

Микросервисы в бизнес-приложениях

Олег Кутырин, SAP
28 октября, 2016



Об авторе

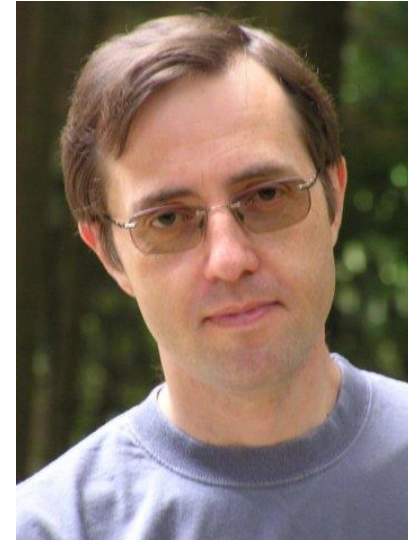
Родился и вырос в Москве

Закончил МАИ

С 2000 г. в SAP

Области интересов:

- Построение компиляторов
- Cloud
- Ruby on Rails
- Микросервисы
- Domain-Driven Design
- Бильярд (пул, 8)



<https://www.linkedin.com/in/oleg-kutyrim>
email: oleg.koutyrine@sap.com
NO TWITTER

Disclaimer

Никакого маркетинга, никакой рекламы 😊

Исключительно технология и методология

Микросервисы

Основные концепции

Что такое микросервис? (Абстрактно)

Определение М.Фаулера

“**applications** as suites of **services**”

“services are **independently deployable** and **scalable**”

”provides a firm **module boundary**”

”allowing for different services to be written in **different programming languages**”

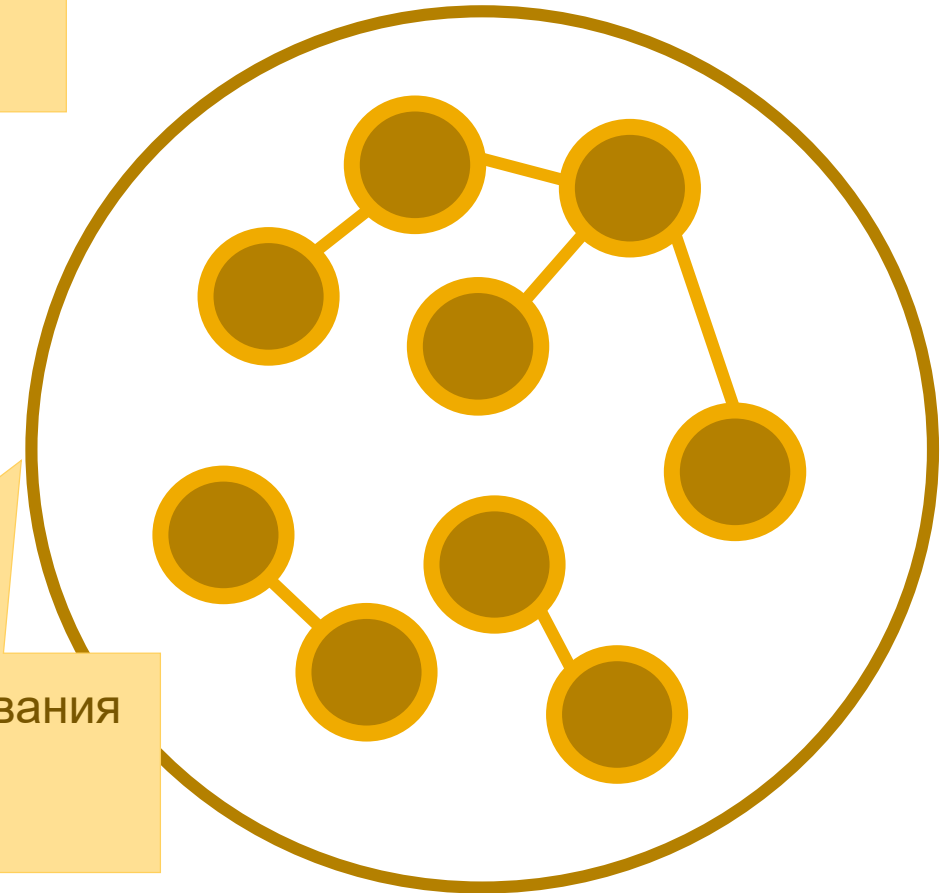
”managed by **different teams**”

<http://martinfowler.com/articles/microservices.html>

Что такое микросервис? (Чуть более конкретно)



- ✓ Один модуль развертывания
- ✓ Компактная архитектура
- ✓ Локальные коммуникации



- ✓ Множество модулей развертывания
- ✓ Распределенная архитектура
- ✓ Удаленные коммуникации

А зачем это все?

Каждый микросервис – это независимый модуль с независимым жизненным циклом

Независимая эволюция

Независимый цикл разработки

Независимое развертывание

Независимое масштабирование

Независимые технологические решения

А что в этом хорошего?

Подходящая Технология Простота
 Поддержки

Скорость Доставки

Обозримость Гибкое
Функционала Масштабирование

И все это бесплатно?

Целостность Данных Надежность
коммуникации

Общая Сложность

Производительность Требования к
Инфраструктуре

А когда оно имеет смысл?



А в общем и целом...

