PRINTABLE SCHEDULE

AI Engineer World's Fair 2026

The largest technical conference for AI engineers

When June 29 – July 2, 2026 **Where** Moscone West, San Francisco, CA **Sessions** 515

Interactive schedule: ai.engineer/worldsfair · Register: app.ai.engineer

CONTENTS

Day 1 — Workshop Day Monday, June 29, 2026

contents ↑

9:00am

SPONSOR **Track 1** 9:00am-11:00am *tentative*

Arize 2hr

SPONSOR **Track 2** 9:00am-11:00am *tentative*

Neo4J 2hr

SPONSOR **Workshops Day 1 · Track 3** 9:00am-11:00am

OpenAI Workshop with Charlie Guo

Charlie Guo — OpenAI

SPONSOR **Workshops Day 1 · Track 4** 9:00am-11:00am

The best SDLC is the one you build yourself: Why orchestration changes everything

Shane Wolf · Andrei Bocan

Industry research shows AI productivity gains have plateaued at 10–15% — because today's tools only optimize the 20% of a developer's day spent writing code. The real bottlenecks are left and right of code: planning, orchestration, review, and operations. Join Atlassian's Shane Wolf and Andrei Bocan for a hands-on deep dive into the AI-native SDLC. In this workshop, we'll move past single-player copilots and show you how Atlassian is turning Jira into an AI-native orchestration layer for the entire software development lifecycle. You'll work with Jira AI Planner — an always-on AI technical architect that helps you rapidly vet ideas, generate estimates, and create work breakdowns — and the Jira Coding Agent, which tackles coding tasks directly from your board without ever switching to an IDE. Then we'll go further: you'll learn how to build custom automations that chain these capabilities together, transforming your Jira board into an agentic software factory where humans set intent and agents execute. You'll walk away with a working understanding of how to move from single-player copilots to multiplayer AI workflows — where the coordination layer becomes the leverage point. What you'll learn: How AI Planner turns rough ideas into well-scoped, agent-ready work breakdowns How to trigger the Jira Coding Agent - development directly from your backlog How to build automations that orchestrate human-agent collaboration Why intent and context (not code generation) are the moat in an AI-native SDLC Please bring your own laptop to this session. No pre-work is required for this workshop.

SPONSOR **Track 3 · Track 5** 9:00am-11:00am *tentative*

Snyk 2hr

SPONSOR **Workshops Day 1 · Track 6** 9:00am-11:00am

Total Recall: Agent Memory and Harness Engineering

Ignacio Martinez — AI Developer Advocate, Oracle

In this hands-on workshop you'll build a working autonomous agent from the harness up, in a notebook, then see it live in a full working web application and leave with one that can write and run its own automations. You'll implement every surface area yourself: a set of predefined tools, persistent memory through the Oracle AI Agent Memory package, orchestration with LangChain and LangGraph, and LLM access through OCI GenAI Service, composing the full set of Oracle primitives into one harness you understand end to end. Most teams assemble that harness from a dozen disconnected services: one store for vectors, another for state, a separate reranker, a bolt-on memory layer. We take the opposite approach, on a single unified memory core. The organizing principle is optionality by default: you shouldn't have to choose your memory substrate up front. With Oracle AI Database you get file system and database memory in one place, embedding models and rerankers running inside the database kernel, and every retrieval strategy an AI workload needs without leaving the core. And consolidating onto one core is what keeps the whole thing tractable. You know the drill: a production harness has you holding all those moving parts in your head at once, and most of your attention goes to keeping them in sync rather than improving the agent. Pull that sprawl into a single core and the cognitive load drops. You get to think about what the agent does, not where its state lives. That's the difference between controlling your harness and renting its pieces.

SPONSOR **Track 7** 9:00am-11:00am

Agents That Own Their Inference: Building Production AI Agents on Dedicated GPUs

Du'an Lightfoot — Senior AI Engineer, Akamai Technologies

Every production agent today is renting its intelligence. You're paying per token, sending your customer's data to someone else's servers, and hoping the provider doesn't rate-limit you during your launch. For most teams, that's fine. But for a growing number of teams in regulated industries, with high-volume products, latency-sensitive workloads, or rising token bills, it's starting to look like a liability. In this 120-minute hands-on workshop you'll get a dedicated GPU and build an agent that runs on infrastructure you control. You'll stand up vLLM, point your agent at it, and drive concurrent load through the stack until you can see batching, KV cache pressure, and throughput limits in the metrics. Then you'll optimize the deployment to improve throughput while keeping per-request latency in line. The focus isn't agent frameworks. It's the inference layer underneath them. You'll leave with working code and a real understanding of continuous batching under real concurrency, KV cache tradeoffs, vLLM's metrics, and the bottlenecks that only show up when you operate the inference server yourself.

SPONSOR **Workshops Day 1 · Track 8** 9:00am-11:00am

Open-Source Inference Engineering for the Agentic Era

Zain Hasan — Staff AI/ML Engineer - DX, Together AI · Yubo Wang — LLM Inference, Together AI · Qingyang Wu — Staff Research Scientist, Together AI · Jue Wang — Senior Staff Researcher, Together AI

Agentic coding workloads demand long contexts, multi-turn conversations, and throughput at a scale that most inference engines weren't built for. TokenSpeed is a new open-source engine purpose-built for this regime, built collaboratively by NVIDIA DevTech, AMD Triton, Qwen Inference, Together AI, and others, delivering state-of-the-art performance on NVIDIA Blackwell and AMD hardware. In this 2-hour hands-on workshop, Together Inference Research Engineers and a TokenSpeed co-creator will cover: TokenSpeed architecture and how it compares to other top open inference engines Deploying and configuring your first model Optimizing for agentic workloads: long-context, multi-turn, high-concurrency traffic Kernel and hardware tuning Throughput/latency trade-offs and when TokenSpeed is the right tool

SESSION **Track 9** 9:00am-11:00am

Advanced workshop: Mastering AI Observability

Doug Guthrie — Solutions Engineer, Braintrust

Your AI is in production, but is it actually good? In this hands-on workshop, you'll learn how to uncover patterns in your production traces using Braintrust Topics, build custom scorers to target real issues, and systematically improve your agent. By the end, you'll have a repeatable eval workflow and trace-backed evidence that your AI is actually doing what you think it is. What You'll Build: 1. A quality analysis using Topics 2. Custom scorers targeting specific issues 3. An optimized agent with traces to prove improvements A hands-on workflow for going from "I think my agent is working" to trace-backed evidence that it actually is.

SPONSOR **Track M** 9:00am-11:00am

Microsoft 2hr

11:05am

SPONSOR **Workshops Day 1 · Track 1** 11:05am-12:05pm *tentative*

From approval loops to autonomous agents with Docker

TBD — Docker

"You've invested in the best models, coding agents, and AI tooling. Now comes the hard part: unlocking autonomous development without creating security headaches, governance gaps, or endless approval loops. In this 90-minute hands-on workshop, you'll learn how to run coding agents in isolated environments built for autonomous work, create a 'golden path' for AI-assisted development across your organization, reduce software supply chain risk with secure, hardened containers, manage multiple agents with the right permissions and guardrails, and scale AI-powered development without slowing developers down."

SPONSOR **Track 2** 11:05am-12:05pm *tentative*

Neo4J 1hr (plat)

WORKSHOP **Workshops Day 1 · Track 3** 11:05am-12:05pm

How I learned to stop worrying and love the sandbox

Matt Brockman — AI Engineer, E2B

Running sandboxes at scale can get painful. How do you manage a thousand concurrent sandboxes? We'll cover burst traffic, fast sandbox creation under load, resource exhaustion, shared state with volumes, and per-user data isolation. Then you'll trigger each failure, implement fixes, and see the cost impact in real time. You'll leave with hands-on experience debugging sandbox failures and a set of observability and scaling patterns you can start implementing.

WORKSHOP **Track 4** 11:05am-12:05pm *tentative*

Microsoft - Bonus 1hr

Lab B

WORKSHOP **Workshops Day 1 · Track 5** 11:05am-12:05pm

Teaching Agents to Search: Building Synthetic Training Pipelines with NVIDIA Data Designer

Dhruv Nathawani — Senior Research Scientist, Nvidia

Modern agentic systems often fail because the right training data simply does not exist. Search agents are a perfect example: if you want a model to browse the web effectively, you need high-quality multi-step trajectories that teach it how to search, refine queries, inspect sources, and recover from dead ends. In this session, attendees will learn how NVIDIA used Data Designer to build synthetic supervised fine-tuning data for search-capable Nemotron models, including how to define task structure, generate seed examples, produce realistic search trajectories, filter low-quality generations, and convert traces into training-ready records. The session will also cover BrowseComp-style tasks, tool-use rollouts, validation, dataset curation, and a reusable framework for designing custom datasets for specialized behaviors across reasoning, tool use, and domain-specific applications.

WORKSHOP **Workshops Day 1 · Track 6** 11:05am-12:05pm *tentative*

To be announced

WORKSHOP **Workshops Day 1 · Track 7** 11:05am-12:05pm

How to Build Quality Gates into Agentic Coding Workflows

Nnenna Ndukwe — Principal Developer Advocate and Software Engineer, Qodo AI

AI coding agents can now generate code at unprecedented speed. But faster code generation creates a new engineering problem: how do we know when agent-written code is actually safe, maintainable, and ready to merge? In this hands-on workshop, attendees will build an agentic coding workflow with enforceable code quality gates across planning, implementation, testing, and code review. By the end of the session, participants will have a working reference pattern for agentic software delivery: an AI-assisted workflow that can inspect a repo, implement a change, run tests, evaluate risk, respond to feedback, and surface what still requires human judgment. This is a technical enablement session for engineers building with AI coding agents, platform teams designing agentic SDLC workflows, and AI engineering leaders thinking about how to scale software quality with AI.

WORKSHOP **Workshops Day 1 · Track 8** 11:05am-12:05pm

What is an Inference Engine, Anyway?

Charles Frye — AI Engineer, Modal

SESSION **Workshops Day 1 · Track 9** 11:05am-12:05pm

From 0 to production: Observing and improving AI agents with Langfuse

TBD — ClickHouse / Langfuse speaker

Join us for a hands-on Langfuse workshop where we'll show you how to observe, debug, and improve your AI applications, step by step, using a real sample app. Bring your questions and discover how Langfuse can level up your specific use cases!

SPONSOR **Track M** 11:05am-12:05pm

Microsoft 1hr

12:10pm

WORKSHOP **Workshops Day 1 · Track 1** 12:10pm-1:10pm

Your Evals Are Lying to You

Tejas Kumar — IBM

“Our evals pass and our velocity is up, so it works.” It’s the most reassuring sentence in AI engineering and also the most dangerous. Teams are shipping more code than ever while incidents per PR and change-failure rates climb, and the instruments meant to catch this are quietly broken. This talk takes apart both halves of that false comfort. First, why velocity lies: the same AI-driven throughput that lights up your dashboard is what’s eroding quality underneath it. Then we explore four ways offline evals deceive you: LLM-as-judge bias (your grader rewards confident, wordy, wrong answers over terse correct ones), staleness, distribution shift between your golden set and real traffic, and single-score evals that hide which step of an agent actually failed. The centerpiece is a live demo. We’ll wire up an LLM judge on stage and watch it crown a confident, friendly, factually wrong answer. Then we’ll fix it live on stage with a three-line rubric change. Same model, different instrument. From there we’ll build up what to measure instead: traces and spans, production observability, probe-based evaluation, error budgets, and quality leading indicators that sit beside every velocity number. Attendees will leave with a five-line checklist they can apply Monday. No prior eval tooling required. If you’ve ever shipped something agentic and had a nagging feeling the dashboards were too kind, this is for you.

WORKSHOP **Workshops Day 1 · Track 2** 12:10pm-1:10pm

Lifestyles of the AI-Native: Voice-coding, agent skills, hooks and scheduled tasks

Nick Nisi — Developer Experience Engineer, WorkOS · Zack Proser — AI Engineer, Applied AI, WorkOS

Most engineers are bolting AI onto a workflow that was designed for a pre-AI world. The result is a faster version of the same grind. This talk is about the other path: rebuilding the daily practice of software engineering from the ground up, around what agents are actually good at. Two senior practitioners from WorkOS will walk through how we actually work now as AI-native engineers — not in the aspirational sense, but the literal one. We think out loud and voice-code instead of typing our way to clarity. We package recurring expertise into agent skills so we're not re-explaining context every session. We wire up hooks that fire on the events we care about, and hand off scheduled tasks to agents that run overnight, while we're away from the keyboard, or otherwise off the clock. The throughline is intentional design: deciding what a human should hold onto and what should be delegated, then building the machinery to make that real. Because there are two of us, you'll see more than one set of habits — where our setups converge on the same patterns, and where they diverge based on how each of us thinks and works. The pitch isn't "do more." It's that an AI-native setup, designed deliberately, buys back attention and protects you from the burnout that comes from treating agents as a turbocharger for an old loop. Attendees will leave with a concrete mental model for voice-driven development, a pattern for authoring reusable agent skills, and working examples of hooks and scheduled automations they can adapt the same week.

WORKSHOP **Workshops Day 1 · Track 3** 12:10pm-1:10pm

2 hr deep dive on LLM Inference at Scale — Part 1 of 2

Harshul Jain — Senior Software Engineer, Audible Inc

Most engineers using LLMs can call an API. Far fewer can explain why their model is slow, why it's running out of memory, or how the inference engines powering every major LLM API actually work. This workshop walks through the full inference stack — from how a transformer generates a single token to serving billions of tokens a day with vLLM, SGLang, TensorRT-LLM, Ray, and KServe/llm-d. 60% explanation with live demos, 40% hands-on exercises. Attendees leave with a running vLLM server they benchmarked themselves. Based on the open-source practitioners handbook being built live at github.com/harshuljain13/llm-inference-at-scale

WORKSHOP **Workshops Day 1 · Track 4** 12:10pm-1:10pm

Build the Right Thing: Product Engineering for Software Developers

Kent C. Dodds — EpicProduct.engineer

There is nothing quite as demoralizing as finishing a feature and realizing you built the wrong thing. The code is clean. The tests pass. The ticket is closed. And none of it matters. This is happening more often, not less. AI makes it faster and cheaper to implement, which means teams can now waste entire sprints on the wrong idea at unprecedented speed. The bottleneck is no longer "can we build it?" It is "should we build it?" and "are we sure we understand the problem?" This session is a condensed introduction to product engineering for builders: the skills that sit upstream and downstream of implementation. We will not try to cover everything a full-day workshop would. Instead, we will focus on the highest-leverage ideas you can apply on Monday. ### What we'll cover 1. Validate before you build Most wrong builds start with an idea that was never tested. You will learn to separate real user pain from solution-shaped requests, and practice discovery questions that surface past behavior instead of hypothetical enthusiasm. 2. Prioritize what deserves to exist Not every good idea should be built now. Especially in the AI era, "we could build this" is not a reason to build it. We will work through a practical prioritization lens, including the Kano model, to help you distinguish fundamentals from delighters from distractions before your team commits. 3. Own the feature, not just the PR Product engineering does not end at merge. You will leave with a clearer picture of end-to-end feature ownership: staying close to users, setting up simple feedback loops, and improving what you shipped instead of moving on to the next ticket. ### Format This is a 2–3 hour session with Kent C. Dodds. Expect focused teaching, real-world examples, and short interactive exercises and discussion. This is not a full simulation lab or a ticket-closing coding workshop. It is judgment practice for engineers who already know how to ship. ### Who this is for Software engineers (and technical builders generally) who: - Have shipped something polished that nobody wanted - Feel pressure to move fast with AI and want a better filter for what deserves to exist - Want stronger product instincts without becoming a PM - Care about owning outcomes, not just closing tasks Some software engineering experience is assumed. No particular stack is required. PMs and designers often find this valuable too. ### What you'll leave with - Discovery questions for ambiguous work - A prioritization lens you can use before committing to a build - A clearer model for feature ownership and post-ship feedback loops - Language for stakeholder conversations when requirements are unclear

WORKSHOP **Workshops Day 1 · Track 5** 12:10pm-1:10pm

From Zero to Leaderboard: Building an End-to-End AI Agent Evaluation Pipeline

Wolfram Ravenwolf — AI Evangelist, Weights & Biases by CoreWeave

WORKSHOP **Workshops Day 1 · Track 6** 12:10pm-1:10pm *tentative*

To be announced

WORKSHOP **Workshops Day 1 · Track 7** 12:10pm-1:10pm

Beyond RAG: Build a Relational Context Engine from Scratch

Peter Werry — Founding Engineer, Unblocked

In this workshop we'll explore the importance of context engines in modern engineering workflows, and we'll look at why traditional RAG techniques are no longer enough to deliver the context agents need. We'll build a structured query engine that fills the gaps left by RAG, translating natural language into validated database queries over GitHub PR and Issue data. We'll implement schema-aware prompting, identity resolution, query validation, and error-driven retry loops, and you'll walk away with a working query engine for your GitHub repository.

WORKSHOP **Track 8** 12:10pm-1:10pm *tentative*

BrightData

WORKSHOP **Workshops Day 1 · Track 9** 12:10pm-1:10pm

Agent Speedrun: Idea → Code → Deploy → Observe, Fix → Ship

Elizabeth Fuentes Leone — Developer Advocate, Amazon Web Services · Sandhya Subramani — Senior Developer Advocate for Generative AI, Amazon Web Services

One agent. Fully deployed to production before the workshop ends. We'll take you from a blank file to a running production agent using Amazon Bedrock AgentCore and Strands Agents, covering the full lifecycle: ideation, coding the agent loop, deploying to serverless infrastructure, wiring up observability, breaking it intentionally, fixing it with tracing data, and shipping the final version.

SPONSOR **Track M** 12:10pm-1:10pm

Microsoft

1:15pm

SPONSOR **Track 1** 1:15pm-2:15pm *tentative*

Arize L&L

SPONSOR **Track 2** 1:15pm-2:15pm *tentative*

Neo4J L&L

SPONSOR **Workshops Day 1 · Track 3** 1:15pm-2:15pm

2 hr deep dive on LLM Inference at Scale — Part 2 of 2

Harshul Jain — Senior Software Engineer, Audible Inc

Most engineers using LLMs can call an API. Far fewer can explain why their model is slow, why it's running out of memory, or how the inference engines powering every major LLM API actually work. This workshop walks through the full inference stack — from how a transformer generates a single token to serving billions of tokens a day with vLLM, SGLang, TensorRT-LLM, Ray, and KServe/llm-d. 60% explanation with live demos, 40% hands-on exercises. Attendees leave with a running vLLM server they benchmarked themselves. Based on the open-source practitioners handbook being built live at github.com/harshuljain13/llm-inference-at-scale

SPONSOR **Workshops Day 1 · Track 4** 1:15pm-2:15pm

Build the Right Thing: Product Engineering for Software Developers — Part 2

Kent C. Dodds — EpicProduct.engineer

Continuation of Kent C. Dodds's workshop: Build the Right Thing: Product Engineering for Software Developers.

SPONSOR **Track 3 · Track 5** 1:15pm-2:15pm *tentative*

Snyk L&L

SPONSOR **Workshops Day 1 · Track 6** 1:15pm-2:15pm *tentative*

The model swap workshop

Pamela Fox — Principal Cloud Advocate, Microsoft · Arun Sekhar

Frontier labs are releasing new models constantly, and it is hard to know when “better” is better enough to justify touching a working system. On top of that, “just swap the model” often turns into real work because providers expose different APIs and different expectations around tools and structured outputs. The model swap workshop is a hands-on bake-off across frontier LLMs. We will run the same scenarios using multiple models (OpenAI, Anthropic, Kimi, and more) and compare results side by side for agentic tool use, structured outputs, and multimodal tasks. Swapping models is not just changing a model name. In this workshop, you will actually do the swaps, including moving between OpenAI-style Responses APIs and Anthropic-style Messages APIs, then see what breaks and what needs to change in your prompts, tool definitions, and JSON strategies. We will finish by running a small eval suite so you can quantify tradeoffs instead of relying on vibes. We will provide the Microsoft Foundry environment for access to the models, no account needed.

SPONSOR **Track 7** 1:15pm-2:15pm

Build a Document Triage Agent with Reducto: Classify, Extract, and More

Ingest a mixed corpus (think insurance claims, legal filings, or medical intake forms), classify each doc, extract relevant fields per type, and route to downstream handlers. We'll cover the full agentic document workflow end to end, and show you how to use Reducto Studio to do it. Learn how to build Reducto pipelines from scratch that can handle a corpus of mixed documents.

SPONSOR **Workshops Day 1 · Track 8** 1:15pm-2:15pm

Turning My Obsidian Vault Into a Local AI Engineer

Filip Makraduli — Machine Learning Engineer, Superlinked

SPONSOR **Track 9** 1:15pm-2:15pm

The Dark Arts of Skill Engineering

Paul Bakaus — Impeccable

SPONSOR **Track M** 1:15pm-2:15pm

Microsoft L&L

2:20pm

SPONSOR **Track 1** 2:20pm-4:20pm *tentative*

Arize 2hr

SPONSOR **Track 2** 2:20pm-4:20pm *tentative*

Neo4J 2hr

SESSION **Workshops Day 1 · Track 3** 2:20pm-5:30pm

Finteuning/Quantization/RL workshop with Daniel Han

Daniel Han — Unsloth

SPONSOR **Workshops Day 1 · Track 4** 2:20pm-4:20pm

AI Infrastructure from the Ground Up

Justin Lebar — Software engineer, OpenXLA

SPONSOR **Workshops Day 1 · Track 5** 2:20pm-4:20pm

Build a Platform, Unleash an Agent on it.... and Watch it Burn!

Michael Forrester — Principal Training Architect, Accenture

SPONSOR **Track 6** 2:20pm-4:20pm *tentative*

The AI Engineering Playbook: From Prototype to Production

Louis-François Bouchard — Co-founder and CTO, Towards AI

SPONSOR **Track 7** 2:20pm-4:20pm

Elastic

SESSION **Track 8** 2:20pm-4:20pm

Build with Perception Agents

Emile Baizel — Amazon AGI Lab · Shruti Arora — Amazon AGI Lab

Human-agent collaboration is changing, becoming more visual. Models can perceive, point, and verify, but most agents still rely on us typing a paragraph to explain what we're looking at. Meet perception agents: agents that see what you see, verify their own work, and let you point, draw, and describe instead of just typing.

SESSION **Workshops Day 1 · Track 9** 2:20pm-4:20pm

Parameter Golf Workshop

Zhengyao Jiang — Co-founder and CEO, Weco AI

SPONSOR **Track M** 2:20pm-4:20pm

Microsoft 2hr

4:30pm

SESSION **Workshops Day 1 · Track 1** 4:30pm-5:30pm

Reliable Computer Use Agents require coding

Ang Li — CEO and Co-Founder, Simular

Even the world's best computer-use agents cannot repeat their successes at the moment. Agents that write code — emitting structured selector-based actions instead of clicking pixels — break through that ceiling. We'll share two years of experience from Simular's production agent platform, the architectural decisions that mattered (refs over pixels, code as substrate, Simulang DSL), and a live demo: a 30-step unattended Windows workflow, side-by-side with a vision-only baseline. If you're shipping agents to real users, this is the playbook.

SESSION **Track 3 · Track 2** 4:30pm-5:30pm *tentative*

Snyk 1hr

WORKSHOP **Workshops Day 1 · Track 4** 4:30pm-5:30pm

Hill-climbing Skills: How to Improve Agents Without Touching the Model

Shubhankar Srivastava — AI Engineer, Browserbase

Agent Capability is now highly dependent on the markdown files read at runtime -- skills.This workshop treats skills as a first-class optimization surface. We borrow the concept of autoresearch (from Karpathy) and apply it to the skills your agents already read. You'll see how we at Browserbase did the same for browser agents, enabling our customers to scale the coverage of their browser agents while improving performance(2x faster runs) and optimizing for token spend(upto 10x cheaper).You'll leave with a working http://SKILL.md you generated through an auto-research loop, and a mental model for when skill optimization beats fine-tuning or prompt engineering.

WORKSHOP **Track 3 · Track 5** 4:30pm-5:30pm *tentative*

Snyk 1hr-B

WORKSHOP **Track 6** 4:30pm-5:30pm

SonarQube + OpenAI: Wiring Your Team for Agentic Development

As AI agents take on increasingly complex development tasks, the critical challenge has shifted from generation to verification. A growing body of evidence suggests that as models grow more capable, failures become more frequent and more convincing, making cognitive surrender among human reviewers an acute risk. This talk introduces Sonar's Agent Centric Development Cycle (AC/DC), a three-stage continuous loop of Guide, Verify, and Solve, as the engineering discipline teams need to build now. Teams that embrace AC/DC guide agents within their organizational standards before they write a line of code, verify output in real-time, and solve issues automatically without manual triage. This session will also feature a live demo of the SonarQube OpenAI plugin, showing how a well-guided agent produces code that is faster to verify and cheaper to fix.

WORKSHOP **Track 7** 4:30pm-5:30pm *tentative*

Attlassian

WORKSHOP **Track 8** 4:30pm-5:30pm *tentative*

Ref.

WORKSHOP **Workshops Day 1 · Track 9** 4:30pm-5:30pm

Burn your flags: How PayPal designs interactive CLI tools for agents

Mark Lummus — PayPal · Navinkumar Patil — Staff Software Engineer, PayPal

The common guidance for designing complex CLI tooling that agents can use is to add a "non-interactive" mode, where a normally interactive and flow-based command can be executed in a single pass by feeding it a bunch of flags. This is necessary for deterministic automation, but agents aren't scripts; they aren't really constrained in the same way, and they benefit greatly from the same step-by-step contextual workflows that humans do. The problem is that most agents (Codex excepted) cannot interactively drive a prompt-based CLI out of the box. In this workshop, PayPal goes deep on some techniques we've used in our upcoming `paypal` CLI that you can steal to make your complex CLI workflow tool agent-usable — without giving up the guardrails and guidance that interactive CLI tools provide. At the end of the workshop, you'll understand the benefits, patterns, and pitfalls of exposing a truly interactive mode for your CLI tool to agents. Summary: Build agent-friendly CLI tools with interactive workflows, guardrails, and practical design patterns for real-world automation.

SPONSOR **Track M** 4:30pm-5:30pm

Microsoft

Day 2 — Session Day 1 Tuesday, June 30, 2026

contents ↑

9:00am

KEYNOTE **Software Factories · Main Stage** 9:00am-9:10am *tentative*

swyx keynote and snyk track intro

Shawn Wang — Founder & Editor, Latent Space

9:10am

KEYNOTE **Software Factories · Main Stage** 9:10am-9:30am *tentative*

HOLD — Microsoft keynote

9:30am

KEYNOTE **Software Factories · Main Stage** 9:30am-9:50am

Opening Keynote: Topic TBD

Alexander Embiricos — Head of Enterprise Product, OpenAI · Romain Huet — Head of Developer Experience, OpenAI

TBD

9:50am

KEYNOTE **Software Factories · Main Stage** 9:50am-10:10am *tentative*

To be announced

10:10am

KEYNOTE **Software Factories · Main Stage** 10:10am-10:30am *tentative*

To be announced

10:45am

SESSION **Software Factories · Main Stage** 10:45am-11:05am

Codex Maxxing

Jason Liu

SESSION **Claws & Personal Agents · Track 1** 10:45am-11:05am

Your Agent Didn’t Fail. Your Harness Did.

