

MTC True Tech

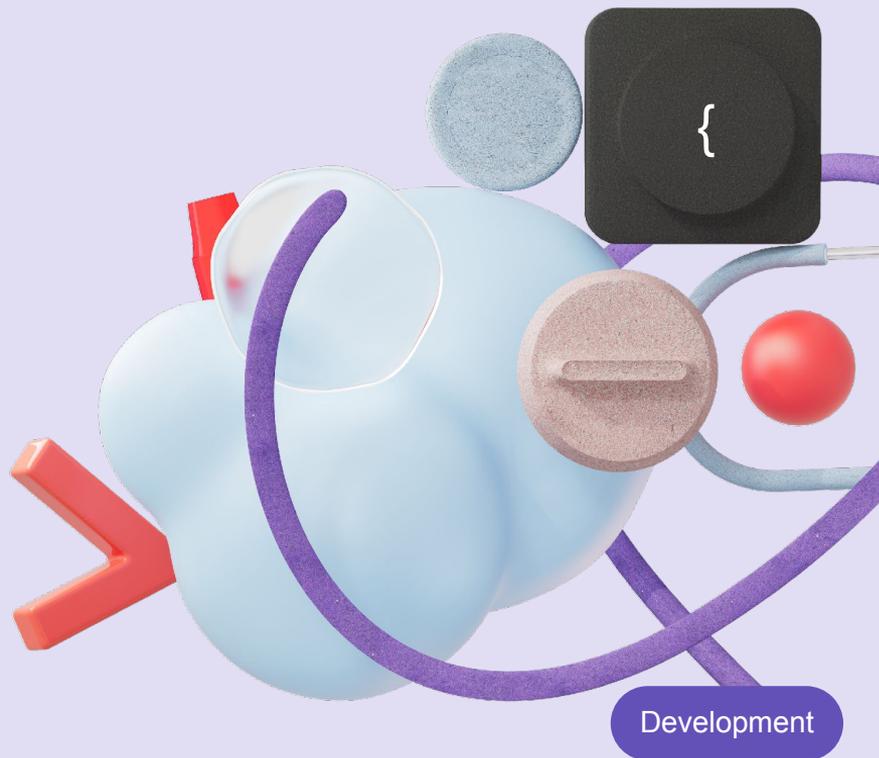
Day

+

БОЛЬШИЕ ПРОЕКТЫ: СЕРВИСЫ ИЛИ МИКРОСЕРВИСЫ?

Алексей Рыбак

- devhands.io, основатель (образование и инфраструктурная практика для программистов)
- ex-CTO/VP: Badoo, Vezet, Yum! Brands, Constructor Group

