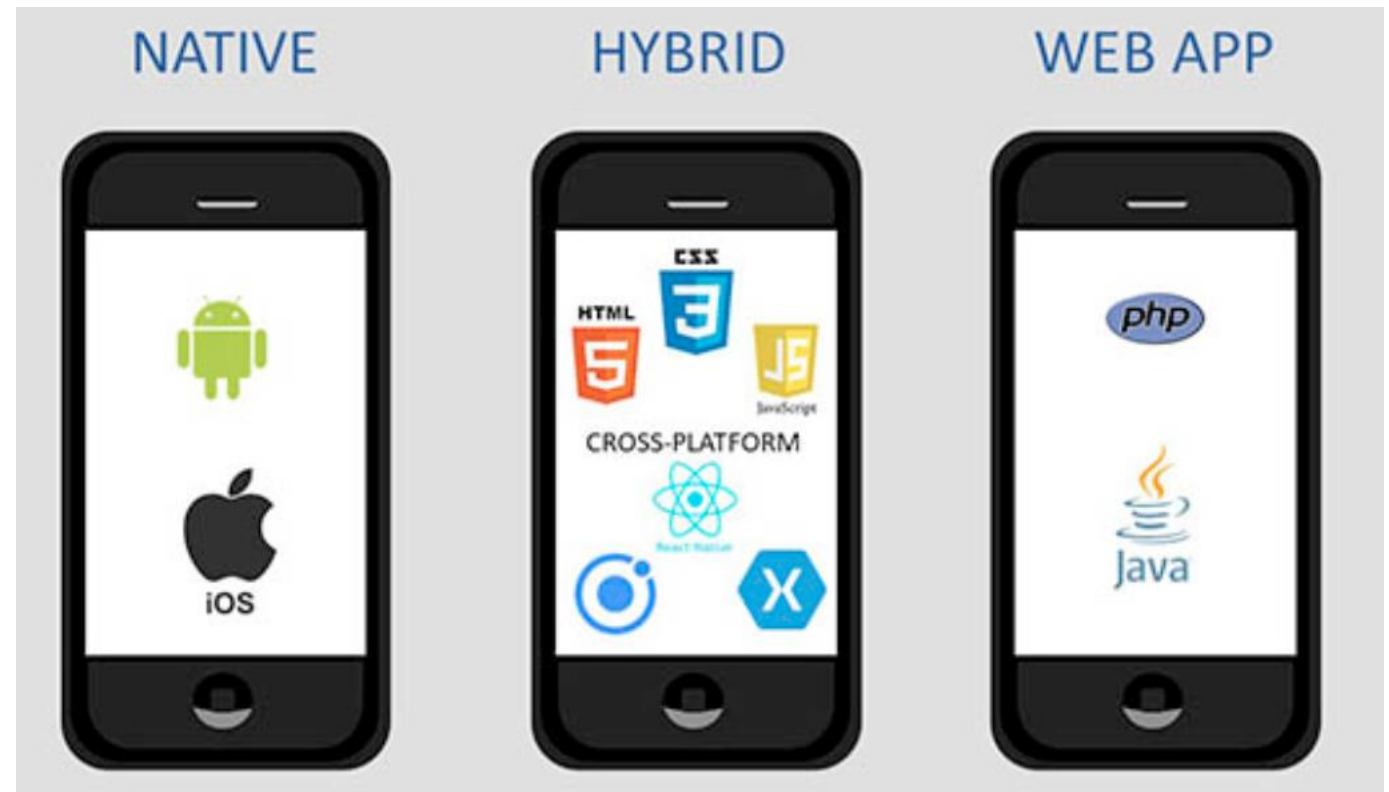


Архитектура спортивного мобильного приложения

Выбор типа приложения

- Web-приложение отпадает т. к. нужно подключать датчики
- Гибридные нежелательны, т.к. предполагается работа в сети с низким уровнем подключения
- Кросс-платформенные приложения нежелательны из-за ограниченного функционала и поддержки фреймворков
- Выберем нативные приложения для Android и iOS



Выбор облачного подхода

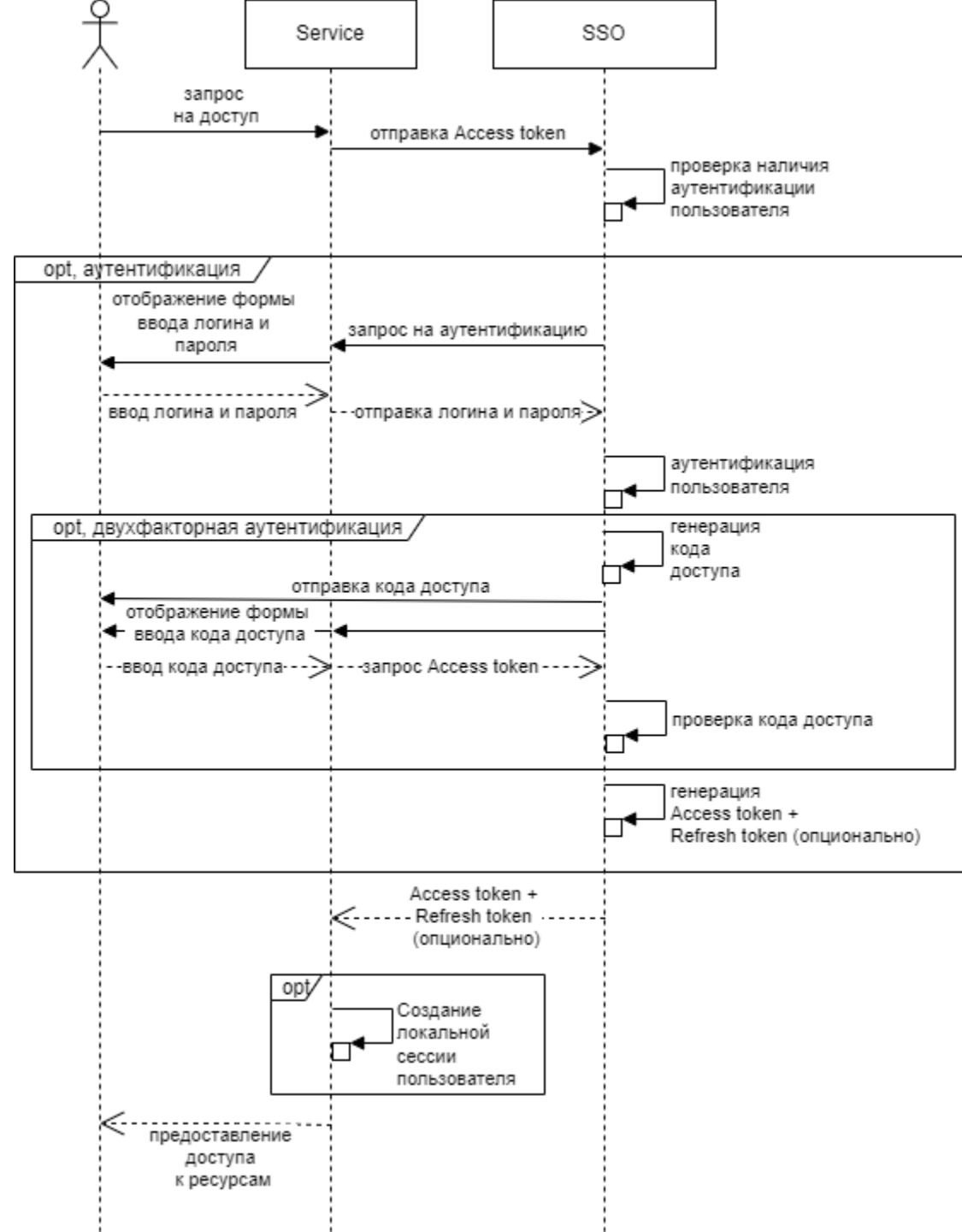
	PaaS	APaaS	IaaS	Гиперконвергентны е решения
Надежность	Высокая надежность определяется инфраструктурой и качеством операционной поддержки со стороны провайдера		Высокая надежность определяется инфраструктурой и качеством операционной поддержки со стороны провайдера	
Масштабируемость	Сокращение времени на выделение дополнительных ресурсов (возможность масштабирования по требованию)		Возможность быстрого наращивания и сокращения ресурсов инфраструктуры провайдеров	Возможность быстрого наращивания ресурсов за счет встроенных механизмов
Гибкость	Возможность быстрого изменения объема и набора услуг по требованию	Предоставление гибких инструментов для разработки по требованию	Возможность гибкой конфигурации используемых инфраструктурных компонентов	
Экономическая эффективность			Снижение общей стоимости владения инфраструктурой за счет эффекта «экономии на масштабе»	Упрощение процесса и снижение стоимости по мере расширения потребности в инфраструктуре
Инновационность	Возможность использования передовых тех. стеков для организации процессов разработки и тестирования	Использование инновационных технологических стеков для разработки и развертывания		Использование современных стеков ПО (виртуализация и др.)
Операционная эффективность			Снижение затрат на развитие и поддержку ИТ-инфраструктуры (задачи на стороне провайдера)	Более простое управление за счет уменьшения числа управляемых систем