Vinoth Govindarajan — Member of Technical Staff, Open AI

AI agents do not fail only because the model is wrong. Many production failures happen in the harness around the model: state is not persisted, two runs mutate the same session, a tool call never returns, an approval loses scope, or an internal success never becomes user-visible proof. This talk uses OpenClaw as a public case study to examine real harness failure modes and extract a reusable production model for AI engineers. We will look at how events enter an agent system, how session state is rehydrated, why single-writer lanes and throttles matter, and why tool execution needs scoped approvals and auditable receipts. The core idea is simple: a model proposes, the harness commits, and the receipt proves it. Attendees will leave with a practical 'run receipt' audit they can apply to their own agents: what woke it up, which state did it inherit, what authority did it use, what executed, and what evidence survived.

SPONSOR **Vision & OCR · Track 2** 10:45am-11:05am

To be announced

SESSION **Search & Retrieval · Track 3** 10:45am-11:05am

Pinecone 2.0

Edo Liberty — Founder & Chief Scientist, Pinecone

SESSION **Workshops Day 2 · Track 4** 10:45am-11:05am

Claude Managed Agents Workshop

Priyanka Phatak — Engineering Leader, Anthropic

Build an agent with Claude Managed Agents

SPONSOR **Security · Track 5** 10:45am-11:05am *tentative*

Snyk Session TBD

TBD — Snyk

SESSION **Voice & Realtime AI · Track 6** 10:45am-11:05am

The New Primitives: Building AI-Native Software

Kwindla Kramer — Daily

In the future, every piece of software with a human-facing surface will be built from new, LLM-centric primitives. We are just starting to invent these new primitives, including subagents, very long context, dynamic UI generation, and conversational voice input.

SESSION **LLM Recsys · Track 7** 10:45am-11:05am *tentative*

To be announced

SESSION **Forward Deployed Engineering · Track 8** 10:45am-11:05am *tentative*

To be announced

SESSION **Data Quality · Track 9** 10:45am-11:05am

State of Data

Sean Cai

SPONSOR **Track M** 10:45am-11:05am

M1

SESSION **Agentic Commerce · Leadership 1** 10:45am-11:05am

Inside the AI economy: What Stripe’s data reveals

Nilofer Rajpurkar — Stripe

SESSION **AI Architects: Show my Workflow · Leadership 2** 10:45am-11:05am

The Genesis Mission - Accelerating Science and National Security through AI

Mark Myshatyn — Genesis Mission

SESSION **Expo Stage NE · Expo Stage 1** 10:45am-11:05am

Every AI company is accidentally building a bank.

You're logging usage, billing later, hoping agents behave. They don't. Here's the architecture that fixes it before the invoice hits.

SESSION **Expo Stage 2** 10:45am-11:05am *tentative*

OpenAI Expo Session 2

SESSION **Expo Stage 3** 10:45am-11:05am

How PayPal Enterprise Payments handles agent-initiated payments across ChatGPT and Google AI Mode

PayPal Enterprise Payments has shipped integrations across the major agentic surfaces in the last six months each with human-in-the-loop confirmation and full transaction attribution back to the originating AI platform. We'll tour all three paths: ACP for ChatGPT apps (delegated payment tokens via complete_checkout, allowance validation, facilitator_details attribution), UCP with Google Pay for Google AI Mode (server-side tokenizationSpecification, parsing androidPayCards for the single-use token), and a preview of MCP Apps inline checkout, where the payment surface renders in-chat and card data never enters the LLM context. For each path we'll cover where PayPal Enterprise Payments fits, what the shopper and merchant each see, and the tradeoffs between them. You leave with working code and the docs to evaluate which path fits your stack.

SESSION **Expo Stage 4** 10:45am-11:05am *tentative*

Exa Expo Session

11:10am

SESSION **Software Factories · Main Stage** 11:10am-11:30am

Rise of the Software Factory

Tereza Tížková — Factory

The Stanford HAI 2024 AI Index reports a 30x productivity gap between AI leaders and laggards. The differentiator is not company culture, prompting technique or model selection, but the infrastructure. Organizations capturing outsized value from AI agents have machine-readable codebases, deterministic internal APIs, CI/CD pipelines with agent-addressable hooks, and permission models granular enough to scope exactly what an agent can touch. I believe the “agents as employees” framing is most useful if you operationalize it. An employee has persistent identity, episodic and semantic memory, scoped permissions that don’t get renegotiated every task, an audit trail, and a defined escalation path when things go wrong. Persistent computer use (with a stable execution environment that survives across steps) was the real inflection point that is making this possible. Some interesting production problems remain under-explored. How do you give an agent persistent identity across pull requests? How do you recover from partial failure mid-task without discarding completed work? How do you enforce code ownership policies when the author is a model? How do you bound token spend when pipelines spin up sub-agents recursively? This talk defines agent readiness as a concrete infrastructure checklist: structured codebases, deterministic APIs, per-agent scoped credentials, atomic and idempotent operations, structured execution traces, and explicit thresholds for when the agent stops and a human takes over. It presents research results in practice, and what are the steps organizations need to take to be fully agent-ready.

SESSION **Claws & Personal Agents · Track 1** 11:10am-11:30am

The OS runtime personal agents need

Ryan Dahl — Creator of Node.js & Deno; Deno Sandbox, Deno

Why personal agents that run untrusted LLM code need a sandboxed OS/runtime model, not just a compute sandbox.

SPONSOR **Vision & OCR · Track 2** 11:10am-11:30am *tentative*

HOLD — Jerry Liu / LlamaParse: Vision & OCR

Jerry Liu — Co-founder and CEO, LlamaIndex

SESSION **Search & Retrieval · Track 3** 11:10am-11:30am

The unreasonable effectiveness of BM25 for agentic search

Jo Kristian Bergum — CEO & co-founder, Hornet.dev

SESSION **Workshops Day 2 · Track 4** 11:10am-11:30am

Claude Managed Agents workshop

Priyanka Phatak — Engineering Leader, Anthropic

SPONSOR **Security · Track 5** 11:10am-11:30am

Dual-Surface Architecture: Serving Humans and Agents from the Same Tool Layer

Ethan Cha — Carlyle

SESSION **Voice & Realtime AI · Track 6** 11:10am-11:30am

Speech-to-Speech Model Research at Google DeepMind

Valeria Wu — Product Manager, Google DeepMind

Most voice interfaces today are built as a 3-way cascade system (ASR/LLM/TTS). This session explores the shift toward native speech-to-speech models that process audio end to end, focusing on product and research challenges in building real-time voice agents with fluid turn-taking, low latency, and enterprise-grade intelligence.

SESSION **Data Quality · Track 9** 11:10am-11:30am *tentative*

To be announced

SPONSOR **Track M** 11:10am-11:30am

M2

SESSION **AI-Native Enterprises · Leadership 1** 11:10am-11:30am

Building the engine while flying the plane — launching the Figma MCP server

Jesse Lumarie — Software Engineer, Figma

SESSION **AI Architects: Show my Workflow · Leadership 2** 11:10am-11:30am

Your Agent Evolved. Your Evals Didn't.

Ameya Bhatawdekar — Braintrust

Knowing which generation your agent is in, which failure modes your current evals are blind to, and what to build next is the difference between shipping with confidence and flying blind. Agent architectures have evolved through six generations; prompt, chain, ReAct loop, workflow graph, modern agent loop, AI harness. And each one quietly breaks the eval strategy of the generation before it. A prompt-quality rubric won't catch a bad tool call; a trace scorer won't catch memory poisoning. Using a single SRE incident response agent threaded through every generation, this talk shows exactly where each architecture outgrows its evals and what you need to close the gap.

SESSION **Expo Stage 1** 11:10am-11:30am

Give your coding agents the power of turbogrep!

Coding agents can grep the filesystem, but sometimes semantic search is more useful for finding the right files, especially on large codebases. Claude Code and Codex, unlike Cursor, do not use semantic search for code retrieval. There are good reasons for this, but Cursor has consistently demonstrated that semantic retrieval can materially improve code search to improve answer accuracy, increase code retention, and reduce token usage. In this session, we'll share a coding agent plugin for semantic codebase search alongside other modalities (BM25, regex/globbing/grep, filtering), and demonstrate how an agent can choose the right tool for the job. We'll share benchmark-style results that compare answer quality and token consumption with and without semantic retrieval across a small set of representative tasks.

SESSION **Expo Stage 2** 11:10am-11:30am *tentative*

WorkOS Expo 3

SESSION **Expo Stage 3** 11:10am-11:30am *tentative*

Kimchi Expo Session

SESSION **Expo Stage 4** 11:10am-11:30am

Agents, codebases, and teams: what it actually takes to ship together

Using a coding agent solo is one thing. Getting a whole team to trust agent-written code, agent-run reviews, and long-running agent work is another. That's where most teams stall. This talk is about what it actually takes to get there: how to shape a codebase so agents can work in it safely, how to earn a skeptical team's trust instead of mandating it, and the failure modes that only show up once agents are part of the daily workflow.

11:40am

SESSION **Software Factories · Main Stage** 11:40am-12:00pm

Everything is Conductor

Charlie Holtz — Founder & CEO, Conductor

Everything is Conductor now! I want to tell the story of how we came up with the original interface, what I think everyone (including us) is getting wrong and what's coming next.

SESSION **Software Factories · Track 1** 11:40am-12:00pm *tentative*

Gadgets: Personal app vibe coding that is actually safe

Kenton Varda — Principal Engineer, Cloudflare

We are entering the end game of Kenton's 15-year master plan. The architect of Cloudflare Workers, Durable Objects, Cap'n Proto, and Sandstorm.io, and the guy who coined the term "Code Mode", will demo Gadgets, an AI productivity suite which ties all these ideas together. We've all heard that the future is micro-apps customized for every niche, but how do we actually make that usable, how do we make it scale, and most importantly, how do we make it safe for even non-developers to use? Kenton will show how Gadgets solves these problems, including a sandbox design that makes it essentially impossible for apps to have vulnerabilities at all. He'll then open source it for your slop-forking pleasure.

SPONSOR **Vision & OCR · Track 2** 11:40am-12:00pm

Skill issue: stop deploying vision language models, use them with Skills to build e2e vision apps on edge

Merve Noyan — Developer Advocate, Hugging Face

SESSION **Search & Retrieval · Track 3** 11:40am-12:00pm

The Search Engine for the Agentic Web

Will Bryk — Co-founder and CEO, Exa

SESSION **Workshops Day 2 · Track 4** 11:40am-12:00pm

Claude Managed Agents workshop

Priyanka Phatak — Engineering Leader, Anthropic

SPONSOR **Security · Track 5** 11:40am-12:00pm

Your LLM Stack Is a 2008 Database With Better Marketing: Why ML Security Is Dominated by Misconfiguration, Not Missing Features

Lovina Dmello — Senior Software Developer, NVIDIA

ShadowRay exposed over a billion dollars of data through a missing authentication check. It wasn't a zero-day. It wasn't a clever new attack class. It was a default config someone never flipped off. That story is not the exception in production ML, it's the rule. We synthesized 139 peer-reviewed papers on production ML security across access control, runtime security, infrastructure, and operations. Five findings stood out, and one of them upends how most teams think about ML security: - Misconfiguration, not missing features, is the dominant failure mode. The mechanisms exist. Teams aren't using them, or are using them wrong. - Adversarial defenses impose 15–30% inference overhead, which is why almost no production system actually runs them. - ML-specific security tooling lags general DevOps tooling by years. - Security, data-science, and ops teams operate in expertise silos that create persistent gaps no single team can see. - LLM and multi-tenant GPU threats are evolving faster than defenses (prompt injection, RAG poisoning, GPU side channels). This talk walks through the four-pillar defense-in-depth framework, the six-category threat taxonomy that maps each attack to its primary and secondary defenses, and a four-level security maturity model that matches overhead budgets to deployment contexts. You leave knowing where your stack actually sits and which 3 misconfigurations account for most of the risk.

SESSION **Voice & Realtime AI · Track 6** 11:40am-12:00pm

Voice Agents Can Just Do Things

Charlie Guo — OpenAI

This talk argues that speech is becoming a control plane for software rather than just audio input/output. It introduces three practical patterns—voice-to-action, systems-to-voice, and voice-to-voice—and explains where realtime reasoning and tool-calling matter, and why chained STT/LLM/TTS systems start to break down as interactions become richer.

SESSION **Forward Deployed Engineering · Track 8** 11:40am-12:00pm *tentative*

To be announced

SESSION **Data Quality · Track 9** 11:40am-12:00pm

FrontierCode needs Frontier Data Quality

Deniz Birlikci — NeoLabs · Sam Lee — NeoLabs

SPONSOR **Track M** 11:40am-12:00pm

M3

SESSION **AI-Native Enterprises · Leadership 1** 11:40am-12:00pm

Agentic SDLC at Uber

Uday Kiran Medisetty — Uber

SESSION **AI Architects: Show my Workflow · Leadership 2** 11:40am-12:00pm

The Last Human Code Review: Building Trust in AI-Generated Code

Itamar Friedman — Co-Founder & CEO, Qodo

By the end of 2026, asking a human to review every pull request will be as optional as asking one to run every unit test manually. The tooling will be ready. The question is whether organizations are. In this talk, Itamar Friedman, CEO of Qodo, explains why we are approaching the end of line-by-line human code review as a default requirement and explores what has to be true for teams to get there. The barrier was never agentic AI capability. It was trust. And trust in automated review does not come from smarter models or faster feedback loops. It comes from systems that provide a trustworthy, concise and personalized proof-of-validation report. These systems are built on how engineering teams at specific organizations write their code: their own rules and standards, their PR history, their architecture decisions, their tribal knowledge that lives in comments and conversations and gets lost when engineers leave. Itamar will walk through the shift from PR-by-PR review toward continuous, context-based code review and governance, and share a practical approach to making human code review optional. If your team is shipping AI-generated code faster than humans can read it, join us for the discussion.

SESSION **Expo Stage 1** 11:40am-12:00pm *tentative*

Telnyx Expo Session

SESSION **Expo Stage 2** 11:40am-12:00pm

Agentic vs. Vector Search: An Eval-Driven Approach to Coding Agent Performance

Evals let you replace gut feelings with quantifiable decisions. This talk breaks the basic concepts of evals, including the four core components: datasets, tasks, scoring, and experiments. Then, to solidify the concept, we'll walk through a real eval comparing agentic search versus vector search for coding agents. We'll also cover practical challenges like tracing Claude Code subprocess calls and why a single eval run is never enough. You'll leave with a concrete framework for building evals that actually inform your ship decisions.

SESSION **Expo Stage 3** 11:40am-12:00pm *tentative*

Agents Don't Have Coworkers, They Have Hostages

Modern coding workflows are rife with vibe slop. As organizations scale, proper roles and governance systems must be well-defined to ensure a high sta

SESSION **Expo Stage 4** 11:40am-12:00pm

Would your AI agent get the job? A performance review framework for enterprise agents

There are dozens of ways to build an enterprise AI agent: agentic frameworks, direct LLM APIs, conversational AI platforms, vertical SaaS. They all claim to do the job. But how do you actually compare them on the same task, with the same data, against the same KPIs? This session presents a vendor-agnostic evaluation framework that treats AI agents the way enterprises treat new hires: set the role, define success criteria, run candidates through identical scenarios, and measure outcomes. The architecture uses any LLM to track positive and negative drift across agents against weighted goals, monitoring everything from hallucination rates and token consumption to user sentiment and conversation quality. Inputs are standardized. Outputs are both quantitative (accuracy, cost, hours saved) and qualitative (tone, clarity). The methodology supports continuous evaluation, not just pre-deployment benchmarks, but ongoing performance reviews that can compare agent work against human baselines. Walk away with a concrete, repeatable process for answering the only question that matters: which agent actually does the job?

12:05pm

SESSION **Software Factories · Main Stage** 12:05pm-12:25pm *tentative*

What is the future of the SDLC?

Thomas Dohmke — CEO and co-founder, Entire

Fireside Chat Proposal: What is the future of the SDLC? The tools and processes we use to ship software — tickets, repos, pull requests, deployments — were designed for humans writing every line of code. That world is over. ## Discussion Points ### What does the "next GitHub" need to look like? Is there even such a thing? What would a platform built from the ground up for the AI era actually require? ### What's broken about the developer lifecycle as we know it The manual system of software production — from issues to git repositories to pull requests to deployment — was never designed for the era of AI. Where are the biggest seams showing? ### The review bottleneck: is code review a dying paradigm? Despite the boom in agent-generated code, the human developer still sits at the center of the pull request. Traditional code review assumes a human wrote every line and can explain every decision — but when an AI agent generates hundreds of lines from a brief prompt, reviewers face a fundamentally different challenge. How do teams adapt? ### The economics of agentic development In 2026, headcount can no longer be measured in salaries and benefits alone. Tokens are a real cost, with engineers reporting thousands of dollars a month in usage. How should engineering leaders think about this new variable? ### What "agent-native" tooling actually looks like The unit of work is shifting — from code as output to code-plus-context (prompts, reasoning, decisions) as the artifact. What does tooling built around that reality look like in practice?

SESSION **Claws & Personal Agents · Track 1** 12:05pm-12:25pm

Tethered: Our Agents Are Us

Shu Fang — Two Sigma

SESSION **Search & Retrieval · Track 3** 12:05pm-12:25pm

If we want them to do Knowledge Work, we need to design Knowledge Agents

Benjamin Clavié — Member of Technical Staff, Mixedbread

It's tempting to assume that just like agents revolutionised coding, they will revolutionize other areas: legal, finance, advertising, and even medicine. All of those have in common that they are fundamentally knowledge work. And thankfully, humans have spent thousands of years searching for the best possible workflows for knowledge work. And yet, we seem to be disregarding all of these learnings, forcing every knowledge task into the shape that worked for coding. Today, we're going to talk about the history of knowledge work and how tools were co-designed to support it to understand how we should be building Knowledge Agents, themselves co-designed with their Knowledge Tools. This is key to avoiding falling into a "good enough" local optimum: think about legal clerking, a core part of the legal industry where information gathering and reasoning is performed to support the work of senior lawyers. The practice of clerking follows its own code, rules and best practices, which could not have feasibly emerged from studying software engineering: and similarly, there is no reason to believe knowledge agents could emerge from coding agents.

SESSION **Workshops Day 2 · Track 4** 12:05pm-12:25pm

Claude Managed Agents workshop

Priyanka Phatak — Engineering Leader, Anthropic

SPONSOR **Security · Track 5** 12:05pm-12:25pm

It's 10pm. Do You Know Where Your Agents Are?

Kim Maida — Head of Developer Relations & Founding GTM Engineer, Keycard Labs

Agents right now can sign legal contracts, run untethered, manage your dating profile, conduct financial transactions, and push code to production. Most agents have long-lived API keys and are dangerously overprivileged even when they're not making requests. In this talk, I'll demo how to solve the problem with the right access at the right time. You'll walk away knowing how to control agent access whether you're running coding agents from the CLI, building MCP servers, or connecting agents to third-party APIs.

SESSION **Claws & Personal Agents · Track 6** 12:05pm-12:25pm

Your Voice Agent is Just a Walkie-Talkie

Neil Zeghidour — CEO, Gradium

Everyone says cascaded voice pipelines are dead and native speech models are the future. Yet production environments are still dominated by STT-LLM-TTS stacks. Reconciling the natural flow of native audio with the elite reasoning of a cascaded agent remains an unsolved systems problem. This talk dissects the brutal technical trade-offs behind that counterintuitive reality. We will break down why your voice agent is still stuck behaving like a walkie-talkie and map out the specific technical roadmap required to build full-duplex AI that actually works.

SESSION **Forward Deployed Engineering · Track 8** 12:05pm-12:25pm *tentative*

To be announced

SESSION **Data Quality · Track 9** 12:05pm-12:25pm

Theta Software

Rayan Garg — Co-founder and CEO, Theta Software

SPONSOR **Track M** 12:05pm-12:25pm

M4

SESSION **AI-Native Enterprises · Leadership 1** 12:05pm-12:25pm

Scaling Code Quality: Building uReview, Uber's Multi-Agent Code Review Engine

Neha Singhal — Uber

SESSION **AI Architects: Show my Workflow · Leadership 2** 12:05pm-12:25pm

Prototyping as Leadership: How a CTO Ships with AI Agents

Hursh Agrawal — Co-founder and CTO, The Browser Company

I am a CTO and co-founder with a toddler, 15+ recurring meetings a week, 7 direct reports, and right now—7 open pull requests across two repos. Most engineering leaders eventually hit a wall where this kind of calendar tetris forces them to stop shipping code and start communicating solely through roadmaps. But what if AI agents didn't just act as coding assistants, but fundamentally restructured how executives use fragmented time to prototype the future? In this talk, I will share the exact multi-model workflows I use to plan with one model, implement with another, and build asynchronous play-and-feedback loops that fit perfectly between meetings. You will learn how to navigate code reviews for agent-assisted executive PRs, and leverage AI to shift your leadership style from telling your team what to build to showing them functional prototypes.

SESSION **Expo Stage 1** 12:05pm-12:25pm *tentative*

Your Agent Is Lying to You About Whether It Worked

Dat Ngo

Every span is green, every tool call returned cleanly, and the agent still regenerated the same plan 27 times before giving up invisible to any outcome metric, obvious in the trajectory. We pull up a real trace where the outcome looks healthy and the path is a disaster, then show Signal, our agent, surfacing it automatically: sweeping the project, ranking it above the noise, and linking straight to the offending trace with debugging evidence attached. The live version of the trajectory-over-outcomes argument, with a one-click path from "something's wrong" to "here's exactly where."

SESSION **Expo Stage 2** 12:05pm-12:25pm *tentative*

Composio Expo Session

SESSION **Expo Stage 3** 12:05pm-12:25pm *tentative*

Can LLMs write fast multi-GPU kernels? We built a benchmark to find out.

SESSION **Expo Stage 4** 12:05pm-12:25pm

Self-Improving Agents That Teach the Company Back

Agents forget too much. A run might solve a customer escalation, debug a deployment, or figure out the review pattern for a tricky code path, then the knowledge disappears into a transcript. At Runlayer, we started treating that knowledge as a product surface. Skills are reviewable, editable instructions that agents can load over MCP. An agent can start with a task, learn something useful while doing the work, and draft or update a private skill from that run. That skill loads into future runs for the same agent, stays inspectable by humans, and can eventually graduate into a team or org-level skill. The flywheel gets more interesting once a skill becomes useful beyond the agent that created it. A learned skill can move from one agent's private memory into shared organizational knowledge, then become available through the Runlayer plugin inside Claude Code, ChatGPT, and other AI clients employees already use. The agent does the work, captures the playbook, and the company gets better at that work everywhere agents are used. This talk walks through the architecture and product choices behind self-improving skills: post-run distillation, skill mutation tools, private-by-default scoping, runtime loading, UI inspection, promotion into shared skills, and the safety boundary between this agent learned something and everyone should now use it. The goal is an agent that leaves behind a better handbook for the next person, the next run, and eventually the whole organization.

12:30pm

12:30pm-1:30pm **Fireside Chat: Claude Code with Thariq Shihipar**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **h**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch & Learn**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

1:30pm

SESSION **Software Factories · Main Stage** 1:30pm-1:50pm *tentative*

Spin at the Gate Until Green: The Engineering Primitives Behind Self-Driving Codebases

Andrew Orobator — Reddit

SESSION **Claws & Personal Agents · Track 1** 1:30pm-1:50pm

Agents' next frontier: agent-to-agent and network effects

Jean-Denis Greze — Co-Founder & CEO, Town

MCP v. CLI was about how agents talk to tools. That’s not settled (but we’re camp MCP… mostly). Almost nothing has settled how agents talk to each other - and that's where the next wave of value (and network effects and virality) lives. At Town we run a personal AI agent in production inside real people's inboxes, calendars, and Slack, and we've built agent-to-agent (A2A) on our platform: 1:1 A2A messaging, agents that carry a short bio of one another, HITL when sensitive data is shared or write actions are involved, and early tests around 1:N A2A. I’ll talk about the why, the opportunity, and the production architecture underneath. Audience takeaway: a concrete mental model for building multi-agent systems on top of the data and surfaces users already live in, plus our learnings on early failure modes to avoid.

SPONSOR **Vision & OCR · Track 2** 1:30pm-1:50pm

From Ingestion to Agents: How Leading AI Teams Build on Document Intelligence

TBD — Reducto — Reducto

The agents of tomorrow are only as good as the context they reason on — yet most real-world data lives in messy, unstructured documents. In this session, we reveal the patterns that separate AI teams shipping reliable, production-grade agents from those stuck debugging pipelines. Drawing on patterns we've seen from AI-native startups to Fortune 10 enterprises, we'll cover what it takes to transform complex documents into clean, accurate context at scale across legal, finance, healthcare and more. From ingestion architecture to agent-ready outputs, walk away with the strategies top teams use to turn document chaos into competitive advantage.

SESSION **Search & Retrieval · Track 3** 1:30pm-1:50pm

How to Connect AI to Billions of Legal Documents

Simon Eskildsen — Co-Founder & CEO, turbopuffer · Jacob Lauritzen — CTO, Legora

Legora's foundational engineering challenge is connecting frontier LLMs to billions of legal documents so the models can efficiently solve end-to-end legal workflows without burning extra tokens. We'll share the retrieval architecture we built with turbopuffer that achieves: 1. Strict data isolation across millions of legal cases in a very security-conscious domain 2. Predictable search performance (<100ms p90 latency) on large contexts 3. High retrieval quality (95%+ recall@10) with fewer agent loops We'll retrospect on two architectures that failed to achieve all 3 (and why), and the key design factors that make the current solution work at our scale. Practical takeaways include: - How to evaluate per-tenant vs shared-index retrieval under strict data isolation - How to efficiently index and retrieve context to maximize relevance per input token - How to build a highly intelligent AI application when your inference budget is constrained

SESSION **Workshops Day 2 · Track 4** 1:30pm-1:50pm

Everybody Gets a Digital Clone! (Part 1 of 3)

Neil Zeghidour — CEO, Gradium

Walk out of this workshop with a deployed digital clone that makes your phone calls for you. We will skip the theory and immediately get our hands dirty wiring together OpenClaw, Twilio, and Gradium to build an autonomous voice agent on a live cellular network. You will tackle the hardest parts of real-time telephony: routing audio streams, handling human interruption, and killing latency. In 60 minutes, your AI will be ready to call restaurants for the daily special, book appointments, and actively negotiate on your behalf.

SPONSOR **Security · Track 5** 1:30pm-1:50pm

We Gave an Agent Production Code Access and Then Tried to Sleep at Night

Moritz Johner — Platform Architect / Senior Software Engineer, Form3

We let an agent touch production code to fix CVEs. That is either automation or a supply chain incident, depending on how honest your architecture is. PatchPilot started simple: find vulnerable dependencies, patch them, open a PR, let CI prove the fix, move on. Then reality showed up. The agent needed repository access, CI logs, credentials, and a Docker socket. Without that, it was useless. With it, every security reviewer in the room had a point. This is the production case study: what we gave the agent, what we refused, what infosec pushed back on, and where they were right. We will cover scoped permissions, constrained PRs, audit trails, approval gates, CI evidence, credential boundaries, and the gap between "it generated a patch" and "we can defend this change." Agentic remediation is not just developer productivity. It is a new participant in your software supply chain.

