

 Hello, Conference!_

**ПРИМЕНЕНИЕ
МИКРОСЕРВИСНОЙ
АРХИТЕКТУРЫ
В ПОТОКОВОЙ
ОБРАБОТКЕ
BIG DATA**

Ефремов Андрей,
Руководитель центра
компетенции java
разработки ООО «МТС
ИТ» (Центр Big Data)

27 августа 2020

МТС

СОДЕРЖАНИЕ

1

ОБ
АРХИТЕКТУРНЫХ
ПАТТЕРНАХ

2

ОБ АРХИТЕКТУРНЫХ
МОДЕЛЯХ

3

О НАШЕМ ПРИМЕРЕ
РЕАЛИЗАЦИИ ПЛАТФОРМЫ

4

О ВЫВОДАХ

5

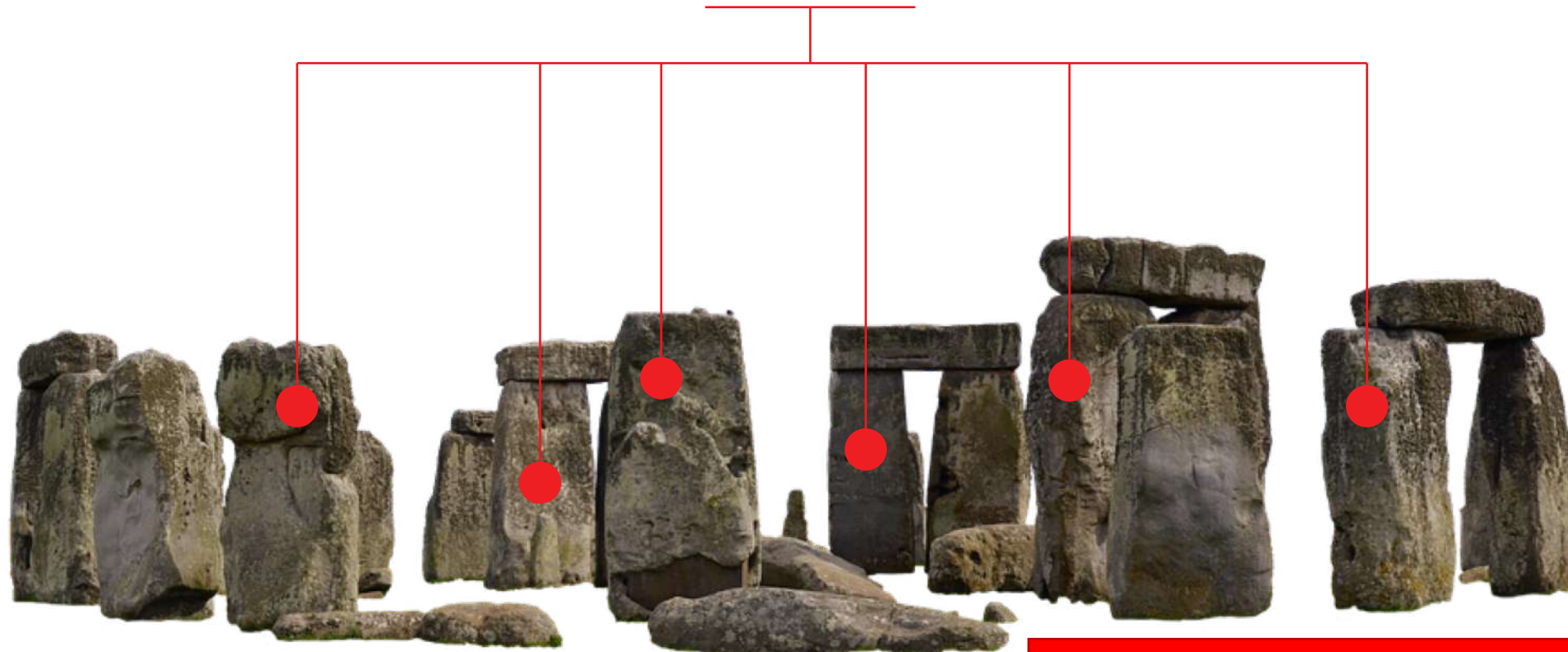
ВОПРОСЫ И ОТВЕТЫ

ОБ АРХИТЕКТУРНЫХ ПАТТЕРНАХ

