

Predictive maintenance

Лекция 1
Введение в Big Data и IIoT

Власов Кирилл Вячеславович



О купце

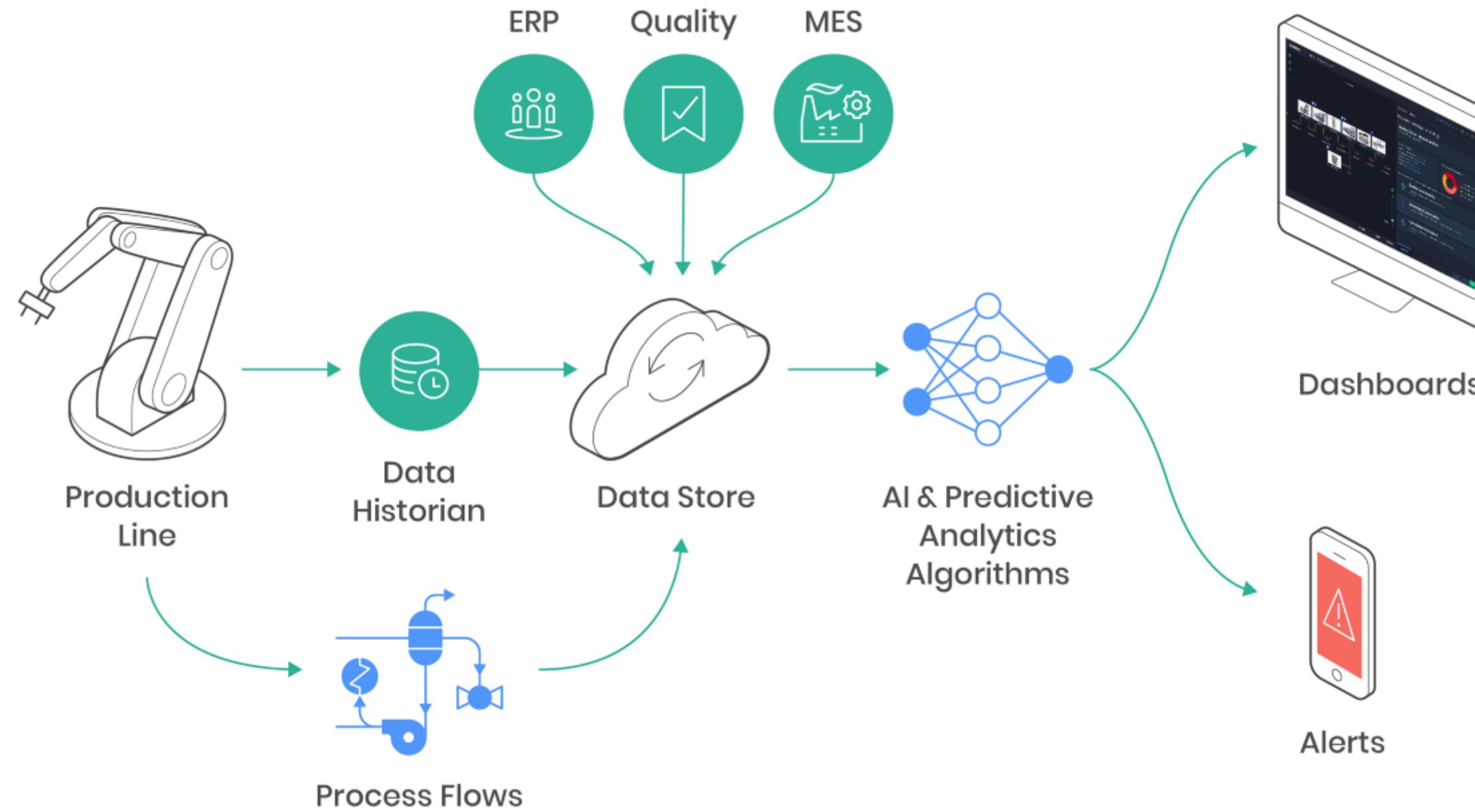


Industrial
IoT

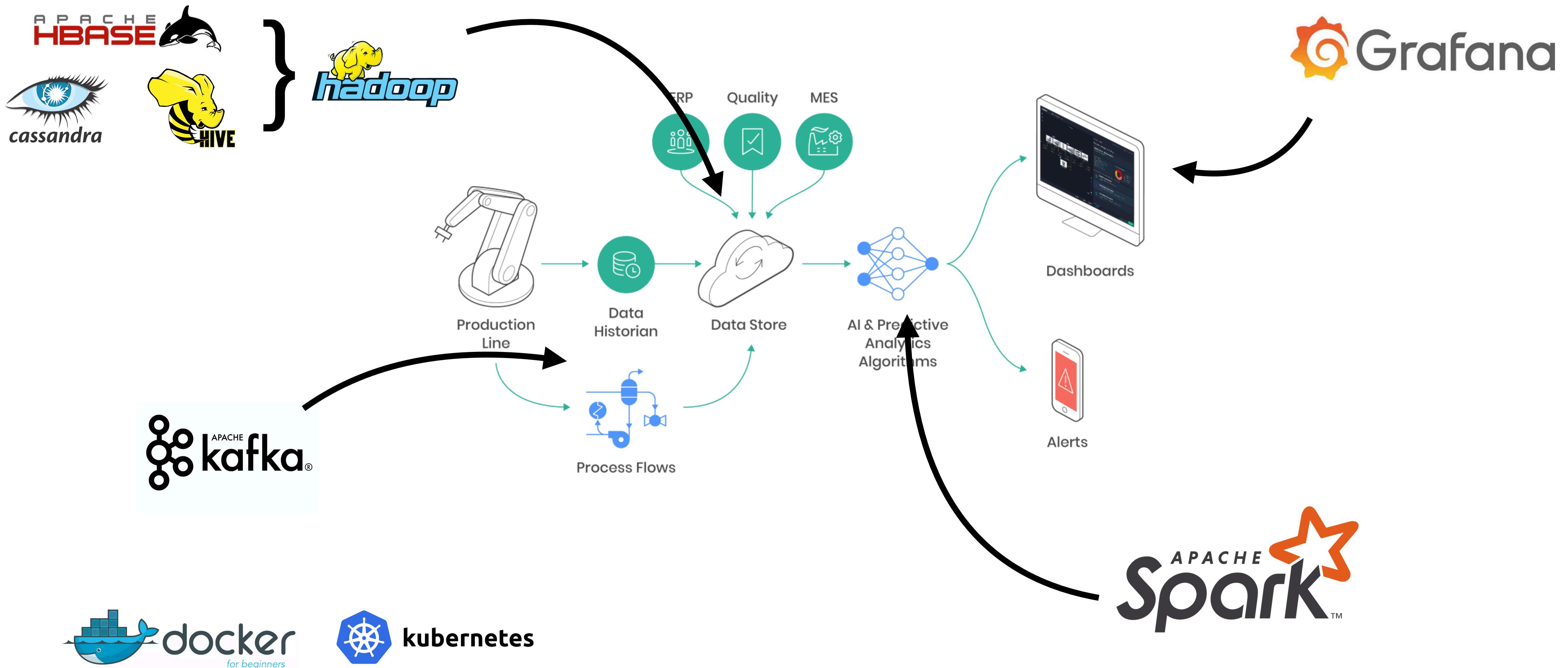


PREDICTIVE
MAINTENANCE

Predictive maintenance



Predictive maintenance



Что это такое эта ваша **BIG DATA** ?



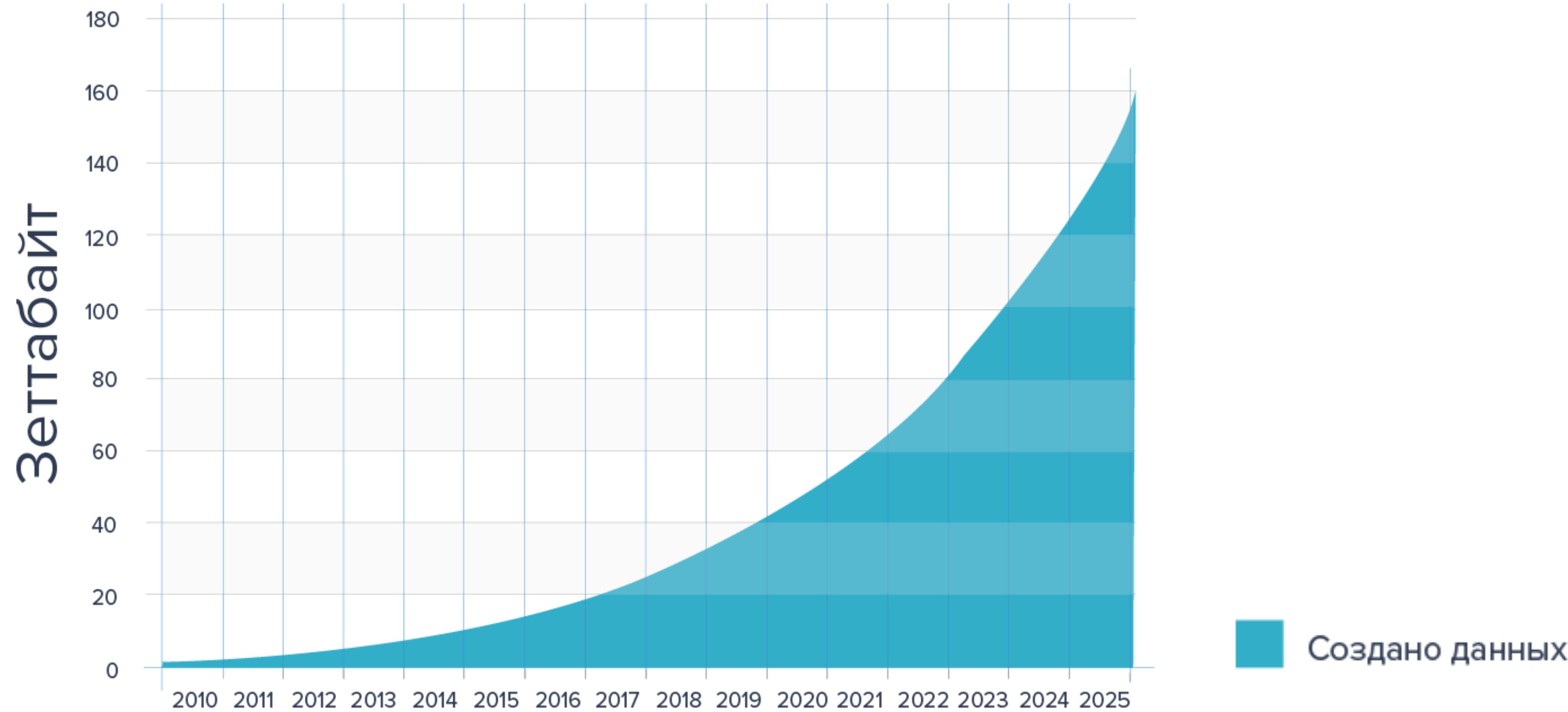


WIKIPEDIA

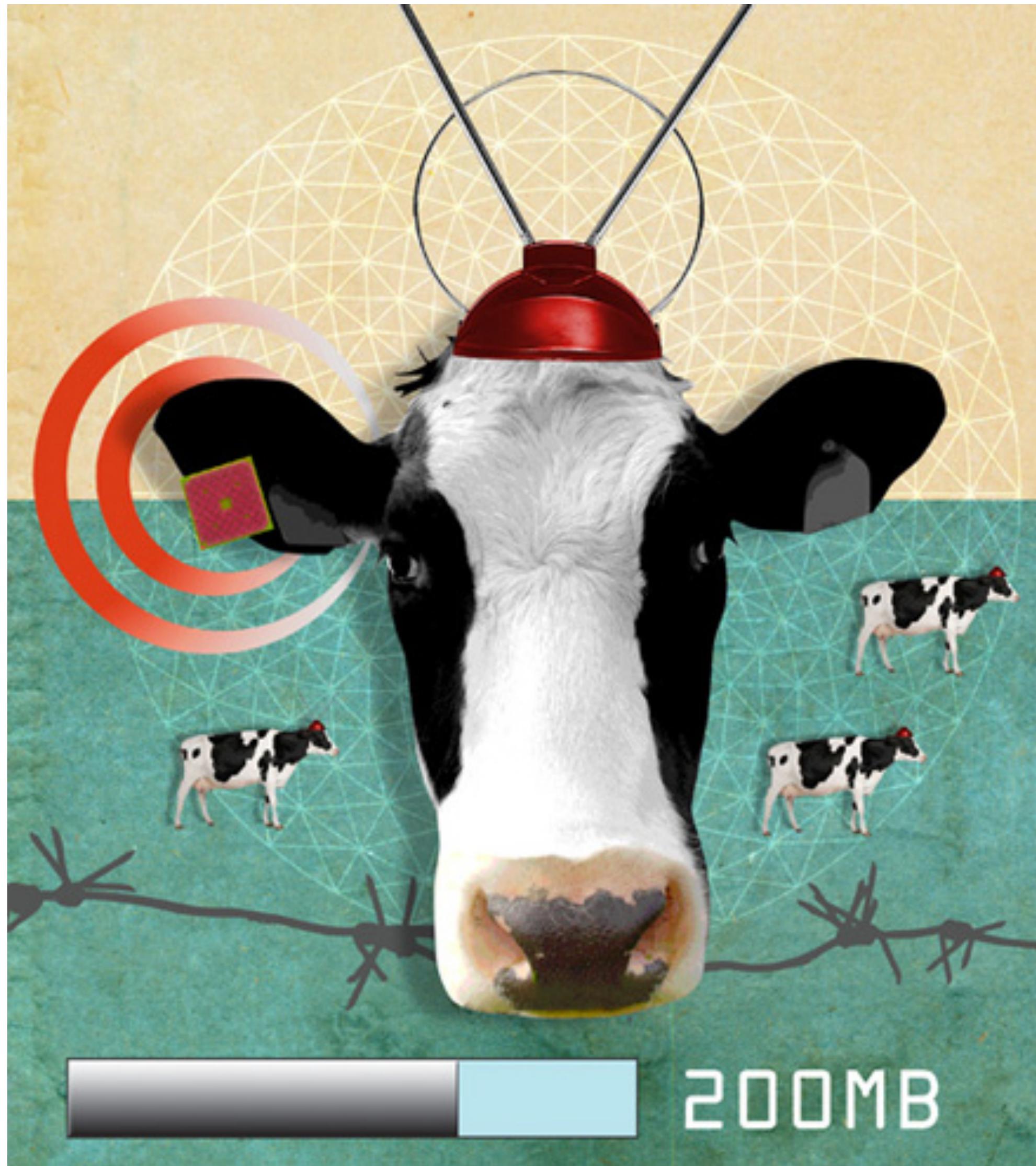


Большие данные (англ. **big data**) — обозначение структурированных и неструктурированных данных огромных объёмов и значительного многообразия, эффективно обрабатываемых горизонтально масштабируемыми программными инструментами, появившимися в конце 2000-х годов и альтернативных традиционным системам управления базами данных и решениям класса Business Intelligence.