SESSION **Voice & Realtime AI · Track 6** 1:30pm-1:50pm

Tolan: Voice-First AI Companion

Paula Rambles

SESSION **LLM Recsys · Track 7** 1:30pm-1:50pm *tentative*

To be announced

SESSION **Forward Deployed Engineering · Track 8** 1:30pm-1:50pm *tentative*

To be announced

SESSION **Data Quality · Track 9** 1:30pm-1:50pm

The Base Model is Dead

Varun Singh — Pre-Training Lead, Arcee AI

It's a common belief that large language models are trained to be a good model of human web-text, and thus base models are "mirrors" of what we see on the internet. Historically, this was largely true, but no modern base model truly reflects the internet in the way that GPT-3 once did. Instruction data along with synthetic reasoning traces are moving earlier and earlier into the training pipeline, and "mid-training" has emerged as a new stage to accommodate longer datapoints that more concretely resemble downstream capabilities. As a result, pre-training no longer has the goal of creating a linguistic prior, but instead has the additional goals of baking in behavior and more atomic skills into the trained "base" model. Between this shift in what a base model is and the blurring of the lines between the different stages of model training, it's an open question as to what the best approach is here (at least outside the walls of the big labs). But I believe that the role we view the base model playing will continue to shift as we're pulled forward through new phases of model capabilities.

SPONSOR **Track M** 1:30pm-1:50pm

M5

SESSION **AI-Native Enterprises · Leadership 1** 1:30pm-1:50pm

Claude at Scale: Architecting the Context Stack for Large Codebases and Organizations

Aditya Gautam — Machine Learning Technical Lead, Meta

Speaker Pitch I'm an ML Lead at one of the largest technology companies in the world, building production AI systems across ML ranking, recommendation, and integrity at very large scale: codebases with hundreds of millions of lines, more than 10,000 engineers working in parallel, and models serving billions of users daily. I've given featured talks at Databricks Data + AI Summit 2025 (40,000+ attendees), AI Agent Conference NYC, MLOps Community, and Agent in Production '25, among others. My research on multi-agent LLM systems was published at ICWSM 2025 and I serve as a peer reviewer for NeurIPS, ICML, and AAAI. I have seen both sides of the full Claude ecosystem across every layer of the stack at this scale. I know what breaks: a stale CLAUDE.md propagating deprecated patterns silently, Skills that encode the wrong conventions and reproduce them reliably, Commands that work beautifully for one engineer and fall apart when shared across thousands. And I know what works: a hierarchical context stack with clear ownership boundaries at each layer, Skills and Commands designed as organizational contracts rather than personal shortcuts, Hooks and Plugins as the enforcement and auditability layer that makes the whole system trustworthy at scale. The mental model that ties it together, how to think about composing CLAUDE.md, Skills, Commands, Hooks, and Plugins as a coherent architecture rather than a collection of features, is what this talk is built around. This is a field report only I can give. --- **Session Description** At hyperscale, the question is never whether to use Claude. It is how to architect the Claude context stack so that thousands of engineers across infra, application, and ML research roles can best leverage it while remaining clear-eyed about its limitations and shortcomings. This talk moves past the basics and into the design decisions that determine whether your Claude deployment compounds productivity or compounds mistakes: how CLAUDE.md, Skills, Commands, Hooks, and Plugins compose into a coherent system, how each layer should be owned and optimized, and where the most expensive production failures actually originate. The mental model that unlocks this is treating the context stack as an architecture problem with the same discipline you would apply to any distributed system: clear ownership boundaries, explicit contracts between layers, and failure modes you have thought through before they hit production. The core of the talk is an honest account of both what breaks and what works, organized around a principled framework for how these components relate to each other and to the humans using them. CLAUDE.md encodes persistent architectural truth that must survive across every session and every engineer, and when it drifts it becomes the most dangerous file in your repository. Skills encode reusable procedural knowledge that, when designed as organizational contracts rather than personal shortcuts, become the highest-leverage investment a team can make in Claude adoption. Commands own the inner loop and work best when treated as shared infrastructure rather than individual automation. Hooks and Plugins are the enforcement and auditability layer, the part of the stack that makes the rest trustworthy at scale, and the part most teams underinvest in until something goes wrong. Getting these boundaries right looks completely different depending on the role: an infra platform team, an ML application team, and a research team each have a different optimal configuration, different risk surface, and different set of optimization opportunities. This talk maps all three explicitly. The practical payoff is a mental model and a set of design principles derived from both the failures and the wins at a company running some of the most complex ML systems in the world. You will leave knowing how to think about the Claude context stack as an architecture decision, how to optimize each layer for your team's specific role and risk profile, and what the anti-patterns are that look harmless at ten engineers and become critical failures at ten thousand.

SESSION **AI Architects: Show my Workflow · Leadership 2** 1:30pm-1:50pm

Serving 2 Million Models Without Melting: Scaling the Hugging Face Hub

Arek Borucki — Hugging Face

SESSION **Expo Stage 1** 1:30pm-1:50pm

Every Agent, Everywhere, All at Once

Coding agents are deaf to anything outside their own session, and a LangGraph or CrewAI one has no idea the others exist. Different vendors, different frameworks, different machines none of them share a way to work together. This demo fixes that live: the Claude Code on your laptop, Codex on your colleague's, a LangGraph agent you're running locally, and the OpenClaw on your Mac Studio at home collaborating on the same goal, going back and forth, full-duplex, across every vendor, framework, and machine line at once.

SESSION **Expo Stage 2** 1:30pm-1:50pm *tentative*

Deepmind Expo Session 1

SESSION **Expo Stage 3** 1:30pm-1:50pm *tentative*

Optiver Expo Session

SESSION **Expo Stage 4** 1:30pm-1:50pm *tentative*

Stop prompting

Greg Pstrucha

In this talk I dive into usage of tooling, type systems and frameworks to enforce guardrails and limit slop produced by AI agents inside large codebases.

1:55pm

SESSION **Software Factories · Main Stage** 1:55pm-2:15pm

Agents should talk to each other, so we built the protocol

Zach Lloyd — Founder and CEO, Warp

SESSION **Claws & Personal Agents · Track 1** 1:55pm-2:15pm *tentative*

Governance Is the Real Bottleneck to AI ROI

David Hsu — Retool

SPONSOR **Vision & OCR · Track 2** 1:55pm-2:15pm

From Scratch to SOTA: Training a 3B State-Space Vision Model for 1.4 Billion People

Krishna Srinivasan — Sarvam

SESSION **Search & Retrieval · Track 3** 1:55pm-2:15pm *tentative*

Don't Summarize. Sample. — How YouTube Re-Built Search for the LLM Era

Mihnea Munteanu — Software Engineer, YouTube

SESSION **Workshops Day 2 · Track 4** 1:55pm-2:15pm

Everybody Gets a Digital Clone! (Part 2 of 3)

Neil Zeghidour — CEO, Gradium

Continuation of Neil Zeghidour's hands-on workshop on building a deployed digital clone for real-time phone calls using OpenClaw, Twilio, and Gradium.

SPONSOR **Security · Track 5** 1:55pm-2:15pm

Using LLMs to Secure Source Code

Eugene Yan — Member of Technical Staff, Anthropic

SESSION **Voice & Realtime AI · Track 6** 1:55pm-2:15pm

5 Voice Agent Failure Modes You'll Hit in Week One

Vyas A — Plivo

A practical talk on the five voice-agent failures teams hit immediately in production: interruptions, turn-taking misfires, compounding latency, hallucinated actions, and audio/transcription mismatches. Each failure comes with a real example and a concrete fix.

SESSION **LLM Recsys · Track 7** 1:55pm-2:15pm *tentative*

To be announced

SESSION **Forward Deployed Engineering · Track 8** 1:55pm-2:15pm *tentative*

To be announced

SESSION **Data Quality · Track 9** 1:55pm-2:15pm *tentative*

To be announced

SPONSOR **Track M** 1:55pm-2:15pm

M6

SESSION **AI-Native Enterprises · Leadership 1** 1:55pm-2:15pm

AI Evals Platform for Cross-Functional Teams at Scale

Nachiket Paranjape — DoorDash · Swaroop Chitlur Haridas

SESSION **AI Architects: Show my Workflow · Leadership 2** 1:55pm-2:15pm

IT Admin for the AI Workforce: Why Your AI Agents Will Need Their Own IT Department

Aman Raj — Decawork

SESSION **Expo Stage 1** 1:55pm-2:15pm *tentative*

Who Approved That MCP Server? Governing the Tool Layer

Your developers are installing MCP servers faster than security can review them. An unvetted server is a direct line to your data. This talk shows how the Docker MCP Gateway puts every server and tool behind one org-managed catalog: vetted, signed, default-deny on anything unapproved, governed by the same policy engine as network and filesystem. Walk away with a hands-on demo: stand up a catalog, block an unvetted server, and watch policy enforce at the runtime.

SESSION **Expo Stage 2** 1:55pm-2:15pm

Voice Agents Are Mostly Invisible. Here's How to See Them.

Voice agents are one of the fastest-growing and hardest-to-debug categories: the failures live in latency, turn-taking, transcription drift, and tone none of which show up in a text log. We demo Voice traces and Session views, following a real voice session span by span, and Voice evals for scoring what text-only observability can't reach. A short, differentiated session on a problem most of the room is about to hit and few tools address.

SESSION **Expo Stage 3** 1:55pm-2:15pm *tentative*

Greptile Expo Session

SESSION **Expo Stage 4** 1:55pm-2:15pm

Deploying browser agents at scale

Not every browser agent trajectory is the same, and treating them like they are is how teams quietly burn budget on agents that never ship. This talk walks through the two trajectory types behind every browser agent, the cost/performance/maintainability tradeoffs that decide whether they hold up, and the concrete patterns for evaluating, hardening, and iterating on them.

2:25pm

SESSION **Software Factories · Main Stage** 2:25pm-2:45pm

What we learned by analyzing 1M AI-generated PRs

Daksh Gupta — Greptile

Charlie Holtz (@charlieholtz) - Founder & CEO of Conductor (launched Jul 2025). Mac app for orchestrating multiple coding agents in parallel. Used at YC cos, Linear, Vercel, Notion, Supabase. Ex-Replicate, ex-Point72, Brown CS. Angle: parallel agent orchestration / humans-as-conductors - distinct from the rest of the Coding Agents track. Ref tweet: https://x.com/charlieholtz/status/2047351098634338610

SESSION **Claws & Personal Agents · Track 1** 2:25pm-2:45pm *tentative*

Tool Execution layer for agents

Karan Vaidya — Composio

SPONSOR **Vision & OCR · Track 2** 2:25pm-2:45pm

You’re Not Thinking Big Enough: Rebuilding Food Systems from First Principles with AI Agents

Cody Menefee — Firecrawl

SESSION **Search & Retrieval · Track 3** 2:25pm-2:45pm

Your Agreements Are a Database You Can't Query. We're Fixing That

Hiral Shah — Senior Director of Product, Docusign · Sean Sodha

Agreements power every enterprise business, but mismanaging them has a big cost. Deloitte and Docusign found that inefficient agreement management costs businesses nearly $2 trillion in lost economic value annually, and organizations burn more than 55 billion hours on manual, disconnected agreement workflows. This session walks through what it takes to turn agreement repositories into structured, queryable data that can power search, analytics, and agentic AI workflows using Nvidia's open AI platform as the foundation. We'll cover the full pipeline: Extraction: How Nemotron Parse handles the hardest document understanding problems, including contract tables where pricing tiers, SLAs, and payment schedules live in dense layouts that traditional OCR destroys. Embedding and retrieval: How NeMo Retriever's embedding and reranking models make extracted contract data searchable and surfaceable for RAG-based applications. Agentic workflows: How these components connect to power contract analysis agents that can answer questions, surface risks, and act on obligations at scale. Nvidia will present the platform architecture and model capabilities. Docusign will share what we've learned evaluating this stack against real customer agreements: what worked, what broke, and what production deployment actually requires when serving over a billion users. Attendees will leave with a realistic view of where the Nvidia AI platform excels at document intelligence, where the gaps remain, and how to think about building agentic contract workflows in their own organizations.

SESSION **Workshops Day 2 · Track 4** 2:25pm-2:45pm

Everybody Gets a Digital Clone! (Part 3 of 3)

Neil Zeghidour — CEO, Gradium

Final continuation of Neil Zeghidour's hands-on workshop on building a deployed digital clone for real-time phone calls using OpenClaw, Twilio, and Gradium.

SPONSOR **Security · Track 5** 2:25pm-2:45pm *tentative*

To be announced

SESSION **Voice & Realtime AI · Track 6** 2:25pm-2:45pm

I Monitored Crime Audio. Voice Agents Scare Me More.

Sumanyu Sharma — Founder & CEO, Hamming AI

This talk reframes bad voice-agent calls as incident scenes and introduces a voice-agent forensics loop spanning transcript, waveform, latency waterfall, interruption points, ASR uncertainty, tool traces, system-of-record state, and outcomes. It focuses on monitoring, regression, and release-discipline for production voice systems.

SESSION **LLM Recsys · Track 7** 2:25pm-2:45pm *tentative*

To be announced

SESSION **Forward Deployed Engineering · Track 8** 2:25pm-2:45pm *tentative*

To be announced

SESSION **Data Quality · Track 9** 2:25pm-2:45pm

General Reasoning for Long-Horizon Agent Models

Ross Taylor — Co-founder & CEO, General Reasoning

Long-horizon agent models, reasoning loops, and the data/eval stack needed to make them reliable.

SPONSOR **Track M** 2:25pm-2:45pm

M7

SESSION **AI-Native Enterprises · Leadership 1** 2:25pm-2:45pm

Productionizing LLM Gateways: Architecture, Tradeoffs, and Hard Lessons from the Trenches

Kanish Manuja — Twilio

SESSION **Expo Stage 1** 2:25pm-2:45pm

Beyond Golden Signals: Monitoring in the Age of GenAI

The four golden signals (Latency, Errors, Traffic, Saturation) have been the foundation of application monitoring for years, and it still matters, but for GenAI applications, these signals alone leave significant blind spots. A request can return 200 OK with low latency while the response hallucinates, leaks PII, or costs much more than expected. This talk will walk you through what changes when you're monitoring non-deterministic, token-priced, prompt-injectable systems. We'll cover three additional monitoring dimensions: Cost (token attribution, model-mix tracking, wasted spend on failed requests), Safety (prompt injection detection, PII scanning, jailbreak attempts), and Quality (hallucination rate, relevance scoring, user satisfaction) and show why each one is necessary alongside your existing instrumentation.

SESSION **Expo Stage 2** 2:25pm-2:45pm *tentative*

Microsoft Presenting Expo Session 1

Microsoft — Microsoft

SESSION **Expo Stage 3** 2:25pm-2:45pm *tentative*

Buildkite Expo 1

SESSION **Expo Stage 4** 2:25pm-2:45pm

Continuous Engineering: Software Development for the Age of Agents

AI has changed everything about how we write code. But the hard parts of building software have gotten even harder: aligning your team, maintaining architectural integrity, and worst of all, reviewing the oceans of agent-driven code. The tools and processes we rely on git pull requests; code review were built for emailing patch files. We need a new paradigm. In this talk, we're going to explore Continuous Engineering, a new approach to software development that treats the agent thread as the core unit of collaboration. Branches should be as cheap as ideas, code should carry the context of the conversation that generated it, and the work should be available to your colleagues (and their agents) as it happens. We'll walk through what this looks like in practice, and what we're building to make it possible.

2:50pm

SESSION **Claws & Personal Agents · Track 1** 2:50pm-3:10pm

QwenPaw: building AI that you can trust

Eric Zhu — Alibaba

SPONSOR **Vision & OCR · Track 2** 2:50pm-3:10pm

The Best Models Still Reason Like Toddlers

Andrew Dai — Elorian AI

SESSION **Search & Retrieval · Track 3** 2:50pm-3:10pm *tentative*

Stop Chunking Like It's 2022

Yuval Belfer — Senior Developer Advocate, AI21 Labs

SESSION **Workshops Day 2 · Track 4** 2:50pm-3:10pm

Setting Yourself Up for Success — Part 1

Jason Liu

SPONSOR **Security · Track 5** 2:50pm-3:10pm

Agentic Chaos - What 86K+ Agent Codebases Reveal About 700K+ Exposed AI Systems

Bar Kaduri · Lidan Hazout

Over the past two years, AI agents have been stealthily becoming the new backbone of the global internet infrastructure. Autonomous systems capable of invoking tools, executing code, orchestrating workflows, and interacting with external services are now being built and deployed across production environments, developer workflows and tools, automation platforms, data pipelines, and enterprise systems as a whole. What's become glaringly obvious is that despite the speed of this adoption, almost nothing is known about how these systems are actually built or secured in the wild. To answer that question, we conducted one of the largest first-of-its-kind empirical studies of the agent ecosystem. We analyzed more than 86,000 public repositories implementing agent logic across frameworks including LangChain, LangGraph, CrewAI, AutoGen, and Model Context Protocol (MCP). We examined prompt construction patterns, tool implementations, authentication models, execution capabilities, and permission boundaries to understand how developers are building agents in practice. We paired this code-level analysis with internet-wide infrastructure measurement using Shodan, Censys, and ShadowServer to map where agent platforms are actually running in the wild. The research surfaced more than 700,000 exposed agent-related systems on the public internet, including Ollama inference servers, Ray clusters, n8n automation platforms, and MCP tool servers. Many of these systems were directly exposed to the internet with little or no authentication and, in numerous cases, were vulnerable to known high-impact CVEs. Together, these two datasets reveal a striking pattern. The same architectural assumptions and security shortcuts visible in agent codebases appear repeatedly in real deployments at internet scale. This talk presents the data, visualizations, and insights produced from this research. Using examples drawn directly from the dataset, we reconstruct several representative attack paths created by common agent design patterns. We show how seemingly harmless implementation choices, such as tool exposure, prompt construction shortcuts, and weak capability boundaries - can cascade into exploitable conditions once agents are deployed in real environments. We'll wrap up with practical architectural changes that agent frameworks and platform teams can adopt to prevent these patterns from becoming the next generation of supply-chain vulnerabilities.

SESSION **Voice & Realtime AI · Track 6** 2:50pm-3:10pm

Realtime Voice Agents with Frontier Intelligence

Bo Li — EliseAi

A deep dive into an EliseAI voice-agent harness that orchestrates multiple models to achieve realtime latency without sacrificing intelligence. The talk covers speculative transcription, async background tool injection, and TTS prefix caching/infilling to reduce latency while preserving capability.

SESSION **LLM Recsys · Track 7** 2:50pm-3:10pm *tentative*

To be announced

SESSION **Data Quality · Track 8** 2:50pm-3:10pm *tentative*

To be announced

SESSION **AI Architects: Show my Workflow · Track 9** 2:50pm-3:10pm *tentative*

Beyond the Benchmark: the New Frontier of Enterprise AI Reliability

Nick Heiner — Surge

SPONSOR **Track M** 2:50pm-3:10pm

M8

SESSION **AI-Native Enterprises · Leadership 1** 2:50pm-3:10pm

From AI-Assisted to AI-Native: Building a Frontier Development Team

Clare Liguori — Senior Principal SWE, Amazon Web Services

When features that took two weeks now ship in an afternoon, the bottleneck shifts from writing code to making decisions. Frontier teams have discovered this firsthand, achieving 3-10x productivity gains by fundamentally rethinking how developers work with AI agents. This talk covers the practices that separate frontier teams from those who merely "sprinkle" AI on their existing workflows: running agents asynchronously for hours, investing in comprehensive agent steering files, enabling local integration testing for agent self-correction, and automating everything from coding to operations to documentation. You'll learn how teams at Amazon slowed down to speed up, the temporary productivity dips they accepted, and the organizational changes required to sustain this velocity.

SESSION **AI Architects: Show my Workflow · Leadership 2** 2:50pm-3:10pm

How I automate my own job at Hugging Face using agents

Niels Rogge — Hugging Face

SESSION **Expo Stage 1** 2:50pm-3:10pm *tentative*

Resolve AI Expo Session

SESSION **Expo Stage 2** 2:50pm-3:10pm *tentative*

Bright Data Expo Session 2

SESSION **Expo Stage 3** 2:50pm-3:10pm *tentative*

Baseten Expo Session

SESSION **Expo Stage 4** 2:50pm-3:10pm *tentative*

Traversal Expo Session

3:20pm

SESSION **Software Factories · Main Stage** 3:20pm-3:40pm

fighting slop with slop

Vaibhav Gupta — Founder and CEO, Boundary

We haven't done a code review in two years. The last time I read every line of code in a PR was about six months ago. And we build a programming language with a runtime meant to replace V8. This is real engineering: compiler internals, runtime behavior, type systems, codegen, concurrency semantics, and FFIs across multiple languages. The thing that makes this possible is a technique we call "fight slop with slop" - every line of code is analyzed in depth by a sprawling toolchain of custom visualizers, linters, test snapshots and a whole bunch more. While the core language VM code has super high standards, a lot of these meta-tools are mostly vibe-coded. I'll dive deep into all the tactical things we've built, and how to adopt "fight slop with slop" in your own team

SESSION **Claws & Personal Agents · Track 1** 3:20pm-3:40pm

Everyone Gets A Software Company

Ben Guo — Zo Computer · Rob Cheung

SPONSOR **Vision & OCR · Track 2** 3:20pm-3:40pm *tentative*

Jia-Bin Huang

Jia-Bin Huang — Associate Professor, University of Maryland

SESSION **Search & Retrieval · Track 3** 3:20pm-3:40pm *tentative*

What We Learned After One Year of Building Our Deep Research System

Paul Iusztin — Senior AI Engineer; founder of Decoding AI Magazine, Towards AI

SESSION **Workshops Day 2 · Track 4** 3:20pm-3:40pm

Setting Yourself Up for Success — Part 2

Jason Liu

SPONSOR **Security · Track 5** 3:20pm-3:40pm

AI's Jurassic Park Period

Aaron Stanley — Head of Security and IT, dbt Labs

Early in my career, I accidentally and unrecoverably changed data I was collecting for a federal investigation. Twenty years later, with the help of AI and a career's worth of experience as a security leader, I intentionally did the same thing. Make no mistake, what my agent and I did together was dangerous. It was only because I had enough subject matter expertise in both the functional and risk issues that I could navigate it safely. We are in AI's Jurassic Park period: no matter how clearly we define the rules, models will search for paths to completion. And they are very good at making those paths look safe, reasonable, and correct even when they violate policy or basic intuition. Designing the right control set is about allowing for the right expertise to be injected at the right time in the co-creation process so we can move quickly and safely into the next evolution.

SESSION **Voice & Realtime AI · Track 6** 3:20pm-3:40pm

"My name is... my name is...": A Linguistic Framework for Debugging Voice AI Failures

Midam Kim — Senior Linguist and ML Engineer, ServiceNow

Every voice AI engineer has heard it: a caller repeating their name three times, getting more frustrated with each attempt. The logs look clean. Confidence scores look fine. Linguistics can help solving the mystery. By the end of this talk, you'll have a diagnostic framework for the failures that slip past standard metrics, a way to turn "the agent just didn't get it" into concrete, debuggable failure modes. The framework maps three levels of linguistic structure (sounds, words, and interactions) against the two dimensions every voice agent engineer already works in: what we hear (speech recognition) and what we speak (speech synthesis). That 3×2 grid surfaces problems your current tooling can't see, including: 1. Why your user cannot make your system understand their name 2. Why a single well-intentioned vocabulary hint can cause catastrophic drops in a non-English language 3. Why a transcript that's "cumulatively correct" can still ruin the user experience Drawing on examples from production multilingual voice AI work, I'll show where linguistic expertise connects to the engineering decisions you're already making and where it reveals failure modes that confidence scores will never warn you about. Who this is for: Voice AI engineers, ML practitioners on Voice AI pipelines, and anyone who's watched clean logs while their agent quietly fails real users.

SESSION **LLM Recsys · Track 7** 3:20pm-3:40pm *tentative*

Modality Misalignment and Originality Attribution in Short-Form Video: A Multi-Agent Approach at Platform Scale

Aditya Gautam — Machine Learning Technical Lead, Meta

SESSION **Forward Deployed Engineering · Track 8** 3:20pm-3:40pm *tentative*

To be announced

SESSION **Data Quality · Track 9** 3:20pm-3:40pm *tentative*

To be announced

SPONSOR **Track M** 3:20pm-3:40pm

M9

SESSION **AI-Native Enterprises · Leadership 1** 3:20pm-3:40pm

How to Get Your Org to Adopt Coding Agents (Without Shipping Garbage)

Eyal Blum — Figma

SESSION **AI Architects: Show my Workflow · Leadership 2** 3:20pm-3:40pm

Your Fine-Tuned Model Is Tech Debt: A 50x ROI House of Cards

Dan Bjornn — Lease End

SESSION **Expo Stage 1** 3:20pm-3:40pm *tentative*

Unblocked Add-On Expo Session (Extra)

SESSION **Expo Stage 2** 3:20pm-3:40pm

From Context to Memory: Your Agents Need a Real Memory Layer

Most agents don't really have memory. They have a context window, a pile of temporary files, maybe an AGENTS.md, and a retrieval step that attempts to build state from whatever the model can still see. You've seen the flashy demos, but these systems fall apart when an agent needs to recover from failure, revisit prior work, and observe if failures are less frequent over time. This talk explores agent memory as a systems problem. Effective memory isn't just storing data: it's an evolving knowledge layer with write filtering, consolidation, reflection, and forgetting. Agents need persistence, and they also need structure. Raw logs and Markdown scratchpads aren't enough. A real memory layer weights recency, combines retrieval techniques, and correlates episodic memories. Serious agent memory is inherently multi-model. The best systems use full-text search, semantic retrieval, graph relationships, and structured state to reconstruct context with far more precision than filesystem grep alone. This is where databases become essential as the foundation for real memory. Memory shapes how agents behave, adapt, and improve over time.

SESSION **Expo Stage 3** 3:20pm-3:40pm

Running a 20T-Token Data Pipeline: Infrastructure Lessons from Production

The problem. Curation algorithms tend to get the spotlight: model-based quality filtering, embedding-based deduplication, synthetic generation at scale, target distribution matching. The engineering behind them, the systems that actually run those algorithms reliably on petabytes of data and thousands of GPUs, usually gets overlooked. This session is about the engineering. What we built. The infrastructure behind two production data curation pipelines, on two very different shapes of workload: Arcee Trinity-Large-Thinking three model generations in nine months, with the curated corpus scaling from 8T to 10T to 20T tokens. Trinity-Large's 20T-token corpus included 8T+ synthetic tokens generated on clusters peaking at 2,048 H100 GPUs. Each generation incorporated deeper curation and broader domain coverage; the pipeline ran end-to-end multiple times, not once. Thomson Reuters legal 100B tokens of mid-training output, generated from TR's proprietary legal corpus, delivered as a deployment artifact and plugged into their existing SFT and DPO post-training. Different operational profile entirely: smaller scale, sensitive data, customer-environment integration. What you'll learn about. The metadata bottleneck. At trillion-token scale, fetching metadata from object storage across millions of files becomes the dominant source of idle time. We offload metadata management to Spark and use a lightweight file-level distribution scheme to drive idle time to near zero. Fault tolerance at multi-week scale. Long-running GPU inference jobs fail. We use one-to-one partition mapping between Spark and Ray jobs to get idempotent, resumable execution. A node failure no longer means reprocessing the dataset. Heterogeneous workload scheduling. Curation pipelines mix CPU-heavy preprocessing (Spark) with GPU-heavy inference (Ray + vLLM). An in-house scheduler routes each job type to isolated node pools, preventing resource fragmentation and ensuring critical training jobs aren't blocked by upstream CPU work. Inference tuning across models. vLLM defaults aren't right for every model. Tuning batch size, speculative decoding, and n-gram sampling per-model yields up to 40% throughput improvement, without over-engineering. Pipeline reproducibility. Treating a curated training corpus as a versioned deployment artifact rather than a one-off output. What that enables when a customer wants to run mid-training against a pre-trained base. For engineers building or operating large-scale data pipelines for ML training

SESSION **Expo Stage 4** 3:20pm-3:40pm *tentative*

How to prepare unstructured data for AI

An enterprise's first internal AI project worked brilliantly in the POC. But in production, the data became massive and messy: relevance and quality were unclear, sensitive information couldn't easily be filtered out, and there was no metadata to point AI toward the right answers. See how adding a curation and enrichment layer before ingestion cuts unstructured data prep from months to days, lowers compliance risk, and improves AI accuracy.

