

## 1. Введение в дизайн микросервисов.

*Понятие микросервисной архитектуры и SOA. История появления. Отличия от ранних решений. Предпосылки и решаемые проблемы. Недостатки. Опыт Банка и причины внедрения. Дизайн микросервисов. Доменная модель. Ошибочные применения. Конвейерные архитектуры и почему их (не)надо делать на микросервисах. Транзакции в микросервисах. Паттерн Saga. Консолидация данных. Агрегаты и reporting. Event Sourcing.*

## 2. Практическая реализация микросервисной архитектуры

*Версионирование и поддержка целостности микросервисной системы. Устранение ада зависимостей на примере решения Банка. Коммуникация сервисов между собой. Почему важны стандарты и какие они бывают. CDC (client driven contracts) и канонические модели данных. Under- и Overfetching данных. GraphQL как современный язык разработки контрактов микросервисов. Многоязычные среды. Альтернативы и проблемы. Событийная модель и брокеры сообщений. Децентрализованный подход к сервисам аутентификации и авторизации. Ролевая модель и уровни доступа. Доступ клиент-сервер и сервер-сервер. Токены, API key, использование различных каналов аутентификации.*

## 3. Упаковка и развертывание микросервисов.

*Зачем нужна контейнеризация. Краткое введение в контейнеры и Kubernetes. Sidecar как модель переиспользования кода в контейнеризованных архитектурах. Три подхода к масштабированию: общие библиотеки, балансировка и подход с координатором. Проблемы решения проблем развертывания без контейнеров. Детали функционирования CI серверов и как они ограничивают процессы работы поверх виртуальных машин. Подход к управлению конфигурацией. Облачная конфигурация против начальной инициализации.*

## 4. Сборка, тестирование и процессы поставки микросервисов

*Чем отличаются CI и CD. Почему их не всегда стоит объединять и когда стоит разделять. Тестирование в микросервисных архитектурах. Функциональное, автоматизированное и на бое. Стратегия полностью зеленых тестов и тоск зависимостей. Почему не работает end to end тестирование. Стратегия непрерывного развертывания. Положительные эффекты и экономическая целесообразность.*

## 5. Организационный дизайн предприятия для эффективного использования микросервисной архитектуры.

*Стратегии масштабирования Agile. SAFe, LeSS и Spotify. Почему архитектура важна на масштабе и почему об этом нельзя не думать заранее. Влияние архитектуры на дизайн команды. Feature team и компонентные команды. Подход Inner source и совместное владение кодом.*