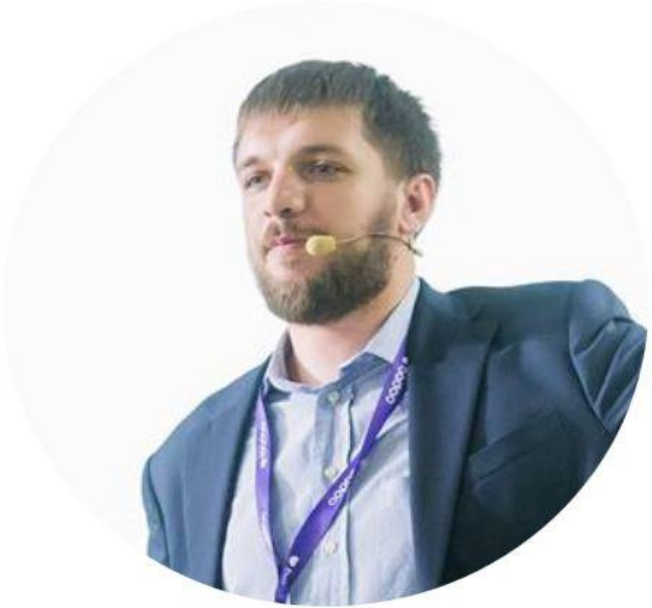
НЕТОЛОГИЯ
групп

Культура работы с данными

Лекция 8



Алексей Кузьмин

Директор разработки; Data Scientist

ДомКлик.ру

aleksej.kyzmin@gmail.com

О ЧЕМ ПОГОВОРИМ И ЧТО СДЕЛАЕМ

Работа с данными

Источники данных

Сбор данных
Лекция 8

SQL-БД
Лекция 1

NoSQL
Лекция 4

MapReduce
Лекция 5-7

Мотивация Big Data
Лекция 3

Data Science
Лекция 2

Отчеты
Лекция 1

Модели
Лекция 2

Люди и процессы
Лекция 9

Примеры кейсов
Лекция 10

ОБСУДИМ

1	2	3	4	5
Что такое большие данные?	Как выбрать где хранить данные?	Как правильно собирать данные?	Data-driven мышление	Как разобраться с бардаком в данных?

1. Что такое большие данные?

Что такое большие данные?

БОЛЬШИЕ ДАННЫЕ (BIG DATA) —

это совокупность методов работы с огромными объёмами структурированной или неструктурированной информации.

Специалисты по работе с большими данными занимаются её обработкой и анализом для получения наглядных, воспринимаемых человеком результатов

Почему данные БОЛЬШИЕ и что с ними не так?

Сколько нужно данных, чтобы они стали большими?

Неформальное определение:

Данные становятся БОЛЬШИМИ, когда их невозможно хранить или обрабатывать на одном сервере.

Примеры

1. Ограничения Excel 1 048 576 строк и 16 384 столбца (на 2018 год)

2. Ограничения MySQL

Максимальный размер таблиц в MySQL 4 гигабайт (2004 год), 256 терабайт (2016 год).

Но на практике ограничения жестче.

2. Как хранить данные?

Экономика больших данных

В обычной экономике: чем больше храним (например, на складе) - тем дешевле стоимость единицы хранения.