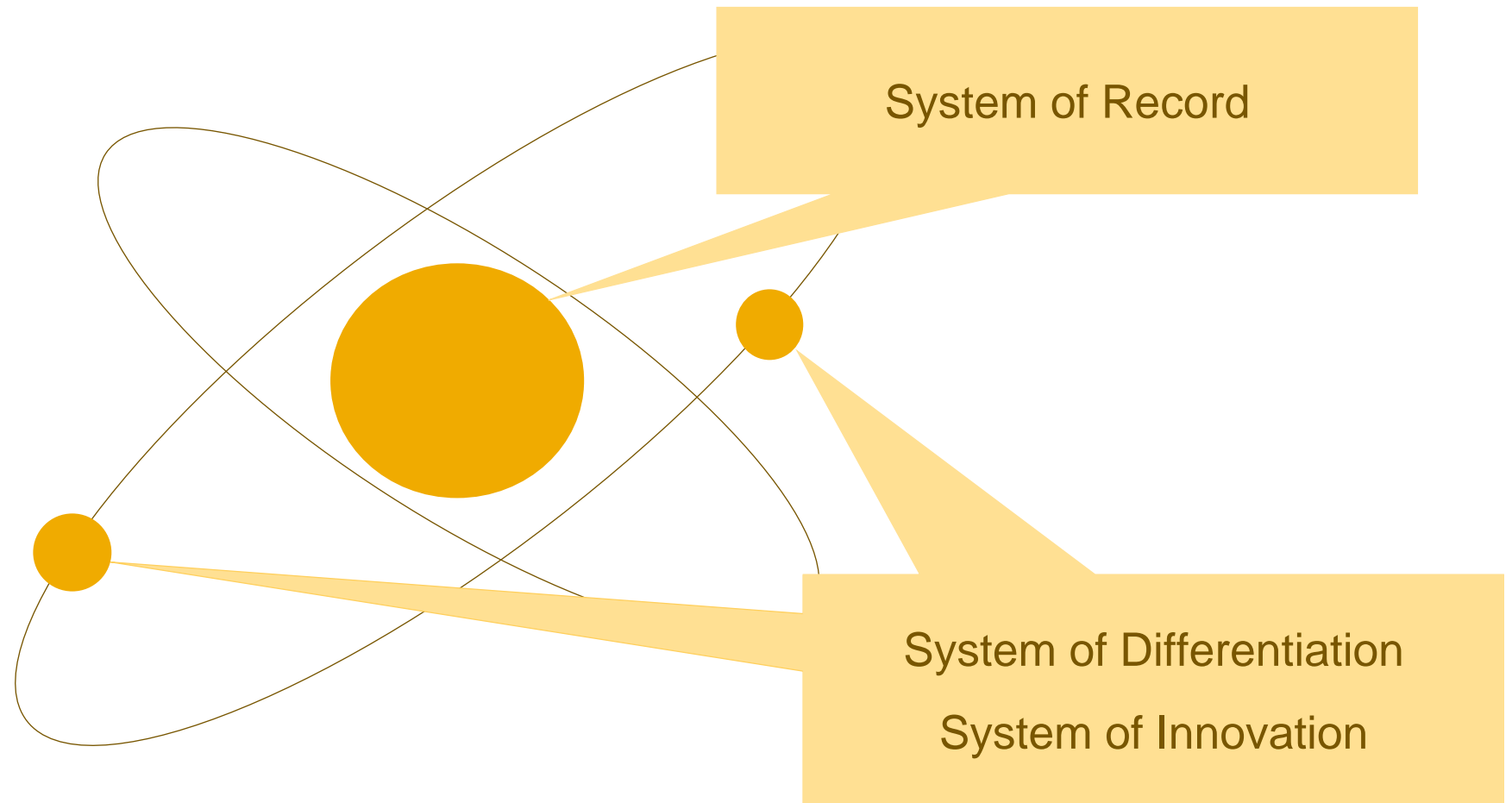
Микросервисы == Скорость инноваций

Микросервисы

Отношение к бизнесу

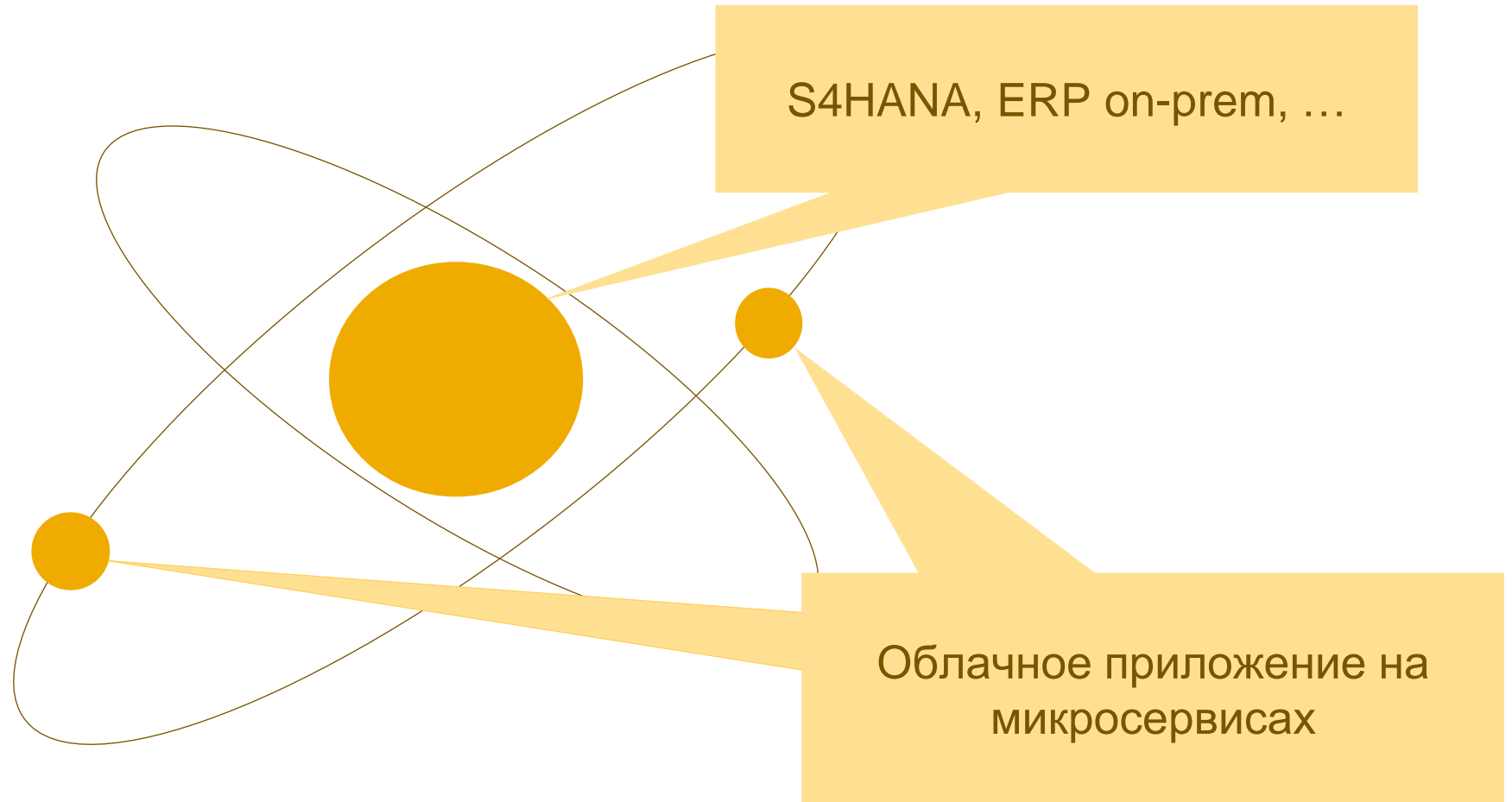


