

Микросервисы в корпоративном ИТ-ландшафте

A large container ship is docked at a port, with a tugboat in the foreground. The ship is heavily loaded with colorful shipping containers. The scene is set during sunset or sunrise, with the sky transitioning from blue to orange. The ship's deck is illuminated by warm lights, and the water reflects the ambient light. In the background, a city skyline is visible under a clear sky.

Максим Смирнов
<https://mxsmirnov.com>

Давайте знакомиться!



Максим Смирнов

<https://mxsmirnov.com>

mxsmirnov@gmail.com

2017-2018: АО "БИНБАНК Диджитал"
Главный архитектор (Начальник управления)

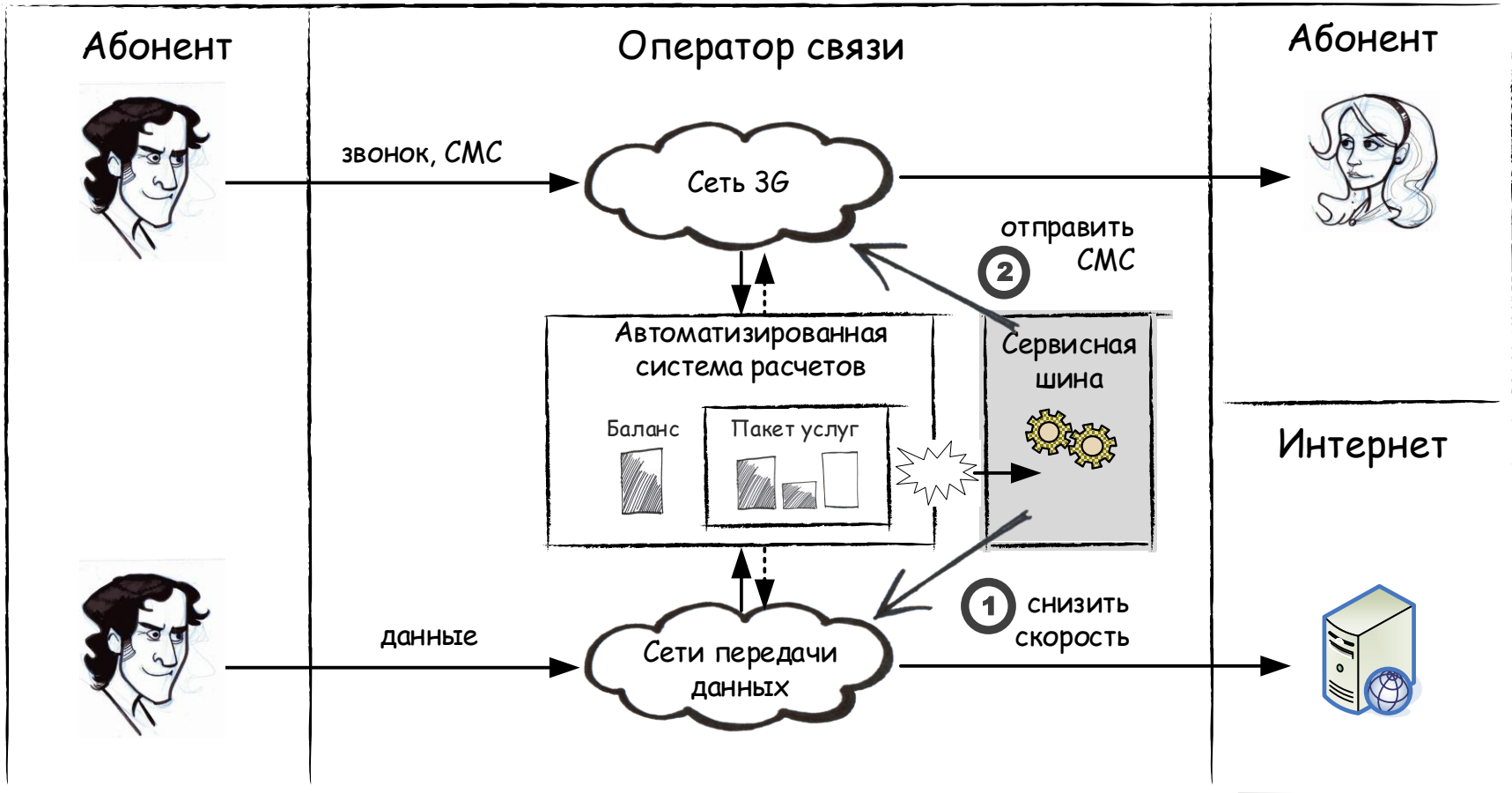
2015-2016: Главный архитектор информационных систем Центрального Банка Российской Федерации (Банк России)

2006-2014: Руководитель Департамента ИТ архитектуры ОАО "ВымпелКом" (торговая марка "Билайн")

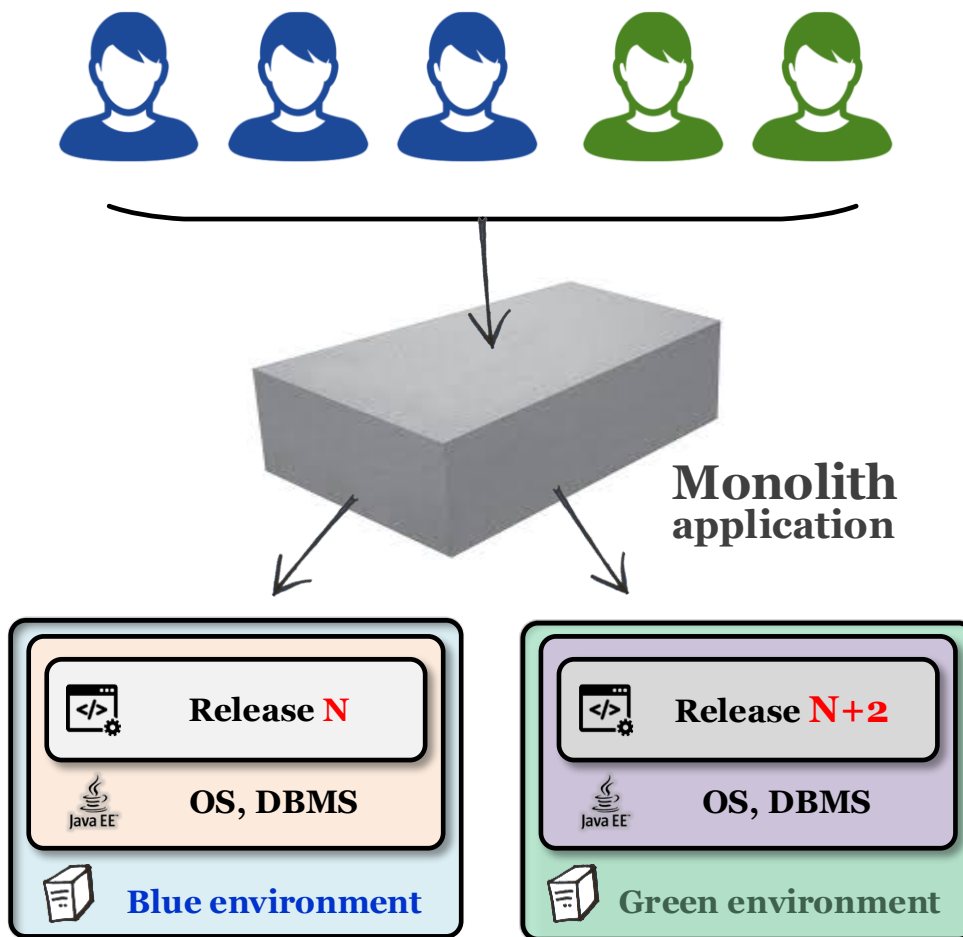
2000-2005: Менеджер по развитию дополнительных услуг связи: платежей, программы лояльности и др.

