МТС Hello, Conference!_

КОНФЕРЕНЦИЯ **АРХИТЕКТУР@ ПРОГРАМ*НЫХ СИСТЕМ**

/ 7 февраля

{_

helloconf@mts.ru

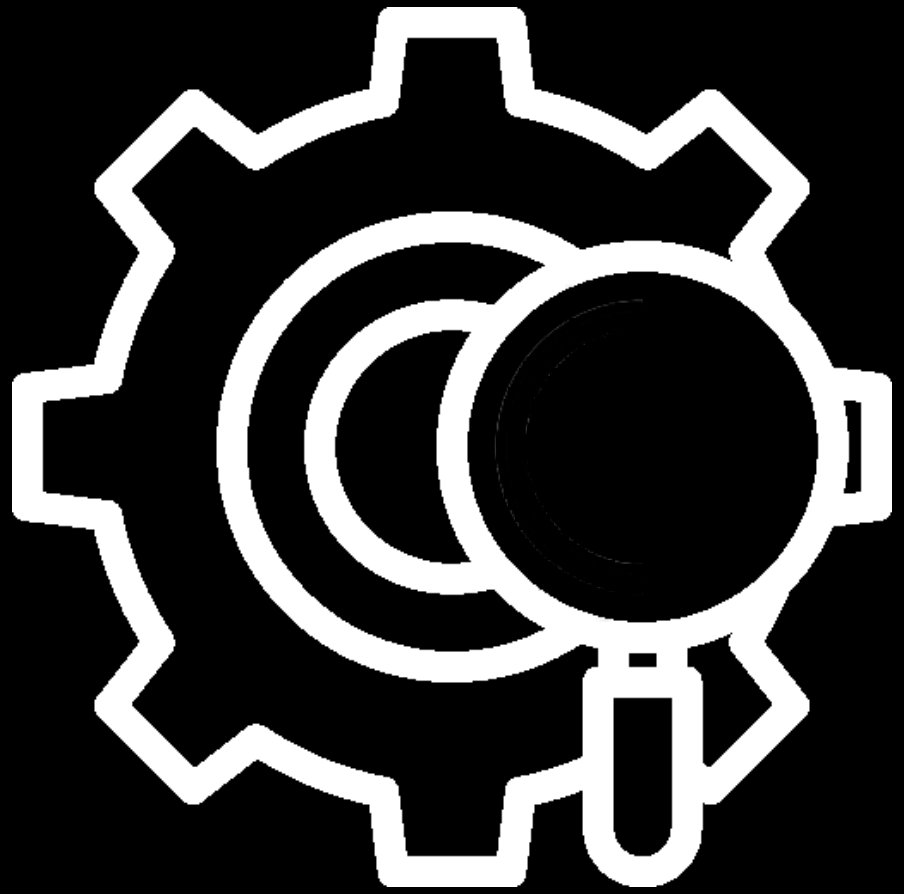
Дзюба Дмитрий

- руководитель центра
- Центр R&D БИТ
- ПАО МТС

Check-list для архитектора

В докладе мы рассмотрим типовые ошибки в области архитектуры, выявленные в процессе внутреннего исследования продуктовых команд.

Исследование внутренних продуктовых команд



Компания снижает риск выпуска некачественного продукта и уменьшает финансовые и репутационные потери от неэффективной организации производства

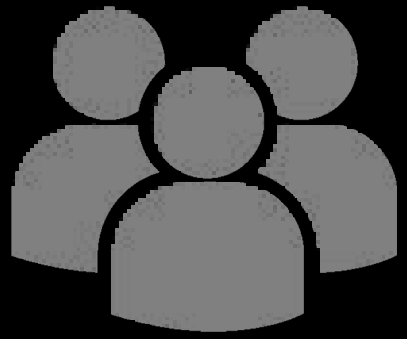
Команды обогащаются опытом и обретают возможность улучшить процессы на основании рекомендаций экспертов