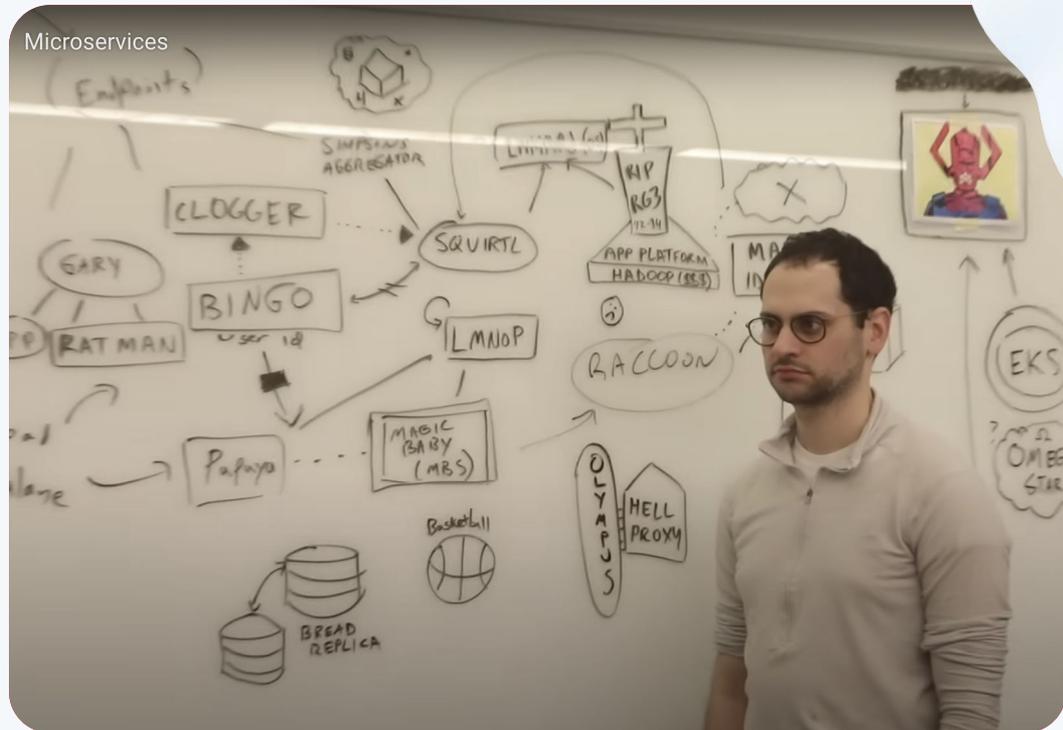


DISCLAIMER (БЕЗ ТРОЛЛИНГА, ПОЧТИ): С МИКРОСЕРВИСАМИ ВСЁ ОТЛИЧНО

Любой архитектурный стиль имеет и преимущества, и недостатки.

Посмотрим на некоторые (возможно, неожиданные) риски использования микросервисов

"It's because of the way our backend works" @KRAZAM
<https://www.youtube.com/watch?v=y8OnoxKotPQ>



СЕРВИСЫ ИЛИ МИКРОСЕРВИСЫ?

Фундаментальные проблемы и план доклада

“Изоляция”: границы и скрытая стоимость

Нет четкого критерия, в какой момент “сервис” становится “микросервисом”. Изоляция небесплатна, приводит к усложнению коммуникаций, повышению стоимости и даже централизации

Основные риски МСА

Основные риски использования микро-сервисной архитектуры (МСА) хорошо известны. Постараемся вспомнить про все в контексте следствий фундаментальных проблем

Три риска разберем подробнее

Поговорим про три риска

- 1. Упрощение:** упрощенный взгляда на архитектуру и ухудшения качества проектирования
- 2. Консистентность:** следствие “распределенности”
- 3. Стоимость:** следствие изоляции

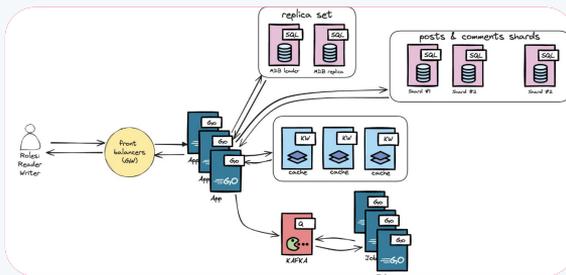


ПРОБЛЕМА №1: НЕТ КРИТЕРИЯ “МИКРО” КАКАЯ ВООБЩЕ БЫВАЕТ ИЗОЛЯЦИЯ



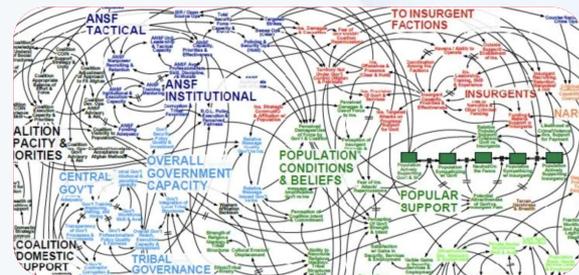
Модули

Изолирован код / API классов и библиотек



Сервисы

Изолированы приложения / сервисные API



Микросервисы

Как сервисы, только микро



ПРОБЛЕМА №2: ИЗОЛЯЦИЯ НЕ БЕСПЛАТНА

Получаем: распределенность и ... централизацию



Распределенность

В “классическом” подходе система становится распределенной по мере роста, в MSA система распределенная изначально. Риски

- Скорость (latency)
- Сложность отладки
- Неконсистентность
- Усложнение ops-механик (управление репозиториями и релизами)

Коммуникация

- Увеличение нагрузки на сеть и промежуточные узлы
- Увеличение нагрузки на процессор из-за преобразований данных для межсервисных API
- Совместимость и версионирование
- Безопасность

Централизация

Единый протокол обмена и брокер - здравствуй, Центральная Шина. Распределенный трейсинг, мониторинг, анализ метрик нужно собирать в “одном месте”. Риски:

