

# Онлайн-образование

# Не забыть включить запись!



# Меня хорошо видно && слышно?

Ставьте + , если все хорошо  
Напишите в чат, если есть проблемы

# Правила вебинара



Активно участвуем



Задаем вопрос в чат или голосом



Off-topic обсуждаем в Slack #канал группы или #general



Вопросы вижу в чате, могу ответить не сразу

# System Design



Юшкевич Виталий

yushkevichv@gmail.com

# Обо мне

- Выпускник ВКА им. А.Ф. Можайского и ИТМО
- Профессиональный опыт с 2009 года.
- Принимал участие в работе над проектами и продуктами разного масштаба и сложности.
- Эксперт в области разработки и масштабирования ecommerce проектов.
- Текущая область интересов - платформенная разработка в высоконагруженных системах
- Выступаю с докладами на темы разработки и управления.
- Хобби:
  - Музыка, играю на ударных инструментах

This is experiment!

# Цели вебинара | После занятия вы сможете

1

Объяснить, что такое System Design

2

Рассказать основные  
этапы System Design

3

Пройти собеседование  
по System Design

# 01. Что такое System design

# Что такое System design

- [wiki]
- Systems design is the process of defining the architecture, product design, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering.

# Что такое *System design*

- Любую "абстрактную" задачу можно решить. Почти всегда - более чем одним способом. Нахождение в условиях ограничений сбалансированных решений - это про системный дизайн.

# Что такое System design

- Подходы к System Design:

# Что такое System design

- Подходы к System Design:
  - Композиция
    - Смотрим на задачу и думаем, как ее собрать, обладая нашими знаниями и кругозором (архитектура, технологии, базы данных и тп)

# Что такое System design

- Подходы к System Design:
  - Композиция
    - Смотрим на задачу и думаем, как ее собрать, обладая нашими знаниями и кругозором (архитектура, технологии, базы данных и тп)
  - Декомпозиция
    - Смотрим на готовую систему; разбираем ее на компоненты; смотрим из чего они состоят и как устроены

# Что такое System design

- Подходы к System Design:
  - Композиция
    - Смотрим на задачу и думаем, как ее собрать, обладая нашими знаниями и кругозором (архитектура, технологии, базы данных и тп)
  - Декомпозиция
    - Смотрим на готовую систему; разбираем ее на компоненты; смотрим из чего они состоят и как устроены
- \* сегодня рассмотрим композицию, на следующих лекциях - декомпозицию
- \*\* на собеседованиях и в жизни чаще будет требоваться композиция

## 02. Примерный план System Design

# План проведения System Design

- План описывает общую последовательность действий
- Для выполнения разбора / проектирования системы всегда есть временные ограничения.
- В случае с собеседованием, как правило, вы ограничены сильнее

# План проведения System Design

- 1. Выяснение требований и граничных условий

# План проведения System Design

- 1. Выяснение требований и граничных условий
  - Важно понимать задачу
  - Всю систему, как правило, невозможно описать в заданное время
  - Открытые вопросы не имеют единственного верного ответа, поэтому выяснение двусмысленностей вначале является критическим.

# План проведения System Design

- 2. Оценки (расчеты)

# План проведения System Design

- 2. Оценки (расчеты)
  - Трафик
  - Хранилище
  - Память
  - Пропускная способность
  - ...
- Возможные доп вопросы:
  - Количество активных пользователей в день
  - Количество новых пользователей в месяц
  - RPS / HPS

# План проведения System Design

- 3. API дизайн системы (модель данных)

# План проведения System Design

- 3. API дизайн системы (модель данных)
  - Разработка API интерфейса (минимум концептуально)
  - Модель данных
    - Отдаем
    - Получаем
  - Потоки данных
    - С какими компонентами / системами связаны и как?
  - На что можем повлиять?