Два подхода

Стремление к Идеалу

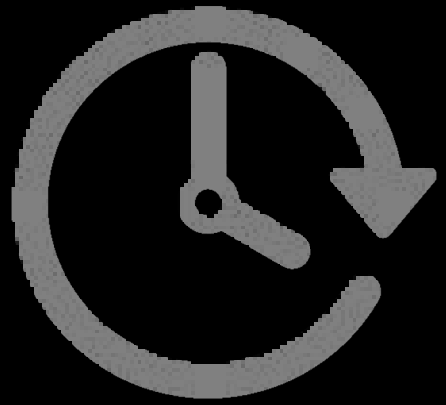
Защита от неудач

Статистика исследования за прошлый год

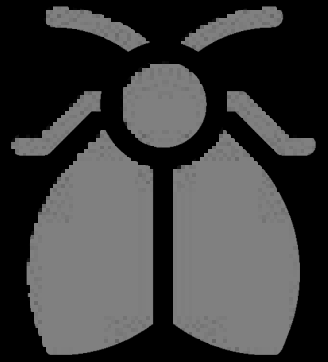


20
Продуктовых команд для аудита

15
Экспертов, проводящих аудит



~ 31
рабочий день на аудит одной команды



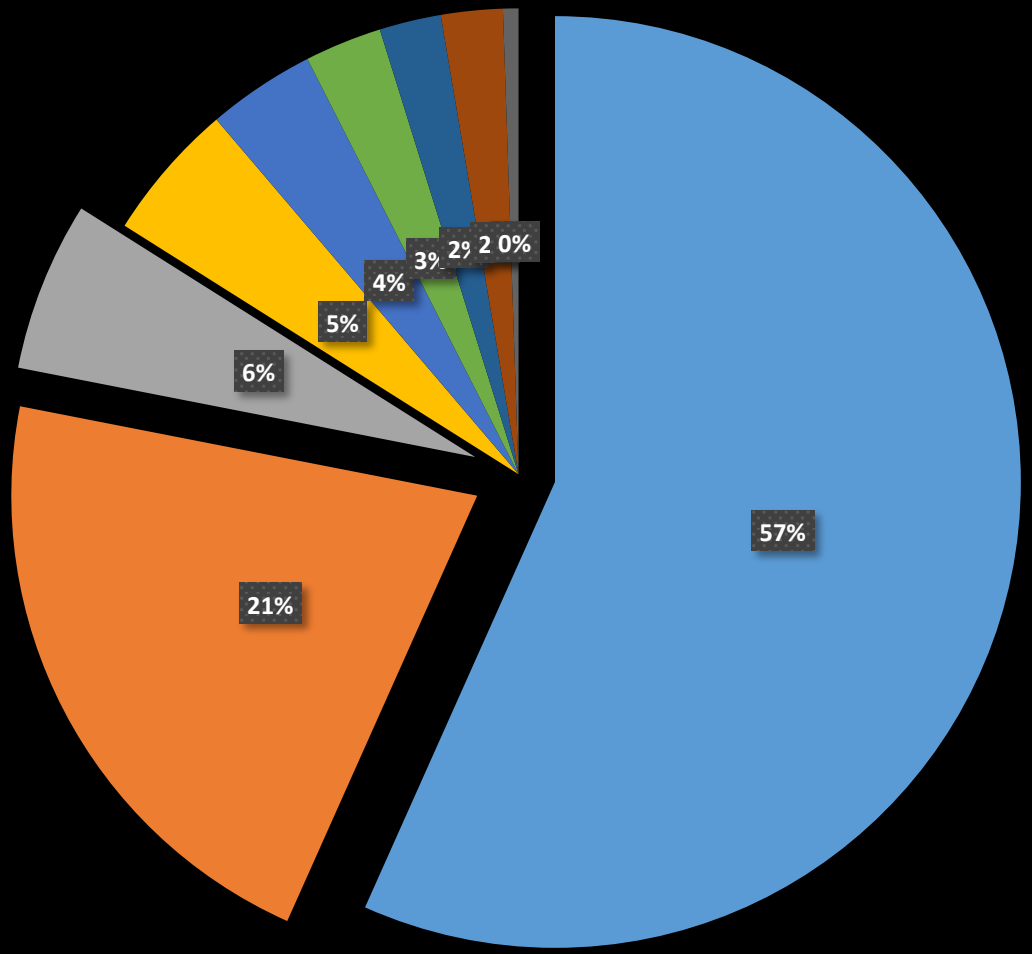
187
рисков связанных с архитектурой

Типы архитекторов/архитектур

1. Enterprise архитектор
2. Solution архитектор
3. Application архитектор
4. Data/Information архитектор
5. Архитектор инфраструктуры
6. Архитектор безопасности
7. Cloud архитектор

наш тип архитекторов

Распределение рисков по группам



- Процесс
- Безопасность
- Производительность
- Модульность
- Качество
- Доступ к данным
- Доступность
- Модифицируемость
- Наблюдаемость

TOP 3

1> Ошибки в процессе проектирования _

1: Процесс>_ как проектировать ?

