CloudWay Digital Inc.
cloudwaydigital.com

Interviewing Senior Technical Talent

How to Interview Staff/Principal Software Engineers, Technical Leads, and Software Architects

✅ -- Free Excerpt -- ✅

Copyright Information

Corporate Use

For corporate, or other organizational use, please contact
licensing@cloudwaydigital.com

Disclaimer

Table of Contents

Let's Get to It! 🚀

Who is This Guide For ?

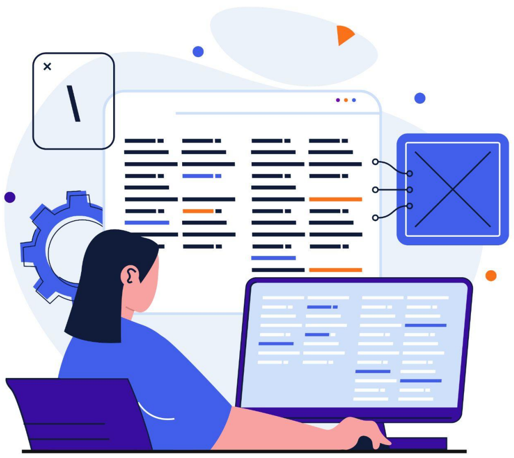
This guide is for any team that wants to improve how they are conducting interviews for very senior technical candidates (think of Staff and Principal engineers, Software Architects, Technical Leads, and similar roles).

The audience of this guide is aimed at both individual contributors (software developers and architects) as well as software development managers who will be taking part in the interviewing process. You are well positioned to get high value from this guide especially if you are interviewing a very senior candidate for your immediate team.

You could also be a very young team of individuals who are at the earlier stages of your career journey. You might not have the best track record or you might have very little experience in conducting interviews for roles that are above the junior/intermediate levels. This would especially be relevant to startups who are moving from an MVP to growth/scaleup mode. In that case, the included Bonus Guide should be of help.

You could also be an experienced team or hiring manager who simply wants to elevate your interviewing game when it comes to evaluating candidates for these very senior technical roles.

If any of the above rings familiar - then this guide is for you!



The Questions

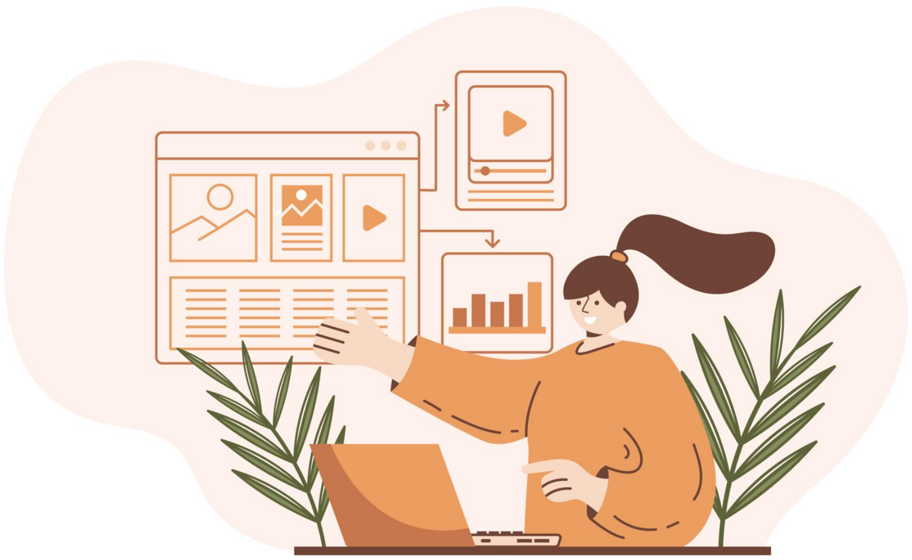
"How would you design [ABC]?" 💻

Technical candidates of the Staff/Principal, Architect, and Lead levels need to work and think widely. They should be able to have a strong grasp of complex system design and architecture topics, alternatives, tools, and technologies.

Most importantly, they should also be aware of the limitations of their knowledge and know how to answer questions that they might not have an answer to at the moment.

They need to be comfortable enough to say:

"I have never dealt with this before, but here is how I would go about it."



In other words, they need to be avid learners and have the ability (and desire) to dive as deep and as wide into new topics as needed. Bonus points if they are also passionate about teaching these topics that they have just learned to others.

Those very senior folks should foresee problems and pitfalls where others do not.

