Искусство System Design. Как построить распределенную систему и пройти собеседование

Владимир Маслов

X5 Tech

t.me/senior_unicorn

vlmr.maslov@gmail.com

JPoint X5 Tech

Обо мне

- ➔ Руководитель команды разработки в X5 Tech
- ➔ Занимался бэкендами в роли ведущего разработчика и мл. архитектора (ex-Сбер, ex-Ланит, ex-ДИТ)
- ➔ Делал стартап про образовательные игры
- ➔ 100+ постов на Habr [@HotWaterMusic]

➔ Автор канала [Сеньор Юникорн]

Сегодня в программе

- ➔ Суть System Design: зачем и как проектировать популярные системы
- ➔ Популярные ошибки на собеседованиях по дизайну систем
- ➔ Как изучить принципы проектирования систем, подготовки к собеседованию и с пользой применять их на практике

Что такое
System Design

Когда пришло письмо счастья

Hello from Google :)

Hi Vladimir,

I’m a software engineering recruiter for core Google Cloud products that support ~1 billion of our users daily through planetary scale distributed systems.

I wanted to check in and discuss positions in our Google Cloud hub in Poland with you, and hear about what you've been working on recently at Sberbank.

Do you have some time over the coming week for a quick call to discuss?

Hope you're keeping well and safe, and looking forward to hearing from you soon :)

AWS - Career opportunities

Hi Vladimir,

Liliia from AWS here. How are you doing right now?

I currently look after Software Engineer hiring for AWS across Ireland (Dublin), Italy (Asti, Cagliari), Spain (Madrid), Germany (Berlin, Aachen, Walldorf), Poland (Gdansk), and the UK (Cambridge). Basically, all of our teams are hiring at the moment! During the process you would get to know the different teams and it's totally down to you to choose which location you'd like to join at the end.

I wanted to keep this discussion open for now and just quickly check if this could actually in general be interesting for you. Just let me know your first feedback here :)

Как же хочется в бигтех (Google/Amazon/Netflix/…)

blind

Lessons learned: Amazon, Facebook, Microsoft, and Google

Google · xms2p

3d 273 Comments

After working for at least a few years in 4 huge software companies, here is the main lesson from each experience.

Amazon: Prioritize or burn out. I learned to prioritize and say no. Each time a manager came with more work, I would show the list of what I was doing and ask for the position of that new item. Always made it clear where was the line of the things I could do.

Microsoft: Tech is not everything. Worked on promising projects that failed miserably in the market, and some didn't even reach the market. Worked on other projects that I couldn't believe were sponsored, and those made it to the short list of products/services with more than $1bi/year revenue.

Facebook: Let the customer decide. Similar to the lesson from Microsoft, but more intense, experimental, and data-driven. The customer may not know what they want or need, and you also don't know. Put the options before the customer, do A/B testing, and look at the results without bias.

3,312 273

clck.ru/33vkDu

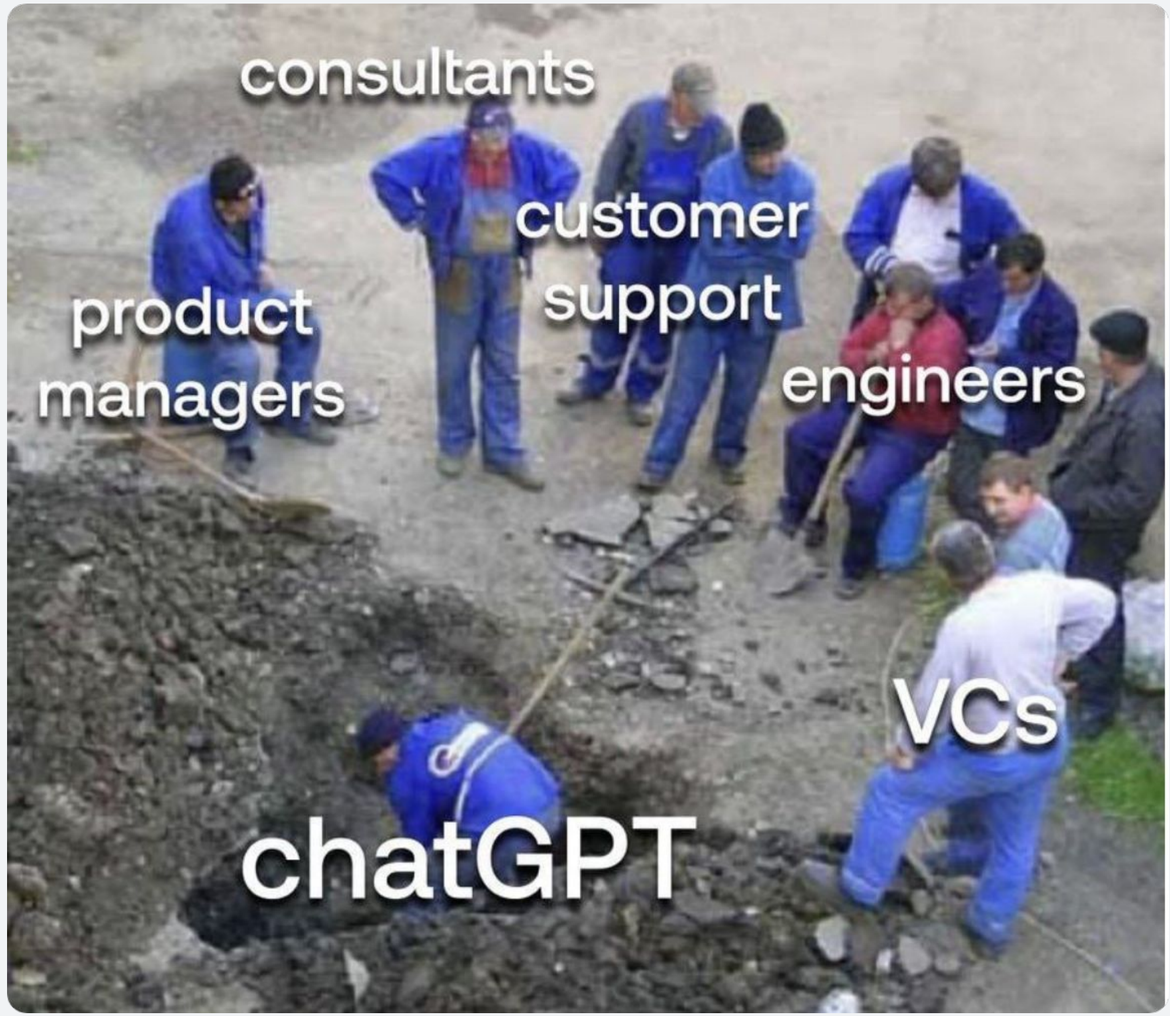
A Day in the Life of a Software Engineer

У самурая нет оффера, есть только собеседования

Увидел интересную вакансию

Увидел количество собеседований

ChatGPT приходит на помощь



Общая подготовка

Wednesday, March 12, 2008

Get that job at Google

bit.ly/3G9VrxW

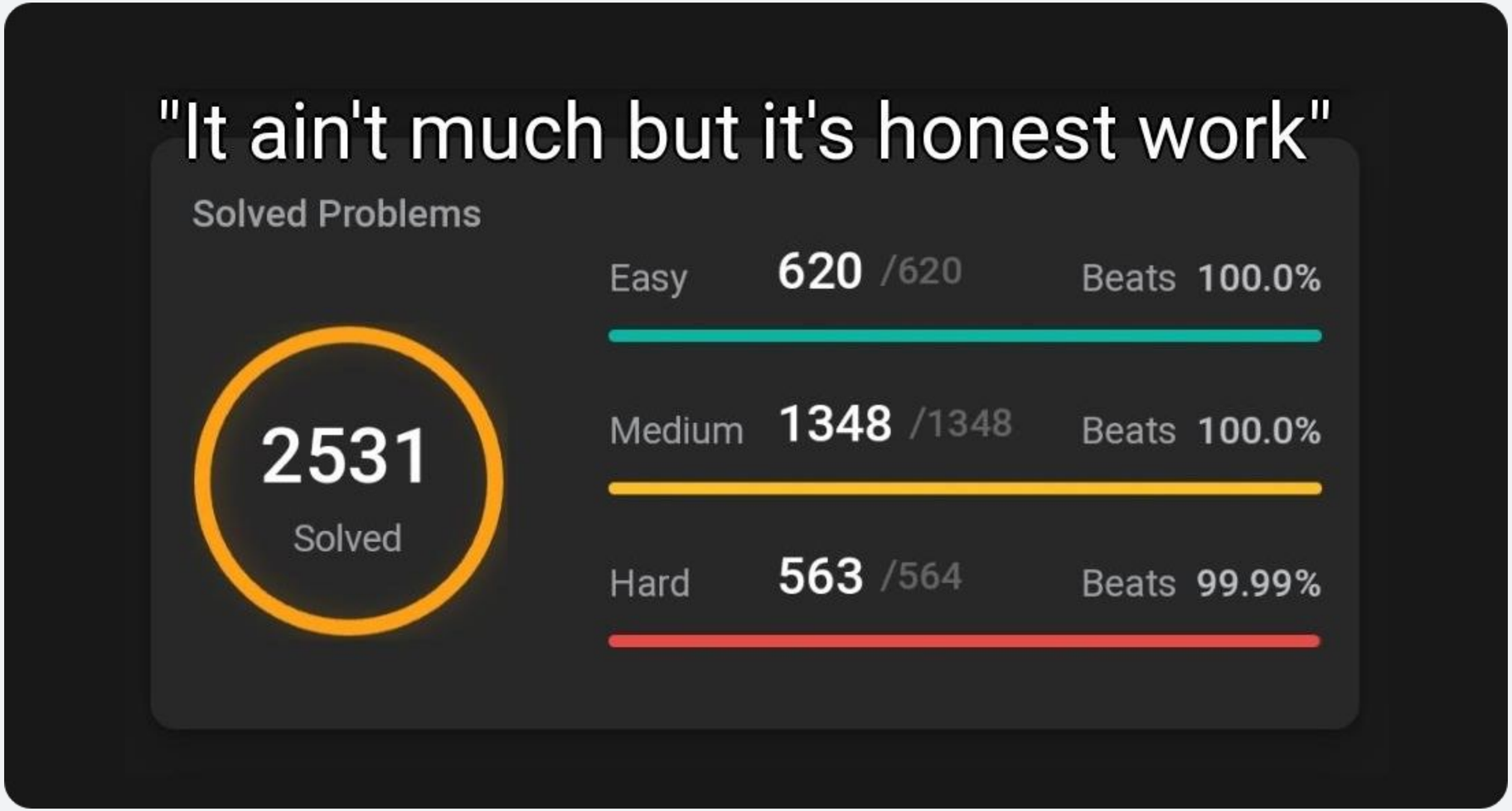
Preparing for the Systems Design and Coding Interview

clck.ru/33z8KF

Тренировки по алгоритмам 3.0

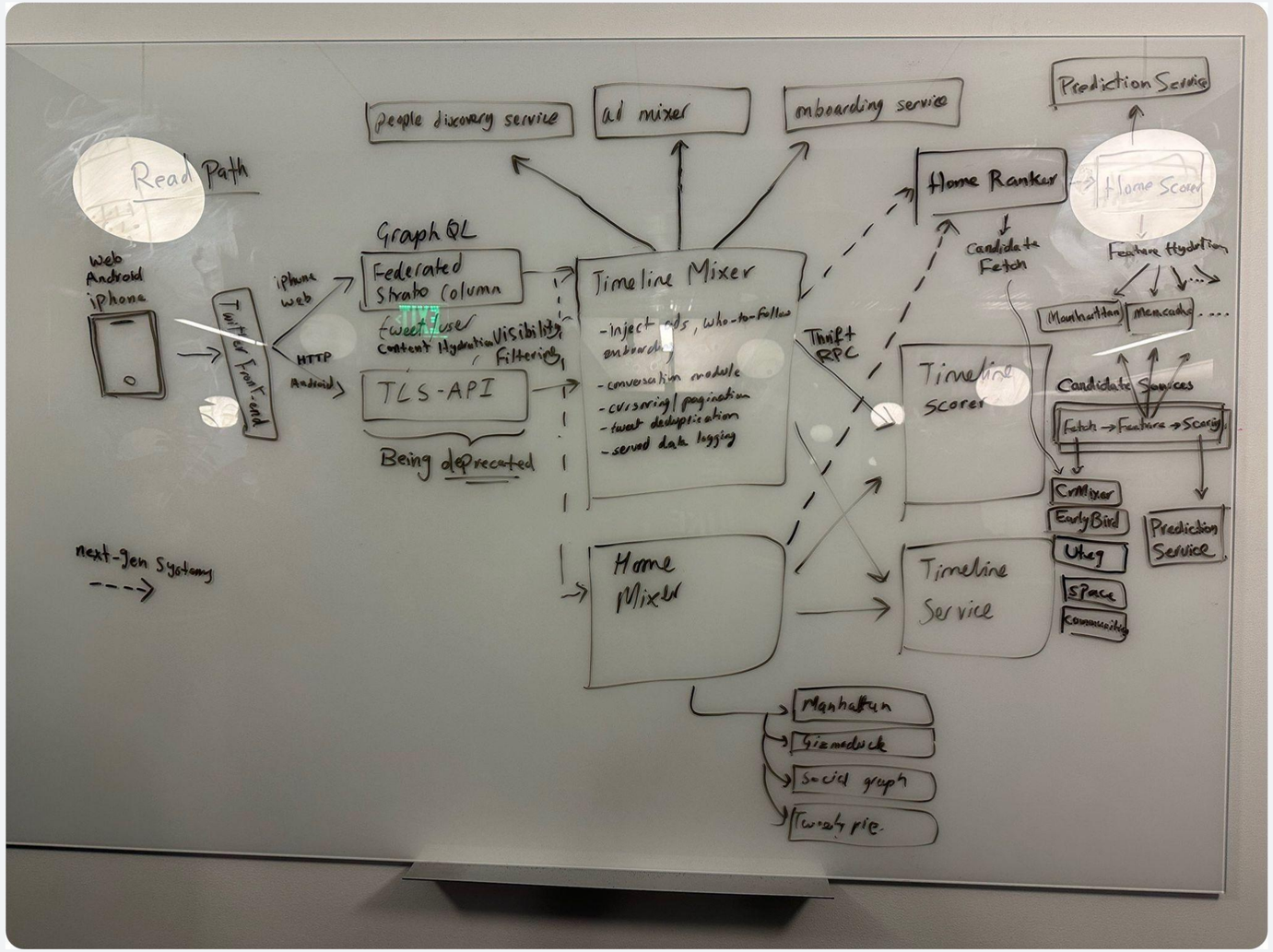
clck.ru/342bpJ

Уже неплохо, но нужно стараться больше

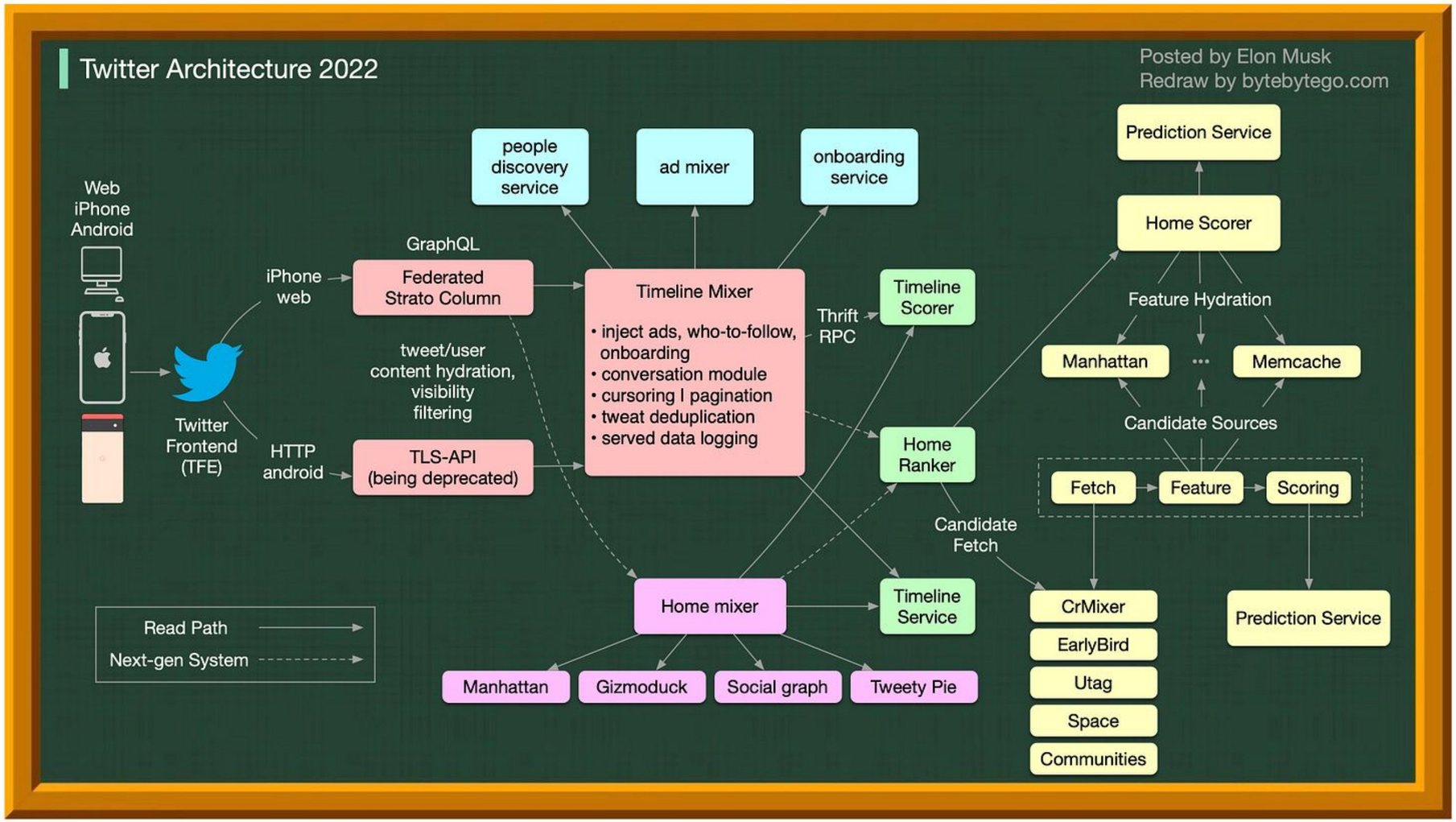


System Design когда вы видите его в первый раз

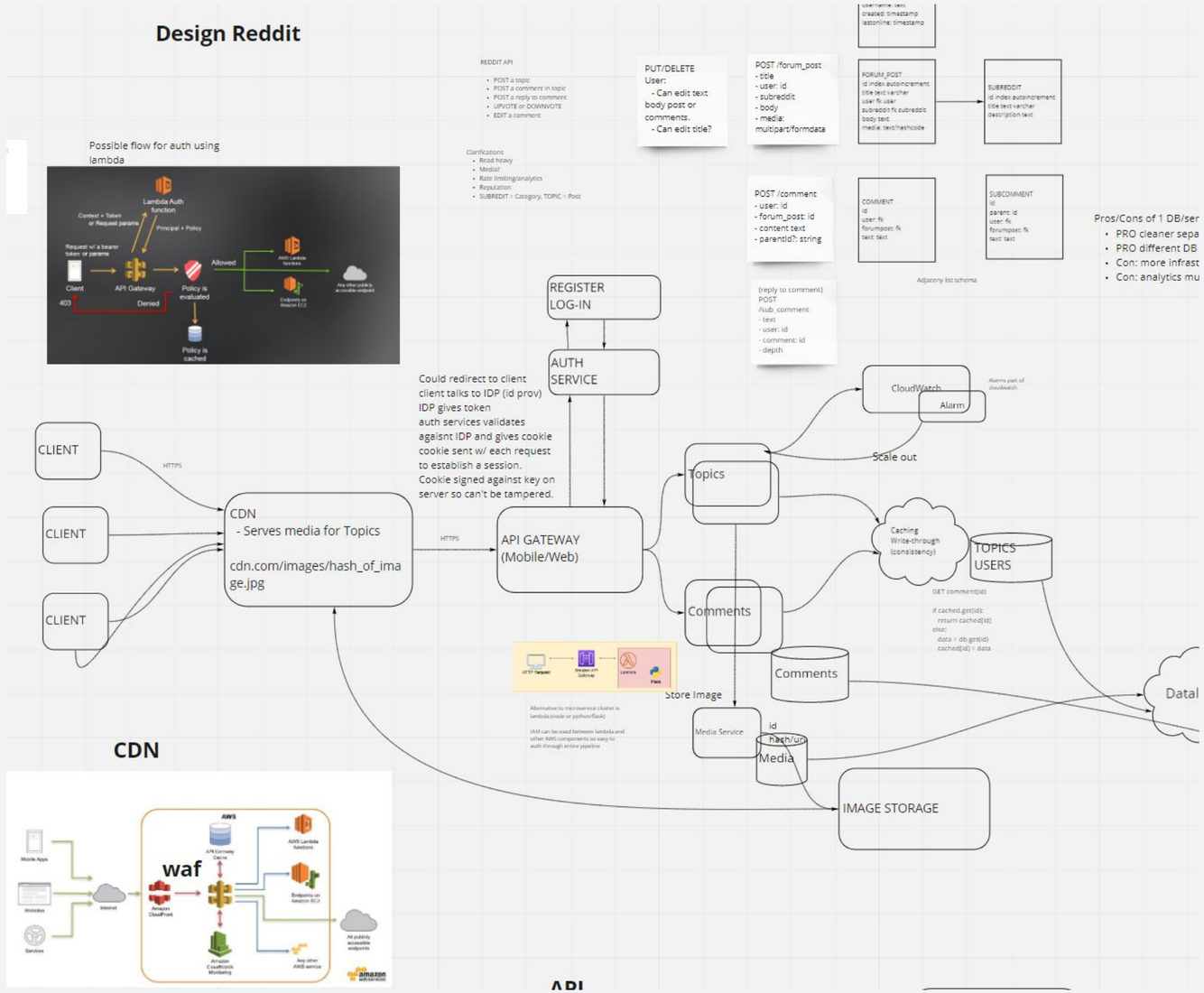
Как мы представляем себе System Design



Если вы вдруг не узнали, это была архитектура Twitter



На самом деле, придется сделать вот это

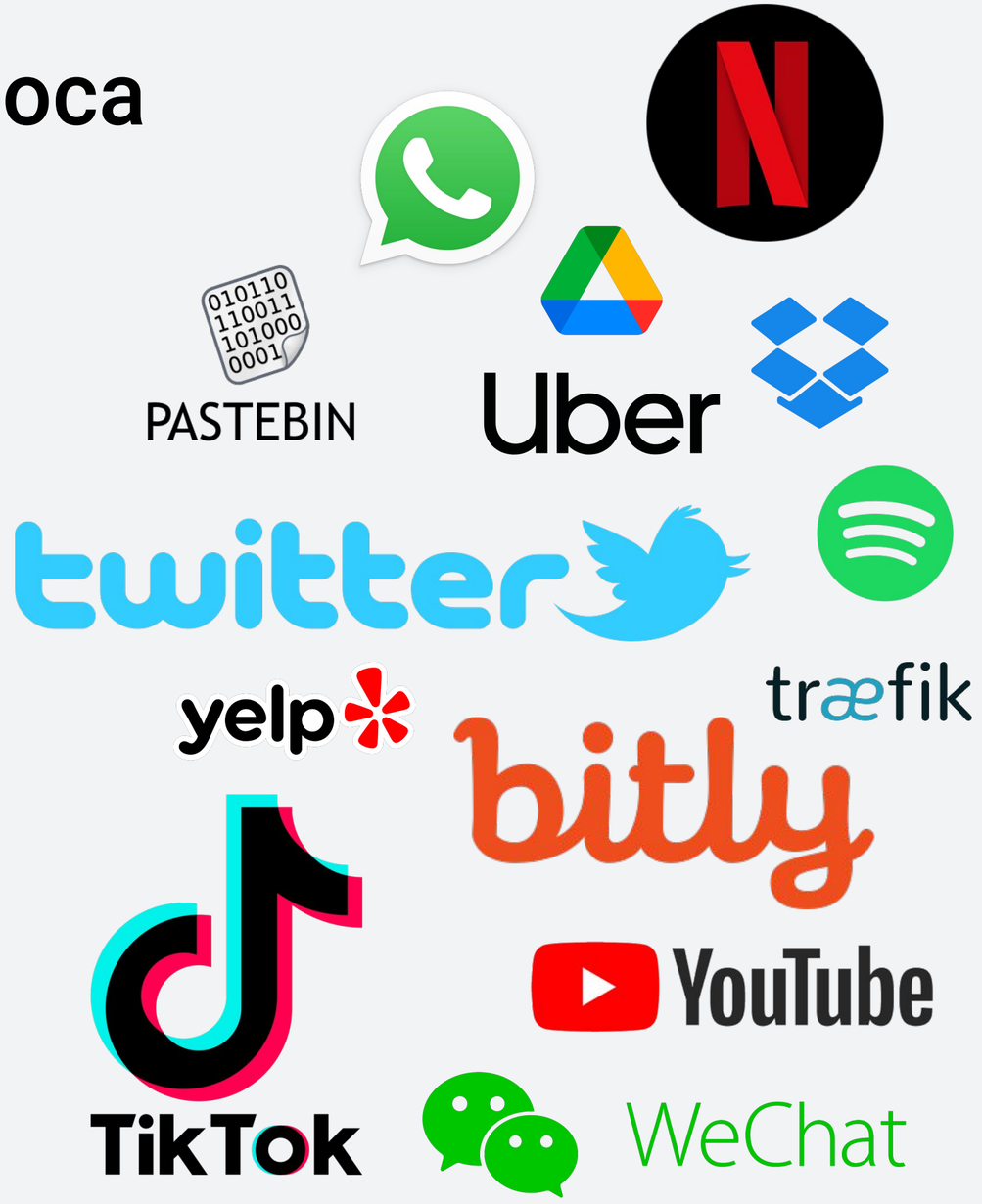


Каким бывает дизайн систем

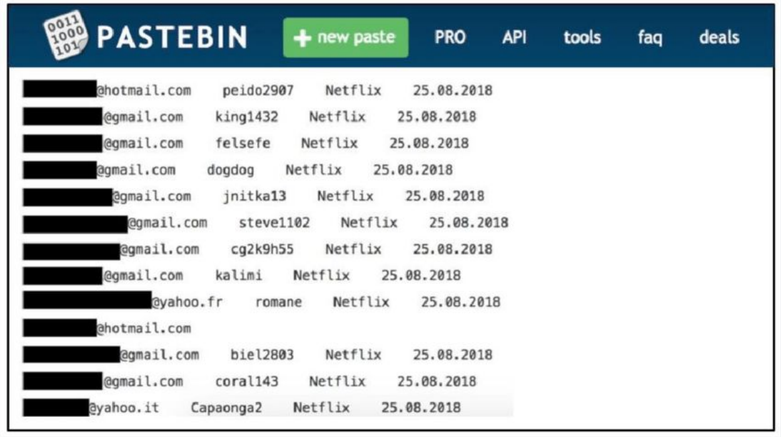
Типология стартовых условий

Всё начинается с простого вопроса

➔ “Давайте мы с вами задизайним систему…”