1

ВСПОМНИМ, ЧТО ЖЕ ТАКОЕ МИКРОСЕРВИСЫ

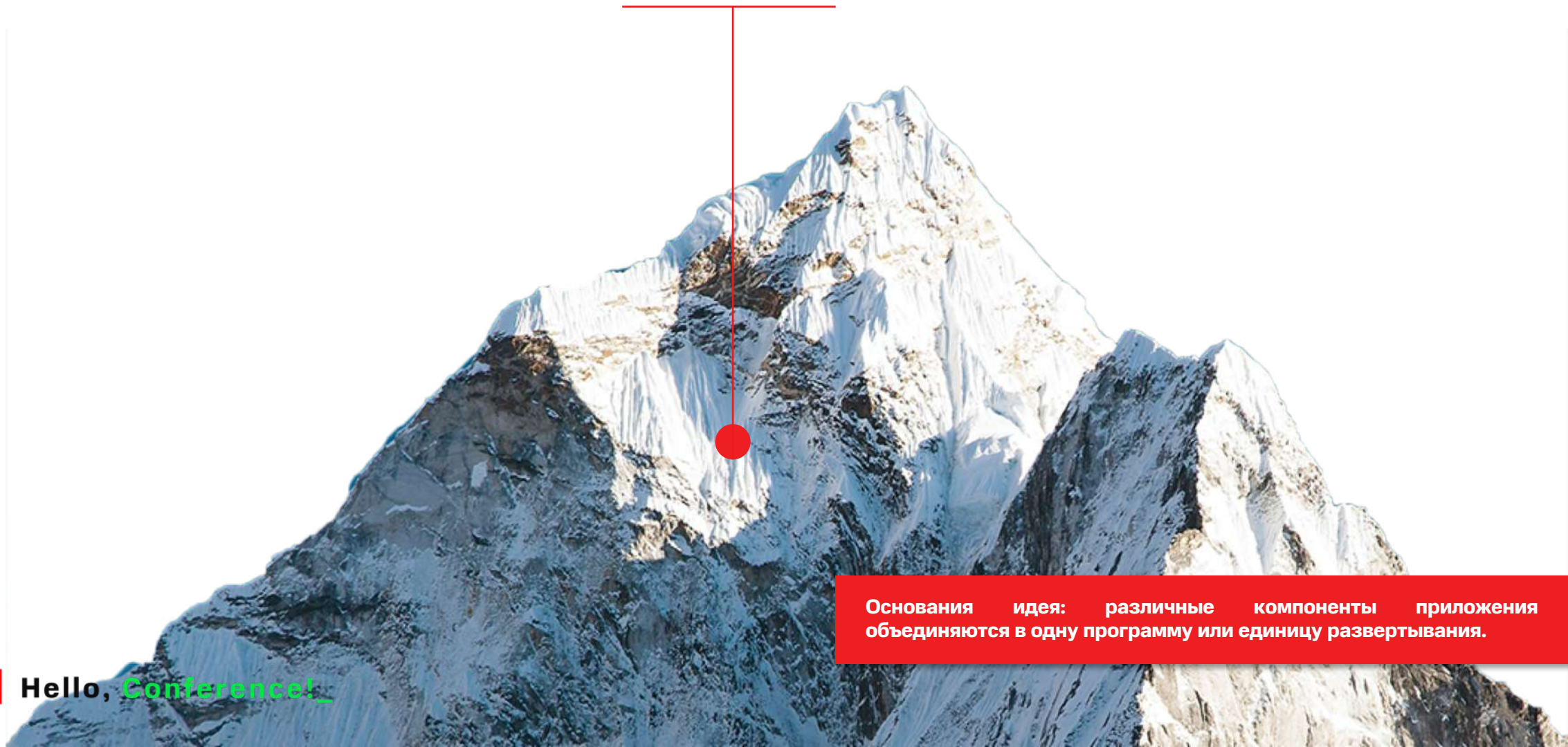
ПАТТЕРН К ПРОЕКТИРОВАНИЮ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, С ПОМОЩЬЮ
НАБОРА НЕБОЛЬШИХ, НЕЗАВИСИМЫХ И ЛЕГКО ИЗМЕНЯЕМЫХ СЕРВИСОВ



Основная идея: за счёт повышения гранулярности модулей
уменьшить степень зацепления и увеличить связность.
Плюсы: позволяет проще и быстрее добавлять и изменять
функции, тем самым сокращая ТТМ.

