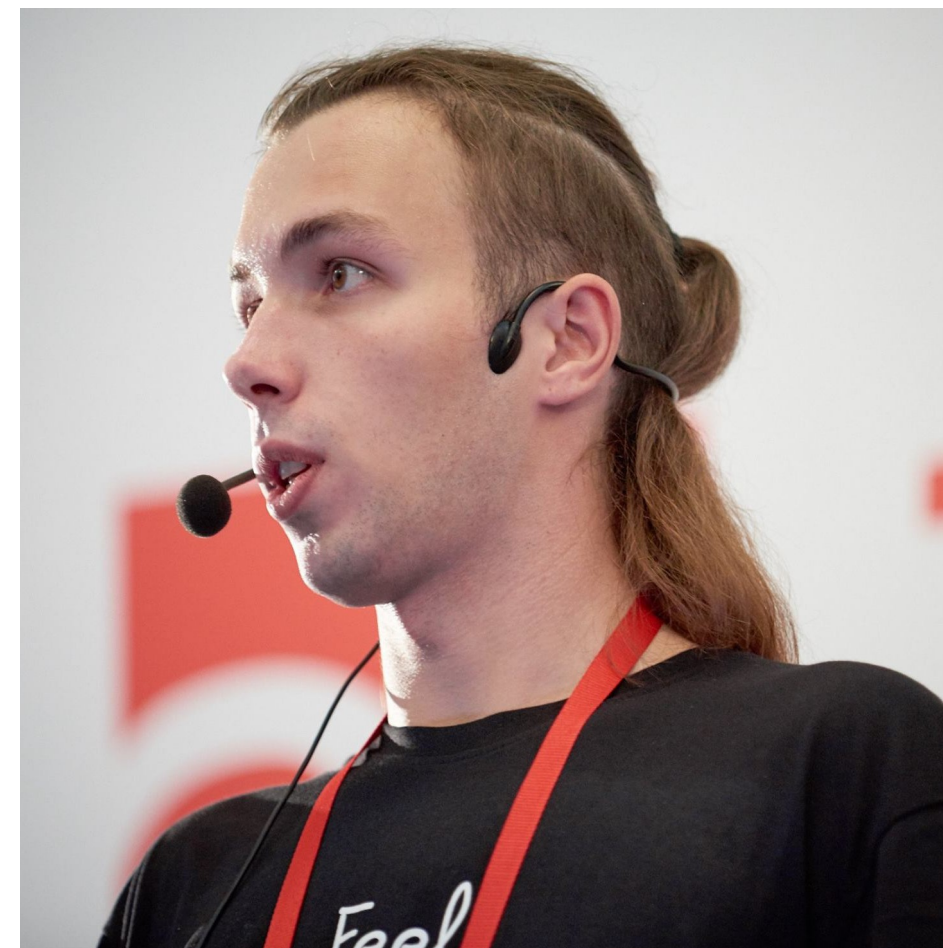


# Применение ClickHouse в Badoo

Александр Крашенинников · 16 мая





# Александр Крашенинников

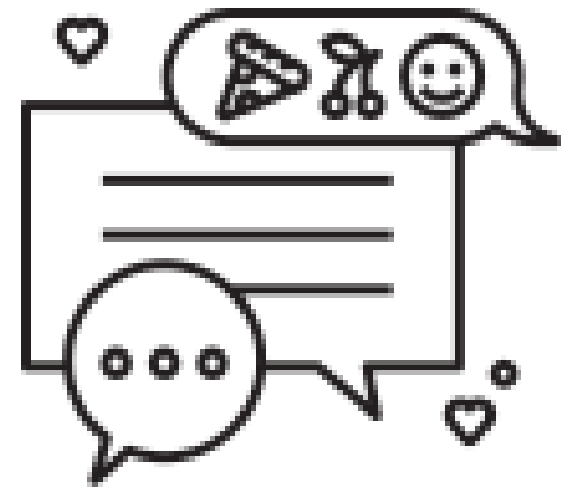
Head of Data Team, Business Intelligence

 [krash@corp.badoo.com](mailto:krash@corp.badoo.com)