# План проведения System Design

- 4. Дизайн БД

# План проведения System Design

- 4. Дизайн БД
  - Один из важнейших пунктов
  - Эффективность решения во многом зависит от хранения
  - Что нужно: БД NoSQL или SQL?
  - Схема данных

# План проведения System Design

- 5. High-level design (потоки данных между компонентами системы)

# План проведения System Design

- 5. High-level design (потоки данных между компонентами системы)
  - Состав компонентов
  - Как взаимодействуют друг с другом
  - \* часто описывается в виде диаграмм

# План проведения System Design

- 6. Детализированный дизайн
  -

# План проведения System Design

- 6. Детализированный дизайн
  - Итерационные проходы
  - Как правило, детализируют лишь определенные компоненты

# План проведения System Design

- 7. Узкие места

# План проведения System Design

- 7. Узкие места
  - Есть ли в вашей системе единая точка отказа?
  - Что будет, если мы потеряем несколько серверов?
  - Достаточное ли количество процессов (инстансов) сервисов у нас запущено на каждом из серверов?
  - Будет ли система работать также стабильно, если нагрузка увеличится? А если она увеличится резко?

# План проведения System Design

- 8. Мониторинг

# План проведения System Design

Подитог

# План проведения System Design

- 1. Выяснение требований и граничных условий
- 2. Оценки (расчеты)
- 3. API дизайн системы (модель данных)
- 4. Дизайн БД
- 5. High-level design (потоки данных между компонентами системы)
- 6. Детализированный дизайн
- 7. Узкие места
- 8. Мониторинг

# 03. Композиция на примере

# Designing Instagram

# Композиция на примере

- Что такое Instagram?

# Композиция на примере

- Требования

# Композиция на примере

- Требования
  - Функциональные:
    - Пользователь может загрузить / скачать / посмотреть изображение
    - Пользователь может искать по определенным атрибутам
    - Пользователь может подписываться на других
    - Система должна формировать и показывать ленту

# Композиция на примере

- Требования
  - Не функциональные:
    - Наш сервис должен обладать хороший доступностью
    - Допустимое время генерации ленты 200ms
    - Консистентность может быть нарушена (в угоду доступности) - если пользователь не видит какое-то время определенное фото - не критично
    - Система должна быть высоконадежной - ни одно фото или видео не должно пропасть

# Композиция на примере

- Требования
  - Не функциональные:
    - Наш сервис должен обладать хороший доступностью
    - Допустимое время генерации ленты 200ms
    - Консистентность может быть нарушена (в угоду доступности) - если пользователь не видит какое-то время определенное фото - не критично
    - Система должна быть высоконадежной - ни одно фото или видео не должно пропасть

# Композиция на примере

- Требования
  - Не рассматриваем (пока):
    - Добавление тегов
    - Комментирование
    - ...

# Композиция на примере

- Общие идеи по дизайну

# Композиция на примере

- Общие идеи по дизайну
  - Read-heavy
  - Может быть много изображений от одного пользователя
  - Ожидается быстрое время ответа при просмотре фото
  - 100% reliable (гарантия не потери данных)