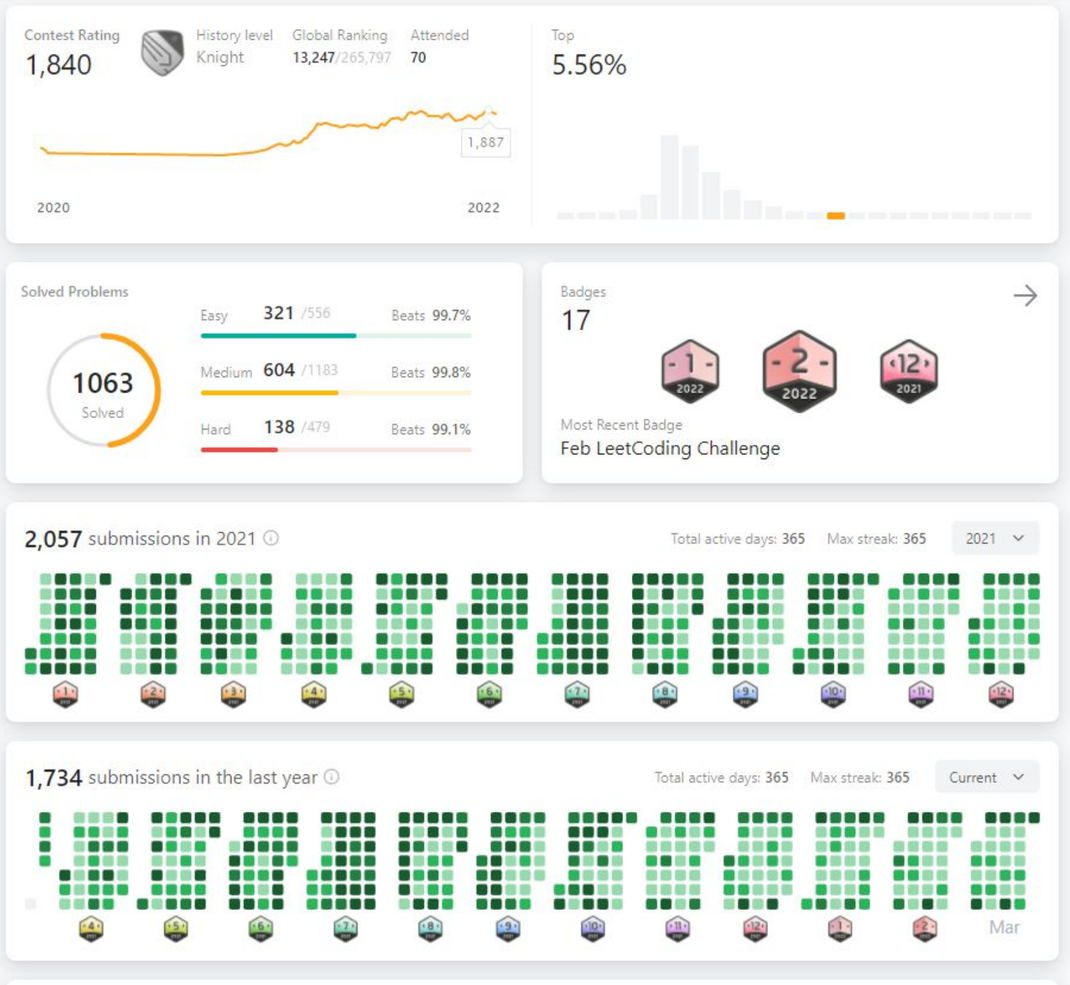


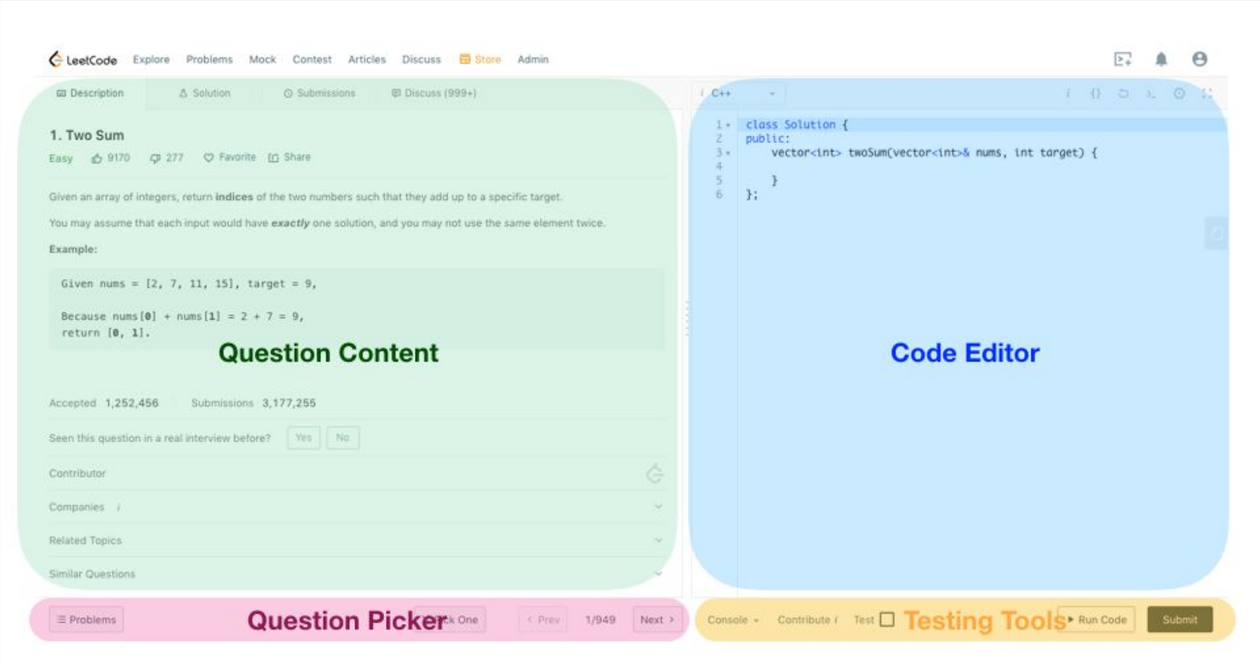
Классические системы



clck.ru/33vi2x

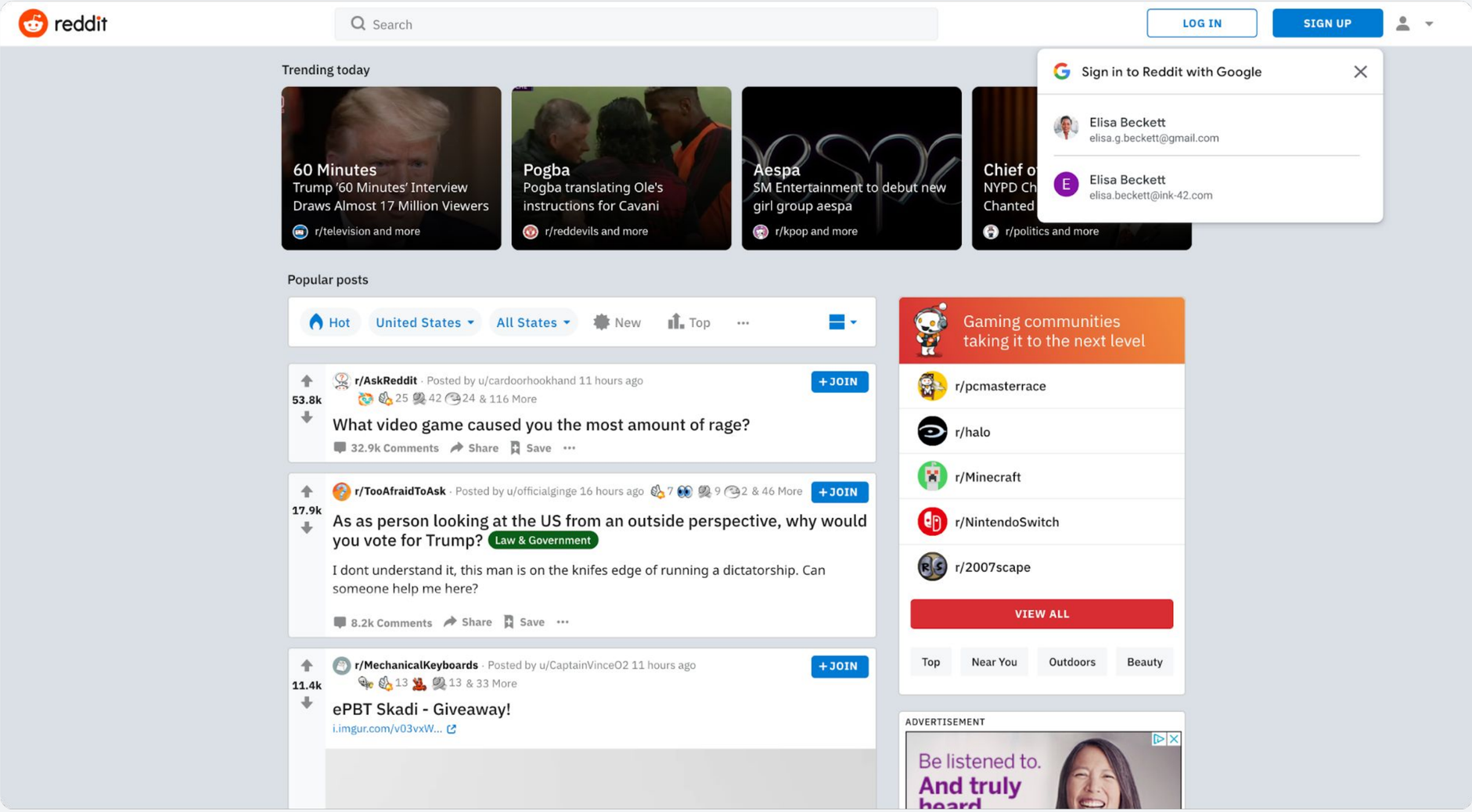
Более редкие варианты





LeetCode

Более редкие варианты



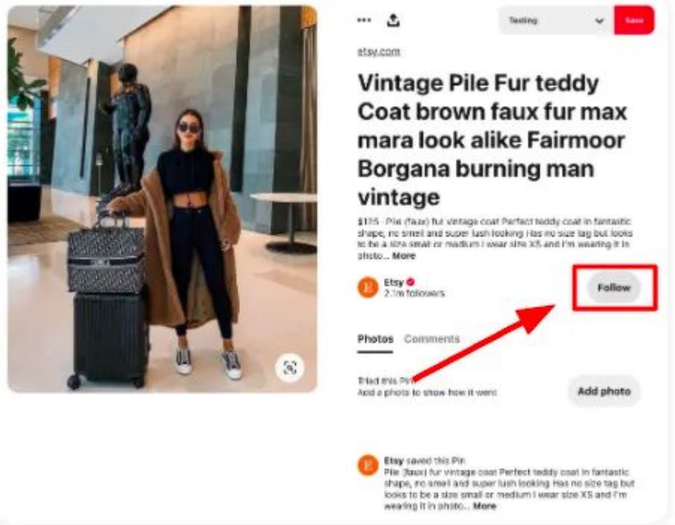
reddit

Новая фича

Нужно спроектировать новую фичу для существующей системы (обычно популярной).

Суть - примерно та же самая.

Q. Come up with a system design for a new feature where users can follow pins or users for updates.



Архитектура вашего текущего проекта

Кому надо проходить System Design

Лестница в небо

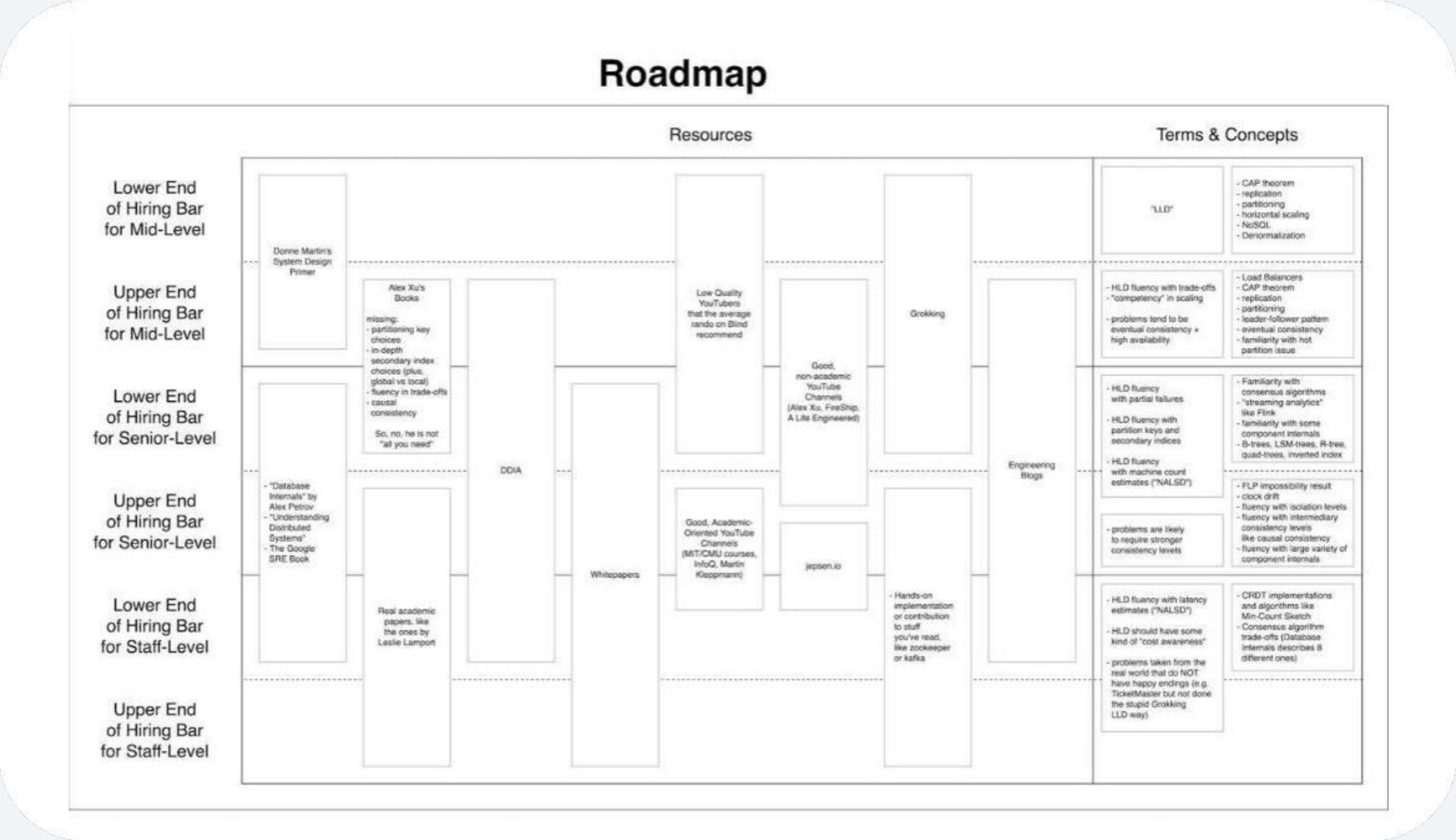
L3/L4 - Основные ожидания: знание алгоритмов, умение писать код

L5 - Способность пройти архитектурную секцию играет важную роль, вместе со скиллами коммуникации

L6 - Оценивается по возможности приносить импакт и брать на себя сложные задачи. Прошлый опыт, уровень ответственности также будут учтены