1995-2000: Эксперт по системам обмена сообщениями, приложениям "Банк-Клиент" и "Банк-Банк" ОАО АБ "Инкомбанк"

Проектирование услуг связи



“Канареечные” релизы





Новое понимание безопасности:

- ✓ **Наблюдаемость:**
способность выявлять надвигающиеся проблемы
- ✓ **Изоляция сбоев:**
сокращение количества затронутых аварией компонент
- ✓ **Устойчивость:**
способность системы к функционированию при неработоспособности отдельных модулей
- ✓ **Авто-восстановление:**
автоматизация сценариев реакции на отказы и сбои

Банки выбирают микросервисную архитектуру

14 ноября 2017 г.
Digital October

Форум про IT-трансформацию банковской платформы

Организатор форума
FutureBanking™



Банку цифрового века
требуется современное цифровое ядро

[Программа](#)

[Принять участие](#)

[Контакты](#)



Алексей Лебедев
Руководитель направления
«Розничные банки»
ЕРАМ

Микросервисная архитектура: мифы и реальность



Андрей Овсянников
Руководитель блока
«информационные
технологии»
Промсвязьбанк

Нужны ли банку микросервисы?



Евгений Цельмин
Директор по производству
ТехноСофт

Микросервисная архитектура. Модный разоряющий тренд или панацея? Стоимость владения и принесённая ценность



Андрей Сабынин
CIO
Альфа-Банк Казахстан

Микросервисы vs интеграционная шина



Андрей Дробышев
СТО
Brainysoft

Использование микросервисов в рамках цифрового периметра МФО



Елена Преподобная
Директор по маркетингу
БКС

Marketing Analytic React System



Алексей Северов
Главный архитектор
Банк «Ренессанс Кредит»

Переход от сервисной шины к новой интеграции – «API*Net»



Михаил Бобков
Системный архитектор
Неофлекс

IT-архитектура предприятия: Микросервисы и управление изменениями

Что же такое микросервисы

“ “ Microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API. These services are built around business capabilities and independently deployable by fully automated deployment machinery. There is a bare minimum of centralized management of these services, which may be written in different programming languages and use different data storage technologies.

Архитектурный стиль микросервисов — это подход, при котором единое приложение строится как набор небольших сервисов, каждый из которых работает в собственном процессе и коммуницирует с остальными используя легковесные механизмы, как правило HTTP. Эти сервисы построены вокруг бизнес-потребностей и развертываются независимо с использованием полностью автоматизированной среды. Существует абсолютный минимум централизованного управления этими сервисами. Сами по себе эти сервисы могут быть написаны на разных языках и использовать разные технологии хранения данных.

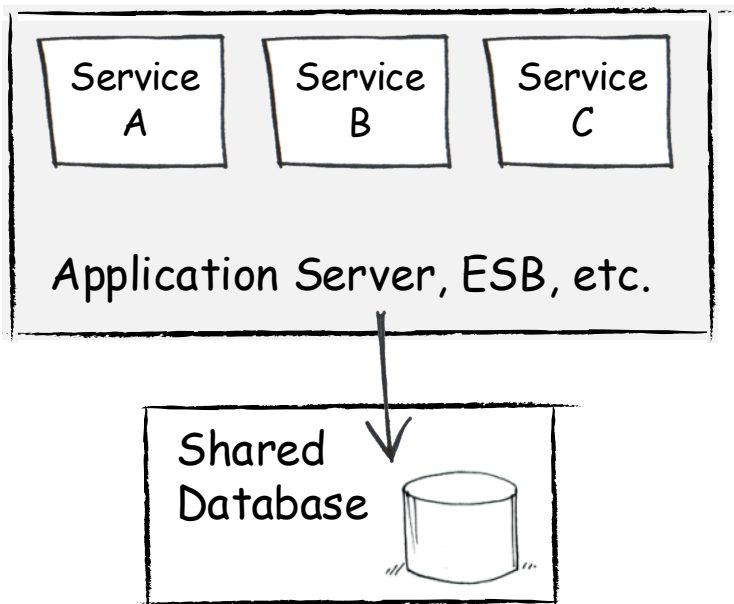
<https://martinfowler.com/articles/microservices.html>

Characteristics of a Microservice Architecture

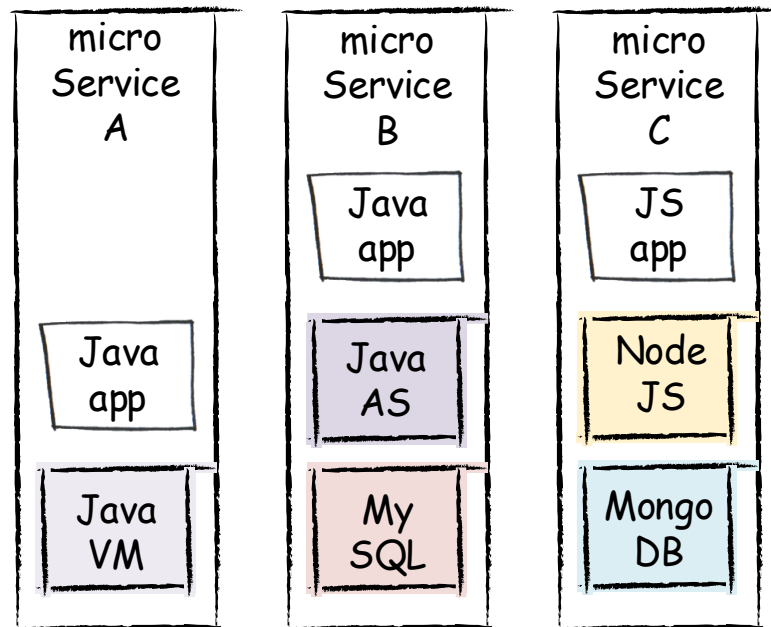
1. Componentization via Services
2. Organized around Business Capabilities
3. Products not Projects
4. Smart endpoints and dumb pipes
5. Decentralized Governance
6. Decentralized Data Management
7. Infrastructure Automation
8. Design for failure
9. Evolutionary Design

SOA и микросервисы

Как нарушить все правила "хорошей архитектуры"