# Композиция на примере

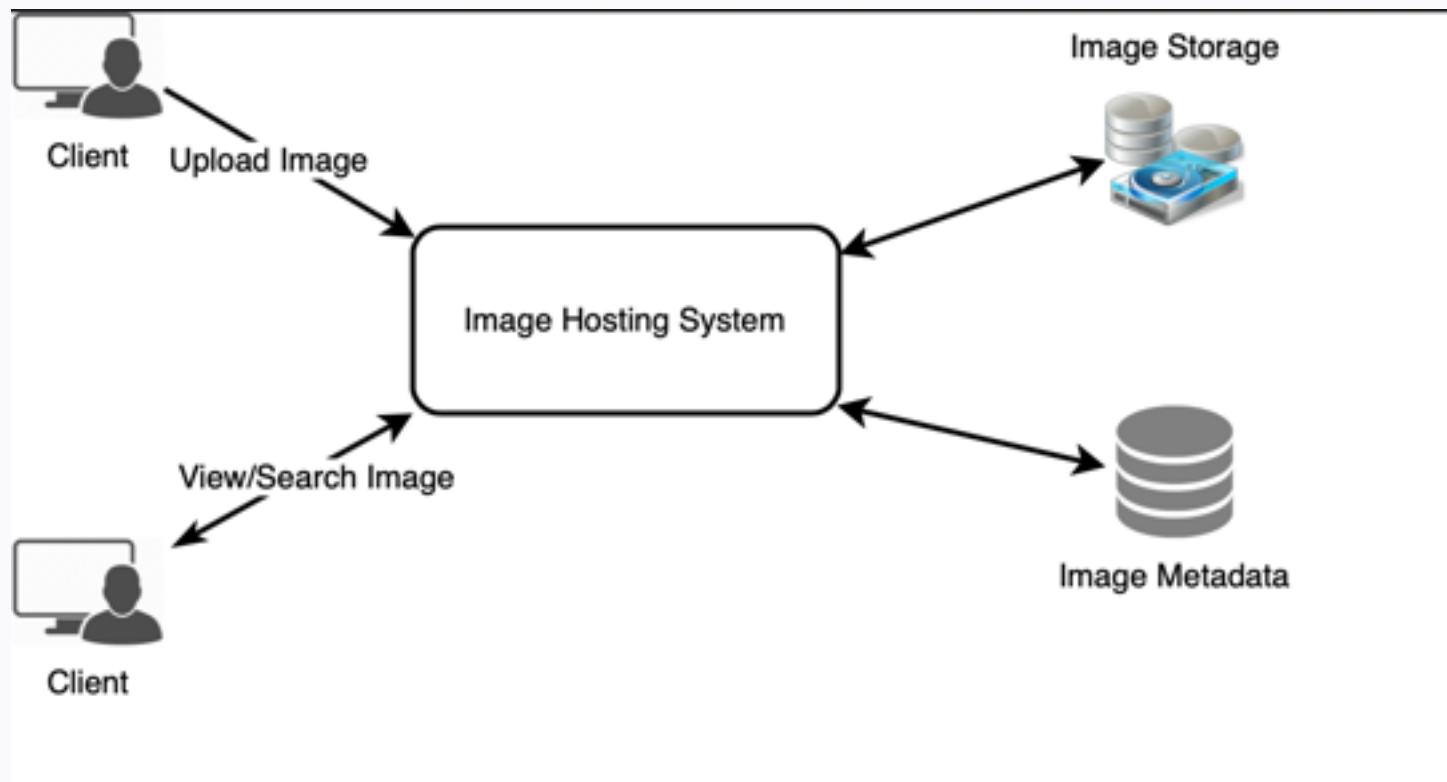
- Оценки

# Композиция на примере

- Оценки
  - 500M пользователей всего, 1M активных ежедневно
  - 2M новых фото в день, 23 новых фото / сек
  - Средний размер => 200KB
- Объем хранилища для фото 1 дня:
  - $2M * 200KB \Rightarrow 400\text{ GB}$
- Общий объем за 10 лет:
  - $400\text{GB} * 365 * 10 \sim 1425\text{ TB}$

# Композиция на примере

- High-level design



# Композиция на примере

- Схема БД

- index on (PhotoID, CreationDate)
- RDBMS storage

Photo	
PK	<u>PhotoID: int</u>
	UserID: int
	PhotoPath: varchar(256)
	PhotoLatitude: int
	PhotoLongitude: int
	UserLatitude: int
	UserLongitude: int
	CreationDate: datetime

User	
PK	<u>UserID: int</u>
	Name: varchar(20)
	Email: varchar(32)
	DateOfBirth: datetime
	CreationDate: datetime
	LastLogin: datetime

UserFollow	
PK	<u>UserID1: int</u> <u>UserID2: int</u>

# Композиция на примере

- Хранение файлов
  - Распределенное файловое хранилище (HDFS / S3)

# Композиция на примере

- Проблемы выбранной схемы:
  -

# Композиция на примере

- Проблемы выбранной схемы:
  - Масштабирование

# Композиция на примере

- Альтернативный дизайн БД

# Композиция на примере

- Альтернативный дизайн БД
  - Распределенное NoSQL key-value
    - Например Cassandra
  - Недостатки
    - Хранить больше данных для эффективной работы
    - Поддержка сложнее

