сопровождение продуктов, управление их данными и пользовательским интерфейсом.

* *Региональные подразделения*

Отвечают за локализацию сервисов, адаптацию к специфике региона и поддержку пользователей на местах. Эти подразделения могут включать центры обработки данных и службы технической поддержки.

* *Пользовательский уровень*

Конечные клиенты, которые взаимодействуют с экосистемой через мобильные приложения, веб-платформы или API, получают доступ к сервисам и предоставляют обратную связь.

**Назначение и основные функции распределенной информационной системы (РИС)**

Назначение распределённой информационной системы (РИС) сервисов цифровой экосистемы Яндекса заключается в обеспечении надёжной, масштабируемой и производительной работы всех цифровых сервисов Яндекса, включая поиск, навигацию, мультимедиа, доставку, финансы, и другие. Основной целью РИС является поддержка большого объёма данных и запросов, с которыми работают эти сервисы, предоставление пользователям непрерывного и быстрого доступа к услугам, а также повышение эффективности за счёт автоматизации и распределённой обработки данных.

Распределенная информационная система сервисов цифровых экосистем должна обеспечивать непрерывное взаимодействие между всеми участниками экосистемы и эффективное управление ими. Основные функции РИС включают в себя:

* Поддержку взаимодействия и коммуникации между пользователями.
* Обеспечение безопасности данных и конфиденциальности информации.
* Управление транзакциями и платежами.
* Аналитику и отчётность по различным аспектам экосистемы.
* Поддержку интеграции с другими системами и сервисами.

Цифровая экосистема помогает поддерживать клиента на всех этапах его жизненного цикла, продавать дополнительные продукты, услуги и сервисы, решать его задачи.

Все сервисы в составе экосистемы заточены на выполнение важной функции — повысить качество обслуживания клиента и продлить время сотрудничества с ним. В свою очередь это ведёт к увеличению прибыли, которую компания получает за все время сотрудничества с данным клиентом.

Каждый онлайн-сервис в свою очередь направлен на решение одной из ключевых задач, среди которых в общем случае выделяют:

* Увеличение скорости запуска продуктов, например, дизайн-система и медиапортал.
* Снижение стоимости привлечения трафика, например, SEO оптимизация сайта, подробный каталог, блог с обзорами автомобилей.
* Повышение конверсии, например, сравнение моделей автомобиля и калькулятор выгод.
* Поддерживающие инфраструктурные проекты, например, различные конструкторы и автоматизации.

**Описание архитектуры, проектируемой РИС.**

Цифровая экосистема Яндекса представляет собой многоуровневую архитектуру, которая объединяет различные сервисы и платформы, обеспечивающие разнообразные цифровые услуги. Важным аспектом экосистемы является тесная интеграция сервисов, их взаимная поддержка и использование общей инфраструктуры.

Принципы, по которым строится система:

#### **1. Микросервисная архитектура**

* Каждый сервис Яндекса реализуется как отдельный микросервис, отвечающий за конкретный функционал (поиск, обработка данных, аналитика и т.д.). Это повышает гибкость и позволяет командам разрабатывать и внедрять новые функции независимо друг от друга.
* Микросервисы взаимодействуют через RESTful API или gRPC, обеспечивая стандартизированное и надёжное взаимодействие между различными компонентами системы.

#### **2. Оркестрация и контейнеризация**

* Контейнеризация: Сервисы развёртываются в контейнерах, что делает их портативными и обеспечивает изоляцию. Это позволяет легко переносить микросервисы между средами разработки, тестирования и продакшена.
* Оркестрация с Kubernetes: Kubernetes управляет контейнерами, обеспечивая автоматическое развертывание, масштабирование и балансировку нагрузки, а также восстанавливает работоспособность микросервисов в случае сбоев.

#### **3. Распределённое хранение данных**

* Хранение данных в облаке: Яндекс.Cloud предоставляет распределённое объектное хранилище, подходящее для хранения больших объёмов данных (видео, аудио, документы).
* Репликация и отказоустойчивость: Данные реплицируются между различными дата-центрами, что обеспечивает надёжное хранение и доступ к ним даже в случае сбоев в одной из зон доступности (например, сбои в Московском дата-центре не отразятся на работе сервисов у пользователей в Санкт-Петербурге).