Выберем IaaS как легко переносимую и финансово оптимальную модель без vendor lock

Интеграция приложений через SSO

SSO позволяет осуществлять бесшовный переход между приложениями, при этом пользователю не нужно вводить логин/пароль. SSO – единая точка отказа по этому требует более высокой доступности. Использует OAuth 2.0. JWT токен доступа и Refresh токен. При каждом запросе сервисы проверяют токен в SSO. Если токен истек, но остался Refresh токен то пользователь может получить новый токен доступа в SSO без ввода логина и пароля.

Для перехода на конкретную страницу будет использована технология DeepLink

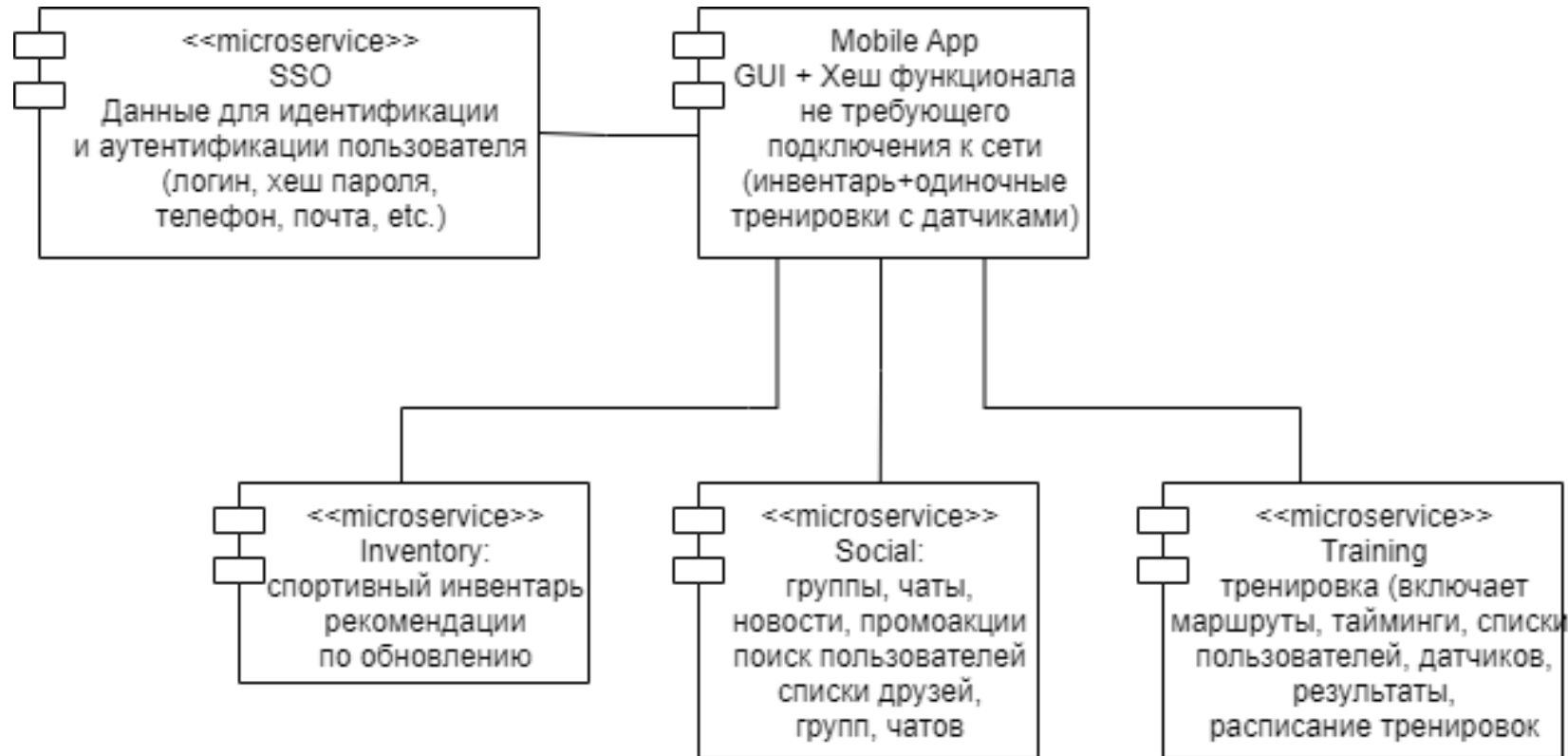


Домены и компоненты

Social – группы, чаты, поиск пользователей, уведомления, списки друзей и групп для пользователей. Работа с этим доменом требует подключения к сети.