This type of question provides fertile ground for exploring the candidate's experience, abilities, and a variety of skill sets. Most importantly, it works on several dimensions as it demonstrates how well the candidate can adapt to new information, their problem solving process, collaboration, and communication skills.

The way to ask this type of a question is to slowly build upon a foundation and expand from there.

For example, you could ask the candidate to design some generic system, or even better, ask them to design one of your own systems - one that is currently in existence or one that you foresee coming in the future.

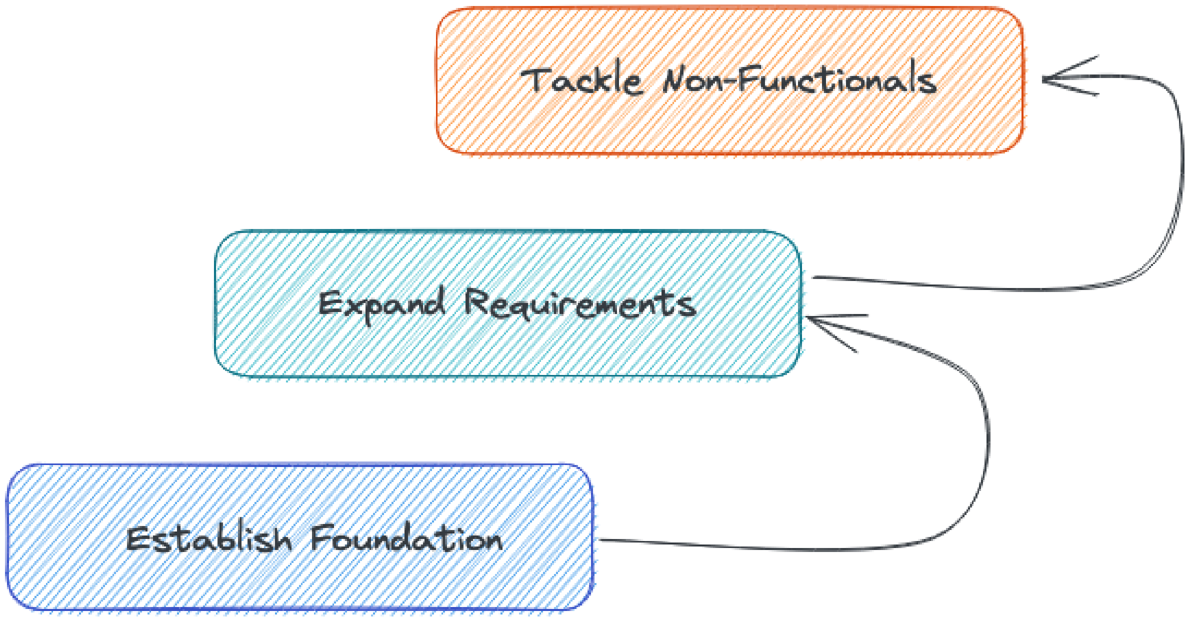
You don't need to break any NDAs here or share more than you can about your organization's specifics. Just point the candidate at an area or business problem that is similar but still generic enough so as not to divulge any confidential information.

The key is to start simple and to add complexity onto the question as you go.

The following illustration is one framework that facilitates this approach.

It consists of three stages:

- ✔ Establish Foundation
- ✔ Expand Requirements
- ✔ Tackle Non-Functionals



Example:

Establish Foundation

☑️ "Say you are building an application that will connect to a set of financial APIs for a particular user account and retrieve the financial details for that user from each API. How would you architect that kind of a system?

You are looking for the candidate to start asking meaningful questions and to break down the problem into manageable pieces.

They should start bringing up things such as

- ✅ Volume of requests and transactions per second
- ✅ Security and auditing of actions
- ✅ Synchronous vs asynchronous ways to integrate with these APIs
- ✅ Ways to integrate and types of third party APIs
- ✅ How often is the integration planned to happen
- ✅ Architectural tradeoffs in designing the system in this or that way

Expand Requirements

☑️ "Let's say that we also need to expose those aggregations to the end user. What would you need to add or change in the application's architecture? What if the application needs to scale to millions of users (hundreds of transactions per second?)"

The candidate should be able to adapt to these new requirements and bring up ways to address them. Ideally, they will be building incrementally on what they were describing prior. Pay particular attention to whether they have awareness of how decisions made earlier can impact the product or team in the future.

Tackle Non-Functionals

☑️ "How would you ensure the security of such an application? How would you deploy it and what kind of infrastructure would you use? How would you make it resilient and highly available?"