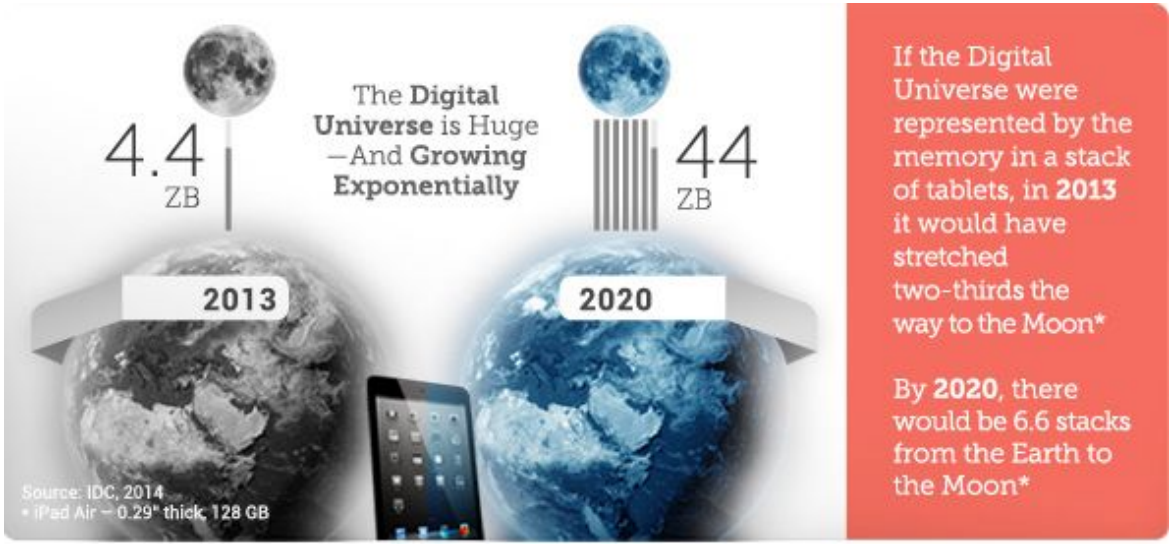
С большими данными все иначе!

Хранить (и использовать!) 1 файл размером 100Тб дороже, чем 100 по 1Тб.

Потому что нужны принципиальные другие технологии.

BigData vs SmartData

“К 2020 году около 1,7 мегабайта новой информации будет создаваться каждую секунду для каждого человека на планете.”



https://www.forbes.com/sites/bernardmarr/2015/09/30/big-data-20-mind-boggling-facts-everyone-must-read/#3543872617b1

Некоторые данные невыгодно хранить

MySpace “потеряла” данные пользователей за 12 лет – 18 марта 2019.

> Бывший технический директор Kickstarter Энди Байо:
>
> “Сильно сомневаюсь, что это было случайно”

https://twitter.com/waxpancake/status/1107511026931490817

Как правильно выбрать хранилище (базу) данных?

Нет единого способа сделать “правильно”.

Всегда нужно смотреть что вы храните и обрабатываете:

- документы
- пользователей
- социальный граф
- банковские транзакции

CAP

Consistency (согласованность данных) — во всех вычислительных узлах в один момент времени данные не противоречат друг другу;

Availability (доступность) — любой запрос к распределённой системе завершается корректным откликом, однако без гарантии, что ответы всех узлов системы совпадают;