Проектирование сверху вниз

Проектирование снизу вверх

1: Процесс>_ выбираем способ

Проектирование сверху вниз

- Большие проекты
- Большие команды
- Понятные и стабильные требования

Проектирование снизу вверх

- Небольшие проекты
- Небольшая техническая сложность
- Небольшая команда
- Изменчивые требования

1: Процесс>_ что нужно для начала проектирования

- ✓ Архитектурно – значимые варианты использования
- ✓ Нефункциональные требования
- ✓ Ограничения и зависимости
- ✓ Срок жизни продукта

1: Процесс>_ что нужно для начала проектирования

- ✓ Архитектурно – значимые варианты использования
- ✓ Нефункциональные требования
- ✓ Ограничения и зависимости
- ✓ Срок жизни продукта

Архитектура должна быть документирована в нотации понятной команде проекта

2> Безопасность _

2:Безопасность>_ это просто

- ✓ Аутентификация
- ✓ Авторизация
- ✓ Principle of least privilege
- ✓ Поддержка конфиденциальности данных
- ✓ Поддержка лога аудита

…

2 Пример> Аутентификация по паролю – 10 проблем _

```
User ID  : Dmitry Dzyuba
Password : **********
```

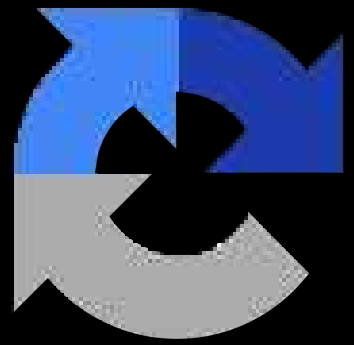
Что тут может быть сложного?

2 Пример> ПРОБЛЕМА №1_

User ID : Dmitry Dzyuba
Password : QWERTY

Веб-приложение позволяет пользователям создавать простые пароли.

2 Пример> ПРОБЛЕМА №2 _



Веб-приложение не защищено от возможности перебора паролей (brute-force attacks).

2 Пример> ПРОБЛЕМА №3_

Веб-приложение само генерирует и распространяет пароли пользователям, однако не требует смены пароля после первого входа (т.е. текущий пароль где-то записан).

2 Пример> ПРОБЛЕМА №4_

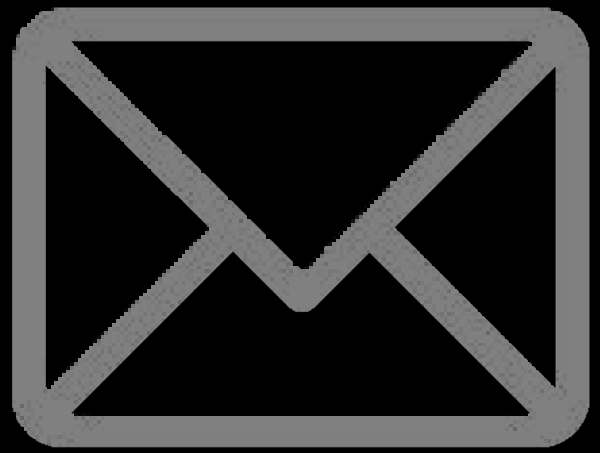
```
http://service.com ? user_id=Dmitriy & password=qwerty
```

Веб-приложение допускает передачу паролей или session token-ов по незащищенному HTTP-соединению либо в строке URL.

2 Пример> ПРОБЛЕМА №5_

Веб-приложение не использует безопасные хэш-функции для хранения паролей пользователей.

2 Пример> ПРОБЛЕМА №6_



Веб-приложение не предоставляет пользователям возможность изменения пароля либо не нотифицирует пользователей об изменении их паролей.

Все еще пример> ПРОБЛЕМА №7_

```
Для восстановления пароля введите имя президента
Российской Федерации: _
```

Веб-приложение использует уязвимую функцию восстановления пароля, которую можно использовать для получения несанкционированного доступа к другим учетным записям.

