Yoko Li - Executive Briefing

Prepared for: Gather Investment Discussion
Date: January 30, 2026

Background

Partner, Andreessen Horowitz (a16z) | AI & Infrastructure Team
Investment Focus: Developer tools, infrastructure, AI, creative tools | Pre-seed to Series B | $500K-$40M checks

Engineering & Product Experience:

- **HashiCorp** - Led product for Terraform Cloud (infrastructure-as-code platform)
- **Transposit** - Founding Engineer/Product Manager (workflow automation startup)
- **AppDynamics** - Software Engineer (sold to Cisco for $3.7B)
- **Education:** Rice University (Engineering)

Still Codes: Active open source maintainer (AI-town, AI-tamago, local-ai-stack) | Technical cartoonist (@stuffyokodraws)

Board Positions

Board Member (Full Seats):

- **Resend** - Modern email API for developers | Replaces SendGrid/Mailgun | 3,000+ customers (Warner Brothers, Supabase)
- **Relace** - (Limited public info)

Board Observer:

- **Clerk** - Auth & user management platform | Drop-in UI components | $84.5M raised | 500+ businesses
- **Mintlify** - AI-powered docs platform | Used by Anthropic, Cursor, ElevenLabs | $21.7M

raised

- **Inngest** - AI workflow orchestration | Runs multi-step agent workflows in production | $10.6M raised
- **Phota Labs** - (Limited public info, likely AI/creative tools)

Other Key Investments:

- **Stainless** (Series A) - Auto-generates SDKs from API specs
- **Upstash** - Serverless Redis/Kafka
- **Nrwl/Nx** (Seed + A) - Monorepo build tools
- **Svix** (Seed) - Webhooks-as-a-service
- **Arcjet** (Seed) - Security/dev tools

Investment Thesis & Key Ideas

Core Philosophy:

"AI agents as the next computing platform" - Not AI as tooling, but AI as foundation for how software gets built

Key Concepts She's Coined/Championed:

1. "Agent Experience (AX)" not just "Developer Experience (DX)"

- Products should be designed for AI agents to use, not just humans
- APIs should be "agent-friendly" from day one

2. "Composable Service Primitives"

- Quote: *"Agents need clean and composable service primitives to scaffold reliable applications"*
- Each company provides a discrete building block (auth, email, docs, workflows)

3. Vertical Integration as Moat

- As LLMs commoditize, owning the full stack creates defensibility

- Don't build middleware - own the complete experience

4. AI-Native Architecture (Not Bolt-On)

- Recent focus on MCP (Model Context Protocol) for connecting LLMs to tools
- Looking for companies rebuilt from ground up for AI, not legacy systems with AI added

5. Infrastructure Over Applications

- Developer tools that solve hard technical problems
- Depth > breadth; primitives > features

Portfolio Pattern Recognition

What ALL Her Investments Share:

- ✅ **Developer-first** (developers are primary users)
- ✅ **Vertically integrated** (own full experience, not plugins)
- ✅ **Replace fragmented systems** (consolidate 5-10 legacy tools into one)
- ✅ **JavaScript/TypeScript ecosystem** (modern web stack)
- ✅ **Composable together** (Clerk + Resend + Mintlify work seamlessly)
- ✅ **Increasingly AI-native** (recent investments designed for agent consumption)

Specific Replacement Patterns:

Her Investment	Replaces
Clerk	Auth0, Firebase Auth, custom auth
Resend	SendGrid, Mailgun, AWS SES
Mintlify	GitBook, ReadMe, custom docs
Inngest	Custom workflow engines, Temporal
Stainless	Manual SDK writing, swagger-codegen

Recent Focus (Last 6 Months)

Podcast & Writing Activity:

- **Dec 2024:** MCP Co-Creator podcast (infrastructure for AI agents)
- **Mar 2025:** Resend + MCP podcast (making systems agent-friendly)
- **May 2025:** "Emerging Developer Patterns for the AI Era" article (agents as platform)

Current Thesis Evolution: Actively seeking infrastructure that enables AI agents to compose reliable applications

---

How to Speak to Her

DO:

- ✅ Lead with technical architecture (she still codes)
- ✅ Show working product/demos (builder credibility)
- ✅ Frame as "primitive" or "infrastructure" not "app"
- ✅ Use "agent-native" language (not "AI-powered")
- ✅ Reference her portfolio companies as analogies
- ✅ Address "why not incumbents?" immediately

DON'T:

- ❌ Pitch as consumer app disruption
- ❌ Focus on TAM/market size over unit economics
- ❌ Compare to Uber/Netflix (use Stripe/HashiCorp instead)
- ❌ Say "AI handles everything" without explaining HOW
- ❌ Dismiss incumbents as "just too slow"

The Perfect Gather Positioning for Yoko

One-Liner:

"We're building the connectivity + intelligence primitive for AI agents - just as Clerk is the auth primitive and Resend is the email primitive, Gather is the telco primitive."