clck.ru/33wEDs

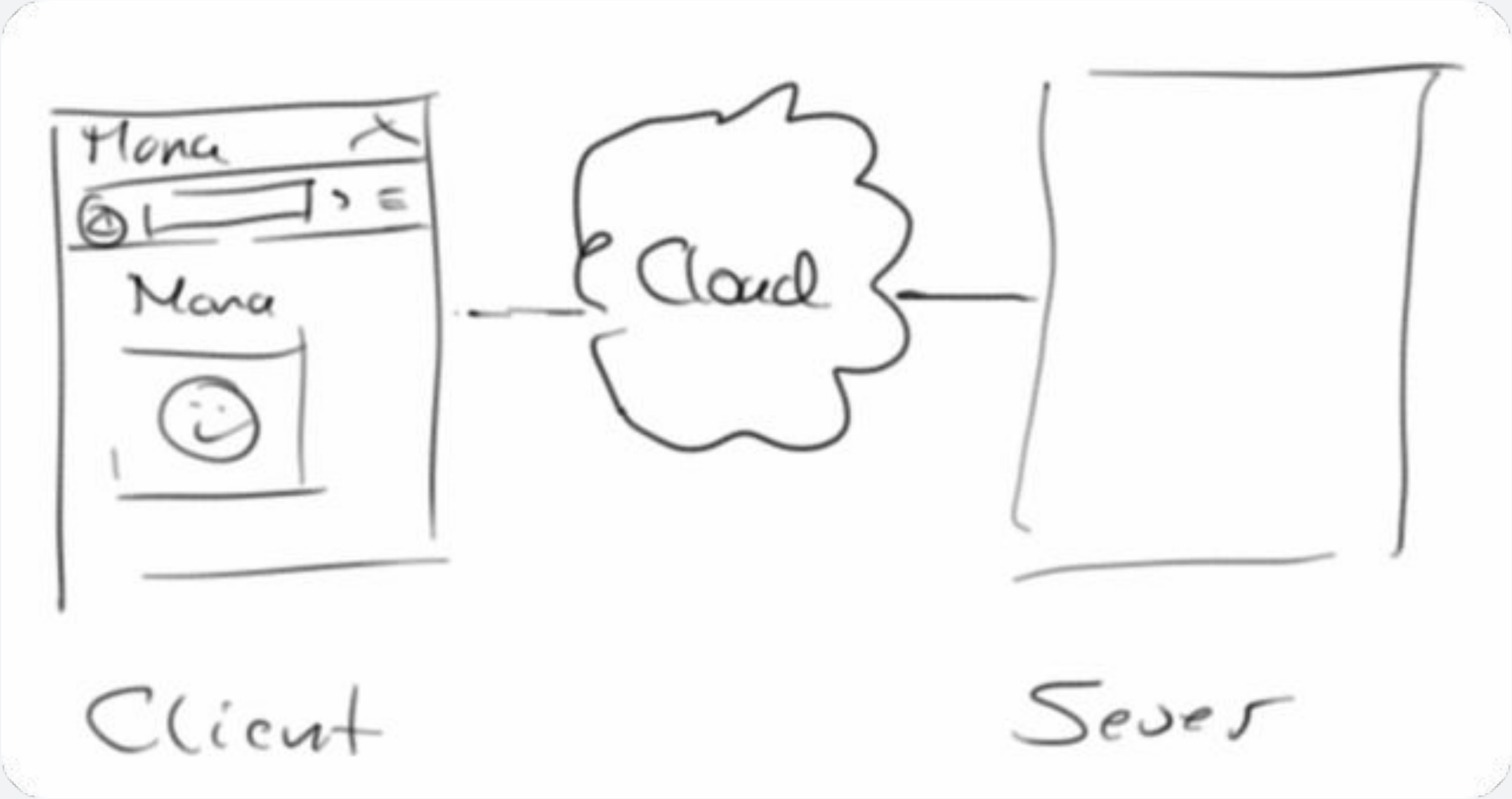
Каждое интервью уникально



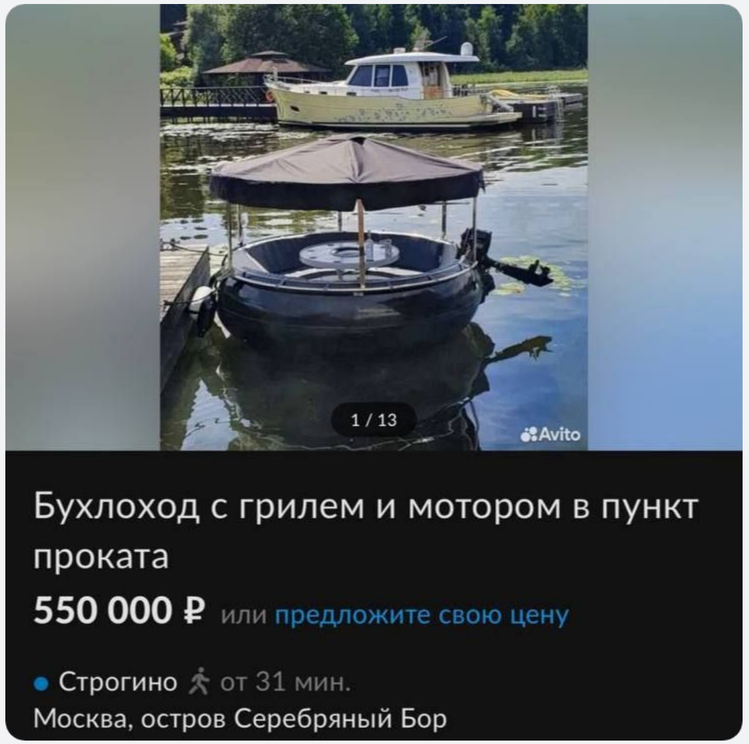
The Key Mistake Nearly Everyone Makes

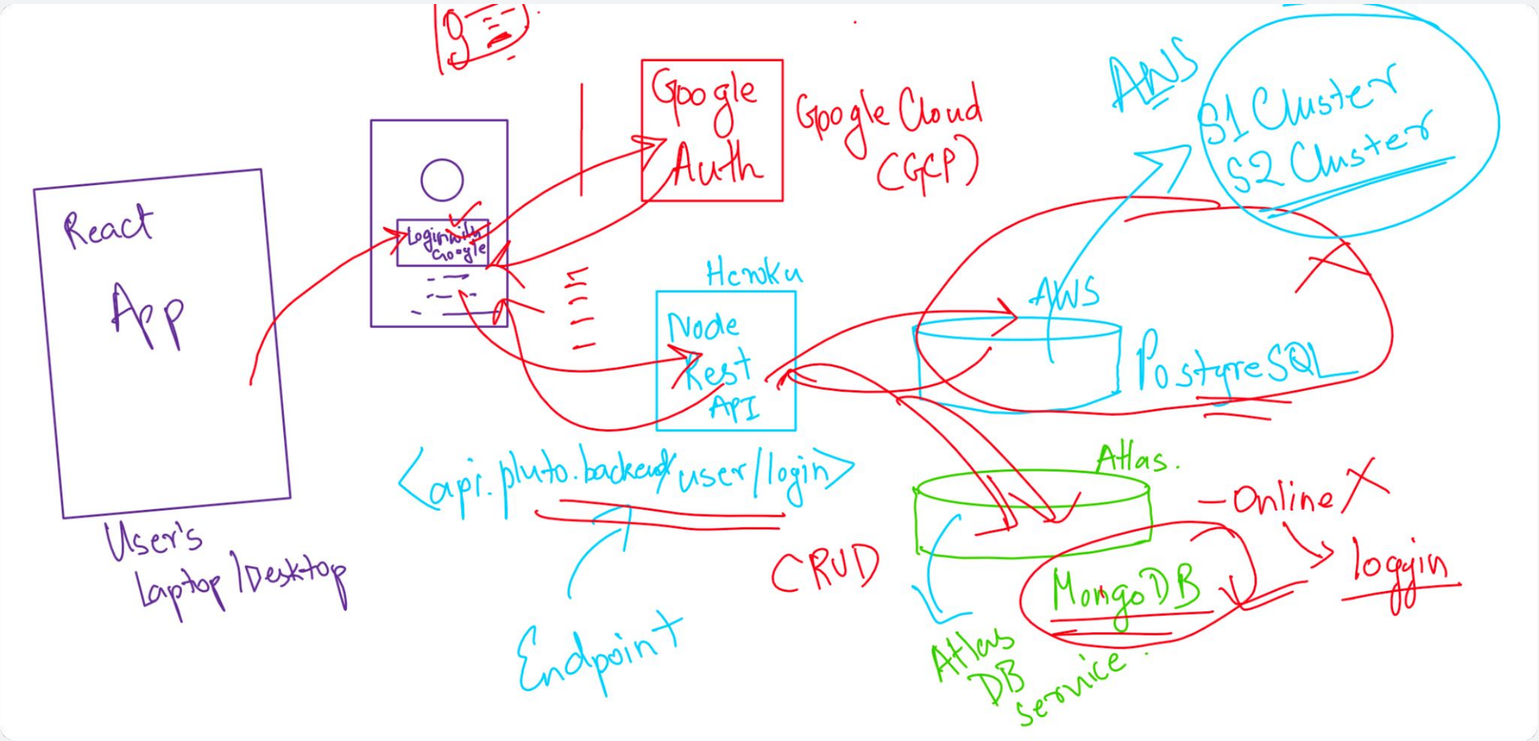
Ключевая ошибка всех новичков

Что мы нарисовали

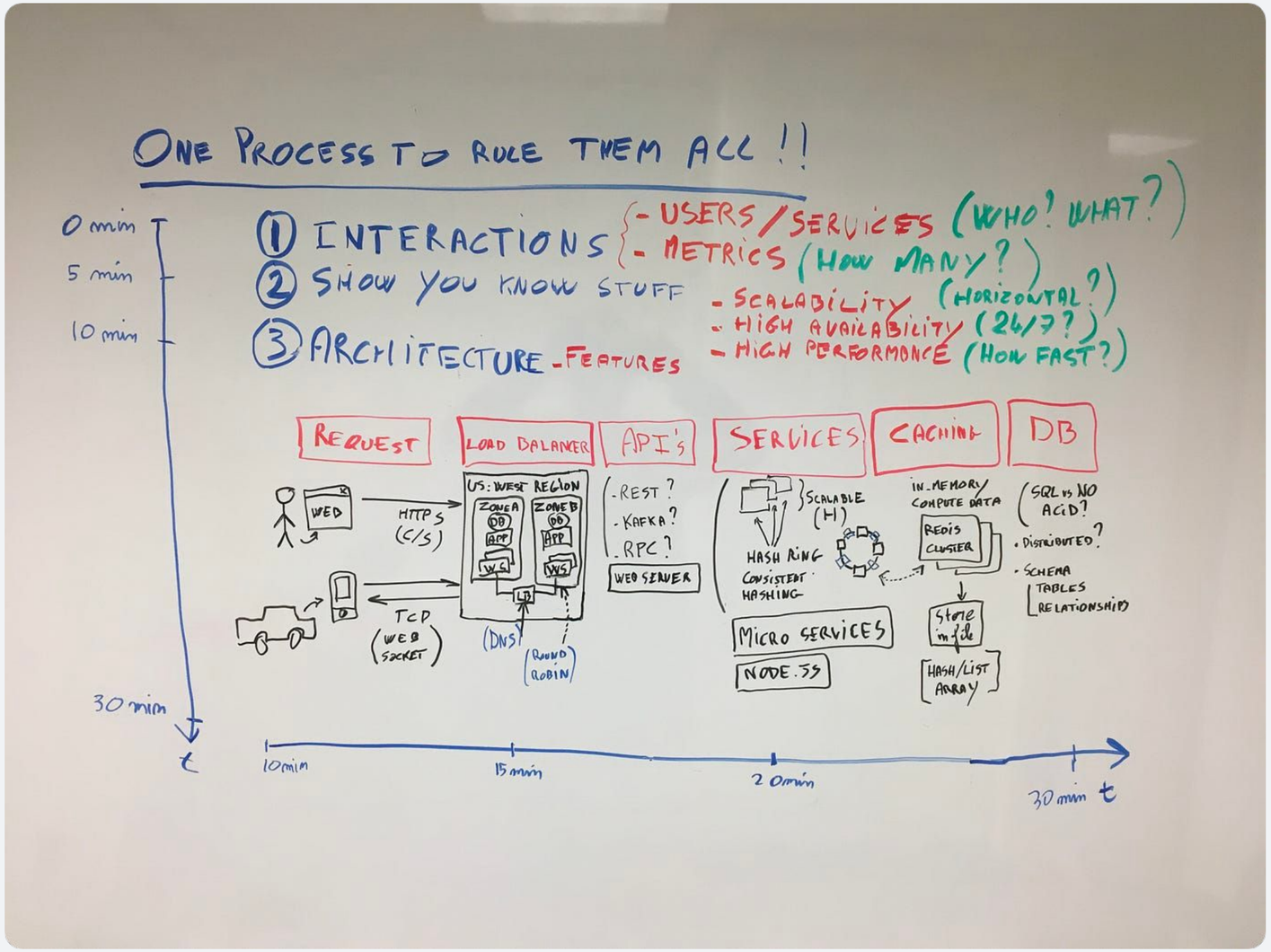


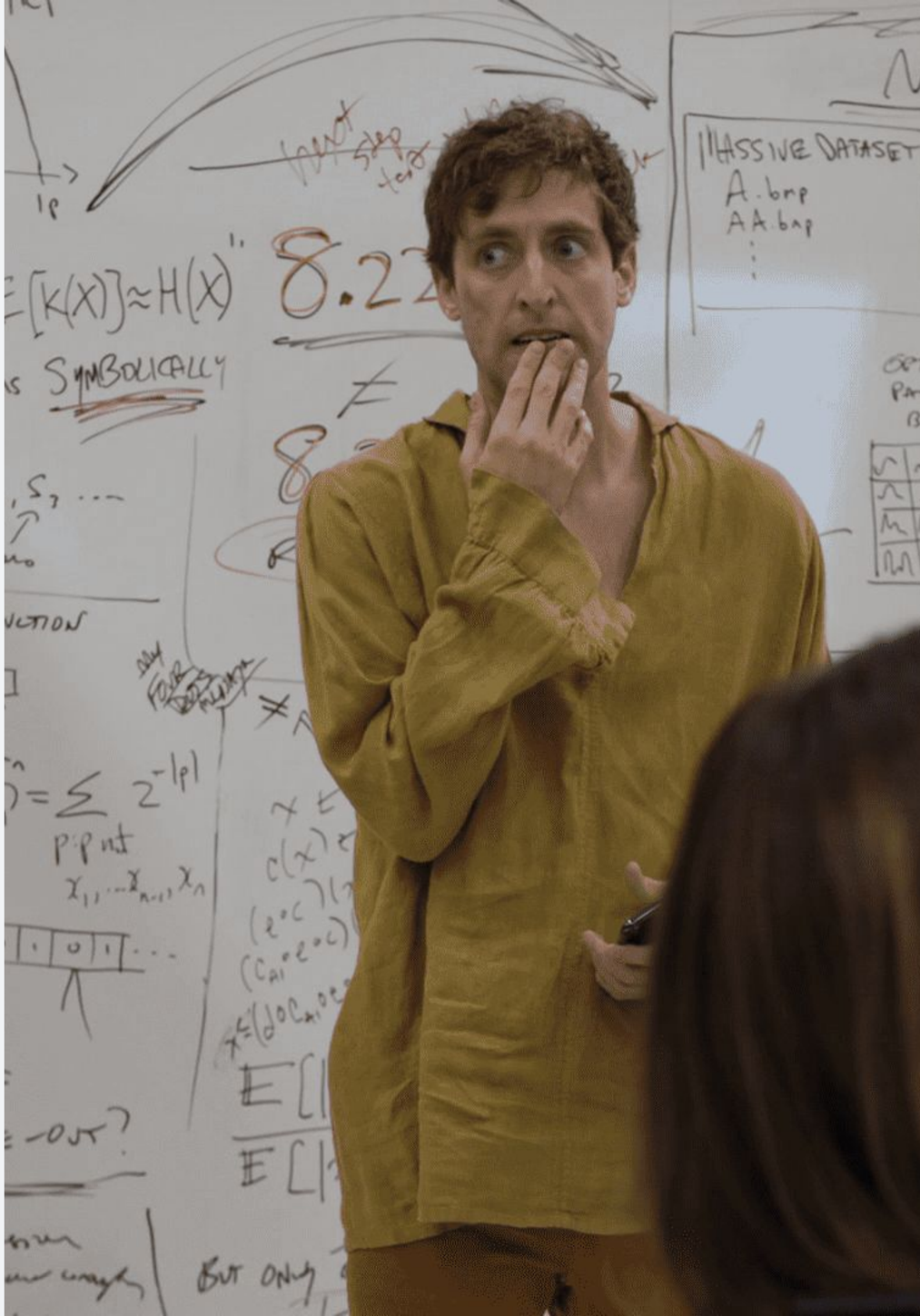
Что ожидал увидеть интервьюер





Finita La Commedia





Коммуникация превыше всего

- ➔ В компаниях MAANG (FAANG) интервьюер не хочет, чтобы вы сразу начали проектировать систему
- ➔ Интервьюер ожидает, что вы вступите с ним в диалог
- ➔ Коммуникация с интервьюером не менее важна для успеха, чем умение строить архитектуру и знания про распределенные системы

One thing you should avoid is "just memorizing" the approaches of the problems. That's far from the point. I made this mistake when I interviewed at Facebook, and was asked to build a part of Instagram. I had done this exercise, and so I just drew out a complicated system. I never talked about constraints or tradeoffs with my interviewer. In fact, I never had a two-way conversation.

A systems design interview is as much about communication with the interviewer as it is about your systems and architecture knowledge. This is why, while the book will help fill gaps you might have on how large systems are built, it won't substitute you collaborating with someone in designing a system.

clck.ru/33z8KF

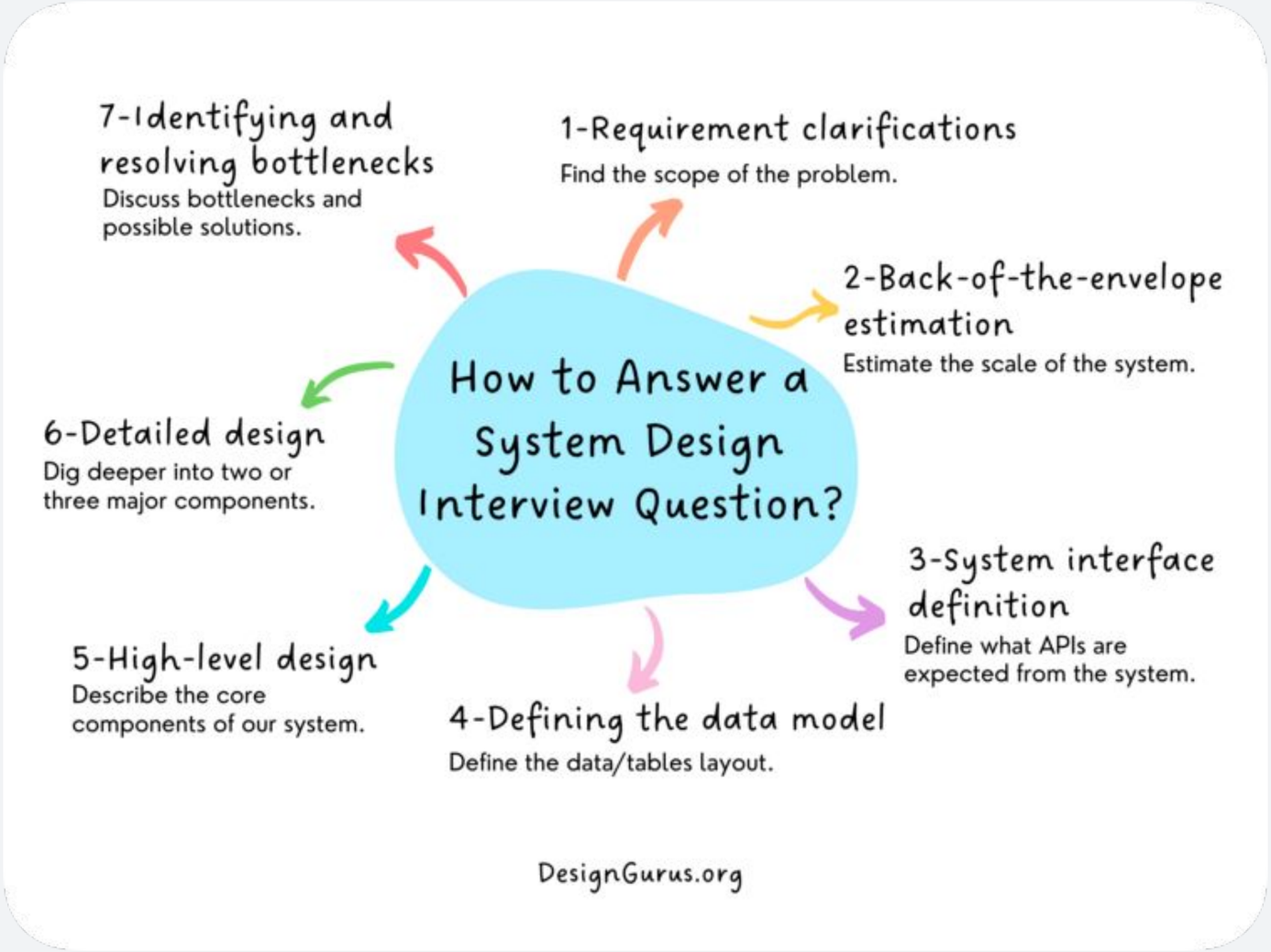
Учимся задавать вопросы на собеседовании

clck.ru/346F5P

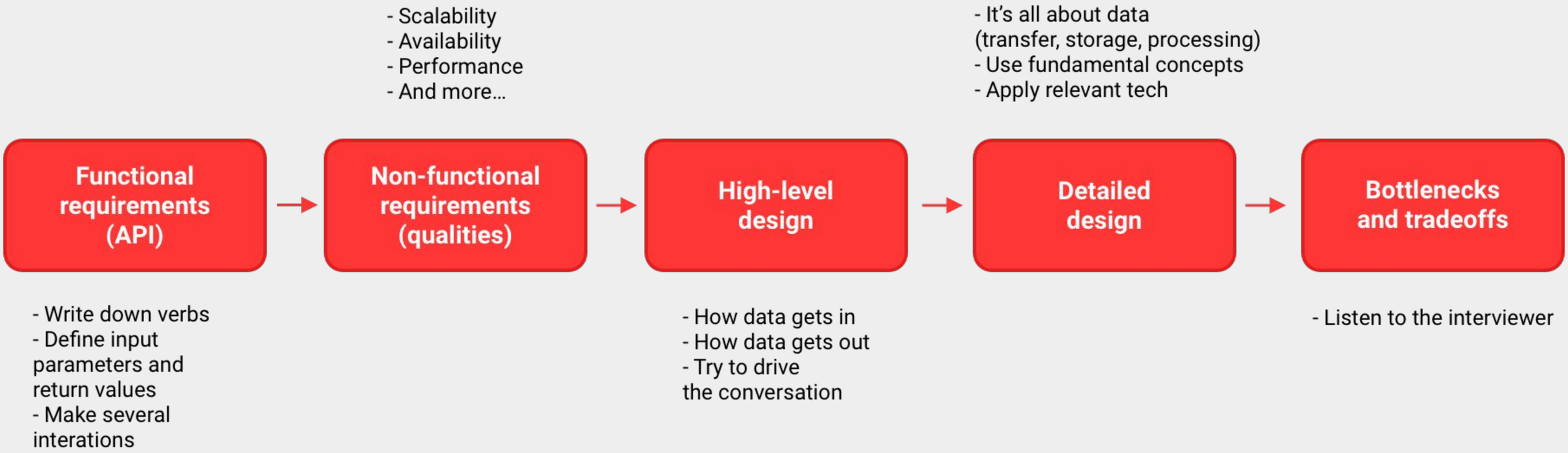
Interview Overview

Структура собеседования

Модель прохождения интервью



System Design Interview in Action



План собеседования

5 минут	Функциональные требования
5 минут	Нефункциональные требования
10 минут	Высокоуровневый дизайн, API
25 минут	Погружение в компоненты, масштабирование, трейдоффы

clck.ru/342bXu

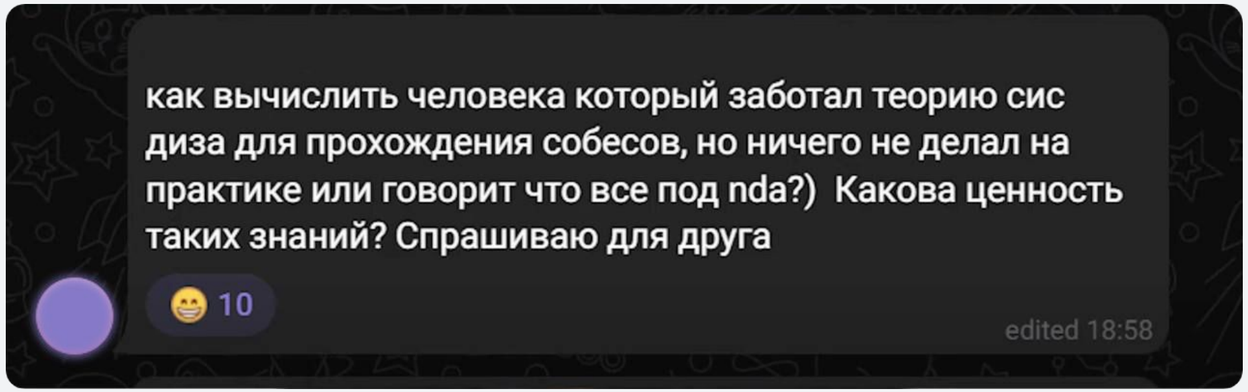
Сигналы, которые хочет увидеть интервьюер

- Умение понятно выражать свои идеи и отстаивать свои технические решения
- Наличие навыка проектирования систем и практического опыта
- Понимание вашего дизайна: сможет ли он масштабироваться, можете ли вы его улучшить, трейдоффы
- Многое другое

What the interviewer said	What the interviewer meant
We still have a lot of candidates to interview	You didn't pass this stage
We'll get back to you in two or three weeks	You didn't pass this stage
We'll be in touch	They won't be in touch
This is a fast-paced role	This is a disorganized chaos
We need someone who can multitask	We don't value prioritization
We need someone flexible	You'll do a bunch of non value-add work
The previous person left for another opportunity	We PIPed the previous person
We plan to promote you within 12 months	Please accept the underleveling
Do you have any questions? (at 30 mins out of 45)	Can we hang up?

Ошибка. Вызубрить готовые архитектуры

- Зазубривание готовых ответов вас не спасет (*спасет, но только на младших грейдах*)
- Вас могут спросить любой из сценариев, с которым вам не приходилось сталкиваться
- Сделайте акцент на системном подходе или методе к собеседованию

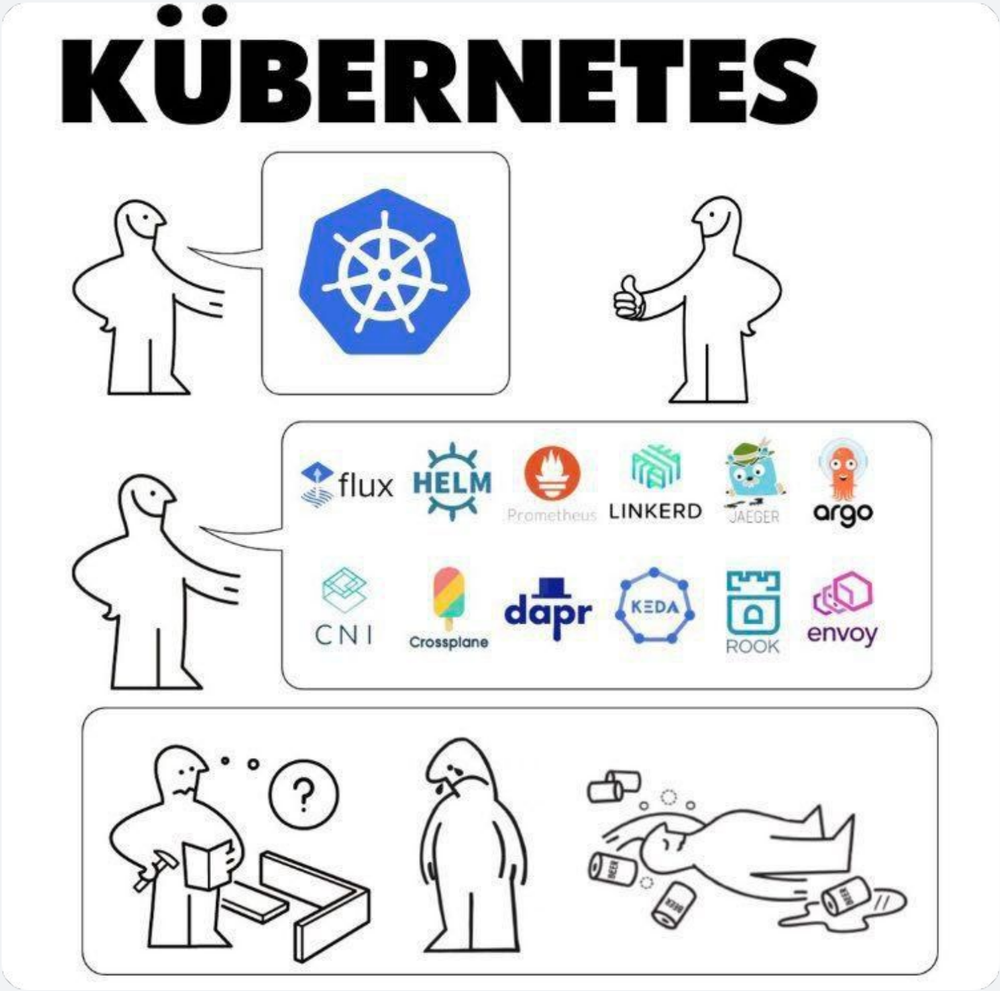


Functional Requirements

Функциональные требования (ФТ)

Функциональные требования

- Какие возможности у системы, которая она дает пользователю?
- Насколько она должна быть надежной, отзывчивой?
- Какие упущения допустимы?
- Какие ограничения должны быть у системы?