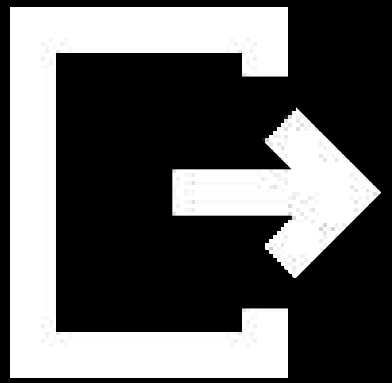
Все еще пример> ПРОБЛЕМА №8 _

Веб-приложение не требует повторной аутентификации пользователя для важных действий: смена пароля, изменения адреса доставки товаров и т. п.

Все еще пример> ПРОБЛЕМА №9 _

- ✓ Веб-приложение создает session tokens таким образом, что они могут быть подобраны или предсказаны для других пользователей.
- ✓ Веб-приложение уязвимо для session fixation-атак (т. е. не заменяет session token при переходе анонимной сессии пользователя в аутентифицированную).
- ✓ Веб-приложение не устанавливает флаги HttpOnly и Secure для browser cookies, содержащих session tokens.

Все еще пример> ПРОБЛЕМА №10 _



Веб-приложение не уничтожает сессии пользователя после короткого периода неактивности либо не предоставляет функцию выхода из аутентифицированной сессии.

3> производительность _

3 Производительность> tradeoffs ….

Performance

Minimal Valuable Product

3 Производительность> правило

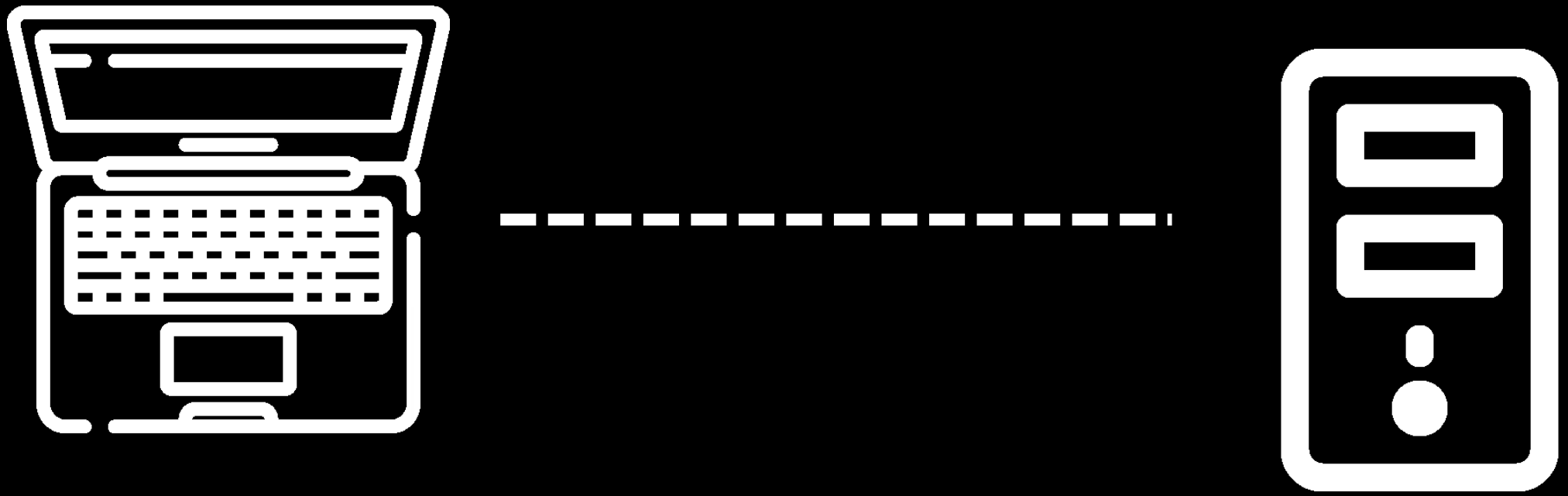
L1 cache reference **0.5 ns**
Branch mispredict **5 ns**
L2 cache reference **7 ns 14x L1 cache**
Mutex lock/unlock **25 ns**
Main memory reference **100 ns 20x L2 cache, 200x L1 cache**
Compress 1K bytes with Zippy **3,000 ns 3 us**

Send 1K bytes over 1 Gbps network **10,000 ns 10 us**
Read 4K randomly from SSD* **150,000 ns 150 us ~1GB/sec SSD**
Read 1 MB sequentially from memory **250,000 ns 250 us**
Round trip within same datacenter **500,000 ns 500 us**