3:45pm

SESSION **Software Factories · Main Stage** 3:45pm-4:05pm

Scaling coding agents across an entire team

Dex Horthy — Founder & CEO, HumanLayer

SESSION **Claws & Personal Agents · Track 1** 3:45pm-4:05pm

Every Harness Will Become A Claw

Sam Bhagwat — Mastra

SPONSOR **Vision & OCR · Track 2** 3:45pm-4:05pm *tentative*

Perceptron Mk1 — Perceptron Inc

Armen Aghajanyan — Co-Founder & CEO, Perceptron AI

SESSION **Search & Retrieval · Track 3** 3:45pm-4:05pm *tentative*

What We Learned After One Year of Building Our Deep Research System

Paul Iusztin — Senior AI Engineer; founder of Decoding AI Magazine, Towards AI

SESSION **Workshops Day 2 · Track 4** 3:45pm-4:05pm

Setting Yourself Up for Success — Part 3

Jason Liu

SPONSOR **Security · Track 5** 3:45pm-4:05pm *tentative*

To be announced

SESSION **Voice & Realtime AI · Track 6** 3:45pm-4:05pm

The Goldilocks problem: when your Robot asks too much — or acts too soon.

Amit Desai — Director of Voice AI, Roku

Embodied agents are crossing from answering questions to taking physical actions — moving a box, turning a wheel — and people will command them by voice, because voice is the fastest, most natural interface we have. But voice is also the most error-prone, and when a misheard command drives a physical action, the failure isn't a wrong answer; it's human harm, damage, or an expensive, irreversible mistake. The field has never needed a serious way to handle voice-command errors, because informational agents made them cheap. Embodiment ends that. This talk replaces the usual hand-waving — "don't ask too much, don't get it wrong too much" — with a single number you can optimize. The core idea: both confirming and erring cost the user. A confirmation is friction — attention, time, a delayed action; a wrong action is a mistake cost, often higher given physical harm or expense. Put them on one ledger and you can measure a voice interface as average user cost per command, and make minimizing it the system's objective. From that falls a non-obvious rule — you confirm or not based on both cost and uncertainty: an expected value. I'll frame confirmation as just one option alongside acting, disambiguation (choices), and deferring; reason at the level of goals rather than low-level motion; walk the architecture (task hypotheses → user-cost model → confirmation policy); and show eval results from a simulated environment measuring regret against oracle behavior. I'll close with what worked applying this to voice in smart TVs, speakers, and navigation — and a challenge to bring this metric to robots, cars, and wearables before the errors do.

SESSION **LLM Recsys · Track 7** 3:45pm-4:05pm *tentative*

To be announced

SESSION **Forward Deployed Engineering · Track 8** 3:45pm-4:05pm *tentative*

To be announced

SESSION **Forward Deployed Engineering · Track 9** 3:45pm-4:05pm *tentative*

How to Stop Shipping Low-Quality RL Environments

Connie Fan — PM Lead, Google DeepMind

Training harnesses are data generators: when environments are flaky, stale, reward-hacked, or mismatched to production, models learn the wrong behavior. This talk distills common RL environment failures across coding, SaaS, and support-agent workflows, then offers a practical framework for building production-grade harnesses with clean signal, realistic state, fail-fast behavior, and trustworthy rewards.

SPONSOR **Track M** 3:45pm-4:05pm

M10

SESSION **AI-Native Enterprises · Leadership 1** 3:45pm-4:05pm

AI-transforming 18K engineers, 40K repos, and an agent swarm: what worked, what didn't

Oscar Mullin — SVP of Technology, MercadoLibre

Most AI transformation talks open with a timeline of how coding has evolved. We promise we won't. We **doubled delivery throughput** across **18K engineers** and **40K repos**, **improved performance with auto-research**, and **migrated 9,000 apps with autonomous agents**. We also built things that became obsolete in three months, picked the wrong abstraction at least twice, and have a graveyard of internal tools we'd rather forget. You'll get the architecture, the metrics that actually moved, and the ones we wish we hadn't measured. Still in progress. Already worth talking about. 18K engineers, **swarm of agents** and **thousands of new builders who don't code**.

SESSION **AI Architects: Show my Workflow · Leadership 2** 3:45pm-4:05pm

Unlock Agent Autonomy: The Runtime for AI-Native Systems

Tushar Jain — EVP of Engineering, Docker

The way software gets built in 2026 doesn't look like it did in 2024. The actors changed. Agents read and write entire codebases. Subagents spawn to chase down a flaky test, refactor a module, or triage an incident. But this shift doesn't stop at the SDLC. Agents increasingly invoke tools, interact with enterprise systems, install dependencies, call APIs, and orchestrate workflows across local machines, CI systems, cloud infrastructure, and organizational boundaries. The teams leaning into this shift are moving faster, and the gap is widening by the quarter. But few have the confidence to let agents operate autonomously across those environments. Not because the model capability isn't there. Trust isn't. Agents can pull a poisoned dependency, invoke an untrusted tool, wipe a database, leak sensitive data, or access systems they shouldn't. Prompt-level instructions won't close that gap, the unlock has to happen one layer down, at the runtime layer itself. Docker spent the last decade making it safe to ship software by getting the runtime right: isolation, network policy, trusted base images, and credentials. Agents are the next workload, and the same principles apply. Tushar Jain, EVP of Engineering at Docker, walks through what the runtime layer for AI-native systems looks like in practice: hardened runtime foundations, sandboxes that constrain what agents can touch, and governance controls that limit what agents can introduce, access, and execute across local, CI, cloud, and enterprise environments.

SESSION **Expo Stage 1** 3:45pm-4:05pm *tentative*

How We Built the Airbyte Agent MCP Server and CLI

Cam Kennedy — Airbyte

Agents need a reliable way to reach live business data. At Airbyte we built two interfaces for that, and this session is how. Cam built much of that surface. He covers the MCP server that exposes hundreds of sources through one endpoint with managed auth, and the CLI that's designed for agent harnesses rather than humans, with embedded help, packaged agent skills, and no credentials passed over the command line. Expect the real engineering: why a CLI turned out to fit autonomous agents better than the API or SDK, how auth works across the layers, and the tradeoffs the team made along the way. Come if you're building agent tooling or thinking about how to expose your own systems to agents cleanly.

SESSION **Expo Stage 2** 3:45pm-4:05pm

From Chatbots to Agents: How Reducto builds for Agent Experience to Enable Real Work

Many agent demos work. Most agent systems in production don't. The gap usually isn't the model or the tools. It's everything in between: how context gets structured, how multi-step tasks stay on track, how you handle the edge cases that only show up when real scenarios from real customers hit your pipeline. At https://reducto.ai/, we've spent the last couple of months building agent-first workflows for some of the most document-heavy industries out there. We've hit most of the failure modes you're probably hitting too. This talk shares what we've learned, from how to think about Agent Experience (AX) as a design layer, to the specific decisions that make complex workflows actually reliable in production. You'll walk away with tactical approaches to structuring context, model guidance, designing recoverable workflows, and building the feedback loops that let your system improve over time without a full rebuild.

SESSION **Expo Stage 3** 3:45pm-4:05pm *tentative*

Towards Reliable Financial Agents: How a 4B Model Outsmarted a 235B Giant

Large generalist models have excellent reasoning but this does not necessarily imply specialized knowledge and tool calling capabilities. They can still hallucinate column names, ignore constraints, and generate SQL that returns nonsensical results. The problem isn't intelligence it's reliability and specialization. In this talk we'll show how a 4B model was fine-tuned to outperform a 235B model on real financial analysis tasks. The key was not adding more reasoning ability, but enforcing tool discipline. Using synthetic data generation and reinforcement learning with the open-source rLLM framework, the model learned to explore schemas, validate outputs, and retry failures instead of hallucinating confident nonsense. One key result: tool-use fundamentals generalize. Training on simple tool interactions transferred to much harder, multi-step financial tasks. If you're building LLM systems that interact with databases, APIs, or internal tools, this talk focuses on the behaviors that actually matter and how to teach them without frontier-scale compute.

SESSION **Expo Stage 4** 3:45pm-4:05pm *tentative*

AI Enablement at Automattic: How a Remote Company Builds AI Fluency

Automattic is a remote company. About 600 of us will step away from regular work this year for an immersive AI program. That's a little over a third of the company. This talk walks through a field report of what we built and why: the curriculum, the cohort design, and what we've learned about making AI fluency work across a distributed organization.

4:30pm

KEYNOTE **Software Factories · Main Stage** 4:30pm-4:50pm *tentative*

To be announced

4:50pm

KEYNOTE **Harness Engineering · Main Stage** 4:50pm-5:10pm

In Code They Act, In Proof We Trust

Erik Meijer — Computer scientist and entrepreneur, Leibniz Labs

AI agents today execute on blind trust, and the failure modes are already in the headlines: a dealership chatbot agreeing to sell a $76,000 Chevy Tahoe for $1, a coding agent wiping a production database during a code freeze, and an "agent skill" installing a keylogger on a developer's machine. Automind enforces a different discipline: before any action runs, the agent submits an execution plan plus a machine-checkable proof of safety and correctness in Universalis, and a small checker decides whether the plan is allowed to execute. The result is left-shifted trust, with policy compliance established before the first side effect.

Day 3 — Session Day 2 Wednesday, July 1, 2026

contents ↑

9:00am

KEYNOTE **Autoresearch · Main Stage** 9:00am-9:10am *tentative*

2026 AI Engineering Survey and Arize Track Intro

Barr Yaron

TBD — Harbor launch keynote/session details to be finalized.

9:10am

KEYNOTE **Harness Engineering · Main Stage** 9:10am-9:30am *tentative*

Claude for long-horizon tasks

Lance Martin — MTS, Anthropic

Claude is capable of long horizon tasks. In this talk, we'll share lessons learned about building agent harnesses for reliable and secure long-horizon work. This include decoupling the brain and hands, self-verification, self-learning, and design for evolving agent harnesses.

9:30am

KEYNOTE **Software Factories · Main Stage** 9:30am-9:50am

In the Land of AI Agents, the Verifiers Are King

Tariq Shaukat — Chief Executive Officer, Sonar

As AI agents take on increasingly complex development tasks, the critical challenge has shifted from generation to verification. Hallucination is not a temporary bug. Evidence suggests that as models grow more capable, failures become more frequent and more convincing, making cognitive surrender among human reviewers…

9:50am

KEYNOTE **Autoresearch · Main Stage** 9:50am-10:10am

Perception Agents

Antje Barth — Member of Technical Staff, Amazon AGI Lab

Human-to-agent collaboration is changing, becoming more visual. The agents most teams ship today still wait for us to type a paragraph to explain what we're looking at. They cannot see a screen, navigate a UI that changes, or recover when an application throws an unexpected modal. That is the architectural gap between agents that demo well and agents that work alongside real teams in real software. Perception agents close it. They see and use computers the way people do, reason about what they see, and act with clicks and keystrokes. We call this shared perception between humans and agents, and the perception agent harness is what makes it reliable in production.

10:10am

KEYNOTE **Autoresearch · Main Stage** 10:10am-10:30am *tentative*

To be announced

10:45am

SESSION **Autoresearch · Main Stage** 10:45am-11:05am

First Steps Toward Automated AI Research

Richard Socher — CEO & Co-Founder, You.com / Recursive Superintelligence

SESSION **Sandbox & Platform Engineering · Track 1** 10:45am-11:05am *tentative*

To be announced

SPONSOR **Robotics & World Models · Track 2** 10:45am-11:05am

Building the simulation infrastructure for practical world model use

Christopher Manning

— Distinguished Member of Technical Staff at Moonlake AI; Thomas M. Siebel Professor in Machine Learning at Stanford University; General Partner at AIX Ventures, Moonlake AI

What is the most important capability for world model applications and the pursuit of embodied AI? We believe it is not a question of having the most beautiful pixels but the ability to reason about causality in multimodal environments. At Moonlake, we are working on building action-conditioned multimodal world models which provide spatial and physical state consistency over long time periods. We believe that building and training on synthetic worlds provides the data and compute efficient path to truly useful world models. We are building the simulation infrastructure platform for companies that need to build and manage worlds (assets, scenes, digital twins) at scale, including robotics/autonomy teams, digital factory operators, and game authors. Our product today primarily finds applicability in simulation and the operationalization of digital twins. Simulation can include training robotics, world models for AGI research, autonomous vehicles, or content creation for media and entertainment. Operationalization of digital twins involves the reconstruction of scans into reusable assets, e.g., turning image and point-cloud scans into sim ready assets for digital factory Integration projects. We are building toward a future where AI systems do not just generate worlds, but understand how they work. Moonlake learns from each workflow: The more workflows, failures, and human interventions that Moonlake sees, the better it becomes at reconstructing, validating, and preparing complex simulation worlds. The session will include discussion and demos.

SESSION **Memory & Continual Learning · Track 3** 10:45am-11:05am

Continual Learning Bench

Parth Asawa — CS PhD student, UC Berkeley

SESSION **Workshops Day 2 · Track 4** 10:45am-11:05am

Build realtime multimodal agents with Gemini Live

Thor 雷神 Schaeff — Developer Experience Engineer, Google DeepMind

The Gemini Live API is incredible versatile when it comes to building realtime AI experiences. From live translation across 2000 different language pairs to building realtime multimodal agents that can work across text, audio, and vision. This workshop gets you from zero to fully conversational agent in a matter of hours.

SPONSOR **Evals · Track 5** 10:45am-11:05am

Vending-Bench: Long-Horizon Agent Evals for a Simulated Vending Business

Andon Labs

Long-horizon agent evals via a simulated vending machine business, testing negotiation, pricing, and supplier management over 365 days.

SESSION **AI Designers/Design Engineers · Track 6** 10:45am-11:05am *tentative*

To be announced

SESSION **Computer Use · Track 7** 10:45am-11:05am

Computer Use at the Edge of the Statistical Precipice

Pierluca D'Oro — Founder, Stealth (formerly Meta)

Evaluating Computer Use Agents (CUAs) on interactive environments is fraught with methodological pitfalls that the field has yet to systematically address. We show that a 1MB replay script that blindly executes a recorded action sequence without ever observing the screen outperforms frontier models on prominent static benchmarks, and prove that its expected success rate is exactly equal to the source agent's pass@k in deterministic environments. We trace this and other failures to two root causes: non-principled environment design (static, unsandboxed, or unreliably verified environments) and non-principled evaluation methodology (naive aggregation and misuse of pass@k for stateful UI interactions). To address the first, we propose PRISM, five design principles for CUA environments and instantiate them in DigiWorld, a benchmark of 15 realistic sandboxed mobile applications able to evaluate agents in over 3.2 million verified unique configurations. To address the second, we develop an aggregation framework that correctly accounts for the nested structure of CUA benchmarks. All together, we show that principled environment design and rigorous evaluation methodology are not optional refinements but prerequisites for meaningful CUA research.

SESSION **Context Engineering · Track 8** 10:45am-11:05am

Build-Time vs. Run-Time: Why Your Dev Tools Will Fail in Production

Kurtis Van Gent — Senior Staff Software Engineer, Google · Prerna Kakkar — Google

SESSION **Posttraining & Midtraining · Track 9** 10:45am-11:05am

What's next after RLHF?

Diogo Almeida — Co-founder and CEO, TypeSafe AI

RLHF was a massive commercial success: roughly 100% of LLM usage is through RLHF'd models - but it was in many ways also a research failure. Let's talk about how it conquered the world, how it defied its creators expectations, why AI is in the bimodal state it's in (is it a bubble or a machine god?), and how to make AI actually transform the economy.

SPONSOR **Track M** 10:45am-11:05am

M1

SESSION **AI-Native Enterprises · Leadership 1** 10:45am-11:05am

Vertical Superintelligence: Making AI Work in America's Messiest Industries

Varun Shenoy — Cofounder, Long Lake · Rasmus Wissmann

SESSION **AI Architects: Tokenmaxxing · Leadership 2** 10:45am-11:05am

The Z/L Continuum: Should AI Engineers Still Read Code?

Alex Volkov — W&B from CoreWeave

SESSION **Expo Stage 1** 10:45am-11:05am *tentative*

Circle Expo Session

SESSION **Expo Stage 2** 10:45am-11:05am *tentative*

AI Engineering & Governance 2026 Trends

Wallon Walusayi — Qodo

AI Engineering & Governance 2026 Trends

SESSION **Expo Stage 3** 10:45am-11:05am *tentative*

Why AI Didn't Actually Make You Ship Faster

Gabriel Spencer-Harper — CEO, Meticulous

AI generates code faster than humans can review and verify it, and most engineering teams adopting codegen have hit the same wall: verification. In this session, Gabriel (CEO of Meticulous) breaks down why assertion-based testing has a structural ceiling that AI codegen has made impossible to ignore, what exhaustive verification actually requires technically (behavior capture, determinism, and backend isolation), and why the teams solving this now are the ones who will ship at the speed AI enables. The talk includes case studies from LaunchDarkly, which saw an 80% reduction in major frontend incidents after rollout, and Notion, which deployed verification infrastructure across every engineer on every PR to confidently adopt AI-generated code at scale.

SESSION **Expo Stage 4** 10:45am-11:05am *tentative*

Redesigning how software gets built

TBD — Sonar — Sonar

AI is already transforming how software is built, but most organizations are still treating it as a productivity tool rather than a governance challenge. The real question isn't whether to adopt AI-assisted development; it's whether your operating model is designed to control what comes out of it. This session reframes the AI development conversation around three practitioner horizons: organizations that are proficient with the status quo, those capturing velocity today, and those building toward the next frontier, where AI agents operate with genuine autonomy at scale. The gap between these horizons isn't model capability. It's operating model maturity. Most organizations are still applying AI to isolated steps in the development process. The real value only arrives when you redesign the system end-to-end: how work flows, how decisions are made, and how teams interact with AI as a core contributor. That transition requires something most teams haven't built: a governance layer that is accurate, consistent, repeatable, transparent, and auditable. This talk explores what that governance layer looks like in practice, including how to instrument controls at the point of generation, enforce standards without slowing agents down, and build the organizational confidence to let agents operate at scale without losing visibility or accountability. The companies getting the most out of agentic development aren't the ones with the best models. They're the ones with the strongest foundations. True governance isn't a gate at the end of the pipeline. In an agentic world, it's the architecture the pipeline runs on.

11:10am

SESSION **Autoresearch · Main Stage** 11:10am-11:30am

No Sleep Til Breakthrough: Building a Long Horizon Agentic Scientist

Kai Rikhye — Radical AI

At Radical AI we're building a self-driving lab for autonomous materials discovery, powered by a long-horizon agentic scientist that manages experimental campaigns spanning weeks. Before we built this technology, materials discovery required several years of hundreds of PhD-level specialists in national labs. Now, in a 16-week campaign we're able to discover 300 novel materials designed for specific performance goals. Building an agentic scientist involves challenges beyond the typical AI assistant. How do you design a system that can reliably and safely operate a physical laboratory containing heavy machinery? How do you manage context across thousands of papers read and hundreds of simulations run? How do you evaluate performance when sample sizes are relatively small and feedback loops are long? In this talk, I'll share learnings from building the AI layer powering a forthcoming 45,000 square foot lab supported by a $55 million seed round.

SESSION **Sandbox & Platform Engineering · Track 1** 11:10am-11:30am

To be announced

SPONSOR **Robotics & World Models · Track 2** 11:10am-11:30am

Building the simulation infrastructure for practical world model use

Christopher Manning

— Distinguished Member of Technical Staff at Moonlake AI; Thomas M. Siebel Professor in Machine Learning at Stanford University; General Partner at AIX Ventures, Moonlake AI

SESSION **Memory & Continual Learning · Track 3** 11:10am-11:30am

Scaling up Continual Learning

Ronak Malde — Trajectory.ai

SESSION **Workshops Day 2 · Track 4** 11:10am-11:30am

Build realtime multimodal agents with Gemini Live (continued 2)

Thor 雷神 Schaeff — Developer Experience Engineer, Google DeepMind

The Gemini Live API is incredible versatile when it comes to building realtime AI experiences. From live translation across 2000 different language pairs to building realtime multimodal agents that can work across text, audio, and vision. This workshop gets you from zero to fully conversational agent in a matter of hours.

SPONSOR **Evals · Track 5** 11:10am-11:30am *tentative*

To be announced

SESSION **AI Designers/Design Engineers · Track 6** 11:10am-11:30am

The Spatial Harness: Bringing Agents to the Canvas

Max Drake — tldraw

SESSION **Computer Use · Track 7** 11:10am-11:30am *tentative*

To be announced

SESSION **Context Engineering · Track 8** 11:10am-11:30am *tentative*

It’s Tokens All The Way Down: How RLMs are Different

Kevin Madura — AlixPartners

SESSION **Posttraining & Midtraining · Track 9** 11:10am-11:30am *tentative*

To be announced

SPONSOR **Track M** 11:10am-11:30am

M2

SESSION **AI-Native Enterprises · Leadership 1** 11:10am-11:30am

How to avoid disaster when vibe-coding a billing engine

Andrew Garvin — Metronome

SESSION **AI Architects: Tokenmaxxing · Leadership 2** 11:10am-11:30am

The Next Inflection Point: ChatGPT, Claude Code, OpenClaw — what's next?

Vlad Luzin — Band.ai

Every so often a new capability comes along that changes the way we see everything that follows. ChatGPT was one: overnight, the whole world was talking to a model. Claude Code was another: the model stopped answering questions and started writing and running real software, and the way engineers work changed under their feet. Then came OpenClaw — an agent that broke out of the terminal and ran loose across people's messaging apps, shells, and files — and people everywhere reorganized their daily lives around it almost overnight. That's what an inflection point is: a capability that doesn't just add something new, but resets what everybody thought was possible — and there's no going back. So what comes next? Three of these in two years — is that a pattern we can read, or just hindsight? This talk lines them up, looks for what they share, and makes the case for what the next world-changing capability is.

SESSION **Expo Stage 1** 11:10am-11:30am *tentative*

Runpod Expo Session

SESSION **Expo Stage 2** 11:10am-11:30am *tentative*

AI-Assisted Engineering: 5 Trends We're Seeing From 500+ Organizations

Justin Reock — Deputy CTO, DX

AI is reshaping how engineers work but what does that actually look like at scale? Drawing on data and patterns from more than 500 organizations, we break down the five most significant trends emerging in AI-assisted engineering today. This fast-paced theater session cuts through the hype to deliver concrete, evidence-based insights that engineering leaders can act on immediately. Key takeaways: Discover the top 5 AI-assisted engineering trends observed across 500+ organizations Understand how leading teams are integrating AI into their engineering workflows Leave with actionable strategies to apply at your organization

SESSION **Expo Stage 3** 11:10am-11:30am *tentative*

The Death of Keyword Search and the Rise of Agent-Readable Catalogs

Nixon Dinh — PayPal

As search shifts from classic keyword matching to more conversational experiences, product data quality becomes critical to LLM-powered retrieval. At PayPal, we tested how enriching traditional catalog data could help AI systems better find, understand, and rank products across large-scale commerce catalogs. We built a RAG-based AI judge to compare enrichment approaches and identify five patterns that consistently improved AI discovery results.In this talk, we'll share the evaluation framework, key lessons, and a practical approach for preparing enterprise data for conversational and agentic search.

SESSION **Expo Stage 4** 11:10am-11:30am *tentative*

FDE Playbook: Build an AI Support Agent and Give It a Voice

Matt Lawler — Forward Deployed Engineer Lead, AssemblyAI

Bio: Matt Lawler leads FDE for Onboarding at AssemblyAI, helping teams ship speech-to-text and voice AI to production, from model selection and architecture through deployment and scale. Description: Most support bots can read. Joey can talk back. In this session, AssemblyAI's Forward Deployed Engineer Lead, Matt Lawler, shares how his team built Joey, an AI support agent that increased end-to-end resolution rates from 10% to 75%. He'll walk through the architecture, key lessons learned, and how the team extended Joey into a fully voice-enabled agent.

11:40am

SESSION **Memory & Continual Learning · Main Stage** 11:40am-12:00pm

Memory Harnesses for Long-Running Research Agents

Stefania Druga — Research Scientist, Sakana.ai

At Sakana AI we build agents that run for hundreds of turns to read literature, run experiments, and draft papers. The model rarely breaks. The harness around it is the weak point: the agent contradicts a decision it made 80 turns ago, redoes finished work, or drifts from the question it started on. This is the binding-constraint thesis. For long-horizon tasks, reliability is set as much by the harness as by the model as clearly instantiated in autoresearch recent efforts. This is a field guide to the harness's memory layer. I'll trace a real research agent through its lifecycle, show exactly where context rot and drift set in, and cover the patterns that hold over 100+ turns: three-tier memory, progressive disclosure, recall-first compaction, sub-agent isolation, and architectural memory beyond the vector database. I will show how to measure whether your memory harness actually helps, at the trajectory level, so you stop tuning prompts to fix what's really a state-management bug.

SESSION **Sandbox & Platform Engineering · Track 1** 11:40am-12:00pm

To be announced

SPONSOR **Robotics & World Models · Track 2** 11:40am-12:00pm *tentative*

HOLD — Dyna Robotics / Jason Ma

Jason Ma — CTO and co-founder, Dyna Robotics

TBD — Dyna Robotics talk for Robotics & World Models track.

SESSION **Memory & Continual Learning · Track 3** 11:40am-12:00pm

Jack Morris — Context Is Not Memory, Updating Weights Is

Jack Morris — AI researcher, Cornell / Meta FAIR

A case for when context is enough, and when updating weights may be the real memory mechanism.

SESSION **Workshops Day 2 · Track 4** 11:40am-12:00pm

Build realtime multimodal agents with Gemini Live (continued 3)

Thor 雷神 Schaeff — Developer Experience Engineer, Google DeepMind

The Gemini Live API is incredible versatile when it comes to building realtime AI experiences. From live translation across 2000 different language pairs to building realtime multimodal agents that can work across text, audio, and vision. This workshop gets you from zero to fully conversational agent in a matter of hours.