Алгоритм сбора требований

- Вспоминаем известные нам свойства системы
- Записываем их на доске
- Уточняем у интервьюера, какие из функций системы будем обсуждать в ходе интервью
- Стараемся, чтобы требования были приоритезированы (в порядке убывания значимости)

Пример: Dropbox / Google Drive

- Пользователь может загружать данные в облако
- Пользователь может управлять данными с различных устройств
- Устройства будут синхронизироваться
- Пользователи могут шарить файлы/папки
- Доступы read/write к файлам/директориям
- Пользователи могут настроить тарифный план
- Поддержка загрузки больших файлов

Пример ФТ и API: Uber

Uber Backend Design
FR
Customer
- request ride.
- types of cabs, seats, pool
- Rate drivers
- Make payments
- Trips History

Driver
- Request for rides
- accept/deny rides
- manage trips once started
- trip history
- payment history

System
- Efficient matching based proximity and on driver and user profile
- Minimise & predict ETAs and delays. (Weather/Events/Roads/Volume)
- Handle surge in demand
- predict, surge pricing
- Maximise profit

NFR
- HA
- Strong consistent (once matched)
- Scalable
- Handle both sparse & dense areas
- Reliable & durable

APIs
Customer
EstimateRide (customerId, source, destination, cabType)
    Response (estimated ETA, price, number of drivers in proximity)

RequestRide (customerId, source, destination, cabType, paymentMode)
    success/failure

Event
    RideFound (rideId, driverId, driverDetails, ETA, price)

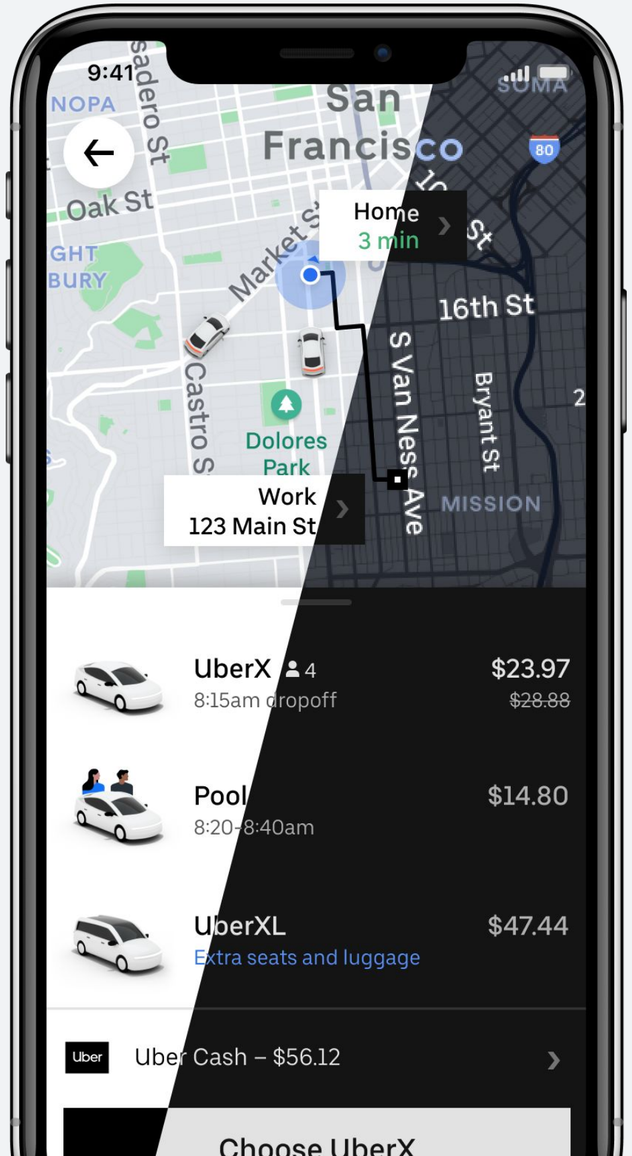
Driver
RequestTrip (driverId, location, preferredDestination)

Event
    RideFound (rideId, tripDetails, pickup

AcceptRide (rideId, driverId)
    Response- customer details, dropLocation
DenyRide (rideId, driverId)
StartTrip (rideId, driverId)

Event
    InterimPickupAdded (rideId, pickup location)

EndTrip (rideId, driverId)
    Response- payment details



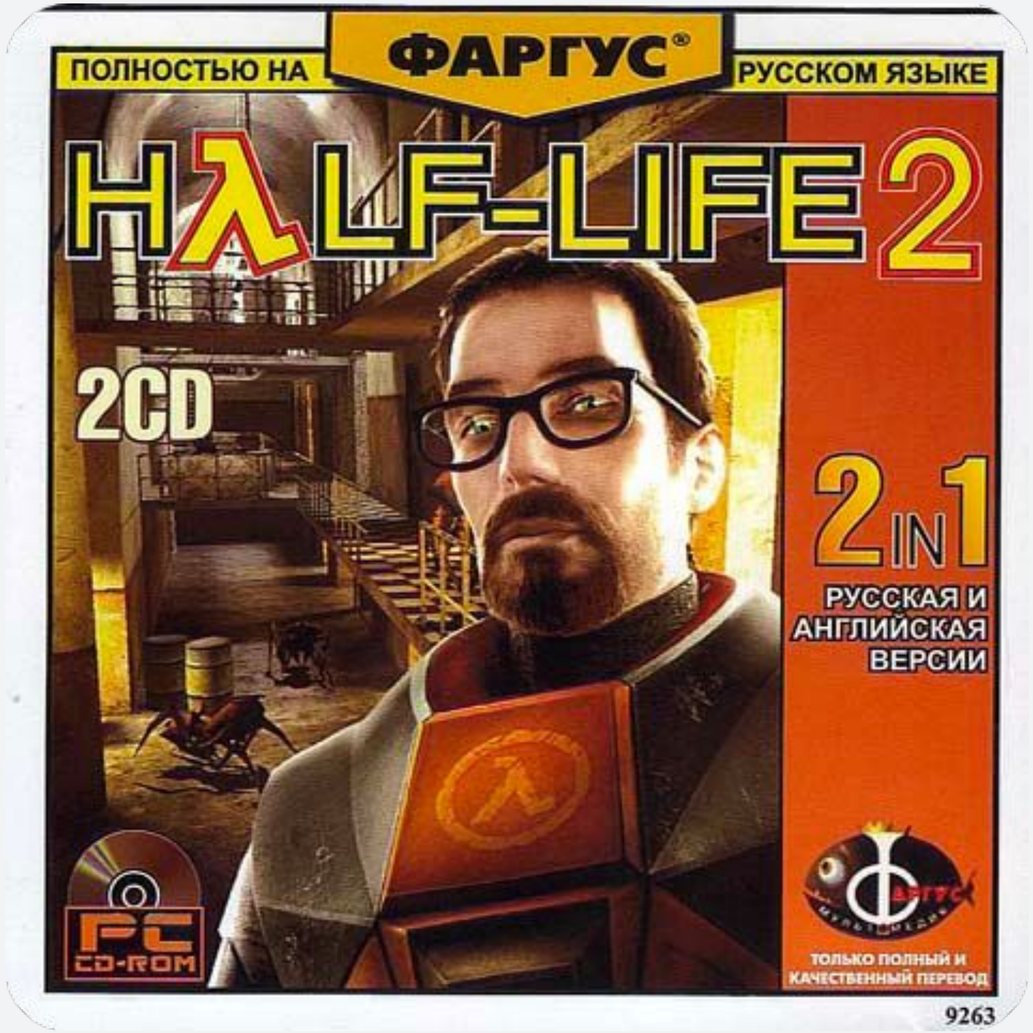
Ошибка. Делать всё

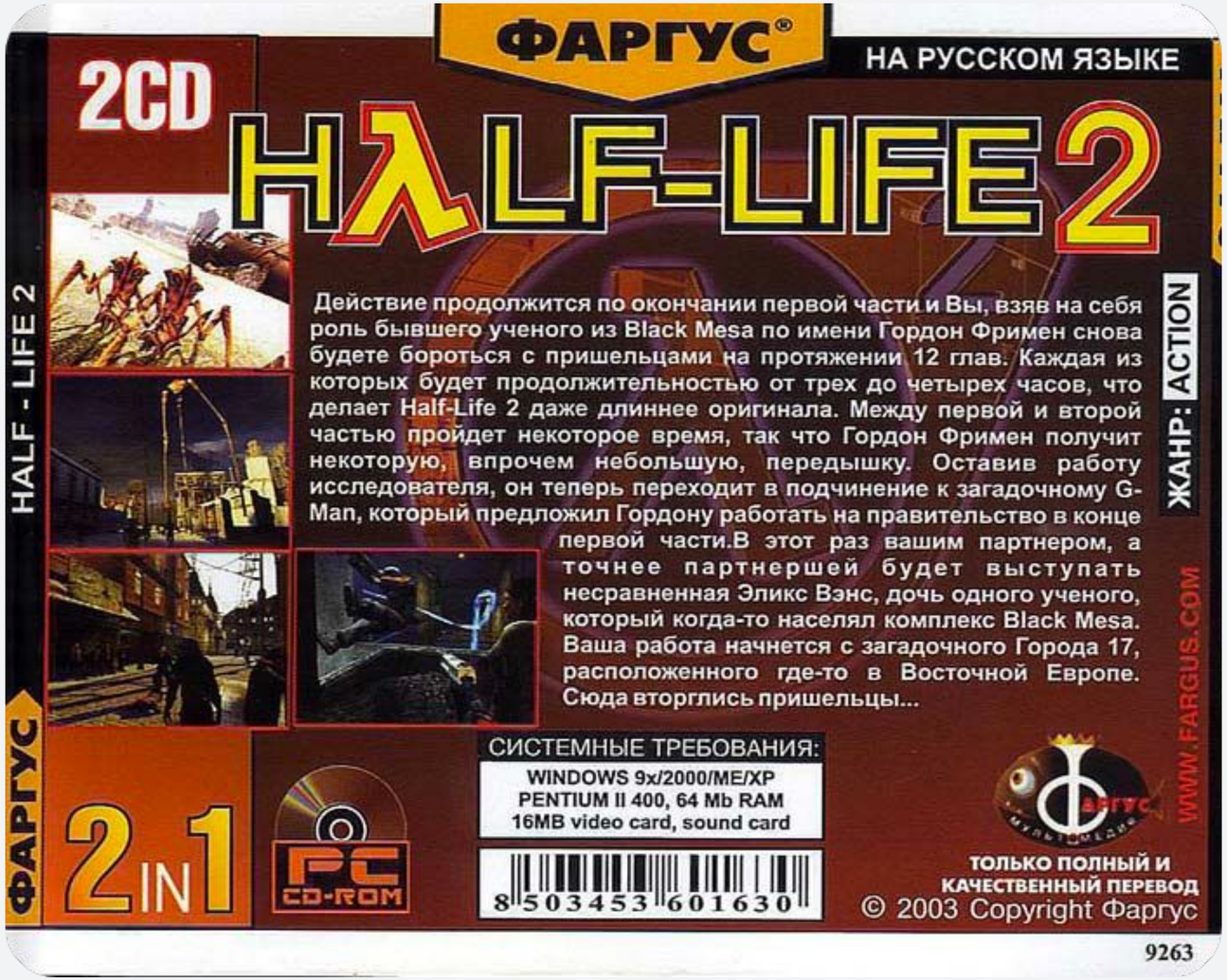
- Не пытайтесь в ходе интервью спроектировать все функции системы
- Сделать всё у вас точно не хватит времени
- Вместо этого уточните у интервьюера, что именно вы собираетесь реализовать в ходе интервью

Non-Functional Requirements

Нефункциональные требования (НФТ)

НФТ


ПОЛНОСТЬЮ НА
ФАРГУС®
РУССКОМ ЯЗЫКЕ
HALF-LIFE 2
2CD
2 IN 1
РУССКАЯ И АНГЛИЙСКАЯ ВЕРСИИ
PC CD-ROM
ТОЛЬКО ПОЛНЫЙ И КАЧЕСТВЕННЫЙ ПЕРЕВОД
9263


ФАРГУС®
НА РУССКОМ ЯЗЫКЕ
2CD
HALF-LIFE 2
HALF - LIFE 2
Действие продолжится по окончании первой части и Вы, взяв на себя роль бывшего ученого из Black Mesa по имени Гордон Фримен снова будете бороться с пришельцами на протяжении 12 глав. Каждая из которых будет продолжительностью от трех до четырех часов, что делает Half-Life 2 даже длиннее оригинала. Между первой и второй частью пройдет некоторое время, так что Гордон Фримен получит некоторую, впрочем небольшую, передышку. Оставив работу исследователя, он теперь переходит в подчинение к загадочному G-Man, который предложил Гордону работать на правительство в конце первой части.В этот раз вашим партнером, а точнее партнершей будет выступать несравненная Эликс Вэнс, дочь одного ученого, который когда-то населял комплекс Black Mesa. Ваша работа начнется с загадочного Города 17, расположенного где-то в Восточной Европе. Сюда вторглись пришельцы...
ЖАНР: ACTION
WWW.FARGUS.COM
СИСТЕМНЫЕ ТРЕБОВАНИЯ:
WINDOWS 9x/2000/ME/XP
PENTIUM II 400, 64 Mb RAM
16MB video card, sound card
8 503453 601630
ФАРГУС
2 IN 1
PC CD-ROM
ТОЛЬКО ПОЛНЫЙ И КАЧЕСТВЕННЫЙ ПЕРЕВОД
© 2003 Copyright Фаргус
9263

SLA/SLO

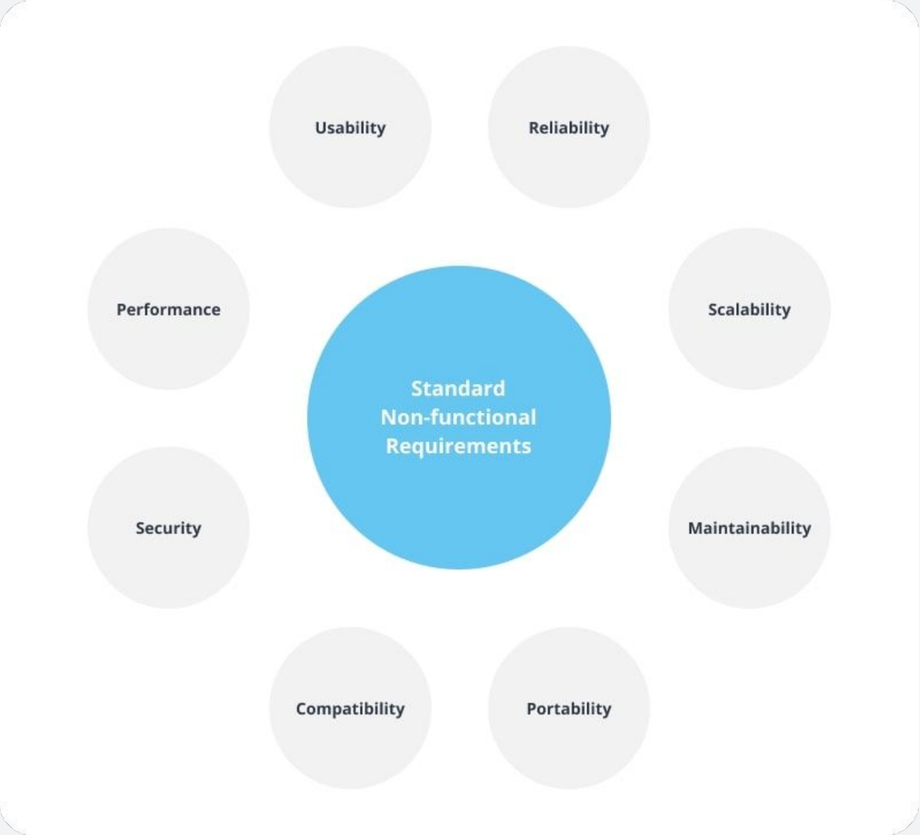
SLA → SERVICE LEVEL AGREEMENT
the agreement you make with your clients or users

SLOs → SERVICE LEVEL OBJECTIVES
the objectives your team must hit to meet that agreement

SLIs → SERVICE LEVEL INDICATORS
the real numbers on your performance

Метрики НФТ

- ➔ DAU/MAU для расчета,
- ➔ кол-во пользователей через 5 (10) лет,
- ➔ кол-во операций пользователем в день,
- ➔ RPS, сетевой трафик и кол-во соединений,
- ➔ CPU,
- ➔ HDD/SSD (дисковое пространство)
- ➔ $$$ (стоимость)



Ошибка. Не уточнить детали

- Решить, что мы строим хайлоад на миллионы / миллиарды пользователей
- Уточнить, сколько юзеров будет у сервиса - надо ли нам дизайнить сначала MVP, а потом скейлить
- Нужен ли этап оценки НФТ, или мы будем сразу рисовать высокоуровневый дизайн системы?

clck.ru/33vkDu

Как получить НФТ

WhatsApp

Metric

Requests/sec (read, write)

- 1 million users
- Receives 72 msg/day
- Sends 36 msg/day

Assumptions

72 million reads/day
~ 833 req/second
READS

36 million writes/day
~ 416 req/second
WRITES

Calculation

REQ / SEC

↓

STORAGE

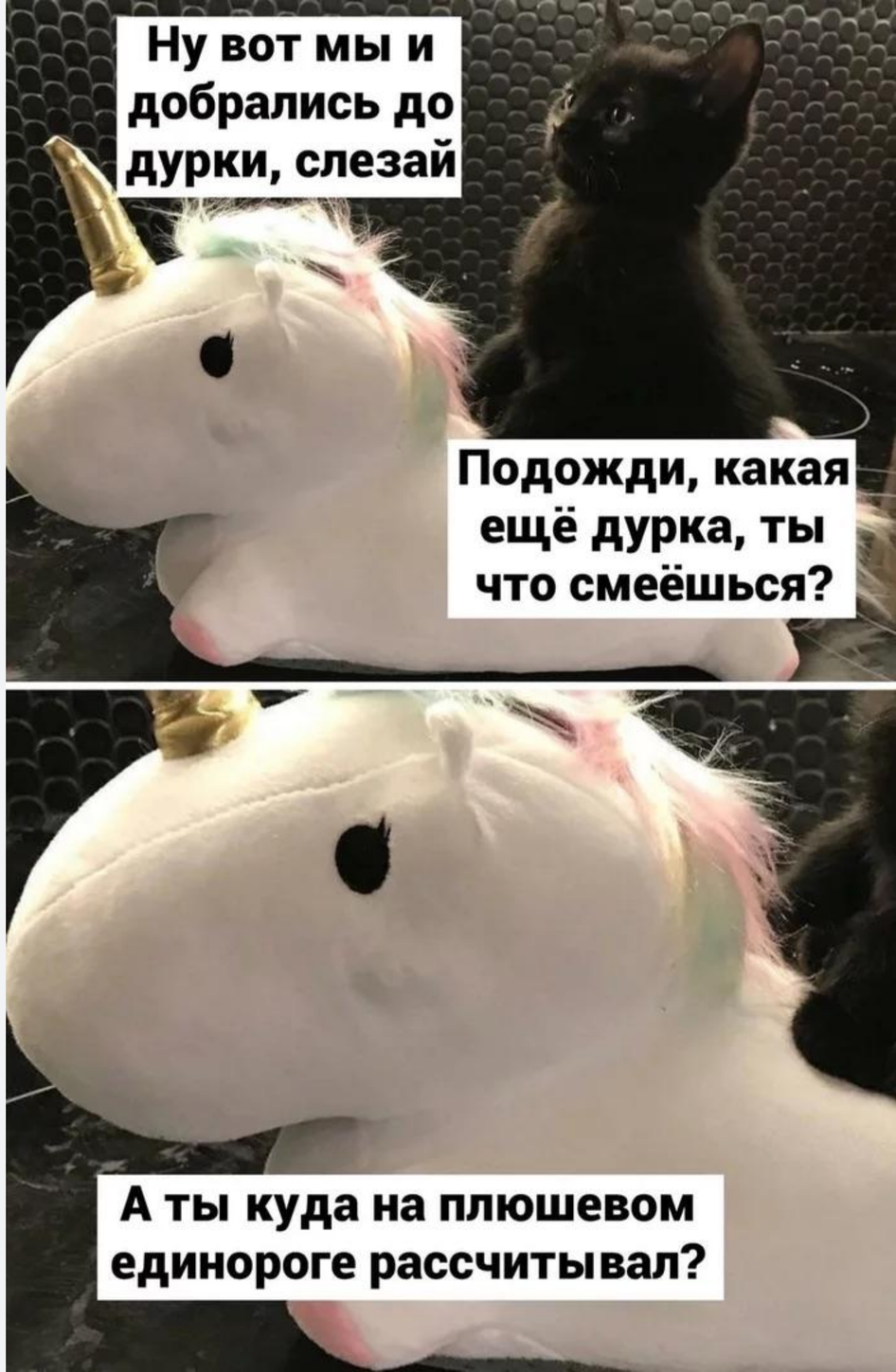
↓

BANDWIDTH

Числа, которые требуется знать

Operation name	Time
L1 cache reference	0.5 ns
Branch mispredict	5 ns
L2 cache reference	7 ns
Mutex lock/unlock	100 ns
Main memory reference	100 ns
Compress 1K bytes with Zippy	10,000 ns = 10 µs
Send 2K bytes over 1 Gbps network	20,000 ns = 20 µs
Read 1 MB sequentially from memory	250,000 ns = 250 µs
Round trip within the same datacenter	500,000 ns = 500 µs
Disk seek	10,000,000 ns = 10 ms
Read 1 MB sequentially from the network	10,000,000 ns = 10 ms
Read 1 MB sequentially from disk	30,000,000 ns = 30 ms
Send packet CA (California) ->Netherlands->CA	150,000,000 ns = 150 ms

clck.ru/348i4k



Хранение данных

- Нужно думать не только про задачи системы, но и про их выполнение на hardware
- Есть ли в системе изображения? Видео?
- Сколько места под данные нам потребуется? Какого рода хранилища?
- Сколько серверов и стораджа нужно нам для выполнения?

Пример расчетов

DAU: 100 миллионов пользователей

Requests per second (RPS): 10K/s

Хранение данных (Storage) (в день): ~400 GB

Хранение данных (Storage) (10 лет): ~1.4 PB для загруженных сообщений и метаданных

Трафик (Bandwidth): ~10 MB/s

WhatsApp or Facebook Messenger

FR
- one-on-one conversations between users
- ack & read receipts should be exchanged
- different types of media can be shared
- support for push notifications when user is offline
- persistent or transient storage
- Backups on cloud drives
- Group conversations?

NFR
- real time chatting experience
- super low latency for exchanging messages
- High consistency
- Availability is desired but can be slightly compromised
- linearly scalable for millions/billions of users

Scale constraints/capacity
- 500M users. Sending 40 messages daily
- 20B messages daily
- 100 bytes per message

Storage
- 20*[9]*100 => 2TB/day
- 2 * 5 * 365 => 36.5PB in 5 years

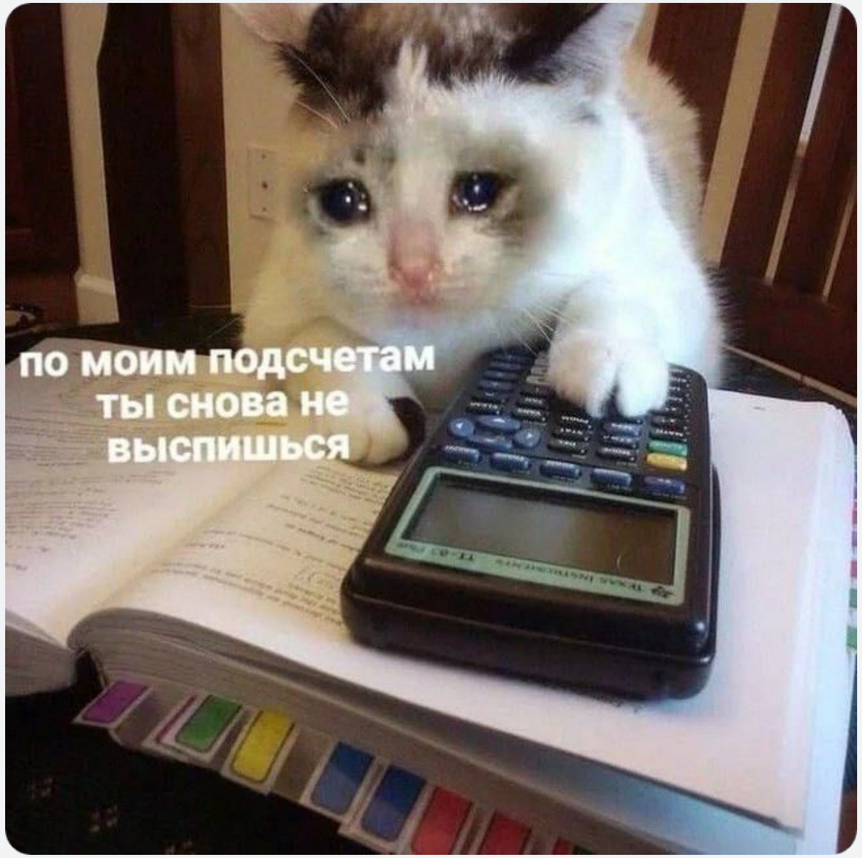
Ingress/Egress/Connections
- 20 * [9]/24*60*60 ~ 25 Mbps for both.
- 500M active connections

- 2 media files daily
- 200 KB per media
- 2 * 200 * [3] * 2 * 500 * [6] => 4 * [14]
~ 400 TB of media storage

clck.ru/33vkDu

Ошибка. Некорректный или медленный расчет

- Этап нужен не для того, чтобы показать, какой вы гигачад в математике
- Научитесь делать расчет быстро - например, быстро возводить в степень (см. Google SRE Book)
- Расчет покажет, где есть ограничения (boundary) в вашей системе (по CPU, стораджу, RAM…)
- На этапе HLD вы решаете, какими средствами вы будете масштабировать найденное узкое место (bottleneck)



Как уточнять НФТ

Users

Who will user the system?

How the system will be used?

Scale (read, write)

How many read queries per second?

How much data is queried per request?

How many video views are processed per second?