Количество данных в мире

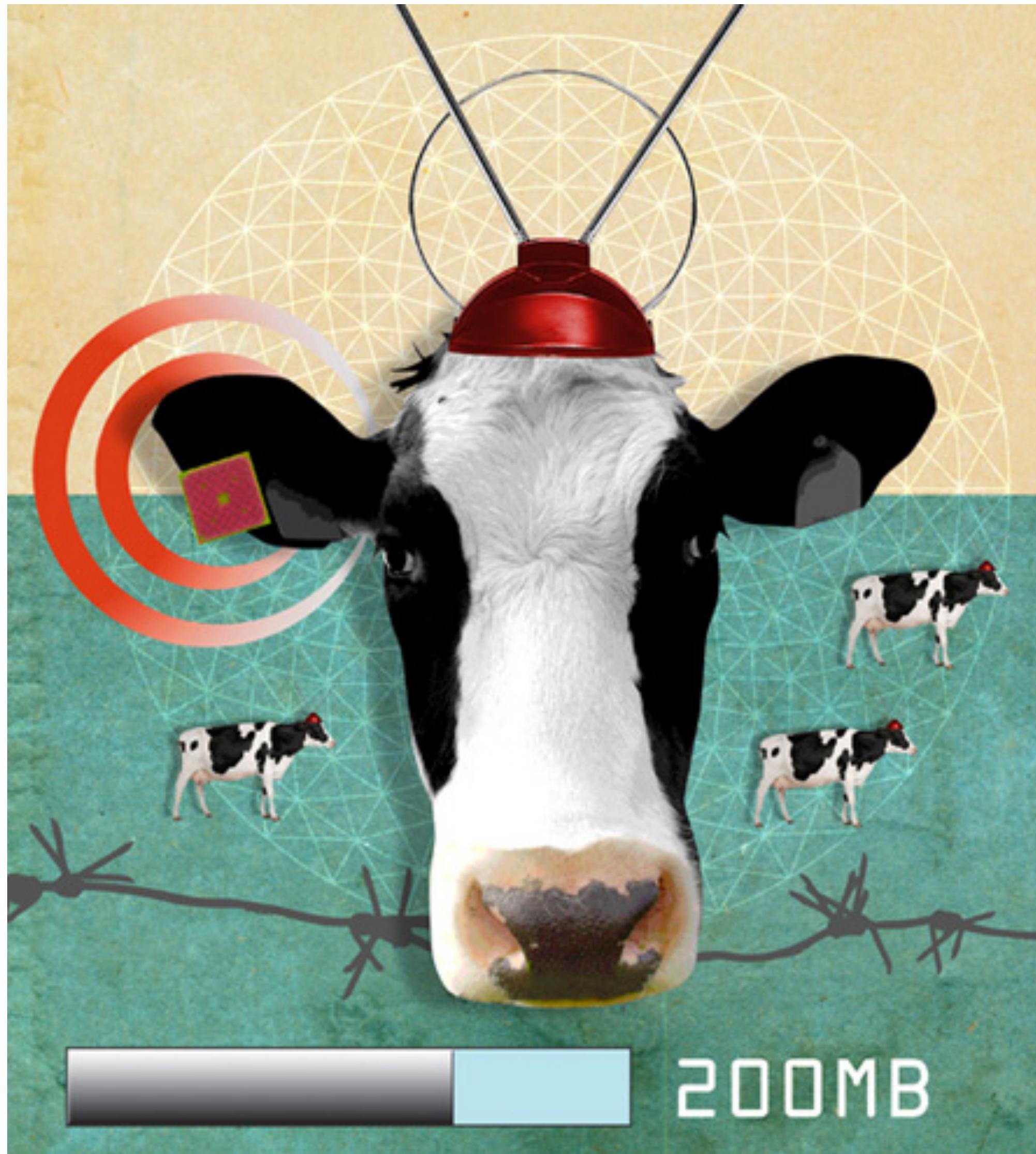


Откуда столько данных?

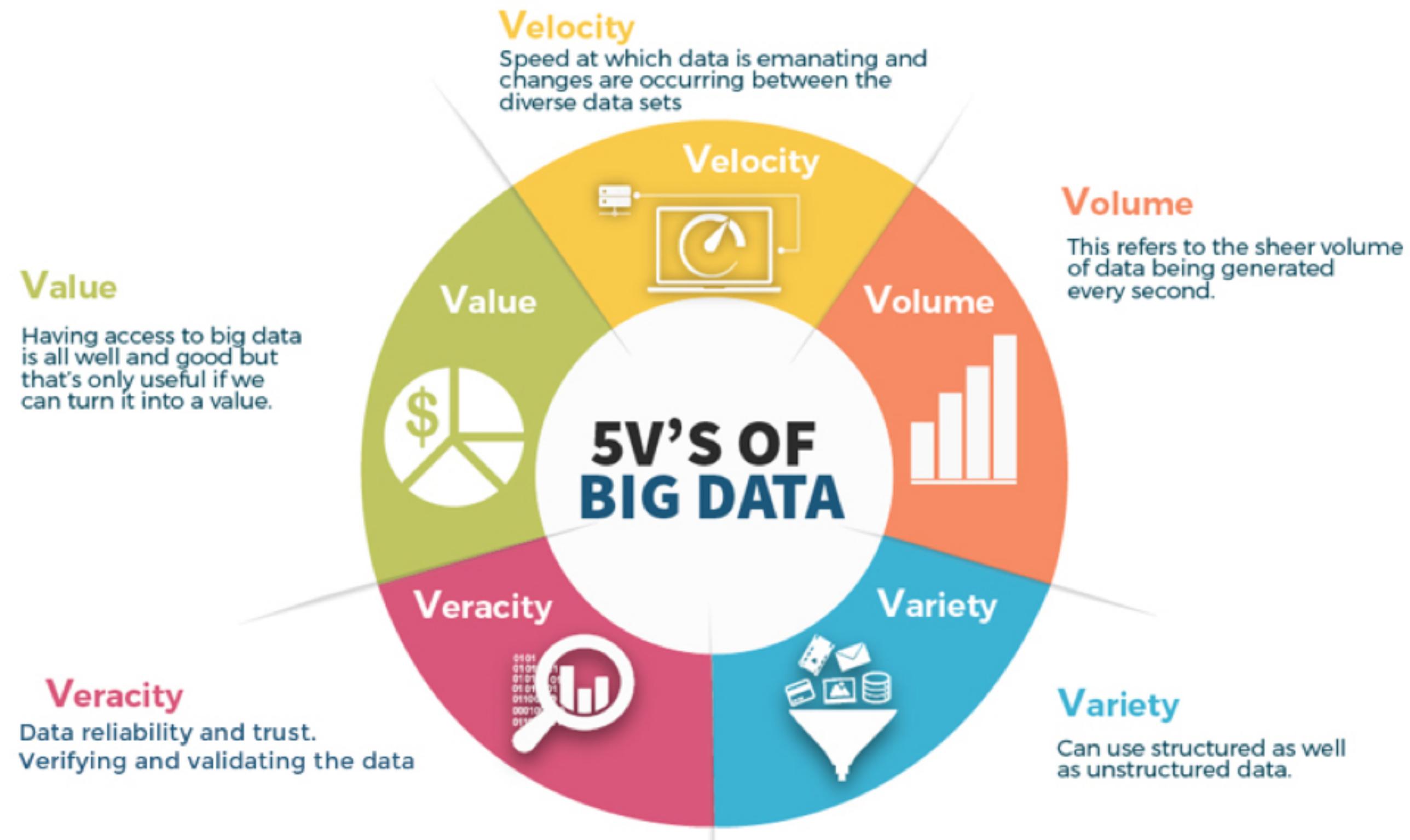


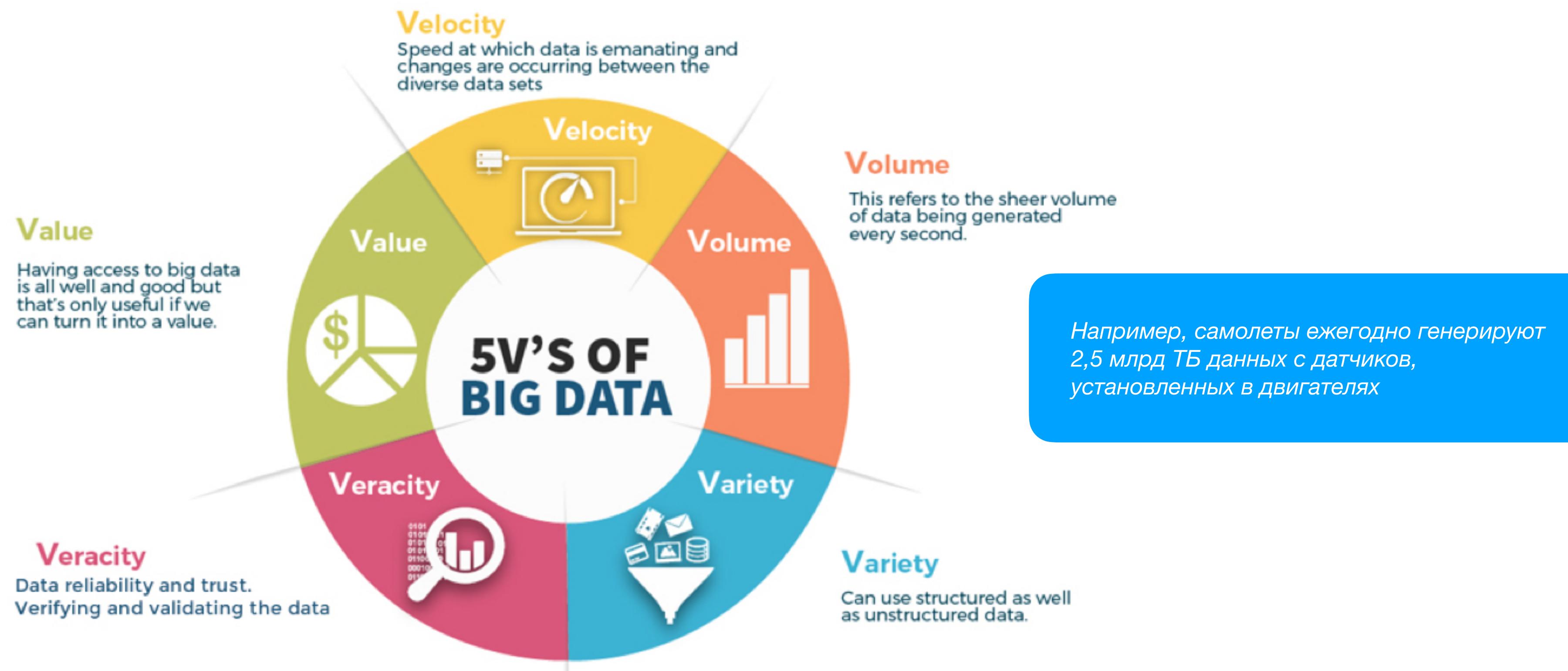
On average, each cow generates about 200 megabytes of information a year.

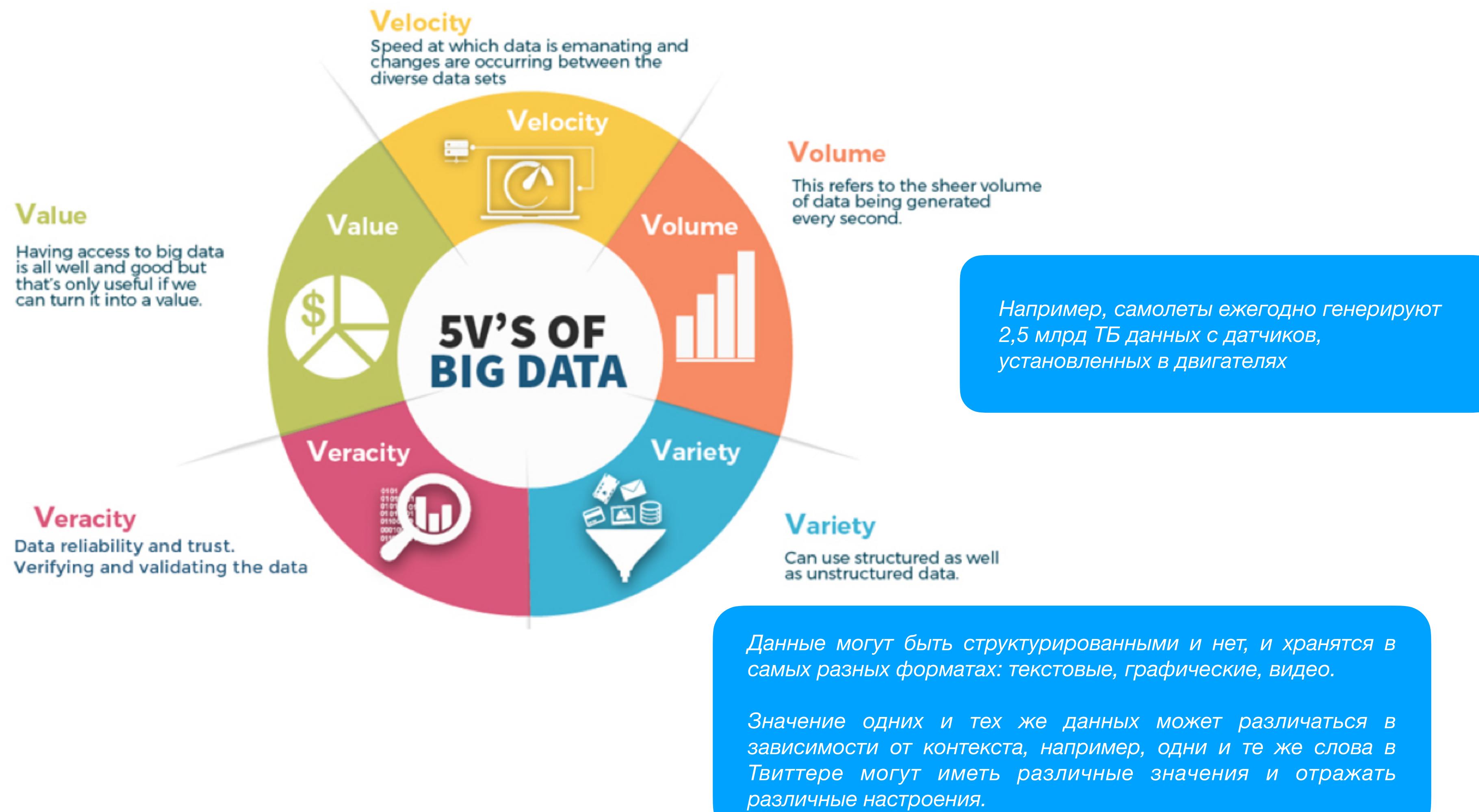
Откуда столько данных?