Training – тренировки и все что с ними связано: подключаемые датчики, расписание тренировок, список участников, маршрут, таблица результатов. Работа с тренировками может производиться автономно на устройстве пользователя. Так же тренировка может быть групповой – тогда требуется подключение к сети.

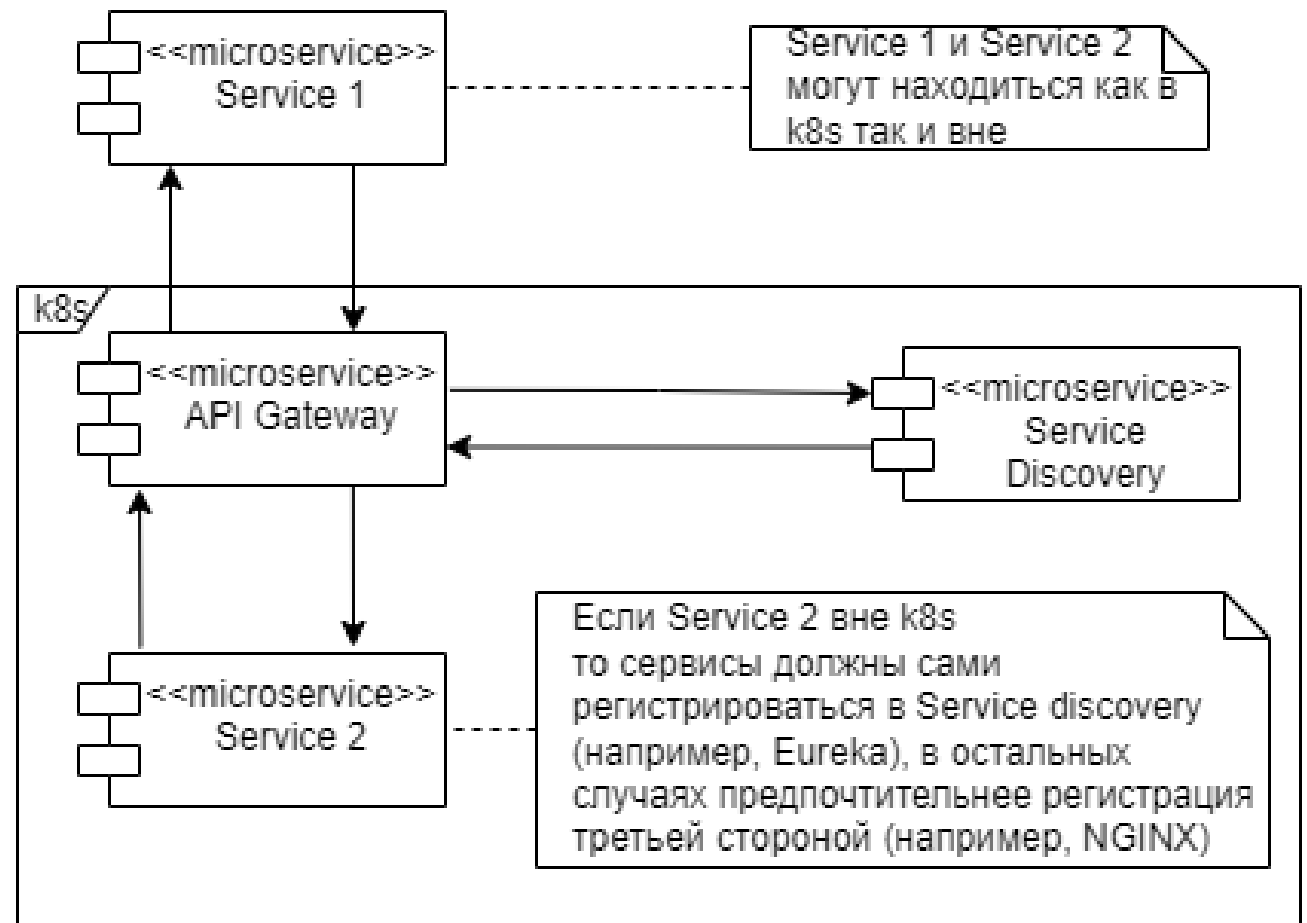
Inventory – то, что связано только с пользователем – его спортивный инвентарь.



Размещение в Kubernetes

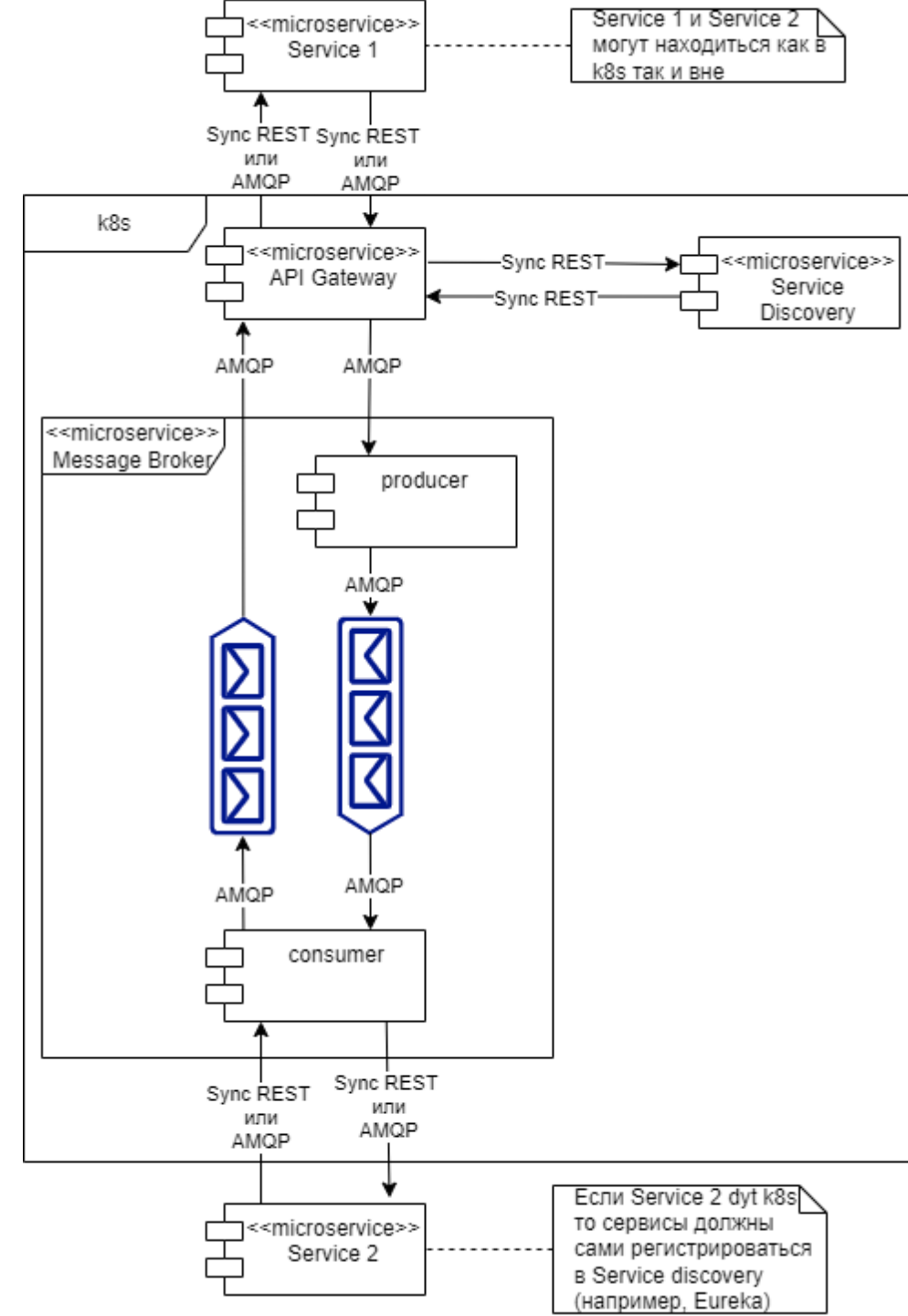
Service Discovery - Сервисы в Kubernetes имеют динамические IP (т.е. после падения сервис восстанавливается на новом IP). Service Discovery регистрирует сервисы – если сервису 1 нужно получить доступ к сервису 2, то он обращается в Service Discovery и узнает где находятся все реплики сервиса 2