Look For:

- ✔ How well can the candidate build upon a foundation?
- ✔ Are they thinking of limitations and edge cases?
- ✔ How do they tackle "unknown unknowns"?
- ✔ How much technical grasp do they have of these topics?
- ✔ Are they only superficially familiar and masking their own lack of knowledge by technical jargon and buzzwords or do they possess depth of knowledge?
- ✔ If only vaguely familiar - are they being honest about that?

Creative Ways to Interview

Putting Them in the Interviewer Seat 💻🤝

Ask the candidate:

"If you were the one interviewing someone else for this role, what kind of technical, or otherwise, questions would you ask to ensure fit?"

This kind of a question might seem strange at first as you essentially are putting the candidate in your seat. What this does though is help uncover the true depth and breadth of the candidate's knowledge.

This is especially useful in situations where you are interviewing a very senior candidate who perhaps is senior to you and so you are unsure of how to actually interview them.

See "**Bonus Guide - How to Interview Someone With More Experience Than You**" For more tips and questions to help interview someone beyond your level of experience

The extent of your experience, being less comprehensive than theirs, creates a lot of "unknown unknowns" in that you do not know what to ask the candidate as you may not have the same level of knowledge that they do.

This type of question allows you to get a full glimpse into the extent of their experience and knowledge without having that experience and knowledge yourself.

For example, let's say you are interviewing someone with a strong emphasis on information security.

Imagine that you are an intermediate software developer interviewing another intermediate developer. If you ask that other developer - "if you were the interviewer for this role, what would you ask the interviewee?" - they would answer something along these lines:

"I would probably ask them whether they know what encryption is. What libraries have they used? How would they store a password and why? What are some common security gaps in Web applications? What are some common encryption algorithms?"

The same question asked from a more senior candidate, would yield something along the following lines:

"I would ask them to describe a common encryption protocol and how it works under the hood. For example, what is mutual TLS, why is it needed and when should you use it? How does it work on a low level? When should you not use it?" What is the zero-trust security practice and is it a viable strategy for distributed applications? How do you achieve a high level of security-mindfulness and strong security posture within an organization and what are some of the key challenges?"

The difference between the two types of question lines is evident. Someone who is less experienced has doubtfully had to deal with team or organization-wide security policies. So they would not have a strong grasp of the zero-trust security paradigm.

They probably also haven't had to implement things like mutual TLS and so aren't aware of why it is needed and how it works. Even less so - about why you would or would not use it.

Since the first developer had no awareness of these, they didn't bring these topics up.

The second developer, being more experienced, did have awareness, and so they did bring up these topics.

The result is that even if you were an intermediate developer yourself, you've now got a pretty solid glimpse into the extent of knowledge for both candidates even though you might not have had full awareness of all of these topics yourself.

The key here is that if the candidate has the same level of understanding of these topics as the interviewer, all of the questions they bring up will match, more or less, what the interviewer already knows.

If, however, the candidate is significantly senior to you (the interviewer), they will raise questions and points that you haven't thought of or haven't come across before.

For bonus points, you can ask the candidate to explain some of these concepts to you and see how they handle it.

There is a Lot More…

The Full Guide Also Includes:

✅ **+7** more in-depth questions with detailed explanations
✅ **+5** creative ways to interview with comprehensive guidelines

… And the Following:

Roles

- ✅ Role Expectations
- ✅ Common Role Types
- ✅ Defining the Role
- ✅ Role Responsibility Template

Interview Structure

- ✅ General Mindset
- ✅ Interview Format and Session Breakdown
- ✅ Who Must Participate

Questions and Their Breakdown

- ✅ General Question Structure
- ✅ Question Categories and What They Reveal

After the Interview

- ✅ Things to Avoid
- ✅ Getting the Right Candidate Interested

Bonus Guide - Interviewing Someone More Experienced Than You

- ✅ Eliminating Fear
- ✅ The Right Mindset
- ✅ 5 Comprehensive Questions with Details and Context
- ✅ How to Get a Full Understanding of the Candidate's Fit for the Role

Get the Full Guide 🚀

CLOUDWAY DIGITAL

CloudWay Digital Inc. is a technology consulting agency that focuses on three pillars.

- ✅ Helping teams with software architecture, governance, and establishing development best practices
- ✅ 10x'ing processes between business and technology to make these significantly more structured, organized, productive, efficient, and delightful for all involved
- ✅ Coaching and mentoring for software developers and architects on an individual level as well as on team level

Visit us at https://www.cloudwaydigital.com/

https://www.cloudwaydigital.com/blog

Or

Give us a shout at hey@cloudwaydigital.com