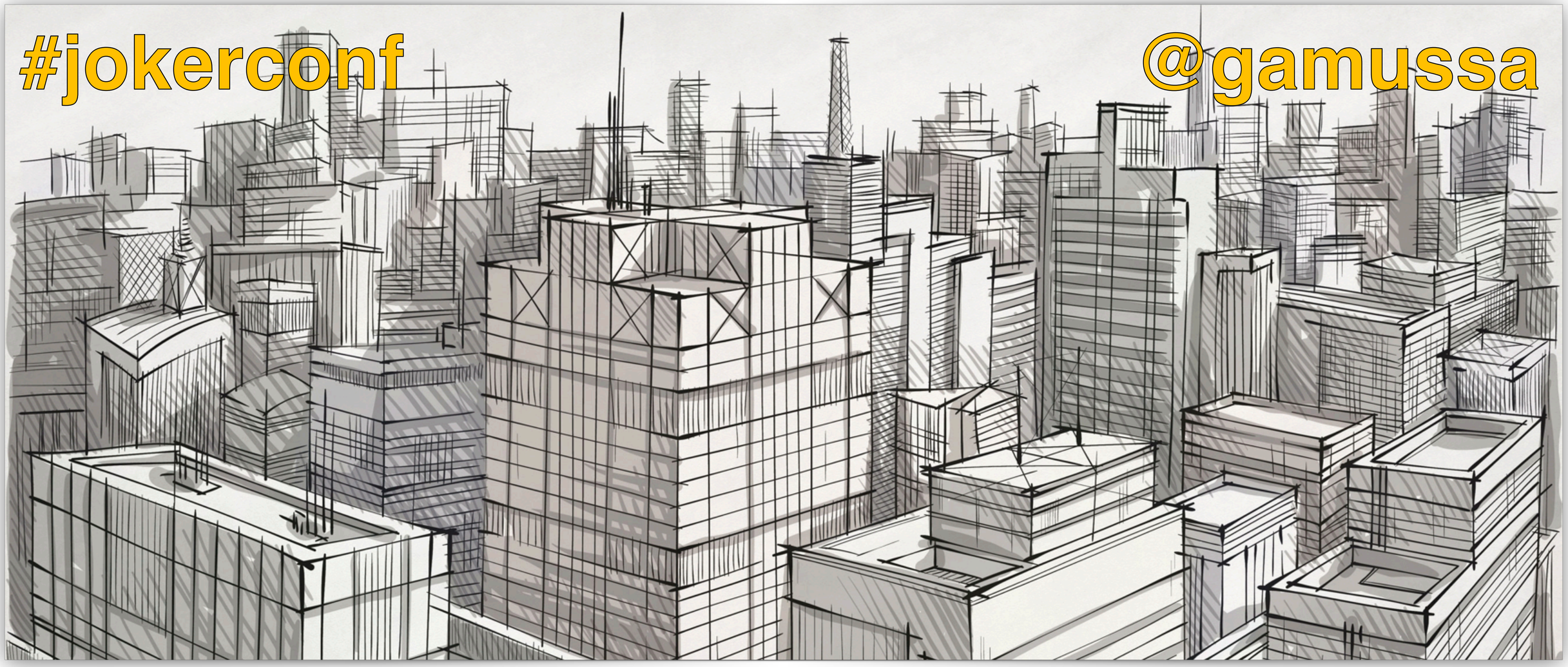
Распределенные Системы

для очень занятого джависта



```
> whoami
```

I'M NOT A SMART MAN...

Почему вы здесь?

- Слушаете `#razborpoletov`
- Нечем больше занять субботним вечером
- Интересуетесь распределенными системами
- For real, люди, почему вы здесь?

Внимание! Disclaimer!

- Распределенные системы – это сложно!
- Теория – **сложна**!
- Имплементация и практика – **еще сложнее**!
- Доклад построен на моем **личном** понимании вещей и личном **опыте**
- А так же, на основе разговоров с моими коллегами из Hazelcast
- Поэтому, я могу быть **не прав или не точен**!

Или немного Капитан О!

О чем пойдет речь

- Модели распределенных систем
- Время в распределенных системах
- Данные в распределенных системах
- Проблемы в распределенных системах

«Разработка ***распределенных систем*** *- это искусство решения задач, которые можно решить, используя* ***один компьютер****, используя* ***много компьютеров****»*

Программы (в т.ч. распределенные)…

- чего-то считают
- что-то хранят

Какие бывают системы

- Синхронные
- Асинхронные

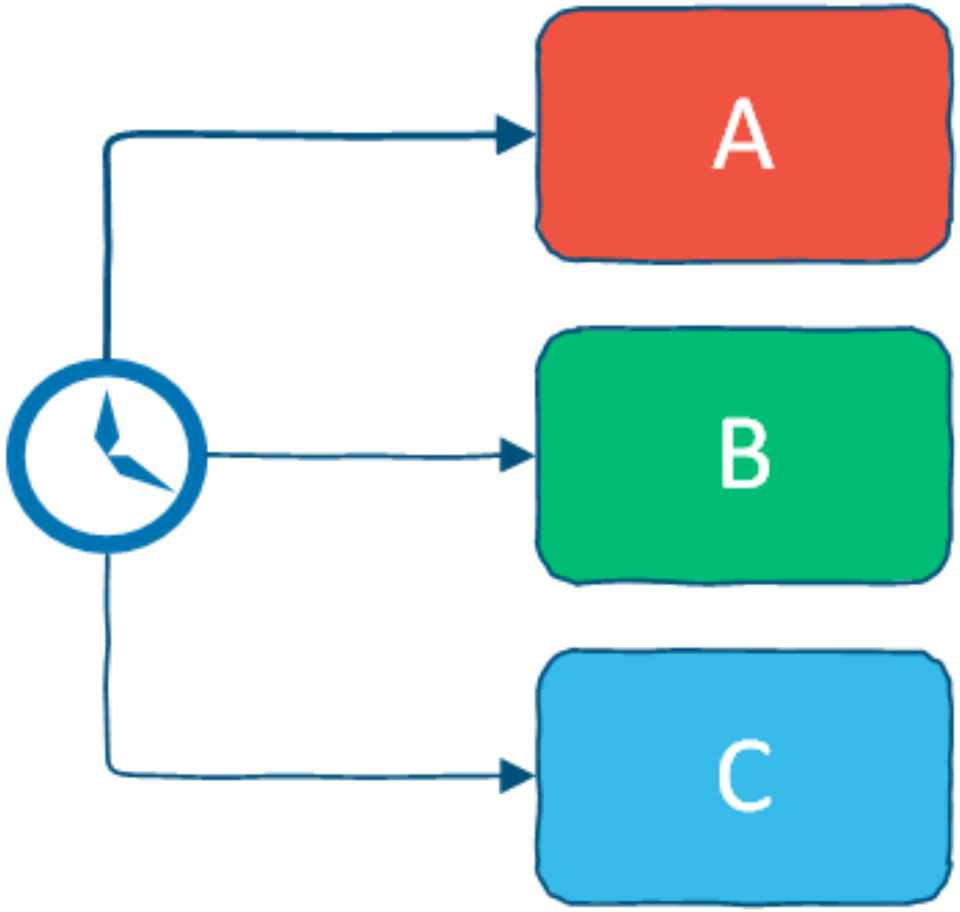
Время и порядок операций

Порядок

Как время используется в распределенных системах?

