Необходимо реализовать приложение, реализующее “турниры” между пользователями.

**Требование к стеку:**

* Платформа: Ubuntu
* ЯП: С++17
* Брокер очередей:  Apache Kafka
* БД: mariaDB (MySQL)

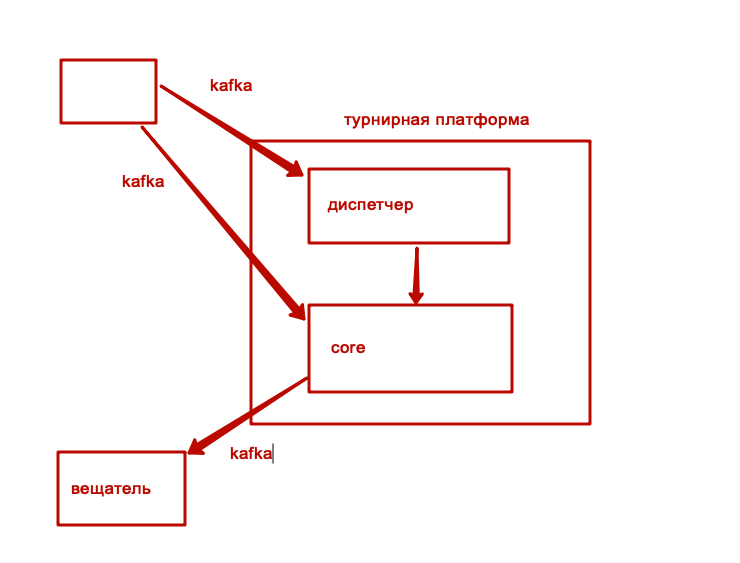
Общие компоненты (не требуется разработка)

* **Action Handler** - сервис, который получает действия игроков, авторизует эти события, и в нужном формате помещает сообщения о действиях в кафку
* **Вещатель** - сервис, который слушает кафку, и реализует WS эндпоинт, для отображения данных на сайте.

Компоненты турнирной платформы:

* **Диспетчер**   
  Один экземпляр на платформу  
  сервис, который реализует административные функции - создание турниров, редактирование, получение информации о турнире, контролирует запуск турнира, обеспечивает контроль запущенных турниров на предмет, что ядро которое должно его выполнять работает и турнир корректно работает  
  Сервис получает сообщения через кафку и выполняет целевые действия. При необходимости ответа, так же отвечает в кафку.
* **Core (ядро)**   
  Несколько экземпляров на платформу  
  Сервис на котором происходит запуск и выполнение турнира.   
  Диспетчер определяет ядро, на котором должен турнир выполняться, и уведомляет целевое ядро о том, что необходимо начать нужный турнир.   
  Ядро сообщает Вещателю таблицу лидеров турнира, при изменении таблицы.   
  Таблица лидеров может быть запрошена вещателем, в любое время, не дожидаясь обновления результатов турнира.  
  Параллельность - несколько турниров обслуживается одним core одновременно.

Схема компонентов турнирной платформы (сервиса)



Этап 1

Разработать схему БД

* таблица турниры (настройки конкретного турнира)
* таблица лидеров (для конкретного турнира)
* таблица участники (зарегистрированные участники - для турниров, где требуется предварительная регистрация)

Диспетчер

Необходимо:

* запуск турнира на конкретном core, если турнир должен начаться
* уведомление core о том, что турнир нужно запустить
* реализация watchdog (время от времени стучимся на целевой кор и определяем что турнир корректно идет)
* если турнир регулярный (повторяемый) - то необходимо реализовать создание копий турнира, для обеспечении регулярности (копия создается в момент начала турнира  - т.е. запускается турнир, у которого есть флаг регулярности, в момент запуска создается такой же турнир. со смещением во времени от момента начала турнира родителя)

Core

Необходимо:

* слушать сообщение кафки на предмет получения сообщений о действиях целевых игроков для данного турнира
* построение таблицы лидеров
* раз в N минут сохранять таблицу лидеров в БД, для того, чтобы если кор упадет, чтобы можно было продолжить турнир с минимальной потерей данных - обсудить
* обработка команды на получение свежей таблицы лидеров
* отправка таблицы лидеров в кафку (прим: определить формат сообщения и дополнить задачу)
* подсчет “баллов” игрока, на основе одного из алгоритмов подсчета в зависимости от типа турниров

ВАЖНО: желательно стараться меньше дрочить базу, чтобы снизить на нее нагрузку, т.е. сам турнир может проходить на базе кор, в памяти, но раз примерно в минуту (или 30 сек - вынести в настройку) нужно записывать состояние турнира

Требование к логам

Каждое приложение должно подробно описывать свои действия в лог stdout в формате json, в одну строку, чтобы файлбит мог нормально их забирать

формат лога (пример):

{

  "date": "26.07.2023",

  "time": "07:36:02.395",

  "level": "trace",

  "event": "action",

  "t\_id": 123,

  "player": "player\_string\_id",

  "log": {

  }

}

Турнир

**Настройки турнира**

* ид зала, к которому принадлежит турнир
* Название
* Тип турнира (алгоритм начисления очков: **Grow better,** **Break the Bank, Spin to win**)
* Регистрация (не требуется, требуется)
* Настройки времени запуска турнира
  + Повторяемость (0, нет повторяемости, разовый турнир)
  + Время начала (дата + время)
  + Продолжительность (в часах)
  + Смещение по времени с момента старта турнира, при создании нового. (в минутах)
* Мин.ранг участника (не требуется, требуется, мин ранг)
* Набор игр участвующих в турнире (любая, либо список игр)
* Настройки приза
  + Тип призового фонда (фиксированный, динамический)
  + Тип приза (кредиты, купоны)

Примечание: каждый турнир принадлежит залу

Примечание: каждый игрок принадлежит залу

Формат входящего сообщения в кафке о действии игрока

{

   "data":{

      "bet\_data":{

         "bet":1111111111,

         "win": 22222222,

         "game":"game\_string\_id",

         "player":"myTestPlayer"

      },

      "extra":null,

      "id\_data":{

         "client\_id":1,

         "currency":"GEL",

         "hall\_id":1

      }

   },

   "from":"spin\_handler",

   "to":"jp\_core",

   "type":"spin"

}