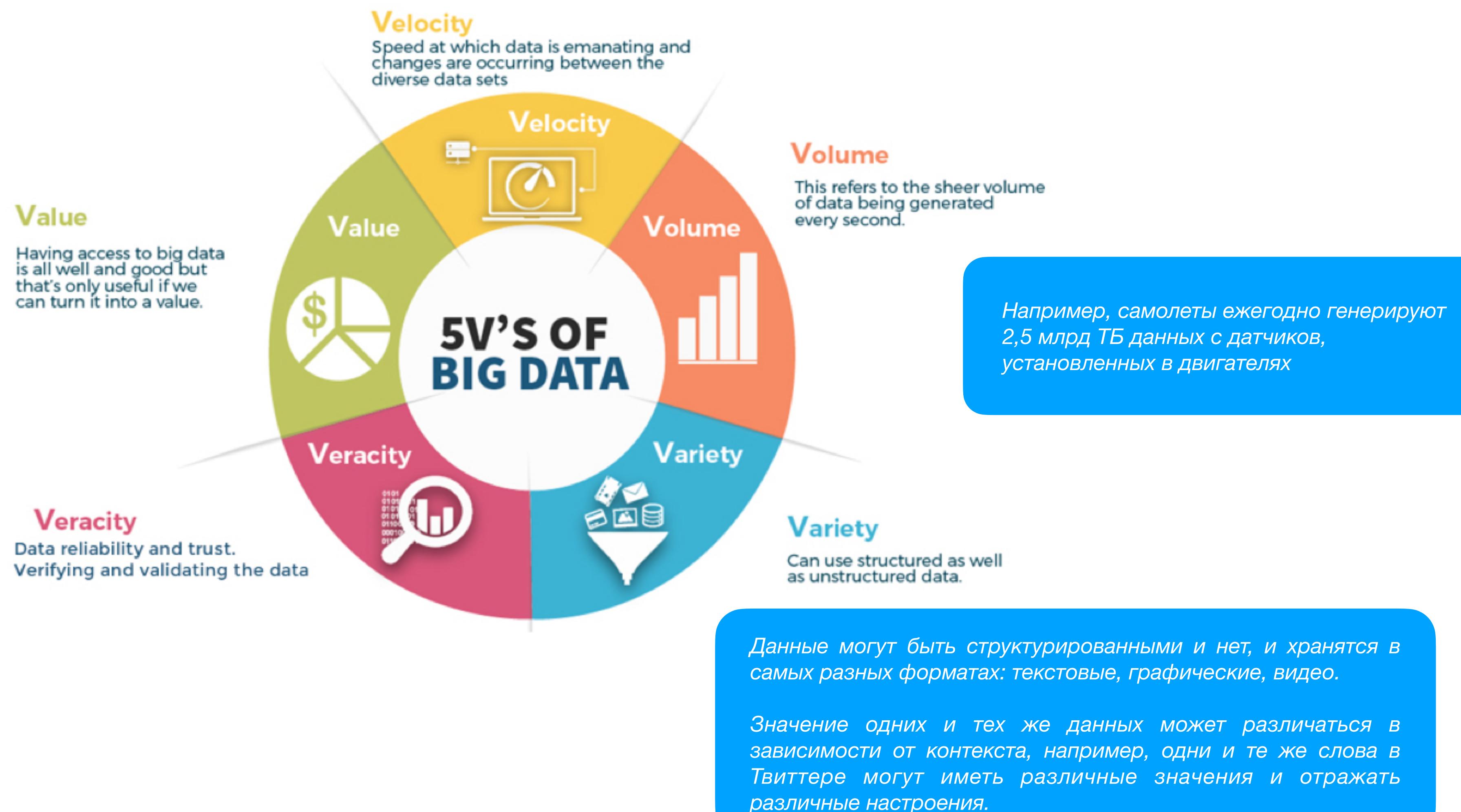
В мире IoT даже коровы будут связаны. В специальном отчете The Economist под названием «Дополненный бизнес» описано, как будут контролироваться коровы. Голландская стартап-компания Sparked внедряет датчики в уши крупного рогатого скота. Это позволяет фермерам следить за здоровьем коров и отслеживать их передвижение, обеспечивая людям более здоровый и обильный запас мяса. В среднем каждая корова генерирует около 200 мегабайт информации в год



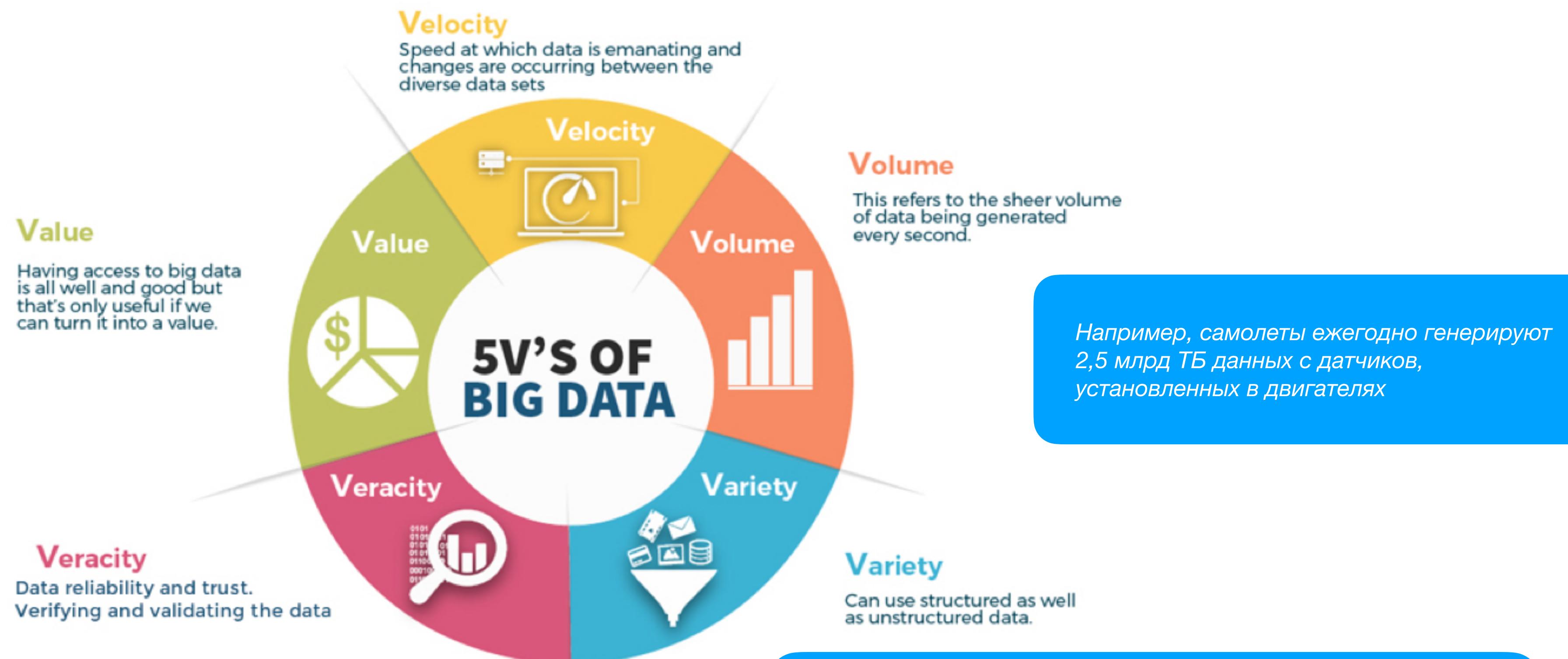




Данные постоянно изменяются, например каждую минуту в мире выполняется почти 2,5 миллиона запросов к поисковой системе Google



Данные постоянно изменяются, например каждую минуту в мире выполняется почти 2,5 миллиона запросов к поисковой системе Google



Достоверность: безусловно, эта характеристика является крайне важной, поскольку любой анализ будет совершенно бесполезен, если данные окажутся недостоверными.

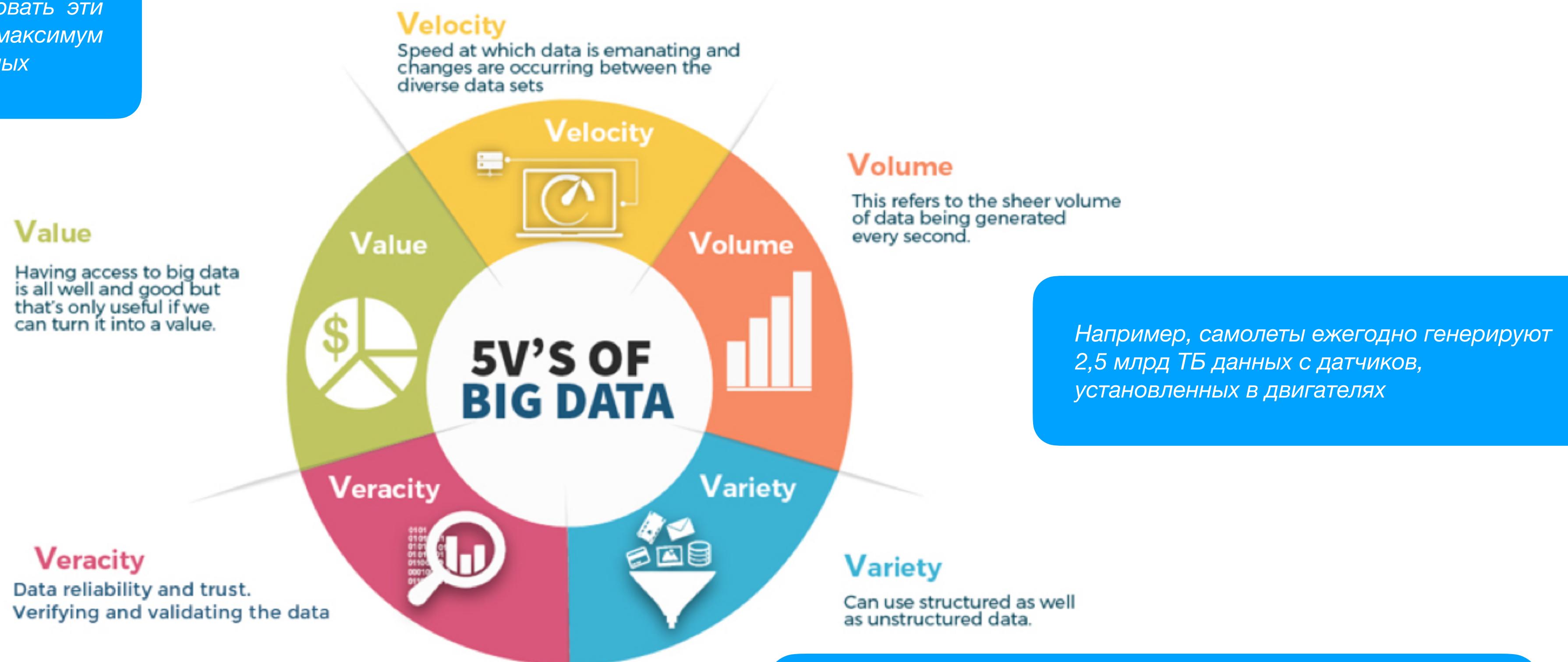
Данные могут быть структурированными и нет, и хранятся в самых разных форматах: текстовые, графические, видео.

Значение одних и тех же данных может различаться в зависимости от контекста, например, одни и те же слова в Твиттере могут иметь различные значения и отражать различные настроения.

Например, самолеты ежегодно генерируют 2,5 млрд ТБ данных с датчиков, установленных в двигателях

Важно то, как Вы будете использовать эти данные, т.к. необходимо извлечь максимум пользы из результатов анализа данных

Данные постоянно изменяются, например каждую минуту в мире выполняется почти 2,5 миллиона запросов к поисковой системе Google



Достоверность: безусловно, эта характеристика является крайне важной, поскольку любой анализ будет совершенно бесполезен, если данные окажутся недостоверными.

Данные могут быть структурированными и нет, и хранятся в самых разных форматах: текстовые, графические, видео.

Значение одних и тех же данных может различаться в зависимости от контекста, например, одни и те же слова в Твиттере могут иметь различные значения и отражать различные настроения.

Например, самолеты ежегодно генерируют 2,5 млрд ТБ данных с датчиков, установленных в двигателях

Примеры в авиации

ИИ победил человека в симуляции воздушного боя

Подробнее: <https://habr.com/post/395525/>

Оценка действий лётчика на этапе посадки

Подробнее: <https://habr.com/post/281455/>

Rolls-Royce: Диагностика самолетов

Подробнее: <http://azure.rbc.ru/business-practices/kak-sekonomit-na-diagnostike-samoletov/>

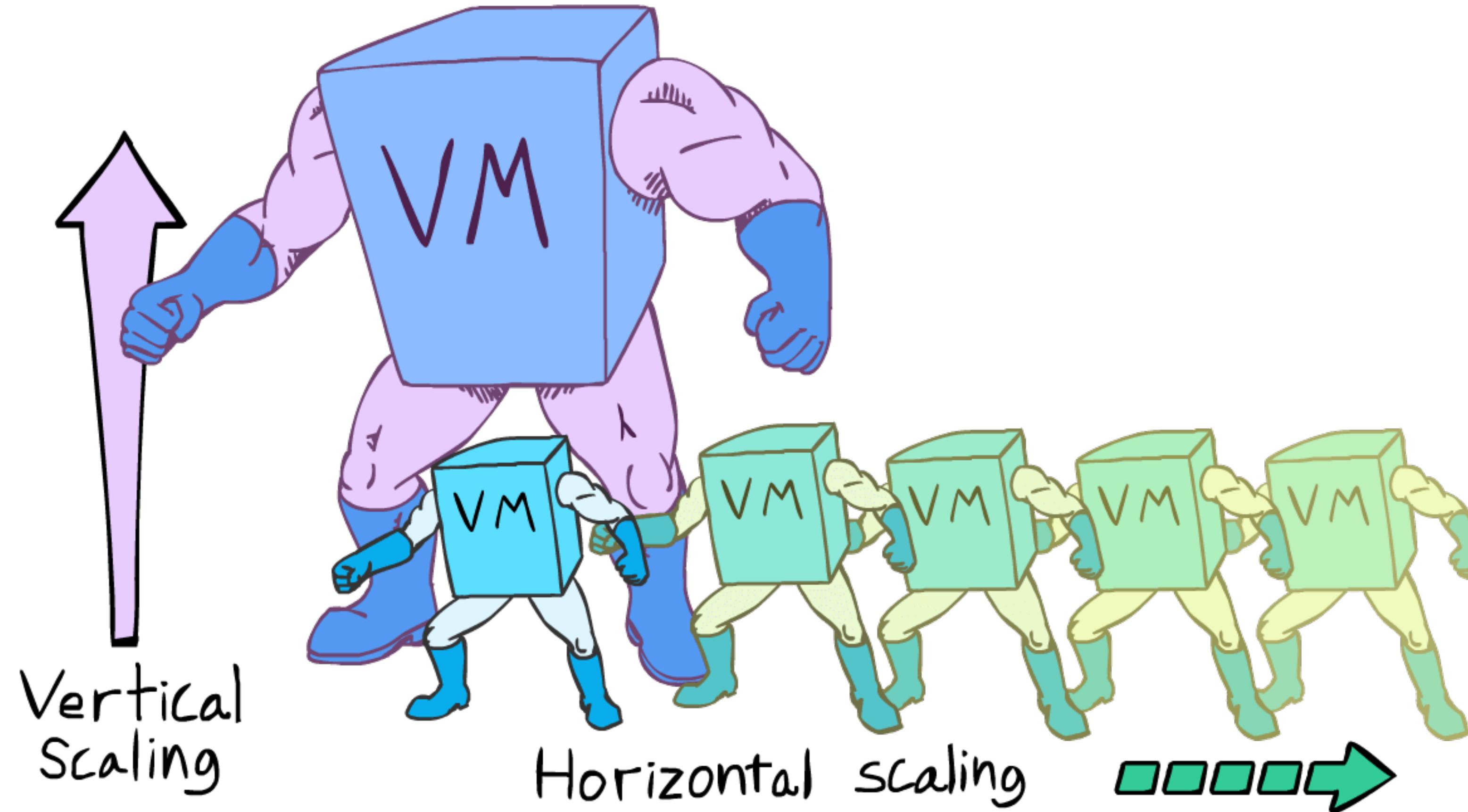
Dubai airport: Оптимизация выходов на посадку и полетов

Система диагностики и прогнозирования технического состояния узлов и агрегатов воздушных судов в Аэрофлоте.

С борта самолета в наземные службы автоматически поступает информация о его техническом состоянии, чтобы задолго до поломки знать об износе деталей. Благодаря предупреждениям, можно заранее подготовить детали, которые необходимо заменить или отремонтировать. Это повышает эффективность использования воздушного судна, сокращая время егоостоя на вынужденный ремонт. Подробнее: https://www.tadviser.ru/index.php/Статья:ИТ_в_авиации

Как это устроено?