Вспомним, что же такое монолит

ПАТТЕРН К ПРОЕКТИРОВАНИЮ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ДОСТАВЛЯЕМОЕ ЧЕРЕЗ ЕДИНОЕ РАЗВЕРТЫВАНИЕ



Основания идея: различные компоненты приложения объединяются в одну программу или единицу развертывания.



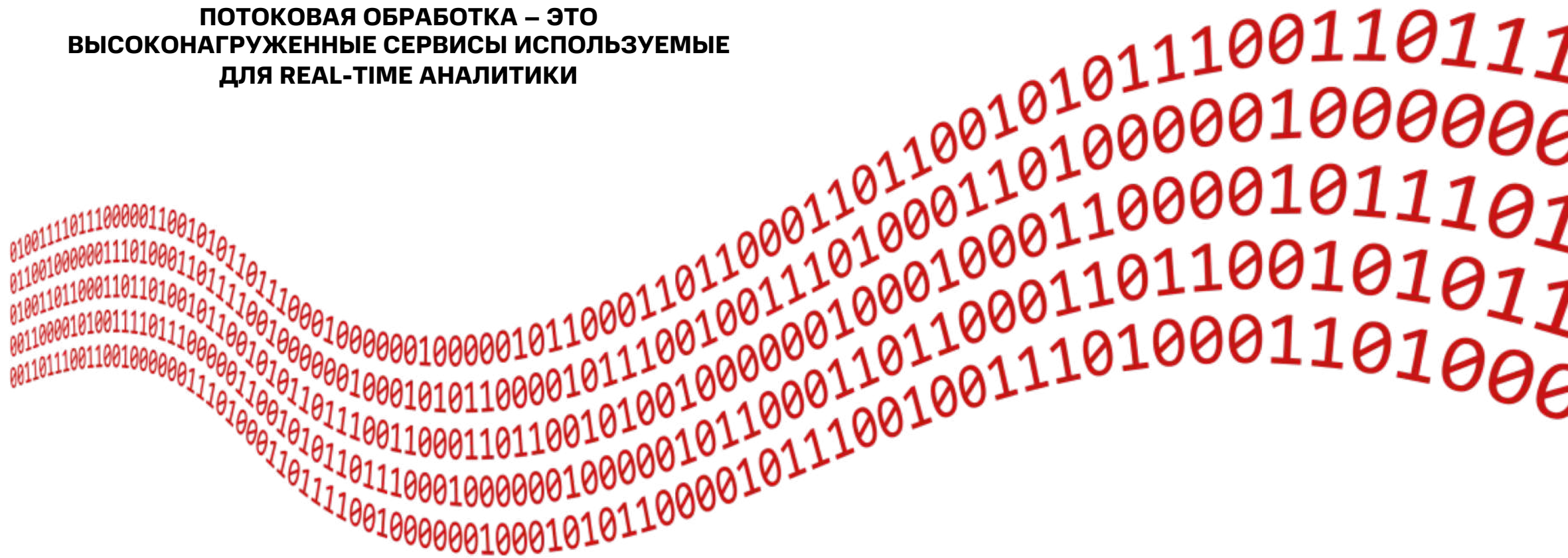
Hello, *Conference!*

ОБ АРХИТЕКТУРНЫХ МОДЕЛЯХ

2

ЧТО ТАКОЕ ПОТОКОВАЯ ОБРАБОТКА BIG DATA

ПОТОКОВАЯ ОБРАБОТКА – ЭТО
ВЫСОКОНАГРУЖЕННЫЕ СЕРВИСЫ ИСПОЛЬЗУЕМЫЕ
ДЛЯ REAL-TIME АНАЛИТИКИ