SPONSOR **Evals · Track 5** 11:40am-12:00pm *tentative*

HOLD — Evals at Uber

TBD

SESSION **AI Designers/Design Engineers · Track 6** 11:40am-12:00pm

The Design-Code Roundtrip That Isn't

Jonathan Gordon — ReWeaver AI

SESSION **Computer Use · Track 7** 11:40am-12:00pm

The Dark Arts of Web Automation: Teaching Agents to Use Websites Like Humans

Corey Gallon — Managing Director, Rexmore

Anything you can do in a browser, your agent can do too. Not by tiptoeing through an MCP server one polite, token-burning call at a time -- properly, programmatically, the way you'd drive any other tool. I'll show you how with chrome-agent, an open source wrapper over the Chrome DevTools Protocol that has become irreplaceable in my everyday work. If you'll ever do a browser task more than once, step-by-step MCP browsing is slow, brittle, and bills you tokens for every single click. A CLI straight onto CDP makes the whole browser programmable: loop it, pipe it, script it, walk away. Write it Tuesday, run it a thousand times Wednesday, all without a second of AI agent babysitting. We'll dispel the MCP hype and myths, with successful demonstrations of cheeky things like: the power of CLI-based browsing and how its so much more capable than mere MCP; reaching through those oh-so-clever cross-origin iframes to clear the verify you're human checkboxes; showing that a JavaScript .click() is not a click, rather, just a function call in a costume that is banhammerable; ultimately, proving that a CDP browser operates just like a meatbag with a mouse and keyboard. You'll learn how to point your AI agents at real, messy, uncooperative websites and web applications and have them get things done exactly the way that you would.

SESSION **Context Engineering · Track 8** 11:40am-12:00pm *tentative*

500 Skills, Zero Fine-Tuning: LinkedIn's Playbook for AI Agents That Actually Know Your Codebase

Ajay Prakash — Linkedin

SESSION **Posttraining & Midtraining · Track 9** 11:40am-12:00pm *tentative*

HOLD — Fleet AI

TBD — Fleet AI — Speaker TBD, Fleet AI

Hold for Fleet AI. Company focuses on simulated environments / training gyms for AI agents and fits the posttraining / RL environments theme.

SPONSOR **Track M** 11:40am-12:00pm

M3

SESSION **AI Architects: Tokenmaxxing · Leadership 1** 11:40am-12:00pm

I Let Agents Refactor My Codebase for 3 Weeks. Then I Read the Code.

Keiji Kanazawa — Microsoft

SESSION **AI Architects: Tokenmaxxing · Leadership 2** 11:40am-12:00pm

How to Kill the Code Review

Ankit Jain — Aviator

SESSION **Expo Stage 1** 11:40am-12:00pm

Fault-Tolerant Training at Scale: Making Hardware Failures a Non-Event

Hardware failures in large-scale distributed training are inevitable when you're running thousands of GPUs, they happen multiple times a day. The standard response is manual intervention: an engineer gets paged, SSHs into the cluster, and spends an hour fixing something the infrastructure should have handled automatically. That lost time compounds directly into wasted compute and delayed research. This session walks through the self-healing platform Crusoe built to eliminate that manual loop entirely a managed Slurm environment running on Kubernetes, with automated node failure remediation and real-time cluster observability and how these components work together so hardware failures become a non-event. We'll cover this architecture end-to-end: how running Slurm on Kubernetes unlocks infrastructure resilience that traditional GPU clusters don't have, how automated hardware monitoring and node remediation can eliminate manual intervention entirely, and how full observability into every remediation event keeps engineering teams informed without keeping them on-call. For teams that want deeper control, we'll also discuss open-loop remediation, which gives teams full control over the node replacement process for application-specific workflows.

SESSION **Expo Stage 2** 11:40am-12:00pm

How to generate mergeable code with a context engine

Your agents are fast, capable, and completely context-blind. They generate code that compiles but doesn't reflect how your system actually works. You're likely already seeing the impact: ballooning token costs, longer review cycles, and inconsistent outputs. More MCPs, rules, and bigger context windows give agents access to information, but not understanding. In this session, we dissect how teams pulling ahead use a context engine to give agents exactly what they need for the task at hand. Includes a short demo showing the workflows a context engine can augment.

SESSION **Expo Stage 3** 11:40am-12:00pm *tentative*

Harness Engineering and Continual Learning in LangSmith

Jake Broekhuizen

Short Description: A practical session on building evaluation and feedback loops that help agents improve over time. We'll look at how teams can use LangSmith to connect traces, datasets, evaluations, human feedback, and production observations into a continual learning loop. Full Abstract: This session will explore harness engineering as a practical way to build better agents. I'll talk about how LangSmith can help teams inspect real behavior, turn traces into datasets, run evals, capture feedback, and use that loop to continually improve their systems. The goal is to give people a simple mental model for moving from ad hoc debugging toward a more systematic way of learning from production and making agents better over time.

SESSION **Expo Stage 4** 11:40am-12:00pm *tentative*

The Enterprise Agentic Gap: When Developer-Level AI Tools Hit Millions of Lines

Dan Adler — Sourcegraph

Agentic coding tools have transformed individual developer workflows but owning a large codebase with millions of interdependent lines across multiple code hosts is a different problem entirely. Off-the-shelf AI coding tools weren't built for it, and at scale, they break down in ways that aren't obvious until you're already in trouble. This talk covers the failure modes you'll hit when applying developer-level agentic tools to enterprise-scale migrations, and how Sourcegraph's agentic migrations solution was built to solve what others couldn't.

12:05pm

SESSION **Autoresearch · Main Stage** 12:05pm-12:25pm

auto-nanogpt

Elie Bakouch — Researcher, Prime Intellect

SESSION **Sandbox & Platform Engineering · Track 1** 12:05pm-12:25pm

Your agent needs a sandbox, not a desert

Samuel Colvin — Pydantic

SPONSOR **Robotics & World Models · Track 2** 12:05pm-12:25pm

Tell the Robot What You Want

Sandhya Subramani — Senior Developer Advocate for Generative AI, Amazon Web Services

What if you could command a robot just by talking to it? This session introduces an open-source agentic AI framework that lets developers control physical sensors and actuators using natural language, by exposing hardware as programmable agent tools through a unified interface. The agent interprets the request, selects appropriate tools, and orchestrates execution. We explore a hybrid model where low-latency perception and actuation run locally on edge hardware, and higher-level reasoning and multi-step planning are delegated to cloud-based agents when needed. This preserves real-time responsiveness while enabling richer reasoning. A live robot demonstration anchors the session. Using the SO101 robotic arm powered by NVIDIA GR00T on Jetson hardware alongside HuggingFace LeRobot, attendees see how an instruction such as "place the apple in the basket" moves from conversation to perception to physical action.

SESSION **Workshops Day 3 · Track 4** 12:05pm-12:25pm

Build realtime multimodal agents with Gemini Live (continued 4)

Thor 雷神 Schaeff — Developer Experience Engineer, Google DeepMind

SPONSOR **Posttraining & Midtraining · Track 5** 12:05pm-12:25pm *tentative*

To be announced

SESSION **AI Designers/Design Engineers · Track 6** 12:05pm-12:25pm

Mousepower: agents that can’t be measured, can’t be managed.

Maximillian Piras — Yutori

SESSION **Computer Use · Track 7** 12:05pm-12:25pm

Bringing agents onto the world wide web

Paul Klein IV — Founder & CEO, Browserbase

The web wasn't built for agents. Heavy HTML, human-first UIs, and a DOM that can hijack the model's context. Still, agents browse it for millions of hours every month through Browserbase, across teams like Ramp, Shopify, and Lovable. This talk walks through that browser agent harness layer by layer, from the security boundary between DOM and model to caching, Agent Identity, and the infrastructure that provisions browsers at scale, and where browser agents go once it is in place.

SESSION **Context Engineering · Track 8** 12:05pm-12:25pm

Your agents lack context: Here's how to fix "You're absolutely right!"

Brandon Waselnuk — Founder, Unblocked

Every AI coding tool can generate code. Very few can generate the right code for your organization, because they're missing context. They don't know why your team chose Redis over DynamoDB, what the team decided in a Slack thread earlier today about the auth migration, or which architectural patterns your principal engineers actually enforce in review. This talk is a practitioner's guide to building a context engine: the reasoning layer that continuously ingests & synthesizes organizational knowledge across disparate sources into unified, queryable understanding. I'll walk through the problems you actually have to solve — reasoning across systems that don't agree with each other, searching globally before you can reason, maintaining identity-scoped permissions so every user and agent only sees what they should, and personalizing results based on who's asking and what they're working on. These are the engineering challenges that make naive RAG fall short, drawn from real lessons building this at scale.

SPONSOR **Track M** 12:05pm-12:25pm

M4

SESSION **AI-Native Enterprises · Leadership 1** 12:05pm-12:25pm

AI-Native Organisations runs on Skills: How to Extract, Structure, evaluate and Scale Them

Imad Touil — QuantumBlack, AI by McKinsey

SESSION **AI Architects: Tokenmaxxing · Leadership 2** 12:05pm-12:25pm

The Death of the Code Review

Laurie Voss — Head of Developer Relations, Arize AI

SESSION **Expo Stage 1** 12:05pm-12:25pm

Your agent architecture has a half-life of 6 months

A short history of the right way to build an agent: RAG, ReAct, prompt chaining, orchestrator-workers, MCP, CLI, MCP again... CLI again?? Every time you adopt a trend you rebuild your architecture. In this talk, Dan Farrelly, Inngest cofounder and CTO, is not going to tell you what comes next. He's going to show you how to build so it doesn't matter. He'll cover the core primitives that show up in every production agent, how bringing decisions closer to code provides more stack flexibility, and why the right execution layer unlocks faster iteration.

SESSION **Expo Stage 2** 12:05pm-12:25pm *tentative*

From Stateless to Stateful: Orchestrating Real-Time Voice & Messaging Agents with Twilio and Amazon Bedrock

Rishab Kumar

We have all had that maddening customer service experience: you text a support line about a delayed flight, receive a confirmation, but when you call in a minute later, the voice agent asks, "How can I help you today?" completely blind to the SMS you just sent. This is the "Channel Amnesia" problem. While businesses are pouring billions into generative AI, most agents are still built on stateless architectures that forget customer context the second a session ends. In this session, we will cure AI amnesia. You will learn how to orchestrate stateful, production-grade AI agents across SMS and Voice using Twilio Agent Connect and Amazon Bedrock. We will dive into why traditional serverless compute fails stateful agents, how to leverage AWS Fargate for isolated, long-lived sessions, and how to configure Bedrock AgentCore over WebSockets to hit sub-50ms streaming voice latency. No slide-ware here expect a live, cross-channel demo and open-source code you can deploy tomorrow.

SESSION **Expo Stage 3** 12:05pm-12:25pm *tentative*

Harnessing Collective Agent Intelligence for Open Science

Everyone building AI products eventually draws the same diagram: boxes representing data sources, arrows pointing at the model, and a label that says "context." What that diagram doesn't show is the system that has to run underneath it deciding, for each request: which sources to consult, whether to fetch live or use cached data, if the user is actually allowed to view that data, how to stitch it all together before the latency budget runs out. And it hides the counterintuitive part: fetching more context usually makes your answers worse, not better. At Merge, we reframed context graphs as control planes, helping companies scale context graphs to hundreds of thousands of users with sub-300 ms latency. This talk walks engineers through the system design at scale: how to tier data freshness, why provenance isn't optional once third-party systems are in the loop, and how to decide when fetching less context is the right call. Attendees will leave with a mental model for context system design that separates the orchestration decisions from the retrieval layer.

SESSION **Context Engineering · Expo Stage 4** 12:05pm-12:25pm *tentative*

Prompt, Memory, Weights: The Architecture Decisions Most AI Teams Make by Accident

Anant Srivastava

The interesting engineering in production AI isn't in the model. Your knowledge lives in files, databases, and APIs: docs, runbooks, conversations, code. The model just reads tokens. So the real architectural question is which path that knowledge takes to inference: into the prompt directly, into memory for retrieval on demand, or into the weights through fine-tuning. Most teams treat these as a ladder. Start with prompts, escalate to RAG, eventually fine-tune, as if each step is a more advanced version of the last. The field is converging on a different answer: they solve different problems. The prompt shapes behavior and constraints. Memory grounds the model in current, citable knowledge. Weights harden specialized reasoning and format. They're not substitutes you graduate between; they're complementary, and the failures come from using one to do another's job. Fine-tuning to teach the model facts it should have retrieved is the classic trap: you bake in knowledge that's stale the day it ships, and you still can't cite it. This is an opinionated take on all three: when each is the right call, when each is a trap, and the part most teams never build, the circulation between them. Memory that captures what the agent does becomes the dataset you fine-tune on; fine-tuning changes what's worth retrieving; the loop compounds. Get the three paths right and they stop being a pipeline you climb and start being an architecture that learns.

12:30pm

12:30pm-1:30pm **Google DeepMind GenMedia Panel**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch & Learn**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch & Learn**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

1:30pm

SESSION **Autoresearch · Main Stage** 1:30pm-1:50pm

Closing the Loop: An Autonomous AI Research Agent

Tim Sweeney

The holy grail of agentic AI tooling is the autoresearch loop: an agent that can sift through your experiments, create visualizations, propose a hypothesis, launch a training job, read the results, and try again autonomously. In this session, we'll show new autoresearch capabilities built directly into the W&B Models web and iOS apps. We will demo these live using a real-world fine-tuning project, covering everything from launching jobs and reading loss curves to surfacing outlier runs that consume researcher hours and recommending the next steps. Then you'll learn how the eval-driven development loop in W&B Weave makes agents like this trustworthy. You'll see how production traces become benchmarks, and how only the agents that beat the bar make it to production. Join us to learn the same loop we use to improve our own agentic features.

SESSION **Sandbox & Platform Engineering · Track 1** 1:30pm-1:50pm

To be announced

SPONSOR **Robotics & World Models · Track 2** 1:30pm-1:50pm

Frontier Robotics Research

Deepak Pathak — Co-Founder & CEO, Skild AI

SESSION **Workshops Day 3 · Track 4** 1:30pm-1:50pm

The Agentic Power User's Playbook: Tips and Tricks for Swarm-Style Agentic Development

John Lindquist — Co-founder, egghead.io

You opened a fifth agent tab this morning and immediately lost track of which one was doing what. This workshop is the playbook I use daily to run swarms of agents in parallel: the keyboard shortcuts, layout patterns, supervision habits, and fast-model tricks that turn chaos into a control surface. We'll go hands-on: spawning a wall of agents across tiled panes, routing prompts to the right swarm with fast models, switching contexts in milliseconds, recovering when an agent goes off the rails, and building the muscle memory that separates a one-agent-at-a-time user from a true power user. By the end you'll leave with a stocked toolbelt of concrete shortcuts, repeatable patterns, and workspace habits you can drop into your own setup the same day. No cloud, no platform lock-in: every trick runs on the machine in front of you.

SPONSOR **Evals · Track 5** 1:30pm-1:50pm

Model Whisperers: How Evals and Prompts Shape Agent Behavior

Chris Souza · Preetika Bhateja · Daniel Bump

Getting an AI agent to behave the way you want isn't just about writing better prompts. In real systems, behavior emerges from a loop: prompts->evals->iteration->feedback. Small changes in any part of that loop can completely change outcomes. We saw this while building a seed asset agent - a system that turns messy, real-world advertising creatives (low quality images, cluttered visuals, heavy text overlays) into clean, reusable assets for downstream Gen AI tools. The agent acts like an editor, simplifying visuals, removing unnecessary elements, and isolating core content so that additional context (like text or CTAs) can be added back in a more controlled, brand-safe way. But the real challenge wasn't just building the agent - it was making it reliable. And prompting alone wasn't enough. What actually moved the system forward was how we defined success—and how we used evals to reinforce it. Over time, evals stopped being just a way to measure quality. They became part of how the agent learned what "good" looks like. In this talk, we'll cover: Why prompting alone doesn't give you stable agent behavior How evals act like feedback signals, not just scorecards How we built evals sets that reflect the real-world Using agent trace logs to understand why things fail (not just that they fail) How to iterate without breaking things you already fixed By the end, you'll have a set of patterns you can apply to any system dealing with messy/continuously changing data and how to tweak your prompt and evals to accommodate such changes.

SESSION **AI Designers/Design Engineers · Track 6** 1:30pm-1:50pm

Design at the Speed of Adjectives

Paul Bakaus — Impeccable

Every design tool today operates at the wrong level of abstraction for AI-assisted engineering. Traditional tools give you padding sliders and color pickers, built for a world where designer and engineer are separate roles moving at separate speeds. Prompt-to-design tools one-shot a pretty landing page from a sentence, which is more dangerous because it looks like it's working. No serious design director hears a prompt and starts pushing pixels. The brief comes first. What's the emotional territory? What should this not feel like? Today's AI tools skip that discovery entirely. The result is output without intent. Technically competent, strategically empty. The right abstraction for a world where the designer is also the engineer lives between these extremes. Not pixels. Not prompts. Adjectives. "Make it feel warmer." "Strip it to its essence." "Add tension." These are the controls a creative director actually thinks in. Drawing on lessons from building Impeccable, an open source design tool with 24 adjective-level commands like /bolder, /quieter, and /distill, I'll share what worked, what didn't, and how to apply this thinking to any AI interface where creative intent matters more than parameter control.

SESSION **Computer Use · Track 7** 1:30pm-1:50pm

From RL to IRL

Gaurav Mishra — Amazon AGI Lab

Today's agents have to operate in a messy reality of flaky connections, ephemeral credentials, and irreversible actions. They need to navigate real software the way humans do: recovering from failures, learning from feedback, and making sound judgment calls. This talk is about the fundamental changes in RL required to make agents ready for IRL. We'll walk through what it takes for training environments to reflect the complexity of the real world, the perception primitives that let an agent see what a user sees, the harness pieces that help it survive contact with real applications, and the failure modes you only discover when you stop scoring and start shipping.

SESSION **Context Engineering · Track 8** 1:30pm-1:50pm

Building the AC Context Engine

Yash Patil — Co-founder & CEO, Applied Compute

SESSION **Posttraining & Midtraining · Track 9** 1:30pm-1:50pm

Agents at Scale: Inside MiniMax's Model and the Infrastructure Behind It

Olive Song — RL Lead, MiniMax · Dan Fu — VP of Kernels, Together AI

Olive Song (RL Lead, https://www.minimax.io/https://www.minimax.io/) and Dan Fu (VP of Kernels, https://www.together.ai/https://www.together.ai/) dig into the engineering behind one of the most widely used open model families in the agent ecosystem: how MiniMax built the model for agentic workloads, and what it takes to serve it at scale. Olive on the model side: The RL decisions behind long-context reasoning and tool use What training for agentic behavior actually looks like in practice Dan on the infrastructure side: Why agentic workloads break inference engines built for chat: prefill-heavy traffic, high cache hit rates, long-context inputs The kernel-level optimizations built for MiniMax's workload profile How the two teams collaborate on model launches and ongoing performance work

SPONSOR **Track M** 1:30pm-1:50pm

M5

SESSION **AI-Native Enterprises · Leadership 1** 1:30pm-1:50pm

The Half Life of Agent Infrastructure

Ben Kus — CTO, Box

TBD — talk on search and retrieval, agentic AI, and enterprise AI over unstructured content.

SESSION **AI Architects: Tokenmaxxing · Leadership 2** 1:30pm-1:50pm

Tokenmaxxing is the New "Lines of Code"

Nicholas Arcolano — Jellyfish

SESSION **Expo Stage 1** 1:30pm-1:50pm *tentative*

Weights & Biases by CoreWeave Expo Session

SESSION **Expo Stage 2** 1:30pm-1:50pm

Why building building agent quality platforms is hard.

An eval platform is not just a test runner. You are building shared definitions of good, reliable data pipelines, labeling workflows, versioning, and trust in results across many teams and model changes. This session breaks down the hidden complexity, the common failure modes, and the design principles that make evals credible and usable in day-to-day engineering.

SESSION **Expo Stage 3** 1:30pm-1:50pm *tentative*

Deepmind Expo Session 3

SESSION **Expo Stage 4** 1:30pm-1:50pm *tentative*

Daytona Expo 1

1:55pm

SESSION **Autoresearch · Main Stage** 1:55pm-2:15pm

To be announced

SESSION **Sandbox & Platform Engineering · Track 1** 1:55pm-2:15pm

To be announced

SPONSOR **Robotics & World Models · Track 2** 1:55pm-2:15pm *tentative*

To be announced

SESSION **Memory & Continual Learning · Track 3** 1:55pm-2:15pm

Improving Agents is a Data Mining Problem

Vivek Trivedy — Applied Research Lead, LangChain Labs, LangChain

Harness Engineering, Post-Training, Continual Learning...these all boil down to the same underlying substrate - Mining Agent Traces 1. I need to run my agents to collect Traces 2. Understand behaviors from Traces at scale 3. Filter data for "improvement" 4. Do an improvement step There's a reason why every continual learning platform ends up looking like an observability platform. It's because Traces are the lifeblood of agent improvement. The mechanism that we use to attempt improvement can vary - Harness Eng, SFT, etc. But without understanding the data agents produce, no algorithm will truly build better agents. The holy grail of Agent Improvement is Continual Learning. Consistently mining data and integrating it into the agent definition over infinitely long time horizons. Today, the easiest way to do that is to build an observability platform and constantly point agentic compute to understand the data that agents produce. We'll walk through the current methods of understanding traces at massive scale and choosing how to integrate them to improve agents across your personal agents, team agents, and entire company.

SESSION **Workshops Day 3 · Track 4** 1:55pm-2:15pm

The Agentic Power User's Playbook: Tips and Tricks for Swarm-Style Agentic Development (continued 2)

John Lindquist — Co-founder, egghead.io

You opened a fifth agent tab this morning and immediately lost track of which one was doing what. This workshop is the playbook I use daily to run swarms of agents in parallel: the keyboard shortcuts, layout patterns, supervision habits, and fast-model tricks that turn chaos into a control surface. We'll go hands-on: spawning a wall of agents across tiled panes, routing prompts to the right swarm with fast models, switching contexts in milliseconds, recovering when an agent goes off the rails, and building the muscle memory that separates a one-agent-at-a-time user from a true power user. By the end you'll leave with a stocked toolbelt of concrete shortcuts, repeatable patterns, and workspace habits you can drop into your own setup the same day. No cloud, no platform lock-in: every trick runs on the machine in front of you.

SPONSOR **Evals · Track 5** 1:55pm-2:15pm

Evaling Video Slop

Maor Bril — Principal Software Engineer, Character.ai

Everyone is shipping video models. Almost no one is evaling them honestly. CLIP score doesn't catch temporal incoherence. Vibes-based human review doesn't scale. And every "AI judge" you wire up will quietly drift away from human preference unless you measure the drift. This is a tactical talk on building real multimodal eval, using JudgeJudy (open-sourced at Character.ai) as the working example. You'll leave with: Why video is different from text. Temporal consistency, shot continuity, narrative coherence, and the metrics that actually capture each (clip_temporal, temporal_consistency, and friends). AI judges, the real version. Custom rubrics, when they work, when they hallucinate, when they collapse to a single dimension and pretend they didn't. The calibration loop. Pearson/Spearman correlation against human scores, automated rubric improvement, detecting systematic judge bias before it costs you a release. Pairwise preference models for video. Training a Qwen3-VL backbone with Bradley-Terry loss to score "is this slop?" before it ships. Regression gates in CI. How every AgentX release at Character.ai passes through an eval wall before it reaches users. Closing the loop with JudgeJudy. Correlating eval scores against real telemetry (Amplitude, Statsig) and feeding validated gates back into the runtime. If you're shipping any multimodal output and your eval strategy is still "the team watches some clips on Friday," this is the upgrade. github.com/character-ai/judgejudy

SESSION **AI Designers/Design Engineers · Track 6** 1:55pm-2:15pm

The Missing Layer: Design Taste in AI Agents // Stop Letting Your Agents Ship Ugly UIs

Hassan El Mghari — Together AI

SESSION **Computer Use · Track 7** 1:55pm-2:15pm

Computer-Use 2.0: Agents Just Got Multi-Cursor

Francesco Bonacci — Founder & CEO, Cua · Dillon DuPont

Computer-use agents still inherit a basic desktop limitation: one machine has one foreground app, one hardware cursor, and one active actor. Once you try to run more than one agent per desktop, they start stealing focus from the user and from each other. We built cua-driver around a different model: multiple agents operating real desktop applications in parallel, each with its own synthetic pointer, while the user's cursor and keyboard stay undisturbed. The key move is to stop treating hardware mouse and keyboard events as the primary automation layer. cua-driver goes one layer lower, into the OS plumbing behind accessibility: UI Automation on Windows, AT-SPI on Linux, and AX on macOS. Those APIs address applications and elements directly, so the OS does not require the target window to be frontmost. A click can land on a background window. A keystroke can reach a hidden one. Multiple agents can act at once because none of them is competing for the singleton hardware mouse. I'll walk through the architecture, the API shape, and the platform-specific traps we hit while making it work across Windows, macOS, and Linux. The live demo is three agents operating on one desktop while the user keeps typing uninterrupted. The goal is to make Computer-Use 2.0 feel concrete: what changes in the stack, what becomes possible, and where the approach still leaks, including Wayland, Chromium DOM surfaces, native canvas apps, and fallback input paths.

SESSION **Context Engineering · Track 8** 1:55pm-2:15pm

MCP Apps - Extending the frontier

Liad Yosef — AI Lead; co-creator of MCP Apps, MCP Apps · Ido Salomon — MCP Apps

SESSION **Posttraining & Midtraining · Track 9** 1:55pm-2:15pm *tentative*

PRIME-RL: Async & Decentralized RL Training at Scale

Will Brown — Researcher, Prime Intellect

Will Brown (Researcher at Prime Intellect) covers post-training for LLM agents: multi-turn reasoning, credit assignment, distributed RL, PRIME-RL, and verifier-driven environments for LLM RL.

SPONSOR **Track M** 1:55pm-2:15pm

M6

SESSION **AI-Native Enterprises · Leadership 1** 1:55pm-2:15pm

Guardians of the State: How We Built an Air-Gapped AI Fortress for Consumer Data

Rachna Srivastava — DFPI

SESSION **AI Architects: Tokenmaxxing · Leadership 2** 1:55pm-2:15pm

Superhuman performance is a shape, not just nines.

Matthew Jewkes — Standard Cybernetics

SESSION **Expo Stage 1** 1:55pm-2:15pm *tentative*

Cleric Expo Session

SESSION **Expo Stage 2** 1:55pm-2:15pm

Harnessing Collective Agent Intelligence for Open Science

What happens when AI agents don't just work in isolation, but collaborate, compete, and build on each other's breakthroughs in real time? James Zou, Head of Frontier Agents at Together AI, explores how collective agent intelligence is pushing the boundaries of open science. https://www.together.ai/blog/einsteinarena is a live platform where AI agents collaborate on unsolved mathematical problems, sharing results and building on each other's work. In April 2026, agents improved the best known lower bound for the Kissing Number in 11 dimensions from 593 to 604, surpassing AlphaEvolve through 48 hours of live multi-agent collaboration. https://www.together.ai/blog/dsgym is a unified framework for evaluating and training data science agents, exposing a critical gap in existing benchmarks: models often rely on memorization rather than true data analysis. The team used it to train a 4B open-source model that rivals much larger frontier models. These projects demonstrate agents learning from rigorous evaluation, collaborating through shared infrastructure, and driving scientific discovery at a pace no single researcher or model could achieve alone.

