Проект микросервисной архитектуры к игре «Танки»

В данном документе приведено описание подхода к построению микросервисной архитектуры для игры «Танки», созданного в соответствии с предъявляемыми требованиями.

Предъявляемые требования

К игре предъявляются следующие требования, которые необходимо учесть в рамках разрабатываемой архитектуры:

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

Предлагаемое решение

Схема предлагаемого решения изображена на рисунке 1.

1. Для аутентификации и авторизации пользователя предполагается использовать JWT токены, выдаваемые отдельным сервисом «Authentication Service», который позволяет аутентифицировать пользователя по его данным (пара «Имя пользователя» и «Пароль пользователя») и, в случае удачной аутентификации, выдаёт пользователю JWT токен, используемый для дальнейшего взаимодействия с прочими сервисами игры. Данный сервис может взаимодействовать с отдельно стоящей базой данных (БД), что позволяет ввести дополнительные разграничения доступа к важной для пользователя информации. Данное решение позволяет закрыть первую часть п. 1 требований, предъявляемых к игре.
2. Для получения информации о турнирах, просмотра результатов прошедших турниров, а также создания новых турниров предлагается использовать один или несколько отдельных сервисов «Tournament Service», количество которых можно масштабировать в зависимости от загрузки. Конкретный экземпляр сервиса, с которым будет взаимодействовать данный пользователь, определяется посредством предварительного вызова балансировщика. Данное решение позволит более равномерно распределять нагрузку. Для доступа к данным игр, рейтингам пользователей, а также к прочим пользовательским данным, необходимым для проведения турниров, предполагается совместное использование БД с сервисами, обеспечивающими непосредственно игровую механику.

При создании новых игр, проводимых в рамках турниров, сервис турнира опрашивает балансировщик игрового сервиса с целью получения информации о наименее загруженных сервисах для последующего создания новой игры. При завершении игры от игровых сервисов поступают результаты игры, на основании которых турнирный сервис принимает дальнейшие действия в соответствии с заложенной логикой (в частности, начислять рейтинговые очки участникам турнира).

Данный вид сервисов является одним из наиболее подверженных изменениям из-за возможности появления новых типов турниров, новых типов пользовательских рейтингов, разделения пользователей на лиги и т.д.

Предложенное решение удовлетворяет части требований, изложенных в пп. 1 – 5 требований, предъявляемых к игре.