СРАВНЕНИЕ АРХИТЕКТУРНЫХ МОДЕЛЕЙ

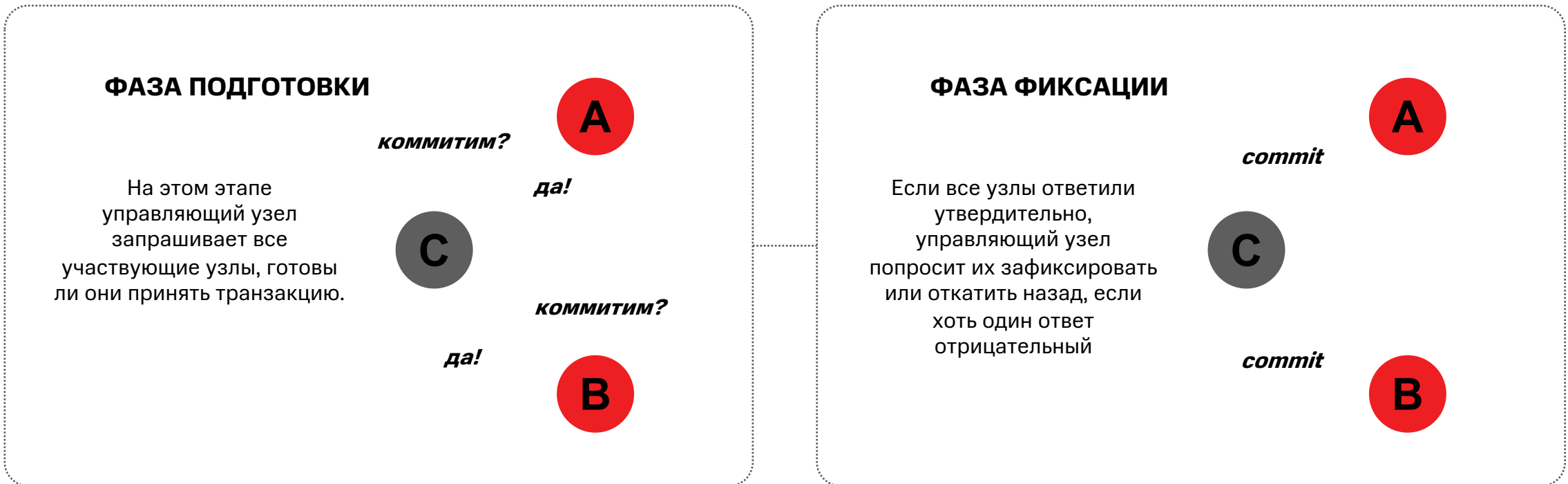


VS



ПОЧЕМУ ОТСУТСТВИЕ ТРАНЗАКЦИЙ ЭТО ХОРОШО

ОРГАНИЗОВАТЬ ТРАНЗАКЦИОННОСТЬ В РАСПРЕДЕЛЕННОЙ СИСТЕМЕ ОЧЕНЬ СЛОЖНО



О НАШЕМ ПРИМЕРЕ РЕАЛИЗАЦИИ ГЕО-ПЛАТФОРМЫ

3

КАКИЕ ПЕРЕД НАМИ СТОЯЛИ ЗАДАЧИ

ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ



Много кейсов объединённых одним бизнес-доменом – «Гео»:

- текущее местоположение,
- вход в гео-зону,
- определение загрузки метро и т.п.

НЕФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ



Производительность (поточная обработка 1 млн. событий/сек, запрос через REST API 10 000 TPS, время отклика до 0,5 сек)



Доступность и надежность (99,99%)

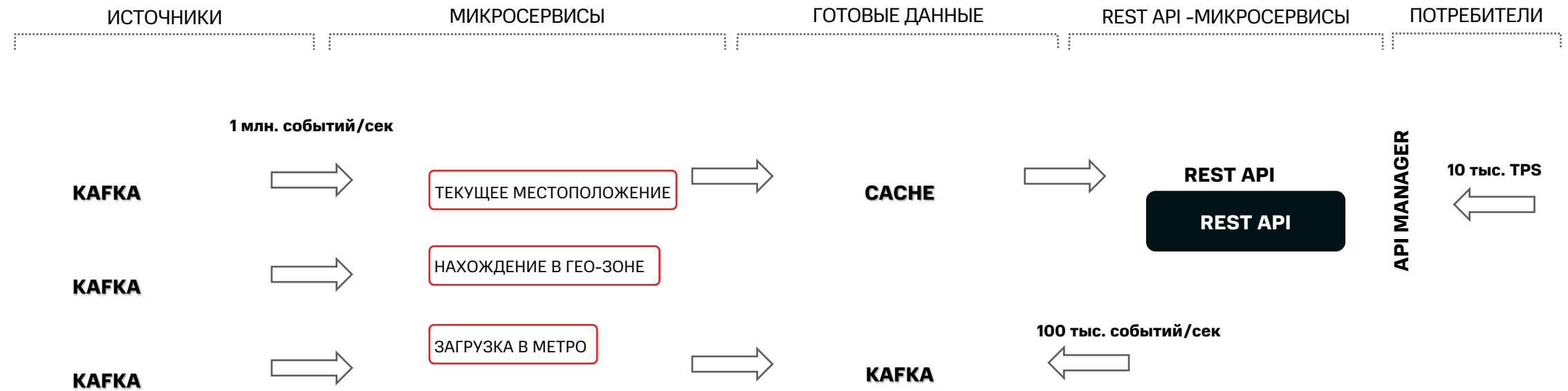


Масштабируемость



Наблюдаемость

КАК БЫ ВЫГЛЯДЕЛИ МИКРОСЕРВИСЫ



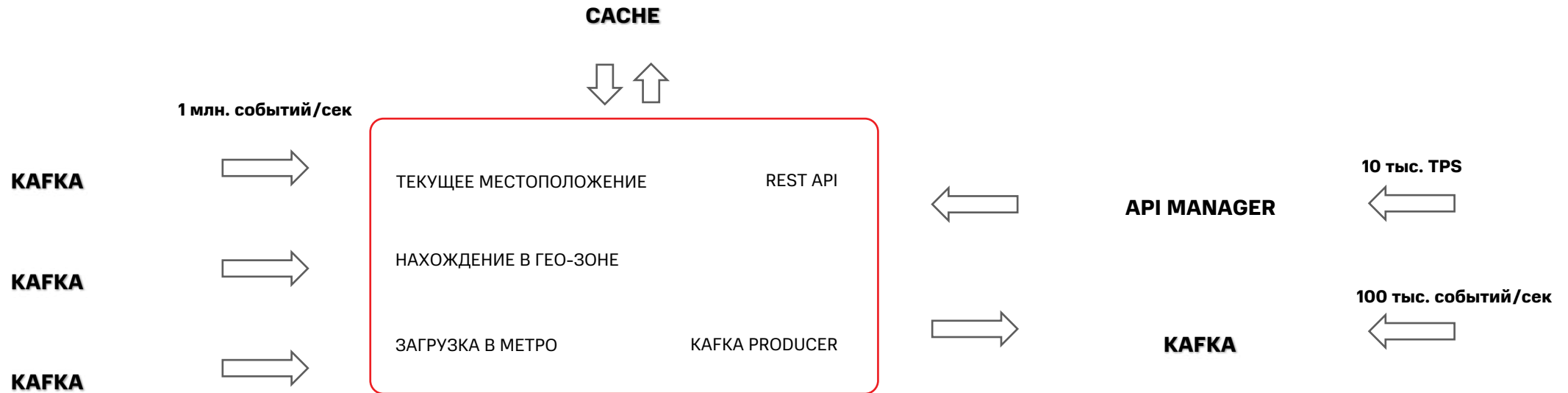
ПЛЮСЫ

- независимое развертывание и обновление (сокращение ТТМ)
- удобное горизонтальное масштабирование - отказоустойчивость
- сервисы легко тестировать
- сервисы могут быть реализованы с использованием различных языков программирования, фреймворков и т.д.
- низкий порог входа для новых разработчиков
- независимая разработка и масштабирование команды