Многоуровневые приложения (Gartner 2012)

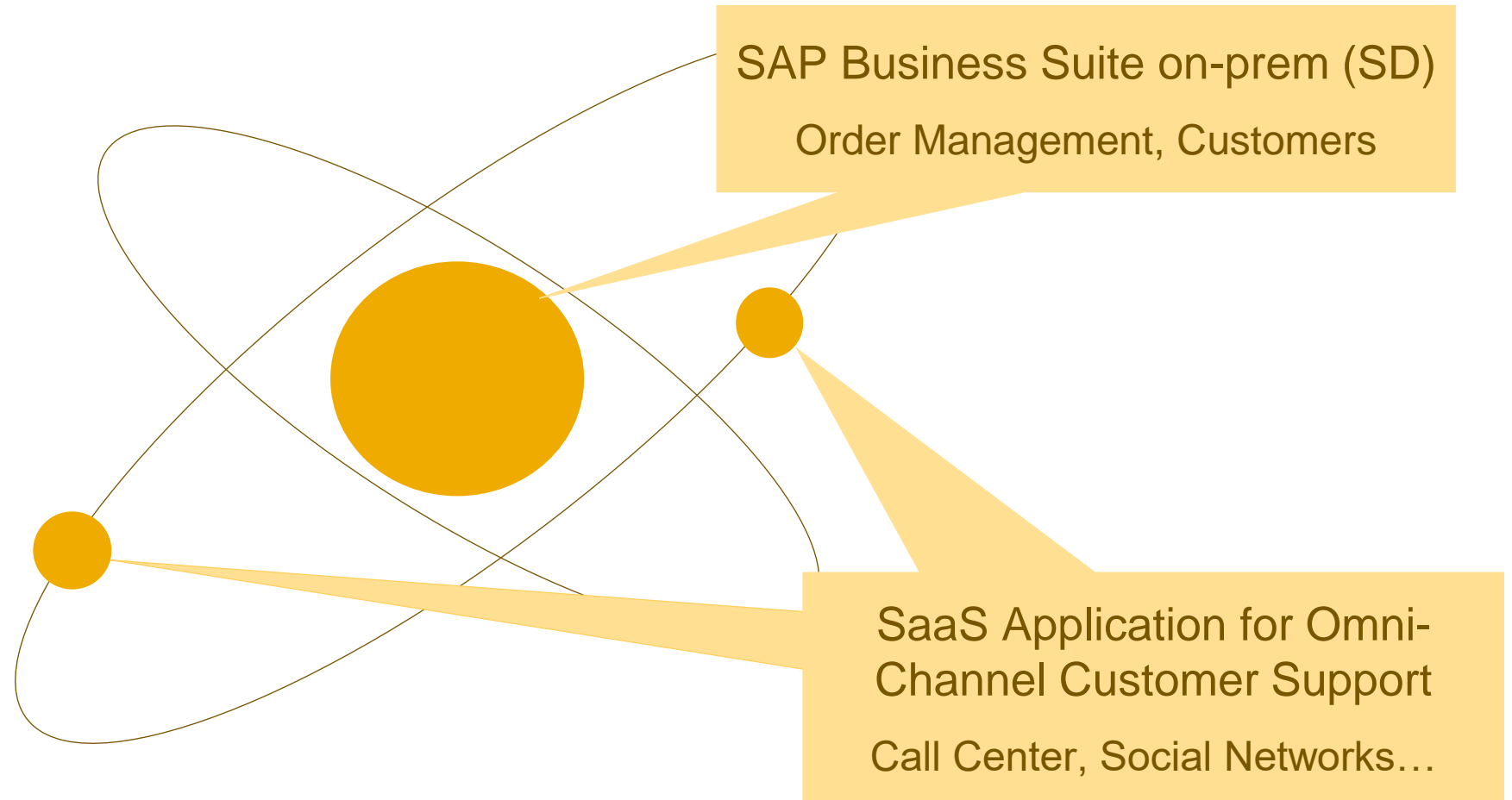


<http://www.gartner.com/newsroom/id/1923014>

Что может, например, означать...