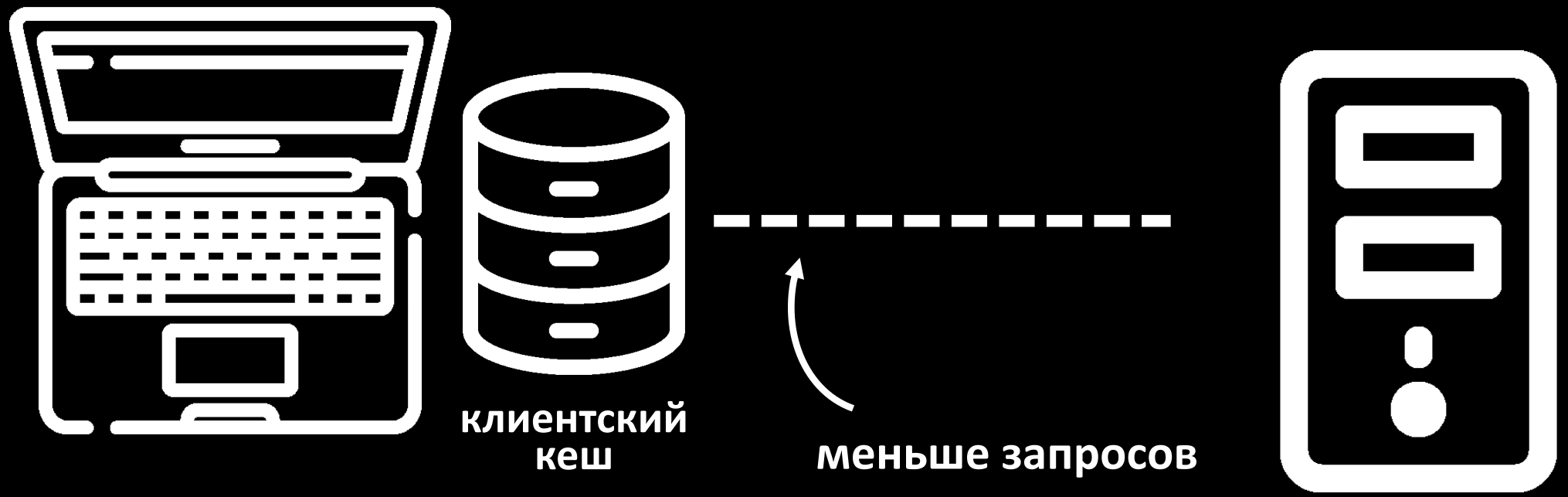
Read 1 MB sequentially from SSD* **1,000,000 ns 1,000 us 1 ms ~1GB/sec SSD, 4X memory**
Disk seek **10,000,000 ns 10,000 us 10 ms 20x datacenter roundtrip**
Read 1 MB sequentially from disk **20,000,000 ns 20,000 us 20 ms 80x memory, 20X SSD**
Send packet CA->Netherlands->CA **150,000,000 ns 150,000 us 150 ms**