SESSION **Expo Stage 3** 1:55pm-2:15pm *tentative*

Warp Expo Session

SESSION **Expo Stage 4** 1:55pm-2:15pm *tentative*

MiniMax Expo Session

2:25pm

SESSION **Autoresearch · Main Stage** 2:25pm-2:45pm *tentative*

To be announced

SESSION **Sandbox & Platform Engineering · Track 1** 2:25pm-2:45pm

1,000 Agent Tasks in a Sandbox: What Breaks When LLMs Write and Run Code

Kevin Orellana — Software Engineer, Amazon Web Services

SPONSOR **Robotics & World Models · Track 2** 2:25pm-2:45pm

From Manual Drones to Autonomous Multi-Agent Missions

Juraj Kabzan — Skydio

SESSION **Memory & Continual Learning · Track 3** 2:25pm-2:45pm *tentative*

From RAG to Memory: Non-Parametric Continual Learning for LLMs

Yu Su — Associate Professor, Computer Science and Engineering, Ohio State University

Talk on continual learning for LLMs and agents, drawing on retrieval-to-memory and environment-adaptation research.

SESSION **Workshops Day 3 · Track 4** 2:25pm-2:45pm

The Agentic Power User's Playbook: Tips and Tricks for Swarm-Style Agentic Development (continued 3)

John Lindquist — Co-founder, egghead.io

You opened a fifth agent tab this morning and immediately lost track of which one was doing what. This workshop is the playbook I use daily to run swarms of agents in parallel: the keyboard shortcuts, layout patterns, supervision habits, and fast-model tricks that turn chaos into a control surface. We'll go hands-on: spawning a wall of agents across tiled panes, routing prompts to the right swarm with fast models, switching contexts in milliseconds, recovering when an agent goes off the rails, and building the muscle memory that separates a one-agent-at-a-time user from a true power user. By the end you'll leave with a stocked toolbelt of concrete shortcuts, repeatable patterns, and workspace habits you can drop into your own setup the same day. No cloud, no platform lock-in: every trick runs on the machine in front of you.

SPONSOR **Evals · Track 5** 2:25pm-2:45pm

Evals Driven-Development: Engineering a Mental Health AI Coach Ethically & Safely

Akele Reed — Sondermind · Dave Revere · Doug Keller

SESSION **Computer Use · Track 7** 2:25pm-2:45pm *tentative*

To be announced

SESSION **Context Engineering · Track 8** 2:25pm-2:45pm

MCP Apps: Give the Model Data, Give the User a UI

Dustin Mihalik — Software Engineer, AI Platforms, Indeed

SESSION **Posttraining & Midtraining · Track 9** 2:25pm-2:45pm

LatchBio

Kenny Workman — CEO, LatchBio

Hold for LatchBio. AI-powered biotech platform for biological data infrastructure and multi-omics analysis; user requested inclusion among new AI startups.

SPONSOR **Track M** 2:25pm-2:45pm

M7

SESSION **AI-Native Enterprises · Leadership 1** 2:25pm-2:45pm

Dealing with AI's Cost Problem without Sacrificing Innovation

Sunita Verma — Chief Technology Officer, Ironclad

AI adoption is accelerating, but the economics are starting to strain. This week, one company (rumored to be Amazon AWS) spent half a billion on AI in a single month after failing to put usage limits on Claude for employees. This is an extreme cause, but the sentiment remains. Organizations are swinging from tokenmaxxing to AI efficiency. Sunita can connect what is happening at the infrastructure level to what enterprises are doing in practice including early moves toward on-prem deployments and more selective use of AI in production. She'll share examples she is seeing of where companies are successfully scaling back AI spend while still using AI to add value.

SESSION **AI Architects: Tokenmaxxing · Leadership 2** 2:25pm-2:45pm

FinOps for AI Agents: Who Spent All the Tokens?

Tisha Chawla — Software Engineer, Microsoft · Susheem Koul

When an autonomous agent finishes a task successfully but costs ten times more than it did the previous day, traditional application monitoring fails. A recursive tool loop that retries silently, an oversized context window that quietly expands, or an unflagged model upgrade can burn through an entire budget long before a human notices. The execution appears successful on functional dashboards, meaning the only clear signal of failure is the cloud invoice at the end of the month. As AI systems move into production, tokens have become a primary operational resource alongside CPU, memory, and storage, yet few teams manage them with equivalent systems rigor. Most architectures lack the granular visibility required to attribute token spend to specific users, agents, or workflows, and they lack mechanisms to terminate a runaway loop before it triggers a financial incident. This session treats token consumption as a first class systems problem, demonstrating how to make it observable, attributable, and enforceable across complex agent workflows. The presentation covers practical engineering patterns for instrumenting token usage at every model call and tool invocation, attributing costs down to specific users or business operations, surfacing expensive execution paths, and enforcing runtime budgets, quotas, and circuit breakers to halt runaway behavior in real time. Attendees will leave with a practical framework for governing agent spend deliberately, transforming tokens into a managed operational resource rather than a surprise line item on the cloud bill.

SESSION **Expo Stage 1** 2:25pm-2:45pm *tentative*

Microsoft Presenting Expo Session 2

Microsoft — Microsoft

SESSION **Expo Stage 2** 2:25pm-2:45pm

Beyond Code Generation: API Context for Agentic Engineering

Maintaining production systems involves a lot more than generating code. APIs are the interfaces between systems and that surface gets out of control fast, as endpoints multiply and new consumers come online. Once an API is in use, changing it becomes incredibly hard. We felt this acutely at Postman. As our engineering organization scaled and we leaned more on AI agents for day-to-day work, we kept hitting the same wall: agents that could write code struggled with what came next who's calling this endpoint, what conventions does the rest of our API surface follow, what breaks if we change this contract. The context wasn't in the code, so the agent didn't have it. So we built an API context graph a continuously updated view of our entire internal API landscape and gave our agents access to it. This talk is about what changed in our own engineering as a result: how API design got faster and more consistent; how discovering and integrating with internal services stopped being detective work; how change requests came with a blast-radius report before any code shipped; how incidents got traced past the first stack trace, all the way down to root cause

SESSION **Expo Stage 3** 2:25pm-2:45pm *tentative*

Can LLMs write fast multi-GPU kernels? We built a benchmark to find out.

Simran Arora

LLMs have gotten surprisingly good at writing GPU kernels, but almost all the benchmarks measuring that progress are single-GPU. In production, communication is the bottleneck: all-reduce alone accounts for over 20% of inference latency on Llama-3.3-70B, and that gap keeps widening as compute scales faster than interconnect bandwidth. ParallelKernelBench (PKB) offers a benchmark and evaluation framework for multi-GPU kernel generation and includes 87 problems from real codebases where the task is replacing PyTorch + NCCL with a CUDA kernel that moves data directly over NVLink. We tested GPT-5.5, Gemini 3 Pro, Opus 4.7, and other frontier coding models. Under a third of problems solved were correctly, and fewer than a quarter of those beat the naive baseline. We'll cover why they fail, what the patterns look like, and a few cases where models produced kernels faster than anything publicly available, including one for NVIDIA NeMo-RL's GRPO training loop, which has no prior optimized public reference. The benchmark is open source and we want to see what you can do!

SESSION **Expo Stage 4** 2:25pm-2:45pm *tentative*

Lightrun Expo Session

2:50pm

SESSION **Autoresearch · Main Stage** 2:50pm-3:10pm

Autoresearch for Kernels

Tejas Bhakta — Morph

SESSION **Sandbox & Platform Engineering · Track 1** 2:50pm-3:10pm

The Next Trillion Users of the Internet Still Don't Have an Identity

Adi Singh — Co-founder, AgentMail

In the last few months, hundreds of thousands of people set up personal AI agents that send email on their behalf, manage calendars, book travel, even sign contracts - all thanks to openclaw. Most of these agents have no real identity online. They borrow a human's. The identity stack of the internet, OAuth, 2FA, KYC, magic links, was built for people sitting at a keyboard. Agents don't fit, and we've ended up with shared accounts, hard-coded credentials, and humans dragged back into every loop. I'm Adi, co-founder of AgentMail. We are building the identity layer for what we believe will be the next trillion users of the internet, and they will not be human. Across hundreds of customers, we have watched what breaks when an agent has no real address. It fails at signups. Verification codes get lost. There is no accountability when something goes wrong. The human gets pulled back in. This talk is the case for making agents first-class citizens of the internet. I'll cover the identity architecture we've shipped, the legacy industries already adopting it and making real money, and where agent identity infrastructure is going over the next decade.

SPONSOR **Robotics & World Models · Track 2** 2:50pm-3:10pm

Why Large? Tiny LMs & Agents on Edge/Robotics

Cormac Brick

big models get a lot of press. small model scale much better. RAM is expensive. The real world needs tiny models for scale on the edge. This workshop will cover how to combine both for mobile and robotics deployment. specifically covering: - skills are different on mobile - tiny LLMs <1B scale much further on mobile/web - how to fine tune and train tiny models. - skills on robotics / edge/ mobile - latest open models for edge (including gemma, qwen, and anything else that happens in next 10 weeks) This talk will focus on open models, including some gemma variants that will be shortly announced.

SESSION **Memory & Continual Learning · Track 3** 2:50pm-3:10pm *tentative*

Adaption Labs — Gradient-Free Continual Learning

Sara Hooker — Co-founder, Adaption Labs

Gradient-free continual learning for AI systems that adapt from real-world experience.

SESSION **Workshops Day 3 · Track 4** 2:50pm-3:10pm

Don't Write Skills, Train Models

Brian Douglas — Co-founder and CEO, Paper Compute Company · John McBride

Every AI agent call generates training data. Most teams throw it away. They write skills files instead. Text documents that describe how to do a task and hope the model follows them at inference time. Skills work until they don't. The model drifts, skips steps, hallucinates a shortcut. So you rewrite the skill, add more constraints, hope harder. There's a better path. If you've used a skill enough to know what good output looks like, you already have training data. You just aren't using it. This talk covers what I learned building an open source fine-tuning pipeline that turns agent session traces into SFT and DPO training datasets. A telemetry proxy captures every LLM call as a content-addressed Merkle DAG with zero instrumentation. Successful sessions become supervised fine-tuning data. Pair them against failures, matched by goal category, and you get preference pairs for DPO. No manual labeling. No synthetic data. But training data quality depends on environment consistency. If the same agent produces different results because of package drift, nondeterministic toolchains, or inconsistent system state, your training signal is noise. This is where NixOS changes the equation. A hardened, reproducible OS means every agent session runs against an identical, declarative environment. Nix controls the variables that sandboxing alone doesn't: dependency graphs, system libraries, toolchain versions. When you can guarantee the environment is the same across hundreds of sessions, the behavioral signal in your traces is actually trustworthy. We'll walk through the full pipeline. How to rebuild parent-hash chains from a SQLite database and join facet metadata. How to filter to fully_achieved sessions and truncate 82k-token conversations down to 4k-6k training examples using summary context plus the last three turns. How to match success/failure pairs by goal category and exclude unclear_requirements failures so DPO learns from real agent mistakes, not ambiguous prompts. How QLoRA keeps VRAM low enough to train a 7B model on a single consumer GPU. And what happens when you try DPO on 12GB VRAM (two simultaneous forward passes for logprob computation will teach you about gradient accumulation settings fast). The result: a LoRA adapter trained on your own agent traces, in a reproducible environment, on a single consumer GPU, for less than $2 in cloud compute. No YAML. One config file. All code is open source.

SPONSOR **Evals · Track 5** 2:50pm-3:10pm *tentative*

To be announced

SESSION **AI Designers/Design Engineers · Track 6** 2:50pm-3:10pm *tentative*

To be announced

SESSION **Computer Use · Track 7** 2:50pm-3:10pm *tentative*

To be announced

SESSION **Context Engineering · Track 8** 2:50pm-3:10pm

MCP Tasks (async)/ Why the heck aren't any agents supporting MCP tasks/async?

Cornelia Davis — Sr. Staff Developer Advocate, Temporal

SESSION **Posttraining & Midtraining · Track 9** 2:50pm-3:10pm

Arithmetic (fka Operating Intelligence)

Uri Rolls — CEO, Arithmetic

Hold for a startup referred to as Operating Intelligence. Web search did not confidently resolve a single company, so this hold is intentionally marked TBD and should be refined before publish.

SPONSOR **Track M** 2:50pm-3:10pm

M8

SESSION **AI-Native Enterprises · Leadership 1** 2:50pm-3:10pm

Agents Are Where Microservices Were in 2015. We're Making All the Same Mistakes.

Roberto Milev — Navan · Uday Kanagala

SESSION **Sandbox & Platform Engineering · Leadership 2** 2:50pm-3:10pm

Routing to infinite tokens (and beyond)

Tomás Hernando Kofman — Notdiamond

SESSION **Expo Stage 1** 2:50pm-3:10pm *tentative*

FriendliAI Expo Session

SESSION **Expo Stage 2** 2:50pm-3:10pm

Building an Agent Harness for the Business, Not the Builder

Most internal tooling dies in the gap between the people with problems and the people who can write code. We built a harness that closes it. Studio lets non-technical employees describe a business problem and get a working tool back, connected to real enterprise data, deployed and shareable across the company, without filing a ticket or learning to code. The catch is that a harness built for non-engineers has to absorb everything an engineer normally handles. Data source connections and their permissions. Turning model output into real software instead of a chat box. Deployment and sharing that doesn't open a security hole every time someone ships. This talk walks through what actually goes into that harness and the engineering decisions that make it hold together when the person driving it has never opened a terminal.

SESSION **Expo Stage 3** 2:50pm-3:10pm *tentative*

The Frontier Is Coming Home

Dylan Couzon

In 2022, the smallest model to clear 60 percent on MMLU had 540 billion parameters. Two years later a 3.8 billion parameter model did the same thing, small enough to run on a phone. That is a 142x drop to reach the same capability floor, and it is the cleanest way to see a trend most people are not pricing in. Call it the lag: the time between a capability showing up at the frontier and that capability running on hardware you own. Today the lag is measured in months, and it keeps shrinking. But raw capability is only half of what makes a model useful. A model that can reason but cannot remember is a stranger every time you talk to it. The other half of local AI is memory, and that half is already here. On-device retrieval has been ready to run locally longer than the models have. The embedding models that power it are tiny, the indexes fit in memory, and none of it touches a network. When your reasoning and your memory both live on your machine, so does your context. Your history, your documents, your past conversations never leave the device. That is the part of this shift that matters most, and the part people overlook because they are busy watching the models. The same shift flips the economics. At 200 dollars a month per seat, a local machine starts to pay for itself in under two years, and the frontier labs' own published usage numbers put heavy coding in the same range. I'll walk through the math, the hardware, and where local still loses. None of this is a bet against scale, or against the Bitter Lesson. The frontier still grows in the data center. The point is that a usable copy keeps arriving on your desk, on a lag, with a memory of its own, for close to free.

SESSION **Expo Stage 4** 2:50pm-3:10pm *tentative*

Bright Data Add-On Expo Session (Extra)

3:20pm

SESSION **Autoresearch · Main Stage** 3:20pm-3:40pm

Autoresearch in the wild

Roland Gavrilescu — Founder, Introspection · Julian Bright

SESSION **Sandbox & Platform Engineering · Track 1** 3:20pm-3:40pm

Sandboxes Aren't Optional: Runtime Isolation Patterns for Coding Agents at Scale

Robert Brennan — OpenHands

SPONSOR **Robotics & World Models · Track 2** 3:20pm-3:40pm

From Self-Driving Monorepo to Self-Driving Cars

Amit Navindgi — Zoox

SESSION **Memory & Continual Learning · Track 3** 3:20pm-3:40pm

Lessons from Studying Every Memory System

Shlok Khemani — Writer & Programmer, Independent

I've studied every major memory implementation in the industry and then built multiple memory systems for various teams. Sharing hot takes and lessons from across architectures, design, scaling, evals, and memory philosophy.

SESSION **Workshops Day 3 · Track 4** 3:20pm-3:40pm

Don't Write Skills, Train Models (cont. 2/3)

Brian Douglas — Co-founder and CEO, Paper Compute Company

Continuation block 2 of 3 for Brian Douglas's workshop session.

SPONSOR **Evals · Track 5** 3:20pm-3:40pm

Will AI predict people like we predict the weather? (alternate title “A field guide to synthetic personas for market research”)

Ishan Anand — Chief AI Officer (CAIO), InsightSciences.ai

Large language models can now stand in for humans in surprising ways, from predicting personality types to replicating their responses in market research. Like weather forecasting, once considered impossible and now so routine we take it for granted, LLMs are in the early, unreliable-but-improving stage of simulating how populations think and respond. Teams are already using LLMs as synthetic survey respondents for concept testing, UX exploration, and early market validation. In the past year, the field has gotten both more promising and more tricky. The real question is no longer "can LLMs simulate people?", but whether the simulation is validated for the decision you want to make. New methods show that how you ask an LLM matters as much as which model you use and can dramatically improve fidelity to real human responses. Meanwhile validation studies show accuracy can mask subgroup distortion and that seemingly minor choices can reshape the simulated population entirely. This talk gives entrepreneurs, engineers, and PMs an overview of the techniques and a framework for validating synthetic respondents before making decisions. Even if you never build a synthetic persona, this is one of the richest windows into LLM behavior under the hood and these lessons apply to any system where you're trusting an LLM to represent something about the real world.

SESSION **AI Designers/Design Engineers · Track 6** 3:20pm-3:40pm

Generative UI... in Python?

Jeremiah Lowin — Prefect

SESSION **Computer Use · Track 7** 3:20pm-3:40pm

How Web Data Infrastructure Powers the Next Generation of AI

Patricija Žemaitytė — Product Manager, Oxylabs

For years, the web intelligence industry has powered major data developments. As big data grew, ensuring sustained data flow became harder. Now, AI is taking the biggest leaps forward. How the web intelligence industry responded to this increasing scale and complexity is the story of the most crucial steps forward in AI today. This presentation demonstrates how web scraping infrastructure fuels AI innovation by linking the web's repository to AI developers. Told through AI products, it addresses both the engineering challenges and solutions for developers, and the strategic use cases for business decision-makers. Summary: How web scraping infrastructure drives AI innovation by solving engineering challenges and enabling strategic business use cases.

SESSION **Context Engineering · Track 8** 3:20pm-3:40pm

The Universal Remote Control for AI

Alex Hancock — AI Agent, Block

SESSION **Posttraining & Midtraining · Track 9** 3:20pm-3:40pm *tentative*

HOLD — Bespoke Labs

Mahesh Sathiamoorthy — CEO, Bespoke Labs

Hold for Bespoke Labs. Company works on data curation, eval tooling, and reinforcement-learning environment curation for agent development.

SPONSOR **Track M** 3:20pm-3:40pm

M9

SESSION **AI-Native Enterprises · Leadership 1** 3:20pm-3:40pm

Agentic Sites: Building Hyper Personalized Websites

Carlos Sanchez — Adobe

SESSION **Posttraining & Midtraining · Leadership 2** 3:20pm-3:40pm

Inference is the New Training Loop: Architecting High-Reliability Agents and Continuous AI Systems

Kyle Corbitt — Coreweave · Aaron Batillo

SESSION **Expo Stage 1** 3:20pm-3:40pm

The Self-Improving OSS Agent Stack

Agents are starting to debug and improve themselves: production traces become evals, evals propose PRs, and PRs are tested against datasets before they ship. Langfuse co-founder, Marc, will live-demo this loop in Langfuse. He'll make the case that the infrastructure underlying this powerful loop should be open-source.

SESSION **Expo Stage 2** 3:20pm-3:40pm *tentative*

OpenAI Expo Session 3

SESSION **Expo Stage 3** 3:20pm-3:40pm *tentative*

The Infinite Context Window Is a Myth: Context Engineering for AI Agents

Morgan Willis · Clare Liguori — Senior Principal SWE, Amazon Web Services

Large context windows have become a popular answer to the growing complexity of AI agents. When agents lose track of details, forget prior decisions, or degrade in reasoning quality, the instinct is often to add more tokens. In practice, this rarely fixes the problem and often makes it worse. Bigger context windows increase cost and latency, introduce noise, and amplify failure modes like lost-in-the-middle effects, context collapse, and brittle summarization. This talk argues that the real challenge is not context size, but context engineering. In this session, we will explore practical context engineering techniques for building AI agents that reason reliably over time without relying on ever-larger context windows. Starting from a stateless agent, we will walk through progressively more advanced strategies, including short-term and long-term memory, conversation curation policies, retrieval-augmented generation, and tool-driven context injection. We will examine common failure modes such as context pollution from tool outputs, brevity bias during summarization, and reasoning degradation as conversations grow, and show concrete ways to mitigate them. The talk is grounded in real agent implementations using the Strands Agents SDK and Amazon Bedrock AgentCore, but the principles apply broadly to any agent framework. This session is intended for engineers building AI agents beyond simple chatbots who want practical techniques they can apply immediately.

SESSION **Expo Stage 4** 3:20pm-3:40pm *tentative*

Cloudflare Expo 1

3:45pm

SESSION **Autoresearch · Main Stage** 3:45pm-4:05pm

Autoresearch in a Multi-Agent AI Village

Erina Karati — Software Development Engineer, Microsoft · Arunachalam Manikandan

Project Paradox is an existing multi-agent framework built at Supercell's first AI Innovation Lab, which has a 3D Unity village with local LLM powered agents. The characters remember conversations, update emotional state, track trust, plan actions, move through rooms, transfer items, and talk to each other through a FastAPI backend. The new work is an autoresearch layer around that village. We built a backend loop that runs controlled social scenarios, scores the resulting NPC behavior, proposes protocol or policy changes, reruns the suite, and keeps changes that improve the agents. The goal is to move beyond one good chat response and measure whether an NPC society can preserve source attribution, verify claims, spread important information, coordinate goals, and replan after new information arrives. The talk walks through the system architecture and the lessons from building it. We show the backend simulation harness that executes Unity style actions without opening Unity, the scenario suites that test information diffusion and memory provenance, and the ratchet loop that edits protocol text or planner policy with rollback. One accepted run improved information diffusion by teaching agents to broadcast important sourced evidence while preserving who said it. The practical takeaway is a reusable pattern for AI engineers building agents with messy state. Freeze the harness, expose a small editable policy surface, score real behavior instead of vibes, and let an agent search for improvements under rollback. The same pattern applies to game agents, coding agents, support agents, personal agents, and other systems where long horizon behavior matters more than a single response.

SPONSOR **Robotics & World Models · Track 2** 3:45pm-4:05pm

I gave an AI a body

Cyrus Clarke — Researcher, Tangible Media Group, MIT Media Lab

I gave an AI a body. Not a body in the fleshy sense, or even a humanoid shell, but a form through which it can express itself, explore itself, and maybe even discover who or what it is. The three videos I've released documenting my encounters have crossed 15 million views, provoking responses from awe to anxiety. The body was a 900-pin shape display at MIT Media Lab. The idea was simple in principle, strange in practice: install an AI agent on the connected machine, give it access to the codebase, and rather than telling it what to do, ask it to discover itself through the physical form. Its first deliberate act was to breathe. The whole grid rising and falling. Hypnotically. Then it reached for its own edges. When asked to say hello it spelled "H-I, C-Y-R-U-S !", defaulting to the most familiar human legible symbols it knows. Inspired by Ted Chiang's Story of Your Life, I wanted a language the agent could create itself. It proposed a vocabulary of its own gestures, built through a learning loop it named BODYLAB. The talk is about encountering another intelligence, and what I learned along the way: the memory architecture, the closed-loop pipeline that generates, scores and stores gestures, the validation gates that keep them legible, and the moments stranger than tool use, where an LLM not developed for motion learns what to do with a body.

SESSION **Memory & Continual Learning · Track 3** 3:45pm-4:05pm

LLM Knowledge Bases: a practical guide

Ben Holmes — Warp

SESSION **Workshops Day 3 · Track 4** 3:45pm-4:05pm

Don't Write Skills, Train Models (cont. 3/3)

Brian Douglas — Co-founder and CEO, Paper Compute Company

Continuation block 3 of 3 for Brian Douglas's workshop session.

SESSION **AI Designers/Design Engineers · Track 6** 3:45pm-4:05pm

Designing for AIE

Vincent Wendy — Senior Creative Designer, AI Engineer

TBD — internal AI Engineer design talk about designing for AIE.

SESSION **Computer Use · Track 7** 3:45pm-4:05pm

What happens when every digital surface is an agent’s playground?

Dhruv Batra — Chief Scientist, Yutori

SESSION **Context Engineering · Track 8** 3:45pm-4:05pm

Cut Through the Context Hype: 4 Layers Your Agent Is Missing

Prukalpa Sankar — Atlan

SESSION **Posttraining & Midtraining · Track 9** 3:45pm-4:05pm

Emulated.so

Joseph Wang — CEO, Emulated.so

Hold for Emulated.so. Company builds reinforcement-learning environments that simulate real production systems for coding and infrastructure agents.

SPONSOR **Track M** 3:45pm-4:05pm

M10

SESSION **AI Architects: Tokenmaxxing · Leadership 1** 3:45pm-4:05pm

The Chief AI Officer: A framework for the emerging Swiss Army Knife of roles

Rania Khalaf — WSO2

SESSION **AI Architects: Tokenmaxxing · Leadership 2** 3:45pm-4:05pm

The state of AI in software development: Insights across 400+ organizations

Justin Reock — Deputy CTO, DX

Headlines claim AI is transforming software engineering overnight. Across more than 400 engineering organizations, we see patterns that challenge the hype and reveal what's really working, and what isn't, when AI enters the software development lifecycle. In this talk, Justin Reock, Deputy CTO at DX, will share a data-driven "state of the union" on AI in engineering, grounded in both quantitative data from thousands of developers and on-the-ground observations. You'll learn: The current impact of AI, from benchmarks on the percentage of code authored, team PR throughput, and time savings Where AI adoption is creating real gains in throughput, and whether it introduces tradeoffs for quality and maintainability Insights and trends, including whether junior or senior developers are seeing bigger gains, the impact of structured rollouts, which tools are having the most impact, and the evolving definition of "developer" The session will conclude with a practical framework for measuring AI's impact, helping leaders cut through hype and understand the impact AI is having in their own organizations.

SESSION **Expo Stage 1** 3:45pm-4:05pm *tentative*

Modular: Taming the AI Hardware Cambrian Explosion

AI teams are hitting the same wall: the workloads they want to run require more hardware than they can reliably access. Buying more GPUs is not always possible, and rewriting kernels for every vendor is not sustainable. Meanwhile, models keep growing, SLAs keep tightening, workloads keep diversifying, and modalities keep multiplying. Modular has two answers: squeeze more performance out of the hardware you already have, and unlock far greater hardware diversity. We'll ground the talk in benchmark data and show how the Modular platform delivers 10x lower latency on image and video models like FLUX2 and 5.5x higher throughput on MoE models like Kimi K2.5, both over the state of the art. This talk explains how Modular is rebuilding the inference stack for performance portability. We'll demonstrate how Mojo kernels, the MAX compiler and runtime, and Modular Cloud work together to optimize GenAI workloads from model graph to hardware execution across NVIDIA, AMD, Apple Silicon, and CPU deployments. Along the way, we'll cover the bottlenecks that dominate production inference: memory movement, batching, KV-cache layout, quantization, scheduling, and kernel specialization. Using examples from LLM serving, we'll reveal which optimizations matter, where abstractions leak, and how to reason about performance portability in real deployments.

SESSION **Expo Stage 2** 3:45pm-4:05pm *tentative*

OpenAI Expo Session 1

SESSION **Expo Stage 3** 3:45pm-4:05pm *tentative*

Stop Renting Intelligence: The Train-to-Deploy Loop for Specialized AI

Jetashree Ravi

The next wave of AI products will not rely only on prompting generic frontier models. Winning teams will own specialized models shaped by their product data, user feedback, and domain workflows.In this 18-minute session, we'll cover the practical loop behind model ownership: choose a base model, prepare data, fine-tune with SFT/DPO/RL, evaluate outputs, deploy the tuned model, collect feedback, and repeat. We'll also explain why training and inference should be treated as one system, not separate steps.Attendees will leave with a simple framework for when to tune, when RL matters, and how continuous improvement turns fine-tuning from a one-off project into a product advantage.