# Композиция на примере

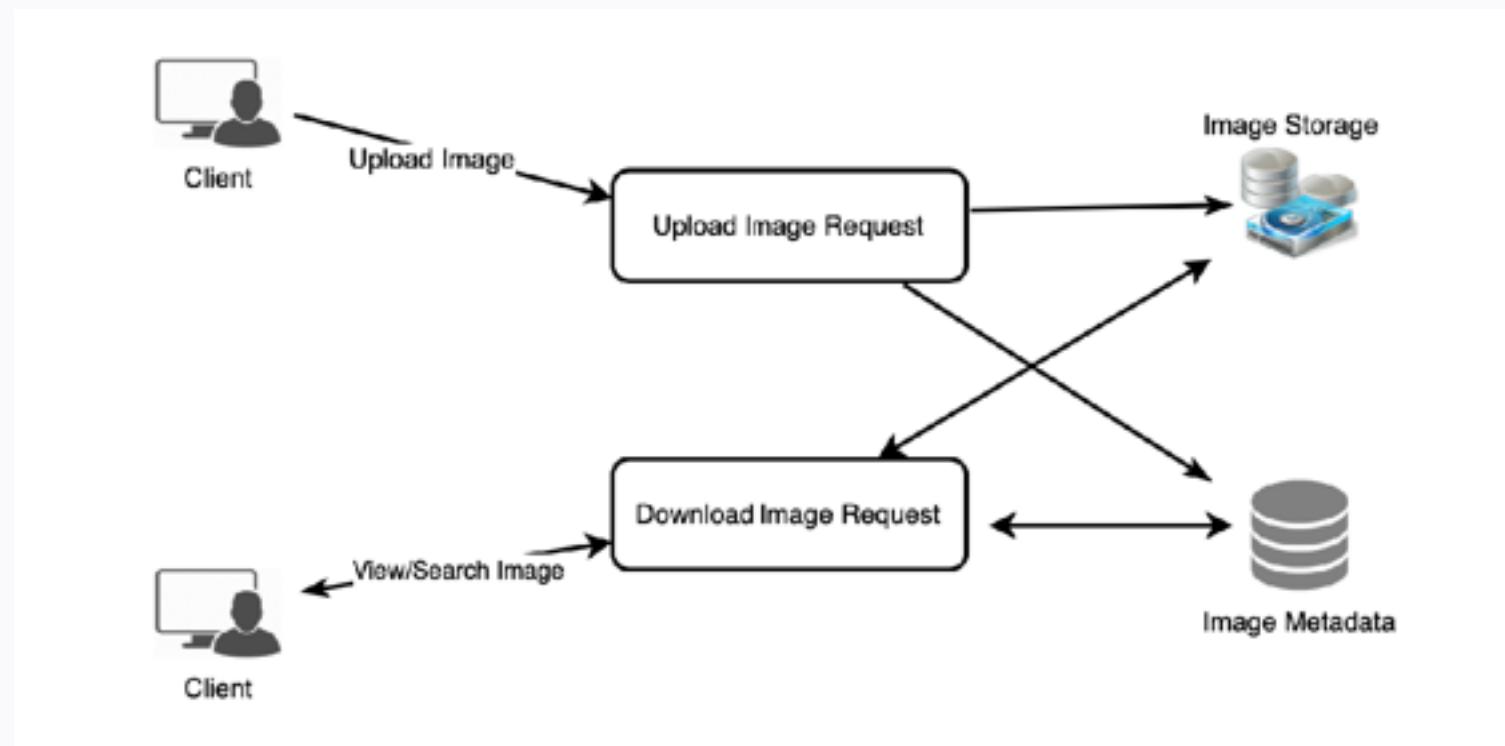
- Оценка объема БД
  - User:
    - UserID (4 bytes) + Name (20 bytes) + Email (32 bytes) + DateOfBirth (4 bytes) + LastLogin (4 bytes) = 68 bytes
    - 500M Users \* 68 ~ 32Gb
  - Photo:
    - Photoid (4 bytes) + UserID (4 bytes) + PhotoPath (256 bytes) + PhotoLatitude (4 bytes) + PhotoLongitude(4 bytes) + UserLatitude (4 bytes) + UserLongitude (4 bytes) + CreationDate (4 bytes) = 284 bytes
    - New uploaded: 2M \* 284 ~ 0.5Gb / per day
    - 10 лет => 1.88TB
  - UserFollow
    - 500 million users \* 500 followers \* 8 bytes ~ 1.82TB
  - Весь объем за 10 лет:
    - 32GB + 1.88TB + 1.82TB ~ 3.7TB