API Gateway или **Ingress Controller** – выполняет роль маршрутизации запросов извне к сервисам (консолидация запросов – все запросы извне идут через единую точку входа) и выполняет балансировку нагрузки между репликами сервиса. Информацию о том, где находятся сервисы он получает из Service Discovery.



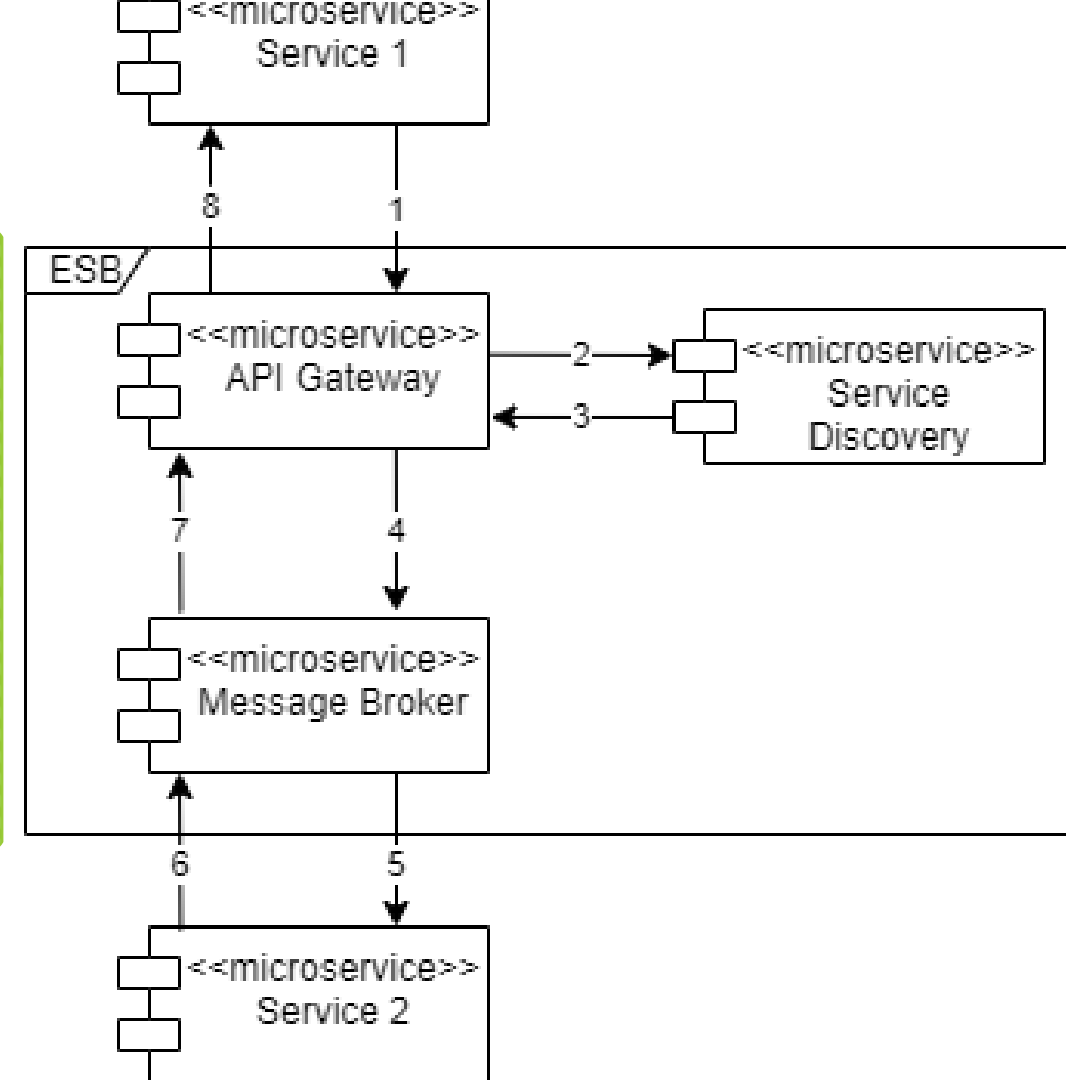
Синхронное VS асинхронное взаимодействие

Предлагается использовать брокер сообщений с гарантированной доставкой (например RabbitMQ или Apache Kafka). При этом сами сервисы могут предоставлять синхронные REST API и API Gateway может выставлять наружу синхронные REST API



Минимальный набор компонент для создания ESB

API Gateway выполняет маршрутизацию запросов согласно данным от Service Discovery, балансировку запросов, преобразование REST в AMQP. Service Discovery хранит данные о всех репликах всех доступных сервисов и отдает их по запросу. Message broker гарантирует доставку сообщений