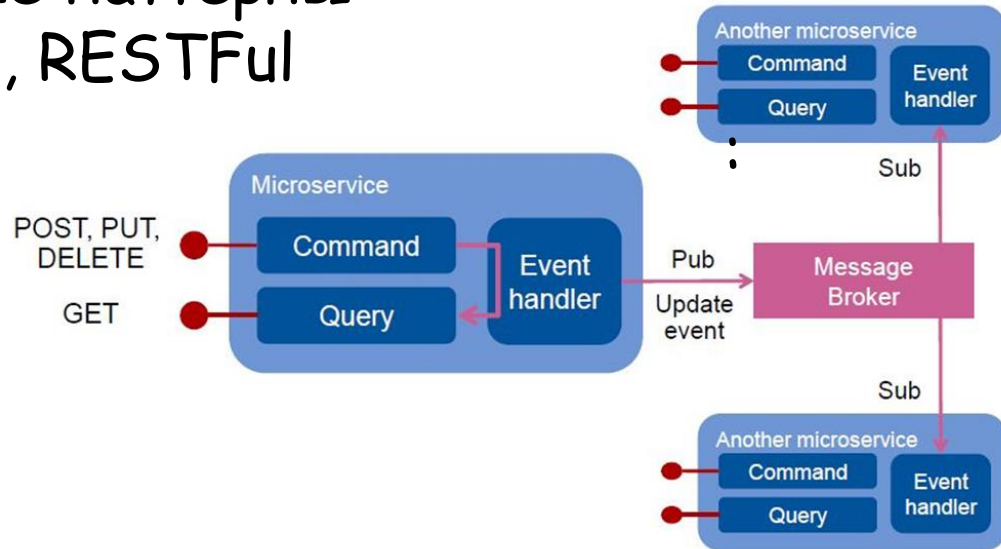
Разработаны для повторного использования



Реализуют расширения, обработку исключений, частную бизнес-логику

Что принесла тема микросервисов

- ✓ Инструменты(cncf.io): k8s, fluentd, linkerd...
- ✓ Набор ops-паттернов: green/blue deployment, circuit breaker...
- ✓ Полезные советы по архитектуре
- ✓ Интеграционные паттерны: Message Broker, RESTful
- ✓ Много другое: Evolutionary Architecture, Monolith First...



Корпоративный ИТ-ландшафт



- ✓ Несколько команд, которые не очень дружат друг с другом
- ✓ “Зоопарк” технологий
- ✓ Постоянная потребность в расширении функционала
- ✓ Унаследованные приложения
- ✓ Spaghetti-like интеграция

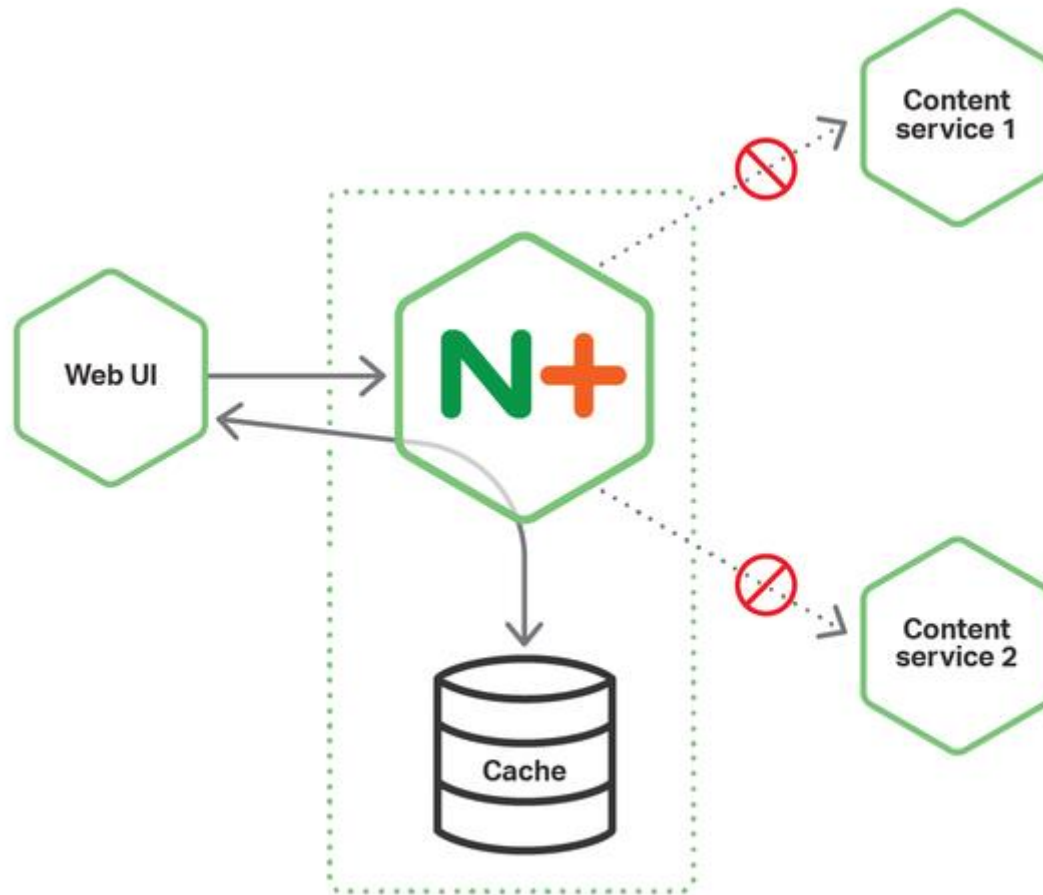
Как меняется интеграция



Реальные проблемы интеграции:

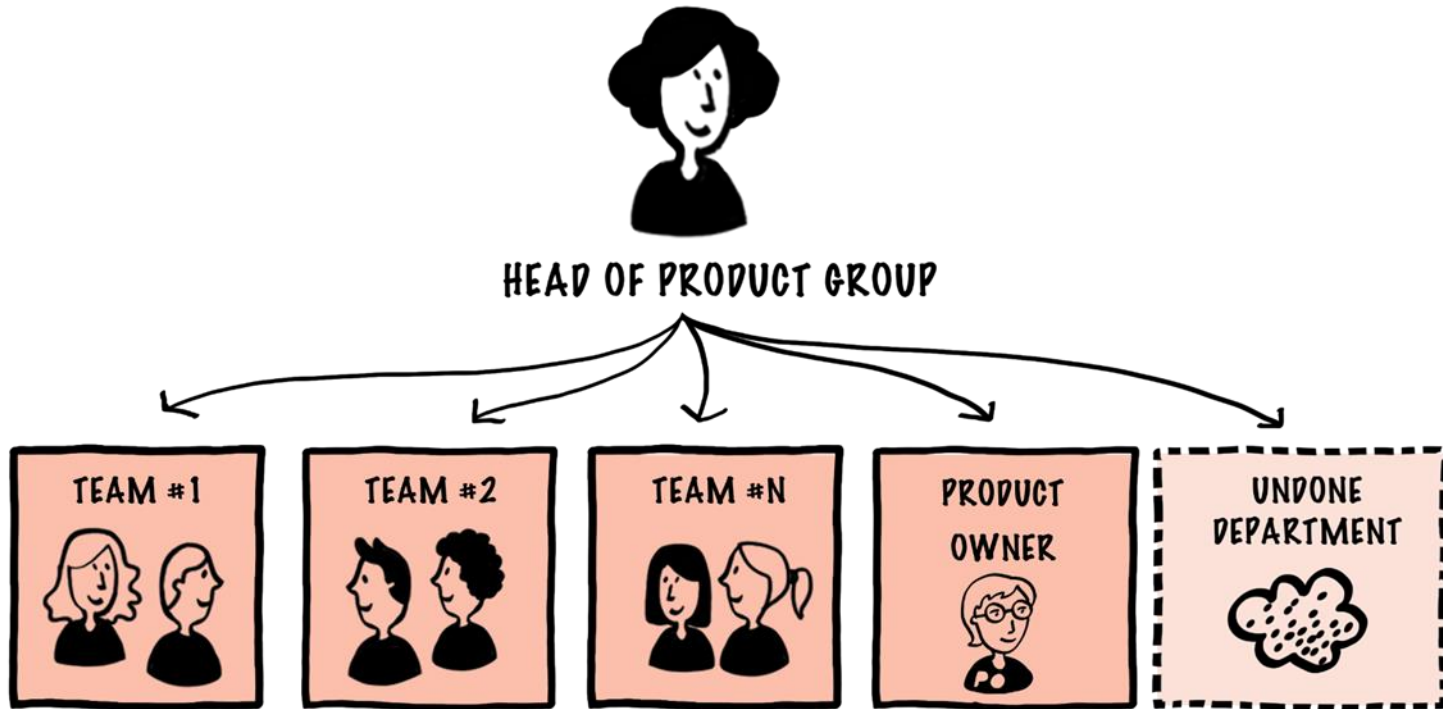
- ✓ устаревшие программные интерфейсы
- ✓ сложные цепочки взаимодействий
- ✓ необходимость проверки условий
- ✓ обработка ошибок и исключений

