EP 14 - How to Prepare for System Design Interviews?

The best way to prepare for System design interviews with topics and resources

JAVINPAUL
AUG 20, 2023 · PAID

8 2 Share

Hello guys, if you are preparing for a Software Engineer interview then you may know how difficult it is to prepare for System Design given its open-ended nature.

In the Software Engineering world, if you are applying for a Senior Engineer / Lead / Architect / or a more senior role, System Design is the most sought-after skill, and hence one of the most important rounds in the whole process. If you mess this up, nothing else would matter.

If you get it right though, you're looking at a raise of at least tens of thousands of dollars annually. So how do you ace your system design round? Well, for that only I have created this guide where I have shared all essential topics, concepts, and resources you can use to prepare for System Design interviews.

In the past, I have shared the **best system design courses**, System Design Books, System Design Cheat Sheets, the best websites to learn System Design as well as the **best software design questions** you can also check them out for better preparation.

But, Before we get into the details though, let's find out **what is a system design interview and** What do the interviewers really expect from the candidates?

What is Expected from Candidate during System Design Interviews?

The first step to preparing for a System design interview is to understand what is expected from you. Once you know that you can align your preparation strategy to meet those expectations.

So here are the key expectations from you on a System design interview:

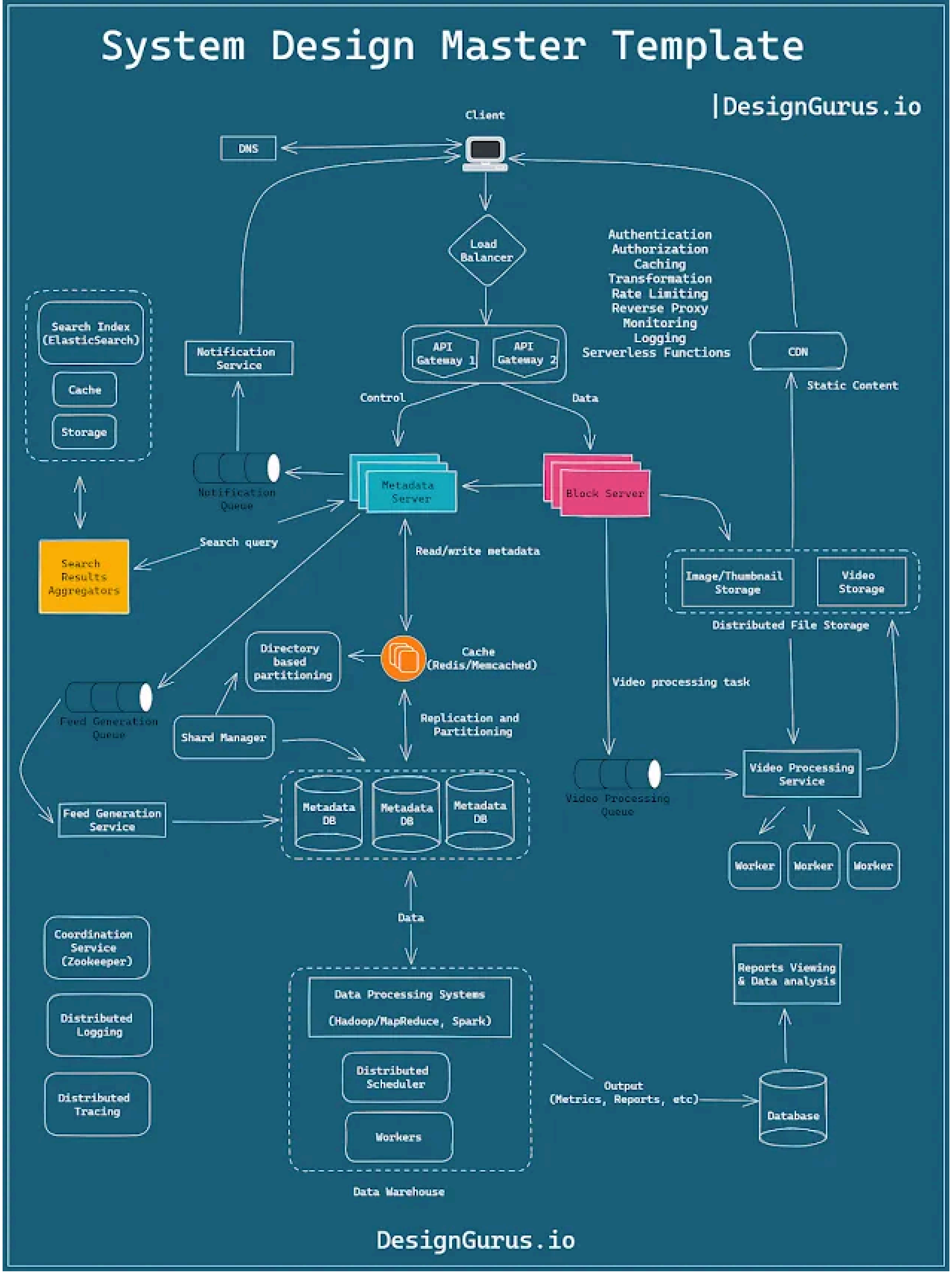
1. You should be able to design a system that satisfies the requirements given to you and scales well, for example, design a vending machine or design a trade position aggregator.
2. Your design should be pluggable and not restrict the addition of new features.

