

Haythem Ben Abdallah

February 1, 2026

1 How I Used AI in This Design Process

1.1 AI Tools Used

I used multiple large language models as design collaborators, including GPT-5, Claude 4.5 Opus (thinking mode), and Gemini 3 Pro, to brainstorm, critique, and stress-test different architectural approaches. I also used Antigravity to generate a runnable prototype once the design had stabilized.

While I usually use Windsurf for development work, this project was a great opportunity to test another tool like Antigravity and evaluate its capabilities in a real design scenario.

Rather than relying on a single model, I intentionally used multiple LLMs with different strengths and compared their outputs to avoid over-indexing on one perspective.

1.2 Methodology

I treated the models as independent reviewers, not sources of truth.

- Explored different architecture options and trade-offs with each model separately
- Asked deep “why” and “what if” questions (latency, caching, rollout strategy, failure modes)
- Shared outputs between models so they could critique and challenge each other’s assumptions
- Iterated on the design by resolving disagreements between models and validating decisions against real system constraints such as latency targets, scale, and operational complexity
- Once the architecture was clear and decisions were locked, I used AI to help generate structured documentation and a prototype aligned with those final choices

This cross-review approach helped reduce single-model bias and surface blind spots earlier in the process.

1.3 What Worked Well

- **Rapid exploration of trade-offs:** AI was effective at quickly enumerating design options and their implications.
- **Challenging assumptions:** Having different models critique each other helped identify weak points, especially around caching boundaries, rollout mechanics, and failure handling.
- **Speed to clarity:** AI significantly reduced the time needed to reach a coherent, interview- and production-ready design.
- **Documentation acceleration:** Once decisions were made, AI was useful for turning them into a clean, structured System Design Document and API contracts.

1.4 What Didn't Work Well

AI does not replace engineering judgment. Models can confidently propose solutions that sound reasonable but don't hold up under real operational constraints.

Performance assumptions, scaling limits, and failure thresholds still required human experience and validation. The best results came from iteration, review, and cross-checking, not from first responses.

1.5 Human Supervision & Code Quality

All AI-generated code was reviewed under explicit human supervision before being considered complete. While the generated code typically achieved high test coverage and passed automated tests, this was treated as a necessary but not sufficient condition for production readiness.

In practice:

- All AI-generated logic was manually reviewed for correctness, edge cases, performance characteristics, and operational safety.
- Tests were used to validate behavior, but design intent, failure modes, and scalability concerns were verified through human review.
- In a small number of cases, AI-generated implementations passed tests but revealed issues during review (such as inefficient data access patterns, missing failure handling, or unclear ownership boundaries). These were corrected before acceptance.

With this review process in place, AI-generated code can reach production-quality standards, but only when paired with clear ownership, strong test coverage, and deliberate human validation. **Responsibility for correctness and reliability ultimately remains with the engineer.**

1.6 AI Assistance for Teams

I would position AI as a force multiplier, not a decision-maker.

AI works best for early-stage design exploration, documentation drafts, and prototype scaffolding. I would encourage multi-model or peer-review usage to reduce bias and overconfidence, while keeping humans responsible for final decisions—especially around scalability, security, and operational risk.

Lightweight guardrails are essential: no PII exposure, clear review ownership, and explicit validation steps. Used this way, AI helps teams think faster and more broadly, while engineering judgment determines what ultimately ships.

1.7 Summary

I used multiple AI models as independent design reviewers to explore and challenge architectural choices, cross-validated their outputs to reduce bias, and then leveraged AI to accelerate documentation and prototyping once the design was finalized.