FRONTIERLOOP · FREE SAMPLE

Inside the Frontier

Preface & Chapter 1 from the field guide to software-engineering interviews at Anthropic & OpenAI. The full guide is at frontierloop.dev.

Preface: How to Read This Book

Most interview prep is written by people who are guessing. This isn't. Every claim, question, score, and quote in this book comes from someone who actually sat in the loop at Anthropic or OpenAI and wrote down what happened. That's the whole premise. If a thing isn't in the sources, it isn't in this book.

Who this is for

You're a software engineer interviewing at Anthropic, OpenAI, or both. The book treats the two as co-equal targets and makes the contrast explicit, because the loops reward genuinely different things: Anthropic punishes over-engineered system design and rehearsed values answers; OpenAI punishes coding solutions you don't finish completely and technologies you can't defend. Knowing which company you're walking into changes how you prepare.

The book leans toward **senior, staff, and principal** candidates. At that tier the bar shifts from "can you solve it" to "can you operate, own, and lead" (Google/MorganVale). New rounds appear, the phone screen can turn into system design, and you're expected to drive the interview with zero hand-holding. If you're earlier-career, the process chapters still map your loop precisely; you can skim the staff-specific material until it's relevant.

The method

This book is built from roughly **66 real candidate accounts**, gathered two ways.

First, primary sources: LeetCode Discuss posts and Blind threads collected first-hand, where engineers describe their loops in their own words. These are messy, specific, and occasionally contradictory, which is exactly why they're valuable. Second, published interview guides: interviewing.io, Exponent, IGotAnOffer, InterviewQuery, and others, plus compensation data from levels.fyi and Blind.

Three rules govern everything that follows:

- **Nothing is invented.** No fabricated questions, no "model answers," no made-up statistics. Where the sources go quiet, so does the book.
- **Everyone is anonymized.** Candidates appear exactly as their sources name them: "a Meta engineer," "Airbnb/57995378," "geekasm on LeetCode," "an Anthropic employee (d2ario)." Both labs require NDAs, and several candidates explicitly declined to post exact question wording for that reason. We respect that line.
- **Conflicts are flagged, not smoothed.** When sources disagree, you'll see it. Anthropic's online assessment is reported as scored out of 600 by some candidates and out of 800 by others. References are quoted as "2" on Blind and "3" in external guides. Rather than pick a winner, the book shows you the disagreement so you can ask your recruiter.

How to use the callouts

Five recurring callouts carry the load-bearing detail. Read them even when you skim the prose around them.

> **TIP**
>
> **Field Note** flags the non-obvious truth most candidates miss (e.g., near-perfect on Anthropic's OA still isn't enough on its own). **By the Numbers** gives you concrete reported figures: durations, scores, comp. **War Story** is a real anonymized experience or verbatim quote with attribution, like the Lyft candidate who finished 2 of 3 coding parts at OpenAI, added extensive tests, and was still rejected. **Tip** is a specific prep action grounded in what worked for real candidates. **Watch Out** is a documented failure mode: the take-home that landed in a Gmail Promotions folder, the scripted STAR story Anthropic flags as its #1 behavioral failure. Every chapter ends with **Key Takeaways**.

An honest word on limits

You should know what this book is and isn't.

The sample is **small, self-selected, and anonymized.** Sixty-six accounts is a lot for this kind of book and nowhere near a statistically clean dataset. People who post about interviews skew toward the memorable outcomes, the brutal rejection and the surprise offer, which means the loudest stories aren't always the typical ones. Anonymity is the price of candor: you get honesty, but you can't independently verify any single account.

Processes also **drift.** Round counts change. Tools change (Anthropic has used CodeSignal, Colab, and Replit; OpenAI uses CoderPad and Excalidraw). Both labs interview enormous volumes, with one secondhand report describing a team screening "tens and sometimes 100 to 200 candidates" for a single job, and a pipeline at that scale gets retuned constantly. Timelines reported here run from under two days end-to-end to more than three months for the same company. That spread is real, not noise to be averaged away.

So treat this book as **calibrated intelligence, not gospel.** The patterns are strong and repeated: Anthropic's culture round really is described, repeatedly, as the hardest to pass; over-engineering really is the most-documented system-design failure; OpenAI's "finish all the parts" rubric really did sink strong candidates. Use those patterns to prepare with intent. But when it comes to your specific loop, the format of your rounds, the number of references, the current cooldown policy, your recruiter is the source of truth. The sources themselves say it best: "recruiter is the best source of truth for your specific loop."