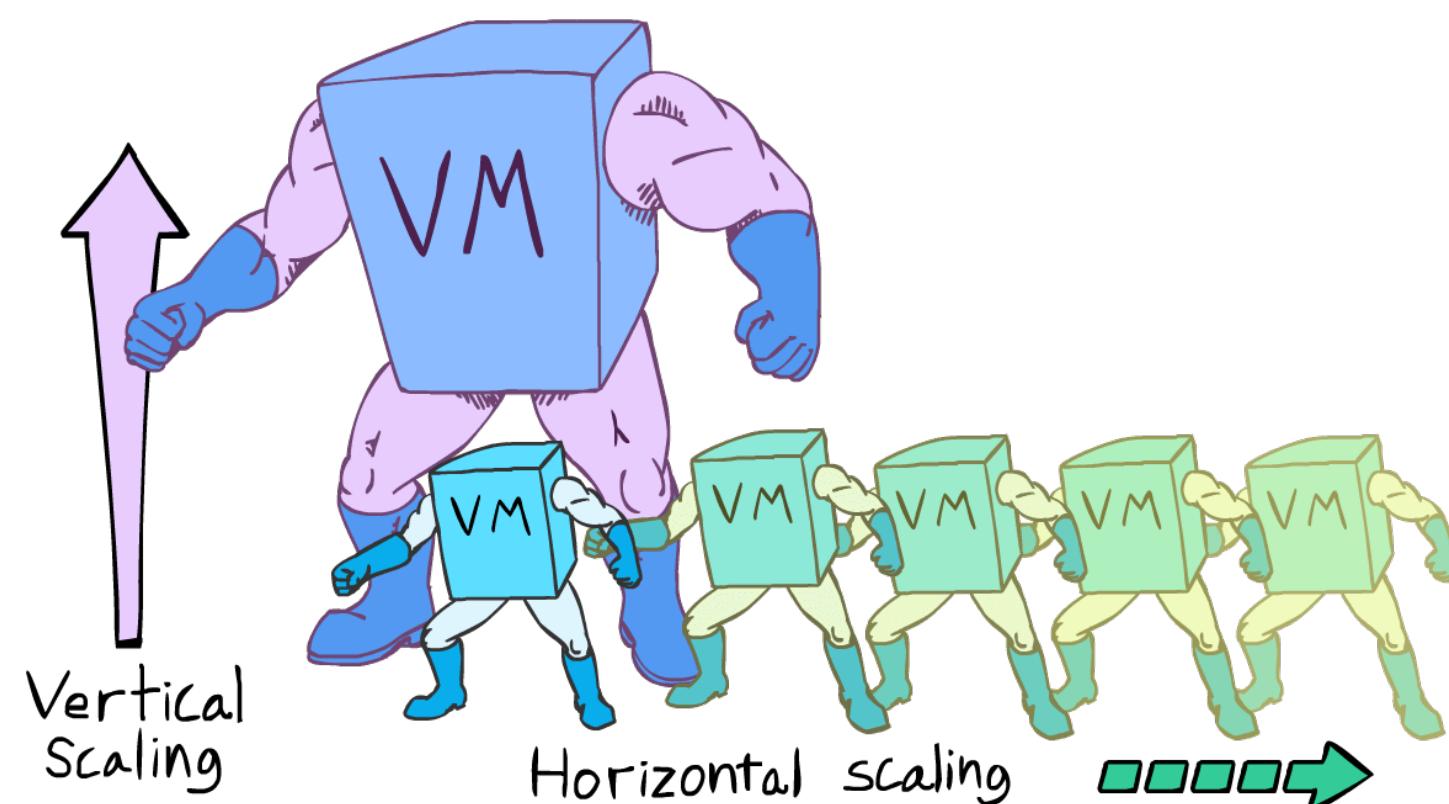
Масштабируемость



Масштабируемость

Vertical Scaling

- + низкая задержка (latency) при доступе к данным для записи и чтения
- нельзя бесконечно увеличивать вместимость и мощность
- рост объема данных может быть быстрее скорости апгрейдов.



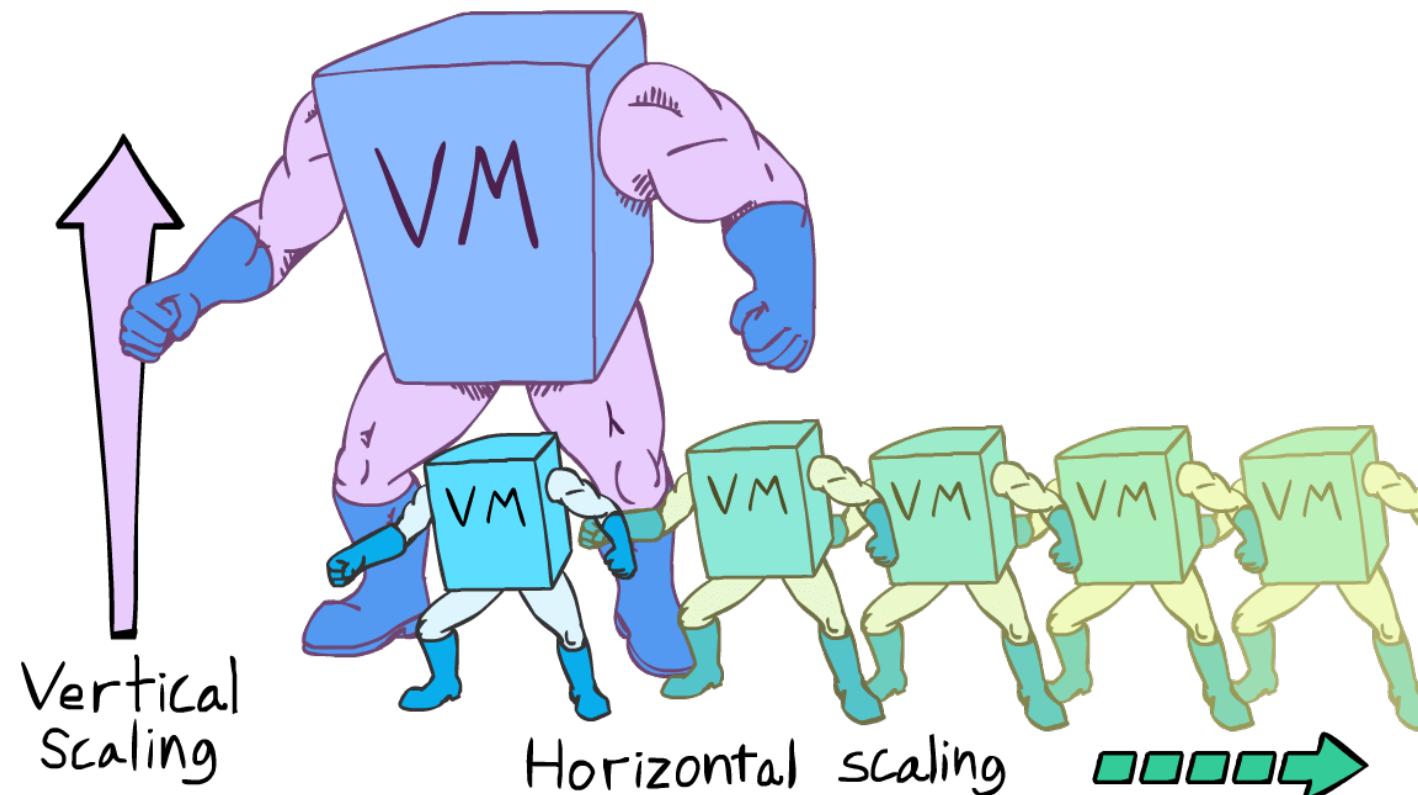
Масштабируемость

Vertical Scaling

- + низкая задержка (latency) при доступе к данным для записи и чтения
- нельзя бесконечно увеличивать вместимость и мощность
- рост объема данных может быть быстрее скорости апгрейдов.

Horizontal scaling

- + легко добавлять дополнительные и убирать неиспользуемые ресурсы из кластера
- + мощность кластера можно увеличивать почти бесконечно.
- в связи с необходимостью работы с данными на разных серверах увеличивается задержка при передаче данных по сети.



Замечание: при проектировании горизонтально масштабируемой системы, нужно помнить что в среднем один HDD ломается раз в три года, значит в кластере на 1000 HDD каждый день будет выходить из строя один диск.

Как это работает?

"As a manager who worked under Michael Fisher and Marty Abbott during my time at PayPal/eBay, the opportunity to directly absorb the lessons and experiences presented in this book is invaluable to me now working at Facebook."

—Yishan Wong, former CEO, Reddit, and former director of engineering, Facebook



THE ART OF SCALABILITY

Scalable Web Architecture, Processes, and Organizations
for the Modern Enterprise

S E C O N D E D I T I O N

MARTIN L. ABBOTT

MICHAEL T. FISHER

FOREWORD BY MARTY CAGAN, FOUNDER, SILICON VALLEY PRODUCT GROUP

"As a manager who worked under Michael Fisher and Marty Abbott during my time at PayPal/eBay, the opportunity to directly absorb the lessons and experiences presented in this book is invaluable to me now working at Facebook."

—Yishan Wong, former CEO, Reddit, and former director of engineering, Facebook



THE ART OF SCALABILITY

Scalable Web Architecture, Processes, and Organizations
for the Modern Enterprise