 <https://linkedin.com/in/alexkrash/>

# About badoo

badoo\_tech



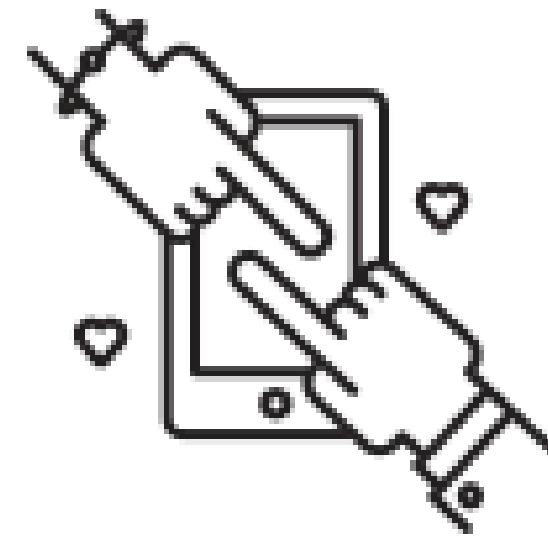
**350m**

Messages per day



**380m**

Registered users  
worldwide



**60m**

Monthly active  
users



**300k**

New daily sign ups

# О чём доклад

Постановка задачи

Архитектура системы

Циферки

**“Хочешь улучшить – измерь!”**

## Ін-һоузе система сбора статистики

Ближайшие аналоги – Яндекс.Метрика, AppMetrica, Google Analytics, Mixpanel, Amplitude

## **Ип-ноузе система сбора статистики**

Ближайшие аналоги – Яндекс.Метрика, AppMetrica, Google Analytics, Mixpanel, Amplitude

## **Realtime обработка данных**

Нет ежедневных/ежечасных задач ETL, Hadoop, Map/reduce

## **In-house система сбора статистики**

Ближайшие аналоги – Яндекс.Метрика, AppMetrica, Google Analytics, Mixpanel, Amplitude

## **Realtime обработка данных**

Нет ежедневных/ежечасных задач ETL, Hadoop, Map/reduce

## **Гетерогенный поток информации**

Однотипно обрабатываются события разного типа и различной атрибутивки



## **In-house система сбора статистики**

Ближайшие аналоги – Яндекс.Метрика, AppMetrica, Google Analytics, Mixpanel, Amplitude