1. Circuit Breaker Pattern




<https://www.nginx.com/blog/microservices-reference-architecture-nginx-circuit-breaker-pattern/>

2. Ограничивайте размер команд

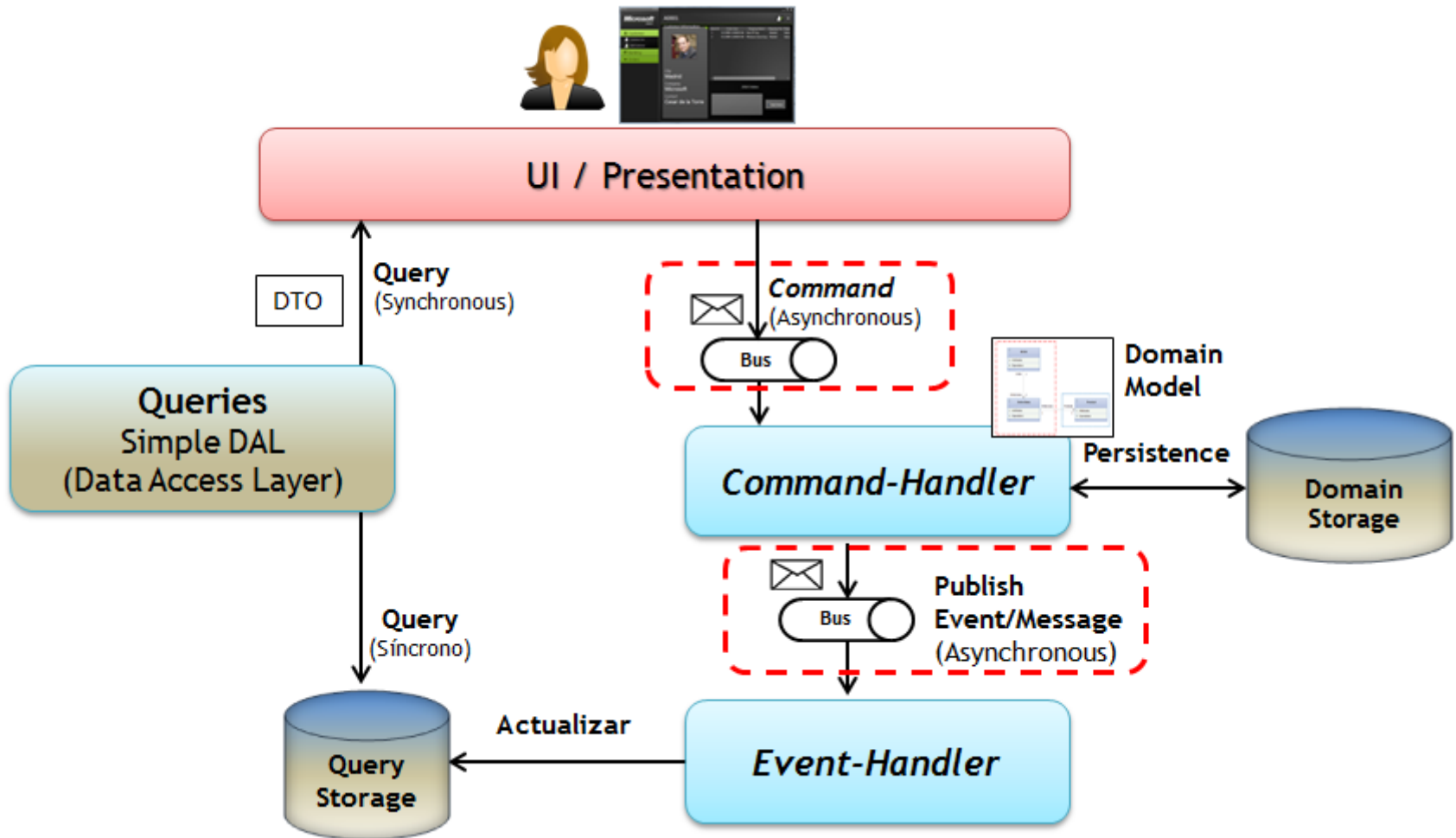


<http://less.works> BY-ND

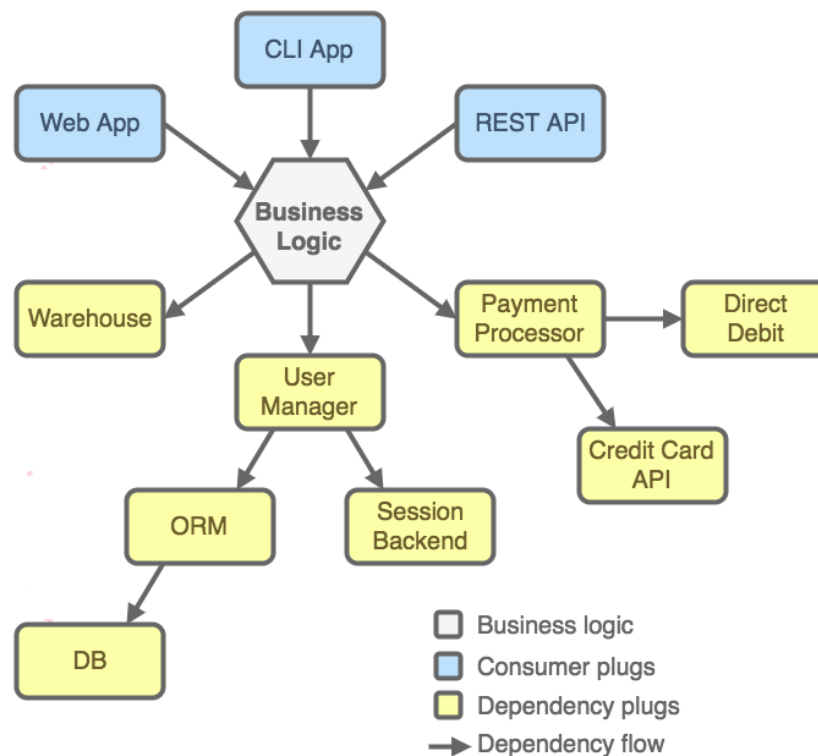
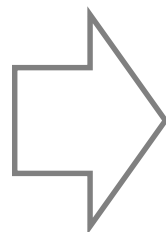
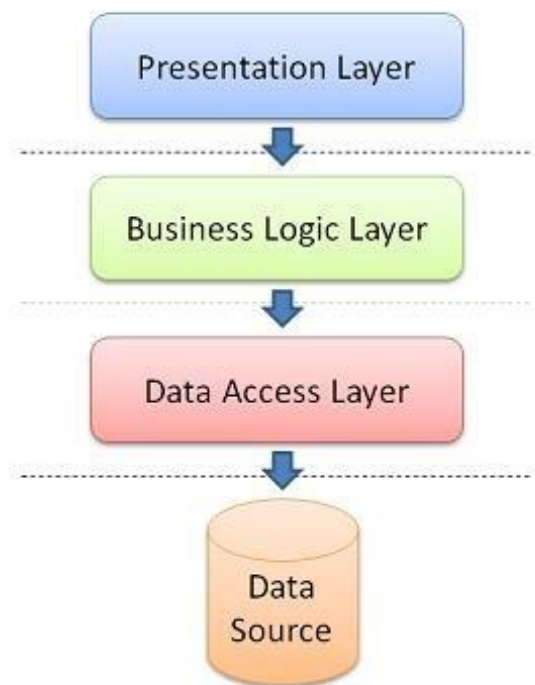
A large crowd of diverse people in business attire is walking away from the viewer, filling most of the frame. In the foreground, a man in a dark shirt and trousers stands with his back to the camera, looking towards the crowd. A speech bubble is positioned to the left of the crowd, containing the Russian text "Все в backlog?".