SAP Hybris Service Engagement Center

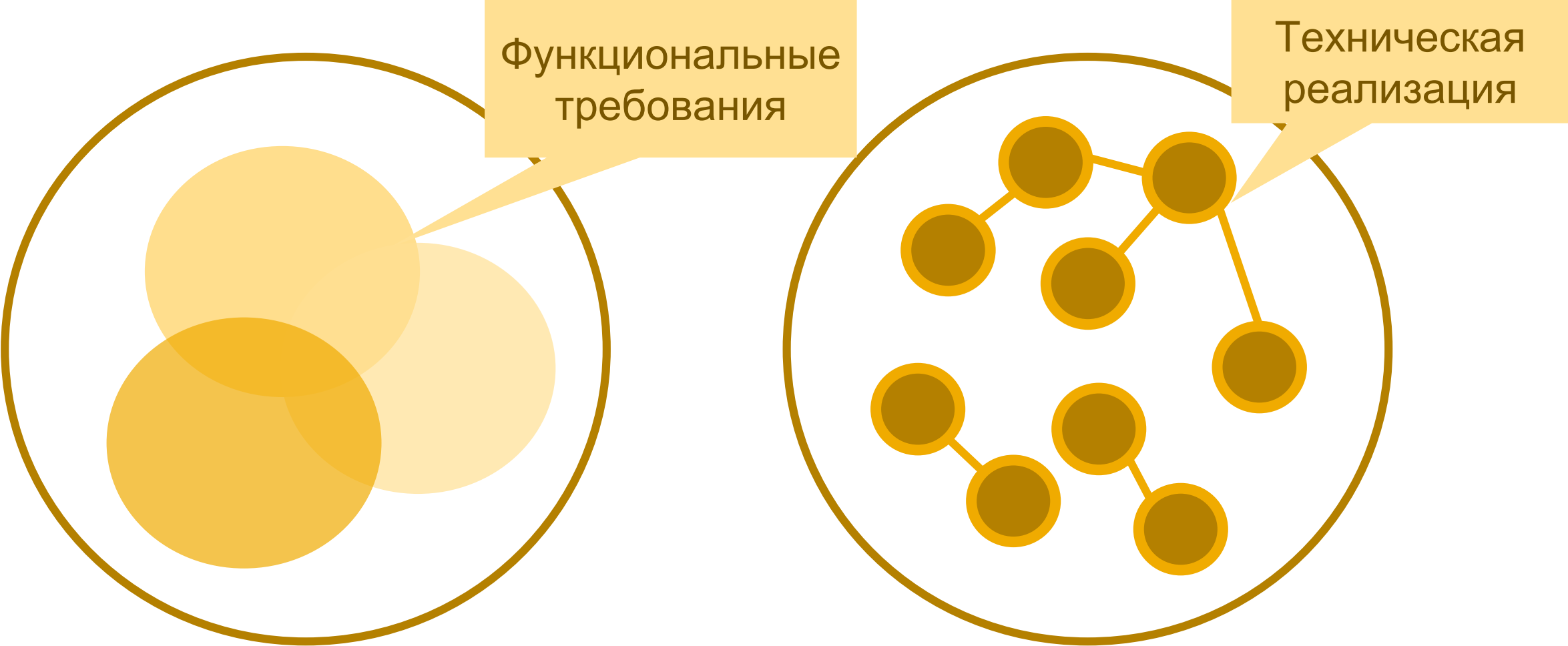


Микросервисы

Архитектура и методология



А как с этим работать?