## **Realtime обработка данных**

Нет ежедневных/ежечасных задач ETL, Hadoop, Map/reduce

## **Гетерогенный поток информации**

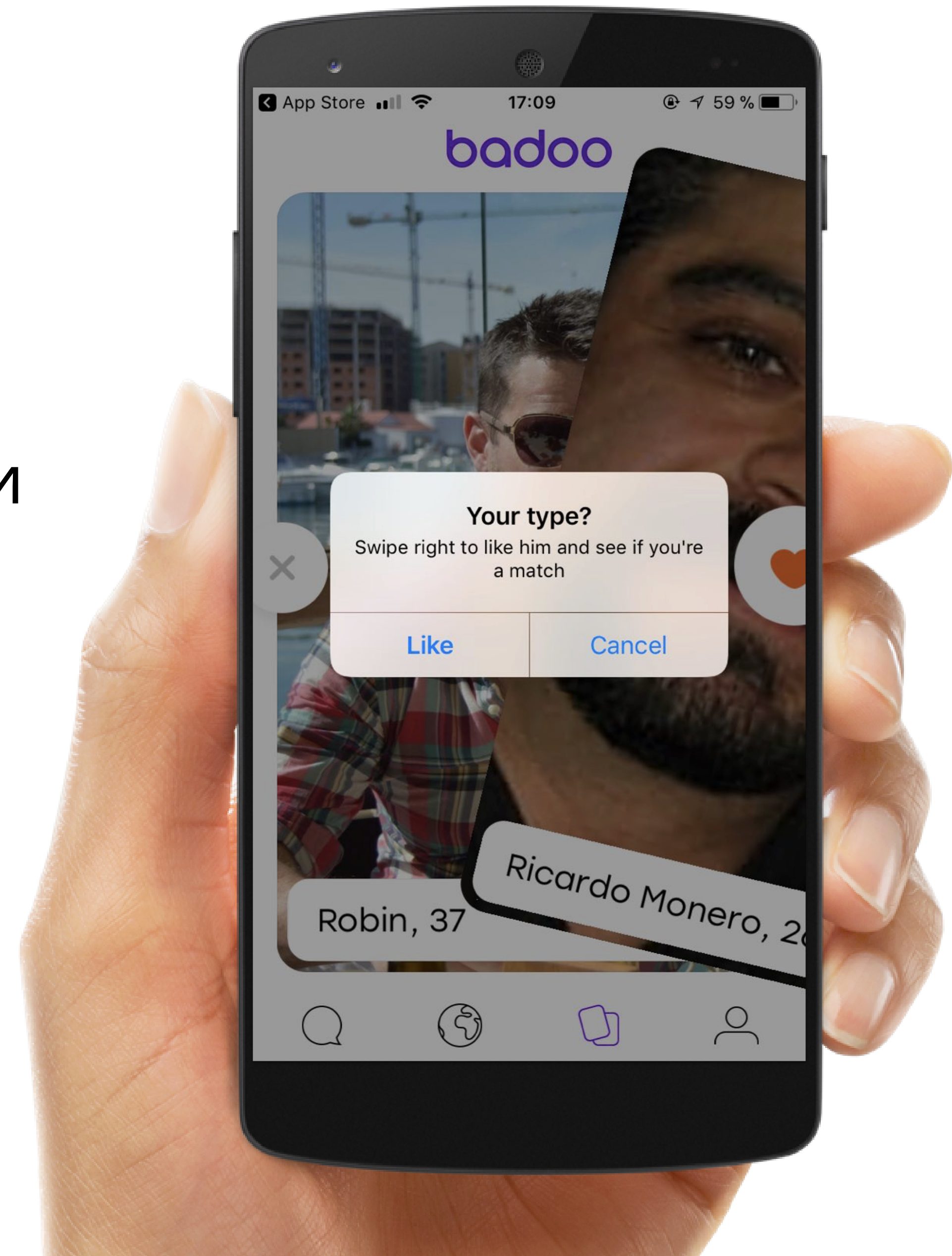
Однотипно обрабатываются события разного типа и различной атрибутивки

## **Много данных**

Цифры позже, но суть – на одном хосте не запроцессить

## Клиентские действия в приложении

- Просмотры экранов
- Взаимодействие с интерфейсом
- UI/UX метрики
- Бизнес-логика



## Серверные метрики

- Всё, что не Zabbix
- Бизнес-метрики (DAU/MAU, application-specific)
- Технические (очереди, performance, etc.)

# Особенности модели данных

## Гетерогенные эвенты

У каждого события – свой набор атрибутов

# Особенности модели данных

## Гетерогенные эвенты

У каждого события – свой набор атрибутов

## Dynamic schema

Набор атрибутов события – колонок группировки и аргументов агрегатных функций, может изменяться по несколько раз в час

# Особенности модели данных

## Гетерогенные эвенты

У каждого события – свой набор атрибутов

## Dynamic schema

Набор атрибутов события – колонок группировки и аргументов агрегатных функций, может изменяться по несколько раз в час

## Материализация агрегатов

Хотим строить поминутные графики “сколько уникальных юзеров голосовало во Франции за последние сутки”

# Особенности модели данных

## Гетерогенные эвенты

У каждого события – свой набор атрибутов

## Dynamic schema

Набор атрибутов события – колонок группировки и аргументов агрегатных функций, может изменяться по несколько раз в час