# Композиция на примере

- Компонентный дизайн
-

# Композиция на примере

- Компонентный дизайн
  - Разделяем read / write нагрузку

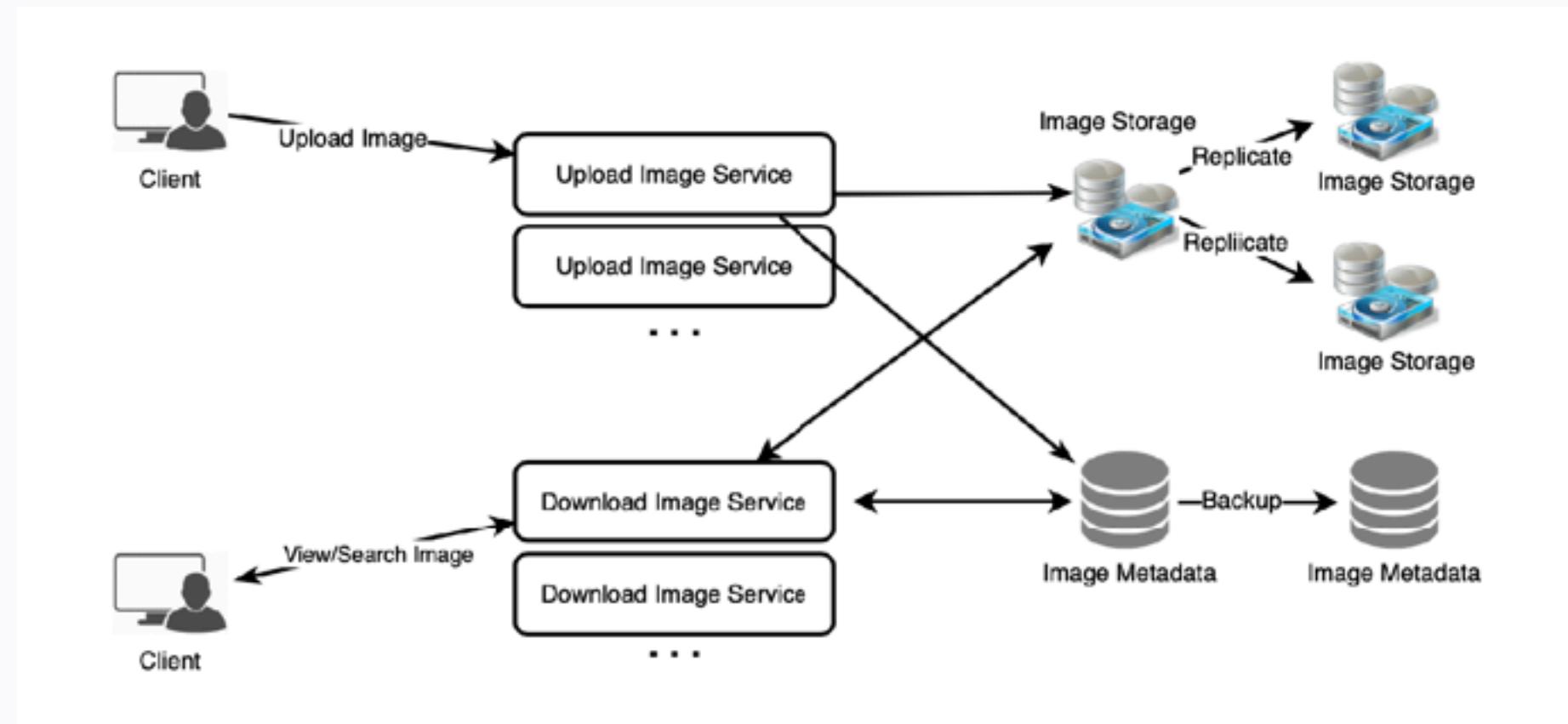


# Композиция на примере

- Надежность и резервирование
-

# Композиция на примере

- Надежность и резервирование



# Композиция на примере

- Масштабирование
-

# Композиция на примере

- Масштабирование. БД