3. You should be able to compare various alternatives and choose the most optimal one. Things like, which database is the most important, which protocol should you use, or what's the best approach to scale a system, etc.
4. You should know the basics that are relevant from a system design standpoint like:
    a. Load balancers
    b. APIs
    c. Caches
    d. Databases
    e. Network Protocols
    f. Message queues
    g. CDNs
    h. High-level details about ML and Big data
    i. CAP Theorem
    j. Monitoring and analytics

By the way, don't get overwhelmed by a number of topics, most of the good System design courses like **System Design by CodeKarle** on Udemy or **Grokking the Modern System Design Interview on Educative** cover all these topics.

4 Steps to Prepare for System Design Interviews?

Here's how you can crack the System design round interview of any FAANG company (Facebook/META, Amazon, Apple, NetFlix, and Google) or get into FAANG



1. Learn Essential System Design Concepts

Any system design interview will definitely require you to come up with a basic high-level design for whatever system you are trying to build. There are some components that will be needed for sure. Make a note of these components.

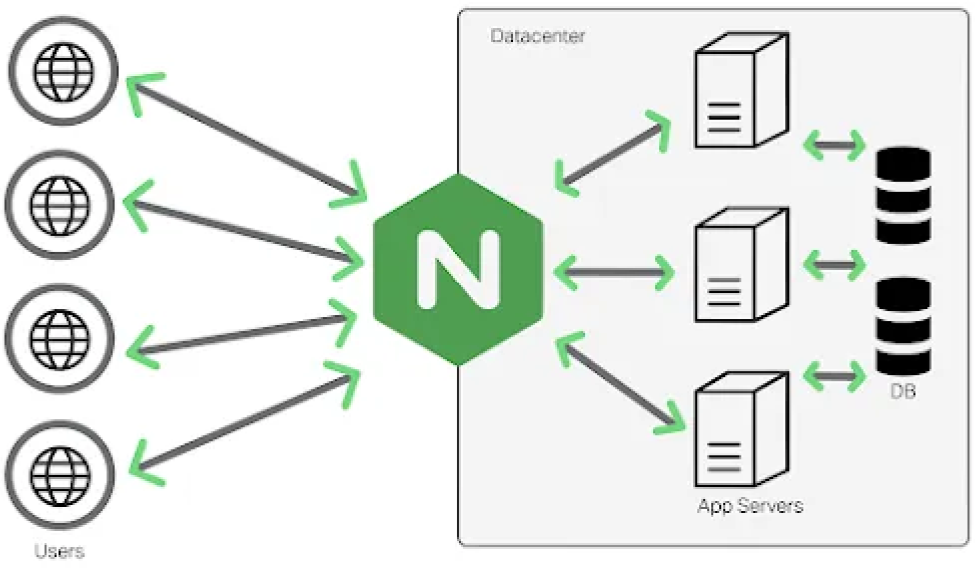
Here are a few -

1. 1 Load Balancer

You cannot have a distributed system without a load balancer to distribute the incoming requests among various nodes. This ensures proper resource utilization and that there is no single point of failure in your system.