МИНУСЫ

- распределенная система - такую систему сложнее разрабатывать
- сетевые задержки
- сложность эксплуатации

КАК БЫ ВЫГЛЯДЕЛ МОНОЛИТ



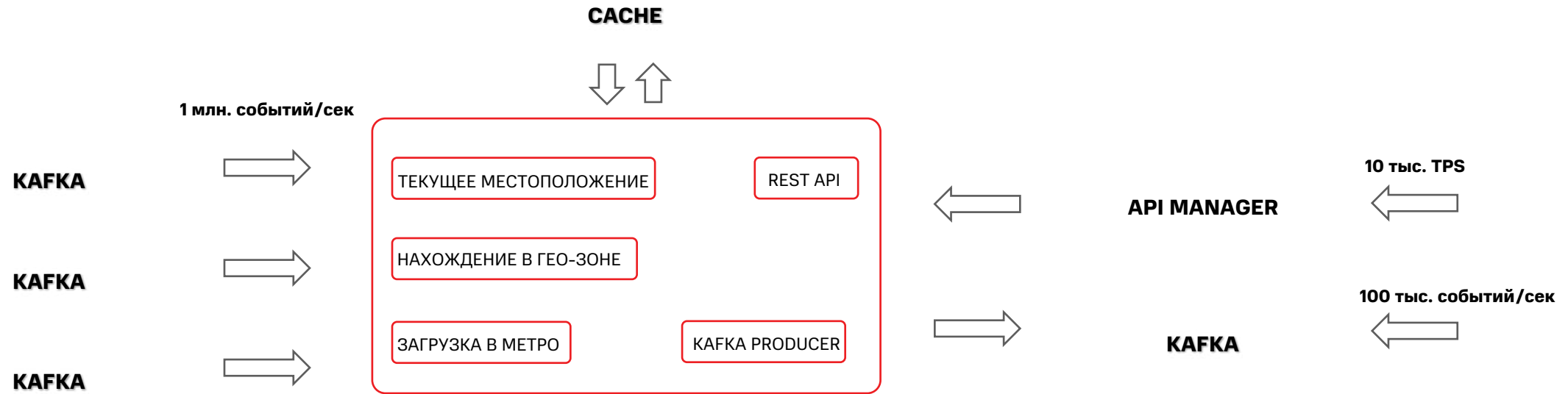
ПЛЮСЫ

- «быстрый старт»
- нет сетевого взаимодействия между модулями монолита

МИНУСЫ

- уменьшение надёжности
- отсутствие гибкости
- сильная связанность
- большой порог вхождения для новых разработчиков
- сложно внедрять новые технологии

А ПОЧЕМУ БЫ НЕ НАПИСАТЬ МОДУЛЬНЫЙ МОНОЛИТ?