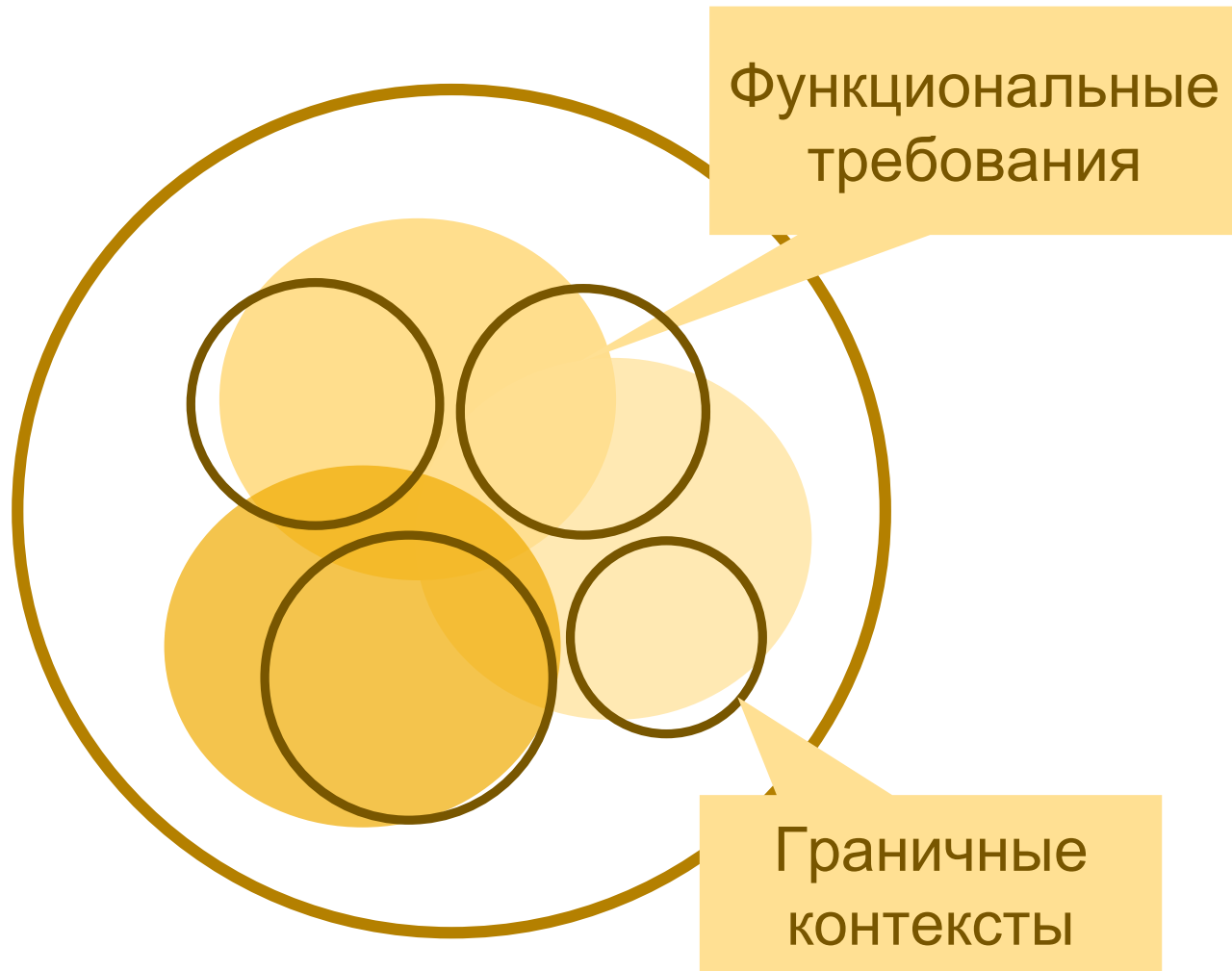
Функциональные
требования

Техническая
реализация

Принципы компонентизации

- **Функциональная нарезка**
 - Domain-Driven Design
- **Нефункциональная нарезка**
 - Особенности команд разработчиков
 - Разделение жизненных циклов
 - Повторное использование

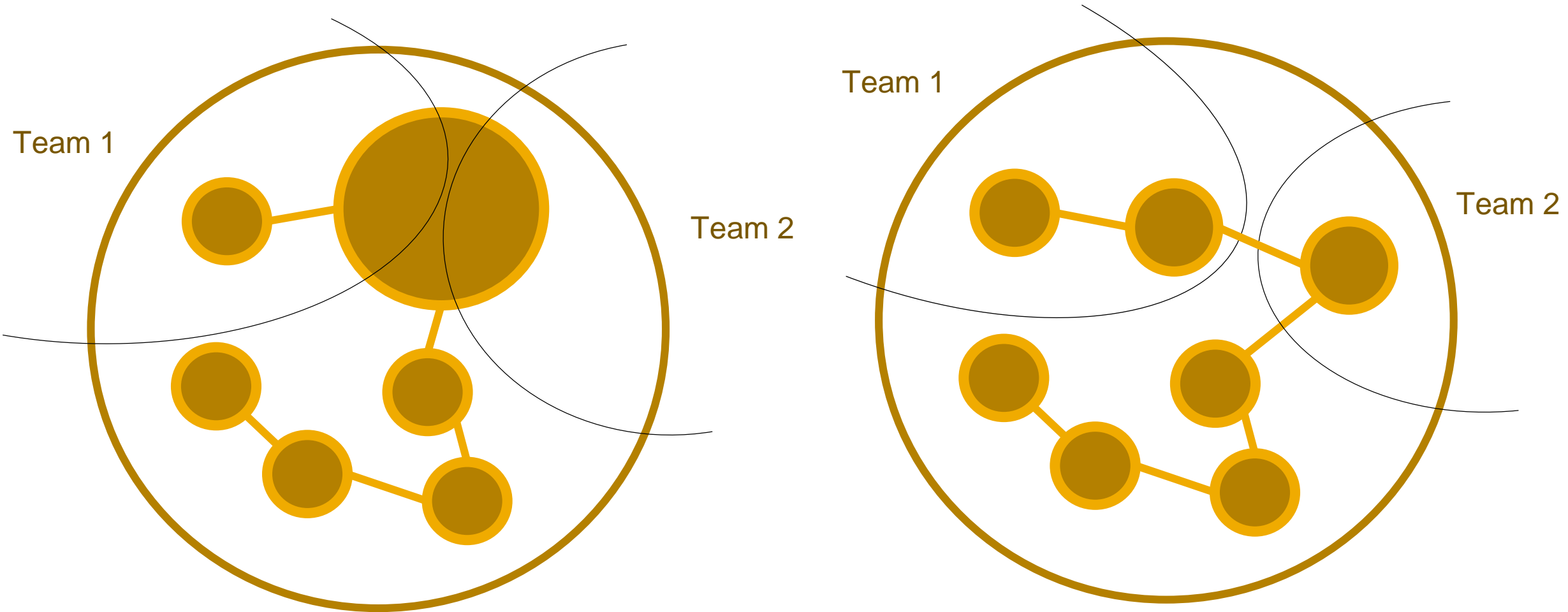
Domain-Driven Design как подход к функциональной нарезке