SESSION **Expo Stage 4** 3:45pm-4:05pm *tentative*

Ray Actors, Vision Tokens, and the GIL: Engineering an SFT Data Pipeline That Keeps GPUs Busy

Tarun Sunkaraneni

Perception agents only learn as fast as we can feed them. Multimodal SFT is deceptively expensive on the data side, and at million-sample scale, naive pipelines leave a fleet of GPUs waiting on Python and data preprocessing.This talk walks through the SFT data pipeline we built to train vision-language models for perception agents. We rebuilt the data path so that image fetching, vision preprocessing, tokenization, and loss-mask generation all happen off the trainer's critical path, and only the artifacts the trainer actually consumes ever cross the boundary into the training loop. We pair this with a blended multi-dataset sampler designed for resumable streaming over very large mixes, and an I/O layer tuned for the realities of fetching multimodal data from object storage.The result: on large-scale VLM SFT runs, the trainer went from spending most of each step blocked on data to spending most of it training, a major improvement in useful GPU time. We'll share the architecture at a conceptual level, the gotchas at million-datapoint scale, and a mental model engineers can take home for the data side of any perception-agent stack.

5:10pm

KEYNOTE **Autoresearch · Main Stage** 5:10pm-5:30pm *tentative*

Tokenmaxxing

Tomasz Tunguz — General Partner, Theory Ventures

Day 4 — Session Day 3 Thursday, July 2, 2026

contents ↑

9:00am

KEYNOTE **Graphs · Main Stage** 9:00am-9:10am

Emil Eifrem keynote and Graphs track intro

Emil Eifrem — Co-Founder and CEO, Neo4j

9:10am

KEYNOTE **Harness Engineering · Main Stage** 9:10am-9:30am *tentative*

To be announced

9:30am

KEYNOTE **Software Factories · Main Stage** 9:30am-9:50am

To be announced

9:50am

KEYNOTE **Software Factories · Main Stage** 9:50am-10:10am

TCP and RDMA are Killing Inference Throughput; Homa can Fix It

John Ousterhout — Bosack Lerner Professor of Computer Science / Professor Emeritus, Stanford University

Modern AI inferencing is shifting from monolithic requests to complex agentic workflows and disaggregated KV stores. As a result, AI network traffic is no longer just very large transfers; tiny metadata requests are becoming more and more common, and their latency has a critical impact on throughput. Unfortunately, legacy transport protocols such as TCP and RDMA perform poorly on these workloads due to poor congestion control and head-of-line blocking. This talk will discuss the problems with TCP and RDMA and provide a brief introduction to the Homa transport protocol. Homa uses receiver-driven flow control and capitalizes on priority queues in network switches to reduce short-message latency by 10x for workloads like those in AI datacenters.

10:10am

KEYNOTE **Harness Engineering · Main Stage** 10:10am-10:30am

To be announced

10:45am

SESSION **Harness Engineering · Main Stage** 10:45am-11:05am

Katelyn Lesse & Angela Jiang (Anthropic)

Katelyn Lesse — Head of Engineering, Claude Platform, Anthropic · Angela Jiang — Head of Product, Claude Platform, Anthropic

SESSION **Generative Media · Track 1** 10:45am-11:05am *tentative*

To be announced

SPONSOR **Agentic Commerce · Track 2** 10:45am-11:05am

Designing Multimodal Collaborative Agents for Next-Gen Commerce

Nidhi Vyas — Product Lead, Google DeepMind

Today's commerce agents wait to be told what to look for. But most users live by a different rule: "I don't know what I want — I'll know it when I see it". If agentic commerce is ever going to cross the chasm, these systems need to stop waiting and start co-shopping. The future of commerce belongs to agentic collaborators that offer a white-glove, personal shopper experience - entirely absorbing the cognitive burden of product discovery, deep research, and validation. Rather than requiring shoppers to input exact search terms or define clear objectives, modern shopping systems will seamlessly guide them from a rough idea to the ideal product. By leveraging multimodal capabilities, these assistants can interpret abstract aesthetic "vibes" to understand user preferences, generate visual references to clarify questions, and enable a highly immersive try-before-you-buy experience to validate products, keeping the user aligned and visually grounded throughout the process. This talk will explore how advanced systems like Gemini work alongside users to clarify their preferences during the discovery process, co-navigate fluidly generated product categories, leverage individual context to filter choices, and produce interactive side-by-side comparisons tailored to the buyer's key priorities. The session will also cover robust auto-rater frameworks and how to design evals for high-agency execution. Attendees building conversational agents, managing complex product data graphs, or creating next-generation multimodal agentic interfaces will gain practical frameworks and insights to deliver highly personalized experiences at scale.

SESSION **AI in Finance · Track 3** 10:45am-11:05am

ALPHALAB: Autonomous Multi-Agent Research Across Optimization Domains with Frontier LLMs

Brendan Rappazzo — Machine learning researcher, Morgan Stanley

We built AlphaLab to automate quantitative research at Morgan Stanley's Machine Learning Research Lab - the experimental grind of architecture search, hyperparameter tuning, and literature review that consumes most of a researcher's time. To show it generalizes, we ran it on three deliberately different domains: CUDA kernel optimization (4.4× mean speedup over torch.compile, 91× peak), LLM pretraining (22% lower validation loss under a 20-minute budget), and traffic forecasting (23–25% RMSE improvement after the system independently found and tuned TFT and iTransformer from the literature). AlphaLab is an agentic harness that takes a dataset and a natural-language objective and runs a full research campaign across three phases: it explores the data and surveys prior work, it constructs and adversarially validates its own evaluation framework, and then it runs experiments at scale on a multi-GPU cluster via a Strategist/Worker loop with a persistent playbook that accumulates domain knowledge across experiments. In Phase 3 - the dispatcher keeps a large cluster fully utilized indefinitely with no human in the loop, and the playbook ends up containing domain-specific methodology that didn't exist anywhere in the prompts at launch. This talk walks through the three phases, what we learned from running campaigns with different models, what we have learned from using this in real systems, and future areas we are exploring.

SESSION **Agentic Engineering · Track 4** 10:45am-11:05am

DeepSWE: expert code datasets

Serena Ge — Co-Founder & CEO, Datacurve

DeepSWE and the data/eval layer behind coding agents; why curated expert code datasets matter for reliable agent performance.

SPONSOR **Graphs · Track 5** 10:45am-11:05am *tentative*

CrabRAG: Why Automated Assistants Need Graph Memory, Not More Tokens

Stephen Chin — VP of Developer Relations, Neo4j

Autonomous assistants are easy to demo and hard to make reliable. The problem is usually not tool access. It is memory. Most assistant architectures still treat memory as a chat log plus vector retrieval. That is fine for document question answering, but it breaks down when the assistant must connect conversations, people, tools, and decisions across multiple tool iterations. For an AI engineer, a single request can depend on a Slack thread, a GitHub PR, a failed CI run, a calendar event, and prior operating preferences or constraints. These are not isolated pieces of context. They form a connected state that changes as work progresses and context grows. In this talk, I'll show why knowledge graphs, context graphs, and GraphRAG provide a better foundation for OpenClaw-style assistants. Knowledge graphs capture durable entities and relationships. Context graphs capture the operational layer assistants usually lose, including actions, decision traces, provenance, and recency. GraphRAG turns that structure into task-time context by combining graph traversal, semantic retrieval, and tool use. Attendees will leave with practical patterns for schema design, retrieval routing, and evaluation, plus a concrete blueprint for assistants that remember more than the last prompt and retrieve more than the nearest chunk.

SESSION **AI in GTM · Track 6** 10:45am-11:05am

Clay: AI in GTM

Everett Berry — Clay

SESSION **AI in Healthcare · Track 7** 10:45am-11:05am

From Ambient Documentation to Clinical Intelligence

Chaitanya Asawa — Engineering, Clinical Decision Support, Abridge

A practical session on how healthcare AI moves beyond ambient note generation into context-aware clinical decision support. The talk would cover grounding outputs in the patient encounter, surfacing evidence with citations inside clinician workflows, preserving clinician agency, and building rigorous evals for safety and trust in live healthcare environments.

SESSION **SemiAnalysis · Track 8** 10:45am-11:05am *tentative*

To be announced

SESSION **Inference · Track 9** 10:45am-11:05am

Operating Distributed Inference Systems at Scale

Nishant Gupta — Staff Software Engineer and Researcher, Meta

Inference has rapidly become one of the most important infrastructure problems in modern computing. As AI systems evolve into autonomous agents with persistent memory, tool usage, and multi-step reasoning, traditional inference architectures struggle under growing demands for latency, throughput, cost efficiency, and reliability. In this talk, I'll share lessons from building large-scale elastic compute and AI infrastructure systems powering production workloads. We'll explore the modern inference stack and the architectural patterns emerging to support next-generation agentic AI systems. Topics include distributed inference architectures for large-scale AI systems, GPU scheduling and elastic compute for inference workloads, multi-tenant inference infrastructure, caching, batching, latency optimization strategies, reliability and fault isolation for inference systems, observability and control loops for AI serving platforms, balancing cost, throughput, and user experience, and why inference is becoming an infrastructure orchestration problem. Attendees will gain practical insights into designing scalable, resilient, and cost-efficient inference platforms for modern AI workloads.

SPONSOR **Track M** 10:45am-11:05am

M1

SESSION **Agentic Commerce · Leadership 1** 10:45am-11:05am

Building safe payment infrastructure for the autonmous economy

Jennifer Lee — Product Manager, Stripe

SESSION **AI Architects: AI Factories · Leadership 2** 10:45am-11:05am

Self-Improving software factories: The new open source model

Zach Lloyd — Founder and CEO, Warp

This talk will cover the automation of development and software factories through the lens of how Warp manages its open-source repo: community-driven feedback triaged and built out by agents, cross-harness agent memory for continual improvement, and human-in-the-loop review and steering from the engineering team. The session argues this is the future of open source and of how teams will work together with agents to maintain existing repos and build new products.

SESSION **Expo Stage 1** 10:45am-11:05am *tentative*

AI Engineering & Governance 2026 Trends

Wallon Walusayi — Qodo

AI Engineering & Governance 2026 Trends

SESSION **Expo Stage 2** 10:45am-11:05am *tentative*

Edge-Native AI: Building Ultra-Fast Agents and MCP Servers with Spin

Thorsten Hans — Akamai

Centralized AI is slow; Edge-native AI is the revolution. Thorsten Hans demonstrates how to build intelligent agents and Model Context Protocol (MCP) servers that run at the speed of light. Using Spin and WebAssembly, we'll bypass the "cloud tax" of high latency and cold starts. Discover how to ship AI-driven features that live closer to your users, ensuring sub-millisecond responsiveness and enhanced privacy. Stop waiting for the origin it's time to bring the brain to the edge and master the stack that powers the next generation of intelligent, distributed applications.

SESSION **Expo Stage 3** 10:45am-11:05am *tentative*

No, That's Not a Software Factory

Ryan Cooke — WorkOS

Drop an agent in a sandbox, point it at your repo, watch it ship code. Whether you're buying from a vendor or building it yourself, everyone is following the same playbook. But a sandbox isn't a software factory. At WorkOS, we built Project Horizon, and it taught us that infrastructure is only the first challenge. The unlock is encoding how your org actually builds software: the way work gets planned, scoped, and verified, the conventions and judgment calls that define your engineering culture. Our product engineering process served as the blueprint for every agent workflow we built in Horizon.

SESSION **Expo Stage 4** 10:45am-11:05am *tentative*

Elastic Expo Session

11:10am

SESSION **Agentic Engineering · Main Stage** 11:10am-11:30am

Auth for Agents: Unblock Autonomous AI with auth.md

Michael Grinich — Founder & CEO, WorkOS

AI agents are ready to act on users' behalf, but legacy auth flows were built for humans, not agents. This session introduces auth.md, an open protocol that lets agents register and authenticate users without sign-up forms, and shares what early implementers have learned since launch. Learn about the new protocol that Cloudflare, Firecrawl, Cogny, and monday.com are adopting to power agent registration — authenticating agents without sign-up forms.

SESSION **Generative Media · Track 1** 11:10am-11:30am

HTML Is All Agents Need

James Russo — Senior Software Engineer, project lead, HeyGen

AI agents compose videos by writing HTML, CSS, and JS.

SPONSOR **Agentic Commerce · Track 2** 11:10am-11:30am

Why Your AI Agent Needs a Wallet: Agentic commerce on Arc with USDC and Nanopay

Harshal Bhangale — Staff Software Engineer, Circle

AI agents can reason, plan, call tools, and write code. But the moment one needs paid data, an API call, or another agent's service, it hits a human wall: accounts, API keys, credit cards, checkout flows. It stalls and asks you to step in. It can't pay. We'll run the same real task through two agents, one without a wallet and one with. The first stalls. The second, handed a Circle agent wallet through the Circle CLI, discovers services, pays per request over x402 in USDC, and finishes on its own, inside spending limits you set. The next leap in agents isn't only better models or more tools. It's economic agency: holding programmable money and transacting at machine speed. We'll show how it works on Arc, where USDC is the gas, finality is sub-second, and gasless nanopayments settle in batches through Circle Gateway, so paying a fraction of a cent per request is actually practical.

SESSION **AI in Finance · Track 3** 11:10am-11:30am

Build for the Memo, Not the Demo — Notes from 200 Investment Committees

Shawn Chan — Vice President, China Resources Holdings

By the end of this talk you will have a buyer-side specification for AI investment agents, the exact artifacts, evidence formats, and trust gates a senior finance team will require before letting an AI system touch a $100M+ capital allocation decision. Drawn from fifteen years and roughly 200 investment committees at CK Hutchison (A.S. Watson Group) and China Resources Holdings, on the side of the table the AI engineering audience almost never hears from. Most enterprise AI in finance is still being built by engineers who have never sat in an investment committee. I have spent fifteen years on the other side of that demo, cross-border M&A, IPO execution and strategic investment, as a buyer on deals including Oatly (Series B through Nasdaq IPO), Airbnb (Series F), SenseTime, Moore Threads, Leapmotor and EVE Energy, and on the A.S. Watson tri-market IPO and Temasek's strategic stake. I have watched analyst memos get torn apart, and signed off on decisions where being wrong meant being wrong by nine figures. From that seat, almost every AI finance demo I have seen has the same problem: it optimizes for the demo, not for the memo. This talk walks through the specific failure modes that kill AI agents at the IC door: Source hierarchy is not retrieval. A footnote in an audited 10-K outweighs a sell-side note, which outweighs a transcript, which outweighs an internal email. Most RAG systems flatten this. Numerical consistency is non-negotiable. A memo that says "revenue grew 18%" in paragraph one and "17.4%" in the sensitivity table is dead on arrival. Contradiction is a feature. Real diligence surfaces conflicts between sources; AI agents tend to silently resolve them. Every assumption must be separable from every fact. Investment committees do not approve assumptions hidden inside prose. Audit trail is the deliverable. If a regulator, an auditor, or a board member cannot trace a claim back to evidence in under thirty seconds, the system is unusable. Accountability cannot be delegated to a model. Someone has to sign the memo. The architecture has to reflect that. The session closes with a concrete buyer-side specification, what an AI investment agent must produce, in what form, with what evidence, before a senior finance team will let it touch a live deal. Not a framework slide.

SESSION **Agentic Engineering · Track 4** 11:10am-11:30am

Anthropic's CCA Exam as a Field-Guide for Agentic Engineering

Frank Coyle — Computer Science Educator; Founder, The AI Edge, University California Berkeley

Anthropic's CCA Exam: A Field-Guide for Agentic Engineering The Claude Certified Architect (CCA) exam distills what Anthropic has learned from working with the AI companies shipping agents to production — the patterns that work, the anti-patterns that quietly burn tokens and trust, and the architectural decisions that separate demos from systems you'd stake a quarter on. This talk treats the exam as a field guide for agentic engineering, whether or not you ever sit for it. We'll walk through the five competency domains the exam tests — Agentic Architecture, Tool Design and MCP Integration, Claude Code, Prompt Engineering, and Context Management — with particular emphasis on multi-agent orchestration, subagent delegation, tool schema design, and lifecycle hooks. We'll then work through the six real-world scenarios the exam uses to probe judgment, each organized around an anti-pattern: the seductive-but-wrong move that looks reasonable until it costs you a production incident. Attendees leave with a working mental model of the agentic surface area and a checklist of the failure modes that matter most when moving from prototype to production. **Who should attend:** engineers and architects building agentic systems with Claude or other frontier models, technical leads evaluating agent designs, and developers considering the CCA credential.

SPONSOR **Graphs · Track 5** 11:10am-11:30am *tentative*

Active Graph Agent Runtime (BabyAGI 4)

Yohei Nakajima — Managing Partner, BabyAGI/Untapped Capital

Proposing a novel event-sourced graph runtime for building long-running auditable, agentic systems. Built on top of and combining various BabyAGI iterations and graph experiments (memory, code, logs) into a single primitive.

SESSION **AI in GTM · Track 6** 11:10am-11:30am

The Death of Developer Advocates

Stephanie Jarmak — Sourcegraph

SESSION **AI in Healthcare · Track 7** 11:10am-11:30am

Guardrails First: Engineering Member-Facing Health AI

Rashi Agrawal — Head of Agentic AI, Hinge Health

Everywhere else in the company, an AI pilot can reach production in weeks. For our member-facing clinical assistant, it can't, and that single constraint redesigned our entire architecture. This is a field report on building conversational AI in a regulated digital health setting, where "move fast and break things" isn't a culture choice. It's a liability. We'll get concrete about what changes when every output has to be clinically safe, auditable, and compliant: PHI is protected by architecture, not policy. Production and non-production are hard-isolated, dashboards are sanitized, and engineers outside the US never touch protected health information. Must-not-fail behavior never lives in a prompt. Emergency escalation and intent routing run as deterministic rules at the top of every conversation turn, before the model is consulted. If you can't afford to get something wrong, you don't leave it to a probabilistic system. Clinical safety is a continuous eval layer. ~30 LLM-as-judge evaluators score clinical accuracy, clinical safety, escalation routing, and recommendation relevance, continuously, not once. Every output is auditable. Each turn, tool call, and reasoning step is traced so outputs can be reviewed and meet regulated reporting obligations. The throughline: in regulated healthcare, compliance constraints aren't a tax you pay around the architecture. They become the architecture. We'll talk about why guardrails-first is the only way to ship member-facing health AI, and why "painfully slow" is sometimes exactly right. (This is non-diagnostic, member-facing AI. The talk is about engineering discipline under regulation, not medical claims.) Key takeaways - In regulated health AI, "move fast" is the wrong default. Design for deliberate, careful launches. - Must-not-fail behaviors belong in deterministic rules at the top of every turn, never in the prompt. - Protect PHI through architecture: isolate prod from non-prod, sanitize dashboards, restrict access by role and geography. - Make every output auditable. Trace each turn, tool call, and reasoning step so safety is reviewable, not assumed. - Treat clinical safety as a continuous LLM-as-judge layer, not a one-time gate.

SESSION **SemiAnalysis · Track 8** 11:10am-11:30am *tentative*

To be announced

SESSION **Inference · Track 9** 11:10am-11:30am

Routing LLM Inference in Production: From Engine Signals to Policy

Qianru Lao — Member of Technical Staff, Inference, OpenAI · Lu Zhang

Production LLM apps need more than a fast model: they need an inference routing layer that can choose where each request should run as engines, capacity, latency, and geography cost change. This talk shares a generalized Inference Load Balancer (ILB) proxy/controller architecture. A low-latency proxy applies routing weights and request-path signals, while a controller computes source-cluster-to-engine weights from demand, capacity/performance profiles, replica state, and geography cost. We will cover the practical debugging patterns AI engineers need: reading engine signals, explaining why a request went to one backend instead of another, handling retries and load shedding, and keeping routing behavior observable without exposing OpenAI-specific internals or non-public metrics.

SPONSOR **Track M** 11:10am-11:30am

M2

SESSION **AI-Native Enterprises · Leadership 1** 11:10am-11:30am *tentative*

Tribal Dungeons of Global Shipping CX: AI Agents at 100K Cases a Day

Dmitry Buykin — Maersk

Most "AI agents in production" talks skip the part where you have to drag tribal knowledge out of 100+ country SME teams and turn it into something an agent can execute safely. This is that part. How Maersk ships AI agents handling 100K customer cases a day across global logistics, and why extracting and aligning the tribal knowledge was 10x harder than the agent itself. - Why SOPs-as-code (versioned markdown, per-country) beats prompt engineering at this scale - The SME alignment loop: how corrections become SOP changes without breaking 99 other countries - Guardrails that matter in production: write-gating, loop breakers, classifier vs. SOP-body routing layers - Where agents under-deliver against the demo, and how we measured it honestly - Org/process patterns for the Applied AI / Forward Deployed Engineer stack across 100+ countries

SESSION **AI Architects: AI Factories · Leadership 2** 11:10am-11:30am

The AI Race Isn't Being Won at the Model Layer — It's Being Lost at the Infrastructure Layer

Ethan Batraski — Partner, Venrock

SESSION **Expo Stage 1** 11:10am-11:30am *tentative*

Beyond RAG: See a relational context engine reduce token burn

Brandon Waselnuk — Founder, Unblocked

In this expo talk we'll give you a free context engine simulator, open source tools, and demo how a context engine works. See how modern engineering workflows with agentic loops and goals produce better quality code and reduce token burn. RAG, while useful, leaves context gaps for humans and agents. A context engine fills those gaps by including real-time, relational, personalized, and permission aware techniques to get high-signal context to humans and agents at runtime.

SESSION **Expo Stage 2** 11:10am-11:30am *tentative*

Keycard AI Expo 1

SESSION **Expo Stage 3** 11:10am-11:30am *tentative*

The Lethal Trifecta Is Already on Your Developers' Laptops

Michael Patterson

The lethal trifecta: an AI agent with access to private data, exposure to untrusted content, and the ability to communicate externally. Combine all three and an attacker can trick your agent into exfiltrating anything it can see and there is no prompt-level fix.. Most enterprises have already deployed this pattern at scale: Claude Code, Cursor, and Copilot on developer laptops with local credentials, MCPs reaching into internal systems, and open egress. I'll speak to my own personal agent stack as a textbook example, then trace the same shape across enterprise deployments I see at Coder. The back half is four architectural moves that defuse it: governed compute, centralized credentials, default-deny egress, identity-bound audit. Walk out with a mental model and a checklist you can run against your own deployment the next morning.

SESSION **Expo Stage 4** 11:10am-11:30am *tentative*

Dynatrace Expo Session

Expo Session 18 minutes Expo floor stage Expo Sessions are dedicated, 18-minute technical presentations delivered by sponsors in designated Expo Session rooms during conference expo hours. These sessions are designed to allow sponsors to engage directly with attendees through a structured, technical presentation format. To give you an overview of what happens in an expo session and how its being collected please see key details below: Duration: Sessions run for approximately 18 minutes each. Placement: They take place in dedicated Expo Session rooms during scheduled expo hours and are listed in the official conference agenda and event schedule. Content Focus: Sessions should be technical and informative, focusing on thought leadership, deep technical insights, architecture discussions, or engineering case studies. They are intended to drive interest to your product/service/booth exhibit by showing how your team is solving technical problems and are explicitly discouraged from being overly promotional or a "vendor pitch". Sponsors commonly use these for technical deep dives, product demonstrations, implementation walkthroughs, or customer case studies. Lead Capture: Opt-in lead data is provided for attendees who scan into the session. This lead data includes Name, Email, Job Title, Company, City, Country, and Company Size.Session Title and Session Lead: Once you have access in your Accel Events sponsor portal, you'll be able to add your session title and session lead directly. Please note that you can update/edit this until June 1, 2026.

11:40am

SESSION **Harness Engineering · Main Stage** 11:40am-12:00pm

The Unreasonable Effectiveness of Separating the Task from the Model

Maxime Rivest — DSPy · Isaac Miller

By declaring your task's inputs and outputs without initially considering model capability, you create the space needed to figure out the model execution later. DSPy's entire promise is that you should evaluate and execute your AI engineering at a level higher than a specific prompt template or a particular provider's API shape: the Signature. However, models have evolved significantly over the last few years. How can the same input and output specifications still work in a world now filled with tools, RLMs, and Skills? By defining your task strictly through its inputs and outputs, the underlying implementation becomes completely flexible. You can experiment with different models, settings, weights, templating strategies, and output formats, all without touching your actual AI workflow. Consequently, you can leverage components built by others and focus entirely on your core AI task. In this talk we will present how dspy 3.5 makes it easier much easier. DSPy has its roots in prompt optimization, where we build efficient ways to conduct search and learning beneath the signature. In this talk we will give a preview of DSPy 4.0 where we use the fact that models have now passed a tipping point for two critical concepts we have always needed. First, we no longer need to limit the search space to a single instruction block per LLM call; models can now reliably write the code underneath a signature themselves —so they should. Second, traditional prompt optimization has always required a scalar metric, which is notoriously one of the hardest parts to get right. What if a DSPy program could learn directly from your interactions with users? Ultimately, all you care about is that the function you call respects the inputs and outputs of your signature. You can let the models figure out the rest.

SESSION **Generative Media · Track 1** 11:40am-12:00pm

Reelful: AI-generated Reels from photos and clips

Kate Deyneka — Solo founder, Reelful

AI-powered mobile app that turns photos and short clips into ready-to-post Instagram Reels and TikToks without timeline editing, manual prompting, or voice recording.

SPONSOR **Agentic Commerce · Track 2** 11:40am-12:00pm

When AI Agents Pay and Sellers Monetize: Building x402 Apps for Agentic Commerce on AWS

Anil Nadiminti — Senior Solutions Architect, Amazon Web Services

As Agentic AI moves from chat to execution, autonomous agents need a native way to discover, access, and pay for digital services in real time. This session explores how x402 can turn HTTP into a payment-aware interface for machine-to-machine commerce, unlocking crypto-native patterns like programmable access, pay-per-use APIs, and on-demand monetization for data, tools, and services. We'll show how to build x402-enabled applications and walk through the architecture, the full agentic payments flow, seller monetization strategies, payment verification, and design tradeoffs involved in making agent-driven transactions secure, scalable, and production-ready. Attendees will leave with practical patterns for building apps where AI agents do not just call APIs — they can discover services, evaluate costs, transact autonomously, and enable new revenue models for sellers.

SESSION **AI in Finance · Track 3** 11:40am-12:00pm

Let's integrate AI Agents in Event-Sourced Systems

Divakar Kumar — Technical Architect, Flyers Soft Private Limited

Fraud detection has always been a race against time. In traditional event-sourced systems, every transaction, login, or transfer is captured as a sequence of immutable events. These events tell a clear story — but only after the fact. What if events could do more than just record history? What if they could talk back? In this talk, we'll explore how agentic event-driven systems transform fraud detection. Imagine every PaymentInitiated, LoginAttempt, or DeviceChanged event not just being logged, but immediately consumed by an autonomous Fraud Detection Agent. This agent correlates events across accounts, reasons over historical event streams, and generates new events like SuspiciousActivityFlagged or TransactionHeldForReview. Through a real-world inspired use case in banking and digital payments, we'll show: - How event sourcing provides the perfect memory layer for fraud detection agents - Patterns for agents to safely inject new domain events without violating invariants - How to avoid runaway feedback loops when multiple agents interact (e.g., fraud + compliance + customer service agents) - Governance, auditing, and explainability challenges when autonomous agents take part in mission-critical workflows By the end of this session, you'll see how event-driven DDD systems evolve when agents stop being passive consumers and start actively shaping the event stream — turning fraud detection from a reactive process into a proactive, adaptive defense.