S E C O N D E D I T I O N

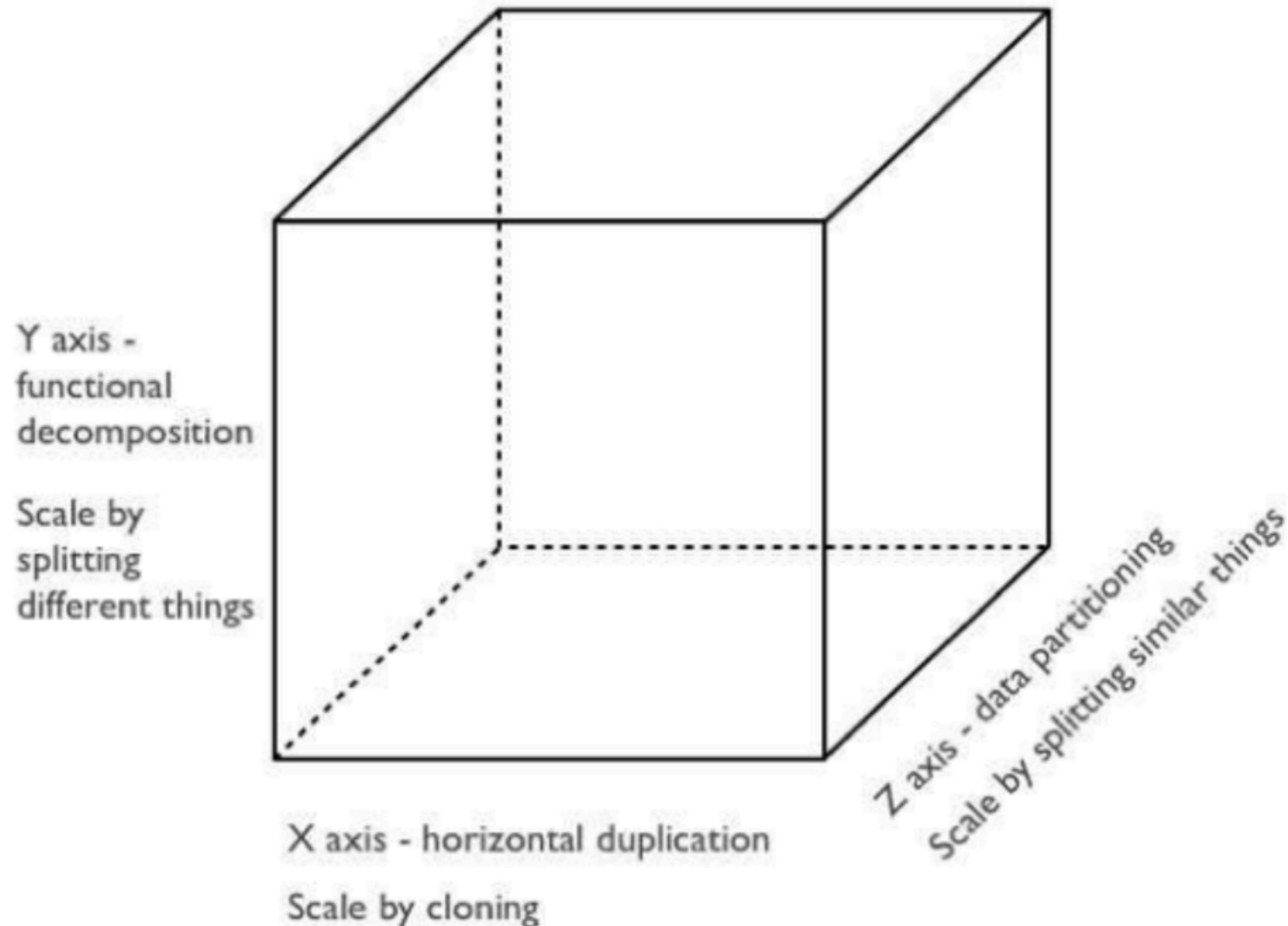
MARTIN L. ABBOTT

MICHAEL T. FISHER

FOREWORD BY MARTY CAGAN, FOUNDER, SILICON VALLEY PRODUCT GROUP

Три измерения масштабирования

Три измерения масштабирования



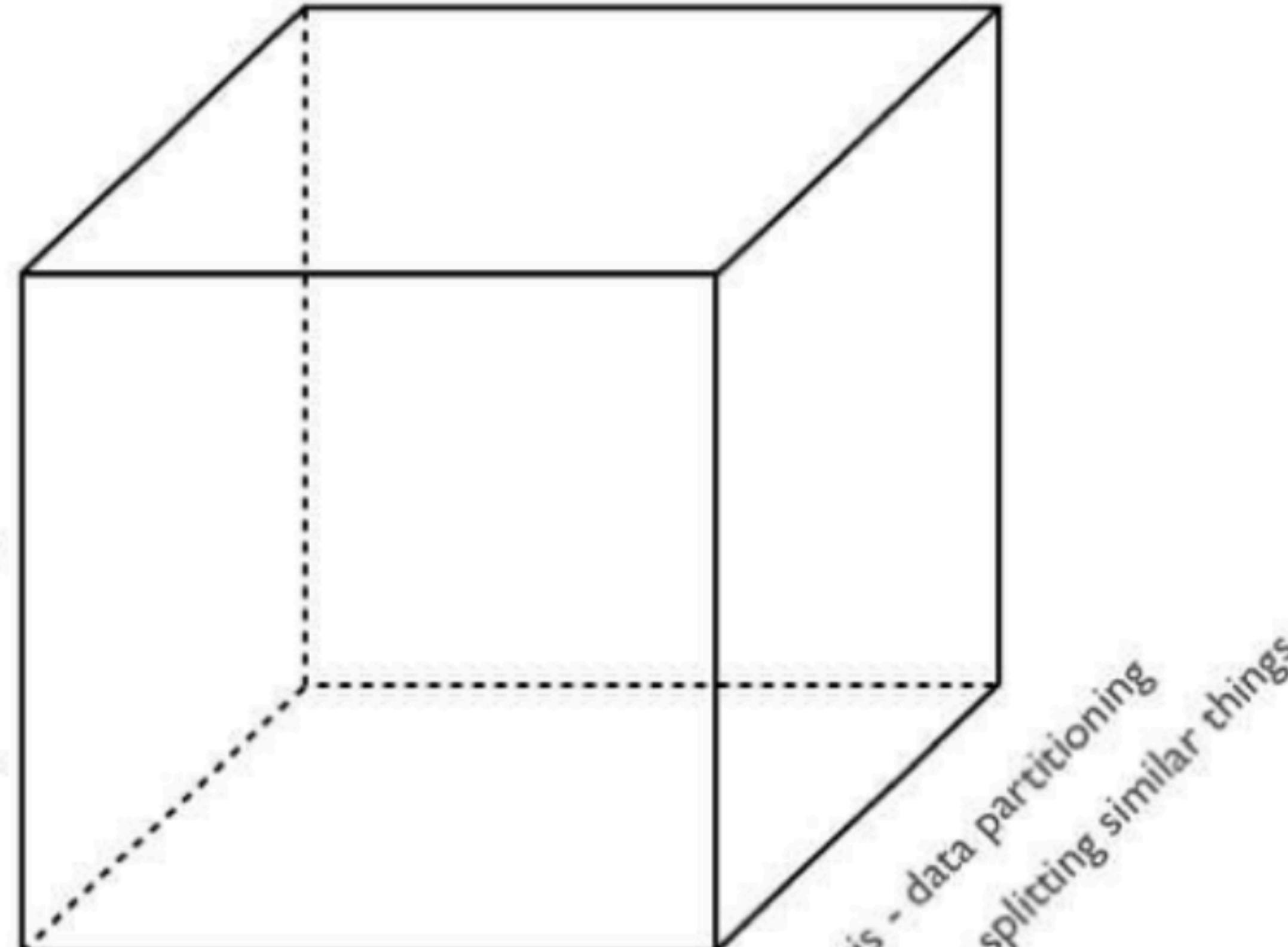
Три подхода к масштабированию

Mirroring

горизонтальное
клонирование всех данных



Y axis -
functional
decomposition
Scale by
splitting
different things



X axis - horizontal duplication
scale by cloning

Z axis - data partitioning
Scale by splitting similar things

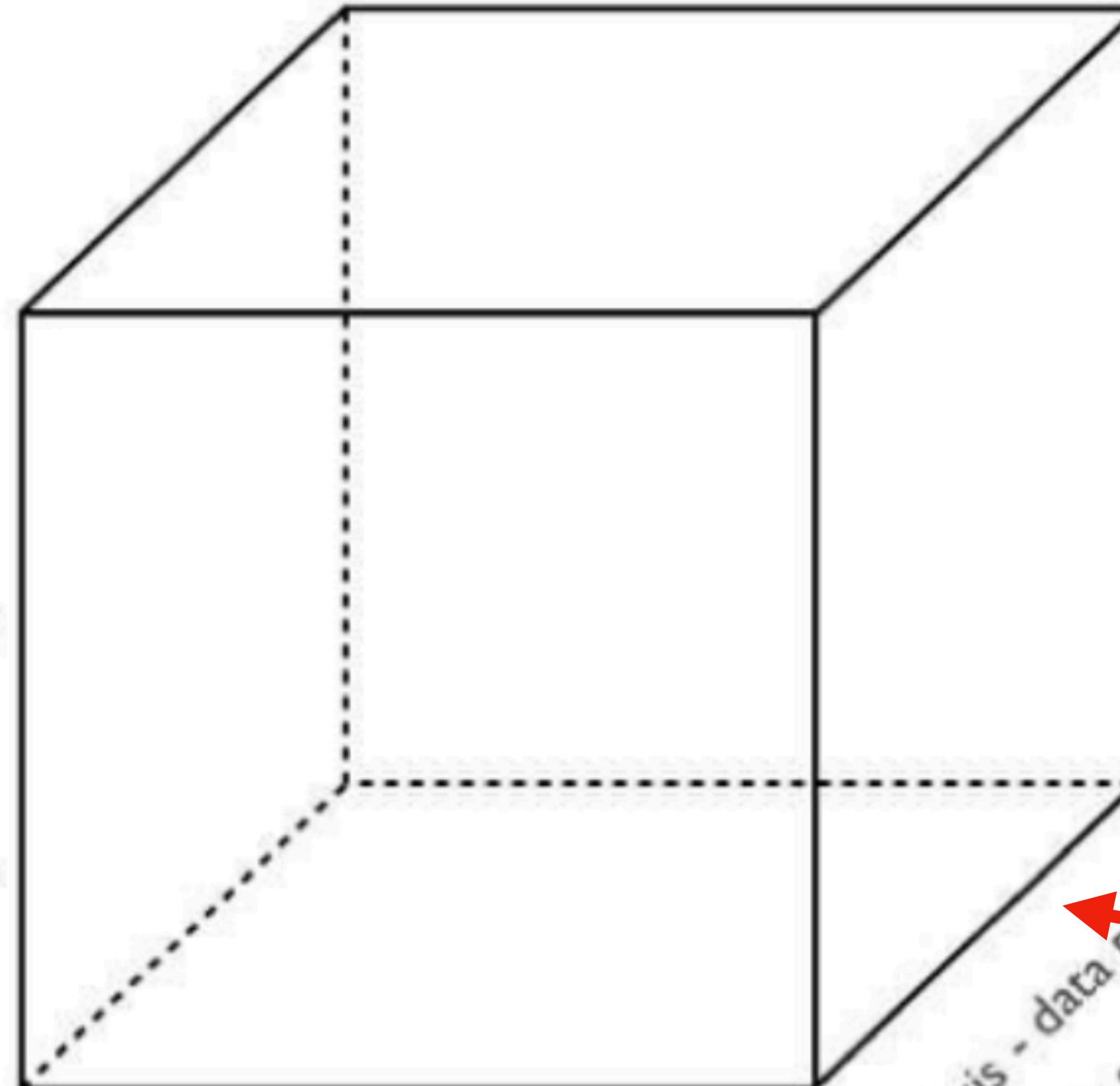
Три подхода к масштабированию

Mirroring

горизонтальное
клонирование всех данных

Y axis -
functional
decomposition

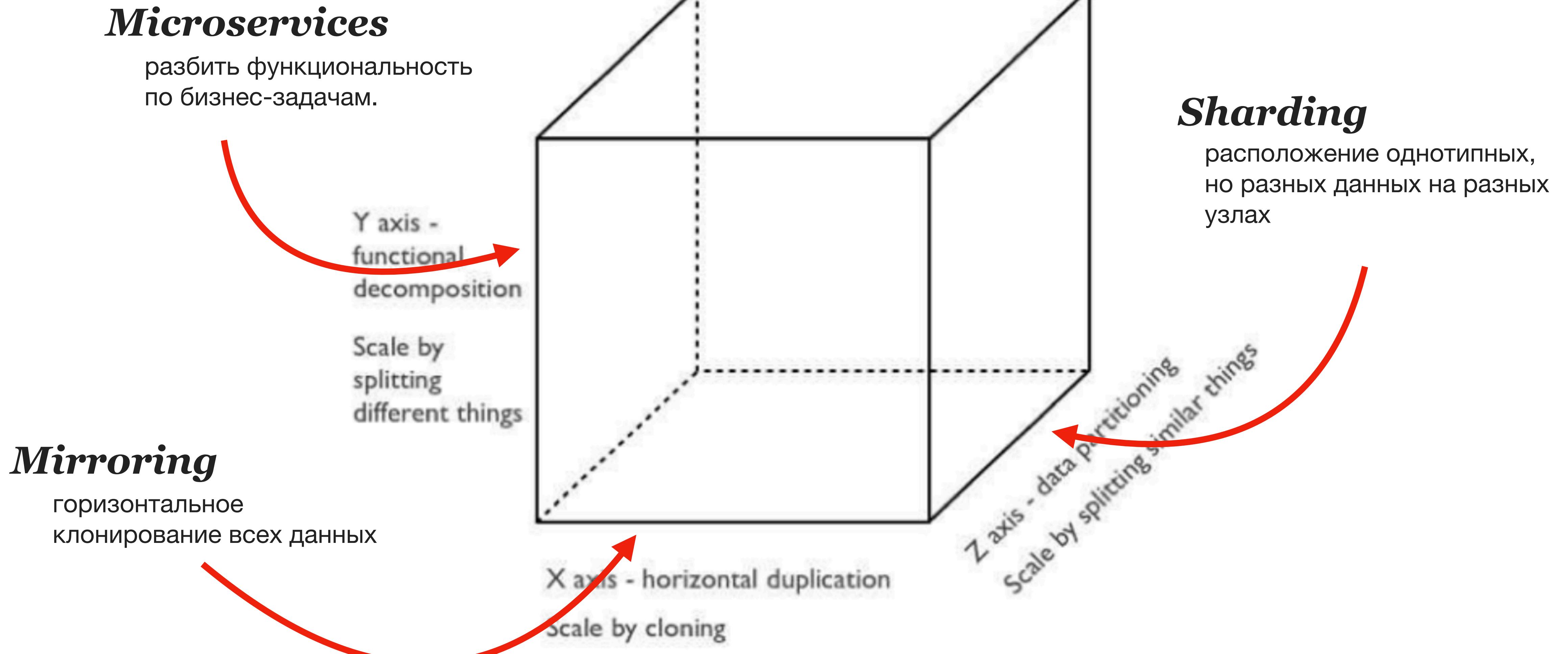
Scale by
splitting
different things



Sharding

расположение однотипных,
но разных данных на разных
узлах

Три подхода к масштабированию



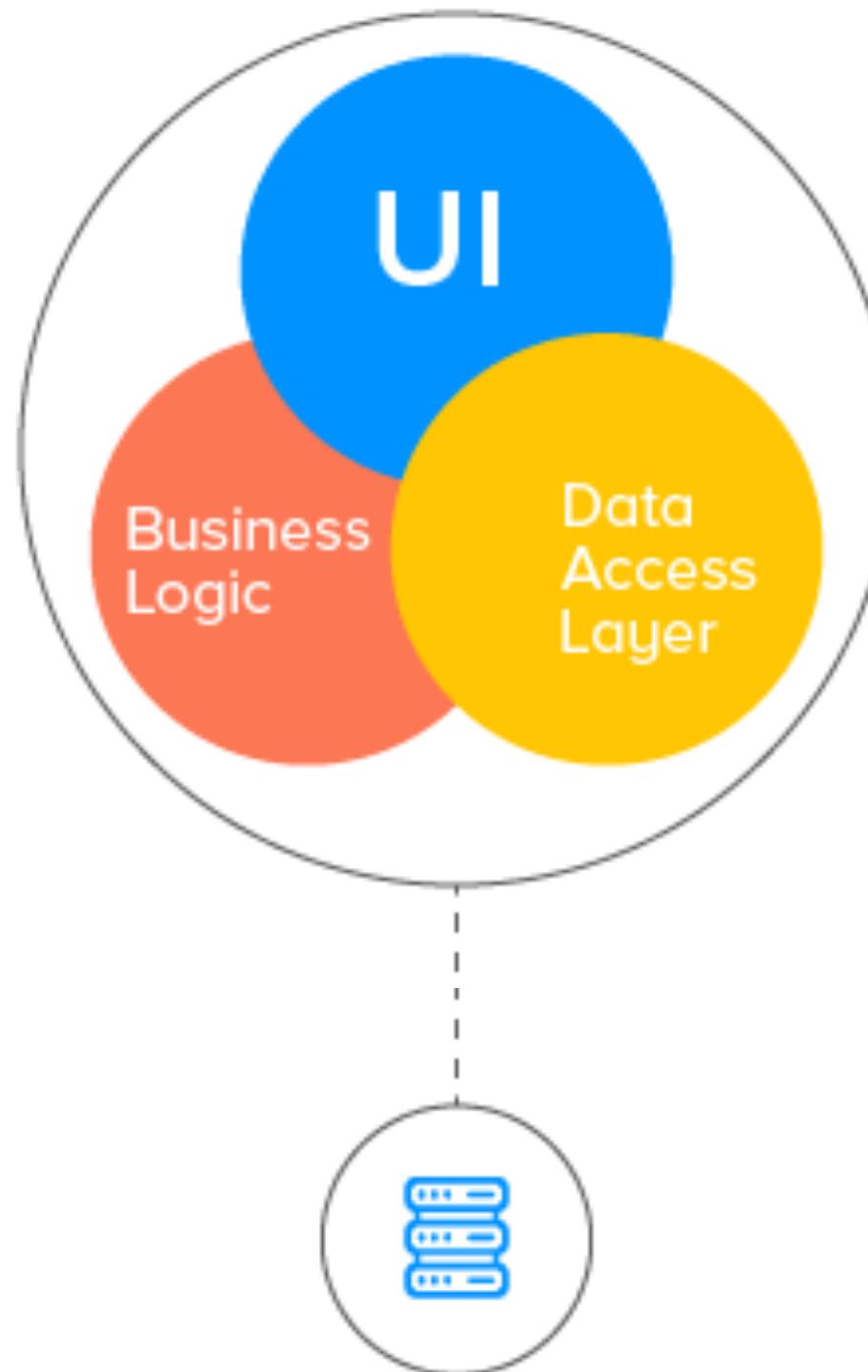
Microservices



Monolith

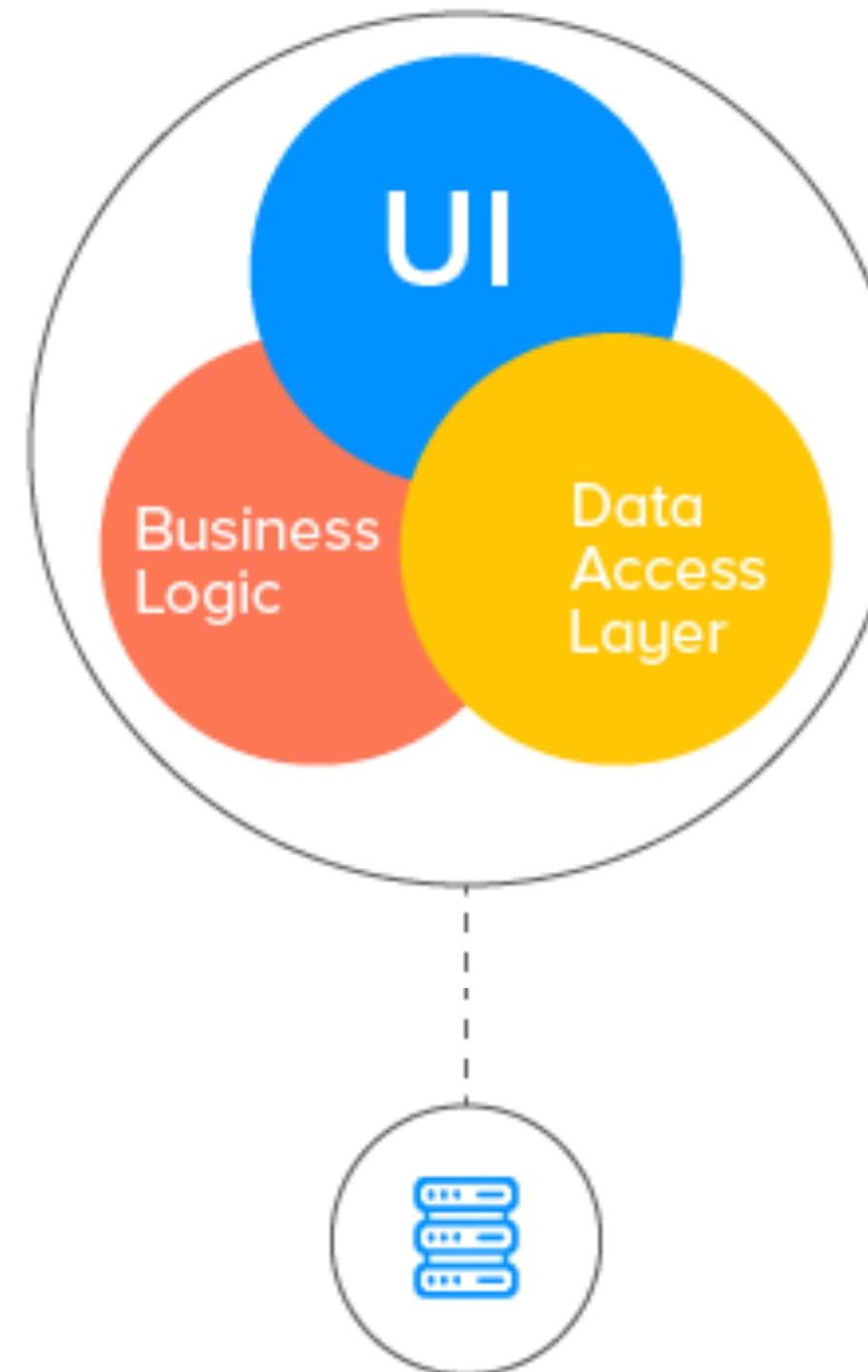
Microservices vs. Monolith

При монолитной архитектуре все контексты приложения запускаются на сервере с помощью внутрипроцессных взаимодействий.