- SPOF
- объемы данных
- стоимость решений



МИКРОСЕРВИСЫ РИСКИ АРХИТЕКТУРНОГО СТИЛЯ



Усложнение системы: распределенность, неконсистентность, сложность отладки и большой объём данных (распределенный трейсинг)



МИКРОСЕРВИСЫ РИСКИ АРХИТЕКТУРНОГО СТИЛЯ



Усложнение системы: распределенность, неконсистентность, сложность отладки и большой объём данных (распределенный трейсинг)



Увеличение стоимости ops-процедур: совместимость/версионирование, управление репозиториями, пайплайнами, security



МИКРОСЕРВИСЫ РИСКИ АРХИТЕКТУРНОГО СТИЛЯ



{ Усложнение системы: распределенность, неконсистентность, сложность отладки и большой объём данных (распределенный трейсинг)

{ Увеличение стоимости ops-процедур: совместимость/версионирование, управление репозиториями, пайплайнами, security

{ Увеличение стоимости облачных ресурсов (особенно managed DB при полной изоляции БД)



МИКРОСЕРВИСЫ РИСКИ АРХИТЕКТУРНОГО СТИЛЯ



{ Усложнение системы: распределенность, неконсистентность, сложность отладки и большой объём данных (распределенный трейсинг)

{ Увеличение стоимости ops-процедур: совместимость/версионирование, управление репозиториями, пайплайнами, security

{ Увеличение стоимости облачных ресурсов (особенно managed DB при полной изоляции БД)

{ Упрощение архитектуры и снижение качества проектирования



НЕОЖИДААННЫЙ РИСК#1: УПРОЩЕНИЕ УХУДШЕНИЕ КАЧЕСТВА АРХИТЕКТУРЫ

Продуктовые компании:
специальный формат
архитектурных сессий:
“Системный Дизайн”.

Формат представления
архитектуры, чтобы снизить
вероятность “булшита”.

MCA: внезапное регулярное
ухудшение качества дизайна в
парадигме MCA

Сорвите уборы: на маминых
полках – “рисователи кружочков”

Допустим, нужно спроектировать
платформу для много-
пользовательских блогов

Что у нас есть? собственно
платформа, пользователи, там,
посты, комментарии...

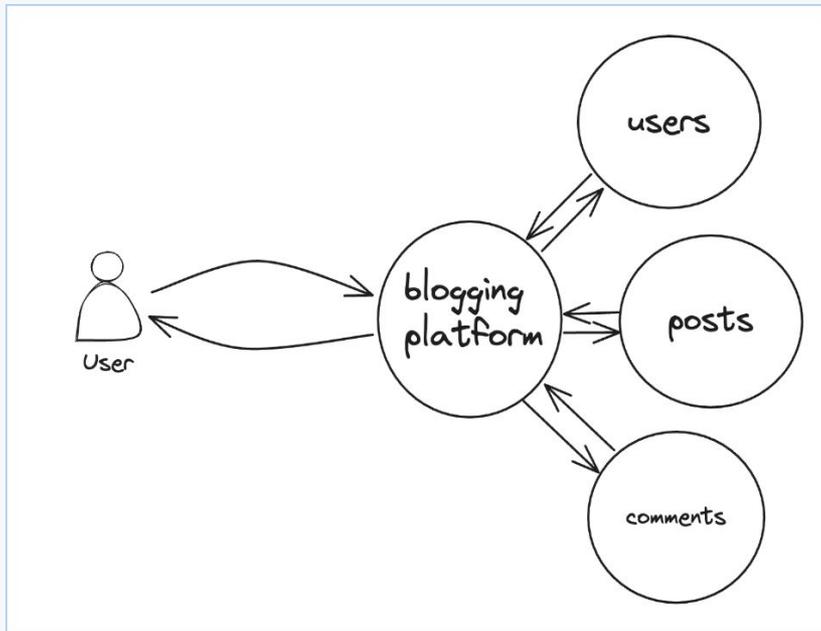


НЕОЖИДААННЫЙ РИСК#1: УПРОЩЕНИЕ УХУДШЕНИЕ КАЧЕСТВА АРХИТЕКТУРЫ

Продуктовые компании:
специальный формат
архитектурных сессий:
“Системный Дизайн”.

Формат представления
архитектуры, чтобы снизить
вероятность “булшита”.

МСА: внезапное регулярное
ухудшение качества дизайна в
парадигме МСА



ВЫ МНЕ, ВОЗМОЖНО, СКАЖЕТЕ: ЧТО ЗА АНЕКДОТ?