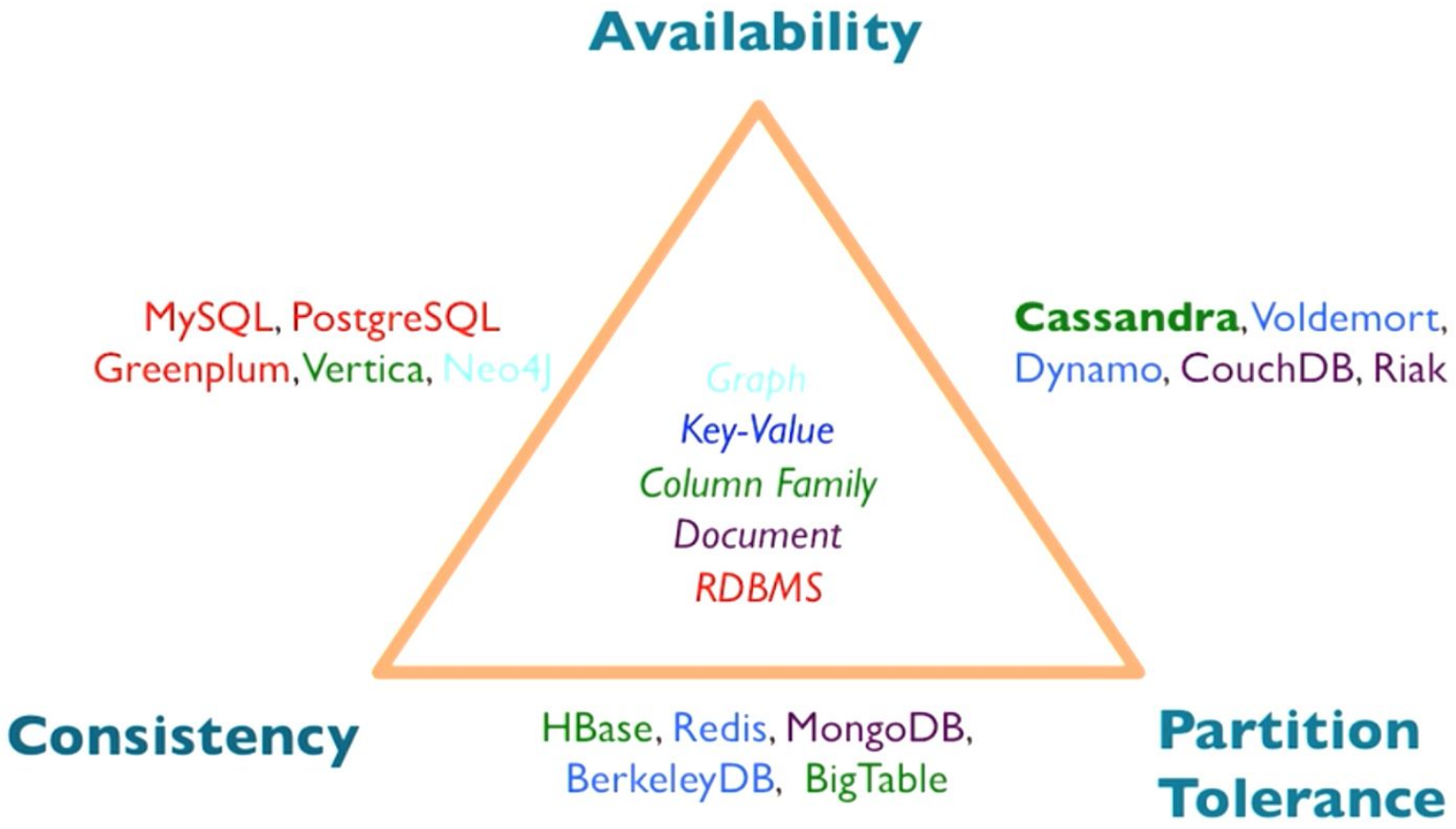
Partition tolerance (устойчивость к разделению) — расщепление распределённой системы на несколько изолированных секций не приводит к некорректности отклика от каждой из секций.

CAP-теорема

CAP Теорема (теорема Брюера) — утверждение о том, что в любой реализации распределённых вычислений возможно обеспечить не более двух из трёх следующих свойств: **C**onsistency, **A**vailability, **P**artition tolerance.

*Строго говоря: это **гипотеза**, т.к. Теорему никто не доказал, но и не опроверг.*

См. также: Теорема PACELC


Availability
MySQL, PostgreSQL
Greenplum, Vertica, Neo4J
Cassandra, Voldemort,
Dynamo, CouchDB, Riak
Graph
Key-Value
Column Family
Document
RDBMS
Consistency
HBase, Redis, MongoDB,
BerkeleyDB, BigTable
Partition
Tolerance

Нужно задать 3 вопроса при выборе

1. Скорость ответа?
    a. Real time
    b. Near-Real time
    c. Long-time
2. Критичность ошибки в данных?
    a. Ошибки недопустимы
    b. Допускаются небольшие ошибки
    c. Допускаются ошибки, но нужна “Согласованность в конечном счёте”
3. Критичность потери в данных?
    a. Near-Real-time копирование
    b. Отложенный backup

На практике обычно используют

Для продакшена:

AC - если возможно

AP - иначе

Для логов, аналитики и тд:

CA - если возможно

CP - когда, данных много

3. Как правильно собирать данные?

Перед тем как собирать данные, нужно посчитать окупаемость:

- сбора
- хранения
- эксплуатации

Если вы Яндекс, Google, Mail.ru или Facebook - собирайте все.

Простые метрики оценки эффективности данных

Затраты	Выручка
1. Необходимые данные a. Инфраструктура b. Программное обеспечение c. Специалисты поддержки	Baseline
2. Данные для real-time a. Инфраструктура b. Программное обеспечение c. Разработка алгоритмов	рост от Baseline
3. Данные для долговременного хранения a. Инфраструктура b. Программное обеспечение c. Разработка алгоритмов	рост от Baseline

Как считать (оценивать) прибыль от данных?