- Время может определить порядок операций в системе
- Время может определить пограничные условия для операций

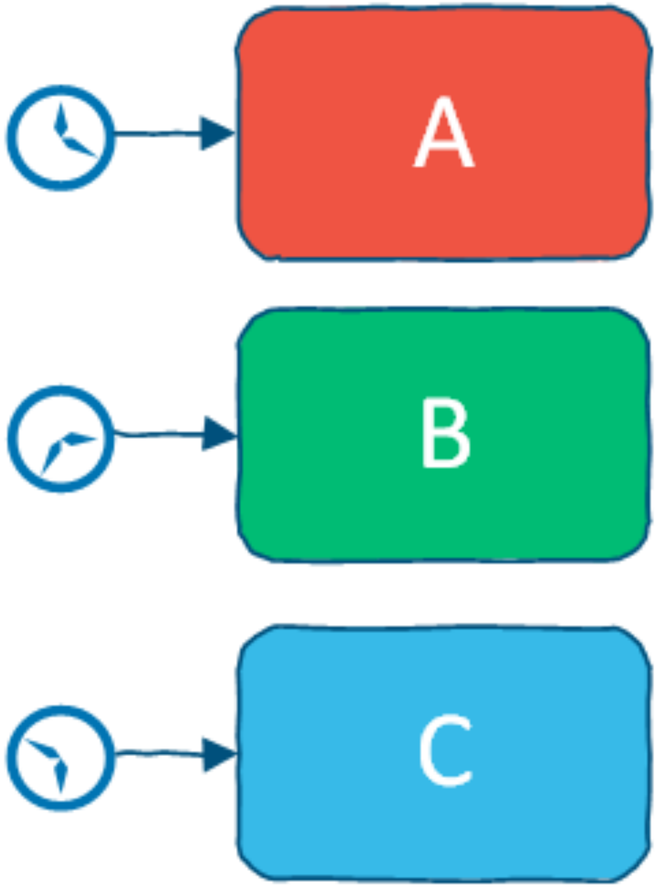
Идеально точные глобальный часы



Используется

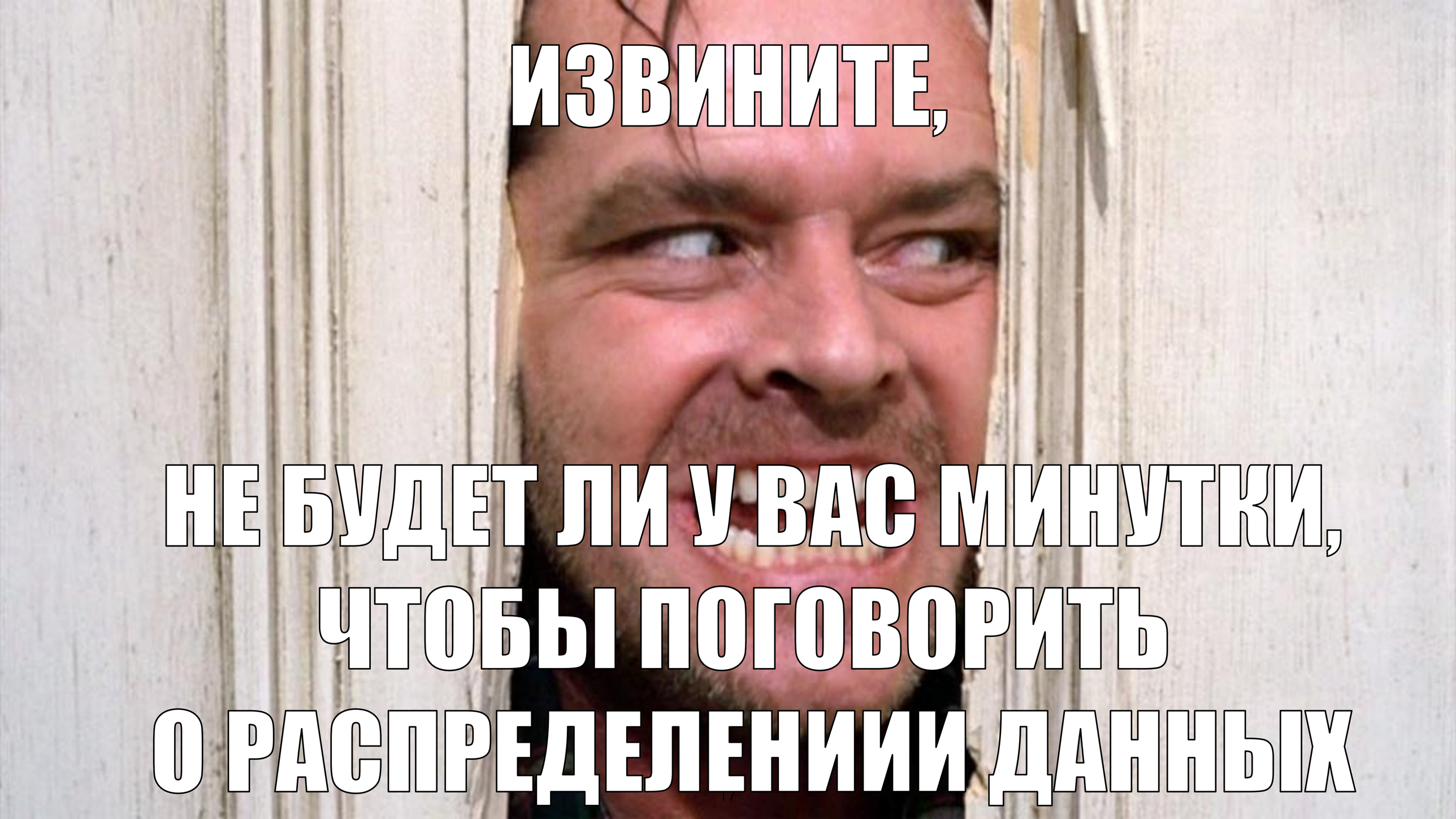
- Apache Cassandra (использует NTP)
- Google Spanner: Globally-Distributed Database (использует GPS, атомные часы)

