

# Система Образовательного менеджмента

Архитектурное решение



# 01 Бизнес архитектура

# Описание проекта

**Разработать систему образовательного менеджмента для школ и университетов, поставляемую как сервис**

**Пользователи:** преподаватели, сотрудники и родители студентов

## **Требования:**

- Отслеживание отсутствий, опозданий и их причин (данные вводятся родителями, преподавателями или сотрудниками)
- Управление данными от 1000 до 1 млн. студентов
- Составление отчетов по данным студенческой успеваемости
- Отслеживание оценок и заданий учащихся (выполненных и заданных)
- Форумы для общения учителей и родителей
- Развертывание системы как SaaS-продукта из хостинг-центра

## **Дополнительный контекст:**

- Агрессивная маркетинг-стратегия
- Основной конкурент пострадал из-за утечки данных
- Новый CIO
- Основные конкурентные преимущества – гибкость, возможность настройки под нужды клиента, безопасность

# Бизнес-драйверы

## 01

### Недостатки существующих систем

- Долгое развертывание
- Внедрения новой функциональности занимает продолжительное время
- Сложности с масштабированием

## 02

### Ключевой конкурент

Позиция ключевого конкурента ослаблена в результате утечки данных

## 03

### Бюджет

Бюджет разработки неограничен

## 04

### Маркетинг

Сильная команда маркетологов

# Атрибуты качества (1/2)

## Надежность

- Допустимая потеря данных (RPO) - не должно быть вообще
- Допустимое время восстановления системы после сбоя - 1 час
- После установления соединения с сервером, система проводит проверку на возможную подмену данных и синхронизирует данные, сохраненные оффлайн на машине клиента во время недоступности базы данных или API. При восстановлении сетевой связности, система уведомляет пользователя о том, что есть неотправленные данные, и предлагает отправить их на сервер для сохранения. Это гарантирует, что данные не теряются и сохранены в базе данных в соответствии с требованиями безопасности и целостности.

## Безопасность

- Личные данные пользователей должны быть надежно защищены в соответствии с законодательством РФ
- Двухфакторная идентификация пользователей
- Недоступность баз данных снаружи

# Атрибуты качества (2/2)

## Производительность

- Ответ основных ручек API - до 1 с в 80 %
- Ответ основных ручек API – до 3 с в 95 %
- Построение отчетов - асинхронное. Пользователь получает пуш, когда отчет готов.
- Нагрузка на систему сильно возрастает во время сессий, нужно уметь динамически расширяться на время прогнозируемых пиков нагрузки.
- Система имеет выраженное снижения нагрузки в ночные часы и выходные дни. Это время стоит максимально использовать для тяжелых вычислений (составления отчетов).

## Модифицируемость

- Стоимость пятикратного увеличения производительности системы не должна превышать 400% от стоимости эксплуатации на момент сдачи

# Роли и права доступа в учебной информационной системе

- **Администратор учебного заведения:**
  - Имеет полный доступ ко всем данным студентов
  - Может редактировать и обновлять информацию о студентах
  - Формирует отчеты и аналитику по учебному процессу
- **Преподаватель:**
  - Имеет права на чтение данных студентов
  - Загружает информацию об успеваемости студентов
  - Формирует отчеты по успеваемости
  - Может общаться с родителями/студентами старше 18 лет через форум
- **Родитель или студент старше 18 лет:**
  - Имеет доступ только к своим личным данным
  - Может отправлять запросы на изменение своих персональных данных
  - Может общаться с преподавателями через форум
- **Студент до 18 лет:**
  - Имеет доступ только к ограниченному набору данных, таким как домашние задания и оценки
- **Менеджер заказчика:**
  - Имеет доступ к данным учебных заведений и агрегированной статистике (общее количество студентов, затраченные вычислительные мощности на создание отчетов)
  - Не имеет доступа к персональным данным студентов