1. Смотреть успешные кейсы (и делить на 2)
2. АБ-тесты
3. Рыночная стоимость данных (DMP)

DMP (Data Management Platform) - *платформа управления данными*.

Общие правила сбора данных

1. У данных должен быть потребитель
2. У данных должен быть единый источник (мастер-дата)
3. Атомарность
4. Гибкая структура
5. Версионность
6. Валидация

И конечно: документация! :)

Общие правила обработки данных

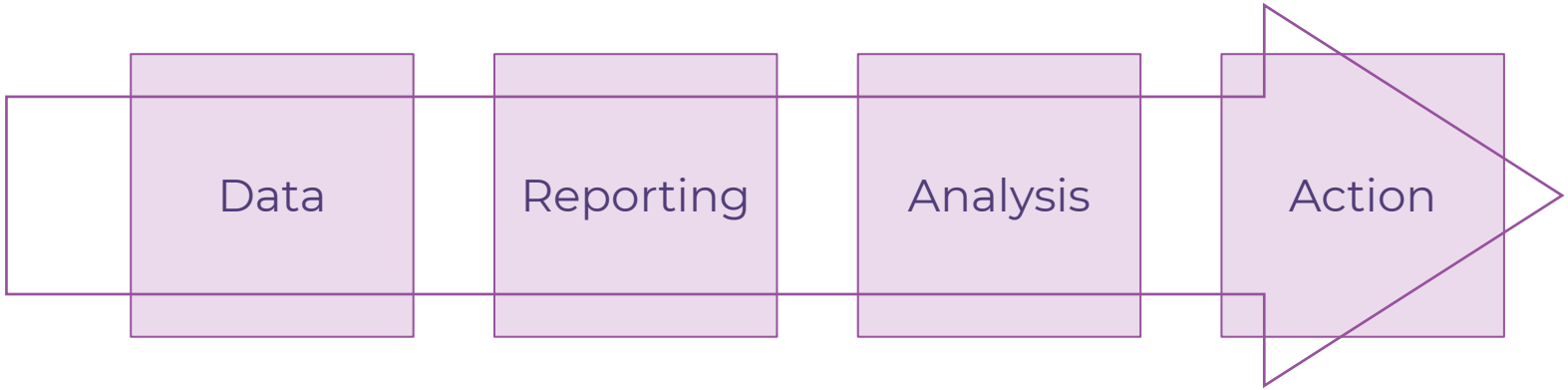
1. Для каждого потребителя свой ETL*
2. Версионность ETL
3. Вся обработка данных в “слоях” (промежуточные таблицы, БД)
4. Обработку данных делать на там же где и хранить
5. Для частых запросов создавать агрегаты

И конечно: документация! :)

