



The Interview Delivery Playbook

What to say, and when, in every round of a
senior software engineering loop

Free sample

A free sample

This is one chapter of The Interview Delivery Playbook: the 45-minute plan for a system design round. Practise it out loud against a timer before your next round.

Loop Ready is independent and is not affiliated with, endorsed by or sponsored by any company named here. The examples are our own illustrations, not any company's rubric. Nothing here guarantees an offer.

© 2026 Loop Ready.

The 45-minute plan

What to do, and when, in a design round.

Knowing the material and being able to deliver it are different skills. This is the shape of a round, with an example of what to say at each stage. Each italic line shows the move: copy the move, not the words, and say it with your own numbers.

-
- | | |
|---------|--|
| 0–5 min | Clarify. Ask questions, agree the scope, write down 2–3 core features. Never start drawing. |
|---------|--|
-
- | | |
|----------|--|
| 5–10 min | Estimate. Users → QPS → storage. State the read:write ratio out loud. |
|----------|--|
-
- | | |
|-----------|---|
| 10–15 min | API & data model. A few endpoints, the core entities, the partition key. |
|-----------|---|
-
- | | |
|-----------|--|
| 15–28 min | High-level design. Happy path first, then attack the bottleneck your numbers found. |
|-----------|--|
-
- | | |
|-----------|--|
| 28–40 min | Deep dive. Go where they push, or pick the riskiest component yourself. |
|-----------|--|
-
- | | |
|-----------|---|
| 40–45 min | Trade-offs, failure modes, and what you'd do next. |
|-----------|---|

Opening: the clarifying questions

Say this

"Before I design anything, let me make sure I'm solving the right problem. Can I ask a few questions?"

"What scale are we targeting – roughly how many users, and is this a read-heavy or write-heavy workload?"

"Is there a latency target I should design to? And an availability target?"

"Are there parts of this where correctness matters more than availability – anything involving money or compliance?"

"I'll assume for now that [X] is out of scope so I can go deep on the core flow. Does that work, or would you rather I cover it?"

That last question is the most useful one in the whole round. It makes the interviewer a collaborator and gets you explicit permission for your scope.

Estimating out loud

Say this

"Let me size this before I draw anything, because the numbers will tell me where the pressure is."

"Say 10 million daily actives. If each does 50 reads and 2 writes a day – I'll round a day to 100,000 seconds to keep the arithmetic clean – that's 5,000 reads per second and 200 writes per second on average, call it 15,000 reads at peak."

"So it's about 25 to 1 read-heavy. That tells me caching and read replicas will buy me far more than write optimisation, and I'll design around that."

Round aggressively and say you're rounding. Nobody wants precision; they want to see that you reason quantitatively and that the numbers then *drive* a decision. An estimate you don't use is wasted time.

Drawing the high-level design

Say this

"I'll draw the simplest correct version first, then make it fast."

"Client, load balancer, a stateless application tier, and a database. That works and it's correct – but at 15,000 reads a second the database is the bottleneck, so let me fix that."

"I'll put a cache in front of the read path. I'm choosing cache-aside because the app can control what gets cached, and I'll accept a cold-start burst in exchange for simpler invalidation."

"I'm adding this component to solve a specific number I just computed – I don't want to add anything I can't justify."

Handling a question you can't answer

Say this

"I haven't worked with that directly, so let me reason from first principles and you can tell me if I'm going wrong."

"I don't know the exact figure, but the order of magnitude should be around [X], and the reason that matters here is [Y]."

Reasoning transparently from an admitted gap scores better than confident invention. Interviewers are calibrating how you behave when you don't know something, because that's most of the job.

Closing strong

Say this

"Let me summarise the three decisions I'd defend and the two I'd revisit with more information."

"The main risk in this design is [X]. If I were building it, the first thing I'd do is instrument [Y] so we'd find out early whether that assumption holds."

"If traffic grew ten times, the first thing to break would be [Z], and the fix would be [W]."

The meta-skill

Narrate your reasoning continuously. An interviewer cannot score thinking they can't hear. Say what you're considering, what you're rejecting and why, and name which level you're at – "that's the high-level shape; do you want me to go into the class design?" Silence reads as being stuck even when you're thinking clearly.

The full playbook

Nineteen chapters: the minute-by-minute plan for a design round, the twenty mistakes that cost the hire, Senior and Staff answers side by side, the coding and low-level design rounds, and the behavioural rounds from "tell me about yourself" to the hiring manager. The link is in the description of any Loop Ready video:

youtube.com/@LoopReady.