# Контекстная диаграмма



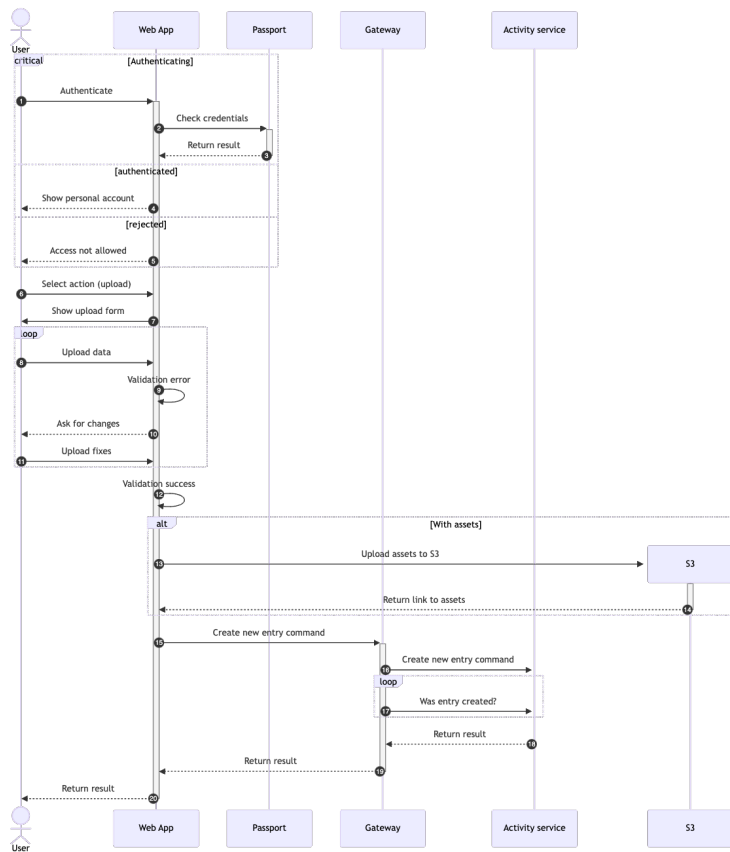




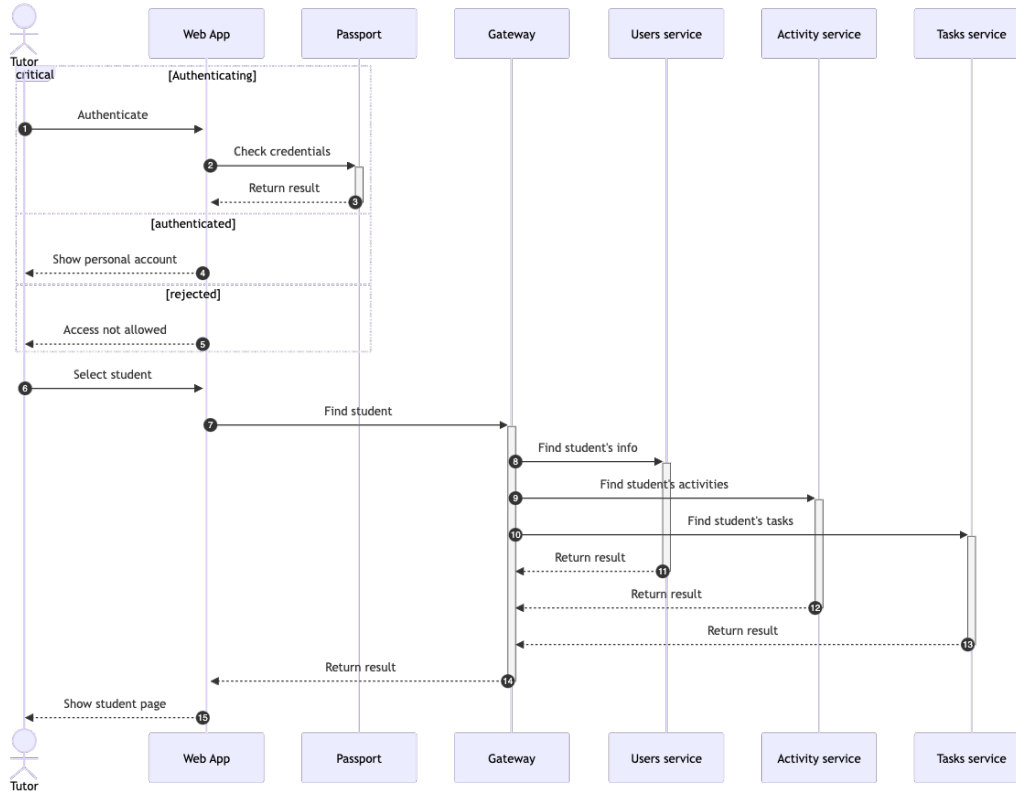
02

# Сценарии использования

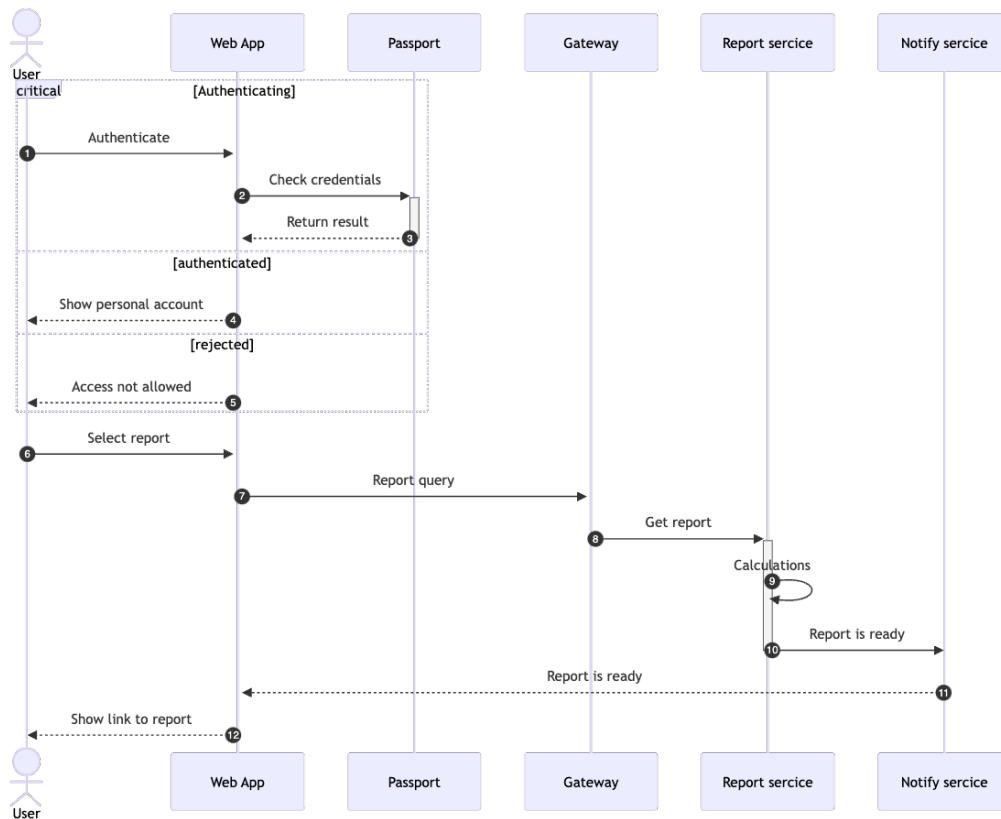
# 1. Пользователь загружает данные



## 2. Пользователь просматривает данные о студенте



### 3. Пользователь запрашивает и получает отчет



# Сценарии атрибутов качества

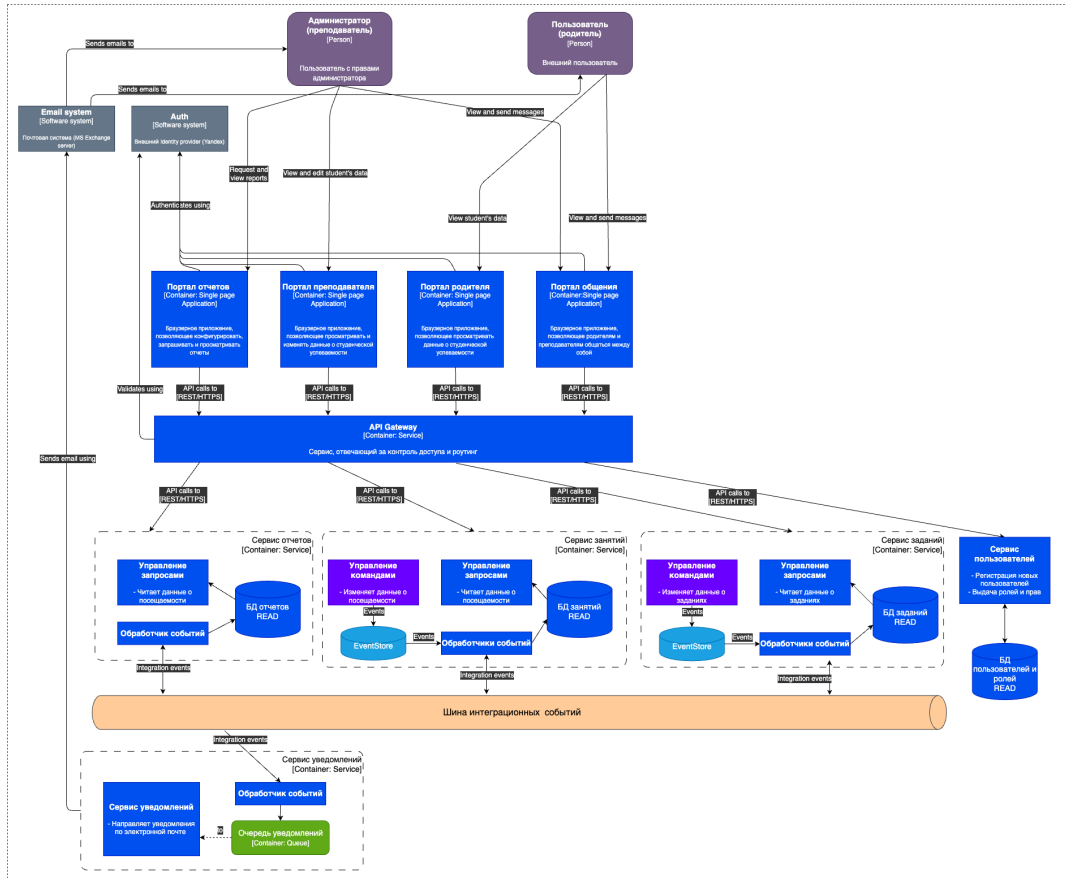
- **Гибкость системы:**
  - Быстрая разработка новых функций: новые функции могут быть добавлены в систему всего за 5 человеко-дней
  - Легкость локализации на другие языки: перенос системы на новый язык занимает 1 человеко-день при условии, что файлы с текстовыми строками уже переведены
- **Эксплуатация системы:**
  - новый релиз системы развернут и доступен всем пользователям в течение одного дня
  - систему можно перенести на новую платформу в течение одной человеко-недели
- **Взаимодействие сервисов:** после развертывания новый сервис сможет взаимодействовать с существующими сервисами с использованием шины событий без дальнейших изменений существующих сервисов
- **Производительность:** пользователь запрашивает отчет во время пиковой нагрузки (сессия) и получает уведомление о его готовности не позднее чем через 10 минут
- **Надежность:** система отслеживает количество 500 ошибок и сигнализирует дежурным о всплесках в течение 30 секунд