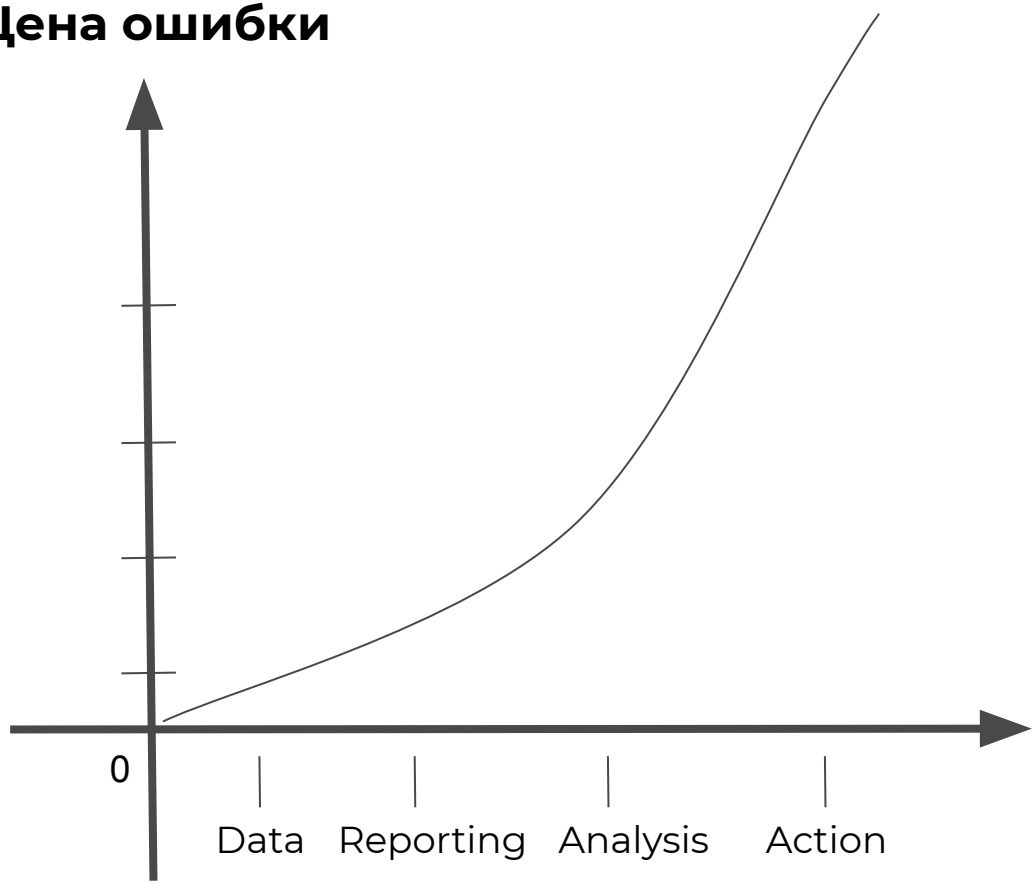
**ETL (от англ. Extract, Transform, Load — дословно «извлечение, преобразование, загрузка»)*

4. Data-driven мышление

Как было бы хорошо и «правильно»



Цена ошибки




1/2
0
Diesel
1
80
60
100
40
120
20
140
160
180
200
220
240
260
km/h
mph
1/min x 1000
READY
OFF
P
EfficientDynamics
mpg
+16.0 °C
50
100
150
05510
000.0
15:29

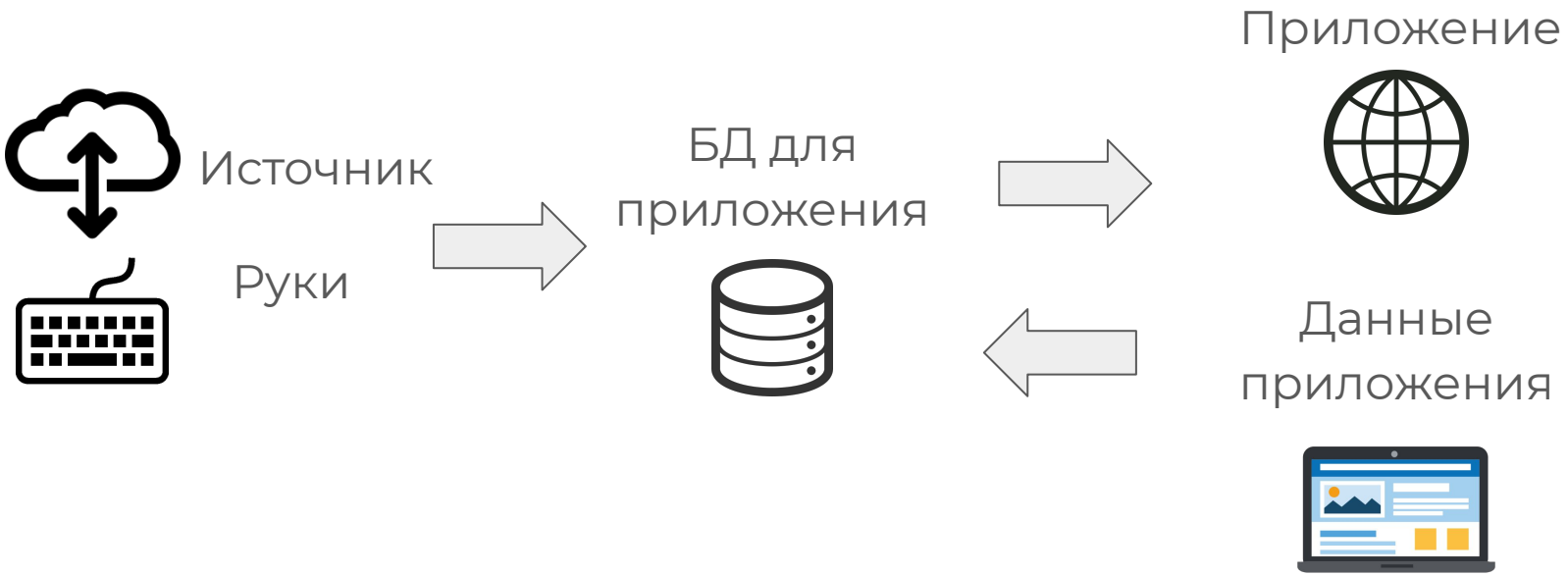
Все данные плохие… всегда!