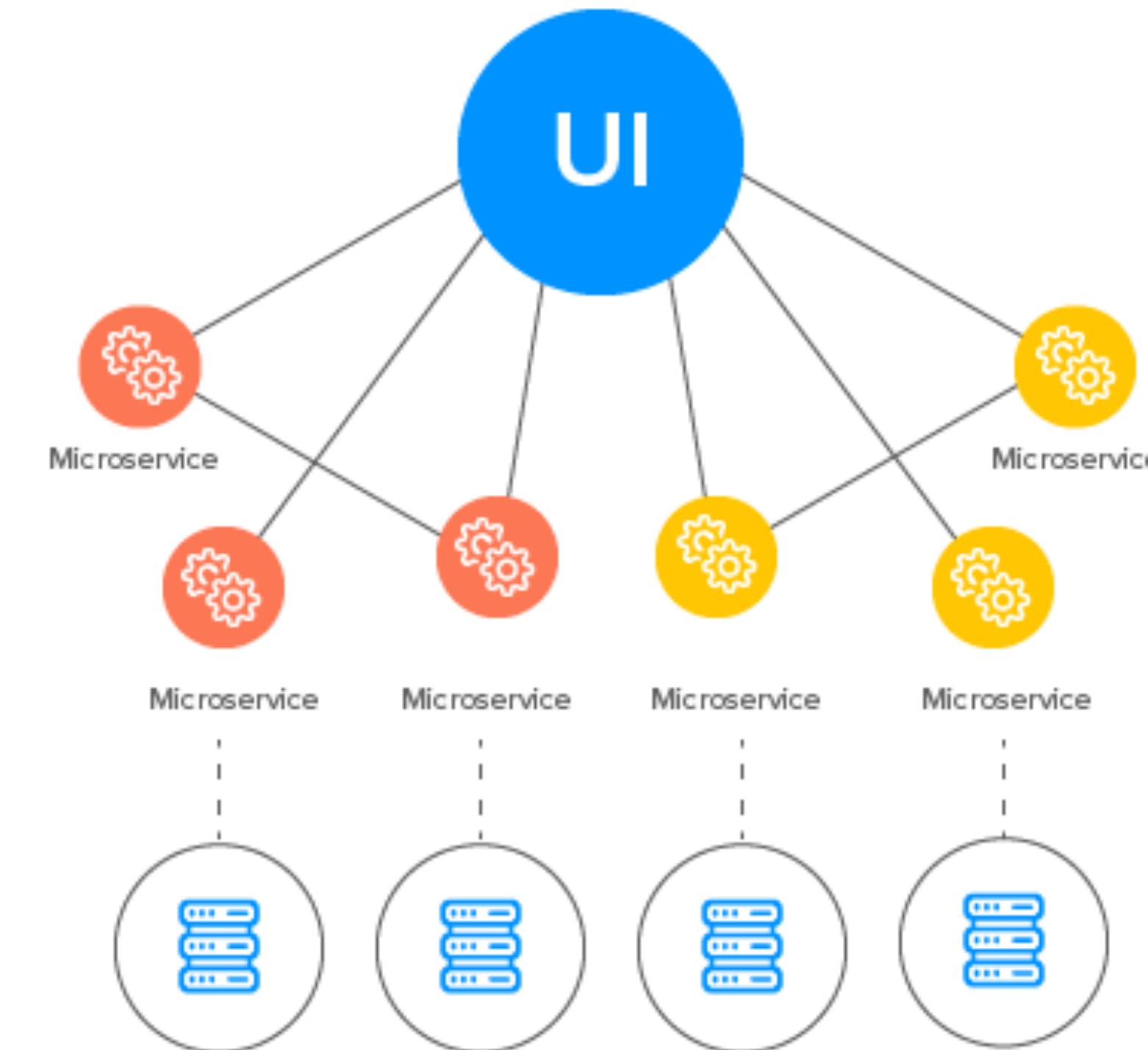


Microservices vs. Monolith

При монолитной архитектуре все контексты приложения запускаются на сервере с помощью внутрипроцессных взаимодействий.



Микросервисная архитектура — это подход, позволяющий инкапсулировать определённые контексты приложения.



Microservices



Monolith

Модульность и независимость
реализаций

Microservices



Monolith

Упрощенная разработка и развертывание
Меньше сквозных проблем
Лучшая производительность

Модульность и независимость
реализаций

Microservices

Сложность разработки из-за наличия различных технологий
Накладные расходы на сериализацию и десериализацию сообщений
Время работы напрямую зависит от того, как сервисы общаются между собой



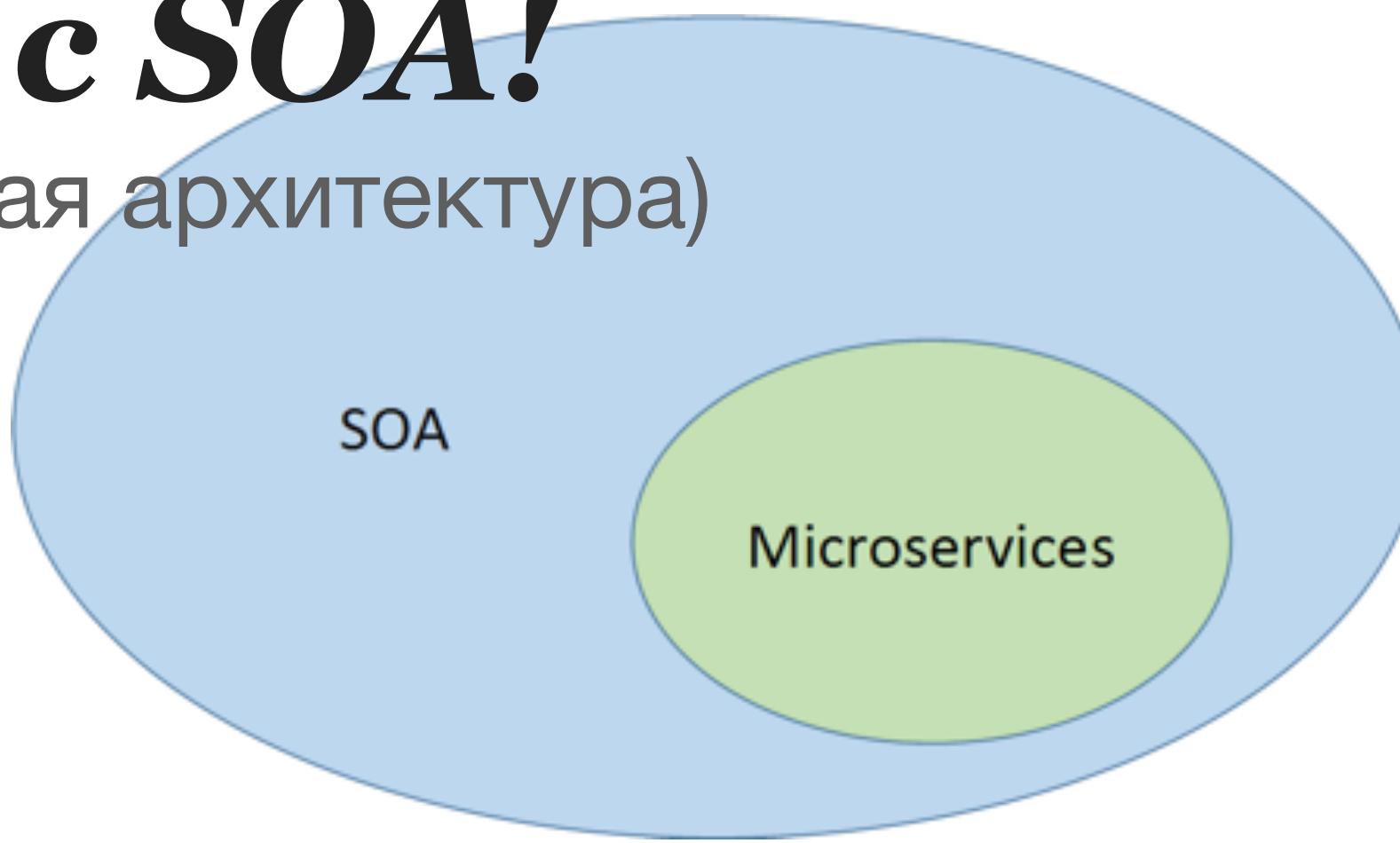
Monolith

Кодовая база со временем становится громоздкой
Сложно внедрять новые технологии
Ограниченнная гибкость

Упрощенная разработка и развертывание
Меньше сквозных проблем
Лучшая производительность

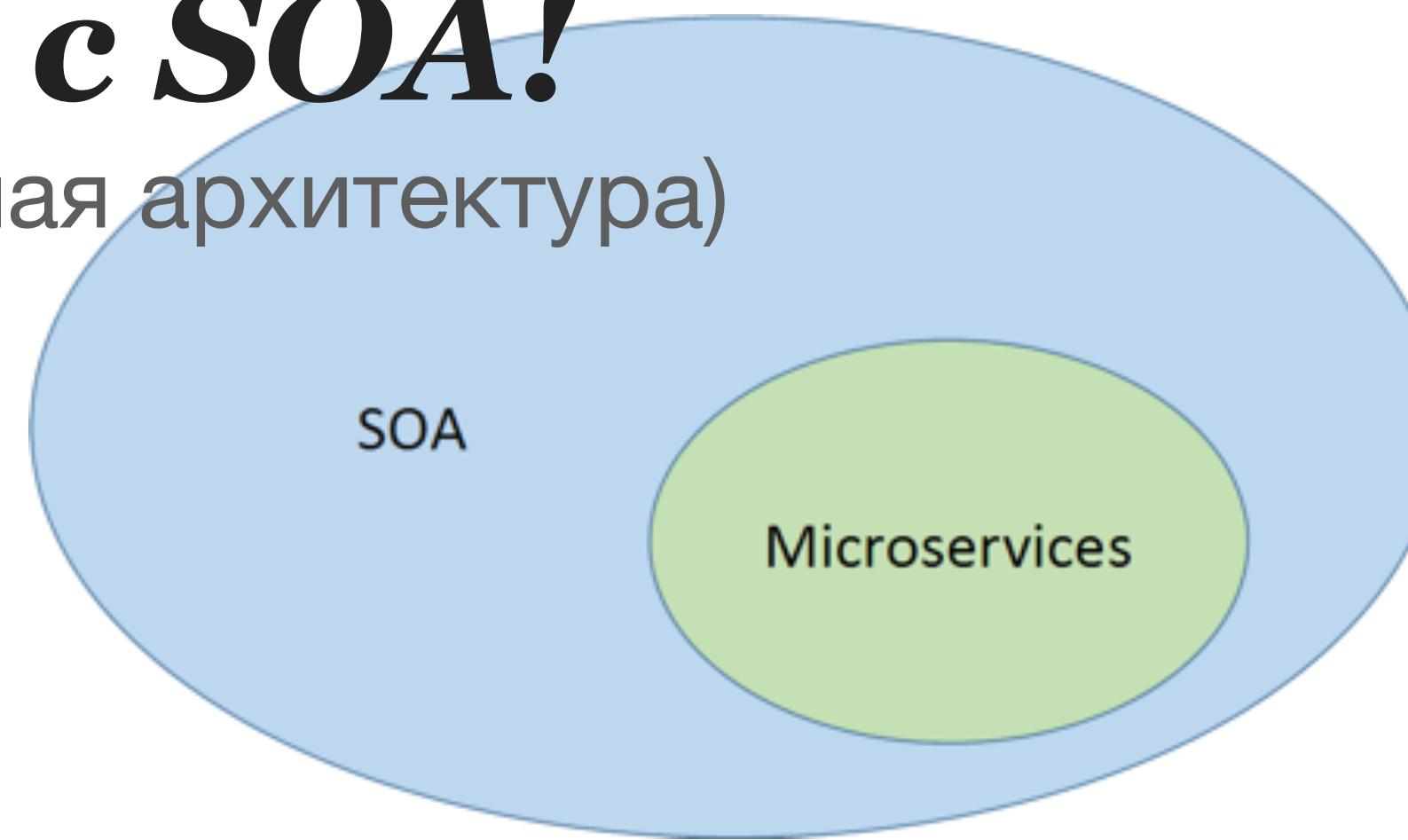
Не путать с SOA!

(сервисно-ориентированная архитектура)



Не путать с SOA!

(сервисно-ориентированная архитектура)



Еще почитать:

<https://habr.com/ru/post/249183/>

<https://habr.com/ru/company/dataart/blog/280083/>

<https://habr.com/en/post/249183/>

Как обеспечить работоспособность, если:

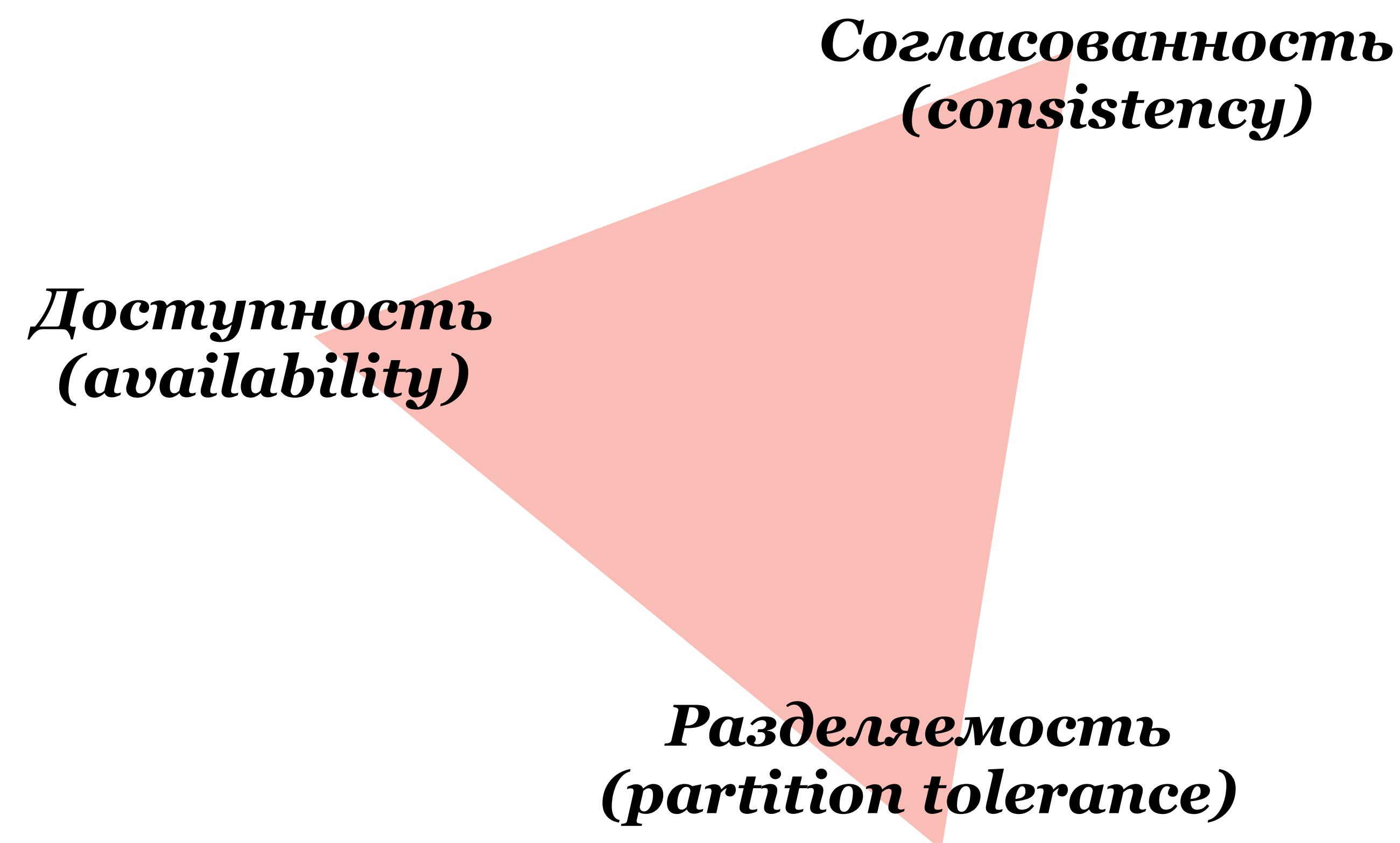
1. серверы в разных регионах
2. нужен стабильный доступ
3. данные должны быть согласованными?