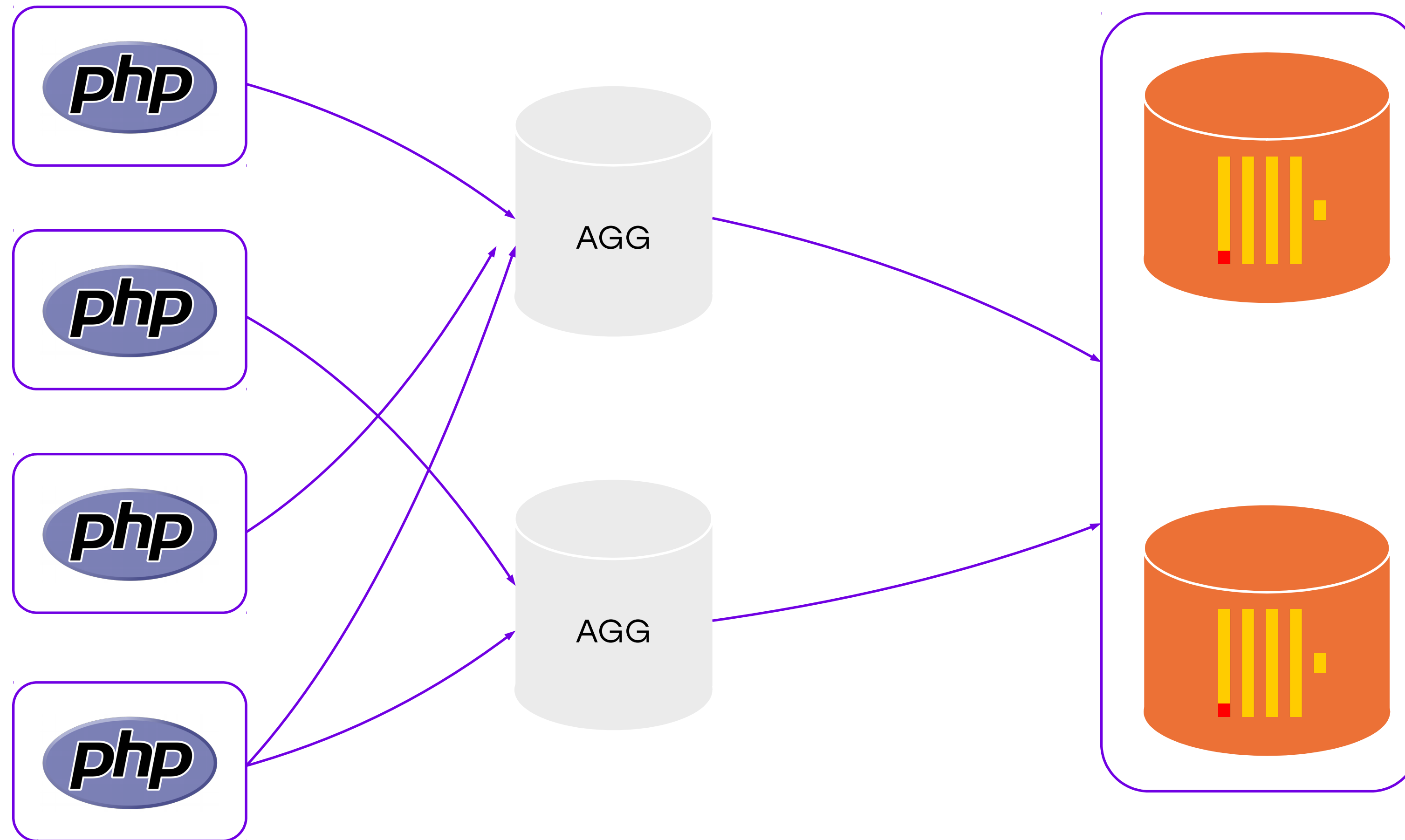
## Материализация агрегатов

Хотим строить поминутные графики “сколько уникальных юзеров голосовало во Франции за последние сутки”