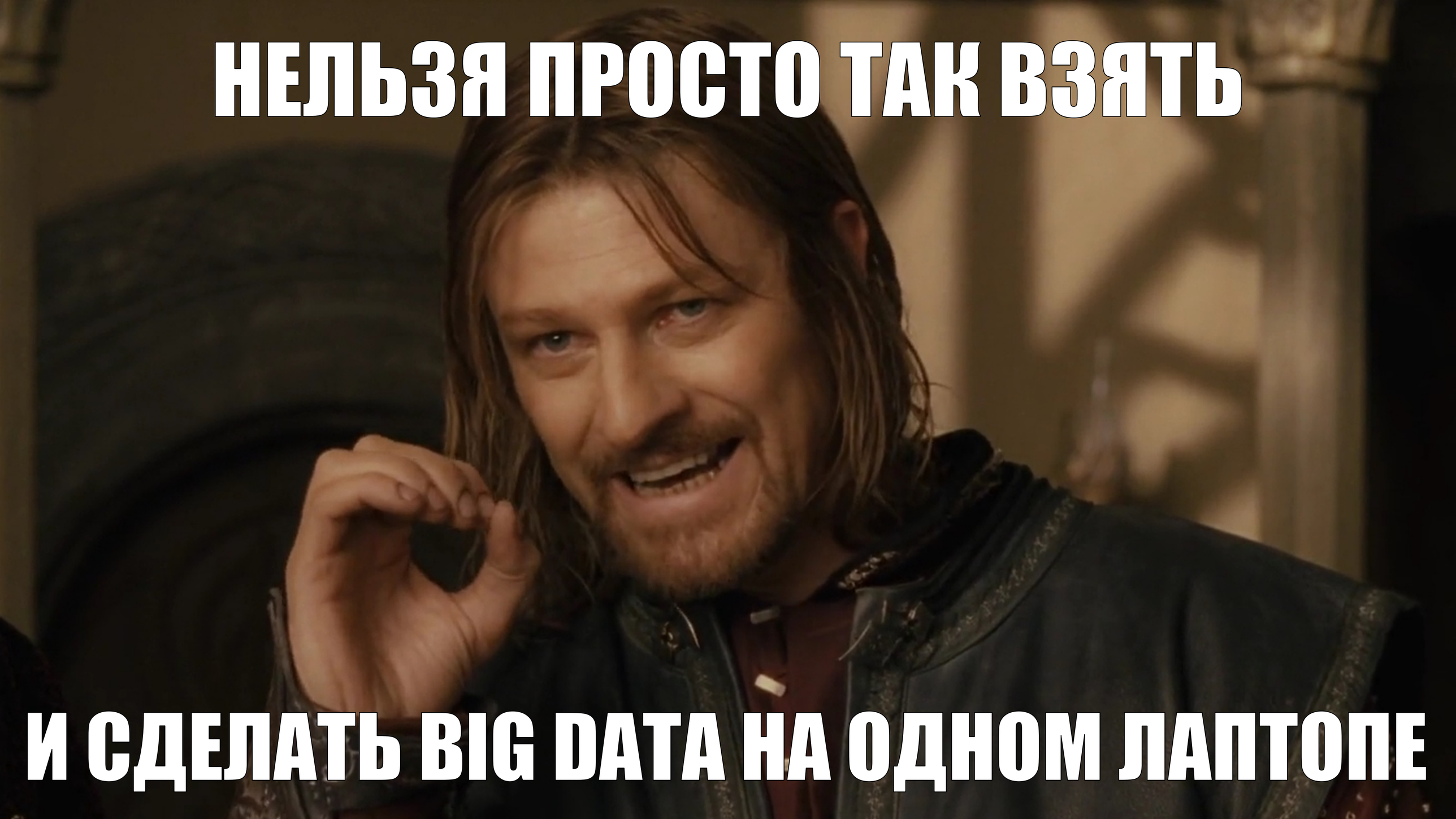
Локальные часы



Часы Лэмпорта

- счётчик увеличивается перед каждым событием процесса
- при отправке сообщения значение счётчика прикрепляется к сообщению
- при получении сообщения значение счётчика процесса-получателя выставляется в максимум текущего и полученного значения и увеличивается на 1