  - Шардинг
  - Партицирование

# Композиция на примере

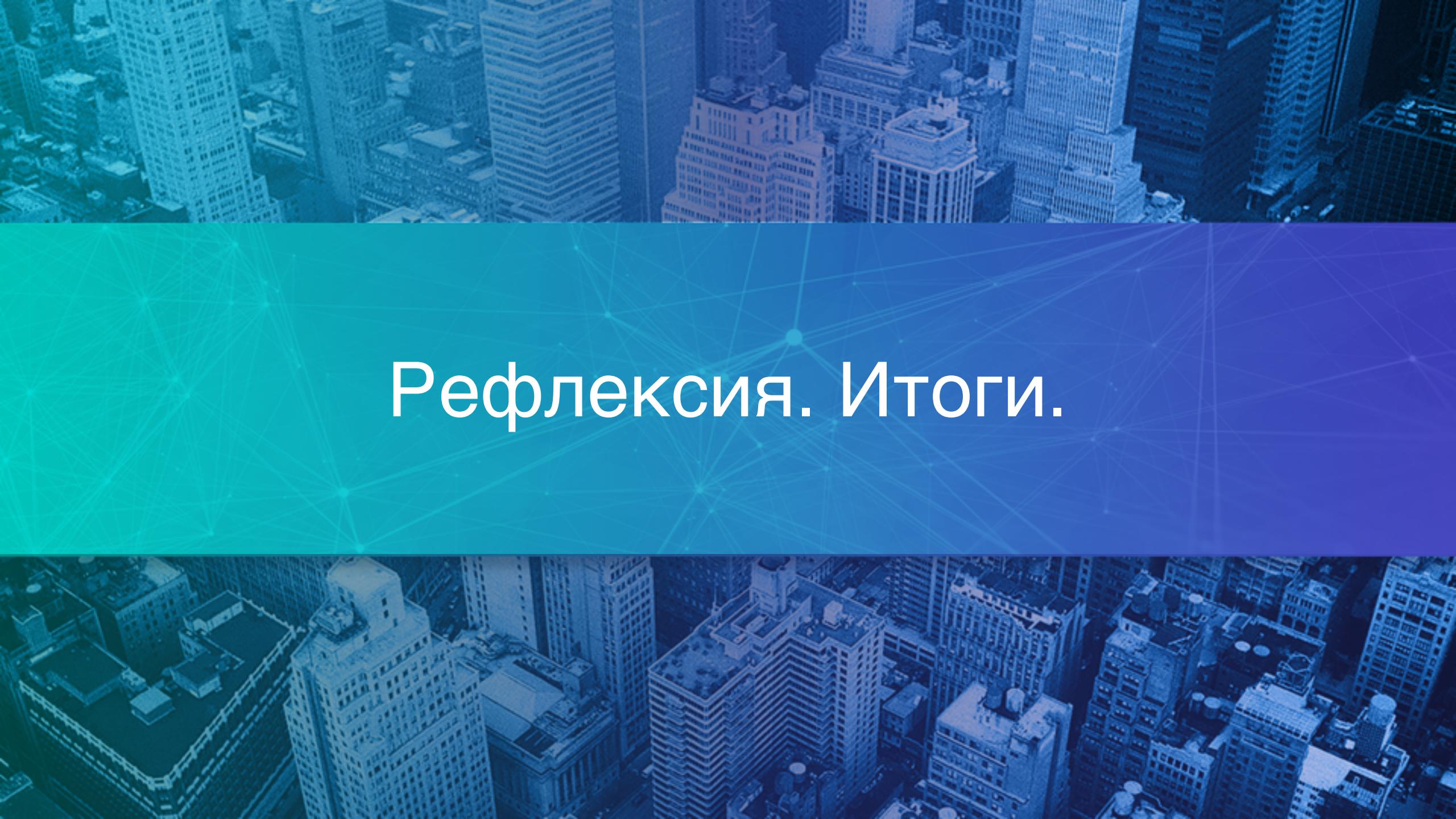
- Масштабирование. Формирование ленты
  - Предварительная генерация
  - Очереди

# Композиция на примере

- Кэширование и балансировка
  -

# Композиция на примере

- Кэширование и балансировка
  - Фото кэш сервера
  - CDN
  - Кэширование мета-данных (LRU)
  -