Внимание!

Черный ящик!

Картинка.

С официального.

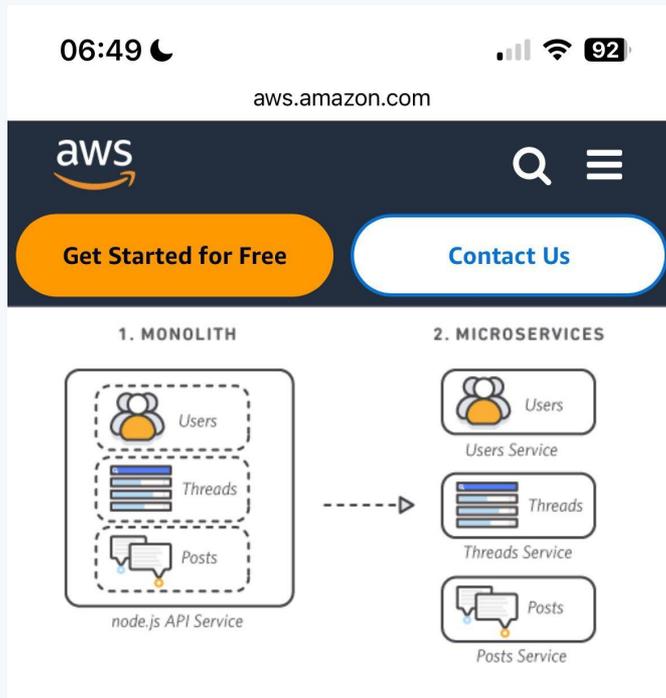
Сайта

AWS.



ВЫ МНЕ, ВОЗМОЖНО, СКАЖЕТЕ: ЧТО ЗА АНЕКДОТ?

Внимание!
Черный ящик!
Картинка.
С официального.
Сайта
AWS.



КАК ЭТО “РАЗВИДЕТЬ”?

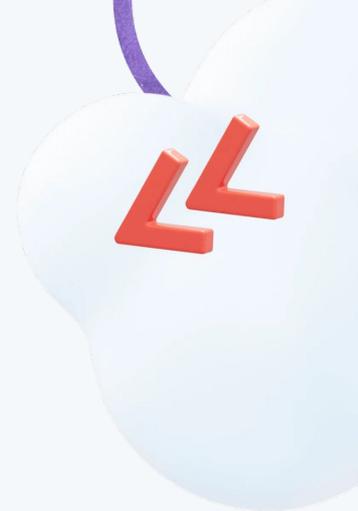
КАК ПЕРЕСТАТЬ БЫТЬ “РИСОВАТЕЛЯМИ КРУЖОЧКОВ?”



Представление: C4Model, а лучше - “роли, сценарии, сервисы”

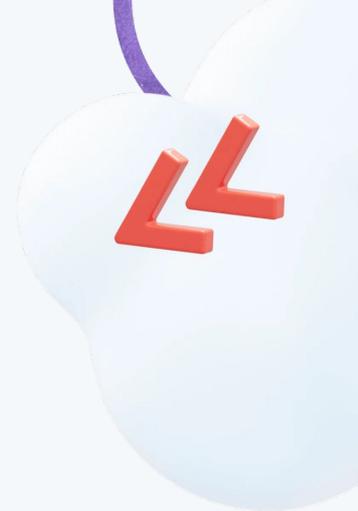


Сервисы: приложения, веб-серверы, СУБД, брокеры, gateways, identity services и тд - не “кружочки”, а по-настоящему deployable components



КАК ЭТО “РАЗВИДЕТЬ”?

КАК ПЕРЕСТАТЬ БЫТЬ “РИСОВАТЕЛЯМИ КРУЖОЧКОВ?”



Представление: C4Model, а лучше - “роли, сценарии, сервисы”



Сервисы: приложения, веб-серверы, СУБД, брокеры, gateways, identity services и тд - не “кружочки”, а по-настоящему deployable components



Каждый сценарий “раскладывается” в запросы и может верифицироваться на уровне API



Наглядное представление движения данных, использования ресурсов, возможно обсуждение планирования мощностей, масштабирования, надежности



КАК ЭТО “РАЗВИДЕТЬ”?

КАК ПЕРЕСТАТЬ БЫТЬ “РИСОВАТЕЛЯМИ КРУЖОЧКОВ?”