## Переиспользование результатов

Результаты агрегации отправляются в time-series базу + доступны для SQL запросов

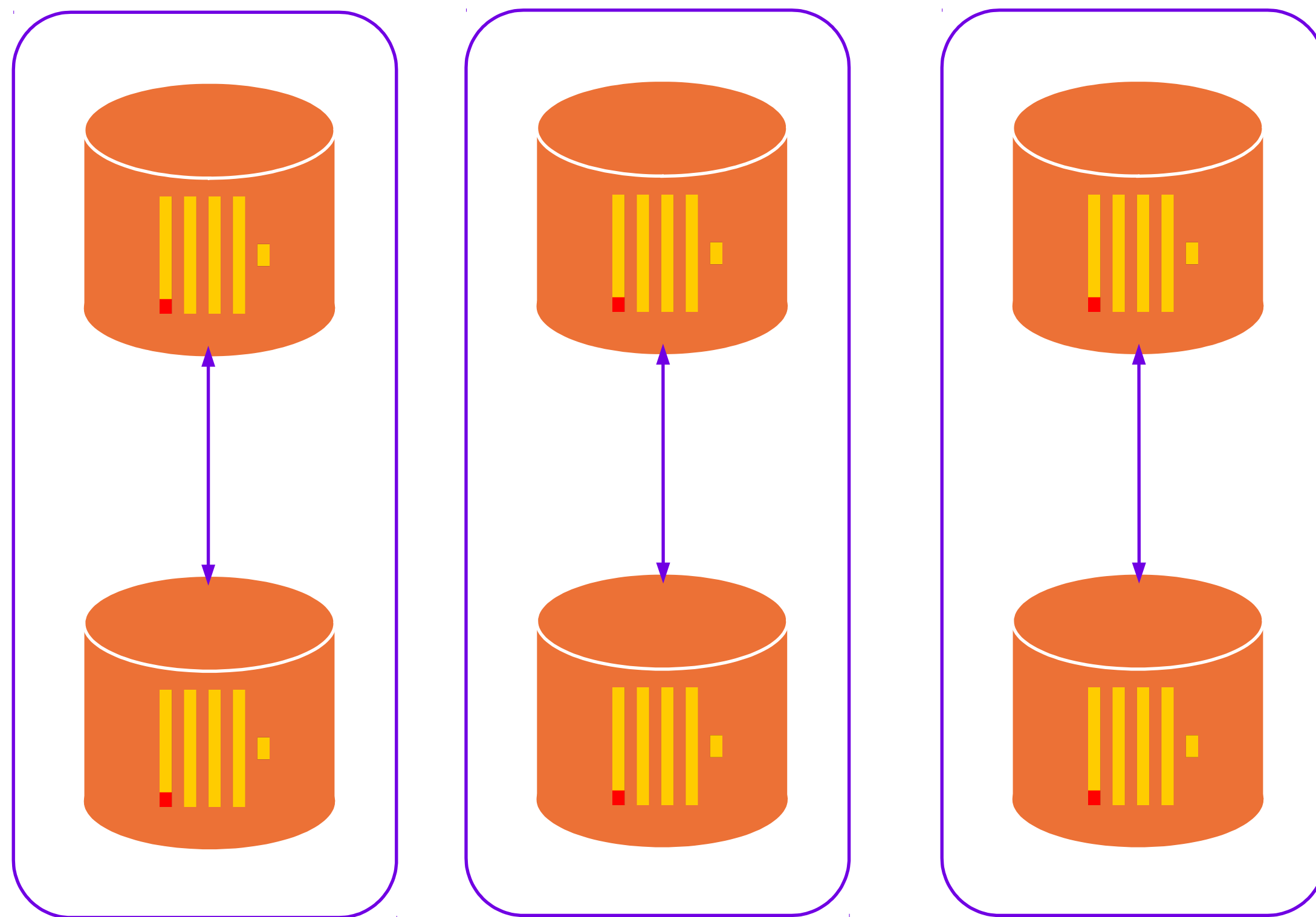
# Поток данных





# Архитектура кластера

Европа

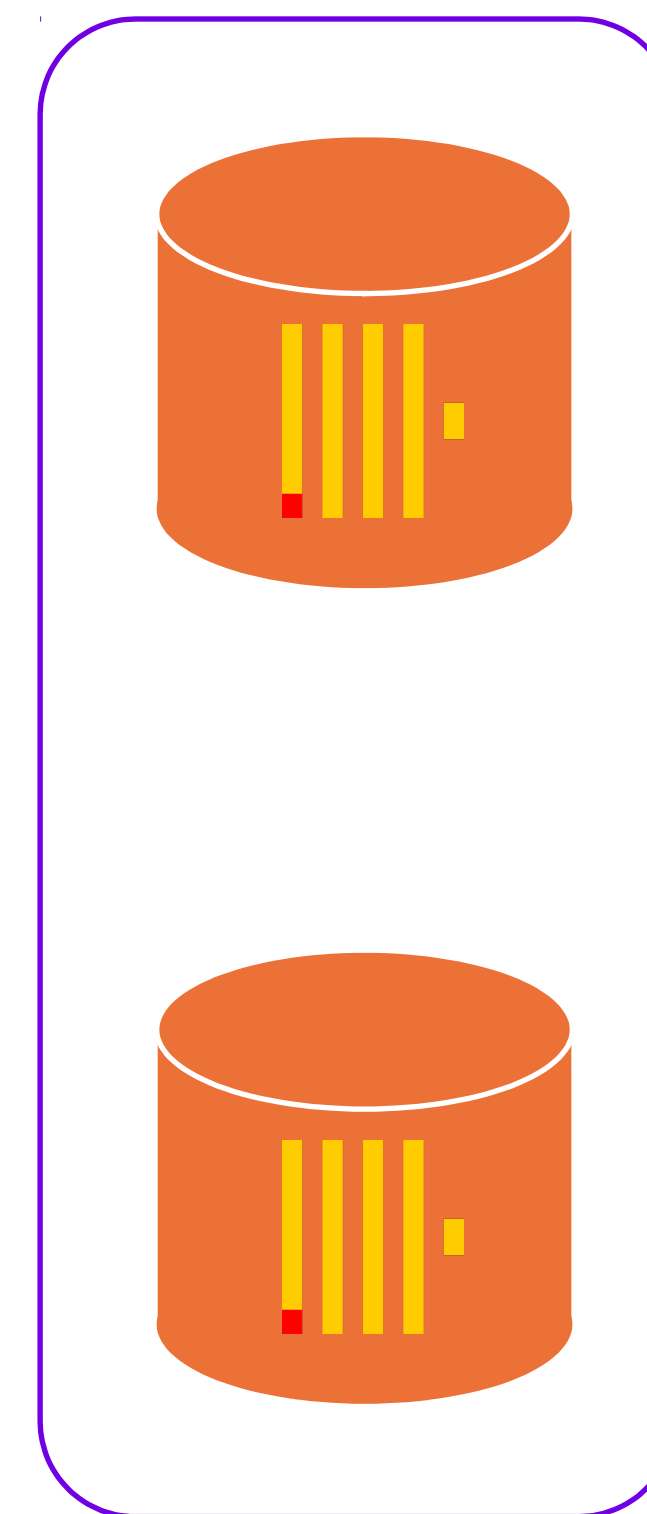


Shard 1

Shard 2

Shard 3

Америка



Shard 1

# Структура хранилища

## Физическая - ReplicatedMergeTree

Хранилище “сырых” событий, replication factor 2

# Структура хранилища

## **Физическая - ReplicatedMergeTree**

Хранилище “сырых” событий, replication factor 2

## **Логическая (запись) – Distributed для датацентра**

Каждый хост-агрегатор пишет в Distributed таблицы, смотрящие на свой ДЦ

# Структура хранилища

## **Физическая - ReplicatedMergeTree**

Хранилище “сырых” событий, replication factor 2

## **Логическая ⟨запись⟩ – Distributed для датацентра**

Каждый хост-агрегатор пишет в Distributed таблицы, смотрящие на свой ДЦ

## **Логическая ⟨чтение⟩ – Distributed world-wide**

Чтение из объединённой таблицы, смотрящей на все хосты двух кластеров

- Получить конфиг обработки события
- Посчитать промежуточные агрегаты из таблицы **Events**