Все в backlog?

3. "Правильный" CQRS Pattern



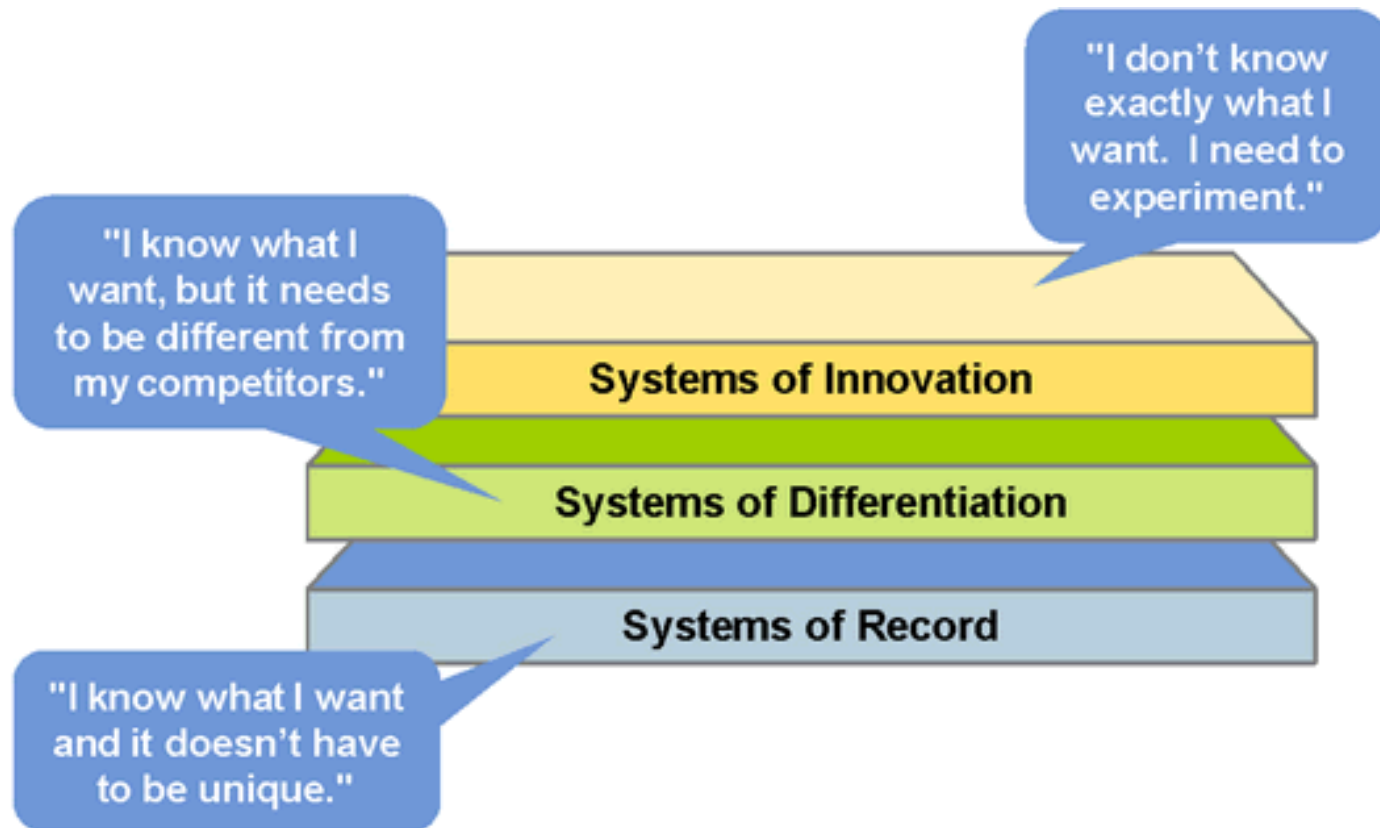
4. Evolutionary architecture



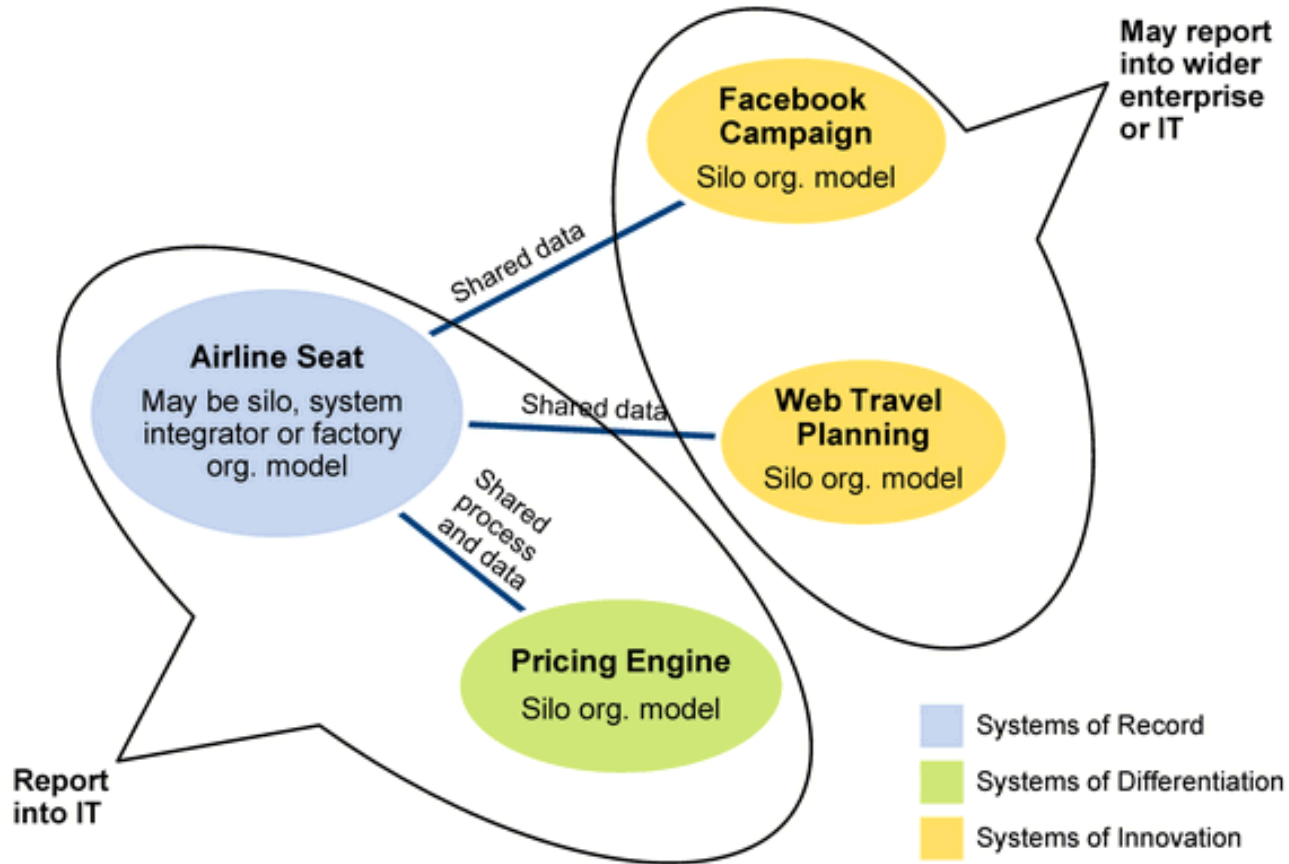


**Паттерн Inversion of Control
проникает в Run-time**