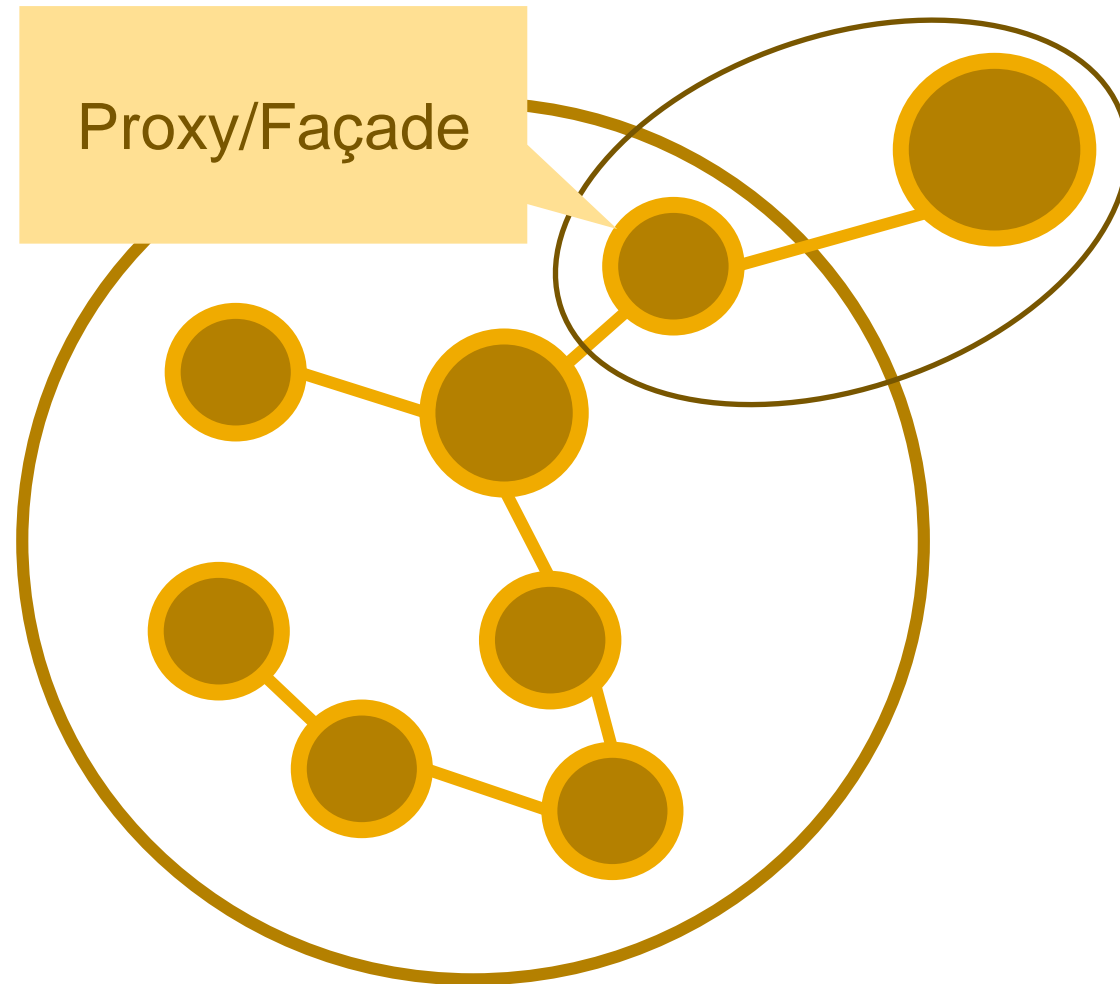
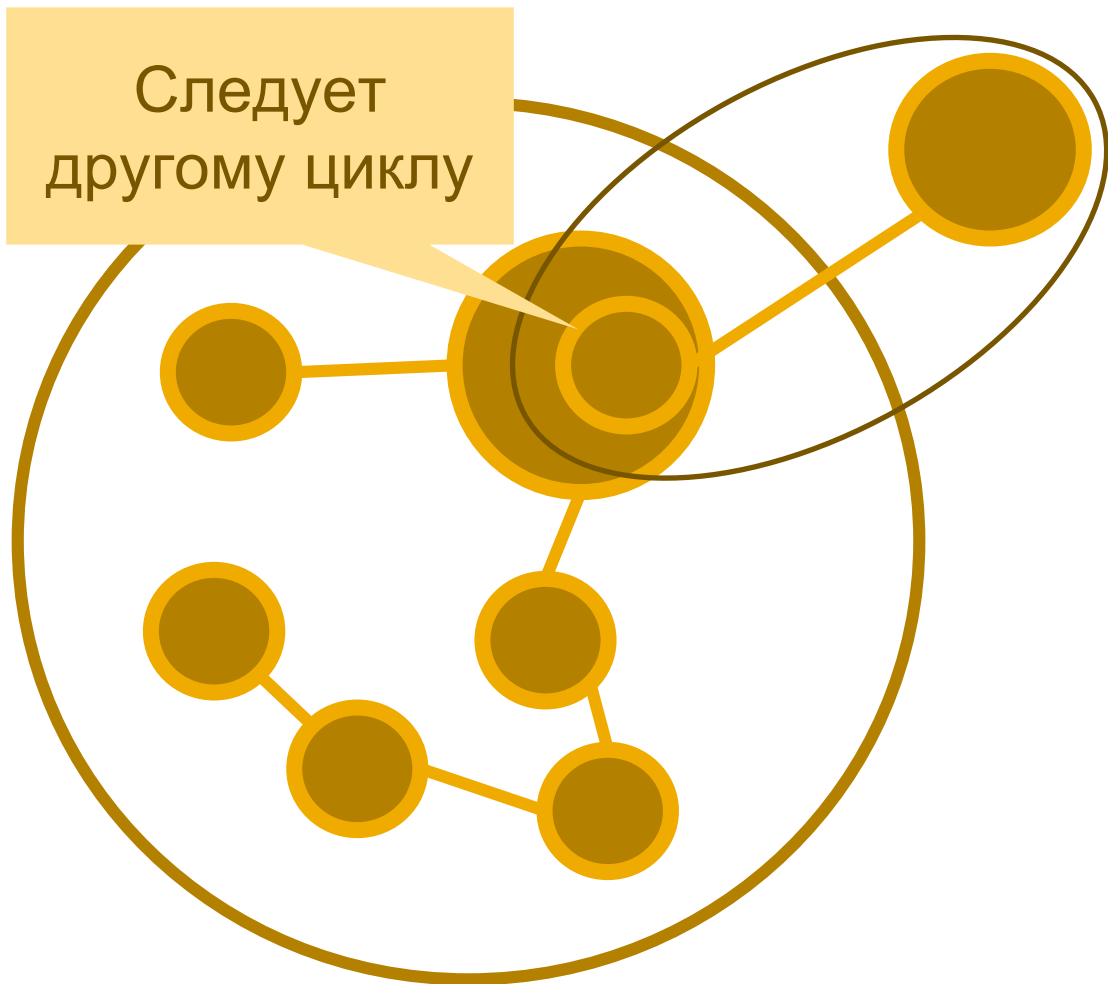
- Предложен Эриком Эвансом в 2003 г.
- Получил вторую жизнь с развитием микросервисов
- Анализ требований на основе семантики терминологии
- Домен является центральным концептом
- Граничный контекст – инструмент для выделения семантически целостных компонент