Can there be spikes in traffic?

Performance

What is expected write-to-read data delay?

What is expected p99 latency for read queries?

Cost

Should the design minimize the cost of development?

Should the design minimize the cost of maintenance?

Джуниор → Миддл → Сеньор

На этом этапе опытный интервьюер уже сделал первые выводы

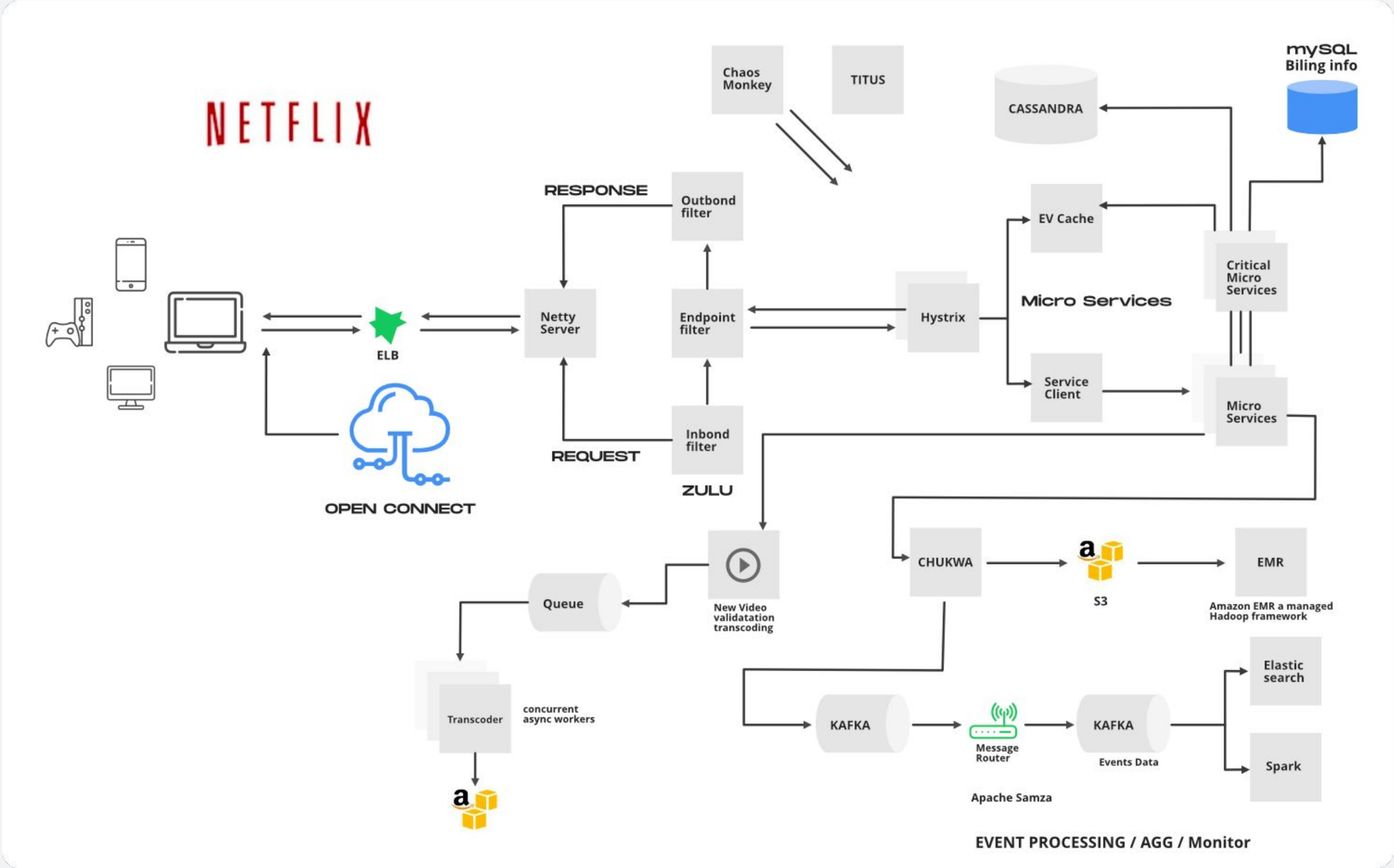
High Level Design

Высокоуровневый дизайн

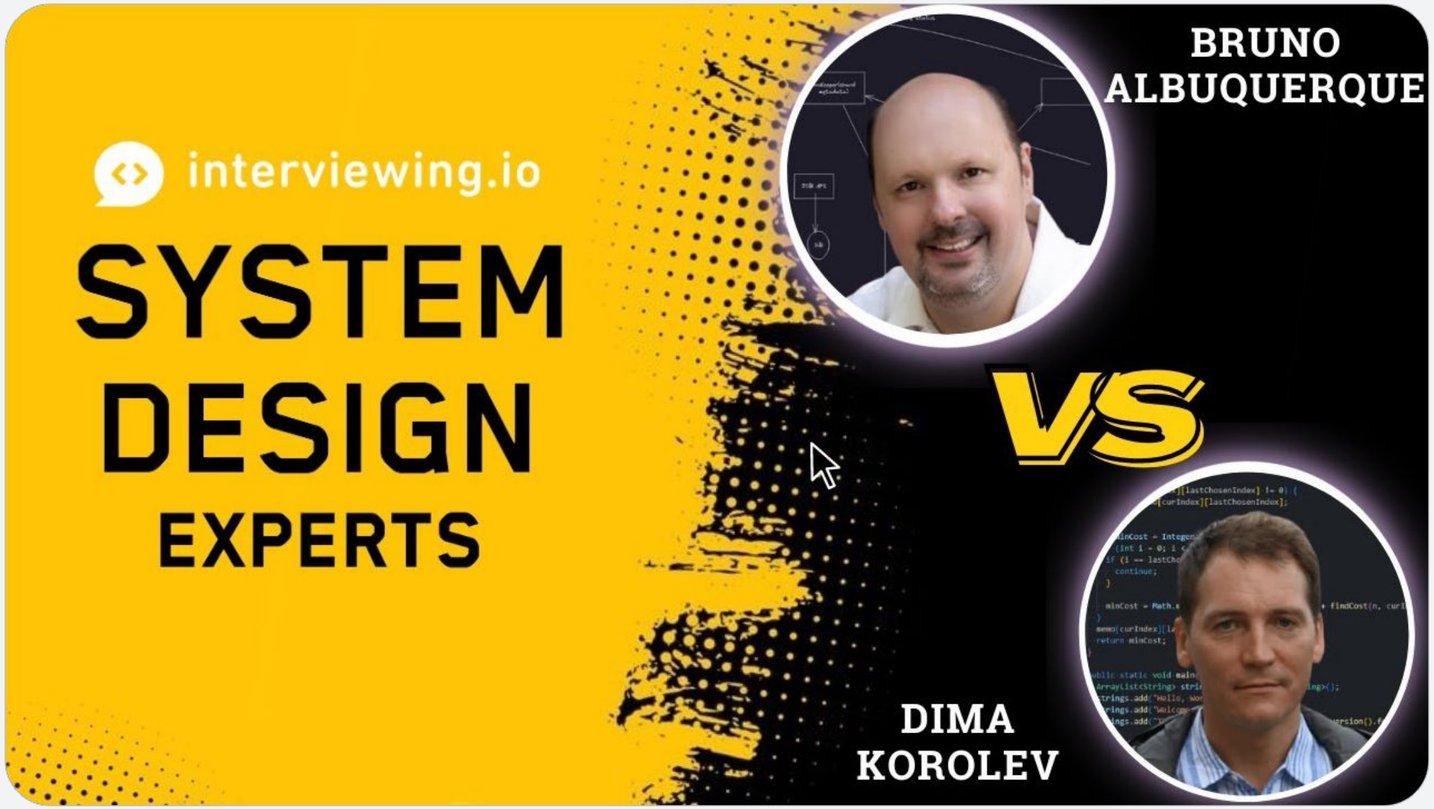
От простого к сложному

- ➔ Сделайте стандартный простой дизайн системы
- ➔ Затем эволюционируйте его

Не бывает идеальных архитектур

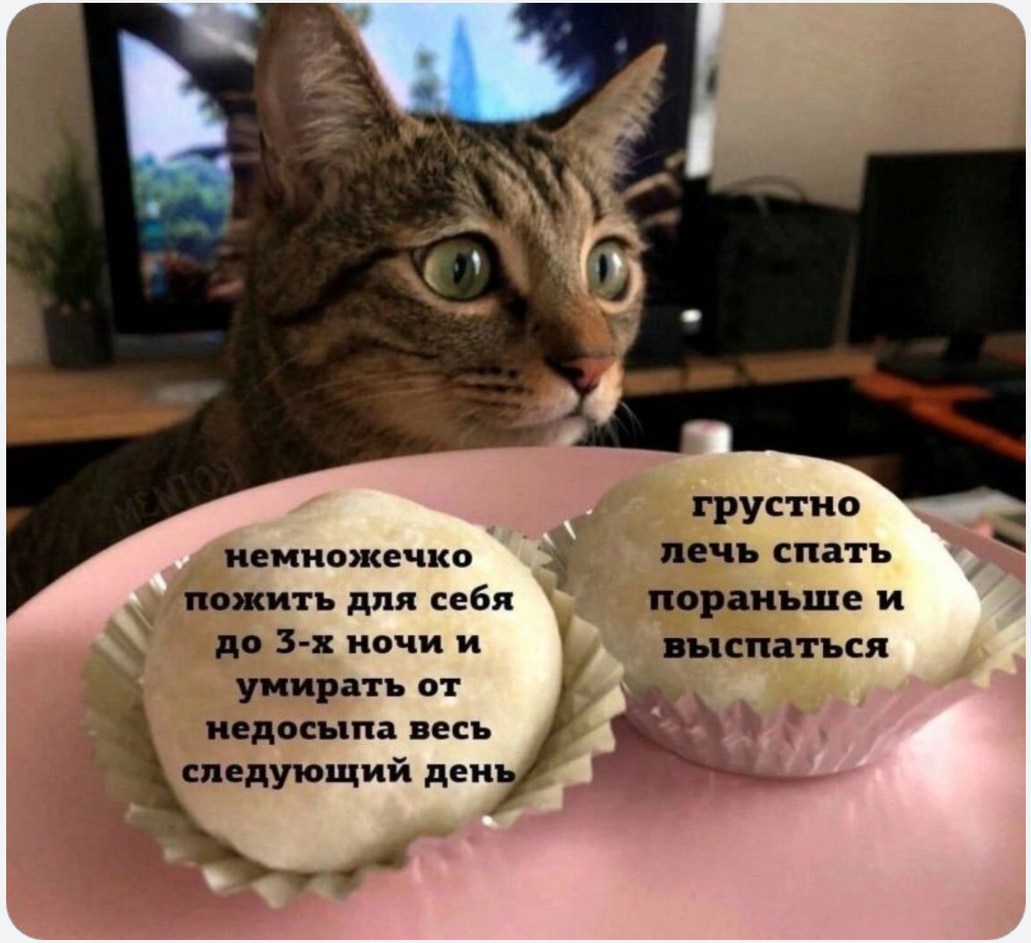


Two Ex-Google System Design Experts Compete: Who will design the better system?

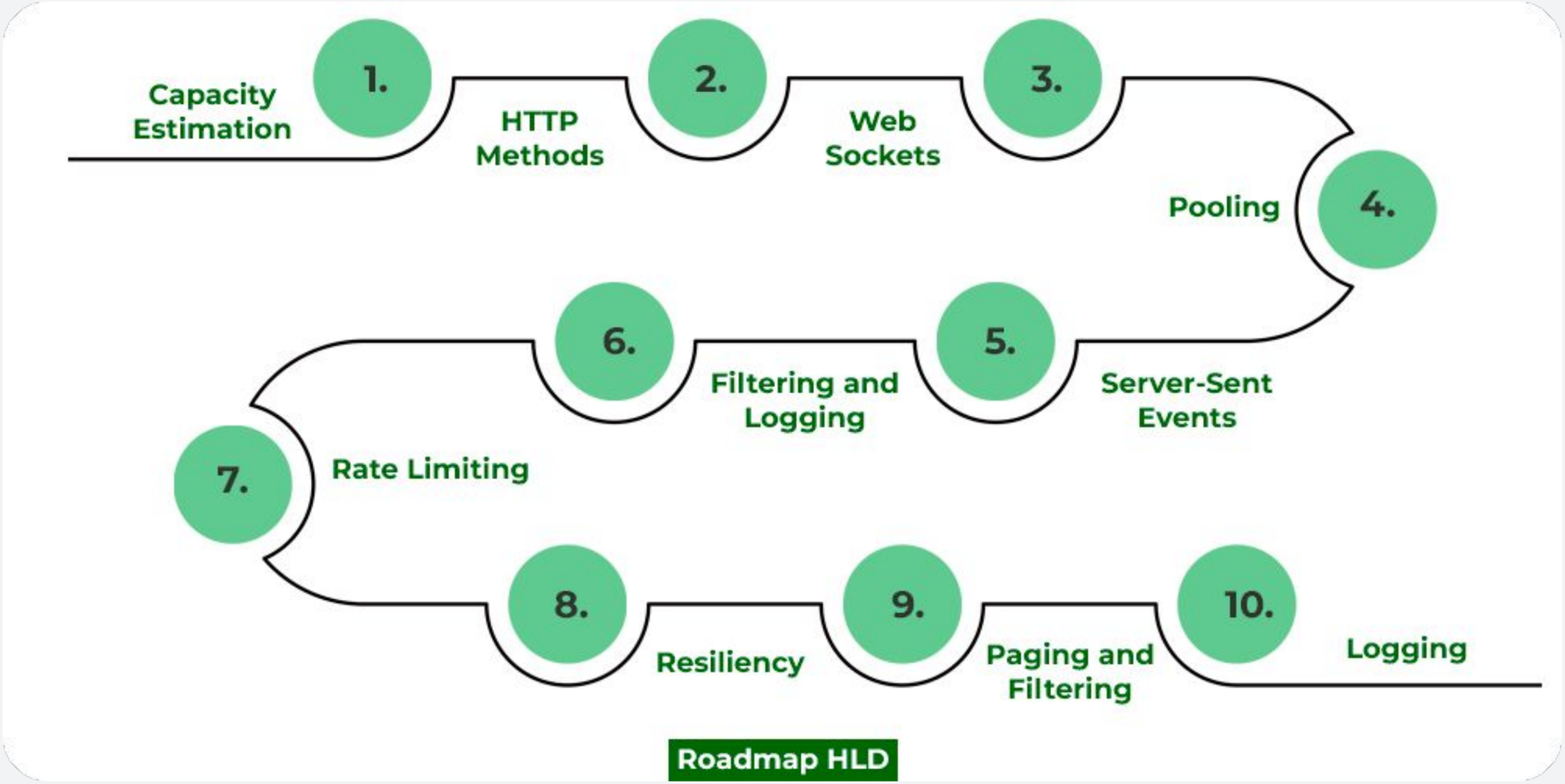


https://youtu.be/Zi0pPkiFemE https://youtu.be/PU_sgwZvm6s

Определите узкие места и трейдоффы



Нюансы HLD



clck.ru/348iNH

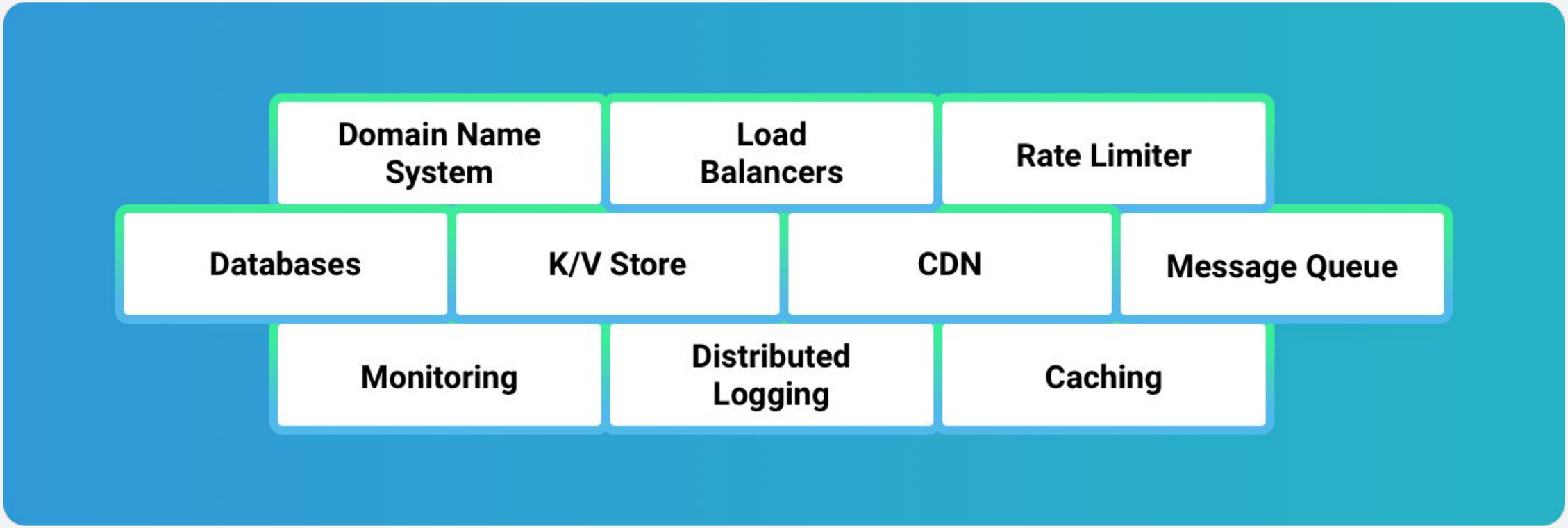
Ошибка. Оценка хранилища данных

- Дьявол кроется в деталях, которые требуются на высокие грейды в MANGA (FAANG)
- Помните, что вам доступно только ~85% объема диска
- За год выходят из строя 3-5% дисков
- Может быть ограничение на макс. размер диска для БД
- Не забудьте умножить получившееся число на фактор репликации

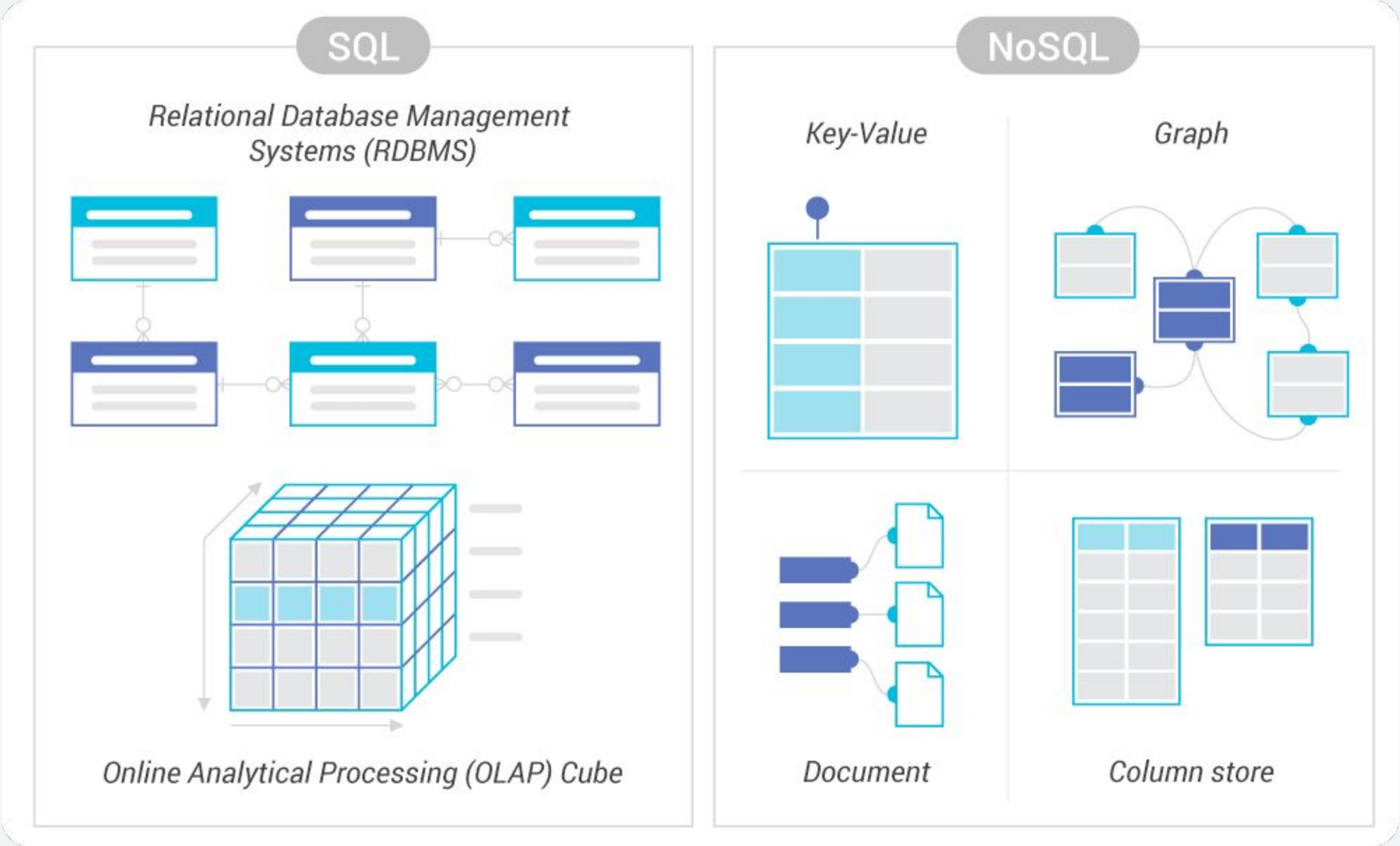
Components

Компоненты системы

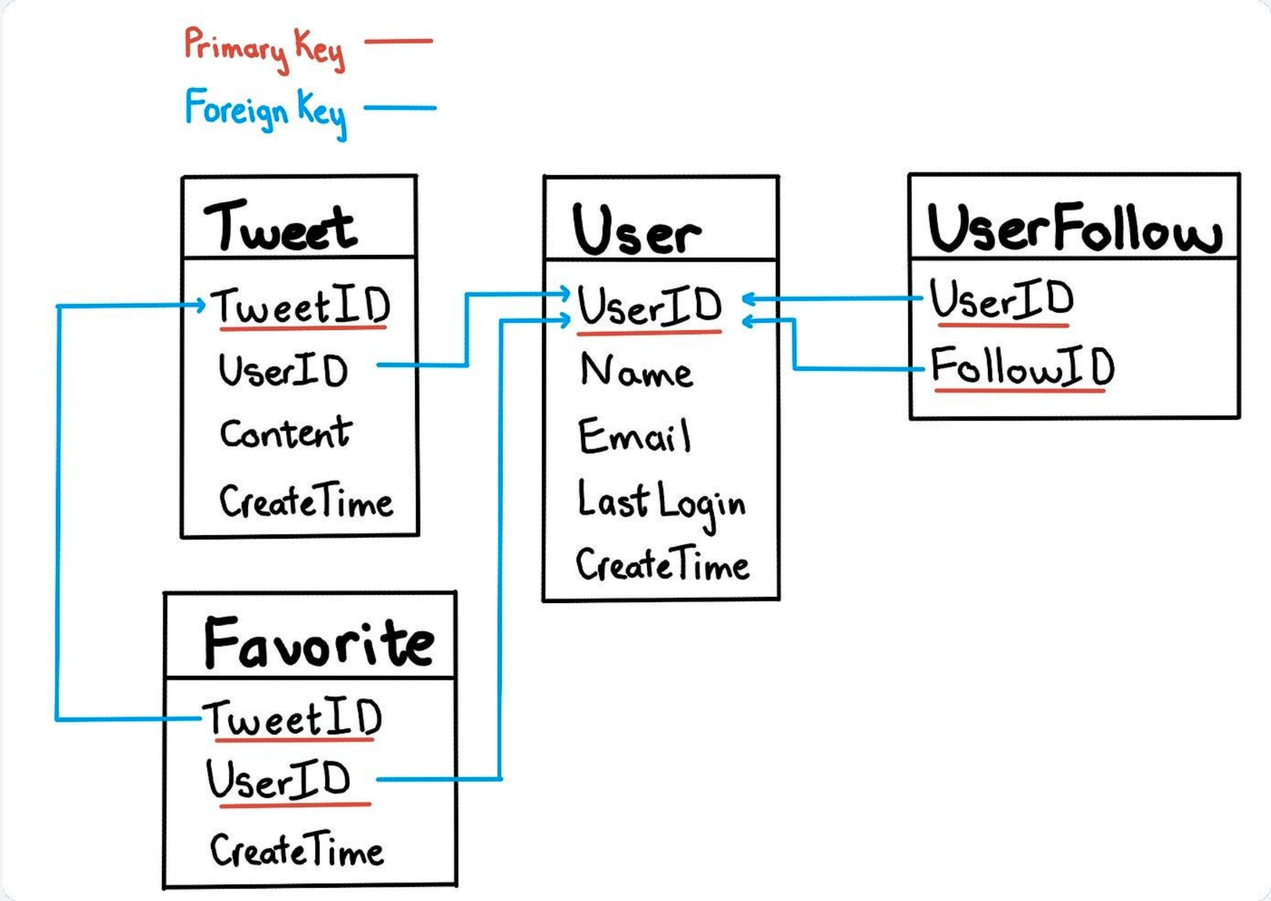
Библиотека компонентов



SQL vs NoSQL


SQL
Relational Database Management Systems (RDBMS)
Online Analytical Processing (OLAP) Cube
NoSQL
Key-Value
Graph
Document
Column store