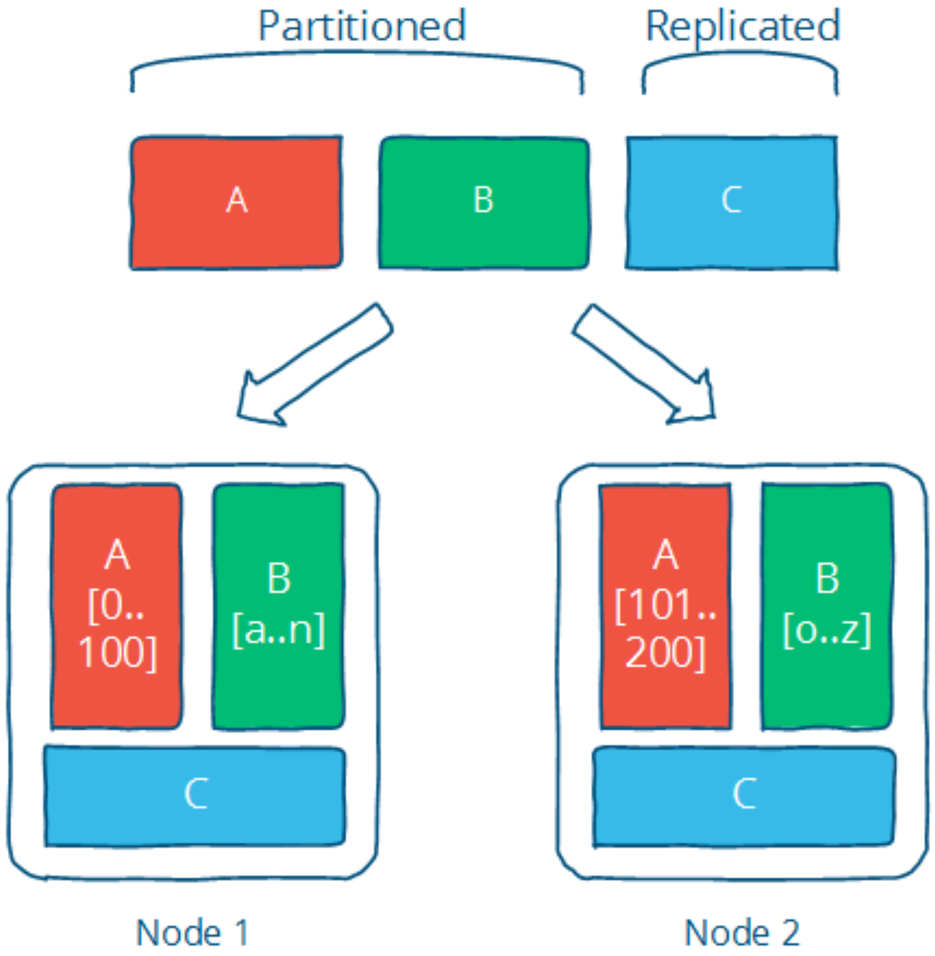

ИЗВИНИТЕ,
НЕ БУДЕТ ЛИ У ВАС МИНУТКИ,
ЧТОБЫ ПОГОВОРИТЬ
О РАСПРЕДЕЛЕНИИИ ДАННЫХ


НЕЛЬЗЯ ПРОСТО ТАК ВЗЯТЬ
И СДЕЛАТЬ BIG DATA НА ОДНОМ ЛАПТОПЕ

РЕПЛИКАЦИЯ

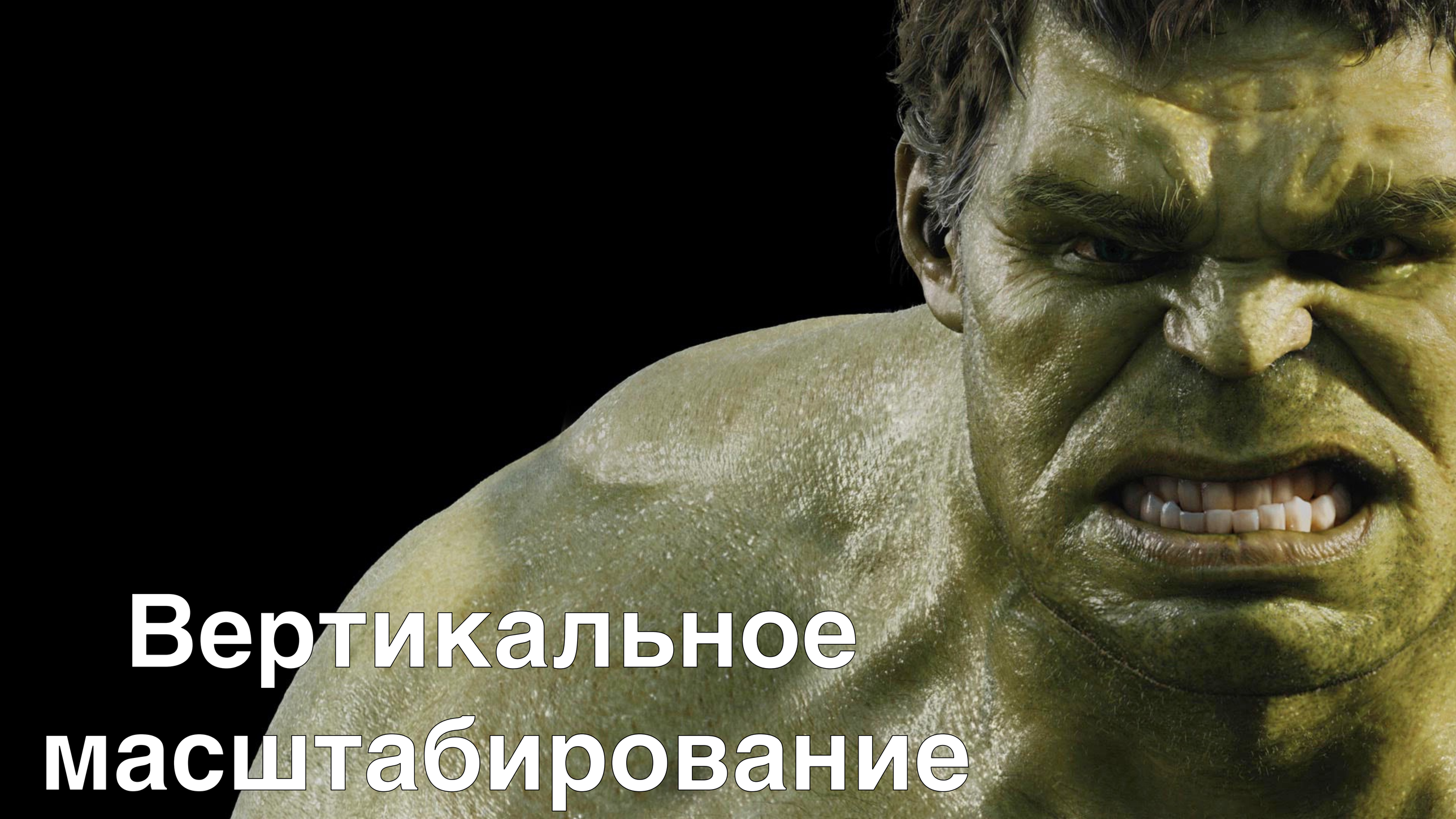
ШАРДИНГ

Репликация или шардинг?



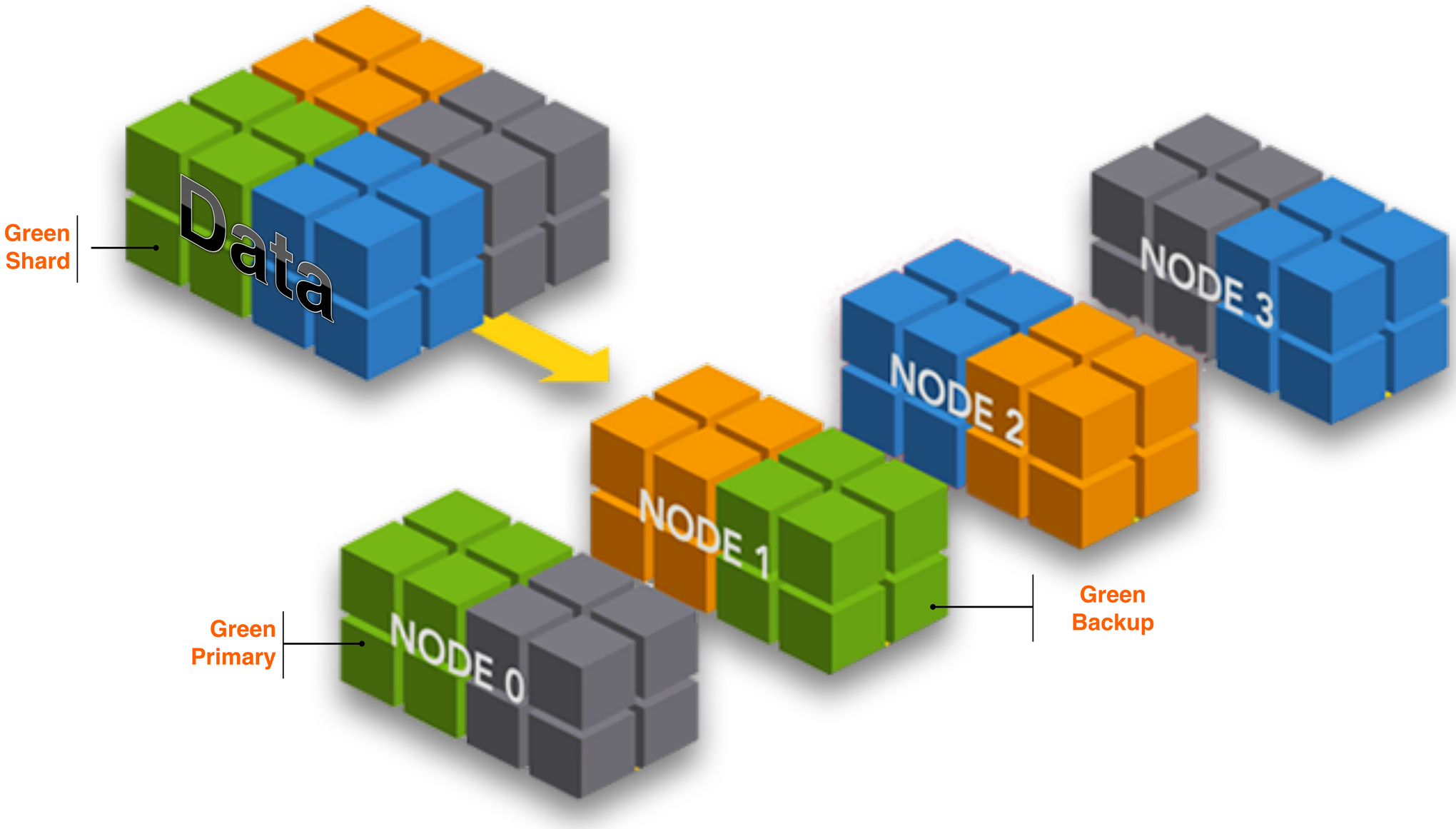
http://book.mixu.net/distsys/single-page.html