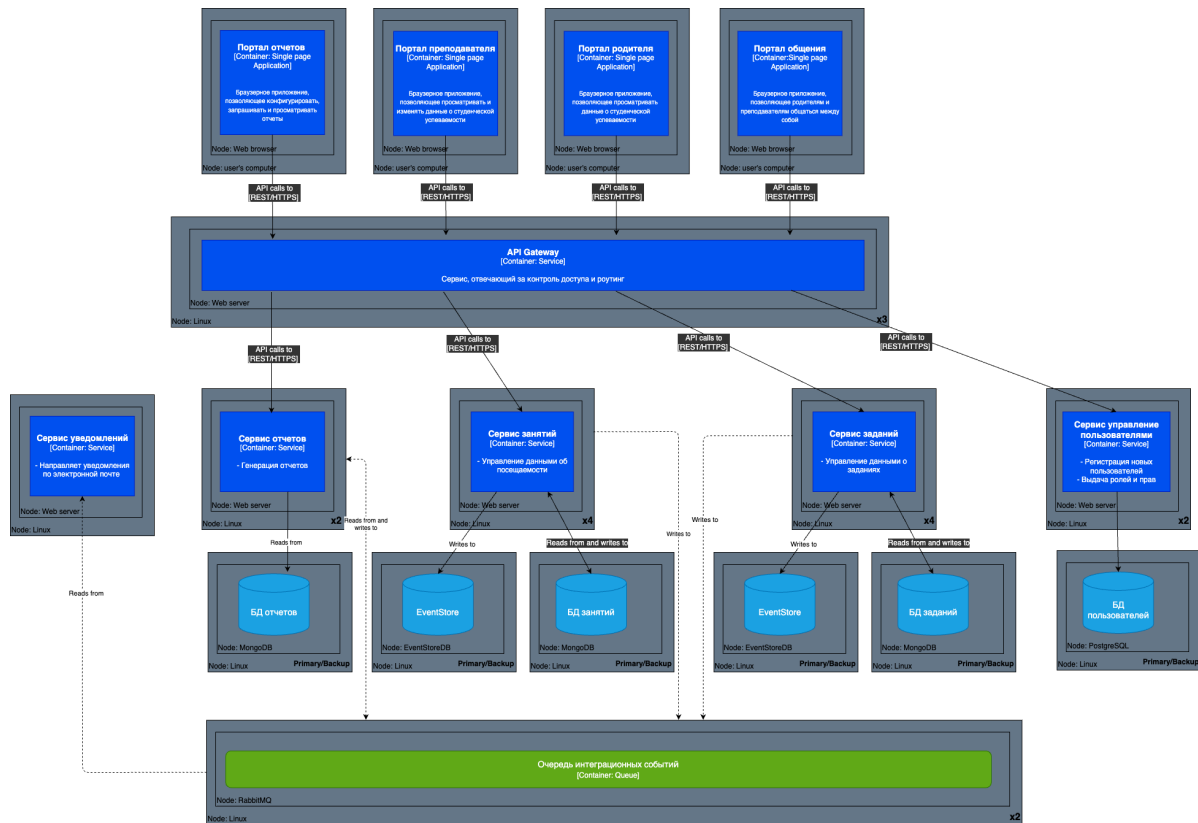
03

# Описание архитектурного решения

# Диаграмма контейнеров



# Диаграмма развертывания







04

**ADR**

# ADR 0

## Система образовательного менеджмента

### Контекст

Необходимо разработать систему образовательного менеджмента для учебных заведений, поставляемую как SaaS-продукт.

Нужно определиться как будем развертывать приложение.

### Альтернативы

0. Публичное облако
1. Гибридное облако
2. Частное облако

### Оценка и сравнительный анализ

#### Публичное облако

За - стоимость, масштабируемость

Против - недостаточный контроль над безопасностью данных

#### Гибридное облако

За - возможность тонкой настройки инфраструктуры, стоимость выше чем у полностью публичного облака, однако существенно ниже чем разворачивать приложение на собственных мощностях

#### Частное облако

За - оборудование в контуре компании, больше никто не имеет доступа к нему

Против - большое количество инвестиций в инфраструктуру, сложно масштабировать

### Решение

Гибридное облако. Персональные данные будем хранить в частном облаке, а данные о занятиях/ заданиях и вычислительные мощности - в публичном облаке.

# ADR 1

## Система образовательного менеджмента

### Контекст

Необходимо разработать систему образовательного менеджмента для учебных заведений, поставляемую как SaaS-продукт.

Нужно определиться как будем реализовывать форум для родителей и преподавателей.

### Альтернативы

- 0. Использование готовой CMS для форума
- 1. Написание форума с нуля
- 2. Использование закрытых telegram-чатов

### Оценка и сравнительный анализ

#### Готовая CMS

За - стоимость, скорость разработки

Против - непонятно как обеспечить синхронизацию аутентификационных данных с нашей системой

#### Написание с нуля

За - абсолютно точно напишем продукт, идеально подходящий под наши требования и синхронизирующийся с нашей системой

Против - стоимость и скорость разработки

#### Использование telegram-чатов

За - скорость внедрения, масштабируемость

### Решение

Будем использовать telegram-чаты для групп. Напишем бота, который будет проверять валидность списка участников (удалять участников, которые были исключены, переведены в другую группу или уволены (преподаватели)). В личном кабинете предусмотрим поле с telegram-аккаунтом, по которому будет проводиться проверка бота.

# ADR 2

## Система образовательного менеджмента

### Контекст

Необходимо разработать систему образовательного менеджмента для учебных заведений, предоставляемую как SaaS-продукт.

Определяем основную архитектуру системы.

### Альтернативы

0. Использование CRUD
1. Использование Event sourcing + CQRS

### Оценка и сравнительный анализ

#### CRUD

За - строгая консистентность данных, ниже стоимость разработки на начальном этапе.

Против - сложнее масштабировать систему, выше связность между компонентами

#### CQRS

За - низкая связность между микросервисами, проще масштабировать систему, аудит лог

Против - большое количество сущностей, сложнее дебажить.

### Решение

Будем использовать смешаную архитектуру. Для сервиса регистрации студентов будем использовать CRUD, так как в этом сервисе не ожидается большой нагрузки и большого количества данных. Для сервиса занятий, заданий и отчетов будем использовать Event sourcing + CQRS, так как нет необходимости в строгой консистентности данных, а также планируется развивать систему и добавлять в нее дополнительные функции (например, добавить библиотеку учебных материалов, анализировать успеваемость студентов, готовить аналитические материалы по улучшению успеваемости с помощью нейросетевых технологий). Архитектура на базе ES позволит более гибко масштабировать и расширять систему.

# Спасибо!

Есть вопросы?

[Elena.V.Makarova@gmail.com](mailto:Elena.V.Makarova@gmail.com)

+7 985 109 35 38