https://en.wikipedia.org/wiki/Domain-driven_design

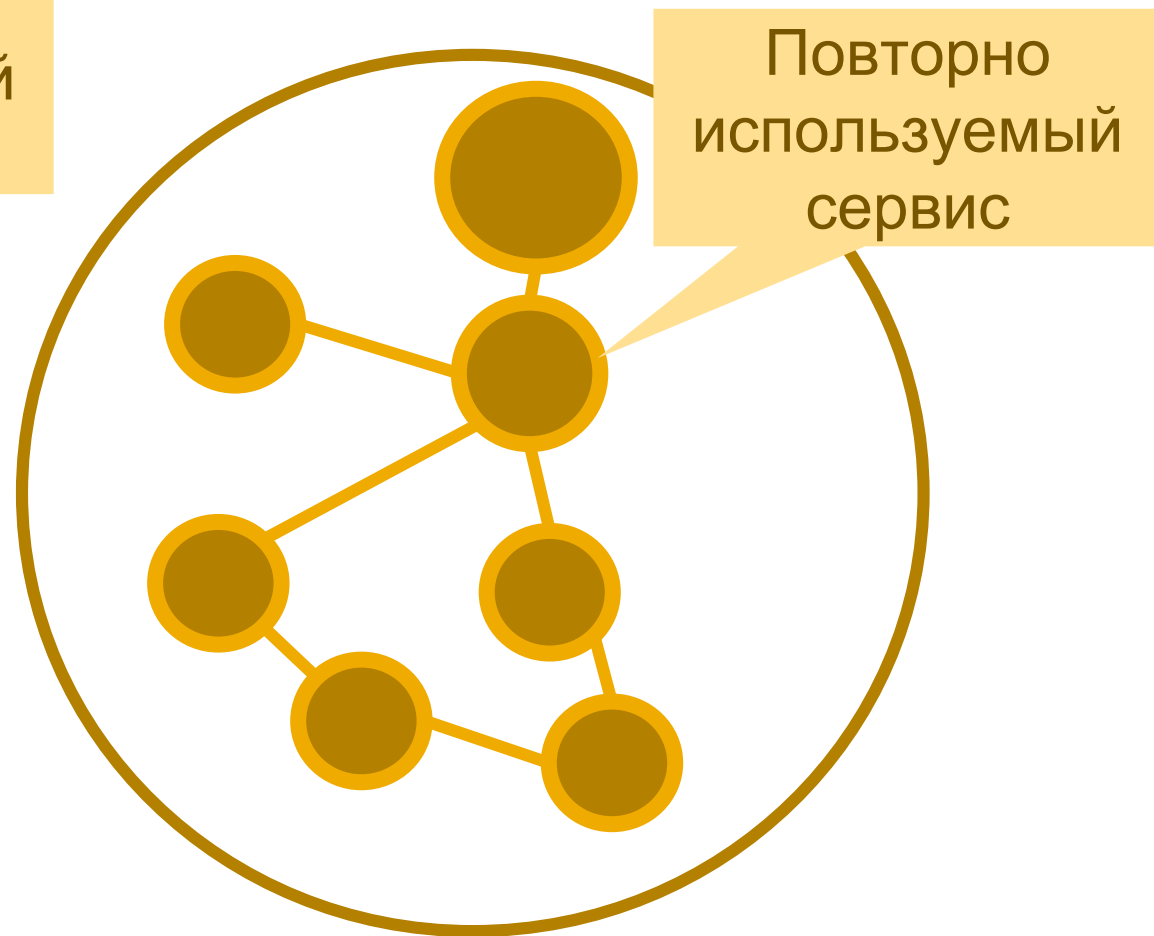
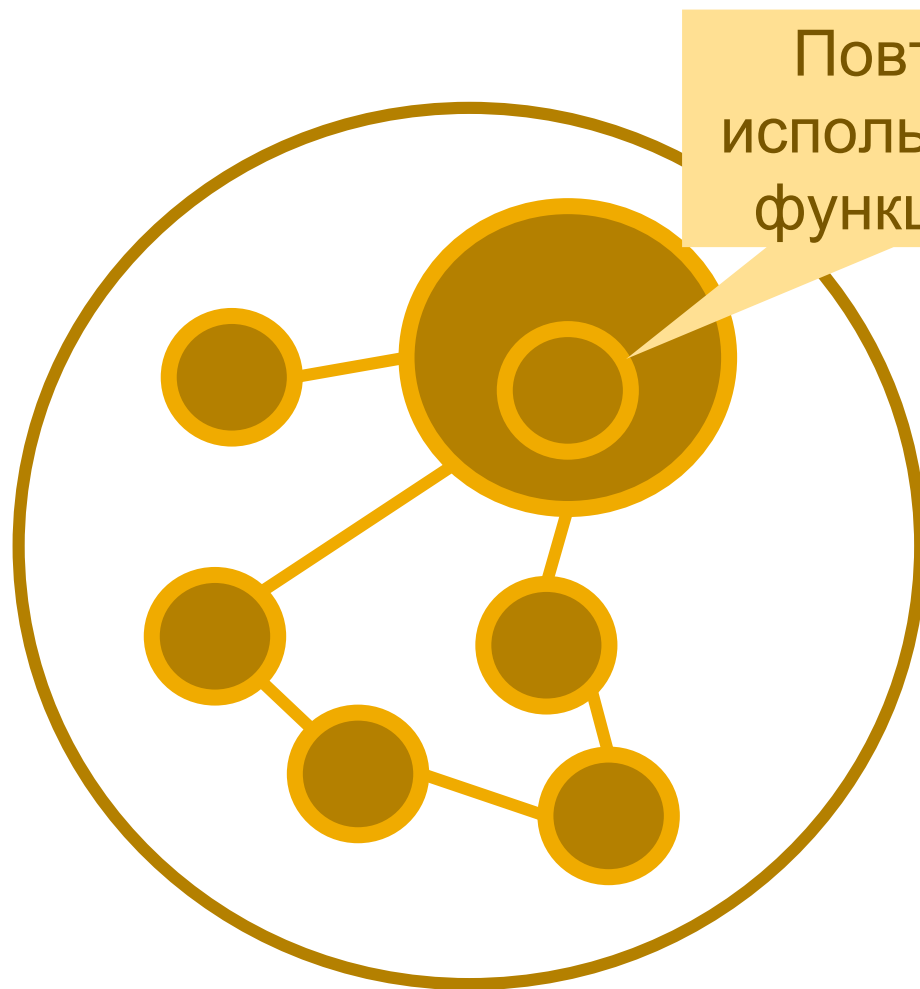
Особенности команд разработчиков