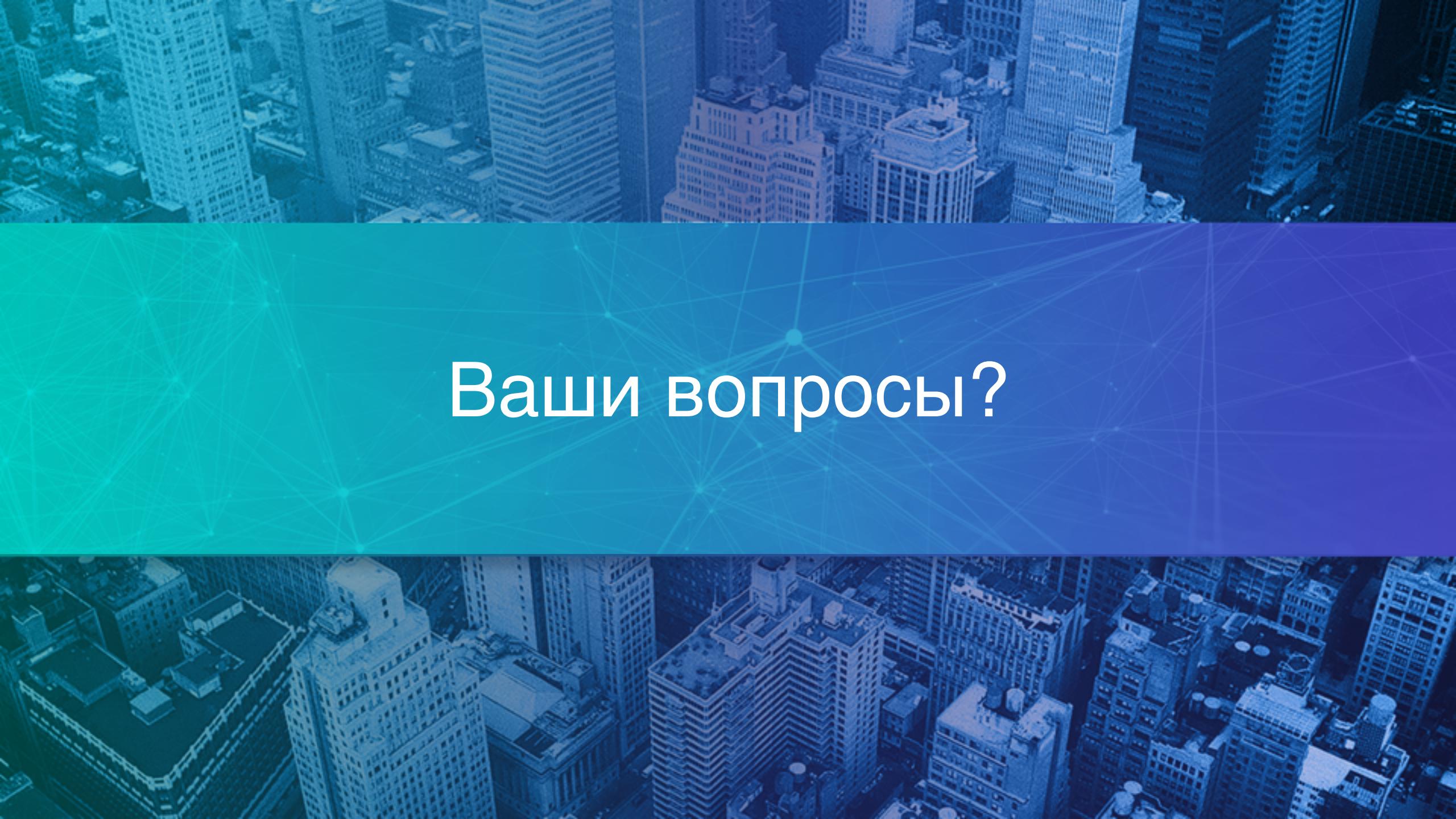
# Рефлексия. Итоги.

# Рефлексия

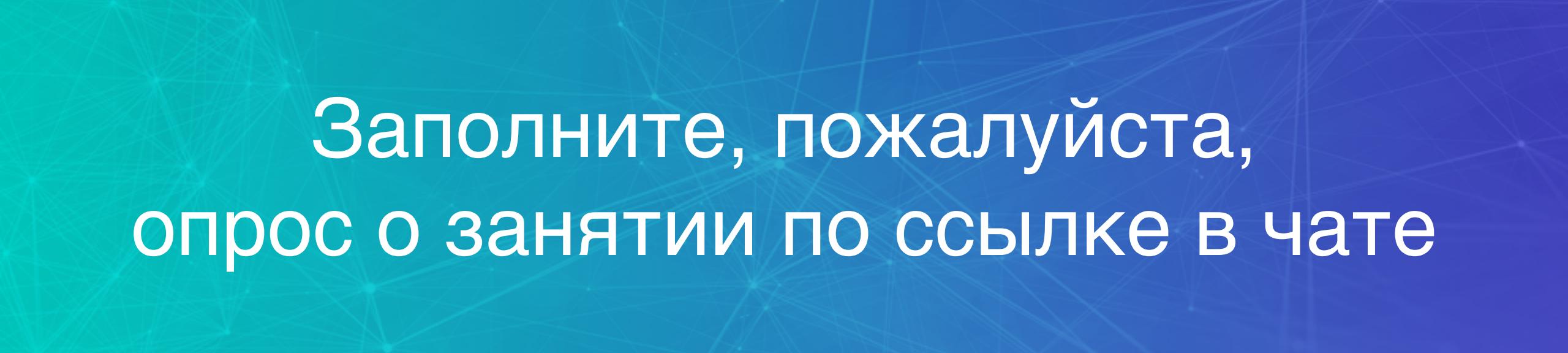
- О чём мы говорили сегодня?