Read the patterns here. Verify the specifics there. Then go operate like someone they'd be comfortable sending to the onsite.

1. Why Frontier-Lab Interviews Are Different

You have done the FAANG loop. You know the rhythm: grind two hundred LeetCode problems, memorize the system-design templates, rehearse your STAR stories, walk in, perform. That muscle memory will hurt you at Anthropic and OpenAI. Not because these loops are harder in the abstract, but because they are testing for different things, and several of the reflexes that got you offers elsewhere are the exact behaviors these labs flag as failure modes. This chapter is the map of those differences. Read it as the thesis for everything that follows.

The four breaks from FAANG

Across roughly 66 first-hand sources — Blind posts, LeetCode write-ups, interviewing.io and IGotAnOffer breakdowns, named candidate accounts — four structural differences show up again and again. They hold at both labs, with the texture varying by company.

1. The coding is build-from-scratch or extend-existing, not LeetCode. Both labs say this in writing on their own pages. Anthropic's candidate guidance is explicit: "*This will be a pure programming problem solving interview which doesn't benefit from memorizing standard algorithms or data structures... your interviews may be more open-ended than standard technical problems*" (Anthropic guidance, via linkjob.ai). OpenAI's guide via interviewing.io: "*You're not going to get questions on string manipulation... they are going to be about stuff you might actually do at work.*" You will implement an in-memory database, a web crawler, a stack-trace parser, a resumable iterator — things that look like a Tuesday at work, not a competitive-programming set.

2. System design is AI-infra-flavored and math-first. The canonical Anthropic prompt is "Design the Claude chat service" or "single GPU, 100 inputs per batch, synchronous waiters." And the math is not garnish. "*You really need your shit together on math — throughput, latency, data size, TPS... The math actually drives design decisions and tradeoffs in substantial ways, unlike many other system designs where it's nice-to-know*" (Blind). Anthropic explicitly disagrees with the popular HelloInterview advice to defer estimation to late in the design (ex-Robinhood/decompiler).

3. There is a values/mission gate that genuinely fails strong engineers. This is Anthropic's signature, and it is real. "*The actual cultural interview is incredibly strange and ultimately pretty much the most difficult interview to pass*" — said by an Anthropic employee (d2ario). FinalRoundAI's summary of the accounts: "*Most candidates are eliminated — not for lack of technical skill, but for insufficient depth on Anthropic's AI safety philosophy.*"

4. AI tools are banned in the live rounds, and the question banks are small and half-public. Anthropic prohibits AI assistance during interviews and uses LLMs to detect gaming — a pointed irony noted by candidates: "*Anthropic ironically doesn't allow the use of AI tools unlike google or meta*" (Apple/Werrkdi). And because the question set is small and largely public, the bar has shifted from *have you seen this* to *can you actually operate.*

> **FIELD NOTE**
>
> The thing you were trained to do at every other company is the thing that gets you rejected here. Scripted STAR stories are Anthropic's documented **#1 behavioral failure mode**. Over-engineering a system design is the **#1 documented SD failure**. Both are FAANG-optimal reflexes. You are not walking into a harder version of the same test — you are walking into a different test that punishes your trained instincts.

Coding: operate, don't recite

The dead giveaway that this is not LeetCode is the *shape* of the problem. Anthropic's most-confirmed online assessment is a single in-memory database built across four progressive levels: basic SET/GET/DELETE, then prefix scans, then TTL with timestamps, then file compression with quotas. Each level extends the code you wrote in the last one. The skill under test is not "can you find the trick" — it is "can you write extensible code from line one so that level three doesn't force a rewrite." Tight code that only passes level one *becomes a liability* (IGotAnOffer).

The contrast between the two labs is sharp. Anthropic lowers time pressure and rewards reasoning: "*Time pressure is lower than at Meta or OpenAI, and you write less code, but the reasoning behind every decision is front and center*" (interviewing.io). A passing phone screen was about 100 lines of Kotlin (Airbnb/57995378). OpenAI runs a completionist gauntlet: the onsite coding round has three sub-parts and you must finish all of them. One candidate finished two of three parts with strong tests and edge-case handling and was still rejected — "*the rubric for that question was specifically to finish all 3 parts to pass*" (Lyft/krkdbw). Adding tests did not buy back the unfinished part.