While learning about Load balancers it's also a good idea to learn about **API Gateway** and carefully understand the *difference between load balancer and API Gateway*, which is quite important from an interview point of view. This question has been already asked me a couple of times in interviews.

Here's how Nginx does it.



1.2 Cache

Most systems have some read-heavy interactions, some information that the user will access frequently but not update as much. It makes sense to cache this information in such a way that it can be easily fetched without the need for a DB lookup.

Think **low latency**. Also, based on your use case, you might need to store more frequently accessed information or more recently accessed information. So, read up on various **eviction policies**.

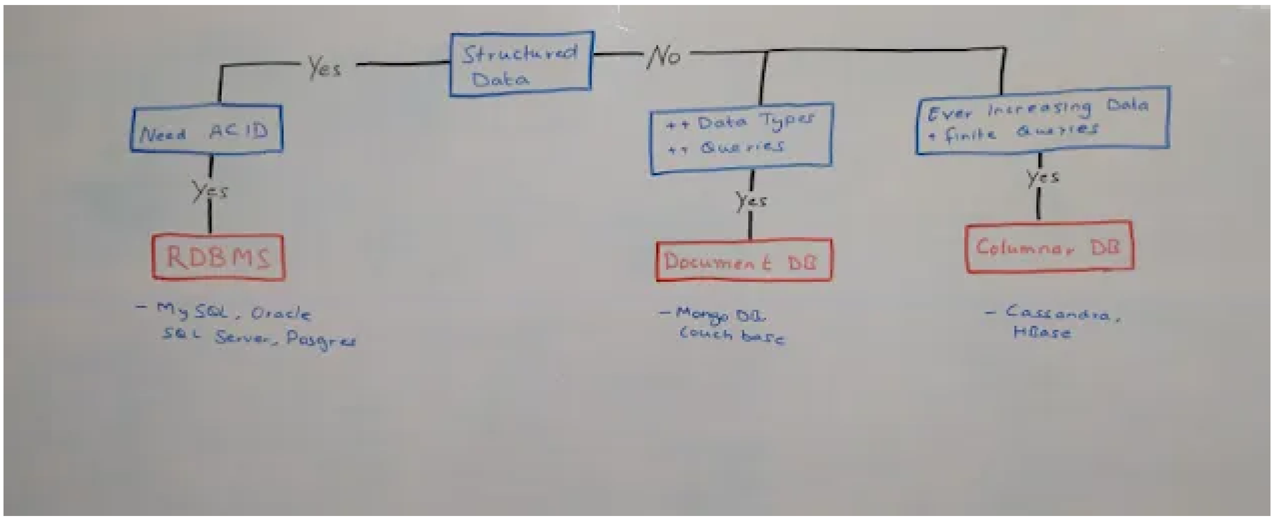
If you want to learn more then the **ByteByteGo System Design course** has a lot of information on Caching strategies and how to effectively use caches. You can refer to them to learn more.

1.3 Database

Again, there can be no system without some form of data storage. Whether you want to store files, images, product information, financial transactions, or simply dump all the data from various user interactions to run analytics later.

It all needs databases. So read up on it. Find out what matters when you are selecting a database, read about SQL/NoSQL, query patterns, and how the CAP theorem might come into play while making tradeoffs.

Sandeep has also explained this quite well in his System Design course on Udemy, if you want to learn further, you can refer to that as well.