CAP-теорема

© Брюер, 2000 г

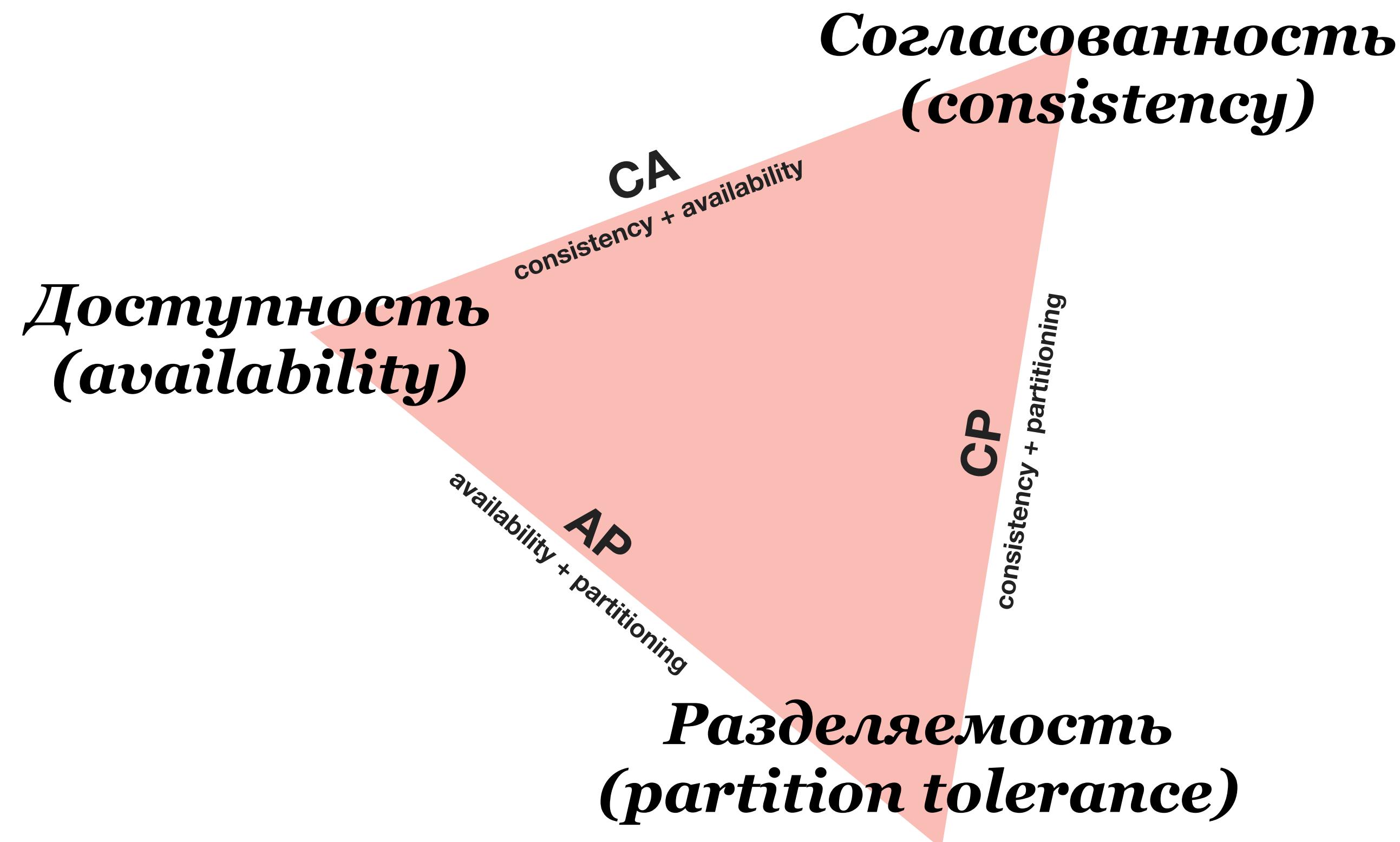
CAP-теорема

© Брюер, 2000 г



CAP-теорема

© Брюер, 2000 г



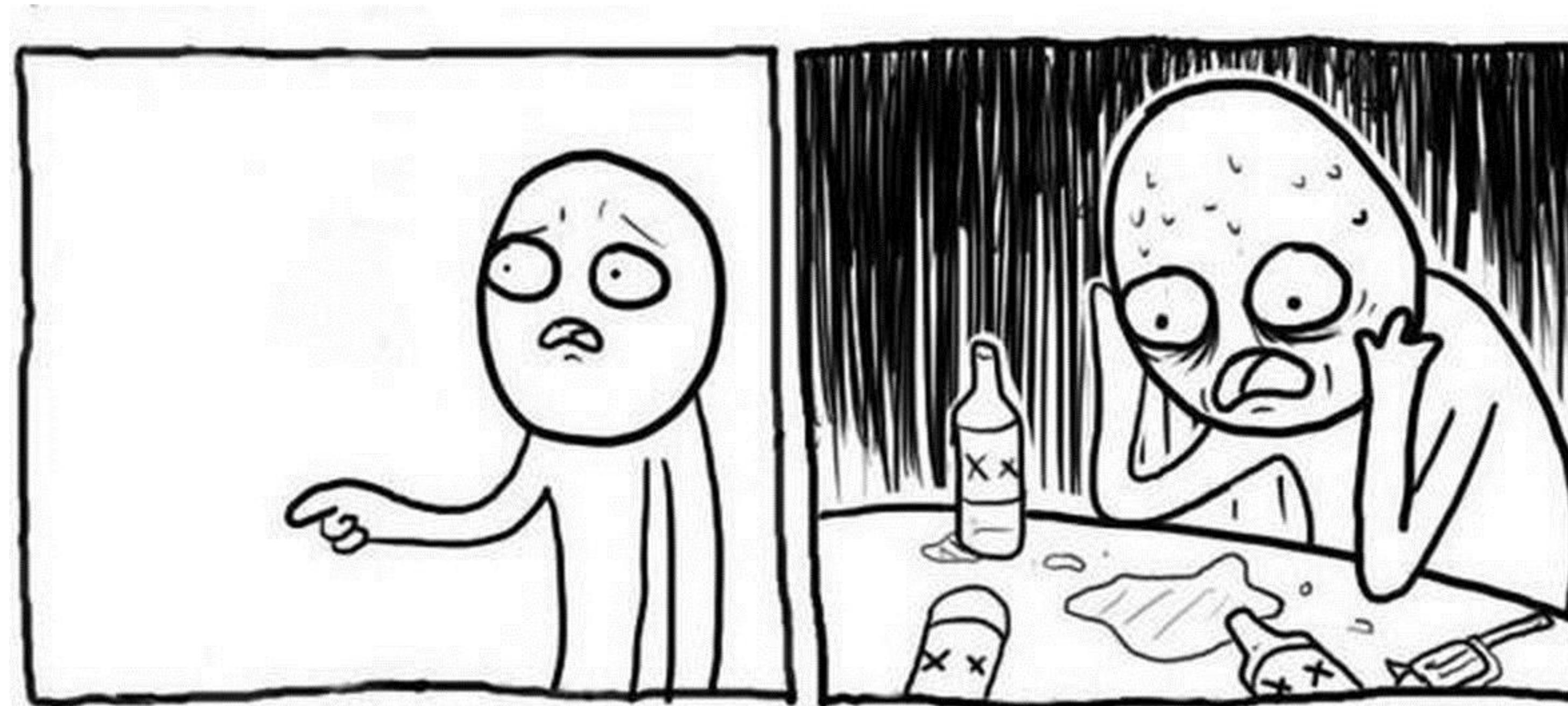
Как обеспечить работоспособность, если:

1. серверы в разных регионах
2. нужен стабильный доступ
3. данные должны быть согласованными?

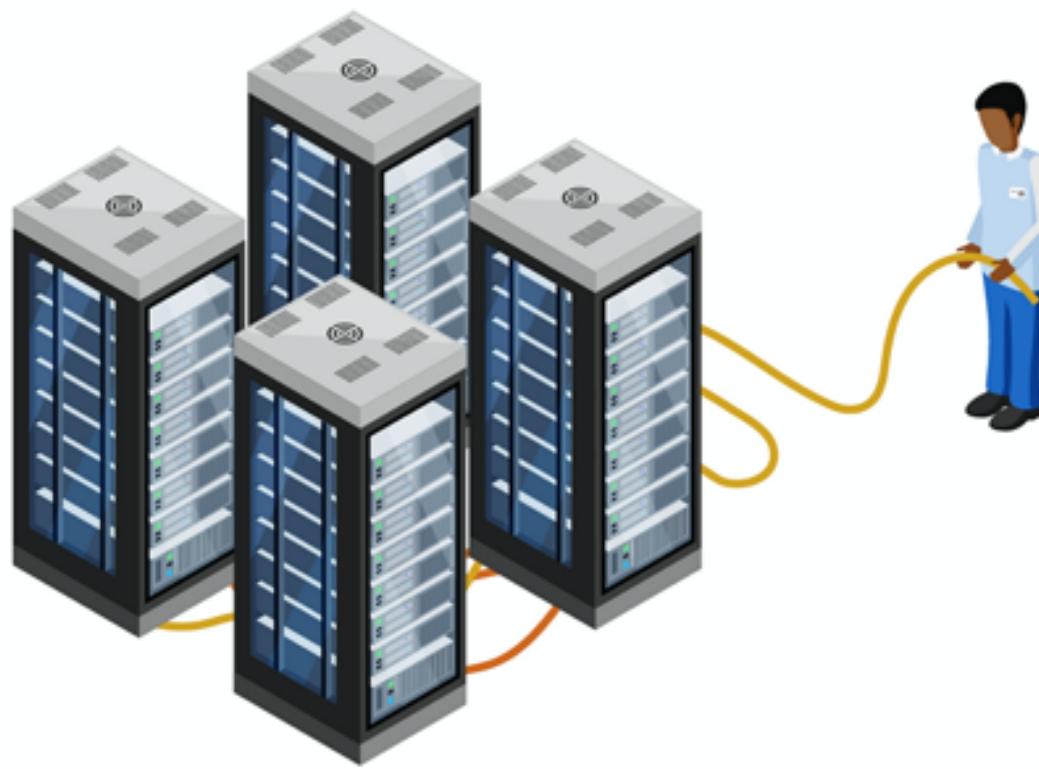
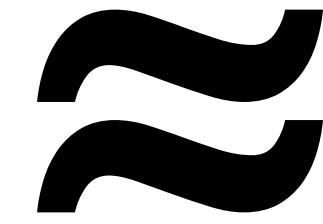
Как обеспечить работоспособность, если:

1. серверы в разных регионах
2. нужен стабильный доступ
3. данные должны быть согласованными?

Никак!

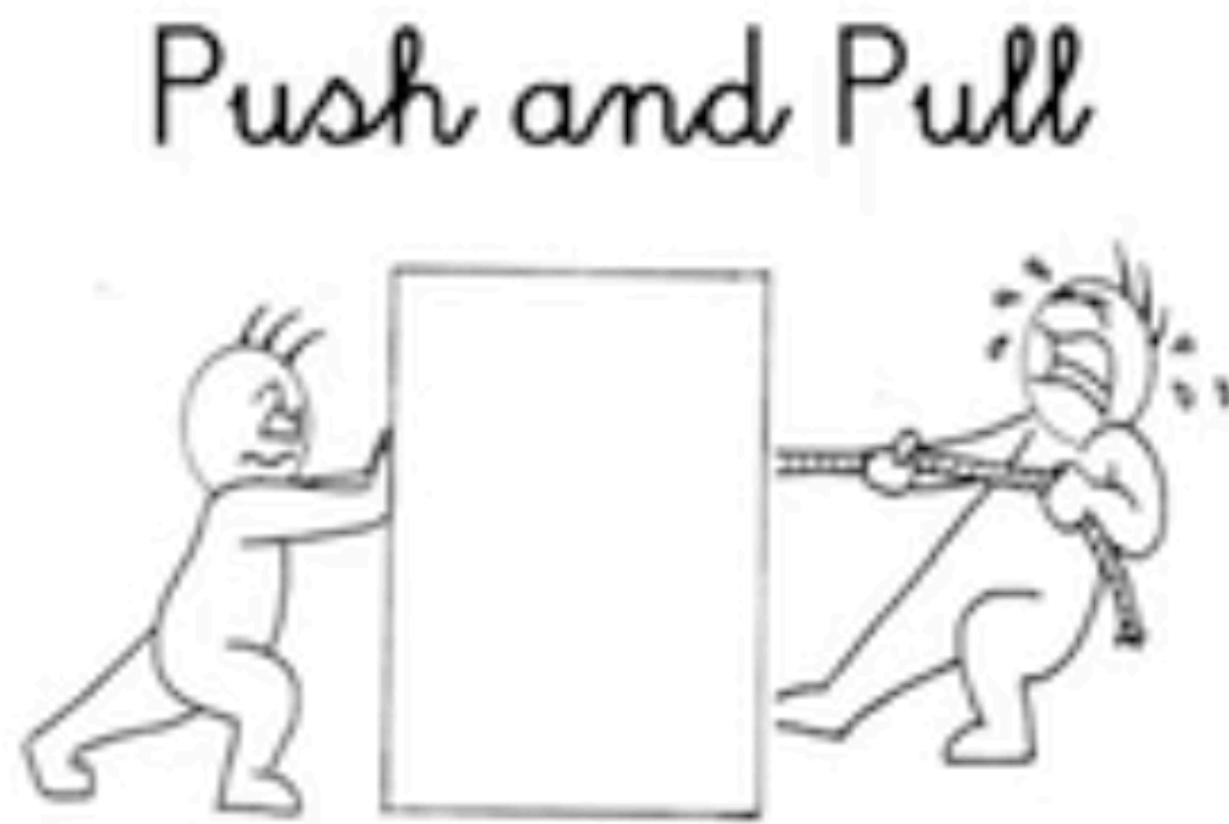


Как обеспечить согласованность конфигураций в локальной среде и на удаленных серверах?



Виртуализация и изолированность

Как настроить удаленную машину?



Раньше: создавали **большой конфигурационный файл**, в котором описываем все настройки, зависимости и библиотеки, и на его основе настраиваем все удаленные машины.