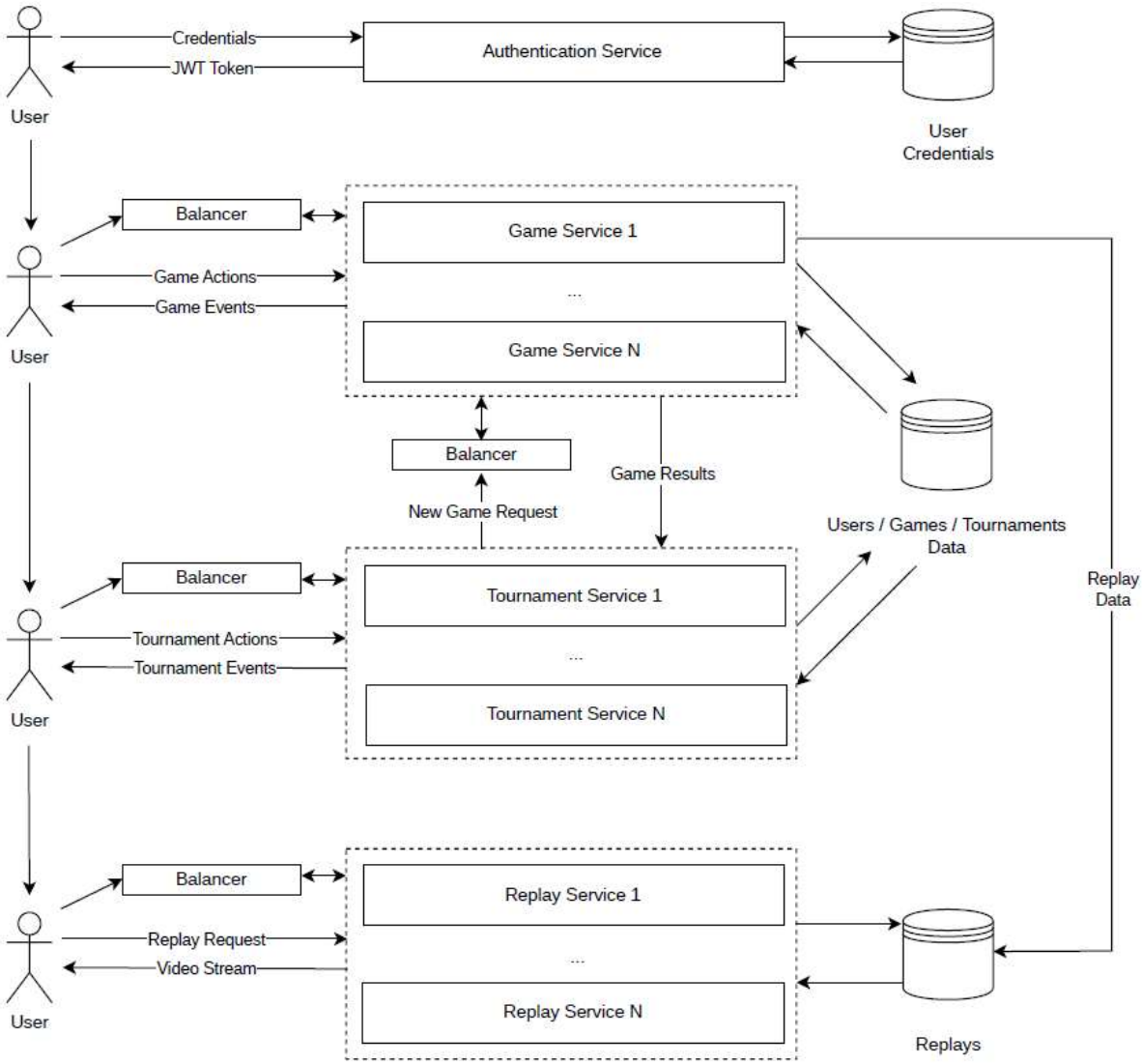

Рисунок 1 Схема разделения игры на микросервисы

3. Непосредственно для игры (выполнения пользовательских команд, расчёта происходящих в игре событий, отсылки информаций об этих событий пользователю и т.д.) используются экземпляры сервиса «Game Service». Пользователь может получать ссылку на конкретный экземпляр сервиса либо обращаясь к соответствующему балансировщику при создании боёв по договорённости или случайных боёв (см. п. 1 предъявляемых требований), либо через турнирный сервис в случае, если пользователь принимает участие в турнире. Кроме взаимодействия с пользователем и расчёта происходящих в игре событий данные сервисы отвечают за сохранение повторов игр в базу данных, из который возможно их извлечение сервисами воспроизведения повторов.

Данные сервисы являются наиболее подверженными изменениям, т.к. в игру могут добавляться новые механики, новые виды техники и т.д. Кроме того, на эти сервисы ложится наибольшая нагрузка, что повышает требования к возможности их масштабирования.

Предложенное решение удовлетворяет части требований, изложенных в пп. 1, 3 требований, предъявляемых к игре.

4. Для просмотра повторов предлагается выделить отдельный вид сервисов «Replay Service», позволяющий пользователям воспроизводить повторы проходивших ранее боёв. Для масштабирования нагрузки при взаимодействии с данным сервисом также предлагается использовать балансировщик.

Предложенное решение удовлетворяет части требований, изложенных в п. 2 требований, предъявляемых к игре.

Потенциальные проблемы и их решение

Первой потенциальной проблемой для данного приложения может стать рост нагрузки на сервисы с увеличением количества пользователей. По большей части эта проблема решается использованием балансировщиков и увеличением количества экземпляров сервисов. Кроме того, при увеличении количества запросов на просмотр повтором может вырасти нагрузка на БД и/или файловое хранилище, содержащие данные повторы. Подобная проблема может быть решена посредством репликации данных.

Компоненты, наиболее подверженные изменениям

Из компонент игры, наиболее подверженных изменениям, можно выделить игровые механики и правила проведения турниров. Для того, чтобы изменения, вносимые при добавлении новых правил и игровых механик, не нарушали Open-Closed Principle, предлагается:

1) использовать минималистичные интерфейсы, удовлетворяющие Interface Segregation Principle;
2) использовать адаптеры к данным интерфейсам;
3) использовать паттерн Command для добавления новых классов, описывающих поведение, характерное для новых механик.