Горизонтальное масштабирование



Вертикальное масштабирование

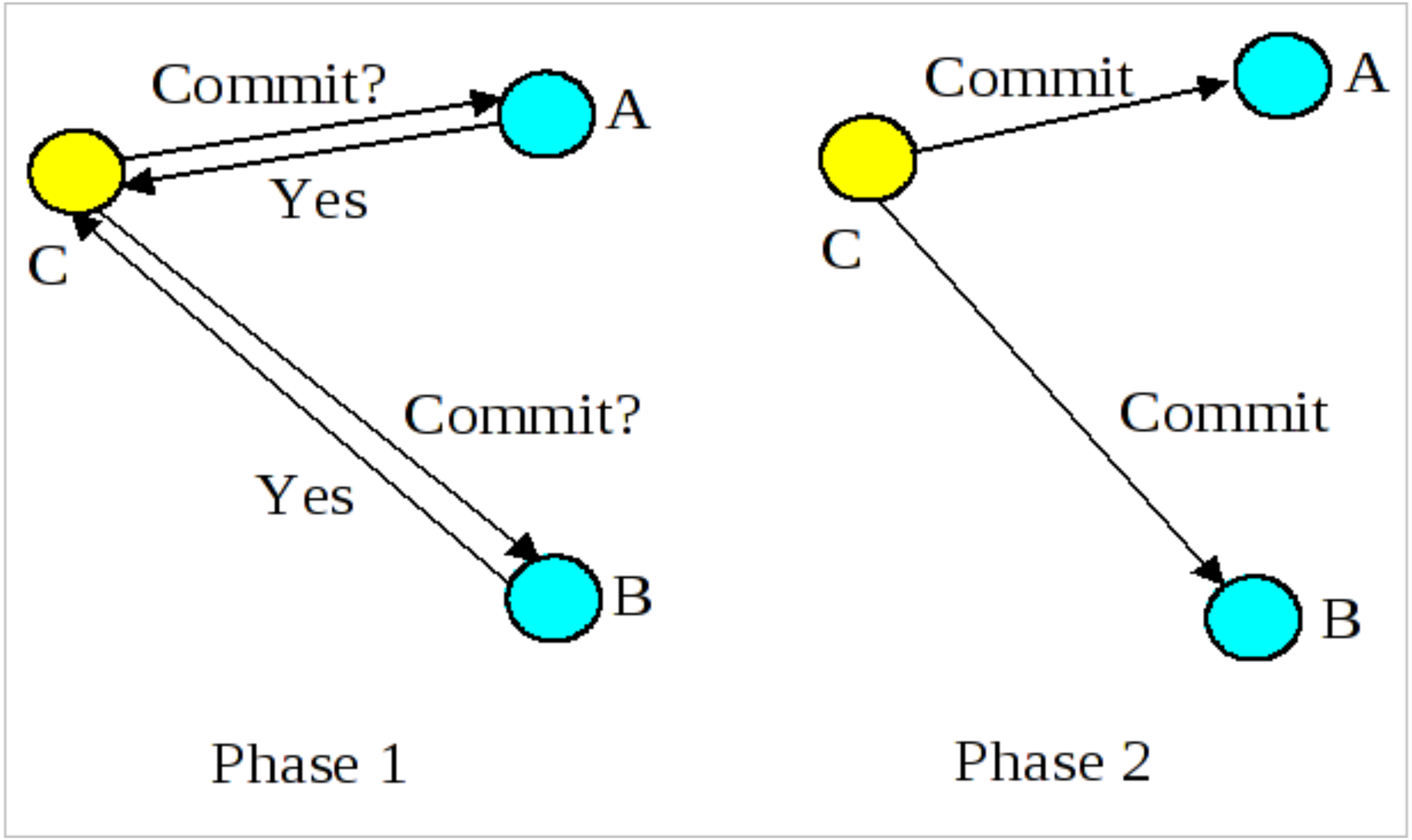
Я НЕ ВСЕГДА ДЕЛАЮ РЕЗЕРВНЫЕ КОПИИ ДАННЫХ
НО КОГДА Я ИХ ДЕЛАЮ, Я ХРАНЮ ИХ В ПАМЯТИ


