



## Описание

### Сервис аутентификации (Auth Service)

- **Ответственность:** Регистрация пользователей, авторизация, аутентификация.
- **Хранение данных:** Учетные записи пользователей, токены.
- **API:**
  - POST /register (регистрация по почте и паролю)
  - POST /login (вход по почте и паролю)
  - POST /oauth (вход через OAuth)
  - POST /refresh-token (обновление токенов)
- **События:** UserRegistered, UserLoggedIn.

### Сервис профилей (User Service)

- **Ответственность:** Управление профилями пользователей (редактирование, поиск).
- **Хранение данных:** Профили пользователей (никнейм, информация о себе, аватарка).
- **API:**
  - GET /user/{id} (получение профиля по ID)
  - PUT /user/{id} (редактирование профиля)
  - GET /users/search?nickname= (поиск по никнейму)
- **События:** UserProfileUpdated.

### Сервис подписок (UserSubscription Service)

- **Ответственность:** Управление подписками (добавление, удаление, подтверждение запросов).
- **Хранение данных:** Запросы на добавление в друзья, список подписок.
- **API:**
  - POST /subscriptions/request (отправить запрос)
  - POST /subscriptions/accept (принять запрос)
  - POST /subscriptions/decline (отклонить запрос)
  - DELETE /subscriptions/remove (удалить)
  - GET /subscriptions/{id} (список подписок)
- **События:** SubscriptionRequestSent, SubscriptionRequestAccepted, SubscriptionRequestDeclined, SubscriptionRemoved.

### Сервис чатов (Chat Service)

- **Ответственность:** Управление сообщениями между пользователями.
- **Хранение данных:** Сообщения, чаты.
- **API:**
  - POST /chat/create (Создание чата)
  - POST /messages/send (отправка сообщения)
  - GET /messages/chat/ (получение сообщений из чата с пользователем)
- **События:** ChatCreated, MessageSent, MessageReceived.

- **События:** `sendMessage, messageSent, messageReceived`.

**Сервис уведомлений (Notification Service)**

- **Ответственность:** Управление уведомлениями (включая пуш-уведомления для мобильных устройств).
- **Хранение данных:** Уведомления.
- **API:**
  - `POST /notifications/send` (отправка уведомления)
- **События:** `NotificationSent`.

Для хранения данных во всех сервисах будет использоваться PostgreSQL.  
**Обоснование:** Надежность, поддержка ACID транзакций.

Для асинхронного общения между сервисами будет использоваться Kafka.  
**Обоснование:** Высокая пропускная способность, масштабируемость, отказоустойчивость.