Выбор БД и консистентность данных

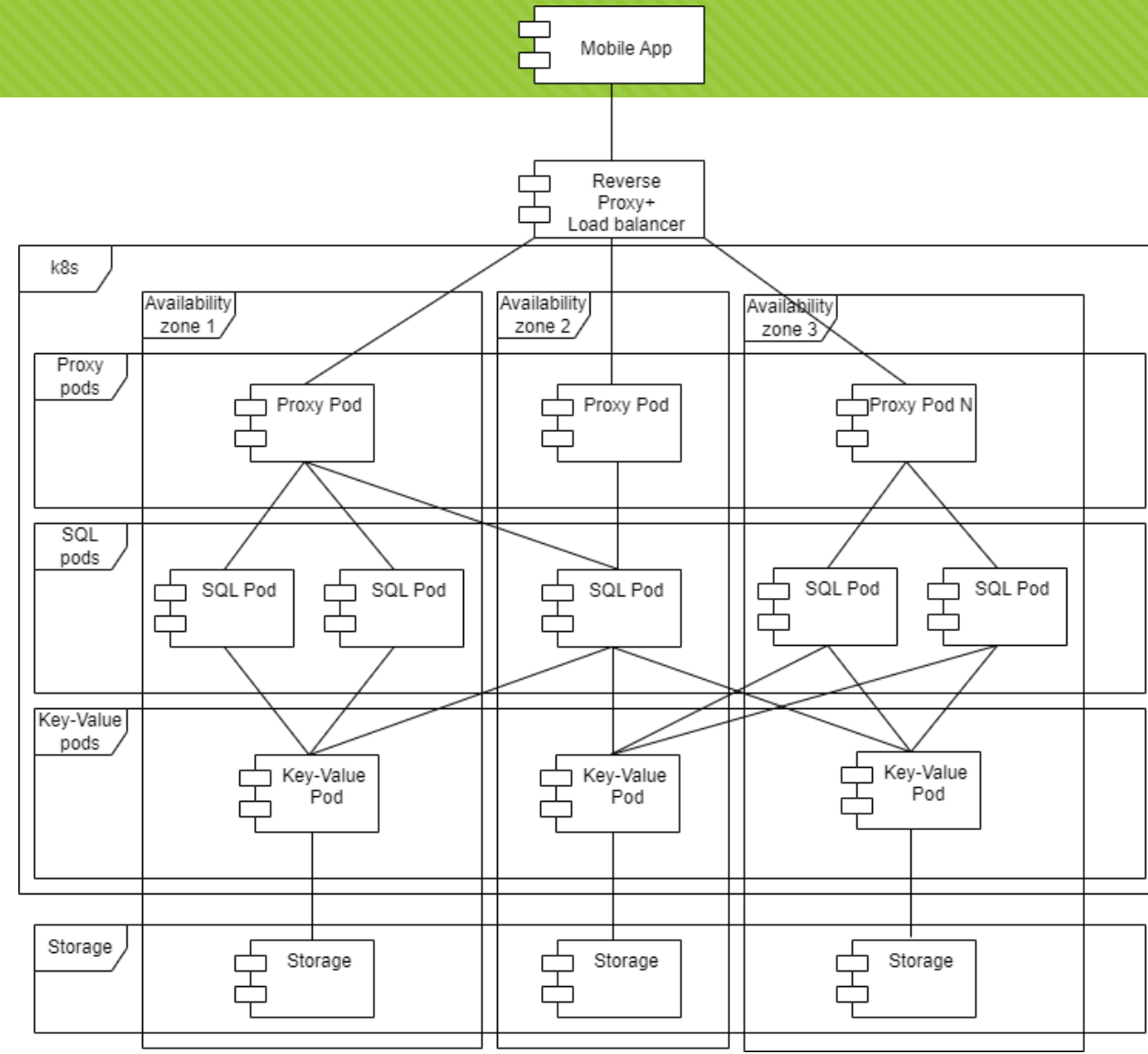
“SQL layer” выставляет наружу SQL API и конвертирует SQL-запросы в низкоуровневые запросы на чтение/запись в нижележащий “слой Key-Value”. Состоит из следующих компонент: “SQL API” – формирует пользовательский интерфейс, “Parser” – конвертирует SQL в abstract layer tree (AST), “Cost-based optimizer” – оптимизирует AST и конвертирует в логический запрос, “Physical planner” – конвертирует логический запрос в физический на одной или нескольких нодах, “SQL Execution Engine” – выполняет физический запрос – чтение/запись в следующий “Слой Key-Value”.

“Transaction layer” реализует поддержку ACID транзакций, путем координации конкурентных операций. Используется протокол коммитов “Parallel Commits”.

“Distribution layer” делает всю информацию в БД доступной из любой ноды.

“Distribution layer” – копирует информацию между нодами для поддержания согласованности. Используется протокол консенсуса Raft.

“Storage Layer” – записывает информацию на диск. Используется движок хранения информации “Pebble”. Информация хранится в формате “Ключ-значение”



Выбор стратегии Observability

Выберем использование стандарта OTL и размещение коллектора рядом с установленным ПО

Само ПО для сбора телеметрии при таком выборе роли не играет -его в любой момент можно заменить. выберем самый популярный стек технологий: Prometheus, Zabbix, Jaeger, Loki, Grafana