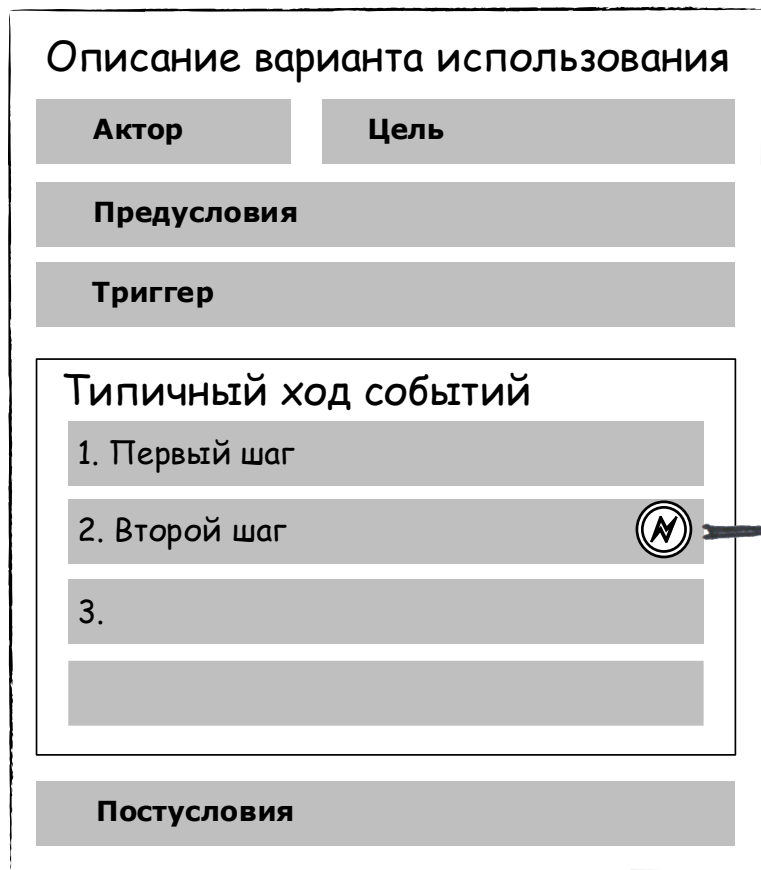
Gartner: Pace-Layered Application Strategy



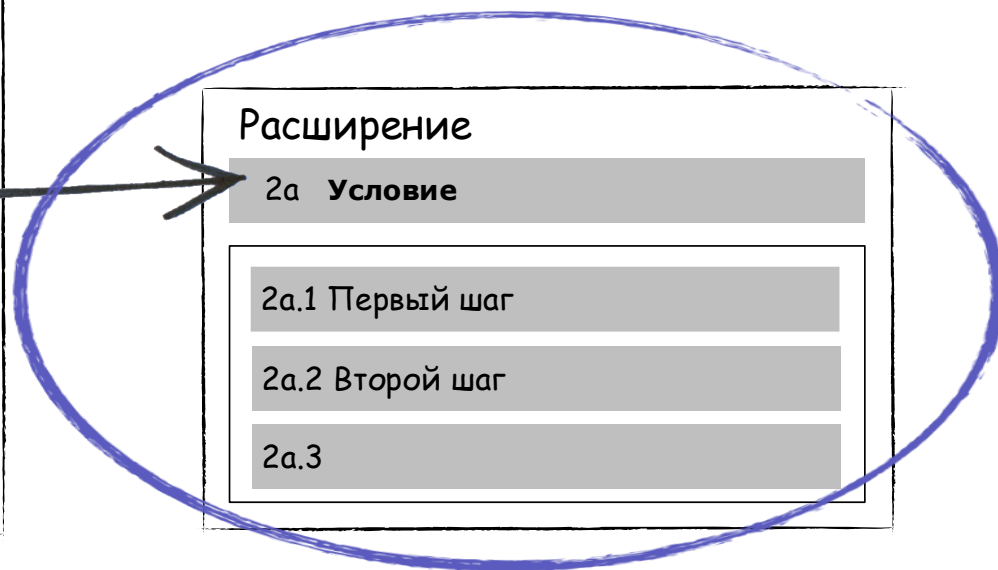
Evolutionary architecture and emergent design