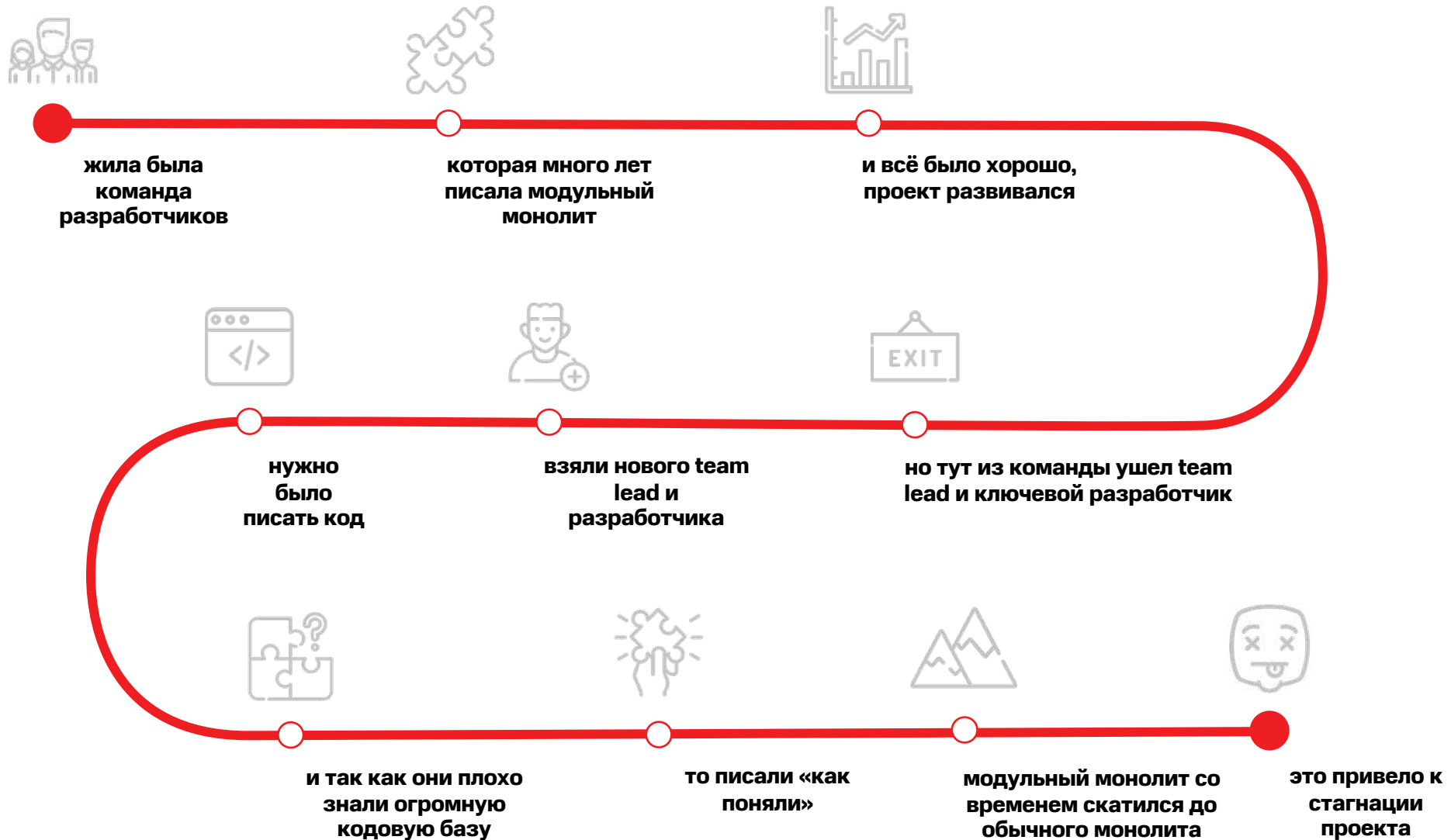
ПЛЮСЫ

- «быстрый старт»
- нет сетевого взаимодействия между модулями монолита

МИНУСЫ

- ~~уменьшение надёжности~~
- ~~отсутствие гибкости~~
- ~~сильная связанность~~
- ~~большой порог вхождения для новых разработчиков~~
- ~~сложно внедрять новые технологии~~
- код для модульного монолита очень сложно писать и поддерживать