- Ошибки в структурировании
- Ошибки в база данных
- Ошибки в программной логике
- «Кто-то что-то поменял»
- Изменение сущностей

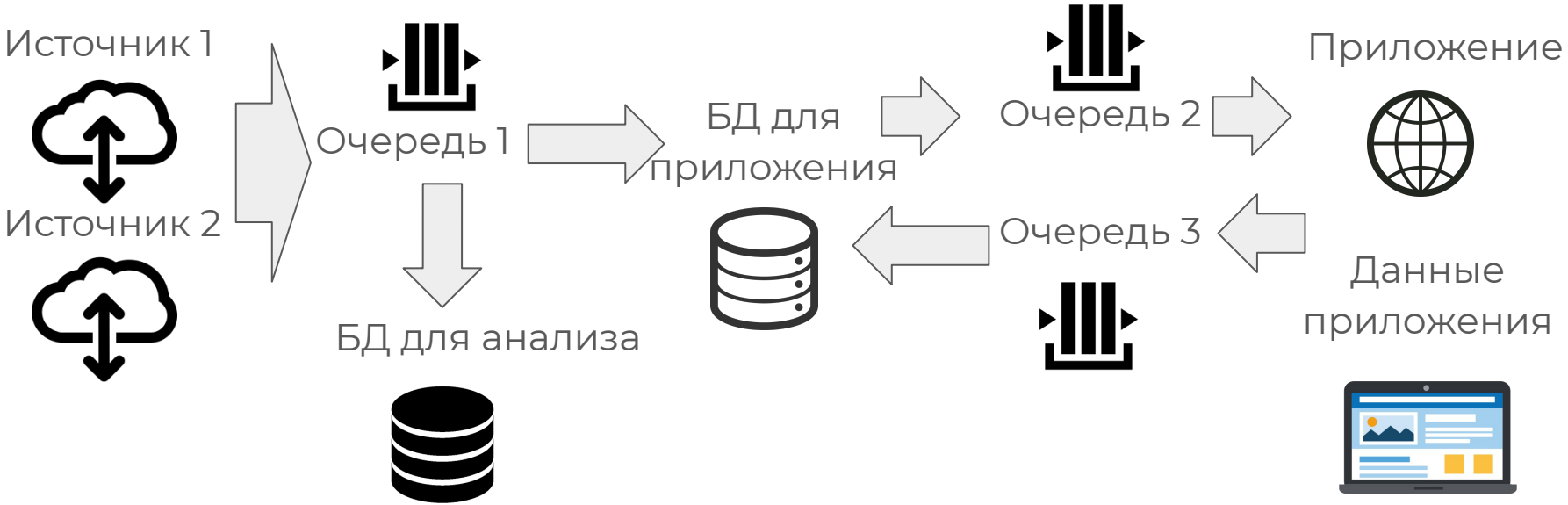
5. Как разобраться с бардаком в данных?

