**Задание ДЗ\_9:**

Танковые бои могут проводить только зарегистрированные пользователи. Танковые бои проводятся как в рамках турниров, либо между любыми пользователями по договоренности.  
Можно посмотреть список будущих турниров, подать заявку на участие в турнире, посмотреть результаты проходящих турниров, уже прошедших.  
Участники боев получают уведомления о приглашении на турнир, решение по заявке на участие в турнире, о завершении танкового боя, о скором начале стартового боя.  
Участник танкового боя может посмотреть прошедший бой. За места в турнире участники получают рейтинговые очки.  
Каждый пользователь может организовать свой турнир. Турниры получают рейтинг, который рассчитывается на основе рейтингов его участников.  
Турнир может быть регулярным, тогда рейтинг его накапливается.  
Игрок участвует в танковом бое посредством программы, которая загружается в специальное приложение Агент.  
Для успешного решения задачи необходимо:

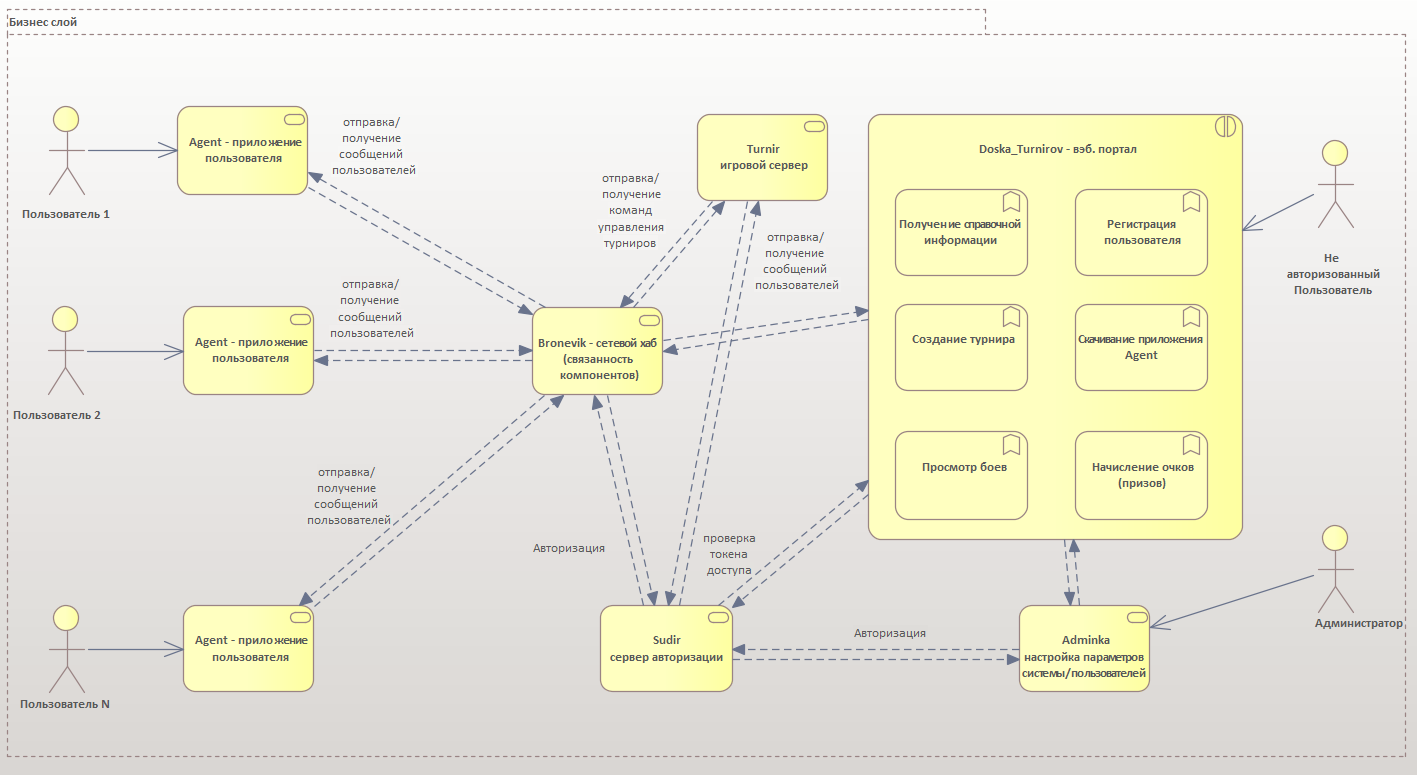
* Определить набор микросервисов и веб-приложений и отобразить их на диаграмме/наборе диаграмм.
* Определить направления обмена сообщениями и Endpoints для каждого микросервиса и приложения.
* Определить узкие места и потенциальные проблемы масштабирования приложения и способы их решения.
* Определить компоненты, к которым чаще всего будут меняться требования, и способы выполнения OCP для них.
* Сделать текстовые пояснения к созданному решению. Все документы, составляющие решение, выложить в git.