- Сохранить состояния агрегатных функций в **Agg\_\$Event**
- Объединить состояния агрегатных функций и вставить в **Results**

# Схема таблиц - Events

```
CREATE TABLE default.events  
(  
    dt Date,  
    ts DateTime,  
    event_type String,  
    cnt Int32,  
    keys Array(String),  
    vals Array(String)  
) ENGINE = ReplicatedMergeTree(...)
```

# Схема таблиц - Agg\_VoteEvent

```
CREATE TABLE default.Agg_VoteEvent
(
    ts DateTime,
    user_gender UInt32,
    user_country UInt32,
    cnt AggregateFunction(sum, UInt32),
    uniq AggregateFunction(uniqCombined, UInt32)
) ENGINE = ReplicatedMergeTree(...)
```

# Схема таблиц - Results

```
CREATE TABLE default.Result  
(  
    ts DateTime,  
    event String,  
    dim_keys Array(String),  
    dim_vals Array(String),  
    agg_keys Array(String),  
    agg_vals Array(Float32),  
) ENGINE = ReplicatedMergeTree(...)
```



# Цифры с нолями

badoo\_tech

7 000 000 метрик на выходе

1 000 000 событий/секунду входная мощность

1 000 типов событий

10 терабайт данных в месяц

# СПАСИБО!