#### **4. Event-Driven Architecture (EDA)**

* Архитектура управления событиями позволяет микросервисам реагировать на события и запускать процессы по мере их возникновения. Например, при создании заказа в Яндекс.Маркете система автоматически уведомляет службы доставки, оплаты и складского учёта.

#### **5. Машинное обучение и аналитика**

* CatBoost: Собственная библиотека машинного обучения Яндекса используется для анализа данных, персонализации рекомендаций и прогнозирования.
* Big Data платформы (Hadoop, Spark): используются для обработки больших данных и выполнения аналитических задач. Эти технологии позволяют обрабатывать данные из различных источников (поисковые запросы, транзакции, пользовательские действия) и создавать модели машинного обучения.

#### **6. Мониторинг и безопасность**

* Prometheus и Grafana: Постоянный мониторинг состояния микросервисов и инфраструктуры позволяет своевременно выявлять и устранять проблемы, обеспечивая бесперебойную работу системы.
* Системы безопасности (OAuth 2.0, OpenID Connect): используются для аутентификации и авторизации пользователей, а также для защиты данных. Все данные шифруются и проходят строгий контроль доступа.

**Описание программных и технических средств поддержки, проектируемой РИС**

1. **Программные средства:**

**- Операционная система и инструменты для разработки:** серверными ОС будут решения из семейства Linux (например, Debian или CentOS). АРМы сотрудников могут быть как ОС семейства Linux, так и Windows или MacOS. Интегрированными средами разработки могут быть продукты JetBrains, VSCode, NeoVIM и т.д.

**- Веб-приложения:** веб-приложения являются основным интерфейсом для взаимодействия пользователей с системой. Они могут включать в себя интерактивные панели управления, формы для ввода данных, отображение информации и т.д. Веб-приложения обычно разрабатываются с использованием современных веб-технологий, таких как HTML, CSS, JavaScript, а также фреймворков для разработки веб-приложений, например, React, Angular, Vue.js, Spring Framework и др.

**- Мобильные приложения:** мобильные приложения предоставляют возможность взаимодействия с системой на мобильных устройствах, таких как смартфоны и планшеты. Они могут иметь те же функциональные возможности, что и веб-приложения, но адаптированы к мобильным устройствам и их особенностям. Мобильные приложения разрабатываются платформ iOS и Android, с использованием языков программирования Swift, Kotlin, Java, Flutter.

**- Бэкенд-сервисы (сервисы бэк-офиса):** бэкенд-сервисы отвечают за обработку бизнес-логики, управление данными, аутентификацию пользователей, обработку запросов и другие задачи. Они обычно разрабатываются с использованием языков программирования, таких как Java, Python, Go, Node.js, а также фреймворков и технологий, таких как Spring, Django, Express.js и др. Бэкенд-сервисы могут взаимодействовать с базами данных, внешними API и другими компонентами системы.

**- Базы данных:** реляционные (например, PostgreSQL, MySQL, Oracle), NoSQL (например, MongoDB, Cassandra, Redis).

- **Платформа контейнеризации:** используются Docker-контейнеры. Чтобы оркестрировать контейнеры, то есть управлять работой их работой, используют кластеры Kubernetes.

**- Балансировщик:** контролирует, чтобы нагрузка на приложение распределялась по облачным ресурсам равномерно. Яндекс имеет собственную инфраструктуры, общее название которой YandexCloud. Для балансировки и маршрутизации запросов между микросервисами и сервисами используются Nginx и HAProxy.

- **Мониторинг и логирование:** стек ELK (Elasticsearch, Logstash и Kibana), Graylog, для дополнительной информации - обычные реляционные БД, например PostgreSQL.

1. **Технические средства:**

**- Серверы:** для развертывания и обеспечения работы программных компонентов системы необходимы серверы. Серверная группа может быть представлена сервером БД, сервером приложений, файловым сервером, веб-серверами (apache, nginx), почтовым сервером, CDN-сервером и т.д. Серверы размещаются на собственной инфраструктуре компании.

**- Хранилища данных:** для хранения данных системы используются хранилища данных, такие как сервера баз данных или облачные хранилища.

**- Сетевое оборудование:** для обеспечения связности между компонентами системы и пользователями необходимо сетевое оборудование, такое как маршрутизаторы, коммутаторы, брэндмауеры и другие устройства.

**Проработка вопроса с позиции иерархии РИС**

Ghbdtn