Детализация схемы данных



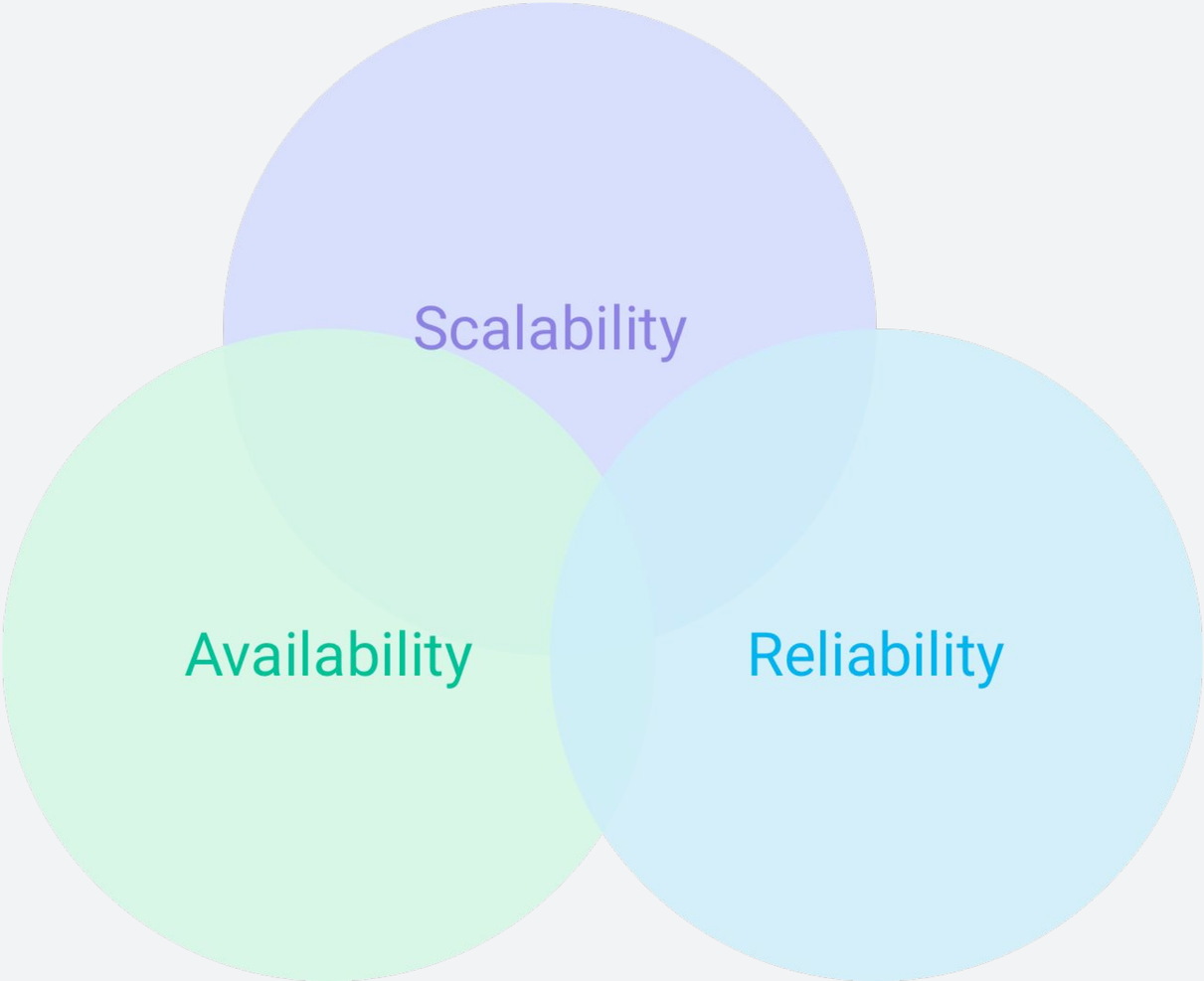
Ошибка. “Хочу и буду”

- Не задумываться над своими выборами
- Будьте готовы к вопросам “зачем?” и “почему?”. Почему React, а не Angular? Почему GET, а не POST? Почему веб-сокеты, а не long polling?
- Представьте, что вы обсуждаете новую фичу с членами вашей команды, и они задают вопросы.
- Залог успеха: то, насколько хорошо вы умеете слушать, раскрывать свои мысли, аргументировать

Scaling

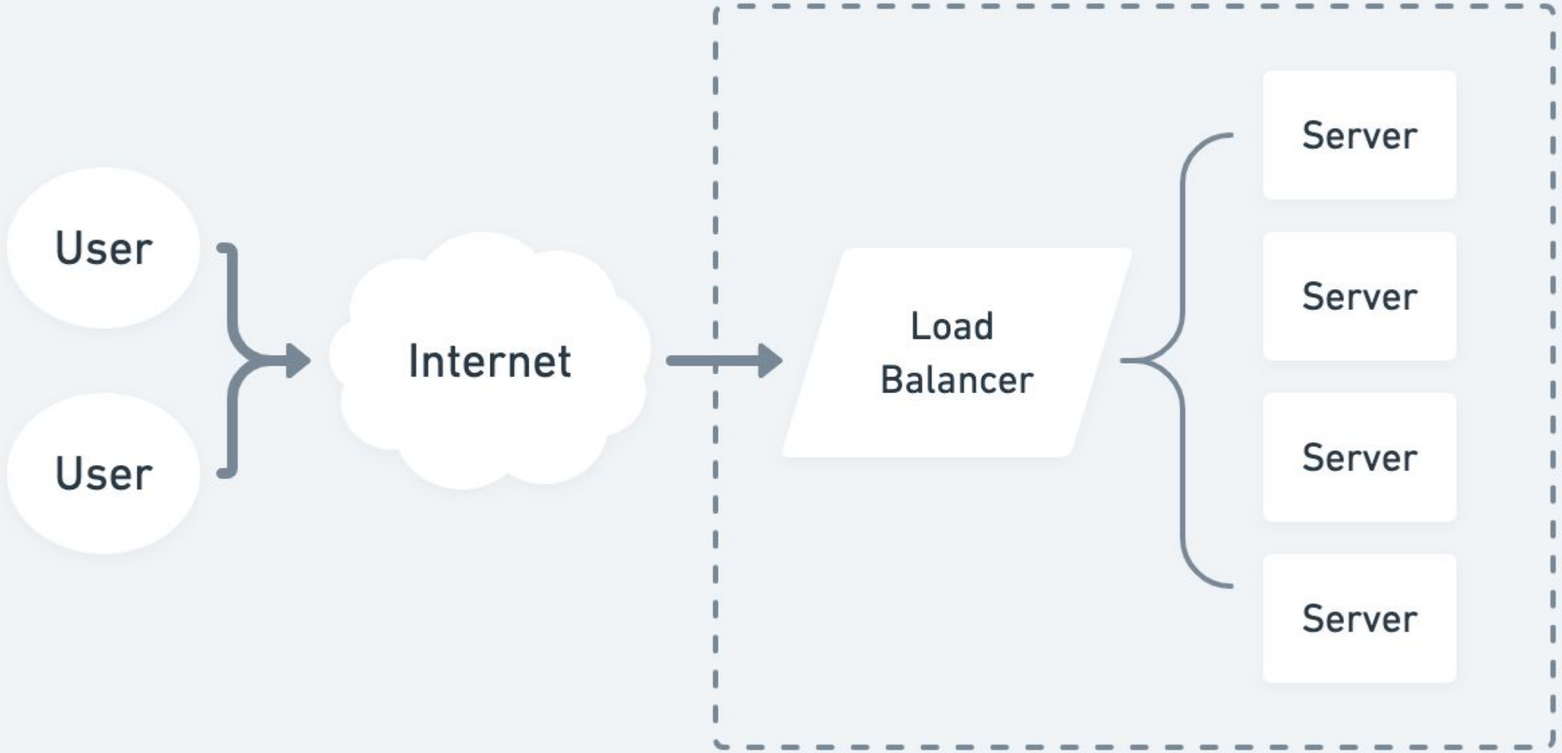
Масштабирование

Ключевые характеристики



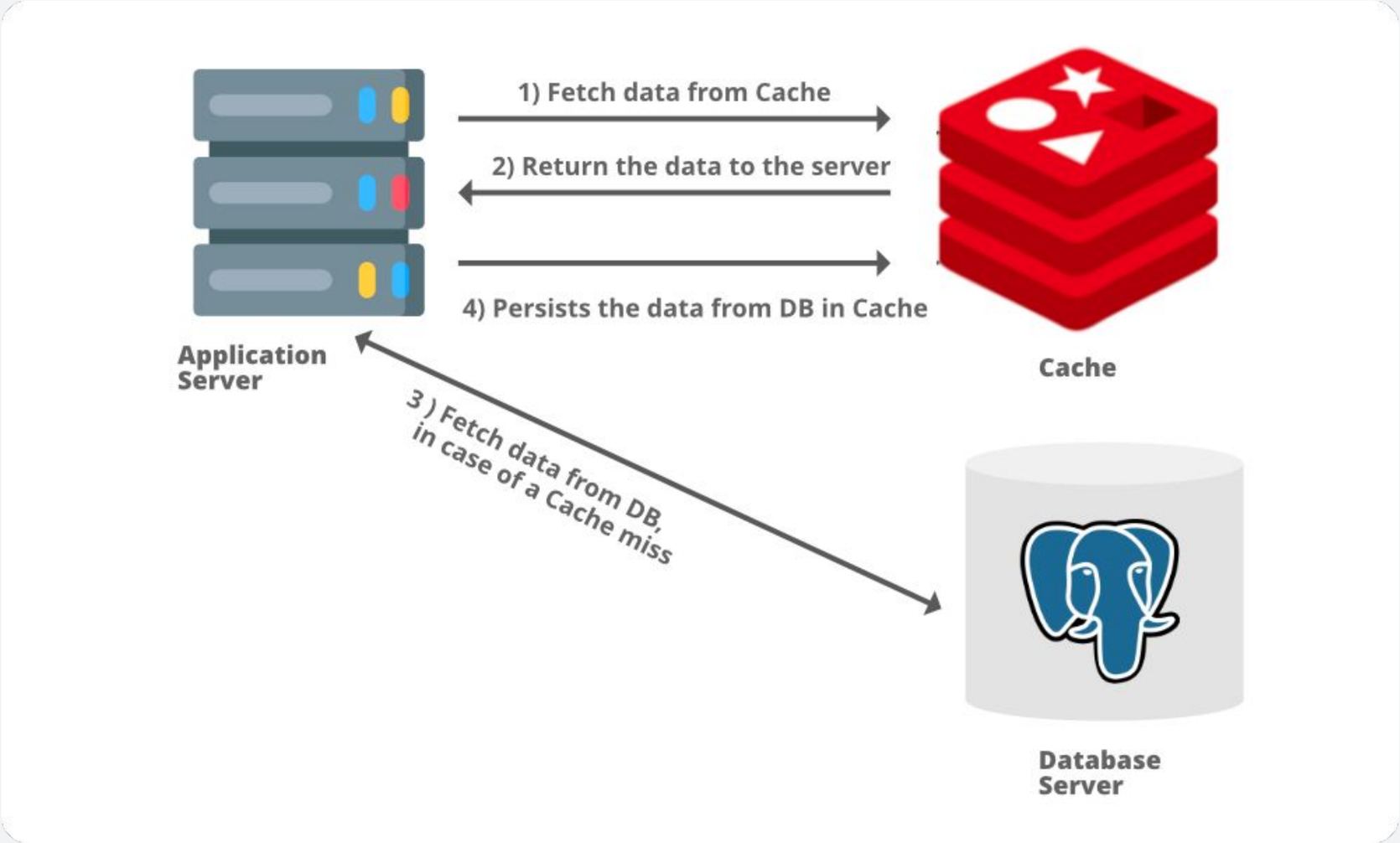
Балансировка нагрузки

Балансировка нагрузки



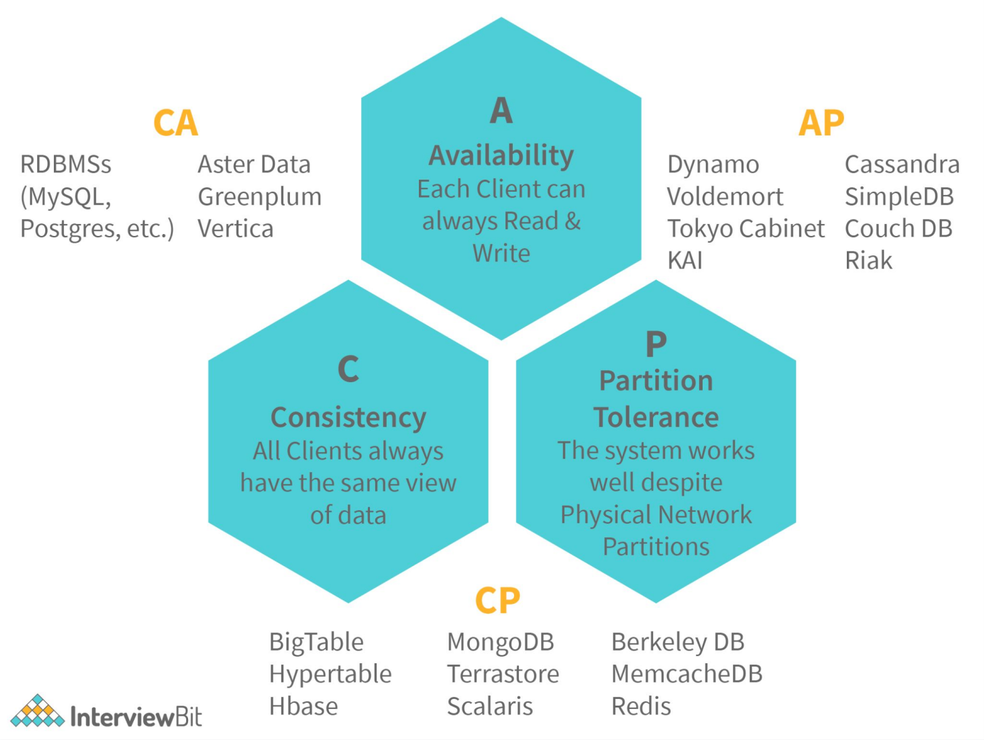
tryexponent.com
/load-balancers

Кэширование

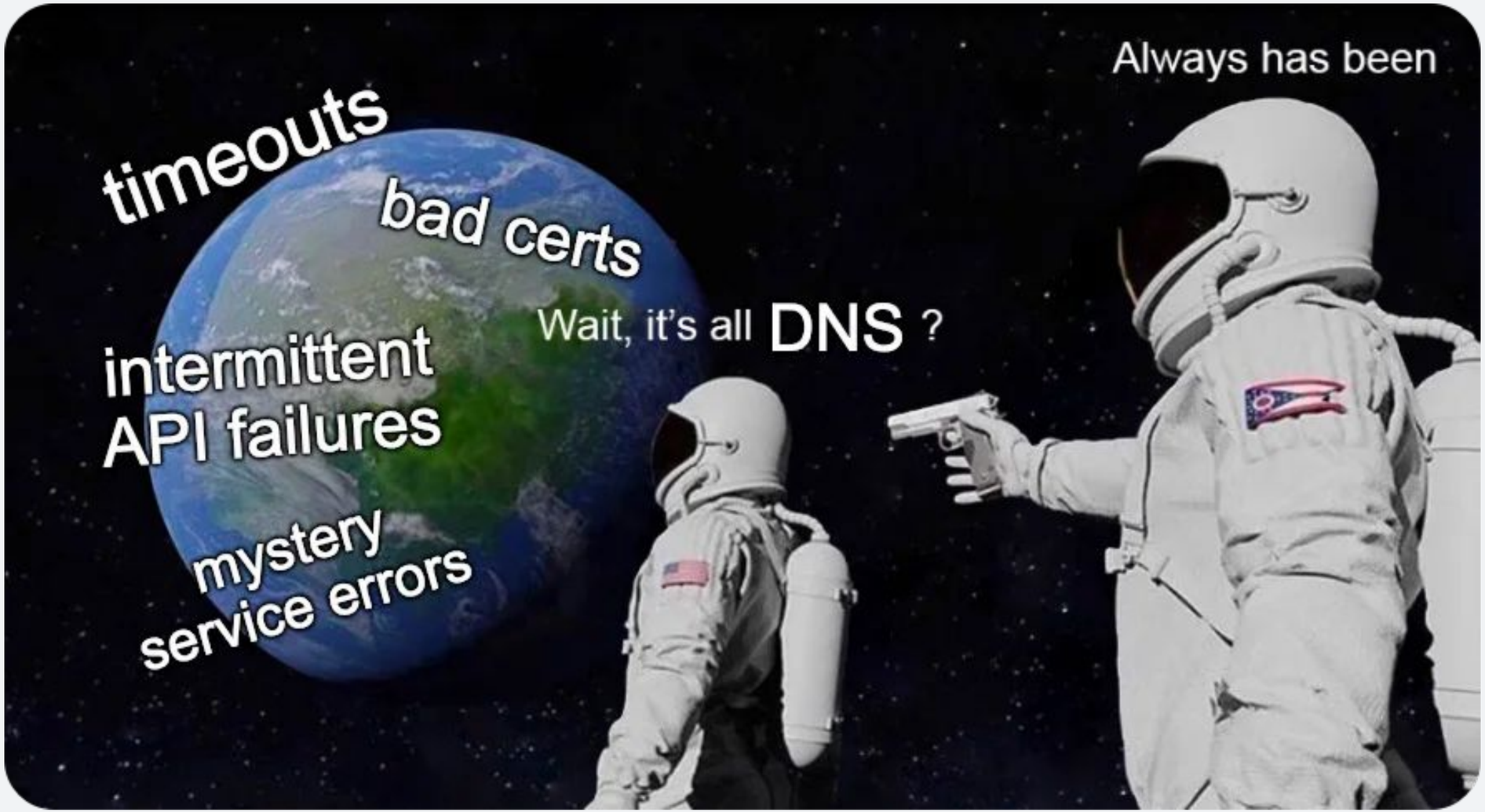


CAP-теорема

Availability
SQL Server,
Postgres,
Oracle,
MySQL...
Cassandra,
SimpleDB,
Riak,
Dynamo...
Pick two
Consistency
MongoDB,
HBase,
BigTable...
Partition tolerance


CA
RDBMSs (MySQL, Postgres, etc.)
Aster Data
Greenplum
Vertica
A
Availability
Each Client can always Read & Write
AP
Dynamo
Voldemort
Tokyo Cabinet
KAI
Cassandra
SimpleDB
Couch DB
Riak
C
Consistency
All Clients always have the same view of data
P
Partition Tolerance
The system works well despite Physical Network Partitions
CP
BigTable
Hypertable
Hbase
MongoDB
Terrastore
Scalaris
Berkeley DB
MemcacheDB
Redis
InterviewBit

DNS

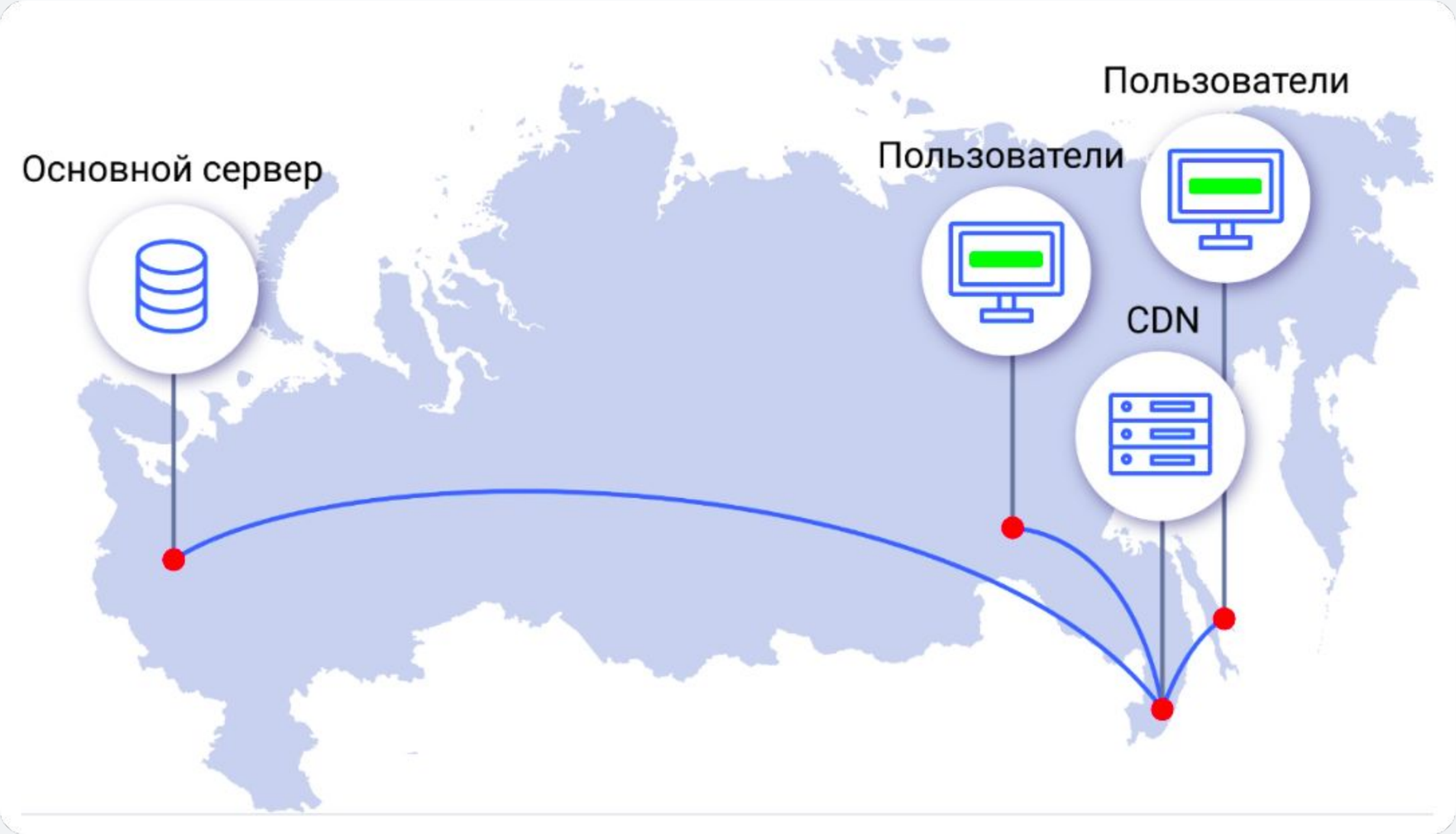

timeouts
bad certs
intermittent API failures
mystery service errors
Wait, it's all DNS ?
Always has been

Полезные приемы

- Кэширование
- Шардирование
- Репликация
- Индексирование
- Consistent Hashing

clck.ru/34897x

CDN



habr.com/463915

Ошибка. Баззворды

- Использовать в дизайне технологии и компоненты, про которые вы слышали, но утверждать, что вы с ними работали

clck.ru/345fE7

Avoid naming names

I highly encourage you to avoid naming specific technologies and just stay with the most generic abstraction possible. Instead of saying “I would cache things in Redis”, just say that you would use an in-memory datastore. Use a NoSQL database in place of “I would use MongoDB”.

The moment you put a name out there your interviewers are going to start asking questions about that specific technology and why you’ve made that specific choice. Restrain yourself from mentioning technologies you are not too familiar with. For this kind of situation, I always say something like:

> *Disclaimer, I’ve never worked with this technology but I read that* ***${BUZZWORD}*** *is generally used to solve this kind of problem.*
>
> *– — Pablo*

You should at least know the very basics of **${BUZZWORD}** technology and why people recommend it for solving this problem. Don’t just talk for the sake of talking. In general, we prefer to work with someone who admits when they don’t know something instead of bullshiting their way out of a situation.

Not knowing is fine, dishonesty is not. This interview can bring up a wide variety of topics and no one is expected to be an expert in all of those. It’s more about how you would tackle the *unknowns* and your thought process for that. Identify single points of failure and bottlenecks. Be mindful of trade-offs and explicitly mention them to your interviewer.

