**1. e-Champs**  
Цифровая платформа для управления спортивными организациями и соревнованиями.

[E-Champs](https://e-champs.com/?utm_source=chatgpt.com)

**Функционал:**

* Прием заявок и онлайн-оплат.
* Автоматическое создание турнирных сеток.
* Публикация результатов.

**Преимущества:**

* Упрощает процесс организации турниров.
* Интеграция с платежными системами.

**Недостатки:**

* Ориентирована в основном на турниры, может не подходить для регулярных тренировок или неформальных игр.

**2. MTGame**  
Платформа для создания спортивных турниров и лиг, сбора статистики и проведения трансляций матчей.

[Start Org](https://start-org.mtgame.ru/?utm_source=chatgpt.com)

**Функционал:**

* Гибкий конструктор лиг и турниров.
* Цифровая заявочная кампания.
* Управление площадками и расписанием.

**Преимущества:**

* Поддержка различных видов спорта.
* Инструменты для сбора статистики и трансляций.

**Недостатки:**

* Может быть избыточной для небольших сообществ или любительских игр.

**3. Юнибор**  
Цифровая платформа для организации спортивных турниров и учета профессионального роста спортсменов.

[Рунибор](https://runibor.com/?utm_source=chatgpt.com)

**Функционал:**

* Автоматизация подачи заявок и регистрации.
* Учет результатов и достижений спортсменов.

**Преимущества:**

* Объединяет федерации, спортивные школы и тренеров.
* Поддержка онлайн-турниров.

**Недостатки:**

* Может не иметь функций для организации неформальных или разовых спортивных событий.

**4. Мой Спорт**  
Платформа для управления спортивными организациями, тренерами и спортсменами.

[Мой Спорт](https://moisport.ru/?utm_source=chatgpt.com)

**Функционал:**

* Электронные дневники спортсменов.
* Управление расписанием тренировок и соревнований.
* Аналитика и отчеты.

**Преимущества:**

* Интеграция с государственными информационными системами.
* Поддержка различных ролей: спортсмены, тренеры, администраторы.

**Недостатки:**

* Ориентирована на государственные и крупные спортивные организации, может быть сложной для небольших групп.

**5. Orgeo.ru**  
Сервис для организации спортивных мероприятий с функцией онлайн-заявок и результатов.

[Orgeo](https://orgeo.ru/about?utm_source=chatgpt.com)

**Функционал:**

* Подача онлайн-заявок на участие в соревнованиях.
* Онлайн трансляция результатов с финиша.

**Преимущества:**

* Удобен для организаторов и участников.
* Поддержка различных видов спорта и регионов.

**Недостатки:**

* Основной акцент на соревнованиях, может не подходить для регулярных тренировок.

**Выводы для вашего проекта:**

Анализ этих платформ показывает, что многие из них ориентированы на организацию турниров и соревнований, часто для крупных или профессиональных сообществ. В вашем приложении можно сделать акцент на:

* **Организацию неформальных или любительских спортивных игр.**
* **Простоту и удобство использования для небольших групп или индивидуальных пользователей.**
* **Интеграцию с картографическими сервисами для выбора мест проведения.**
* **Возможность быстрого создания и управления событиями без сложных настроек.**

**1. UserService**

**Назначение**: Управление пользователями, их регистрацией, аутентификацией и профилями.

**Функционал**:

* Регистрация и вход пользователей (OAuth2/JWT для безопасности).
* Хранение данных профилей (имя, фото, интересы, предпочитаемые виды спорта).
* Управление друзьями и участниками игр.

**Технологии**:

* Spring Security для аутентификации.
* База данных PostgreSQL для хранения данных пользователей.

**2. EventService**

**Назначение**: Создание и управление спортивными событиями.

**Функционал**:

* Создание событий (дата, время, место, вид спорта, количество участников).
* Уведомления о новых событиях для подписчиков.
* Управление списком участников (запись/удаление).

**Технологии**:

* База данных PostgreSQL для хранения событий.
* REST API для взаимодействия с другими сервисами.

**3. NotificationService**

**Назначение**: Оповещение пользователей о событиях и активности.

**Функционал**:

* Отправка email-уведомлений или push-уведомлений.
* Уведомления о новых играх, приглашениях и изменениях.

**Технологии**:

* Spring Boot с использованием SMTP или Firebase для push-уведомлений.

**4. RecommendationService**

**Назначение**: Рекомендация событий и участников на основе интересов.

**Функционал**:

* Анализ интересов пользователей и предлагаемых игр.
* Алгоритм подбора подходящих событий на основе локации, вида спорта и уровня участников.

**Технологии**:

* Машинное обучение или простая фильтрация на основе базы данных.

**5. LocationService**

**Назначение**: Работа с геолокацией и выбором мест проведения игр.

**Функционал**:

* Интеграция с Google Maps API для выбора и отображения мест.
* Сохранение популярных площадок для проведения игр.
* Расчет маршрута до места.

**Технологии**:

* API Google Maps или аналогичные.

**6. ChatService**

**Назначение**: Общение между участниками игр.

**Функционал**:

* Возможность обсуждать детали игр в групповых чатах.
* Уведомления о новых сообщениях.

**Технологии**:

* WebSocket для реалтайм-обновлений.
* Redis для хранения сообщений.

**7. PaymentService (Опционально)**

**Назначение**: Работа с оплатой участия в платных мероприятиях.

**Функционал**:

* Интеграция с платежными системами (Stripe/PayPal).
* Хранение информации о транзакциях.
* Возвраты средств.

**Технологии**:

* Spring Boot с интеграцией внешних платежных API.

**8. AnalyticsService**

**Назначение**: Анализ данных для администраторов платформы.

**Функционал**:

* Отчеты о популярности событий.
* Метрики вовлеченности пользователей.
* Статистика по видам спорта и локациям.

**Технологии**:

* Apache Kafka для передачи событий.
* PostgreSQL или ElasticSearch для хранения данных.

**9. ApiGatewayService**

**Назначение**: Центральный вход для пользователей и маршрутизация запросов.

**Функционал**:

* Обеспечение единой точки входа через REST API.
* Маршрутизация запросов к соответствующим сервисам.
* Глобальная обработка ошибок и логирование.

**Технологии**:

* Spring Cloud Gateway.

**10. EurekaServer**

**Назначение**: Регистрация и обнаружение микросервисов.

**Функционал**:

* Управление доступностью и масштабируемостью сервисов.

**Технологии**:

* Spring Cloud Netflix Eureka.

**Общая архитектура:**

* **Frontend**: React/Angular для клиентского приложения.
* **Backend**: Java + Spring Boot.
* **База данных**: PostgreSQL (основная), Redis (кеширование).
* **Координация**: Eureka Server + API Gateway.
* **Обмен событиями**: Kafka для асинхронного взаимодействия между сервисами.
* **Облачное хранилище**: Amazon S3 или аналог для фотографий/файлов.