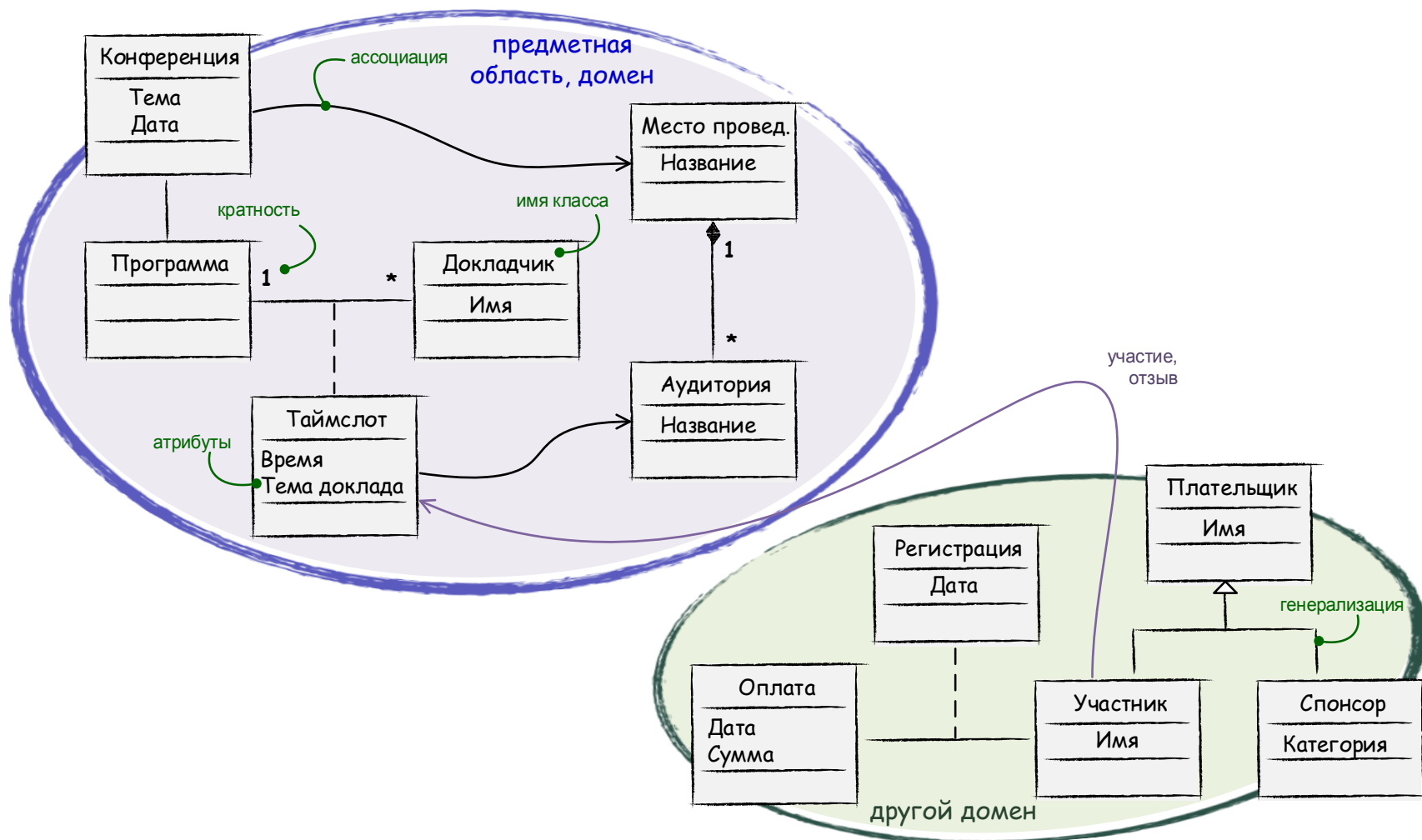
4. Evolutionary architecture



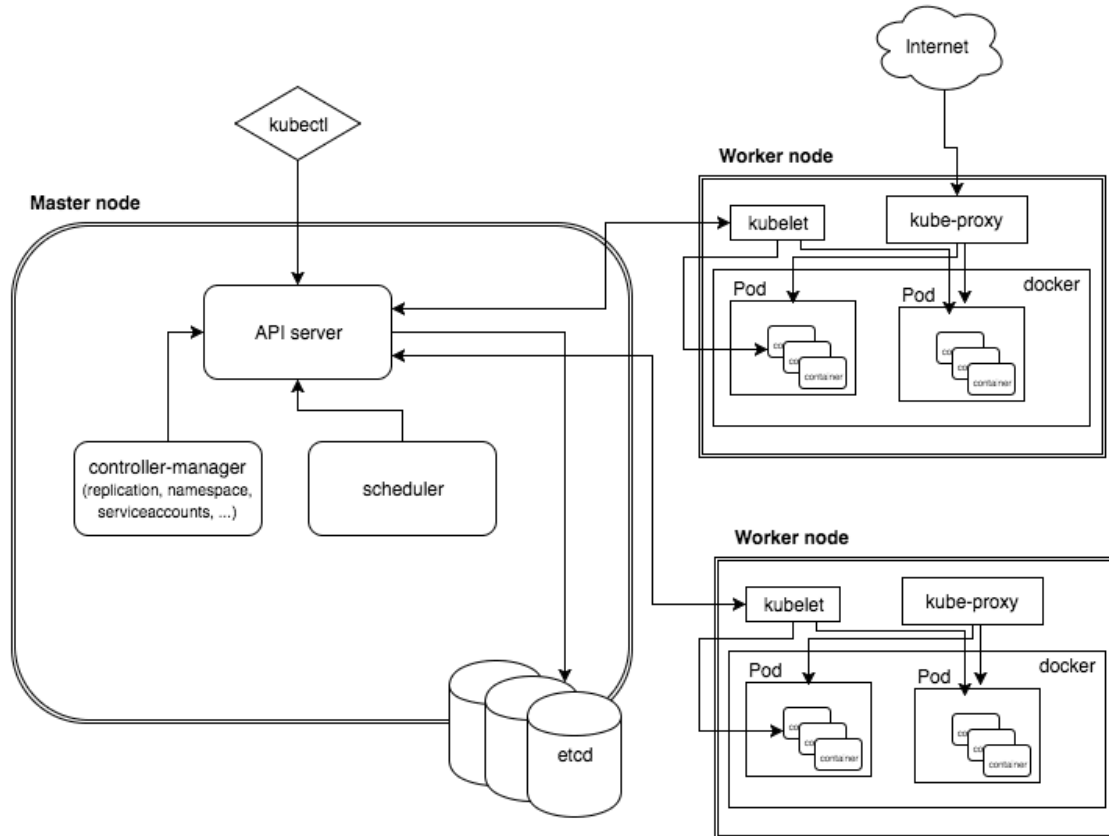
Реализуется как
микросервис



Domain-Driven Design



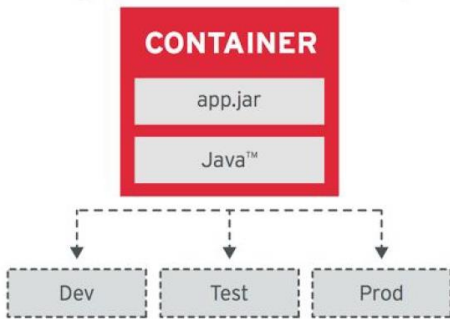
5. The Kubernetes Effect



<https://www.infoq.com/articles/kubernetes-effect>

Container Design Principles

Image Immutability Principle



High Observability Principle



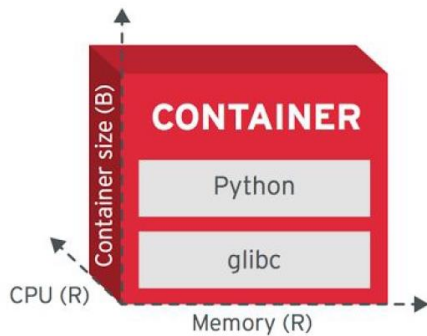
Process Disposability Principle



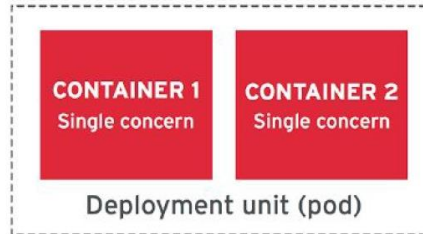
Lifecycle Conformance Principle



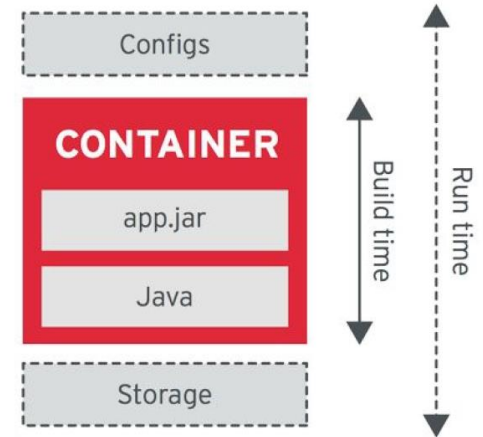
Runtime Confinement Principle



Single Concern Principle