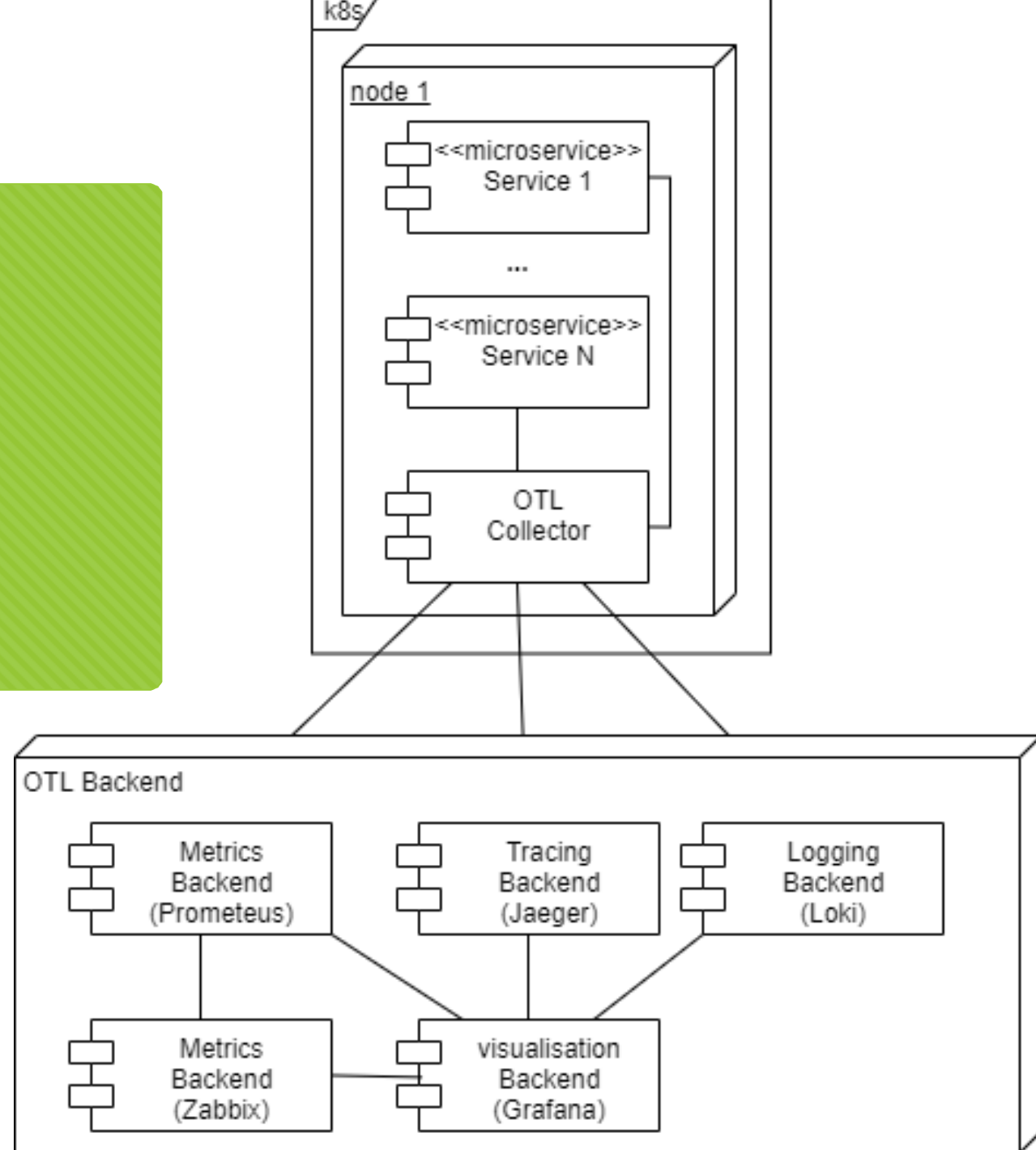


Диаграмма компонент

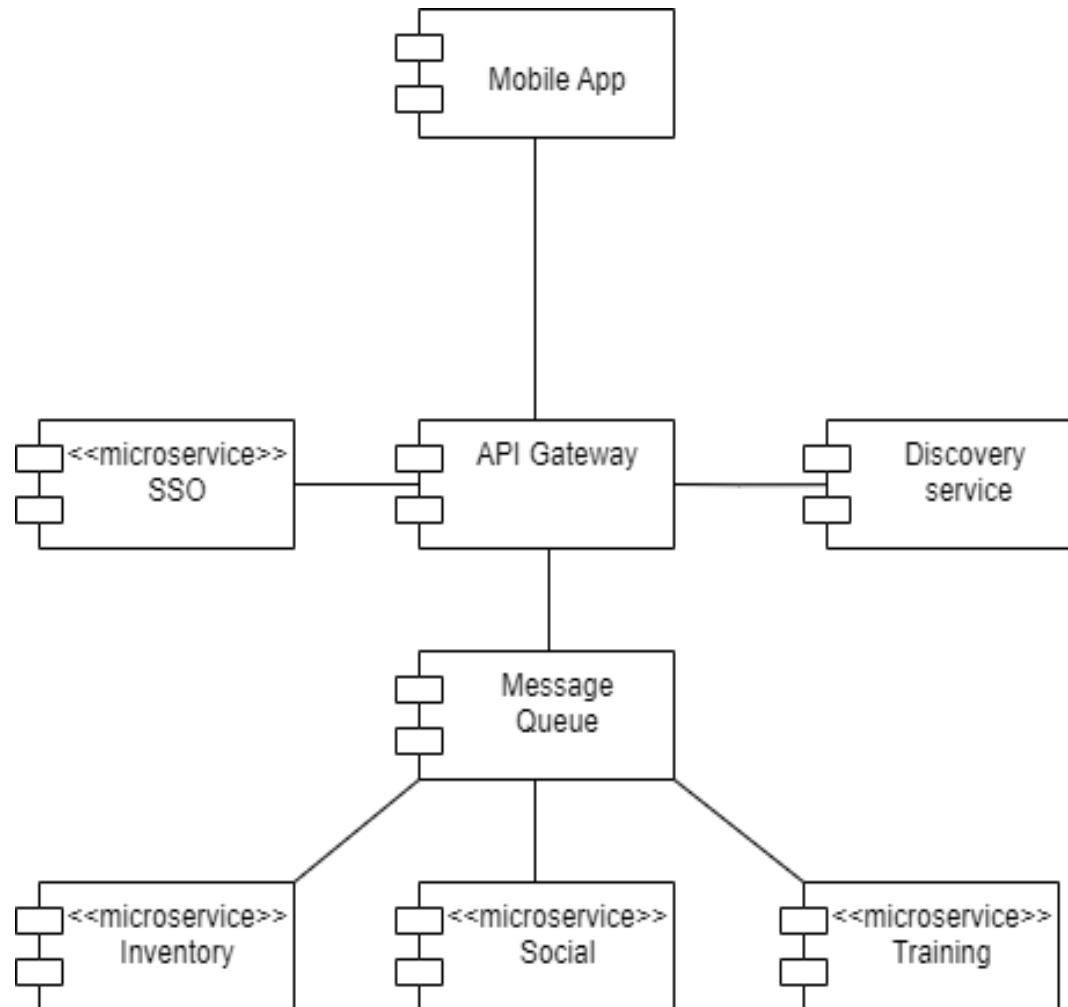