Why This Works:

1. **Primitive Framework** - Speaks her language (infrastructure, not app)
2. **Vertical Integration** - Own full stack like all her investments
3. **AI-Native** - Built from ground up as agent-operated system (not bolt-on)
4. **Developer DNA** - Team built Terraform-equivalent for telco (HashiCorp connection)
5. **Replaces Fragmented Systems** - 4,000 legacy IT systems → one unified platform
6. **Proprietary Data Moat** - 24/7 telco context layer no pure-play AI company can replicate
7. **B2C Wedge → Platform** - Same playbook as Clerk (start focused, expand to platform)

Portfolio Synergies:

Gather becomes showcase customer using Clerk (auth), Resend (email), Mintlify (docs) - demonstrating how primitives compose in AI-native architecture

---

Key Portfolio Analogies to Use in Conversation

When discussing vertical integration:

"It's like how Resend doesn't just wrap SendGrid - they own the entire email sending experience end-to-end. We own the entire telco experience end-to-end, which lets us optimize every interaction for learning."

When discussing developer experience:

"You led product at HashiCorp - you know Terraform succeeded because it abstracted away infrastructure complexity. We're doing the same for telecom operations. Instead of 4,000 systems and manual workflows, it's one unified API that AI agents can orchestrate."

When discussing AI-native design:

"Mintlify doesn't just add AI chat to docs - they designed docs from the ground up to be agent-readable. We designed telecom from the ground up to be agent-operated. The entire system expects AI agents to be the primary 'users.'"

When discussing production reliability:

"We've spent a lot of time thinking about what Inngest solves - running AI workflows reliably in production. That's critical for us because we can't have customer service or network management failing due to LLM hallucinations. Here's our approach..."

Key Questions She'll Ask (and Your Counters)

Q1: "Why not sell to enterprises instead of B2C?"

Counter: "We need real end-user behavioral data to train TIM. Enterprise deals would take 24 months and we'd be integrating with their legacy systems - that defeats the purpose of being AI-native. B2C gives us the fastest learning loop and proves unit economics. Then we can sell the platform to other telcos - same playbook as Clerk going from indie devs to enterprise."

Q2: "What's defensible? Can't OpenAI + AT&T do this?"

Counter: "Three moats: (1) Telco-specific data we're collecting 24/7 - no one else has this context, (2) Regulatory licenses that take 12-18 months to get, (3) Most importantly - AT&T can't reorganize around AI. They have 4,000 systems and would need to sunset them. It's the classic innovator's dilemma. This is infrastructure that needs to be built from scratch."

Q3: "How do you handle LLM non-determinism in production?"

Counter: "Great question - this is where our telco background matters. We run agents in a constrained environment with guardrails: [explain your specific approach - circuit breakers, fallback logic, test suites]. Similar to how Inngest handles workflow reliability, we've built observability and rollback into every agent action."

Q4: "Walk me through cohort retention economics"

Counter: [Have detailed cohort data ready - month 1-12+ retention curves, ARPU trends, LTV calculation methodology]

The Opening (First 3 Minutes)

"Yoko, I wanted to show you what we're building because it sits directly at the intersection of three things you've been writing about: AI agents as infrastructure, vertical integration as moat, and building systems that are agent-native from day one rather than bolting AI onto

legacy architectures.

We're building the world's first fully autonomous telco - but really, we're building the connectivity and intelligence primitive for the AI era. Just like your portfolio companies - Clerk for auth, Resend for email, Mintlify for docs - we're creating a composable primitive, except ours is for connectivity and real-time customer intelligence.

Can I show you the actual system? It's live - we're running about $10M ARR through it right now with our first persona, Meow Mobile."

Then immediately go to demos/screenshots.

What to Have Ready

Technical Deep Dive:

- TIM architecture diagram
- How you handle LLM fallbacks/reliability
- Data collection and privacy approach
- Integration architecture with AT&T

Unit Economics:

- Meow Mobile cohort analysis (month 1-12+ retention)
- CAC breakdown by channel
- ARPU trends over time
- LTV calculation methodology

Product Roadmap:

- Next 3 personas you're launching (with rationale)
- Timeline to 2M subscribers
- Platform/API launch timeline
- When you'll start selling to other telcos

The Ask:

- How much capital for Series A

- What milestones it gets you to
- Who else is in the round
- Why a16z specifically (beyond Yoko)

Post-Meeting Follow-Up

Send within 24 hours:

1. **Technical deep-dive doc** answering any architecture questions she raised
2. **Link to working demo/product** (even if limited access)
3. **One relevant article** she's written + your specific reaction/question about it
4. **Introduction to one of your technical advisors** who can validate the approach

---

Bottom Line

Yoko is an engineer-turned-VC looking for AI infrastructure primitives built by technical founders who've solved the problem before. She invests in companies that:

- Replace fragmented legacy systems with vertically integrated solutions
- Are designed from the ground up for AI agents (not bolt-on)
- Provide composable primitives that other developers/agents can build on
- Have technical founding teams with domain expertise

Gather checks all these boxes. The key is positioning as infrastructure/primitive, not consumer app.

Contact:

- Email: yli@a16z.com
- Portfolio: https://a16z.com/author/yoko-li/
- GitHub: @ykhli
- Twitter: @stuffyokodraws