Green
Shard
Data
NODE 3
NODE 2
NODE 1
NODE 0
Green
Primary
Green
Backup

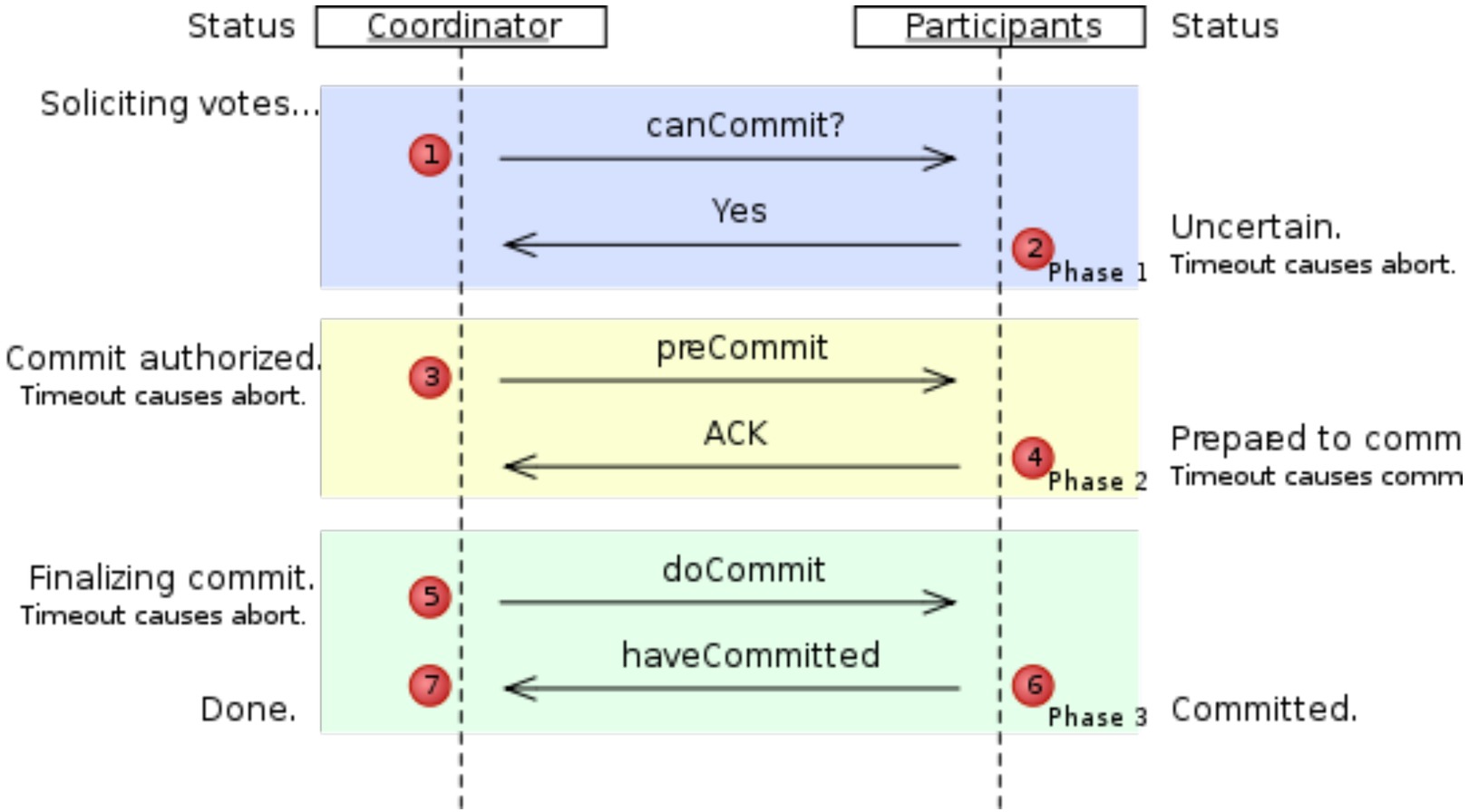
Проблемы консенсуса

- Согласованность / Agreement
  - Процессы должны согласовать непротиворечивость результата
- Валидность / Validity
  - Все процессы предлагающие значение *V*, и все процессы должны согласиться на *V*
- Окончание / Termination
  - Все процессы должны прийти к решению

Двух-фазный коммит



Трех-фазный коммит



CAP теорема

СДУР / CAP

- **С**огласованность / **C**onsistency
- **Д**оступность / **A**vailability
- **У**стойчивость к **Р**азделению / **P**artition Tolerance

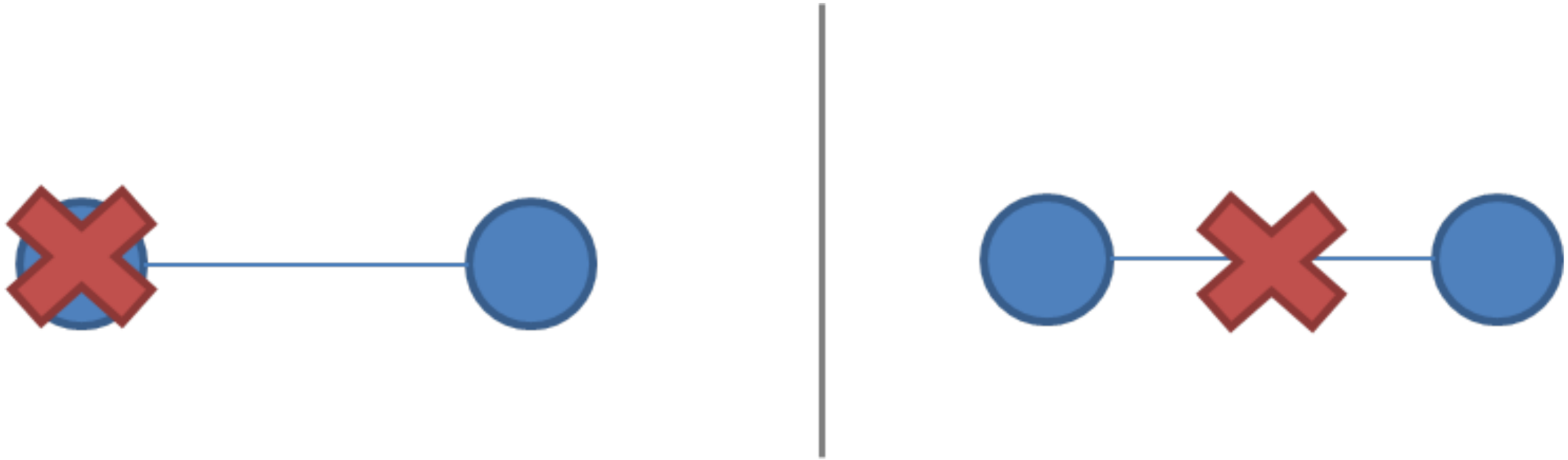
Выбери два…

- *CP*: **C**onsistent & **P**artition-tolerant (жертвуем **A**vailability)
- *AP*: **A**vailable & **P**artition-tolerant (жертвуем **C**onsistency)
- *CA*: **C**onsistent & **A**vailable (жертвуем **P**artition-tolerance)