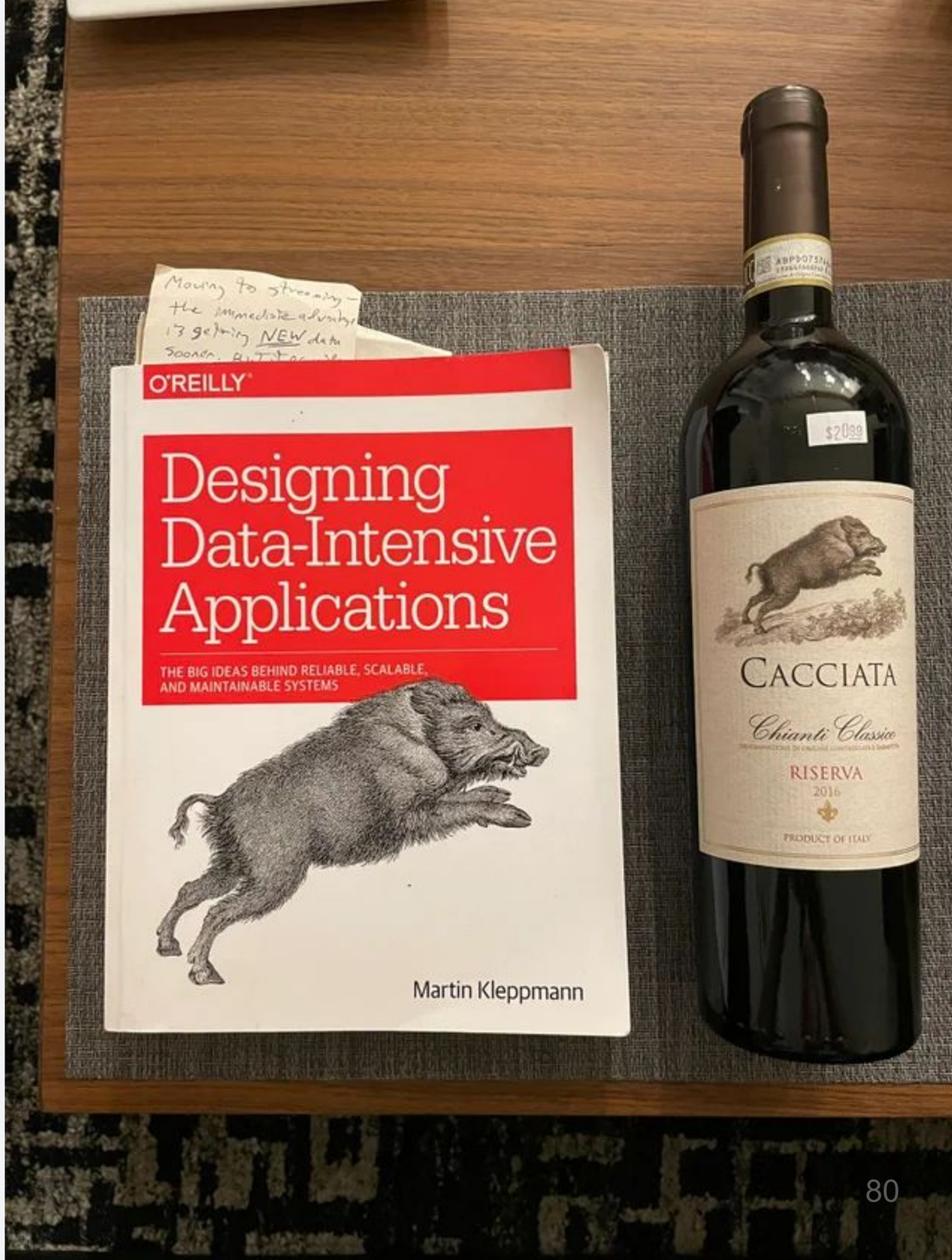
Interview Preparation

Как подготовиться к собеседованию

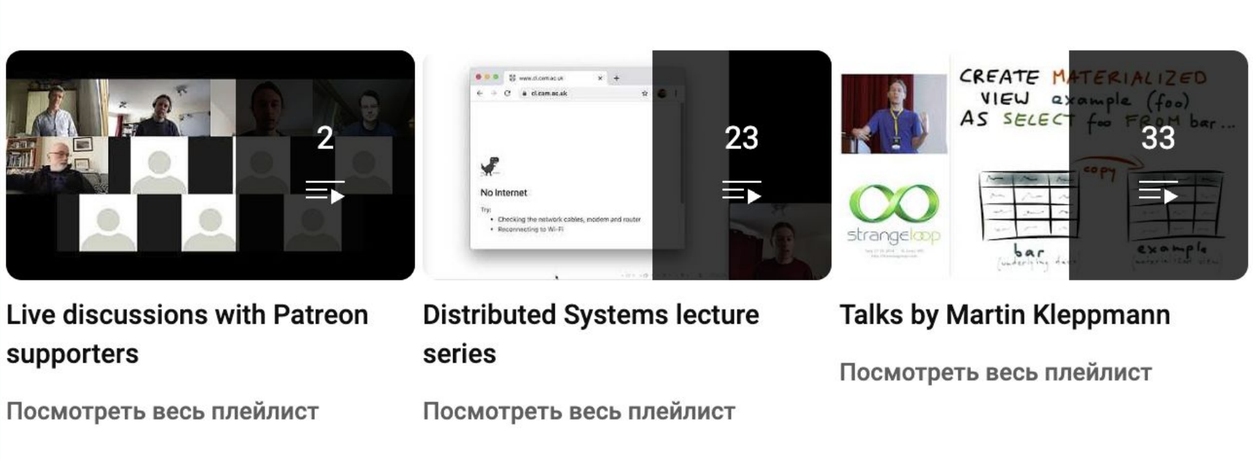
Классический подход

“Просто прочитай книжку с кабанчиком”

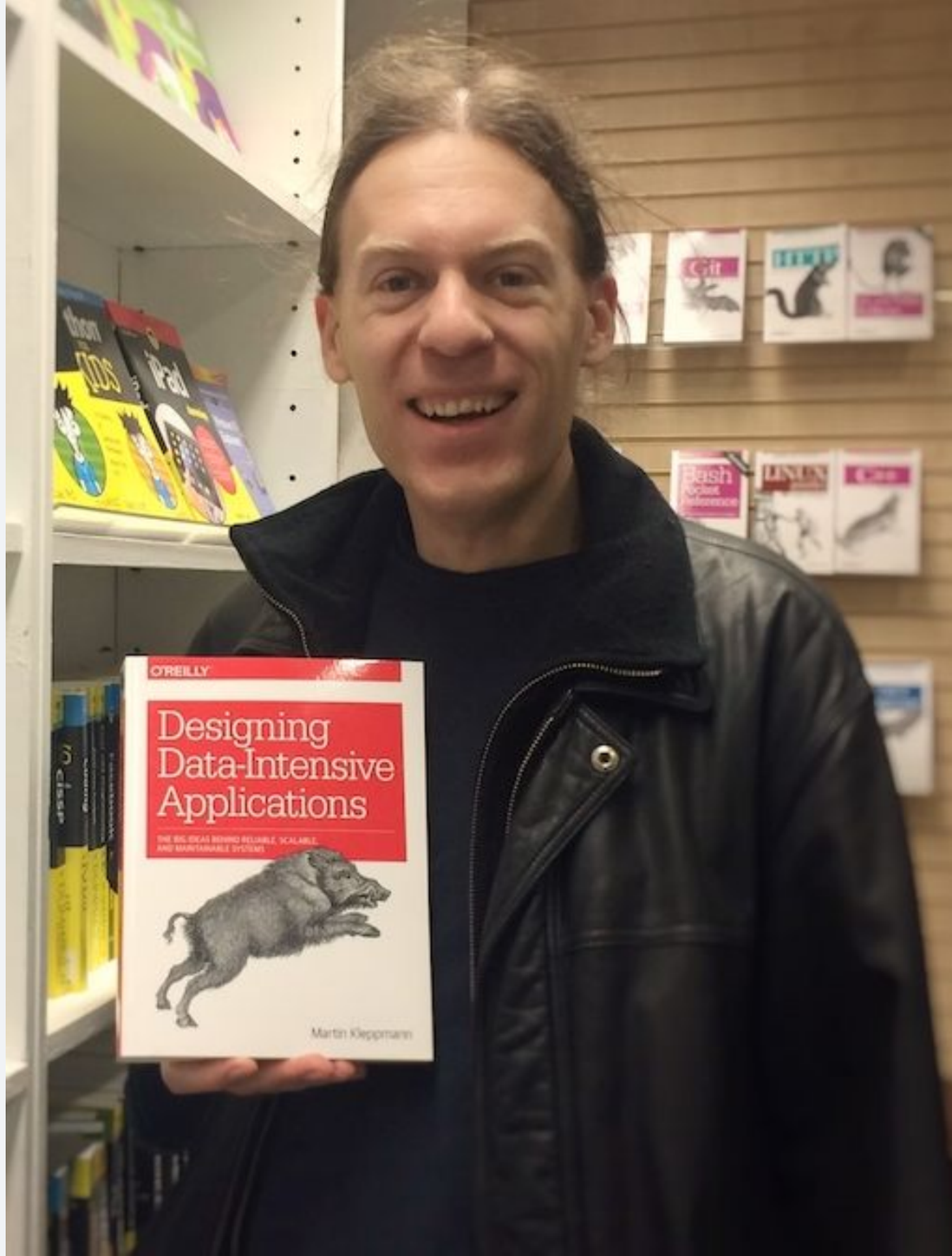
Анонимный интервьюер и доброжелатель



Классический подход



youtube.com/@kleppmann



Курс-легенда. Grokking the System Design Interview

Grokking the System Design Interview

Learn how to prepare for system design interviews and practice common system design interview questions.

Level: None | Study Time: 9h | Lessons: 37

213 479 (Students)

Course Overview

System design questions have become a standard part of the software engineering interview process. Performance in these interviews reflects upon your ability to work with complex systems and translates into the position and salary the interviewing company offers you. Unfortunately, most engineers struggle with the system design interview (SDI), partly because of their lack of experience in developing large-scale systems and partly because of the unstructured nature of SDIs. Even engineers who've some experience building such systems aren't comfortable with these interviews, mainly due to the open-ended nature of design problems that don't have a standard answer.

This course is a complete guide to master the System Design Interview. It is created by Design Gurus who are a team of hiring managers from Google, Facebook, Microsoft, and Amazon. We've carefully chosen a set of questions that have not only been repeatedly asked at top companies but also provide a thorough experience to handle any system design problem.

clck.ru/33zC9L

- System Design Interviews - A step by step guide
- Designing a URL Shortening service like TinyURL
- Designing Pastebin
- Designing Instagram
- Designing Dropbox
- Designing Facebook Messenger
- Designing Twitter
- Designing Youtube or Netflix
- Designing Typeahead Suggestion
- Designing an API Rate Limiter
- Designing Twitter Search
- Designing a Web Crawler
- Designing Facebook's Newsfeed
- Designing Yelp or Nearby Friends
- Designing Uber backend
- Designing Ticketmaster

INTERACTIVE COURSE

Grokking Modern System Design Interview for Engineers & Managers

Intermediate | 175 Lessons | 26h | Certificate of Completion

Takeaway Skills

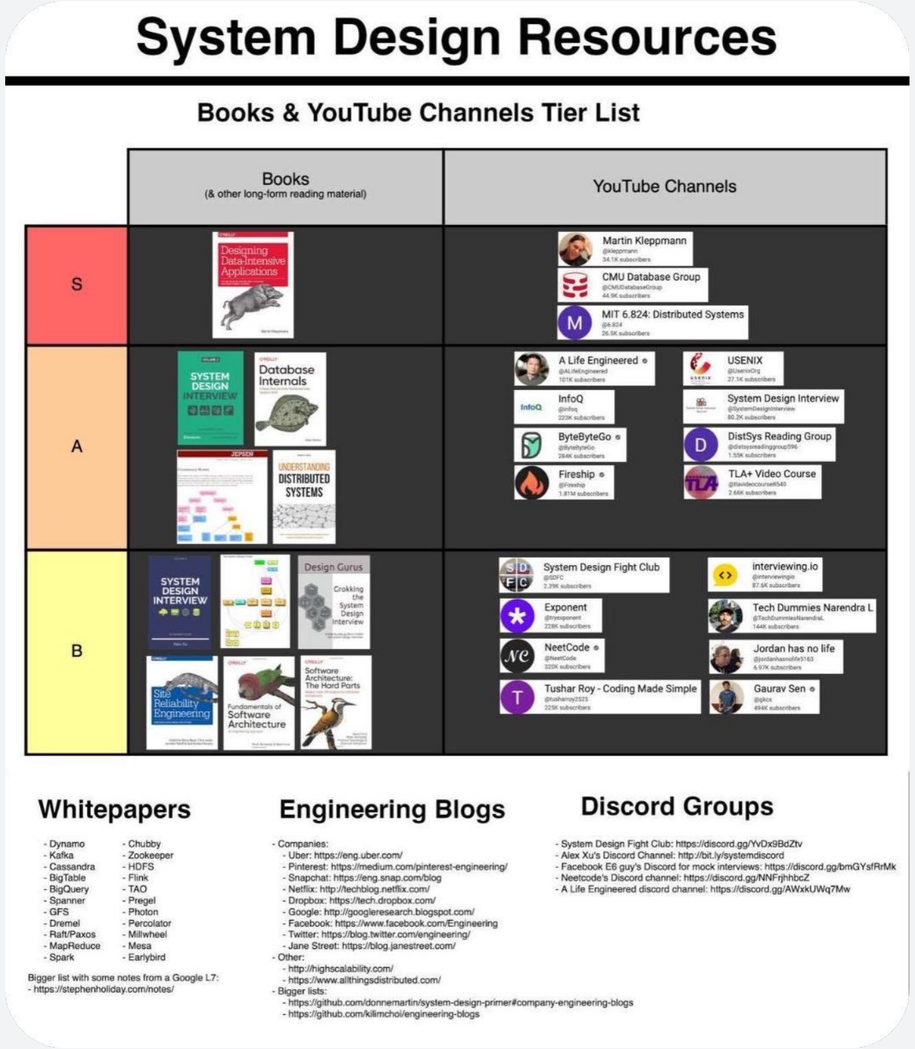
- A modern perspective on designing complex systems using various building blocks in a microservice architecture
- A highly adaptive framework that can be used by engineers and managers to solve modern system design problems
- The ability to solve any novel problem with a robust system design approach using this course as a North Star
- The ability to dive deep into project requirements and constraints
- An in-depth understanding of how various popular web-scale services are constructed

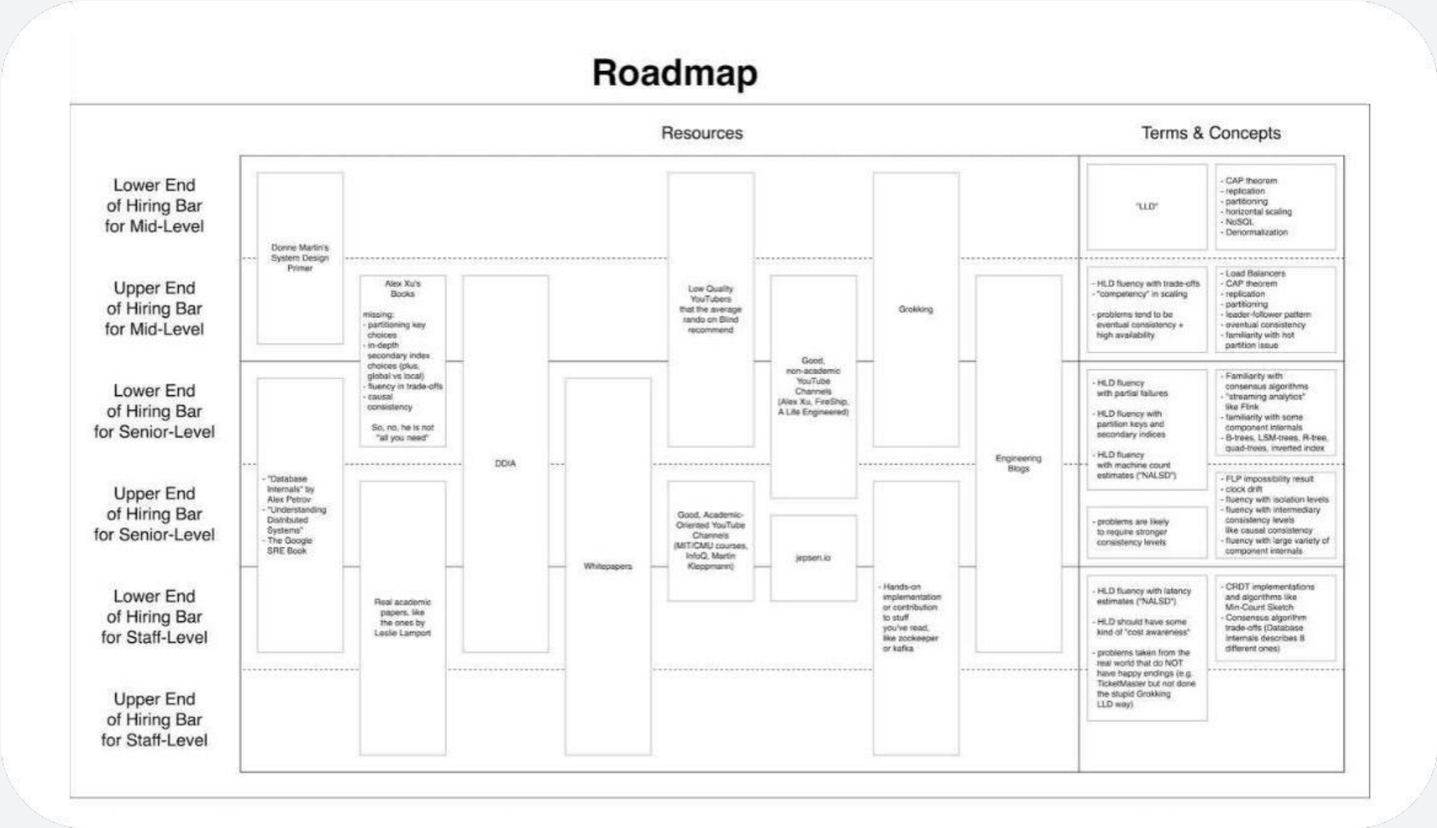
clck.ru/33zD3w

Современный подход

- Martin Kleppmann - DDIA
- Alex Xu - System Design Interview (Volume 1, Volume 2)

Хардкорный подход

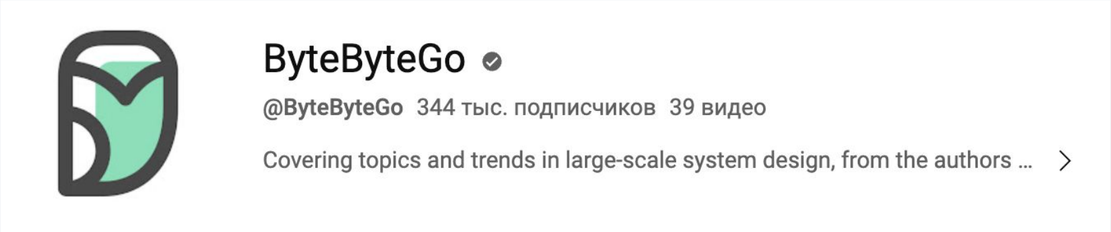




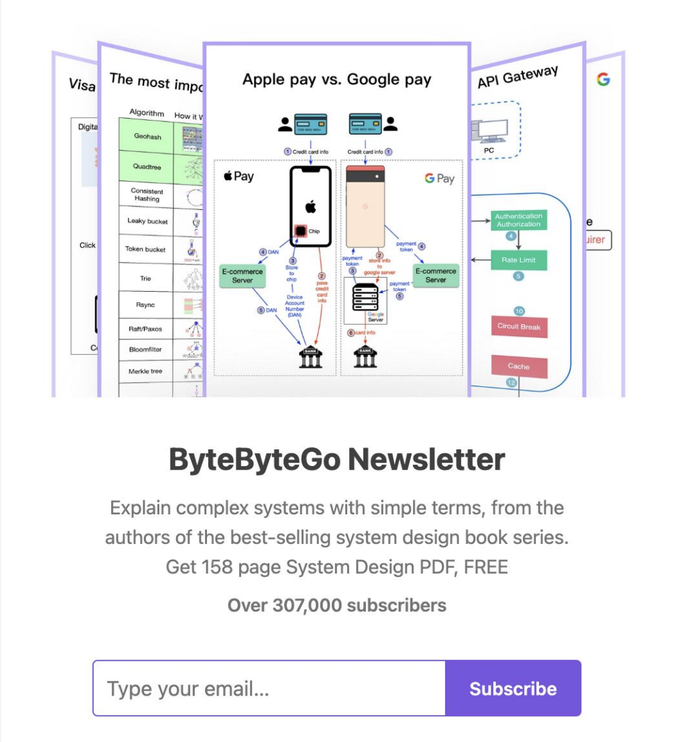
imgur.com/dZEbZaT

Как изучить System Design

Для тех, у кого есть максимум 5 минут в день

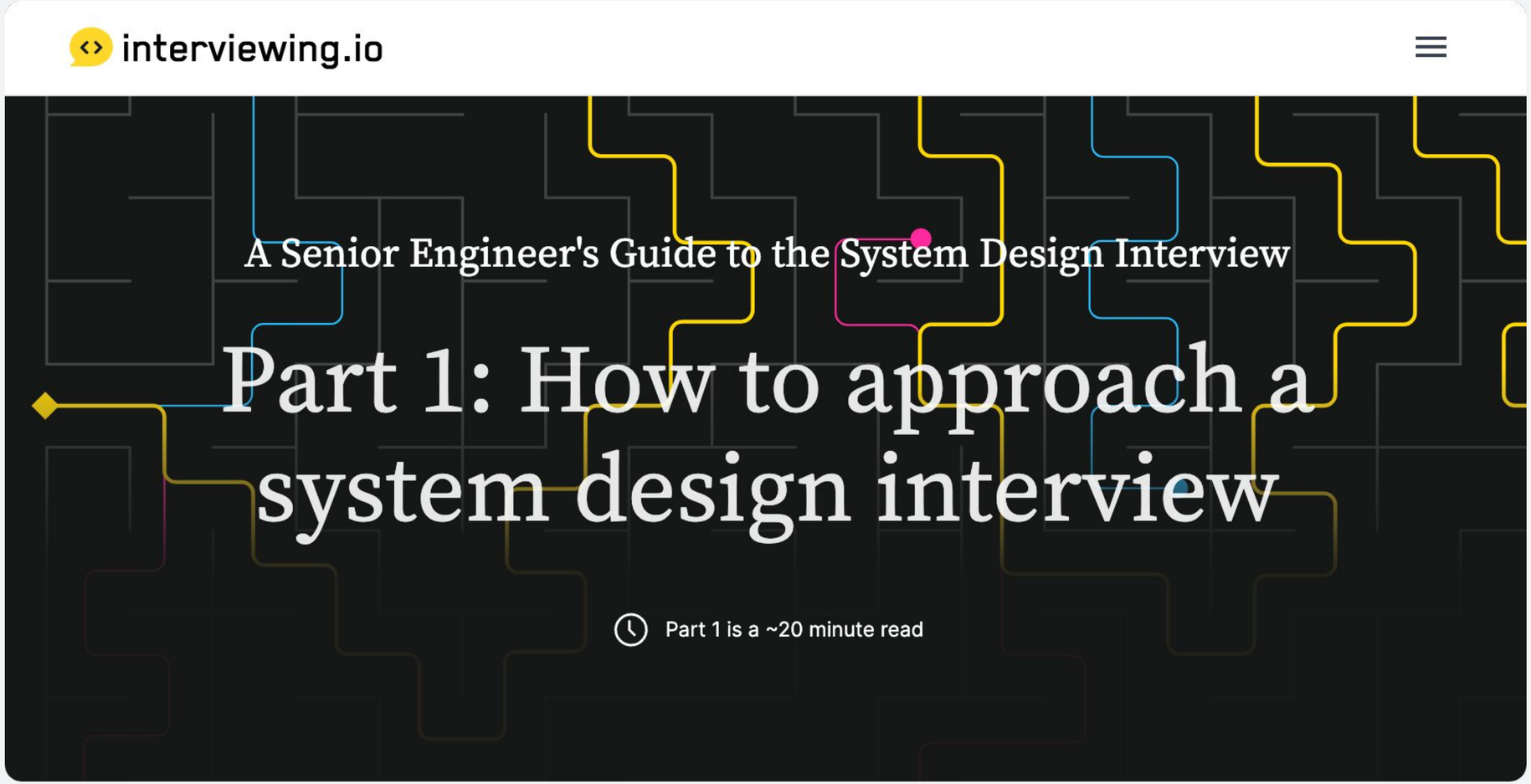


https://www.youtube.com/@ByteByteGo



➔ blog.bytebytego.com/

Полезный гайд для FAANG



interviewing.io/guides

Действительно полезные каналы на YouTube

Exponent
@tryexponent 235 тыс. подписчиков 449 видео
Want to land your dream tech career? Exponent is an online community, co...