> **WAR STORY**
>
> *A Lyft engineer (~$400K TC) on the OpenAI onsite: finished 2 of 3 coding sub-parts, wrote thorough tests and edge cases, even discussed deployment — and was rejected anyway because the rubric required all three parts. The same person's takeaway for system design: "make sure you talk a lot about reliability." (Lyft/krkdbw, Blind)*

One more load-bearing detail: language. At Anthropic, Python is a material advantage, not a preference. "*If you code in Python you are in a great position. I used cpp and it took 3x more characters... massive massive disadvantage. Prepare to write 150-200 line of code if using cpp*" (Uber/Ohfl10). The problems are implementation-heavy, so a concise language directly buys you time and clean code.

System design: math, simplicity, and safety in the flow

Anthropic's system design is best described by a coach's phrase: it is a "*Whiteboard Lab, not Pattern Matching*" (Tarek/IGotAnOffer). You reason through unfamiliar AI-infra problems from first principles rather than reciting how a hyperscaler solved something on its engineering blog. The master move that fits the entire question bank: "*Every Anthropic system design question can be reduced to a*

classic infrastructure problem" — abstract first, do the math, build the simplest correct design, then layer the AI constraints (GPU memory, KV cache, batching, context limits).

Two things make this genuinely different from FAANG. First, simplicity is a *graded signal* and over-engineering is the single most documented failure cause. An infra engineer with 10+ years failed a first-round Anthropic SD specifically for over-engineering: "*I recently failed an Anthropic systems design interview (1st round) due to overengineering... I noticed them struggling to follow*" (ex-Robinhood/decompiler). The internal value is literal: "*We don't invent a spaceship if all we need is a bicycle.*" A correct-but-medium-complex design loses to an elegant one you can explain in fifteen minutes even at a pulse of 170.

Second, safety is a first-class architectural constraint, not a product afterthought. The reported #1 SD red flag is "*prioritizing scale over safety.*" You are expected to design "*safety guardrails directly into the request flow rather than treating safety as an afterthought.*"

> **BY THE NUMBERS**
>
> Anthropic scale anchors that recur across the accounts — internalize them: **1B documents @ 1M QPS**; **10M docs, top-k semantic, under 50ms**; **single GPU, ≤100 inputs/batch** with synchronous waiters; **100,000 req/s** token service; a **p95 latency spike from 100ms to 2000ms** to investigate. OpenAI's anchors run bigger and more product-shaped: **ChatGPT at hundreds of millions of weekly users**; a crawler at **10M req/s**; a vector DB over **billions of embeddings**.

OpenAI's design round optimizes for a different axis: scale and reliability. The load-bearing word is *reliability* — follow-ups push relentlessly into "*how individual services degrade and recover under load — not just high-level architecture diagrams.*" And there is a hard rule on naming technologies: "*If you call out any specific technologies, be prepared to go into detail... It may be best not to bring up specific examples as they like drilling into the pros and cons*" (interviewing.io). Anthropic, by contrast, *rewards* grounded specificity — saying "gRPC sucks for certain use cases, sometimes plain HTTP/1 or TCP is better" reads as operational depth.

> **WATCH OUT**
>
> Do not prep your Anthropic system design by brainstorming with an LLM. The reports are blunt: "*LLMs are still terrible at novel, simple system designs... they default to pattern matching*" and "*add complexity when challenged.*" They will push you toward exactly the hyperscale over-engineering that fails the round. Form the design yourself; use Claude Code only to vibe-code it afterward as a grounding check.

The values gate that fails good engineers

This is the part with no FAANG analog. Anthropic's culture round is universal — "*45 minutes. Every candidate goes through it — PM, engineer, researcher, sales. Every role, every level*" (Ridhima Khurana). Candidates independently reach for the same word: "*closer to a therapy session than a job interview: deeply personal, emotionally probing, conversational.*" There is no case to crack. The

questions feel deceptively simple until, twenty minutes in, you realize the interviewer can see straight through you.

What it actually scores, per the cleanest articulation in any source (Khurana): can you hold complexity without collapsing into a tidy narrative; do you acknowledge what you don't know; can you reason about second-order effects; are you intellectually honest even when it makes you look less polished. The counterintuitive truth that trips up confident candidates: "*Anthropic actively seeks skepticism and pushback over enthusiasm.*" Agreeing with everything Anthropic does — the "dream company trap" — is a red flag. "*Never end an answer with '...and that's why Anthropic is amazing' (please never do this).*"

The signature question, which appears at the recruiter screen, the HM screen, *and* the culture round: "*What are your thoughts on AI safety and the risks of advanced AI systems?*" And the most common way to fail it: "'*AI could be dangerous*' *is not an answer. It's a Wikipedia summary*" (Khurana).