ИСТОРИЯ ПРО ОДИН МОДУЛЬНЫЙ МОНОЛИТ

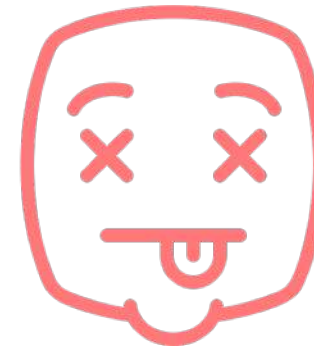


МОРАЛЬ ИСТОРИИ



УХОД КЛЮЧЕВОГО СПЕЦИАЛИСТА

=



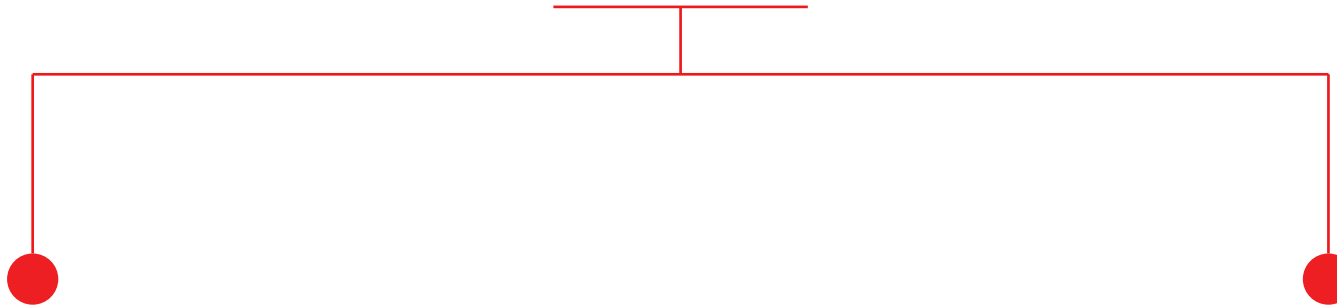
СМЕРТЬ МОДУЛЬНОГО МОНОЛИТА

О ВЫВОДАХ

4

ЗАКЛЮЧЕНИЕ

МИКРОСЕРВИСНАЯ АРХИТЕКТУРА НЕ ПАНАЦЕЯ
решая одни проблемы, получаем другие



КОГДА СЛЕДУЕТ ПОДУМАТЬ НАД ПРИМЕНЕНИЕМ

- большая команда разработчиков или команд несколько
 - команда специалистов высокой квалификации
 - хорошо описан бизнес контекст
- часто меняющиеся бизнес требования и нужен быстрый ТТМ

КОГДА НЕ СЛЕДУЕТ ПРИМЕНЯТЬ

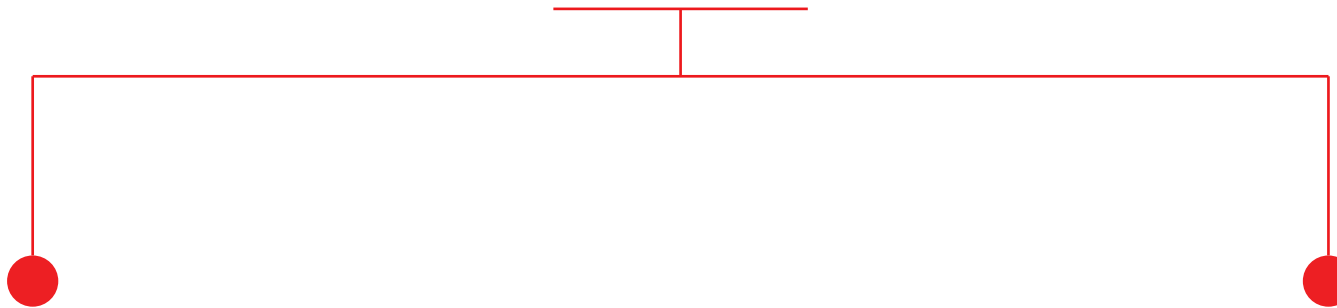
- плохо описана предметная область
 - мало времени на реализацию
- маленькая команд и/или в ней мало специалистов высокой квалификации

ВОПРОСЫ И ОТВЕТЫ

5

ЗАКЛЮЧЕНИЕ

МИКРОСЕРВИСНАЯ АРХИТЕКТУРА НЕ ПАНАЦЕЯ
решая одни проблемы, получаем другие



КОГДА СЛЕДУЕТ ПОДУМАТЬ НАД ПРИМЕНЕНИЕМ

- большая команда разработчиков или команд несколько
 - команда специалистов высокой квалификации
 - хорошо описан бизнес контекст
- часто меняющиеся бизнес требования и нужен быстрый ТТМ

КОГДА НЕ СЛЕДУЕТ ПРИМЕНЯТЬ

- плохо описана предметная область
 - мало времени на реализацию
- маленькая команд и/или в ней мало специалистов высокой квалификации

 Hello, **Conference!**_

СПАСИБО
ЗА ВНИМАНИЕ

MTC