Диаграмма контекста

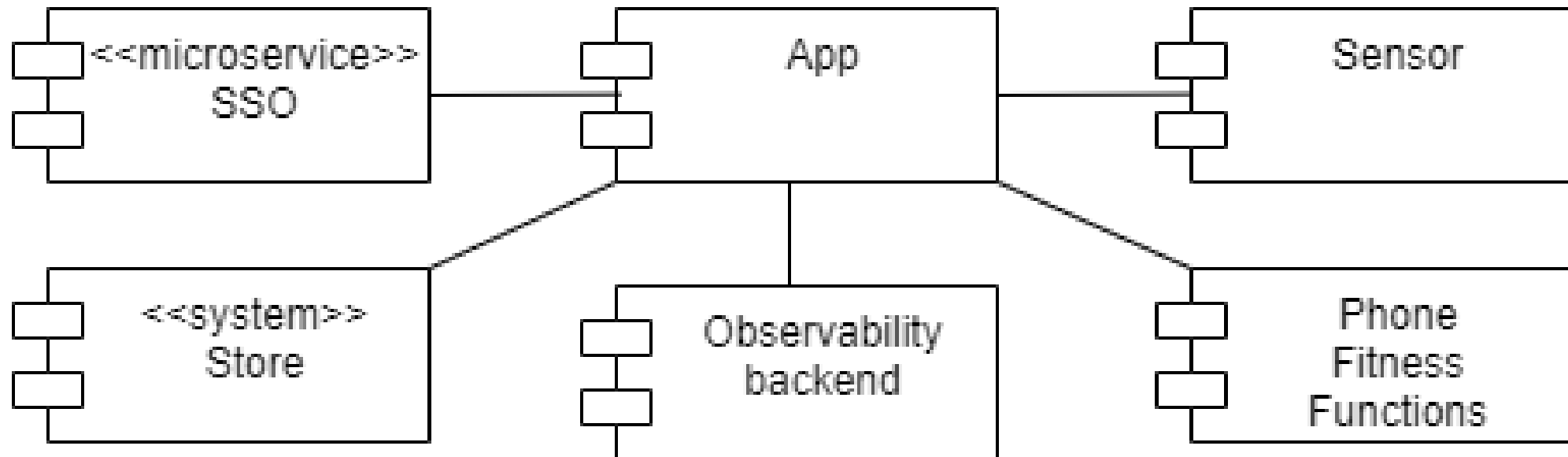
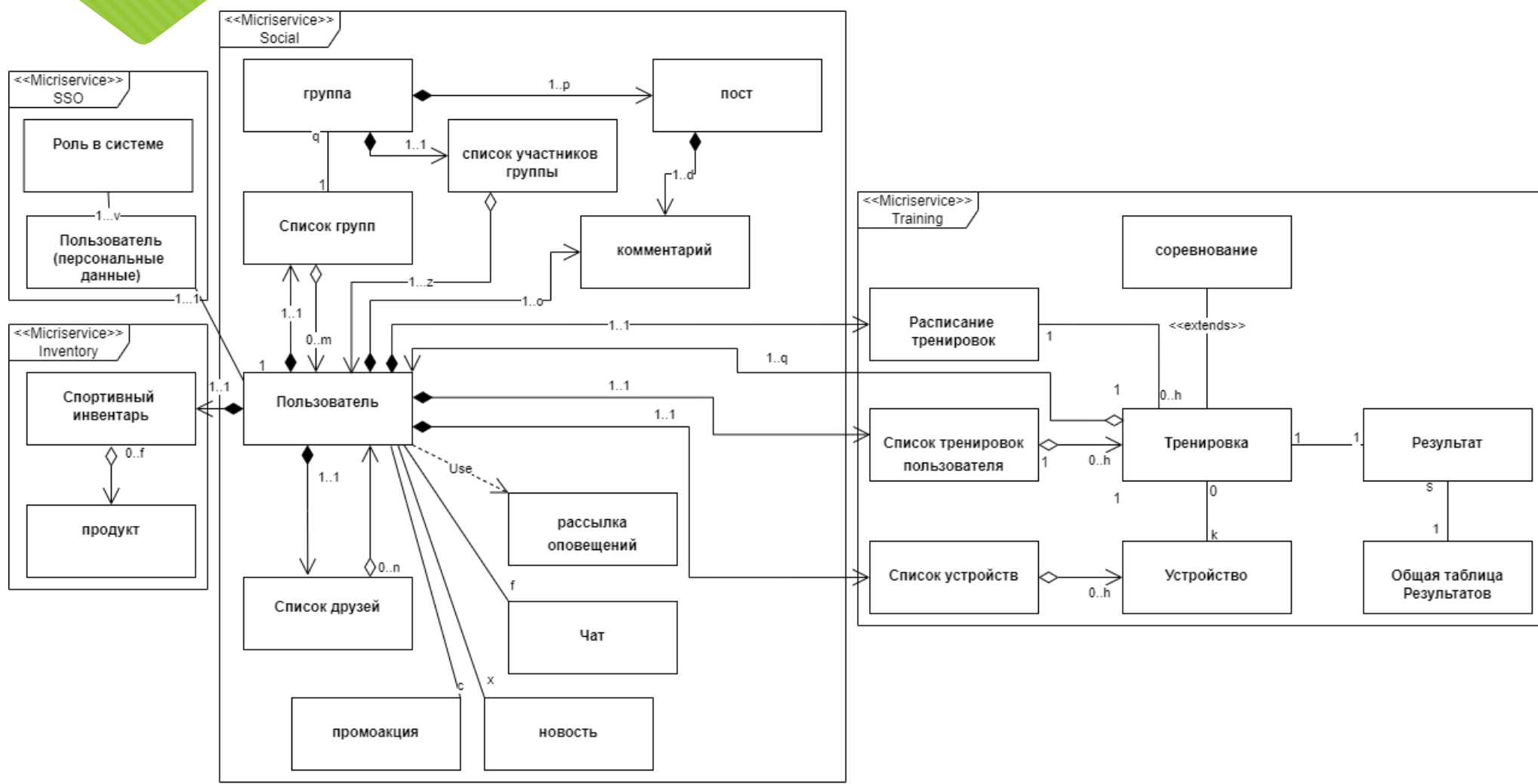