Представление: C4Model, а лучше - “роли, сценарии, сервисы”



Сервисы: приложения, веб-серверы, СУБД, брокеры, gateways, identity services и тд - не “кружочки”, а по-настоящему deployable components



Каждый сценарий “раскладывается” в запросы и может верифицироваться на уровне API



Наглядное представление движения данных, использования ресурсов, возможно обсуждение планирования мощностей, масштабирования, надежности



Нужно понимать про программирование, особенности компонент, API, масштабирование



Идеально поручать этот этап проектирования только людям, имеющим соответствующий опыт и навыки

РИСК#2. КОНСИСТЕНТНОСТЬ БИЧ ВСЕХ РАСПРЕДЕЛЕННЫХ СИСТЕМ

В погоне за изолированностью признали антипаттерном shared database и получили бонусом неконсистентность



Методы решения есть, но они не элементарные: Transactional Outbox, Оркестраторы “сag”



РИСК#2. КОНСИСТЕНТНОСТЬ БИЧ ВСЕХ РАСПРЕДЕЛЕННЫХ СИСТЕМ

В погоне за изолированностью признали антипаттерном shared database и получили бонусом неконсистентность



Методы решения есть, но они не элементарные: Transactional Outbox, Оркестраторы “сag”



Может, дешевле просто пошерить СУБД? Это можно сделать условно-изолированно (через схемы, например)



РИСК#2. КОНСИСТЕНТНОСТЬ БИЧ ВСЕХ РАСПРЕДЕЛЕННЫХ СИСТЕМ

В погоне за изолированностью признали антипаттерном shared database и получили бонусом неконсистентность



Методы решения есть, но они не элементарные: Transactional Outbox, Оркестраторы “сag”



Может, дешевле просто пошерить СУБД? Это можно сделать условно-изолированно (через схемы, например)



Может быть вы боитесь СУБД? DBA не вымерли (пока)



РИСК#3. КОСТ СТОИМОСТЬ ИНФРЫ К ФОТУ, НАПРИМЕР



Кост разный. Расцените это упражнение как пример. Все совпадения - случайны.
Возьмем только стоимость managed DB



РИСК#3. КОСТ СТОИМОСТЬ ИНФРЫ К ФОТУ, НАПРИМЕР



Кост разный. Расцените это упражнение как пример. Все совпадения - случайны.
Возьмем только стоимость managed DB



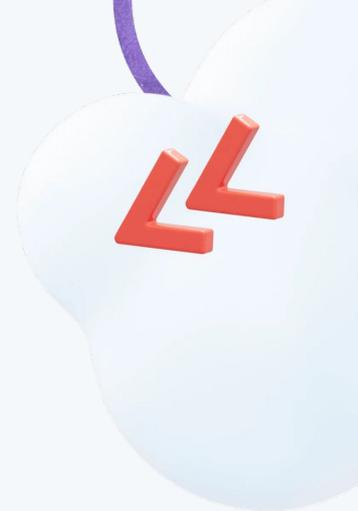
$(100 \text{ инженеров}) \times (400000 \text{ в месяц до ЕСН}) \times 12 = 480 \text{ млн руб в год}$



$(300 \text{ managed СУБД}) \times (10000 \text{ в месяц}) \times 12 = 36 \text{ млн руб в год. } 7.5\% \text{ ФОТ}$



РИСК#3. КОСТ СТОИМОСТЬ ИНФРЫ К ФОТУ, НАПРИМЕР



Кост разный. Расцените это упражнение как пример. Все совпадения - случайны.
Возьмем только стоимость managed DB



$(100 \text{ инженеров}) \times (400000 \text{ в месяц до ЕСН}) \times 12 = 480 \text{ млн руб в год}$



$(300 \text{ managed СУБД}) \times (10000 \text{ в месяц}) \times 12 = 36 \text{ млн руб в год. } 7.5\% \text{ ФОТ}$



Кост инфры / ФОТ: 0-2%% “аскеты”, 2-7%% на своём железе / гибрид, 7-15%% и выше - “чужой” клауд



Это очень примерные цифры. Посчитайте этот процент для себя. Повторите раз в год. <https://a16z.com/the-cost-of-cloud-a-trillion-dollar-paradox/>





MTC True Tech

Day



СПАСИБО ЗА ВНИМАНИЕ!

alexey.rybak@gmail.com

t.me/rybakalexey