Сейчас:

- [Ansible](#). Он позволяет через SSH-соединение «проталкивать» (push) файлы конфигурации на множество машин, и таким образом обеспечивать единообразие
- [Chef](#) «тянет» (pull) конфигурацию с главной машины

Можно ли создать виртуальную,
изолированную от основной системы
среду для каждого проекта. Тогда
каждый из них будет иметь свои
независимые настройки, в том числе
разные библиотеки и их версии?

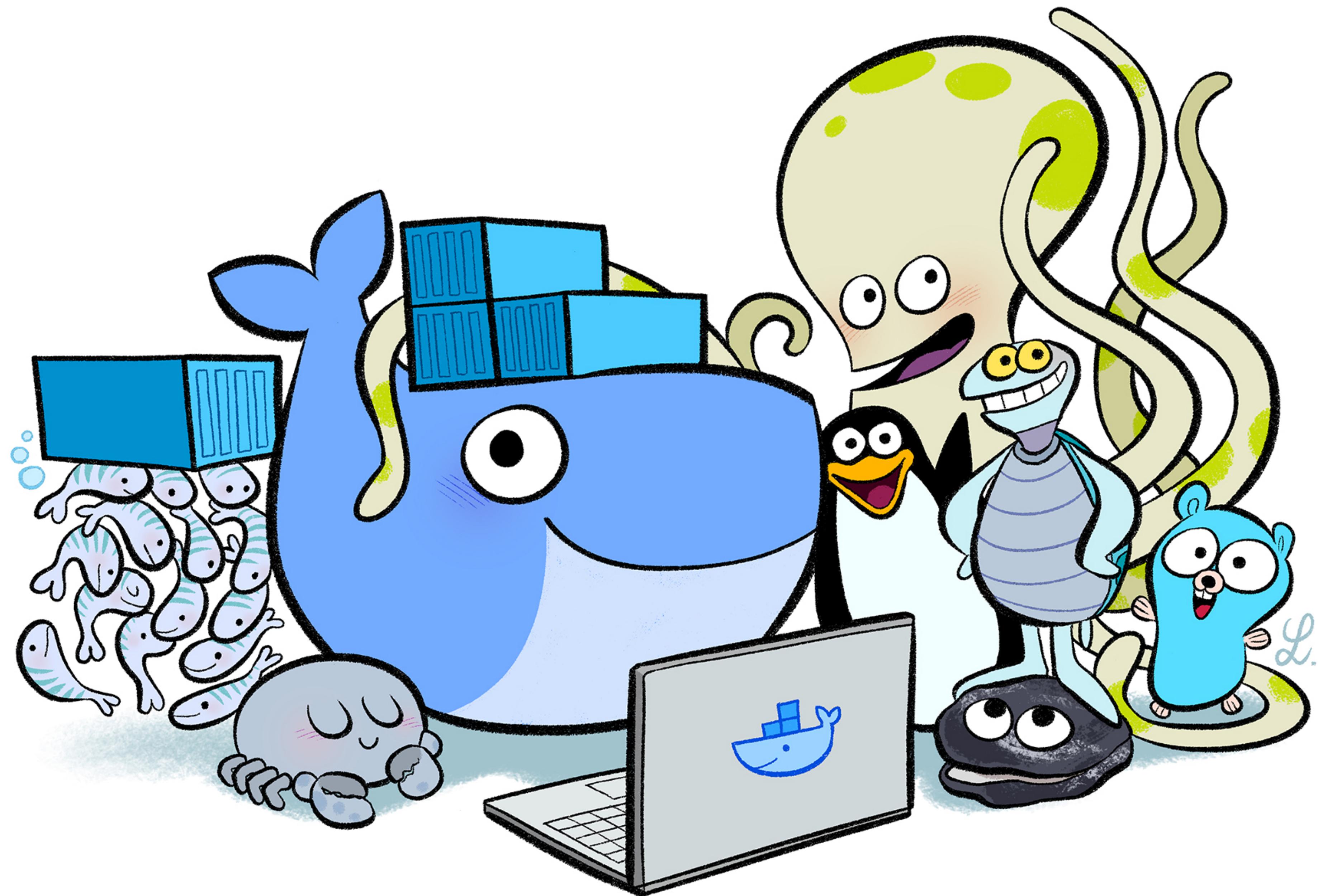


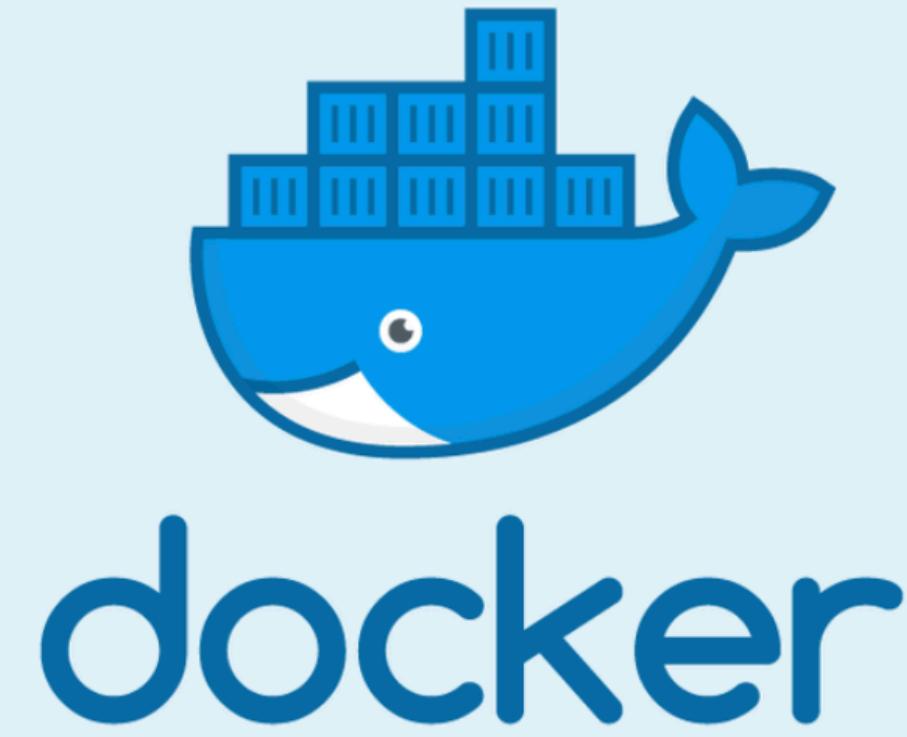
Можно ли создать виртуальную, изолированную от основной системы среду для каждого проекта. Тогда каждый из них будет иметь свои **независимые настройки**, в том числе разные библиотеки и их версии?



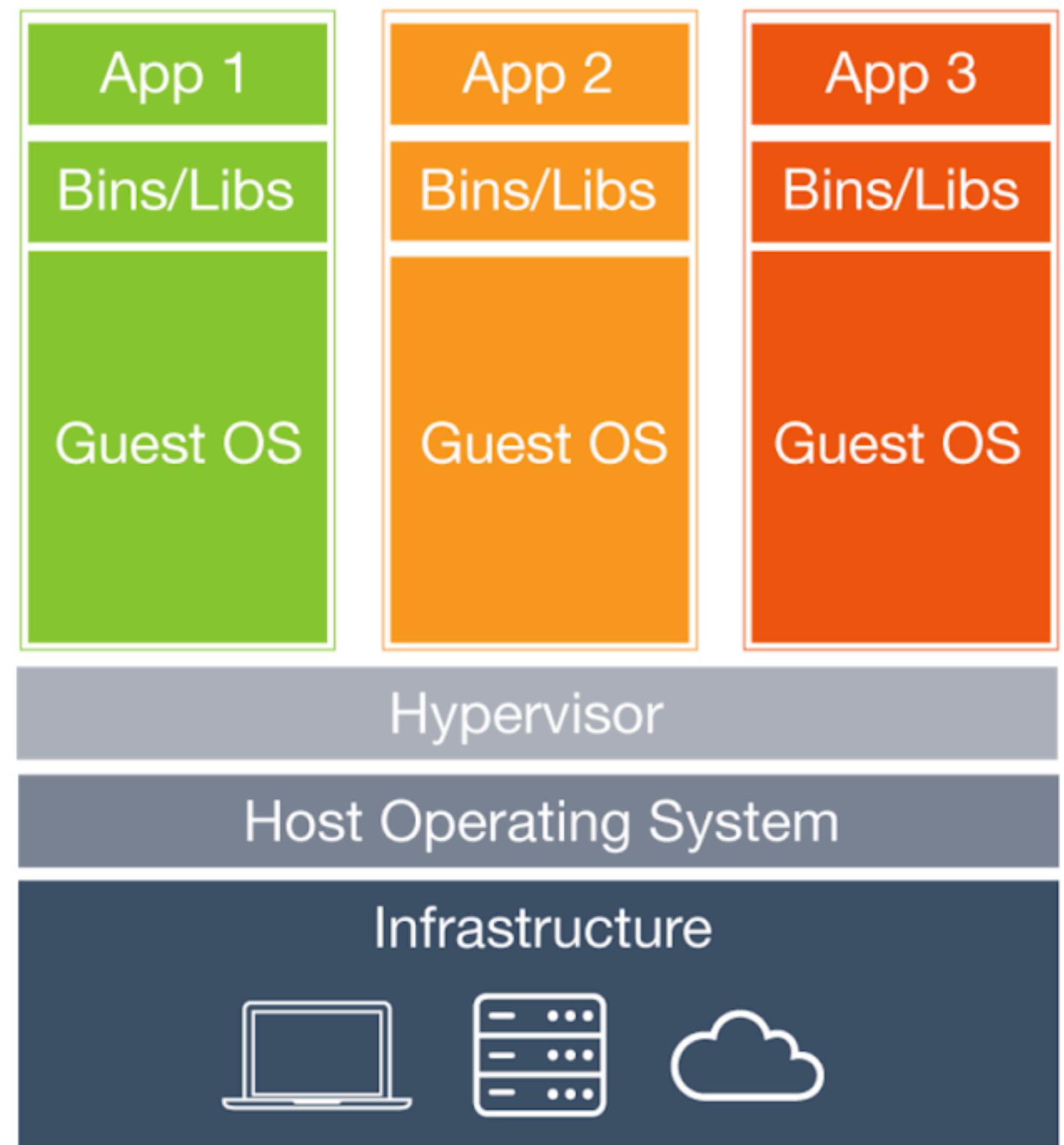
Virtualenv

Как живет реальный мир?

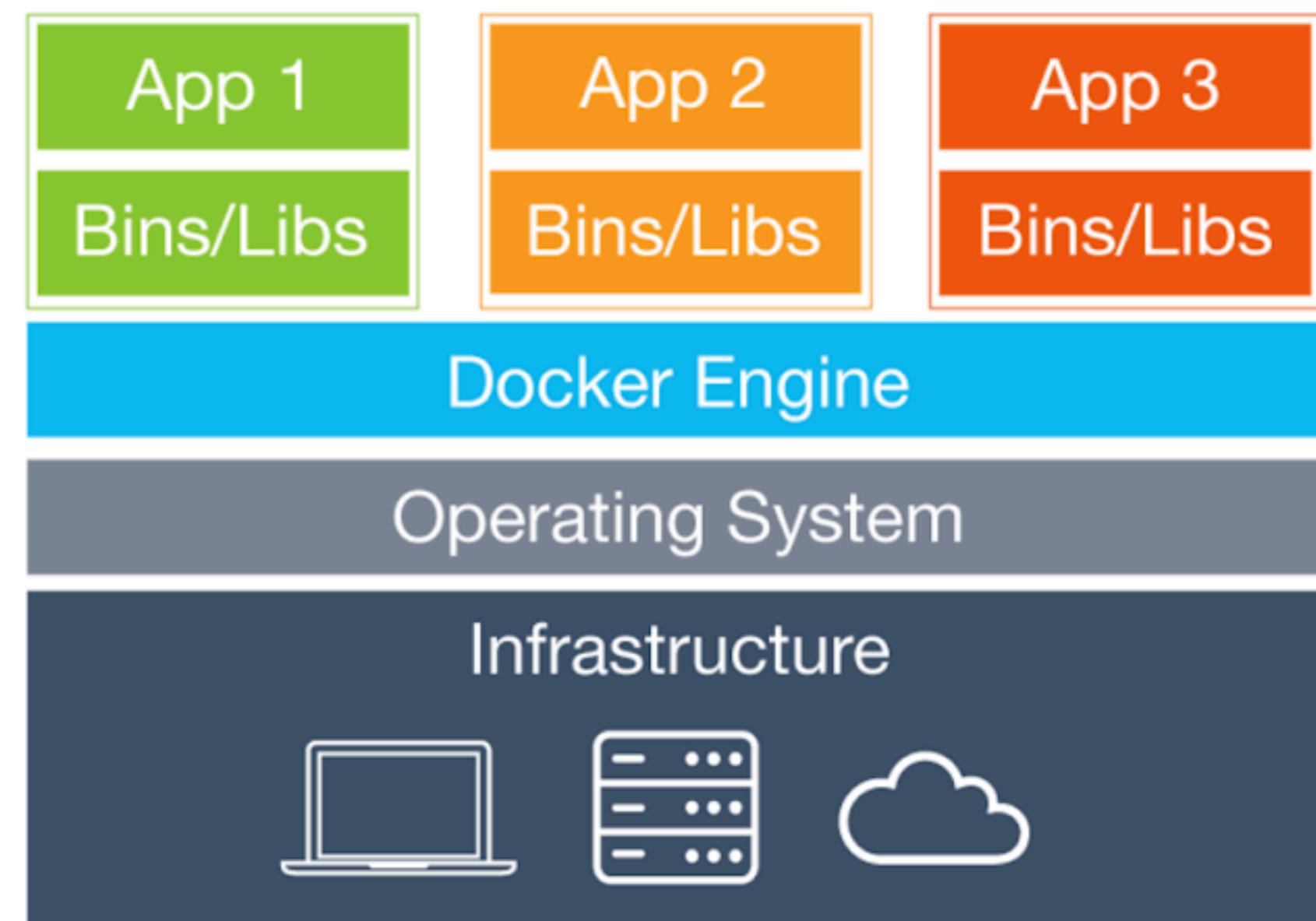




Docker - программное обеспечение, которое позволяет в автоматическом режиме создавать и управлять виртуальными машинами. При этом он делает это более элегантно по сравнению с обычными средствами виртуализации, например *VirtualBox*. Элегантно — потому что виртуализация происходит на уровне операционной системы.



Virtual Machines



Containers

Что будем делать?

Решать кейс!

1. Подключиться к внешнему брокеру сообщений mqtt с потоком данных в режиме реального времени.
2. Реализовать программу для чтения входящих сообщений с данными о двигателях и их сохранение в хранилище.
3. Реализовать хранилище для данных, учитывая их структуру. Предусмотреть, что хранилище должно быть горизонтально масштабируемым. Выбор хранилища требуется обосновать.
4. Проанализировать данные, найти зависимости и закономерности. Сгенерировать новые признаки на основе существующих, расчитывать их в режиме реального времени и сохранять в хранилище.
5. Реализовать алгоритм (например, машинного обучения) для решения задачи классификации (будет ли выведен в ремонт двигатель через n циклов) и/или регрессии (предсказать сколько циклов осталось до вывода в ремонт после прохождения m циклов). Оценить качество работы алгоритма(ов) на исторических данных.
6. Реализовать применение алгоритма из п.5 на потоке данных в режиме реального времени. Собирать оценку качества работы алгоритма(ов) на потоковых данных.
7. Построить систему для визуализации в реальном времени: Данных, прогнозов, метрик и других необходимых параметров

Инструменты?

Инструменты?