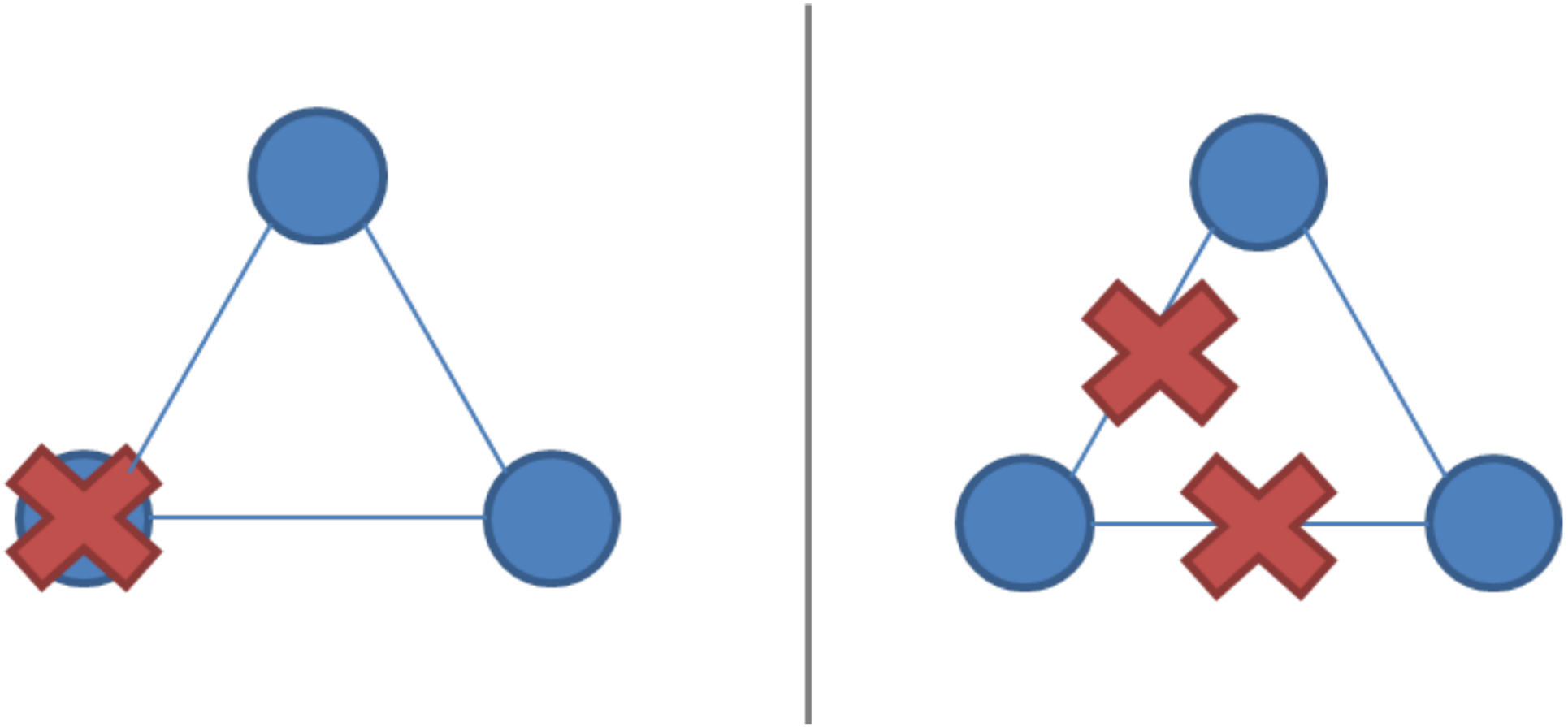
CA

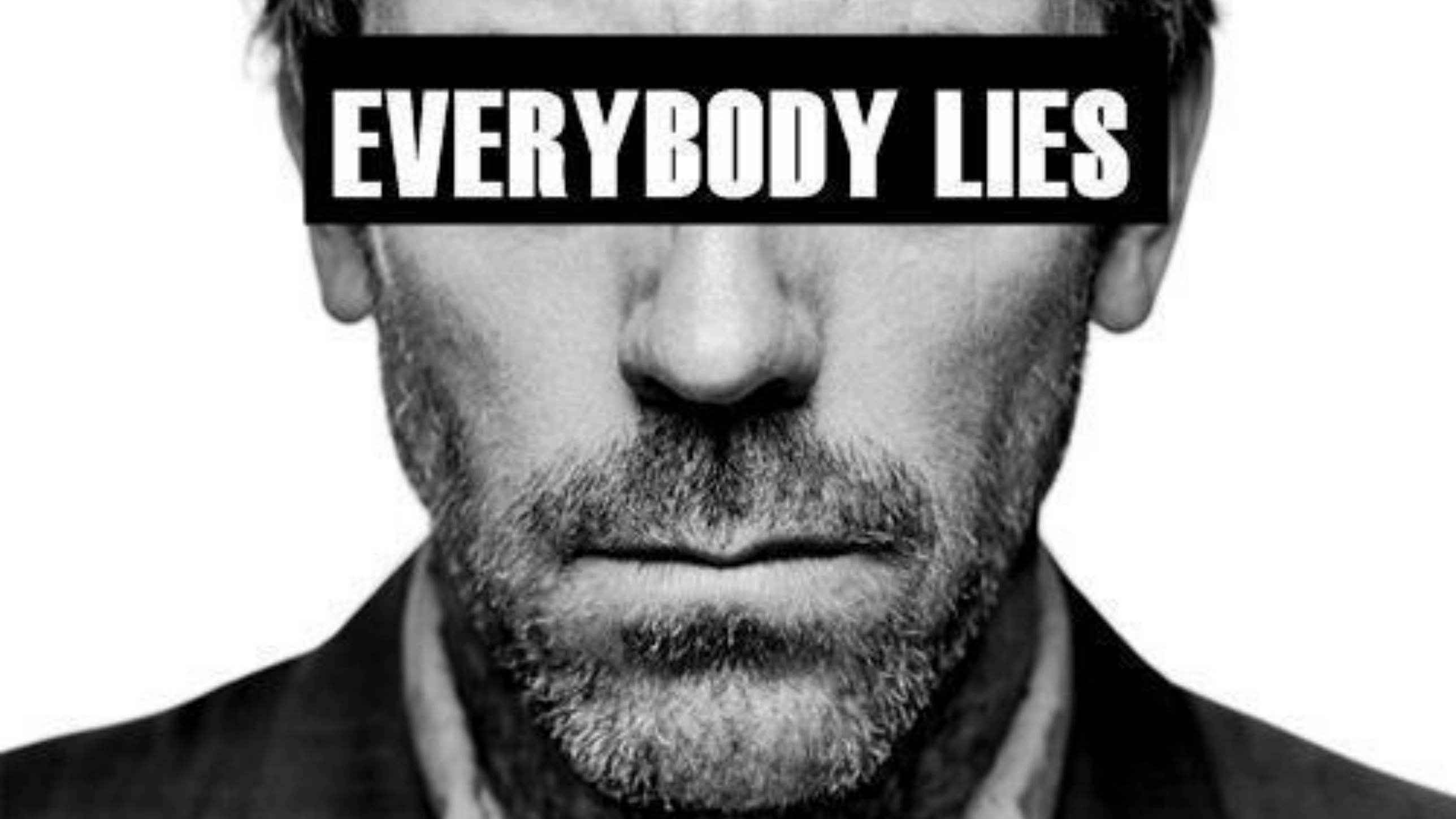
- Single-node база данных, и нечего распределять
- «Сеть надежна» ©

Что такое Partition



Что такое Partition




EVERYBODY LIES

Распределенное вранье

8 врак о распределенных системах

- Сеть надежна
- Задержка равна `0`
- Пропускная способность бесконечна
- Сеть безопасна
- Топология не меняется
- У нас только один администратор
- Транспорт данных не несет высоких накладных расходов
- Сеть однородна

CA

- Single-node база данных, и нечего распределять
- «Сеть надежна» ©
- не получится CA