Разделение жизненных циклов



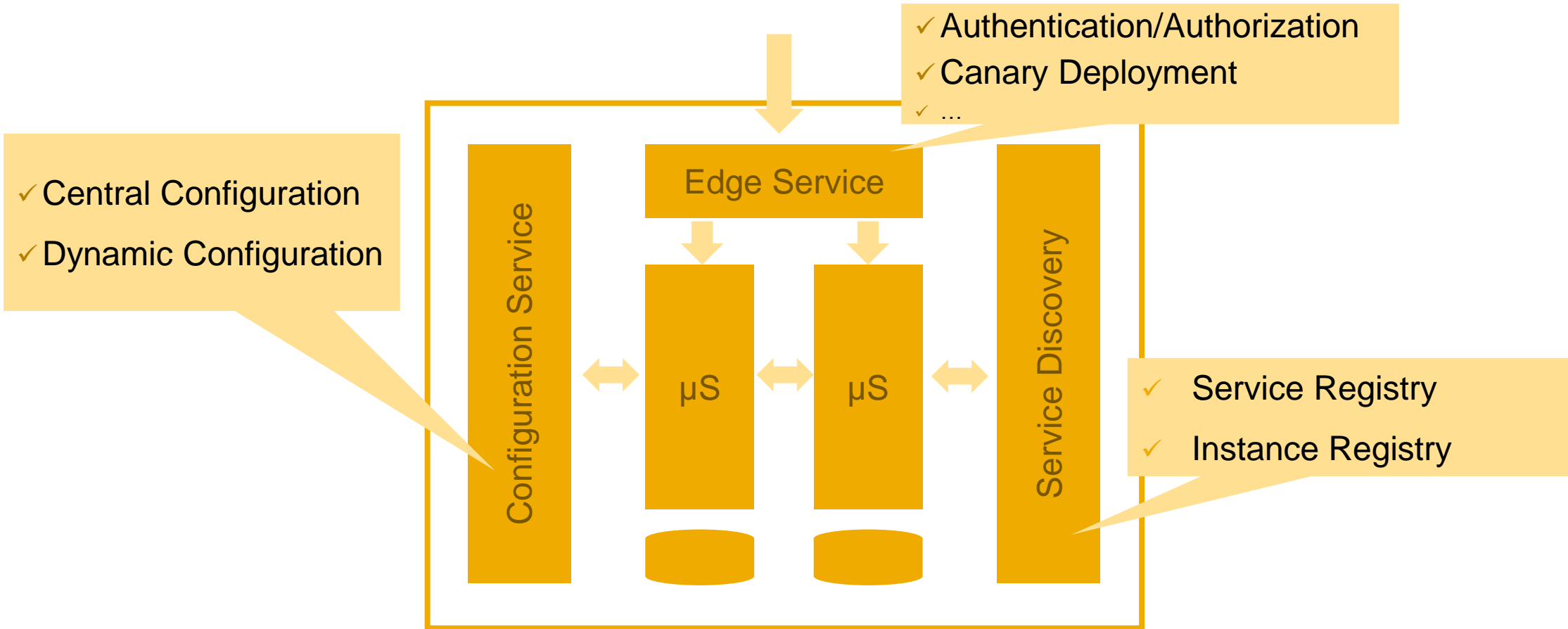
Повторное использование



Микросервисы

Время выполнения

Ландшафт приложения



Вопросы





Спасибо за внимание!

Контакт:

Oleg Koutyrine

Architect

SAP SE

HANA Cloud Platform

oleg.koutyrine@sap.com