Необходимо уменьшить число запросов к медленным ресурсам в ходе работы программы

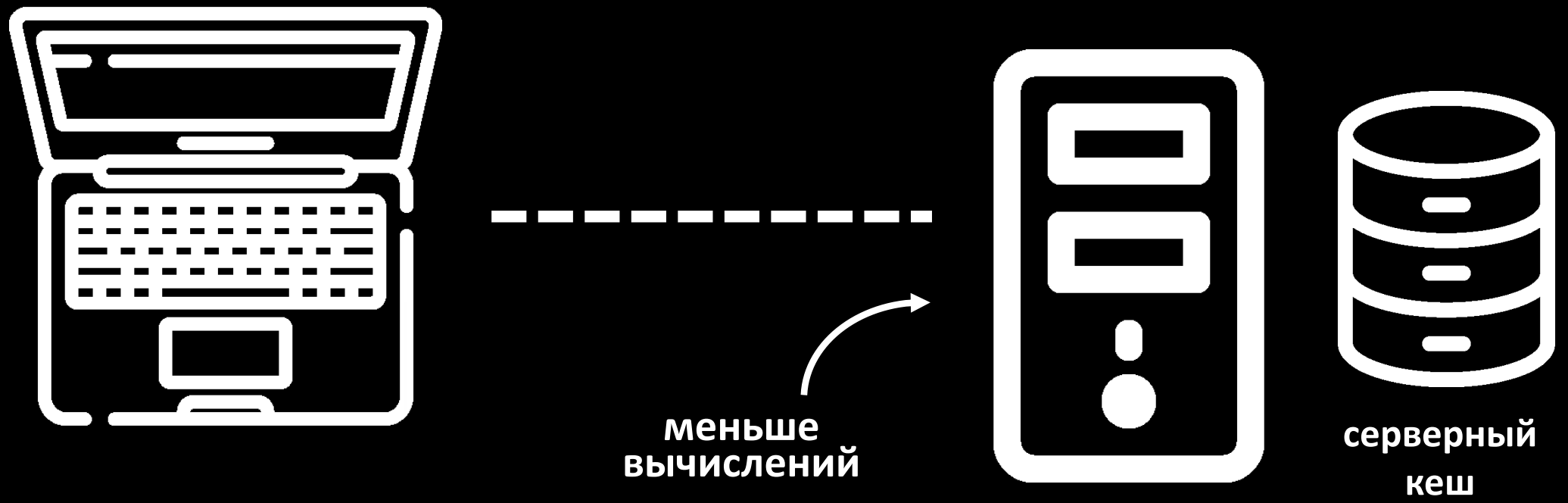
3 Производительность> ускоряем _



3 Производительность> ускоряем _



3 Производительность> ускоряем _

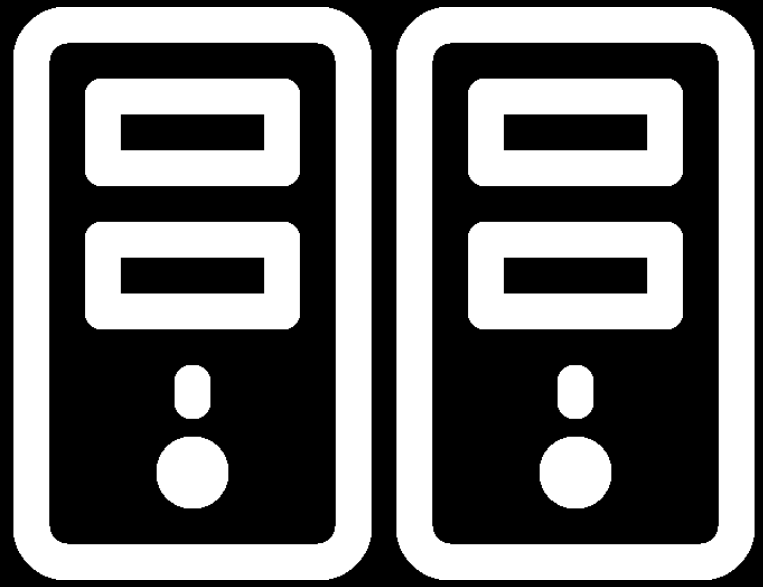


3 Производительность> ускоряем _

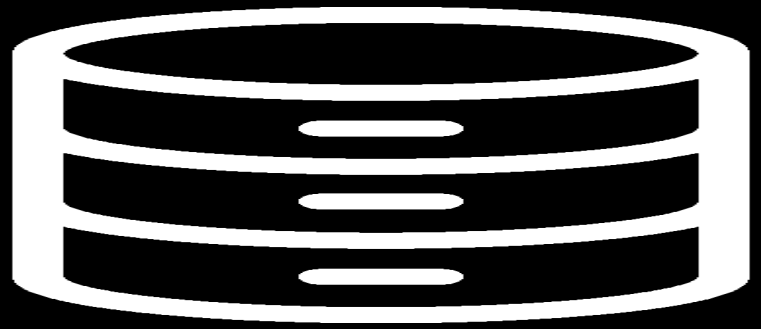
масштабирование

3 Производительность> ускоряем _

много ресурсов



меньше вычислений



общий кластерный кеш

3 Производительность> не забыть

- Предзагрузка кеша
- Обновление данных кеша:
  a. Least recently used (LRU)
  b. Most recently used(MRU)
  c. First-in, first-out (FIFO)
  d. Last-in, first-out (LIFO)
  e. Явное удаление данных

СПАСИБО

dmitriy.dzyuba@mts.ru

helloconf@mts.ru