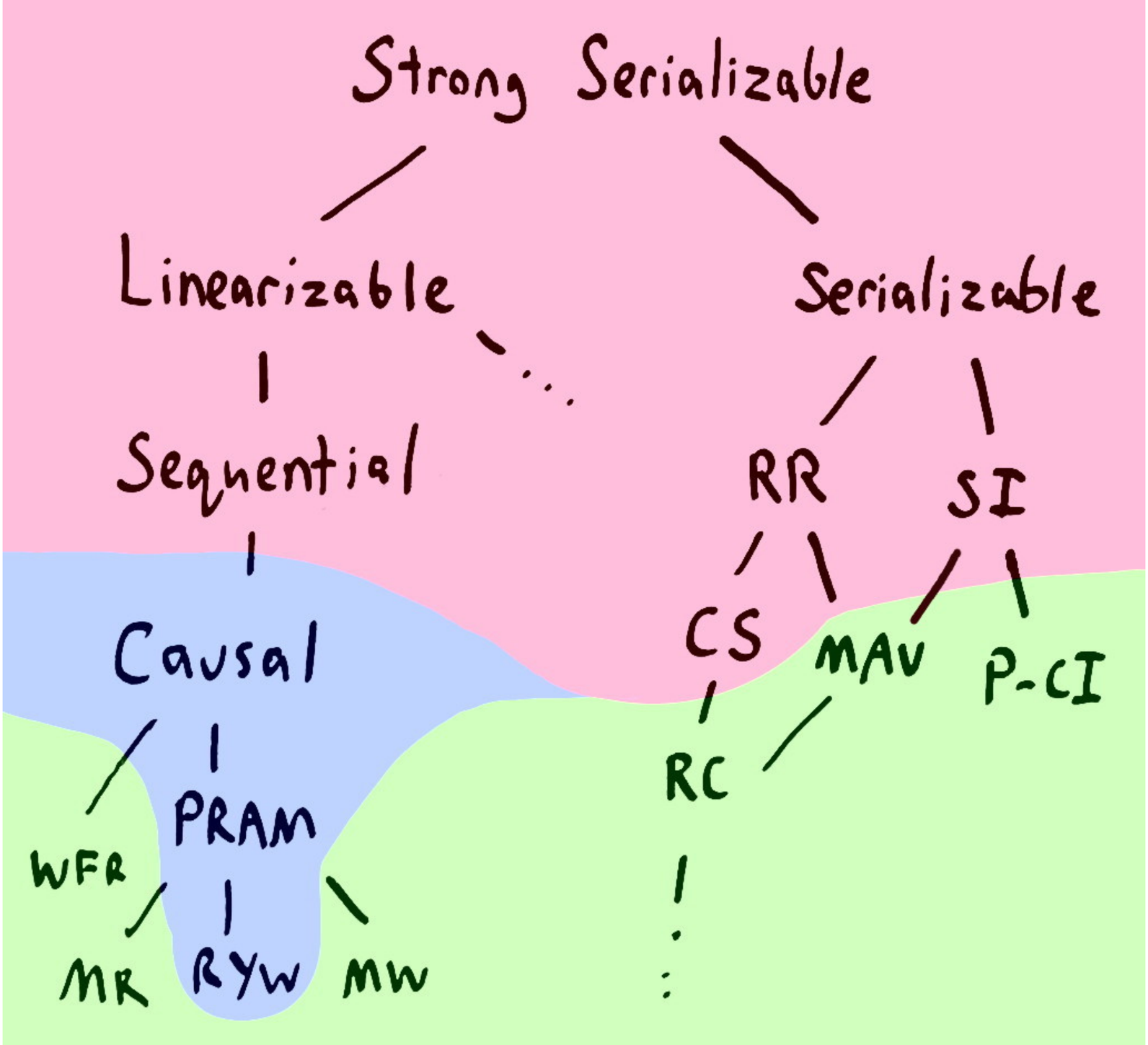
ТАКИЕ ДЕЛА ©

CP

- Согласованность в треминах CAP – Линеаризуемая (строгая) согласованность
- требует распределённую общую память, которая
  - будет вести себя как будто выполняется на одной ноде
  - отвечает на запросы один за раз
- Результат операции будет виден всем нодам

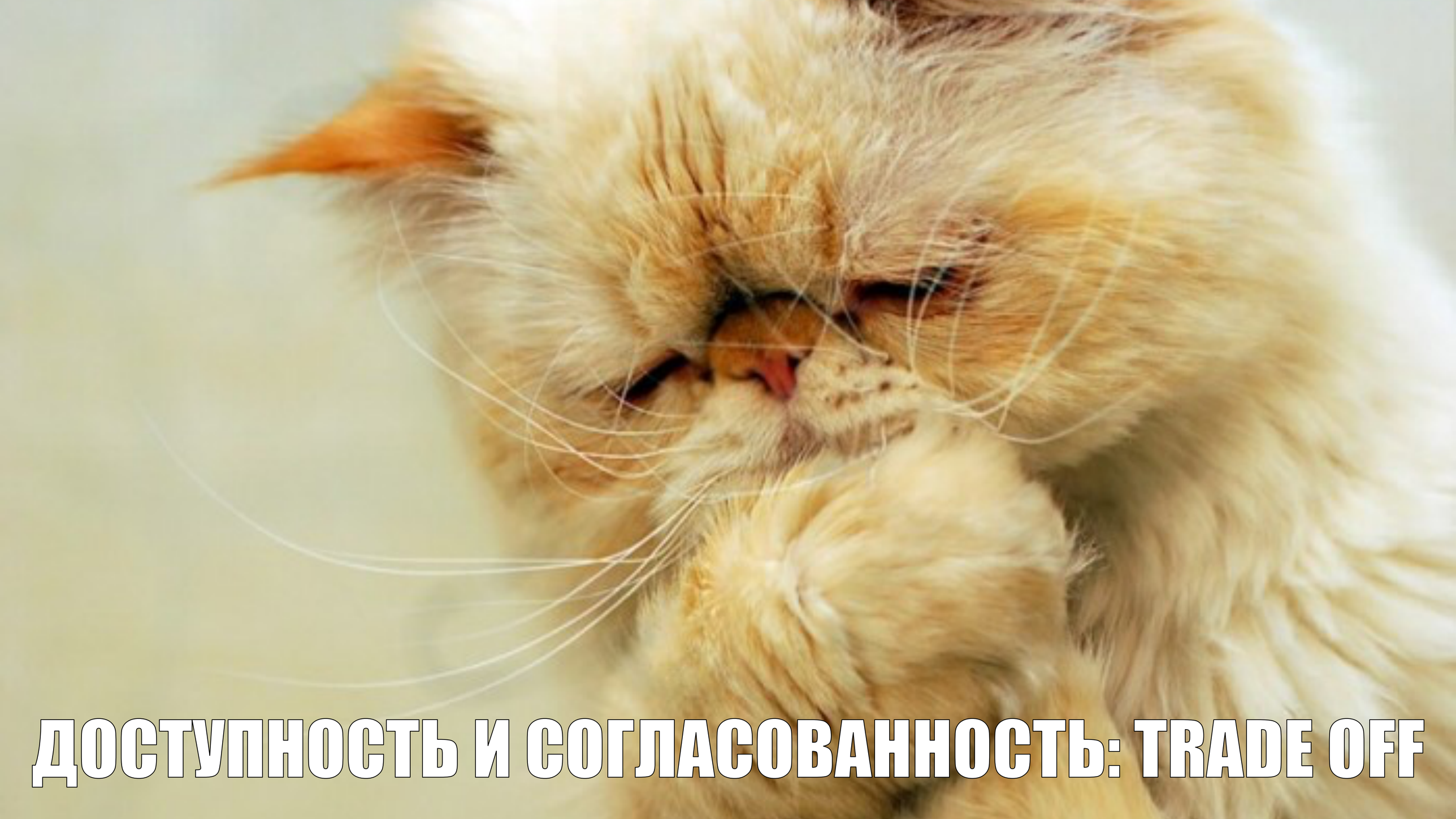
Линеаризуемость

- «Тяжеловато» ©
- Можно построить систему и без этого


Strong Serializable
Linearizable
Serializable
Sequential
RR
SI
Causal
CS
MAV
P-CI
RC
PRAM
WFR
MR
RYW
MW

AP

- каждая рабочая нода должна возвращать результат
- наша задача – High Availability



ДОСТУПНОСТЬ И СОГЛАСОВАННОСТЬ: TRADE OFF

Использованные материалы

1. Distributed Systems for Fun and Profit, http://book.mixu.net/distsys/index.html
2. The Eight Fallacies of Distributed Computing
   https://blogs.oracle.com/jag/resource/Fallacies.html
3. Designing Data-Intensive Applications by Martin Kleppmann.
4. Strong Consistency Models, https://aphyr.com/posts/313-strong-consistency-models

#jokerconf
СПАСИБО ЗА ВНИМАНИЕ!!!
Москва
РОССИЯ 24
ТЕБЕ СПАСИБО!!!
Москва
И ТЕБЕ СПАСИБО!!!
@gamussa
ВСЕМ СПАСИБО!!!
risovach.ru