Gaurav Sen
@gkcs 502 тыс. подписчиков 267 видео
This channel is for software developers looking to get better at algorithms,...

interviewing.io
@interviewingio 87,8 тыс. подписчиков 152 видео
Book anonymous interviews and mentorship sessions on interviewing.io, a...

A Life Engineered
@ALifeEngineered 105 тыс. подписчиков 24 видео
The mission of A Life Engineered is to help software and other knowledge ...

Jordan has no life
@jordanhasnolife5163 7,88 тыс. подписчиков 131 видео
Hey gigachads, welcome to the channel - I was doing a bunch of my own s...

System Design Fight Club
@SDFC 4,73 тыс. подписчиков 43 видео
System design interview coverage by a FAANG Sr SWE that has reviewed o...

Максимальный реализм (MANGA)

System Design Interview Channel

System Design Interview

@SystemDesignInterview 82,1 тыс. подписчиков 7 видео

Preparing for a system design interview? Or simply want to know how distr... >

Course syllabus

System requirements (functional and non-functional requirements)
Functional requirements (how to define, working backwards approach)
High availability (time-based and count-based availability, design principles behind high availability, processes behind high availability, SLO, SLA)
Fault tolerance, resilience, reliability (error, fault, failure, fault tolerance, resilience, game day vs chaos engineering, expected and unexpected failures, reliability)
Scalability (vertical and horizontal scaling, elasticity vs scalability)
Performance (latency, throughput, percentiles, how to increase write and throughput, bandwidth)
Durability (backup (full, differential, incremental), RAID, replication, checksum, availability vs durability)
Consistency (consistency models, eventual consistency, linearizability, monotonic reads, read-your-writes (read-after-write), consistent prefix reads)
Maintainability, security, cost (maintainability aspects (failure modes and mitigations, monitoring, testing, deployment), security aspects(CIA triad, identity and permissions management, infrastructure protection, data protection), cost aspects (engineering, maintenance, hardware, software))
Summary of system requirements (a single list of the most popular non-functional requirements)

[@SystemDesignInterview]

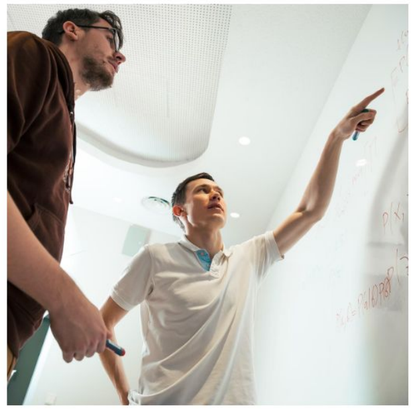
Суровый хардкор (РФ)

kardamanov 24 июня 2021 в 12:27

Как проходят архитектурные секции собеседования в Яндексе: практика дизайна распределённых систем

Блог компании Яндекс, Высокая производительность*, Анализ и проектирование систем*, Промышленное программирование*, Распределённые системы*

✏️ Технотекст 2021



Привет, меня зовут Костя Кардаманов, я работаю в отделе технологий разработки Яндекса. Обычно такой же фразой я приветствую и кандидатов на собеседовании. А сегодня я хотел бы рассказать вам, как и зачем мы проводим интервью по дизайну систем с бэкенд-разработчиками. Сразу скажу: для фронтендеров, мобильных разработчиков и ML-инженеров подобный тип собеседований применим слабо, так что эти специальности мы здесь обсуждать не будем.

Технический уровень кандидата у нас оценивается за счет всего двух типов интервью: секции с кодом и секции дизайна компьютерных систем. Первый тип мы назначаем всем претендентам вне зависимости от их уровня, а вот у кандидатов, которые претендуют на должность старшего специалиста, нужно проверять не только способность писать эффективный и работоспособный код, но и способность разрабатывать сложные системы в целом.

habr.com/564132

Схема изучения

Red Hat · flung

May 3, 2021 832 Comments

Long term prep strategy Part 1 is out: https://www.teamblind.com/post/YA2RKEqY

Coding Part 1 is out : https://www.teamblind.com/post/ikvVrRXa

Part 3 is out : https://www.teamblind.com/post/qubF6fS2

Part 2 is out : https://www.teamblind.com/post/rBrt5bV8

Giving back - how I cleared L6 System Design - Part 3
teamblind

Giving back - how I cleared L6 coding - Part 1 - Google
teamblind

Giving back - how I cleared L6 System Design - Part 2
teamblind

Giving back - long-term interview preparation strategy - Part 1
teamblind

teamblind.com

Подборки на GitHub

https://github.com/donnemartin/system-design-primer

https://github.com/madd86/awesome-system-design

https://github.com/codersguild/System-Design

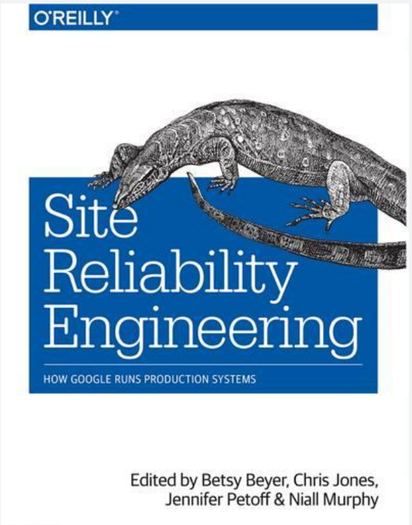
https://github.com/binhnguyennus/awesome-scalability

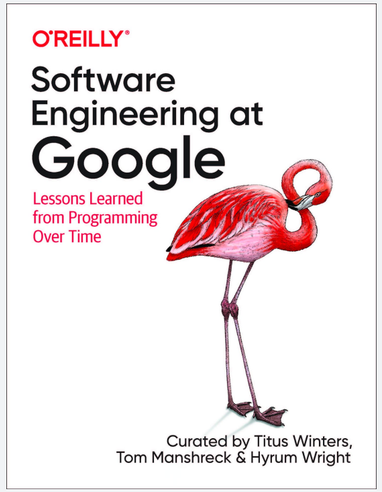
https://github.com/rxin/db-readings

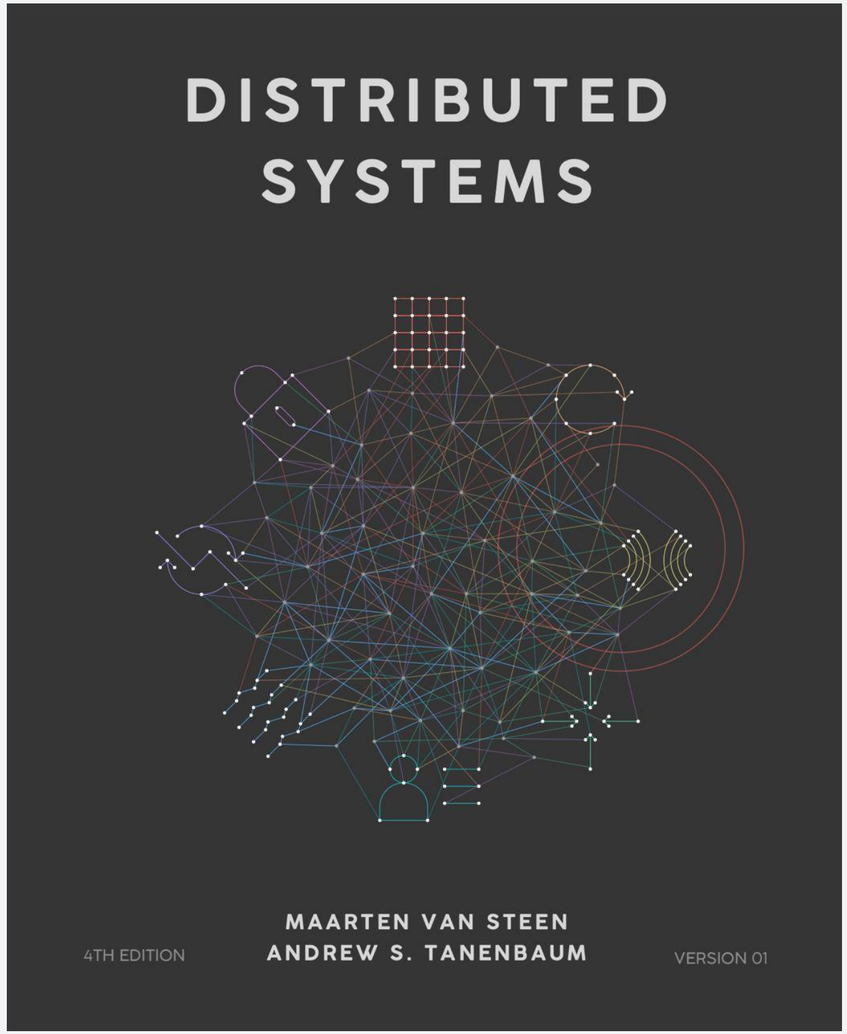
Как стать
Distributed Systems Engineer

Развиваем эрудицию

Читаем книги про SRE и распределенные системы







Блоги компаний

- highscalability.com
- blog.bytebytego.com
- softwareengineeringdaily.com
- quastor.org/
- discord.com/blog/

Супер хардкорный подход

YubiDesu 08/08/2022

Useful Books/Resources:

Most of these books can be found by their ISBNs (with - s removed!) and pirated on libgen.li, but a physical copy is recommended for the better books

- Designing Data Intensive Applications by Martin Kleppmann
  - Amazon: https://www.amazon.com/Designing-Data-Intensive-Applications-Reliable-Maintainable/dp/1449373321
  - ISBN: 978-1449373320
- Database Internals by Alex Petrov
  - Amazon: https://www.amazon.com/Database-Internals-Deep-Distributed-Systems/dp/1492040347/
  - ISBN: 978-1492040347
- System Design Interview (Volume 1) by Alex Xu
  - Amazon: https://www.amazon.com/System-Design-Interview-insiders-Second/dp/B08CMF2CQF/
  - ISBN: 979-8664653403
- System Design Interview (Volume 2) by Alex Xu
  - Amazon: https://www.amazon.com/System-Design-Interview-Insiders-Guide/dp/1736049119/
  - ISBN: 978-1736049112
- Grokking the System Design Interview
  - https://designgurus.org/course/grokking-the-system-design-interview
  - Amazon: https://www.amazon.com/Grokking-System-Design-Interview-Interview/dp/B09NRJT1NF/
  - ISBN: 979-8766433668
  - libgen: https://libgen.li/edition.php?id=138284470
- Grokking the Advanced System Design Interview
  - https://designgurus.org/course/grokking-the-advanced-system-design-interview
  - libgen: https://libgen.li/edition.php?id=138824344
- Donne Martin's System Design Primer
  - https://github.com/donnemartin/system-design-primer (edited)

👏 7

YubiDesu 08/08/2022

Important Whitepapers (WIP, mostly taken from FB E6 guy's channel):

- Dynamo: http://www.cs.cornell.edu/courses/cs5414/2017fa/papers/dynamo.pdf
- Kafka: http://notes.stephenholiday.com/Kafka.pdf
- Cassandra: https://www.cs.cornell.edu/projects/ladis2009/papers/lakshman-ladis2009.pdf
- BigTable: https://static.googleusercontent.com/media/research.google.com/en//archive/bigtable-osdi06.pdf
- BigQuery
- Spanner
- GFS
- Dremel
- Raft/Paxos
- MapReduce
- Spark
- Chubby
- Zookeeper
- HDFS

YubiDesu 08/08/2022

Other useful discord channels:

- Alex Xu's Discord Channel: http://bit.ly/systemdiscord
- Facebvook E6 guy's Discord for mock interviews: https://discord.gg/bmGYsfRrMk

Useful YouTube Channels:

- The Facebook E6 guy: https://www.youtube.com/channel/UCJ8590hU1VY8YcZb7k5-IhQ/videos
- ByteByteGo (Alex Xu): https://www.youtube.com/channel/UCZgt6AzoyjslHTC9dz0UoTw
- InfoQ: https://www.youtube.com/nctw/videos
  - covers Facebook's TAO architecture: https://www.youtube.com/watch?v=sNlvHttFjdI
- Jordan Has No Life: https://www.youtube.com/channel/UCbvDQKzAJ5GwCjTrv4FWkxg/videos
- Usenix: https://www.youtube.com/c/UsenixOrg/videos
  - covers Netflix Zuul Push: https://www.youtube.com/watch?v=6w6E_B55p0E
- MIT distributed systems course: https://youtube.com/channel/UC_7WrbZTCODu1o_kfUMq88g (edited)

YubiDesu 08/13/2022

Engineering Blogs:

- Companies:
  - Uber: https://eng.uber.com/
  - Pinterest: https://medium.com/pinterest-engineering/
  - Snapchat: https://eng.snap.com/blog
  - Netflix: http://techblog.netflix.com/
  - Dropbox: https://tech.dropbox.com/
  - Google: http://googleresearch.blogspot.com/
  - Facebook: https://www.facebook.com/Engineering
  - Twitter: https://blog.twitter.com/engineering/
  - Jane Street: https://blog.janestreet.com/
- Other:
  - http://highscalability.com/
  - https://www.allthingsdistributed.com/
- Bigger lists:
  - https://github.com/donnemartin/system-design-primer#company-engineering-blogs
  - https://github.com/kilimchoi/engineering-blogs (edited)

👍 3

August 25, 2022

YubiDesu 08/25/2022

NALSD / Machine Count Estimation (WIP):

- Site Reliability Engineering: How Google Runs Production Systems
  - Amazon: https://www.amazon.com/gp/product/149192912X/
  - ISBN: 978-1491929124
- The Site Reliability Workbook: Practical Ways to Implement SRE
  - Amazon: https://www.amazon.com/gp/product/1492029505/
  - ISBN: 978-1492029502
- Links:
  - https://us.teamblind.com/s/rRZMVFin
  - https://softwareengineering.stackexchange.com/questions/384732/
  - https://leetcode.com/discuss/interview-question/1161605/Machinesservers-calculation-on-System-Design (edited)

👍 6

YubiDesu 09/05/2022

Interview Leveling Expectations (WIP):

Amazon L5:
- Amazon doesn't seem all that crazy about back-of-the-envelope estimations
- you should be able to pick a database solution and justify your choice
- they usually like to ask custom questions that aren't from grokking/alex xu books
- L5 hiring bar sometimes doesn't require system design round within AWS

Amazon L6:
- Amazon doesn't seem all that crazy about back-of-the-envelope estimations
- cut through *ambiguity* proficiently
- proficiently identifying and talking through *partial failure scenarios* is probably very important
- being familiar with "two-phase commit" is probably sufficient, in terms of consensus algorithms

Facebook E5:
- capable of "identifying important numbers" (back-of-the envelope calculations)
- you generally just need to review all the questions from grokking, IIRC
- they might exclusively ask the standard grokking/alex xu list of design questions

Google L5:
- you might be asked to do latency estimates or machine count estimates (somebody may have once been asked to do the machine count estimates for each microservice of "web crawler" problem)
- Google seems to like it when you can do stuff like explain/recite/"regurgitate" (😂) how paxos works... It's frowned upon to outright ask how paxos works, but it seems like it'd be hard to get a strong hire unless you proactively recite such trivia 🤔

Overall things that companies might expect:
- ability to navigate/cut through "ambiguity"
- partial failure scenarios
- API request & response contracts
- DB schemas
- pick a database solution for a data store & justify your choice
- latency estimates
- machine count estimates
- iteratively identify bottlenecks and address them
- "end-to-end full system HLD + deep dive of one component" (do I need to do *all* of these things for *every* component?) (edited)

👍 19

September 6, 2022

YubiDesu 09/06/2022

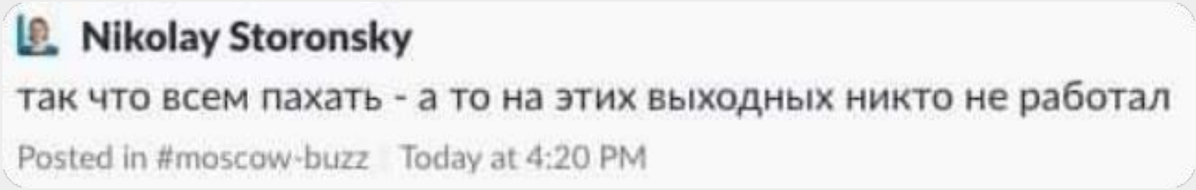
Interview Leveling Expectations, continued (WIP):

Overall tips:
- ask for feedback afterwards (some companies don't, but it doesn't hurt to try)
- formally do the FRs and NFRs step even if it's a grokking problem / "you've seen it before" (some companies have a criteria of "thinking it out" / "careful up-front planning" before jumping right into the problem)
- behaviorals questions can show "scope" (Amazon does this. Also, pinterest gave me mid-level instead of senior-level primarily because I had no stories of leading a project of 5+ engineers despite "doing incredibly well on system design")
- "lead" the interview (instead of "what should I deep dive next?" or "what would you like to prioritize?", try something like "next I'm going to deep dive the partial failure scenarios of the aggregator, is there something you'd prefer to prioritize instead?")
- Reciting trivia and arcane tricks like R-trees *does* seem to impress people.
- They can't *ask* for trivia because that's frowned upon. But, proactively regurgitating trivia seems to score you extra points or a strong hire. 🤔 This seems to me like a workaround to how outright asking trivia is frowned upon. (edited)

🙏 4

Practice

Практика



Мок-собеседования (Mock Interview)

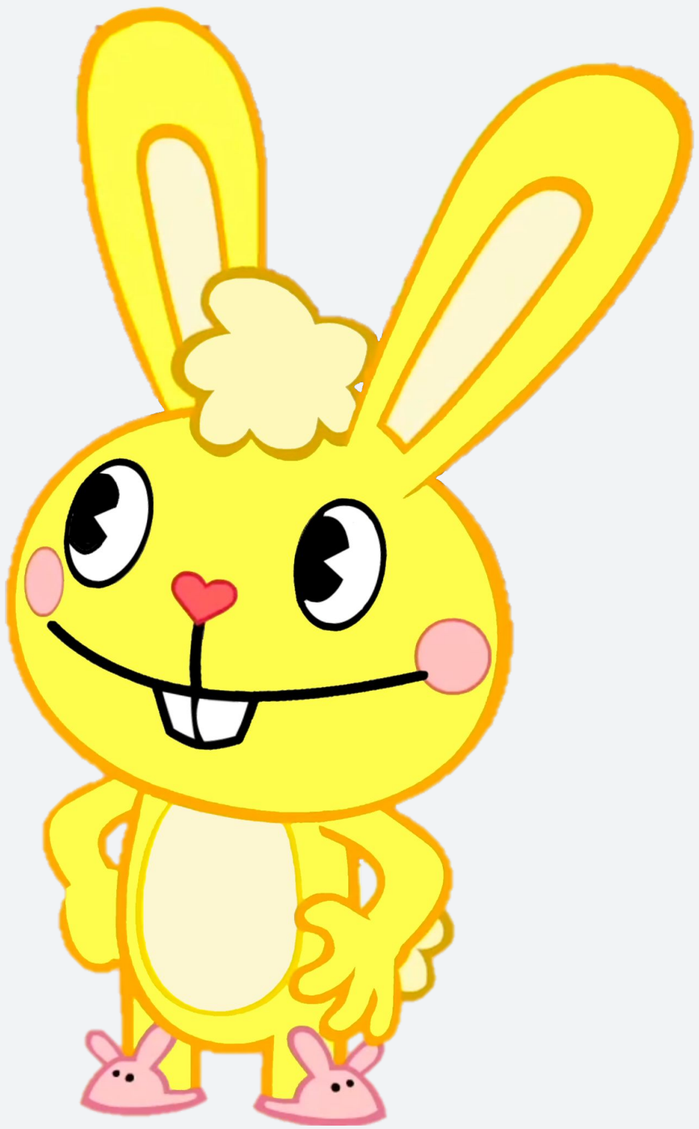
- pramp.com
- tryexponent.com
- interviewing.io
- interviewbit.com
- preplaced.in
- workat.tech
- Discord и Telegram-каналы с энтузиастами

Пет-проекты

Стартапы

Сообщества

- t.me/sysdesign_interview
- {между скобок}
- t.me/system_design_interviews

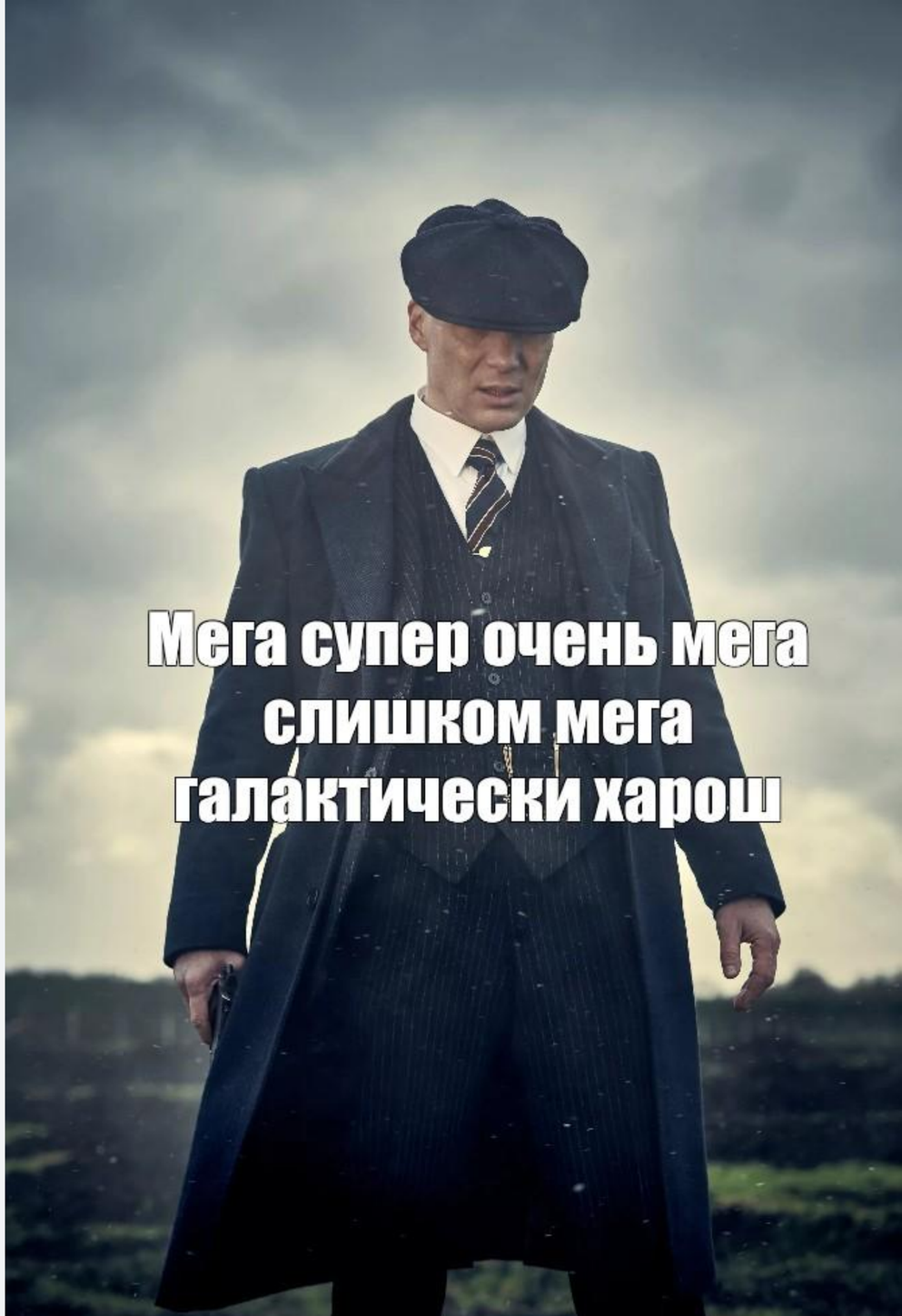


System Design Roadmap

github.com/
system-design-roadmap

Выводы

- ➔ System Design пригодится не только для собеседований, но и в жизни
- ➔ Собеседование тестирует: коммуникацию, навыки проектирования, понимание распределенных систем, опыт работы с ними и масштабирования
- ➔ Хороший интервьюер быстро определит ваши слабые стороны - поэтому знания качать надо не только в глубину, но и в ширину
- ➔ Залог успеха - это правильная подготовка: знания и практика; не стесняйтесь больше практиковаться



Спасибо!

Владимир Маслов

Руководитель команды разработки, X5 Tech

t.me/senior_unicorn