# Рефлексия

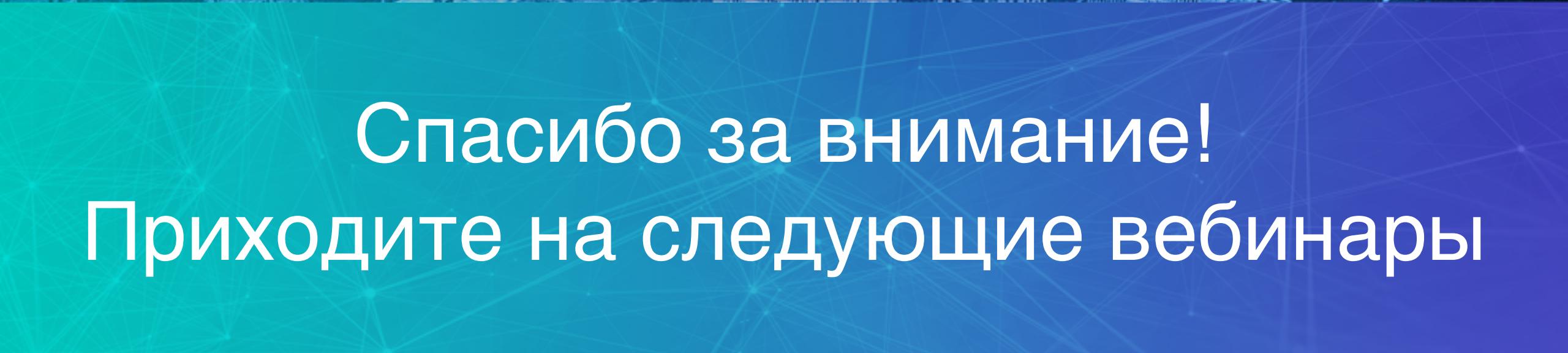
- Напишите, пожалуйста, свое впечатление о вебинаре.
- Отметьте 3 пункта, которые вам запомнились с вебинара.
- Что вы будете применять в работе из сегодняшнего вебинара?



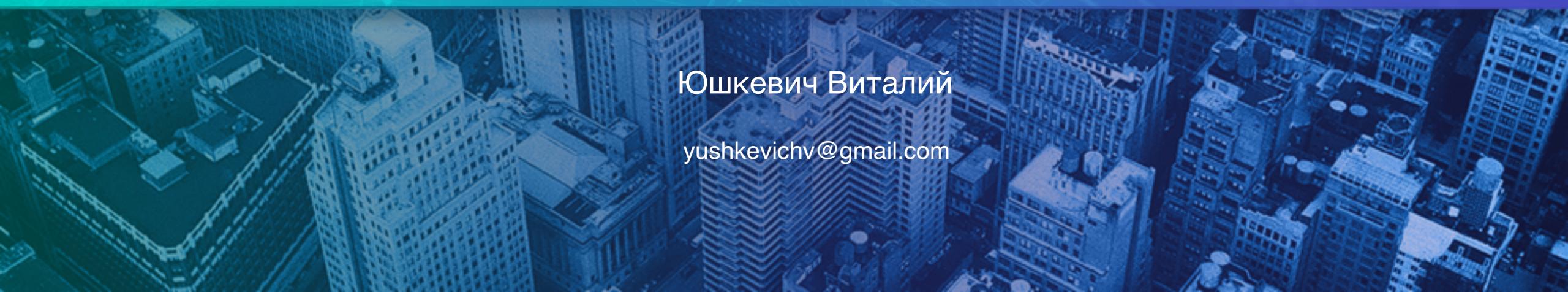
Ваши вопросы?



Заполните, пожалуйста,  
опрос о занятии по ссылке в чате



Спасибо за внимание!  
Приходите на следующие вебинары



Юшкевич Виталий

yushkevichv@gmail.com