> **TIP**
>
> Start the values prep first; it needs the most marinating. Read Anthropic's **Core Views on AI Safety** and **Responsible Scaling Policy** (they pre-share these), and Dario Amodei's **"Machines of Loving Grace"** — one candidate was "*quizzed about Dario's book that I didn't read*" and got no offer. Then build a story bank of 8–12 real, emotionally honest stories you can tell conversationally, not as recited STAR blocks. Have one genuine critique of Anthropic ready to defend.

OpenAI's behavioral round is far more conventional — standard "tell me about a time," with STAR perfectly acceptable. But mission sincerity is still gated hard. "*Why do you want to work at OpenAI?*" is the single most-repeated question in these accounts, and a "*rehearsed answer about career growth or the prestige of the role will not land.*" Notably, OpenAI candidates are told to read the charter and *be ready to compare OpenAI's approach to Anthropic's.*

What each lab optimizes for

Strip away the detail and the two loops resolve into two distinct ideals of an engineer.

	Anthropic	OpenAI
Coding optimizes for	Clarity, calm under ambiguity, clean extensible code, reasoning per decision	Completeness (all parts), no-hints independence, production quality, speed
System design optimizes for	Math-driven first principles + simplest correct design + safety in the request flow	Scale + reliability and degrade/recover mechanics + defensible tech choices
Behavioral optimizes for	Intellectual honesty, ethical depth, non-rehearsed stories, skeptical engagement	Mission sincerity, autonomy ("Find a way"), cross-functional collaboration
Cultural archetype	Mission-first, low-ego generalist who reasons from first principles	Autonomous, fast builder who finishes and defends trade-offs
Biggest single risk	Over-engineering + scripted/fanboy values answers	Not finishing all coding parts + naming undefendable tech

Anthropic optimizes for the **mission-aligned, intellectually honest, low-ego generalist who chooses the simplest correct design.** OpenAI optimizes for the **autonomous, production-quality engineer who finishes completely, defends every trade-off, and ships under shifting priorities.** Both are top-of-market on comp and equity-heavy; both send AI-generated rejections with no feedback and enforce long cooldowns. Neither is impressed by the candidate who is impressed by them.

One candidate who failed five AI labs ranked the difficulty: "*Cursor* > *xAI* > *Anthropic* > *Microsoft AI* > *OpenAI*" (Snowflake/snowsnob) — placing Anthropic as the harder of our two. Take that as one data point in a process where luck and randomness are heavy at both. The rest of this book is about removing as much of that randomness as the sources allow.

KEY TAKEAWAYS

- Both labs run **practical, build-from-scratch coding**, not LeetCode — they say so on their own pages. The bar is "can you operate," not "have you memorized patterns."
- **Anthropic = calm, clean, reasoning-first** coding with lower time pressure; **OpenAI = finish all parts flawlessly** with no hints. Code in Python (a material advantage at Anthropic).
- System design diverges: **Anthropic is math-first, simplest-correct, safety-in-the-flow**; **OpenAI is scale-and-reliability with defend-every-tech-you-name.** Over-engineering is Anthropic's top documented failure.
- The **values round genuinely fails strong engineers** — universal at Anthropic, "therapy-session" in feel, rewarding skepticism and intellectual honesty over enthusiasm. Scripted STAR is the #1 failure mode.
- **AI tools are banned live** at Anthropic (LLM-based gaming detection), and the small, half-public question banks mean memorization won't save you — perturbation exposes it.
- Don't be a fanboy: never end a values answer on flattery, and have a real critique of Anthropic ready.

Keep going — get the full field guide

You just read the opening. **Inside the Frontier** is the complete ~105-page guide to the Anthropic & OpenAI software-engineering loops — synthesized from 60+ real, publicly-reported candidate accounts.

Inside the full guide:

- The **two loops, stage by stage** (Anthropic and OpenAI, side by side)
- The **coding rounds** — the known question families and how interviewers perturb them
- **System & AI-infra design** — the real prompt themes and the simplest-correct rule
- The **values-round playbook** — the #1 filter, and how to prepare for it
- **Compensation, decoded** + a **master question bank** with a confidence rating on every question
- A prioritized **2–4 week prep plan**

Get it at frontierloop.dev

Independent and unofficial. Synthesized from publicly-reported candidate experiences; not affiliated with, authorized by, or endorsed by Anthropic, OpenAI, or any company named.