1.4 Message queues

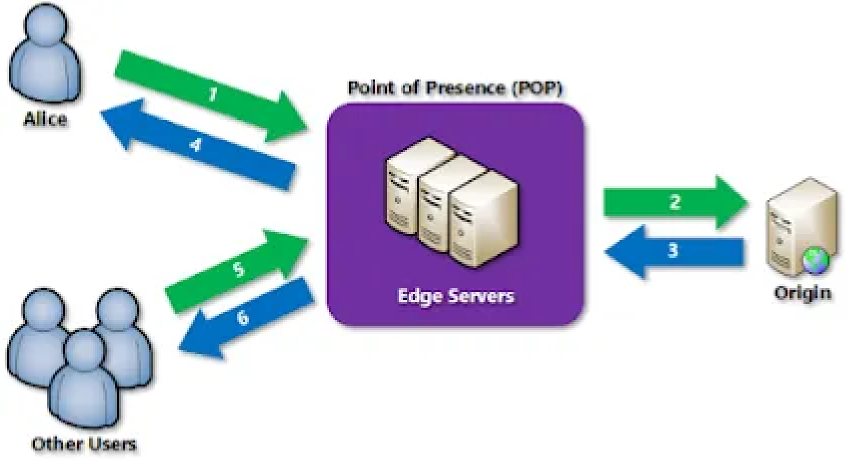
Sometimes our system needs to perform some tasks that need to be done but not necessarily immediately, or their outcome does not affect the user's journey. In such cases, rather than making a service call, and waiting for the response, it makes sense to just write the message to the queue so that it can be executed later.

What if you need to insert information in your database, and bulk insert could be more efficient? It would make sense to just keep track of these inserts in a message queue and perform 1 bulk insert instead of hundreds of 1-to-1 inserts to optimize your resources.

A good knowledge of Messages queues like RabbitMQ, ActiveMQ, and Apache Kafka is essential for any Software Engineer and this is one topic which is covered quite well in Educative's Grokking Modern System Design for Software Engineers & Managers course. If you want to learn about not just message queues but other essential System design topics, this is one resource that I personally use.

1.5 CDN

When your users are distributed geographically, getting your content to them in a reasonable amount of time becomes a real challenge. CDNs allow us to maintain a copy of our data in various data centers located closer to the users' location to reduce the latency. Here is a short video about how Akamai does it.



1.6 Analytics and Monitoring

This is something that is needed in every system you create. This is a hidden requirement, no one calls it out in the requirement gathering but every interviewer wants this. User logs in or log out? Wishlisted an item? Did payment fail? It is all the information for us! If anything of importance happens, fire an event and save it in your messaging queue.

You can perform real-time analytics on data or just dump it in a Hadoop cluster to use later. Similarly, if an API call is regularly failing, or if your servers are about to run out of resources, wouldn't you like to know of it beforehand?

A good knowledge of tools like Grafana, and Prometheus can help you with Analytics and monitoring and also impress your interviewer during a system design interview. Alex Xu has also shared a lot of good information on Monitoring in his book and ByteByteGo course, you can refer to them as well.

1.7 Network Protocols

Based on your requirement, based on the type of content you are sharing, you might need to make a decision on which network protocol to use. Read up on various network protocols and when are they relevant, what might be the compromises you might need to make, etc.

The first step is to know about all these concepts. And by knowing concepts, I don't mean just some theoretical knowledge of what these are, but more practical hands-on experience about what should be used and when.

You need to know things like:

1. Which is the best alternative to choose from, given the use case?
2. What are the tradeoffs that you need to consider while making these decisions?
3. Best practices for certain use cases.

To know most of these things, I'll recommend going through this course on System Design by CodeKarle, which covers all of the above with concrete examples from the real world.

2. Learn from the tech giants (read their engineering blog)

This is probably not going to help in the short term. But in the long term, to become an expert in System Design, it's best to look at the Tech blogs of various tech companies and see how they are solving various technical problems.

This would paint a clear picture of the real problems that they face and how innovatively they solve them. Understanding these things would help you become better at system design and also keep you up to date with the latest innovations in tech.