**Решение:**

1. **Определим набор микро сервисов и веб-приложений:**

* Agent - приложение пользователя.
* Sudir - сервер авторизации.
* Doska\_Turnirov - вэб. страница где производится регистрация, есть возможность скачать приложение, размещены инструкции, создаются турниры, возможен просмотр игр и справочной информации (например, начисленные очки).
* Turnir - игровой сервер.
* Bronevik - обеспечивает связанность компонентов, получение/передачу сообщений (сетевой хаб).
* Adminka - настройка параметров системы/пользователей.

1. **Отобразим микро сервисов и веб-приложений на диаграмме:**



1. **Определим направление обмена сообщениями и Endpoints для каждого микро сервиса и приложений:**

* Endpoints для Agent - приложение пользователя:
* GET/info\_a - получение информации.
* PUT /komand\_a - передача игровых команд.
* PUT /web\_komand\_a – взаимодействие с веб. порталом.
* Endpoints для Sudir - сервер авторизации:
* POST /registr\_s - регистрация.
* GET/token\_s - получение токена.
* PUT /verificat\_s - проверка токена.
* Endpoints для Doska\_Turnirov - вэб. портал:
* GET /info\_d - просмотр информации (база знаний).
* POST /turnir\_d - организация турнира (танкового боя).
* GET /show\_d - просмотр турнира.
* POST /in\_turnir\_d - запись на участие в турнире.
* GET /results\_d - просмотр результатов турнира.
* Endpoints для Turnir - игровой сервер:
* GET /verificat\_t – верификация игрока.
* GET /info\_t - получение информации от портала (турниры, правила игры, пользователи, права).
* GET / komand\_t – получение игровых команд игроков.
* PUT /param\_t – обновление параметров игры.
* Endpoints для Bronevik - сетевой хаб (связанность компонентов):
* GET /poluchit\_b – получение сообщений.
* PUT /param\_b – обновление параметров.
* Endpoints для Adminka - настройка параметров системы/пользователей:
* GET /info\_ad – получение параметров.
* PUT /param\_ad – обновление параметров.

1. **Определим узкие места и потенциальные проблемы масштабирования приложения и способы их решения:**

При масштабировании приложения возрастает количество пользователей, что вызывает повышенные требования к ресурсам микросервиса сетевой связанности (Bronevik) и игровому серверу (Turnir).

Для купирования этих рисков:

* для Bronevik должна быть предусмотрена возможность перераспределения (балансировки) нагрузки и динамического подключения дополнительных мощностей для пиковых нагрузок.
* для Turnir должен быть добавлен контроль времени отклика – на основе данного параметра автоматически выделять требуемые ресурсы и/или формировать искусственной очереди для начала нового турнира (заставка «игра начнется через 5.4.3.2.1 секунды»).

1. **Определить компоненты, к которым чаще всего будут меняться требования, и способы выполнения OCP для них:**

Одним из наиболее часто обновляемым компонентом будет приложение пользователя (Agent).

При реализации приложения Agent необходимо использовать архитектурные решения позволяющие, не менять ранее разработанный код, а создавать новые сущности для описания новых настроек, объектов и способы взаимодействия между ними (OCP). Настраивать алгоритмы с помощью стратегий. Использовать IoC-контейнеры для зависимостей.