Усложнение архитектуры

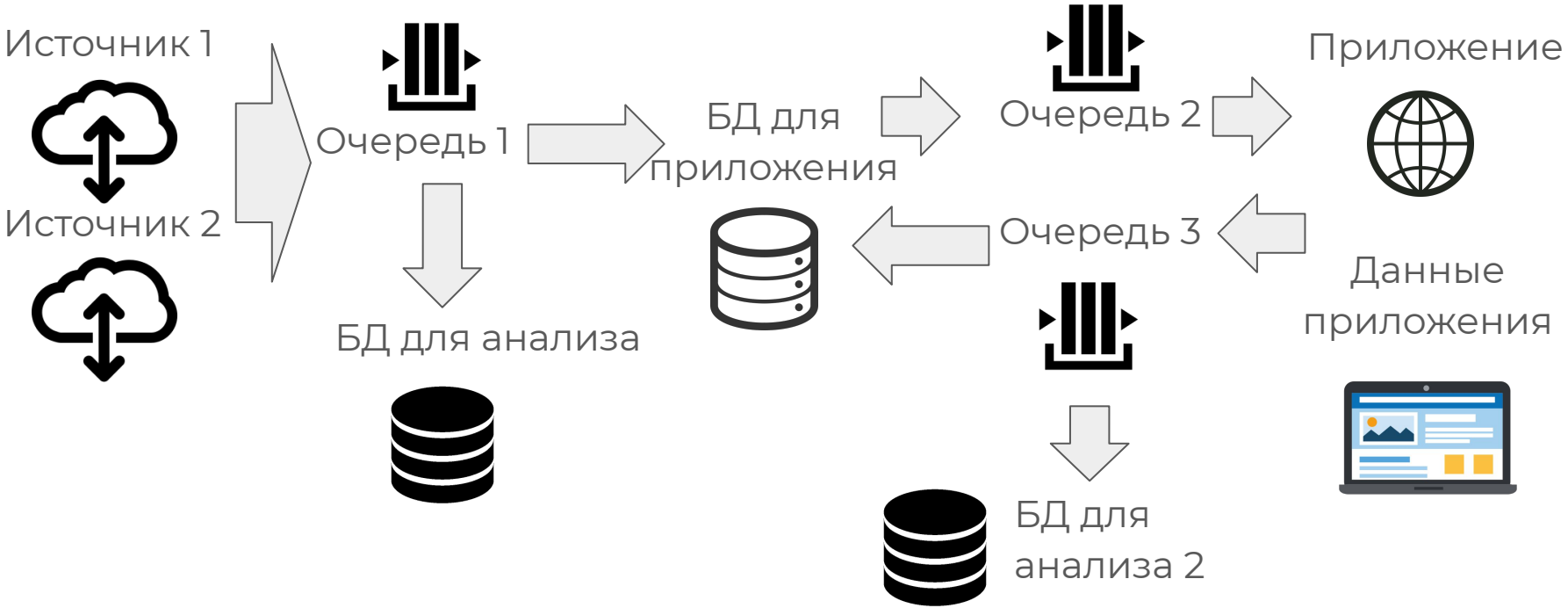
Усложнение архитектуры



Усложнение архитектуры



Усложнение архитектуры



Как в реальной жизни

1. Сотни csv-шных файлов, лежащих на ftp
2. Кто-то генерирует данные “руками”
3. Нет единой мастер-даты
4. Обмены между системам идут очень долго
5. …

Алгоритм расчистки бардака

1. Запрещаем напрямую (“руками”) писать в БД
2. Определить источники для мастер-данных
3. Определить конечных потребителей данных и форматы
4. Настроить очереди от всех источников данных
5. Подключить потребителей данных к очередям
6. ETL на стороне потребителей

После того как все ОК: удаляем старые связи

Плюсы и минусы

Плюсы такого подхода:

Гарантированно приводит к счастью :)

Безопасно

Минусы подхода:

Много накладных расходов

Относительно долго

Заключение

1. Храните и собирайте данные правильно!
2. Культура сбора данных экономит время, деньги и увеличивает счастье!
3. Используйте алгоритмы расчистки бардака.

Домашнее задание

Возьмите свою архитектуру двх. Предложите несколько (5-6) проверок, которые мы можем проводить при записи новых данных, чтобы убеждаться, что все хорошо

НЕТОЛОГИЯ
групп

Спасибо за внимание!

Алексей Кузьмин  aleksej.kyzmin@gmail.com