Диаграмма сущностей



Пример диаграммы последовательности

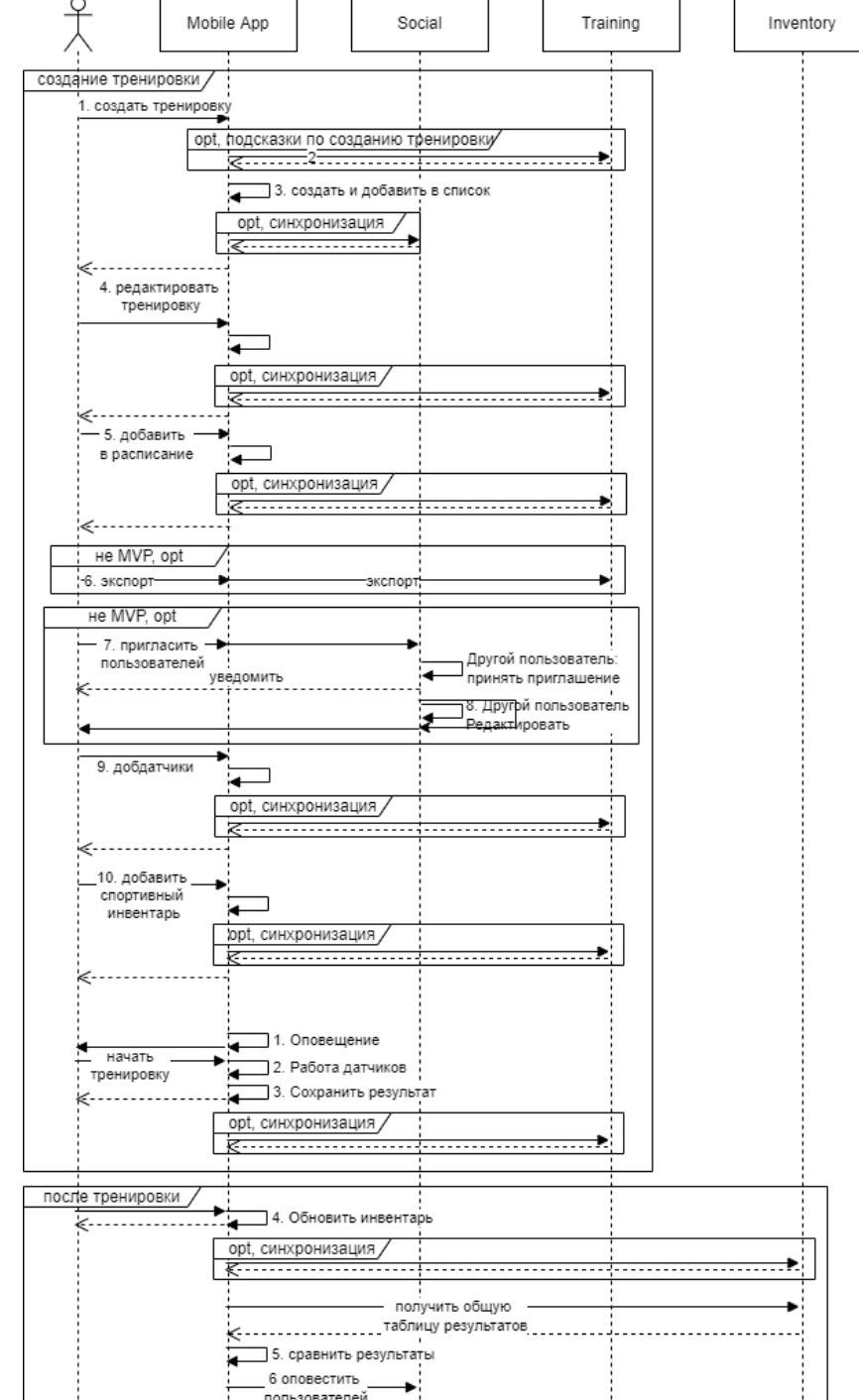
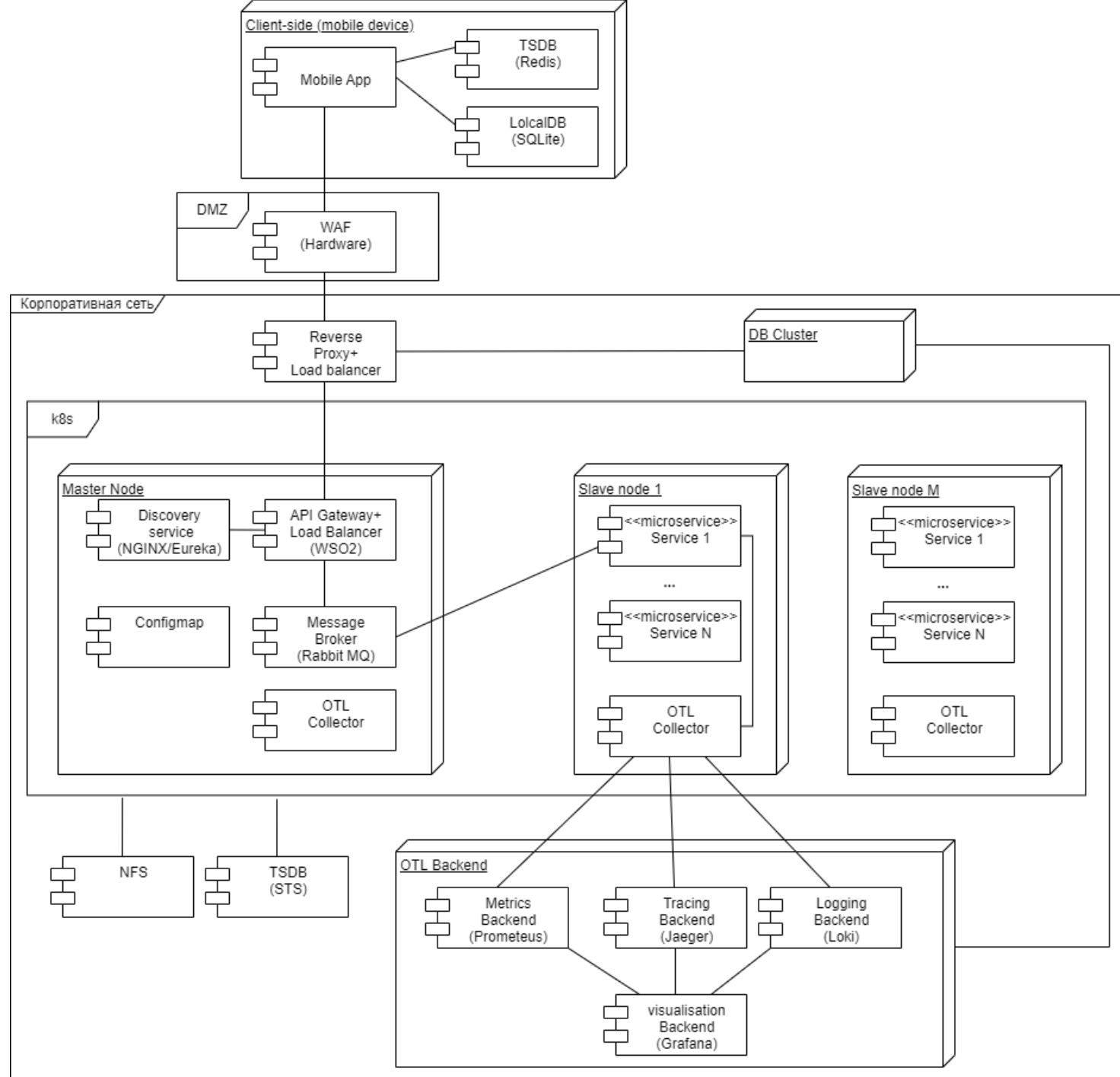


Диаграмма разворачивания

Оценка требуемых ресурсов: Одно ядро CPU в среднем дает 200 одновременных подключений, при этом для стабильной работы требуется около 1Gb оперативной памяти.

Грубая итоговая оценка всей инфраструктуры с БД:

100 CPU, 150 Gb RAM, 30Tb дискового пространства с возможностью добавить 50 CPU 100 GB RAM.



Спасибо за внимание