Some of the best blogs to follow are:

1. Facebook engineering blog
2. Netflix tech blog
3. Uber engineering blog

3. Solve Frequently asked System Design questions

An obvious way to get started with your interview prep! Knowing the basics is definitely important, but it won't be enough! The most basic way to get started with your practice is to look at some frequently asked questions and their solutions.

Most system design interviews revolve around some 5-6 commonly asked System design questions and if you know the solutions to those, you are more than likely to clear this interview.

The most common questions are:

1. Tinyurl System Design
2. Twitter System Design
3. Facebook System Design
4. Whatsapp System Design
5. Airbnb System Design

6. Uber System Design
7. How to design Amazon Prime Video
8. How to design Google Search
9. How to design NetFlix
10. How to Design a Distributed Message Queue

If you need resources to solve these questions, something which not only solves the question but also explains the underlying concepts and approach to solving system design questions then this highly-rated course by CodeKarle discusses most of these case studies and some more problems, which has helped many people crack their interviews for companies like Google, Facebook, Microsoft, Amazon, etc.

4. Practice, Practice, and more Practice

Practice, practice, practice! Did I say practice? There are a lot of resources out there. I have shared a few that discuss some of the most popular system design problems in detail. Once you go through a few of them you will start noticing a pattern and will soon be able to come up with solutions on your own.

Get a better understanding of how your systems are designed in your organization. How are the other teams doing things? **What factors do they take into consideration?** The next best thing is to practice with a friend. Make sure you go through a few mock interviews before your actual interview to avoid some common but easily avoidable pitfalls.

Some most common mistakes that I have seen people make are:

1. Not driving the interview
2. Not asking questions
3. Not structuring the interview properly
4. Running out of time
5. Not considering the requirements
6. Not exploring all the alternate design options

All these mistakes can be easily avoided by having a few mock interviews with someone who knows System Design well. And time your interviews. The target must be to reach a solution within 40 minutes, including time for some discussion.

For mock interviews, you can check out the **Exponent website**, which offers 1–1 coaching and classes for in-person learning and also run a free service called Pramp that offers peer-to-peer interviews for software engineers in data structures and system design.

Best System Design Interview Resources

If you need more resources like books, and online courses to prepare for the System Design Interview here are my recommendations:

- **CodeKarle's System Design Interview Course on Udemy**
- **Grokking the System Design Interview on DesignGuru**
- **Mastering the System Design Interview by Frank Kane**
- **Software Design and Architecture Specialization** [Coursera]
- **System Design Interview course and Mock Interview by Exponent**
- System Design Interview Course on ByteByteGo by Alex Wu
- **Web Application & Software Architecture 101** [**Educative.io**]
- **Pragmatic System Design** [**Udemy Course**]
- Grokking Modern System Design for Software Engineers & Managers

That's all about **how to prepare for System Design Interviews**. We have discussed essential System Design Interview topics, concepts as well as popular System Design questions for practice. With this 4 step process, you'll soon be ready to ace any of your system design interviews! Hopefully, this should be a good starting point for you.

Happy learning! and all the best for your System design interview

Other System Design Resources You may like

If you need more resources to prepare for the System Design Interview here are my recommendations:

- 20 System Design Interview Questions with Answers
- Top 5 Websites to learn System Design in depth
- Is Grokking the System Design Interview worth it?
- Is DeisgnGuru's System Design course worth i?
- 6 Best Courses to Learn Dynamic Programming
- Is ByteByteGo a Good Place to learn System Design
- My favorite courses to learn Software Architecture
- Is Grokking the Advanced System Design course worth it?
- 3 System Design Cheat Sheet You can refer for interviews
- 10 Free System Design Courses for Beginners and Experienced

- Top 5 System Design Courses from Udemy
- 30 System Design Interview Problems for Practice

Thanks for reading this article so far. If you like this System design interview preparation article then please share it with your friends and colleagues who are preparing for tech interviews. If you have any questions or feedback then please drop a note.

All the best with your Tech interviews, may God bless you with growth and success.

8 Likes · 2 Restacks

← Previous Next →

Comments

Write a comment...