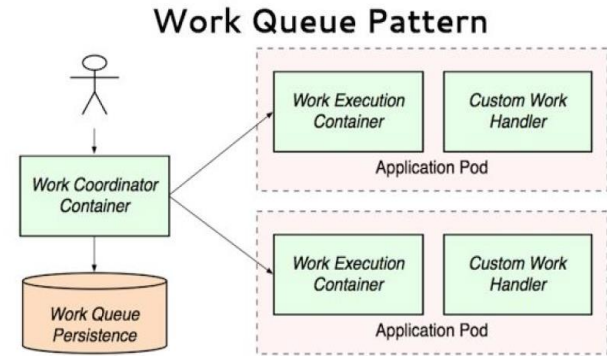
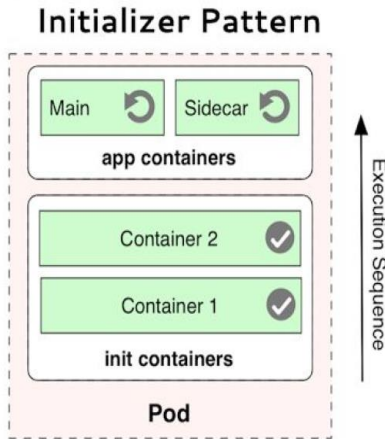
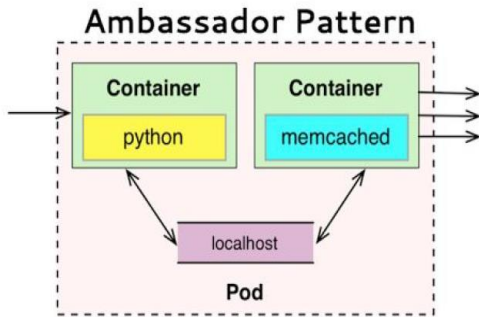
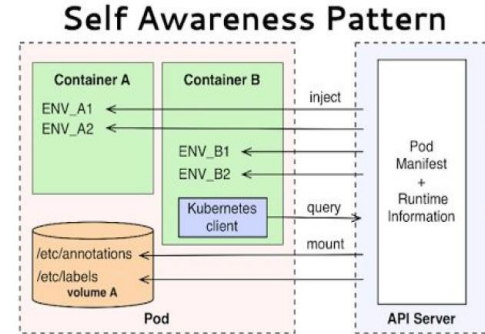
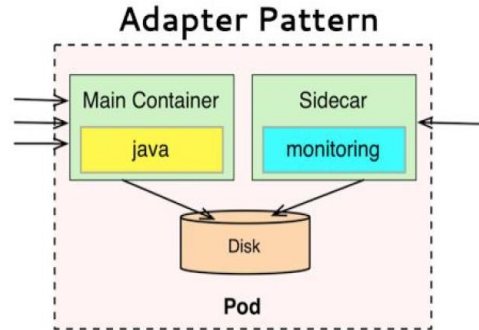
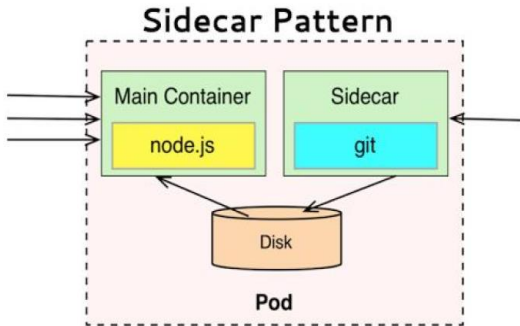


Self-Containment Principle



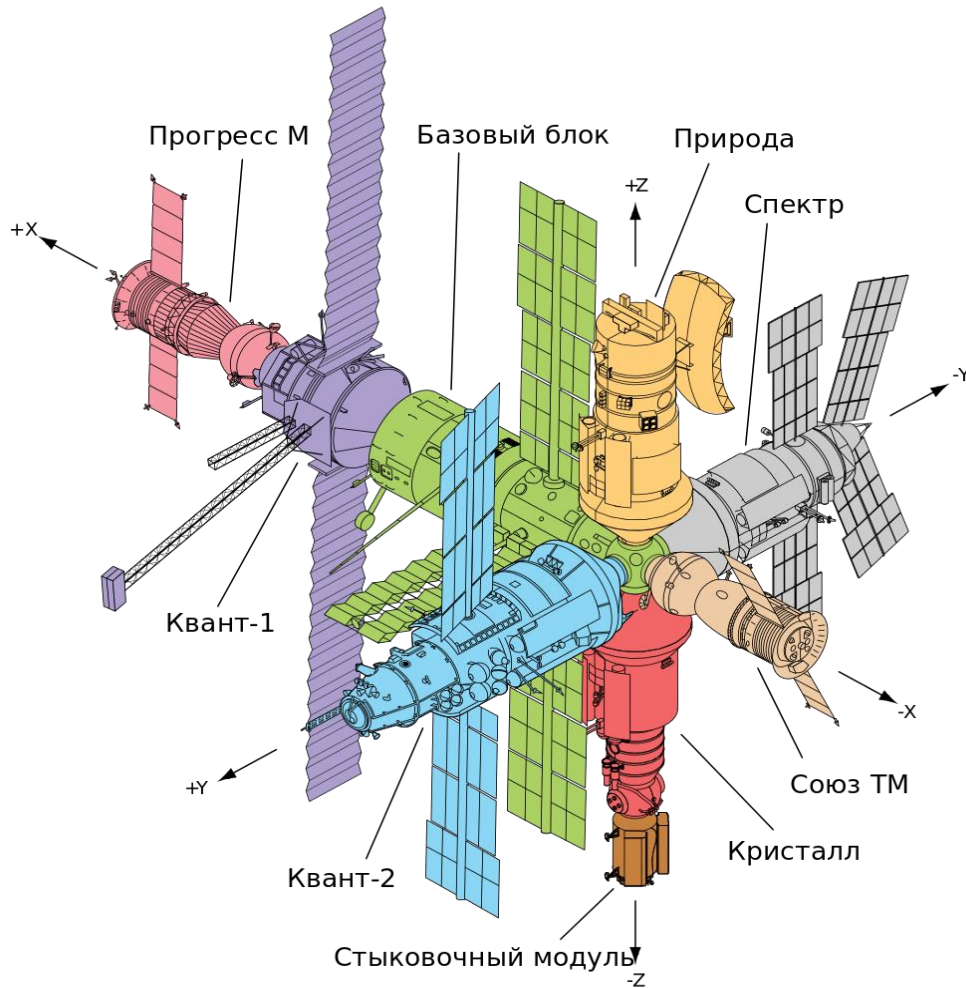
<https://www.infoq.com/articles/kubernetes-effect>

Container Design Patterns



<https://www.infoq.com/articles/kubernetes-effect>

Модули станции "МИР"



- 1986 - Базовый модуль
- 1987 - "Квант"
- 1989 - "Квант-2"
- 1990 - "Кристалл"
- 1995 - "Спектр"
- 1995 - Стыковочный узел для Space Shuttle
- 1996 - "Природа"

Вопросы и комментарии