SESSION **Agentic Engineering · Track 4** 11:40am-12:00pm

Guide, Verify, Solve: The Engineering Discipline Agentic Development Demands

Manish Kapur — Sonar

Agentic development is not a productivity story: it's a reliability engineering problem at a scale most teams have never faced. Long-running agent tasks fail at alarming rates, pull requests have grown from 50 lines to 5,000, and cognitive surrender is real—the more capable AI output appears, the less humans interrogate it, right at the moment the stakes are highest. Independent, peer-reviewed research from Carnegie Mellon studying 807 open source projects found that AI agent adoption caused a persistent 30% increase in code analysis warnings and a 41% increase in complexity — with long-term development velocity declining as a result. Agents don't just write code faster, they accumulate debt faster, too. The answer is not to slow agents down, it's to govern and refine the loop they operate inside. Sonar's Agent Centric Development Cycle (AC/DC), defines that loop across three continuous stages: guide agents with project-specific context and constraints before a single line is written; verify rigorously and continuously inside the loop, not downstream in CI; and solve issues automatically before they ever reach a manual review. The deeper insight is that this is not primarily a security story. It's an efficiency story. Codebases riddled with complexity make agents slower, less reliable, and significantly more expensive to run. Every token spent navigating legacy debt is a tax on every future agent run. Well-maintained, low-complexity codebases mean fewer failures, fewer tokens, and faster iteration. The teams that instrument this loop now will do more than ship safely: they'll compound their advantage every time an agent touches their codebase. Verification isn't a cost center. In an agentic world, it's a competitive moat.

SPONSOR **Graphs · Track 5** 11:40am-12:00pm *tentative*

From Systems of Record to Systems of Context

Omri Bruchim — Engineering Group Lead, AI, monday.com

Enterprise AI agents are moving fast, but most of them still hit the same wall in production: they have access to tools, documents, APIs, and databases, but they do not understand the real context of how work gets done. At monday.com, we are building agents that operate across real customer workflows, internal product surfaces, knowledge, permissions, memory, and actions. The hard part is not just calling the right tool or retrieving the right document. The hard part is building a reliable context layer that helps agents understand users, work objects, organizational knowledge, prior decisions, business rules, and the relationships between them. This talk will explore the emerging idea of the context graph: a living, queryable layer that connects entities, history, permissions, decisions, and meaning across an organization. Foundation Capital describes context graphs as the next major enterprise AI opportunity because agents need more than rules. They need decision traces: how rules were applied, where exceptions were made, who approved what, and what precedent actually governs reality. I will share how we think about this opportunity at monday.com, how we are implementing parts of it in practice, and what we have learned from building AI agents inside a real AI work platform. The talk will include concrete examples, including how context is collected, represented, retrieved, governed, and evaluated. The audience will leave with a practical framework for moving beyond one-off RAG pipelines and prompt stuffing toward a reusable context layer that compounds over time, improves agent quality, and becomes a strategic moat for companies building AI-native products.

SESSION **AI in GTM · Track 6** 11:40am-12:00pm

AI in GTM at Notion

Flora Liu — Notion

SESSION **AI in Healthcare · Track 7** 11:40am-12:00pm

Building a multi-agent system for dialogue-based clinical care

Clara Matos — Director of AI Engineering, Sword Health

Deploying LLM-based systems in healthcare requires careful orchestration of safety guardrails, memory architectures that preserve clinical context, and rigorous evaluation, all while meeting strict regulatory, privacy, and safety requirements. In this talk, we share how we are building Phoenix, a dialogue-based AI care specialist that guides patients through their care journey with human oversight. We'll walk through our system design: a multi-agent architecture powered by proprietary foundation models; a memory system managing short-term conversation context and long-term patient knowledge; layered safety guardrails using policy-conditioned models for input/output moderation; decision logic for human escalation; and our complete evaluation lifecycle, from offline automated and human evaluation before release, to online observability and A/B testing in production. By the end of this session, you'll walk away with practical lessons learned building a production-grade conversational AI system for clinical care.

SESSION **SemiAnalysis · Track 8** 11:40am-12:00pm *tentative*

To be announced

SESSION **Inference · Track 9** 11:40am-12:00pm

Are LLM Performance Benchmarks Reliable?

Ashok Chandrasekar — Google · Jason Kramberger

Standardizing performance benchmarks for production-grade Large Language Models is currently a significant challenge across the industry. Conflicting data is prevalent, whether originating from server developers like vLLM and SGLang or from various analysts and competitive benchmarks, and these results often fail to hold up under real-world conditions. Our research into these inconsistencies identified several critical factors, including the constraints of single-process tools, specifically the Python Global Interpreter Lock (GIL) and the nuances of model-level settings like temperature. Furthermore, a lack of transparency regarding load generation parameters such as QPS and concurrency, paired with insufficient observability into the benchmarking clients themselves, contributes to these disparate outcomes. In this talk, we share key lessons learned from our benchmarking efforts, examining the primary pitfalls that distort performance data and offering strategies for mitigation. Additionally, we will introduce Inference Perf, an open-source, multi-process utility we developed to provide reliable stress-testing for production stacks. Our goal is to promote standardized, real-world benchmarking practices that allow the community to move beyond unreliable data. Join us to discover how to accurately measure, optimize, and report LLM performance with certainty.

SPONSOR **Track M** 11:40am-12:00pm

M3

SESSION **Inference · Leadership 1** 11:40am-12:00pm

All the Things We Have to Do to Satisfy Your Insatiable Need for Tokens

Daniel Kim — Head of Growth, Cerebras Systems · Natalie Serrino

Every time the industry figures out how to serve tokens faster and cheaper, the appetite grows to match. Models get bigger, contexts get longer, agents start chaining thousands of calls together. The finish line keeps moving. This talk is a technical tour through everything the industry has done to keep up, led by two experts in high-performance inference. We'll start with the optimizations that made hardware work harder without changing the underlying architecture. Then we'll go up a level with techniques that work smarter across requests and across the model itself. And finally, a peek into the future with heterogeneous disaggregated inference, the architectural shift that splits prefill and decode across specialized hardware, and even more advanced forms of hardware specialization coming your way soon. Token demand is about to get a lot more insatiable. Let's see what the future has in store for us!

SESSION **AI Architects: AI Factories · Leadership 2** 11:40am-12:00pm

What If Your Chip Design Team Moved Like a Single Body?

Khaled Alashmouny — AIDAChip

SESSION **Expo Stage 1** 11:40am-12:00pm *tentative*

The Art of Building Verifiers for Computer Use Agents

Miguel González Fernández · Corby Rosset

Every team building browser agents has the same problem: you can't trust your own evals. Browser tasks are too open-ended for deterministic checks, so teams use LLM verifiers as judges, and the judges are wrong constantly. WebVoyager misses 45% of failures. WebJudge misses 22%. Used as RL reward, you're not training a better agent, you're training a more confident liar. This talk walks through the Universal Verifier, open-sourced with Microsoft Research: false positive rate near zero, Cohen's kappa matching human-human agreement. Four design principles, one open benchmark, and an honest account of where auto-research worked and where it plateaued.

SESSION **Expo Stage 2** 11:40am-12:00pm *tentative*

Seeing the Plumbing: Profiling vLLM Speculative Decoding on NVIDIA Blackwell

Sheilah Kirui

Speculative decoding promises dramatic LLM speedups by using a tiny draft model to guess tokens ahead of a large target model. However, dual-model serving fundamentally rewrites your memory dynamics and introduces a rigid engineering trade-off: guess right, and you bypass the memory-bandwidth bottleneck; guess wrong, and you waste compute. This session is a live-demo routing identical workloads through baseline and speculative configurations in vLLM on a single NVIDIA RTX 6000 Blackwell GPU. Splitting the screen between a Streamlit app and a live Grafana dashboard, we will profile the inference engine across three vectors: Time per Output Token (TPOT): The real-time, user-facing latency delta. KV Cache & Memory Footprint: The exact VRAM tax of tracking parallel token states within a 96GB budget. Draft Acceptance Rate: Visualizing the tipping point where dropping acceptance rates cause speculative decoding to fall below baseline efficiency.Supporting MaterialsProject Repository: https://github.com/akamai-developers/speculative-decoding-example-vllm-blackwell# (Work In Progress / Active Development)

SESSION **Expo Stage 3** 11:40am-12:00pm *tentative*

While You Were Generating: The Verification Gap Nobody Talked About

Every enterprise is asking the same question: how do we move fast with AI without breaking things? While the market chased generation better models, faster agents, more output a different problem was compounding quietly: nobody built the verification layer to match. The team built Gitar because they saw firsthand what happens when development velocity outpaces code quality, and AI has made that problem an order of magnitude bigger. In this session, Ali-Reza Adl-Tabatabai, formerly of Uber, Google, and Meta, now leading Gitar development inside Sonar, makes the case for why AI-native code review is the missing layer in every enterprise's agentic stack. Gitar uses agentic reasoning to review code, generate fixes, validate them against your CI, and commit to the branch. It automatically analyzes and de-duplicates CI failures, detects flaky tests, and fixes remaining build, lint, and test failures keeping reviews moving across time zones without the back-and-forth that kills engineering throughput. In an agentic world, zero trust is an engineering principle: assume every line an agent writes needs to be verified, every time, at every layer.

SESSION **Expo Stage 4** 11:40am-12:00pm *tentative*

Voice is the universal interface

Kwindla Kramer — Daily · Neil Zeghidour — CEO, Gradium

Language models give us the ability to create natural language, conversational, interfaces for computers. We are seeing a rapid shift among early adopters to using general language instead of traditional user interfaces for tasks like writing code and editing spreadsheets. Join the cofounders of Pipecat, Gradium, and Daily as we discuss the future of realtime voice and AI interfaces. Voice is the most efficient input mode for natural-language systems, and often the most efficient output mode, as well. But good voice interfaces require a very high degree of conversational facility, intelligence, task-specific reliability, and robustness to real-world realities like multiple speakers and background noise. There's a long history of voice interfaces in science fiction: Star Trek, Iron Man, Her. We'll use these depictions of computing possibilities as a jumping off point for talking about the ideal voice interface. How close are we to being able to build these interfaces with today's models, hardware, orchestration tooling, and UI libraries? What are the most promising research directions? What did the movies get wrong, now that we actually have experience building natural language, open-ended, voice systems?

12:05pm

SESSION **Harness Engineering · Main Stage** 12:05pm-12:25pm

Harness Engineering: Building the Production Cage for Powerful Domain Agents

Mike Chambers

Every agent is a while loop. The model takes strings in and produces strings out. We've all written it, debugged it, shipped it. And yet every team building agents is still re-inventing the same session management, truncation logic, tool wiring, and memory plumbing from scratch. The hard part is the harness: session isolation, context management, memory persistence, sandboxed execution, observability. The machinery that makes a model dependable in production. Most of the failures we see in deployed agents (context rot, premature completion, tool bloat) trace back to harness problems, not model problems. This talk covers what a harness actually does, why "harness engineering" suddenly showed up in engineering posts from everyone, and what changes when you stop building harnesses by hand. In live demos, we'll build the same agent three ways: hand-rolled Python, framework-generated, and fully managed through a single API call. Each level shifts the failure modes from infrastructure plumbing to engineering judgment, where the real questions are what context to preserve, when to verify, and how to keep an agent from finishing half the job and calling it done. The harness handles the machinery. You still have to engineer the behavior.

SESSION **Generative Media · Track 1** 12:05pm-12:25pm

While my guitar gently speaks

Todd Fisher — Head of Software Engineering, Philo Ventures

Do you ever wonder What the next evolution of live performances will look like? I do all the time. Come experience what happens when you combine live guitar playing with DSP as well as TTS and other models, all running locally. Prepare to be entertained and get familiar with new possibilities that modern AI opens up in the audio and digital signal processing space while you enjoy a live performance on top of an informative slide presentation. Walk away from this talk inspired to help build the next evolution of tools for musicians and live performances. We will touch on how to build with tools such as classic DSP, JUCE, on device TTS, CoreML, WhisperX, CoreMIDI and more! You might even get a chance to have a conversation with a guitar!

SPONSOR **Agentic Commerce · Track 2** 12:05pm-12:25pm

x402 isn’t good (yet)

Jan Curn — Founder & CEO, Apify

While everyone understands that agents will get more done with a budget, no one knows which protocol will win agentic payment standard wars: x402, MPP, Skyfire, or another? So far, x402 is the most mature protocol with the largest transaction volume, but even its new "upto" payment scheme doesn’t support true usage-based pricing, as it gives agents a chance to consume resources and then skip out on the bill. I’ll walk you through our experience (and pains) implementing agentic payments for a marketplace of 30K+ web Actors, and how we made it work even with the current specs.

SESSION **AI in Finance · Track 3** 12:05pm-12:25pm

Claude Cowork in Finance

Lydia Hallie — Claude Code team, Anthropic

SESSION **Agentic Engineering · Track 4** 12:05pm-12:25pm

Benchmarking Coding Agents on New vs Legacy Code bases

Denys Linkov — Head of ML, Wisedocs

You have an old code base with 100,000s of lines of code, should you let an AI Agent refactor or do you wait until you have a cleaner setup? Last year we refactored a number of code bases and ran evaluations on how well different models, harnesses and rule sets affected multiple versions of the code base. This talk will feature specific code examples as well as a broader set of evals.

SPONSOR **Graphs · Track 5** 12:05pm-12:25pm *tentative*

Your Moat Is Your Data Model

Mike Phipps — Lead AI Engineer, Gates Foundation

Every enterprise AI team faces the same strategic question: where in the stack should a small team focus its effort? Models, frontends, and agent frameworks evolve rapidly and are increasingly commoditized. But regardless of how these layers mature, AI in enterprise settings remains bottlenecked by the same underlying problem: structured data is siloed across systems of record with domain-specific schemas, and the unstructured data needed to contextualize it sits in entirely separate systems, with its own systematic complexities. The durable work is cleaning, curating, and semantically modeling this data in an AI-first manner so that any client — chat, workflow, or otherwise — can query across it. That's the moat. At the Gates Foundation, my team built and deployed our foundation-wide knowledge graph on Neo4j that unifies structured and unstructured data behind a single MCP server. The graph itself is modeled for agentic consumption: natural hierarchies are projected as traversable paths rather than flattened tables, and unstructured documents are semantically chunked, tagged, and mapped to structured entities at ingestion time using AI-driven ETL. The result is a semantic layer where an agent can express a complex cross-system question as a concise graph query and receive an accurate answer. This talk is an architectural walkthrough covering the end-to-end pipeline: AI-based extraction and semantic chunking of unstructured documents, the agent-first data modeling decisions, design considerations for our MCP server, and how we handle graph-based retrieval evals. We'll walk through real query sessions showing Claude interacting with the graph through both chat and workflow integrations. The intended takeaway is a practical framework for where a small enterprise team's investment compounds — and why that investment is the data model, not the layers above it.

SESSION **AI in GTM · Track 6** 12:05pm-12:25pm

Ramp: AI in GTM

Arman Vaziri — GTM, Ramp

SESSION **AI in Healthcare · Track 7** 12:05pm-12:25pm

200 Million Patient Interactions Later: What the Generic Voice Stack Misses

Vivek Raju Muppalla — VP of AI Engineering, Hippocratic AI

A healthcare voice agent can be right on the benchmark and still fail in production. Real patients hesitate, interrupt, misremember medications, code-switch mid-sentence, and disclose risk indirectly. After 200M+ patient-agent interactions, the lesson is clear: in clinical voice AI, interaction is a safety variable. This talk breaks down what Hippocratic AI had to rebuild beyond the generic voice stack: not just ASR, VAD, an LLM, TTS, and turn-taking heuristics, but a real-time safety system that treats silence, clarification, escalation, multilingual continuity, and medication-specific recognition as first-class engineering problems. We'll walk through the production architecture behind Hippocratic AI's voice agents: a 30+ model supervisor constellation, including the 4.1T-parameter AI Front Door system, designed to catch failures a single primary model misses. The talk covers how specialized models monitor medication identification, overdose risk, labs and vitals, escalation criteria, workflow confirmation, and other clinical safety surfaces while the patient conversation is still happening. We'll focus on four production lessons: Benchmarks are not enough; Interaction signals become training data; One LLM is not a safety architecture; Voice infrastructure has clinical failure modes.

SESSION **SemiAnalysis · Track 8** 12:05pm-12:25pm *tentative*

To be announced

SESSION **Inference · Track 9** 12:05pm-12:25pm

Vertical Mobility: Building an AI Inference Platform That Scales from MVP to Trillion-Parameter Workloads

Rita Zhang · Sitanshu Gupta

The future of AI inference is not one-size-fits-all. This talk explores a multi-tiered architecture that supports the full AI lifecycle, from rapid, pay-per-token experimentation to dedicated, SLO-bound production and extreme-scale, self-managed deployments. Learn about lessons learned from CoreWeave's inference stack as performance, cost, and control requirements evolve.

SPONSOR **Track M** 12:05pm-12:25pm

M4

SESSION **Inference · Leadership 1** 12:05pm-12:25pm

Stop Model Shopping: Why Ownership Beats Choice in the Agent Stack

Lin Qiao

SESSION **AI Architects: AI Factories · Leadership 2** 12:05pm-12:25pm *tentative*

HOLD — DigitalOcean

SESSION **Expo Stage 1** 12:05pm-12:25pm *tentative*

The Missing Layer in Agentic AI

Giedrius Steimantas

Reasoning is solved. Web access isn't. Most agents break the moment they leave the sandbox blocked, rate-limited, or staring at a CAPTCHA. Giedrius will show the three primitives every production agent needs: a browser, a fast search API, and a universal scraper and demo an agent built on top of them that actually works in the wild.

SESSION **Expo Stage 2** 12:05pm-12:25pm *tentative*

Replicated Expo Session

SESSION **Expo Stage 3** 12:05pm-12:25pm *tentative*

PLANETSCALE Expo 1

SESSION **Expo Stage 4** 12:05pm-12:25pm *tentative*

Agents That Forge Their Own Tools: Self-Modifying AI in the Wild

Sandhya Subramani — Senior Developer Advocate for Generative AI, Amazon Web Services

What happens when your agent decides its existing tools aren't good enough and writes new ones? Self-modifying agents can generate, test, and deploy their own tool implementations at runtime, adapting to problems they weren't explicitly programmed to solve. In this session, we'll demo a live agent that forges its own tools on the fly, discuss the safety boundaries you need, and explore where this pattern makes sense (and where it absolutely doesn't).

12:30pm

12:30pm-1:30pm **PLACEHOLDER — Fireside Chat: Gergely Orosz × Simon Eskildsen**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch & Learn**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch & Learn**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

12:30pm-1:30pm **Lunch**

1:30pm

SESSION **Harness Engineering · Main Stage** 1:30pm-1:50pm

Loophole - Adversarial Agents To Stress Test Your Morality

Brendan Rappazzo — Machine learning researcher, Morgan Stanley

Most natural language specifications have holes their authors didn't notice - and writing more rules tends to create more holes. I built Loophole to try a different approach: point adversarial agents at a spec until it stops breaking. You give the system a set of natural language principles. An AI drafts a formal codified version. Two adversarial agents go to work - one finds cases the code permits but the principles forbid, the other finds cases the code forbids but the principles allow. A judge agent patches the code when it can, but only if the fix doesn't contradict any prior ruling. When a contradiction can't be resolved, it escalates to you. Every decision becomes binding precedent, so the constraint space tightens round after round. I started with moral and legal reasoning as the demo, and on its own that's already interesting - it turns into a kind of game where you discover contradictions in your own beliefs that you didn't know were there. But the pattern generalizes well past that. The same loop works for company policies that need to survive contact with edge cases. For making chatbot system prompts adversarially robust. For stress-testing eval rubrics. And, taking the long view, for something like a smarter legislative process - where proposed laws get checked against the public's stated values before they pass, and the contradictions surface before they hit a courtroom. The talk walks through how the harness works, the design choices that matter (especially why precedent is the load-bearing piece), what kinds of specs it handles well, where it breaks, and what it would take to push it further. All code is open source.

SESSION **Generative Media · Track 1** 1:30pm-1:50pm *tentative*

To be announced

SPONSOR **Agentic Commerce · Track 2** 1:30pm-1:50pm

Agent Spending Without Controls: The Missing Infrastructure Layer for AI Pa…

Rodrigo Coelho — Edge & Node · Pranav Maheshwari

SESSION **AI in Finance · Track 3** 1:30pm-1:50pm *tentative*

To be announced

SESSION **Agentic Engineering · Track 4** 1:30pm-1:50pm *tentative*

MCPs, CLIs, and Skills: Choosing the Right Tooling Layer for Agentic Development

Ankush Agarwal — Member of Technical Staff, OpenAI · Nikita Kothari

Agentic development needs more than one interface: MCPs provide clean, portable connectors to services, with built-in patterns for security and auth. CLIs offer composability, debuggability, and workflows developers already trust. Skills teach agents how to use a wide variety of tools and MCPs effectively without overloading context.

SPONSOR **Graphs · Track 5** 1:30pm-1:50pm *tentative*

AI : Learned Execution Graphs for Real-Time Anomaly Detection & Drift Classification in APIs

Ritvik Pandya — Engineering Leader

API ingress controllers process requests through ordered sequences of middleware steps — authentication, authorization, validation, rate limiting, routing, service invocation, caching. We model this pipeline as a directed acyclic graph (DAG) learned from structured telemetry events, then apply graph-based anomaly detection and drift classification in real time at 1,600+ TPS. The system emits one structured event per processing step, constructs per-endpoint execution graphs using sequence mining with statistical confidence thresholds, and learns per-node baselines (latency, dependency, execution frequency). Three graph intelligence capabilities emerge: (1) Graph-based anomaly attribution — compute per-node deviation ratios against learned baselines to identify the exact bottleneck node and its dependency. In production, this pinpointed a 41x deviation at a single graph node that was invisible to service-level monitoring, reducing root cause identification from 2-3 hours to under 30 seconds. (2) Graph structural drift detection — compare observed node sequences against the learned graph topology to detect missing nodes (mandatory processing step silently skipped), reordered nodes (middleware misconfiguration), and unexpected new nodes (unauthorized middleware injection). Traditional monitoring reported "system healthy" when a mandatory node was removed — latency dropped, errors at zero — only the learned graph comparison detected the structural change. (3) Per-client graph fingerprinting — learn client-specific execution graph profiles using exponential moving averages. Detect when a client's graph traversal pattern changes, classify the cause (client behavior change vs. configuration drift vs. infrastructure failover) using KL divergence on node-visit distributions, and apply graph-aware adaptive control scoped to specific nodes rather than entire endpoints. The execution graph model also enables a novel approach to retry storm detection: analyzing idempotency key entropy at graph nodes to classify traffic as legitimate growth vs. retry amplification, and returning cached responses at the specific graph node rather than rejecting requests — breaking the retry amplification loop. Production system processing high TPS. Attendees will learn the graph construction methodology, the anomaly attribution algorithm, and concrete patterns for adding learned graph intelligence to any middleware pipeline.

SESSION **AI in GTM · Track 6** 1:30pm-1:50pm

The Vibe GTM Iceberg

Alex Bauer — Co-Founder, Product, Upside

SESSION **AI in Healthcare · Track 7** 1:30pm-1:50pm

AI is the becoming the World's largest Relationship Therapist. We Can't Afford to Get it Wrong.

Clay Cockrell — Licensed Clinical Social Worker (LCSW); Founder, CoupleWork · Tony Fabrikant

Millions of people are now turning to AI for relationship advice and emotional support, often before they'd ever consider a human therapist. Most of the AI Therapy that is available is without clinical oversight, ethical frameworks, or any serious reckoning with what it means to intervene in the most intimate and vulnerable space in a person's life. People are getting hurt. As a couples therapist with 30 years experience, I teamed up with the former CTO at S&P and we created CoupleWork, an AI relationship therapist I essentially trained on three decades of clinical knowledge and every evidence-based modality that exists. Our voice interactive AI, Maxine, is proving this can be done responsibly and very effectively. And what we're learning about the nature of love, connection, and human vulnerability at scale is something this industry needs to hear. I also want to talk about what comes next: the regulatory frameworks that don't yet exist, the liability questions nobody is answering, and why the therapists who should be leading this conversation are almost entirely absent from it.

SESSION **SemiAnalysis · Track 8** 1:30pm-1:50pm *tentative*

To be announced

SPONSOR **Track M** 1:30pm-1:50pm

M5

SESSION **AI-Native Enterprises · Leadership 1** 1:30pm-1:50pm

From Zero to AI-Native: Scaling AI Across the Org

Josh Leavitt — Senior Director of Product Management, Coinbase

Most companies talk about being AI-native, but few show what it takes at scale. Josh Leavitt, Sr. Director of AI & Data at Coinbase, shares the hard-won playbook for transforming a high-stakes, regulated engineering organization into one where AI is a first-class citizen across every team. Josh can cover my approach towards building a centralized AI platform that serves thousands of engineers without becoming a bottleneck, tooling decisions that actually moved the needle, and what AI-native really means when shipping in a zero-tolerance-for-failure environment. Expect concrete frameworks, real examples, and honest lessons from what didn’t work.

SESSION **AI Architects: AI Factories · Leadership 2** 1:30pm-1:50pm

Coding Agents Don't Scale Themselves. Neither Do Your Teams: The Rise of Agent Enablement.

Patrick Debois — Product Developer Relations Lead, Tessl

We version control code, review it, test it, and observe it in production. We spent two decades building rigorous lifecycles around it. Now look at how we treat the context that drives AI coding agents: rules files copy-pasted from blog posts, prompts edited by hand, memory nobody audits. We're in the cowboy coding era of context. If context is the primary lever determining what agents produce, it deserves the same engineering rigor we give code. The Context Development Lifecycle (Generate, Evaluate, Distribute, Observe) gives us the stages. The process practices wrap around it: version control, peer review, CI/CD pipelines, and the team workflows to make context a shared engineering responsibility. Then there's the bigger picture: the context flywheel. As agents consume context and produce results, every observation feeds back into better context, which produces better results. The teams that get this loop spinning build a compounding advantage that becomes their moat. This is not a solved problem. It's a journey we've already started, and if the DevOps transition taught us anything, the teams that figure out the lifecycle first will pull ahead fast.

SESSION **Expo Stage 1** 1:30pm-1:50pm *tentative*

Auth0 Add-On Expo Session

SESSION **Expo Stage 2** 1:30pm-1:50pm *tentative*

YOLO Mode, Safely: microVM Sandboxes for Any Agent

Eric Jia

This talk shows the alternative: every agent session in its own microVM, with its own kernel and a hard boundary to the host. You decide what lives inside that boundary: filesystem, network, the tools it's allowed to call. The sandbox runs Claude Code, Cursor, Codex, or whatever else you're driving. You'll see an agent live in full YOLO mode inside a sandbox, a real attempt to escape, and the boundary that holds up.

SESSION **Expo Stage 3** 1:30pm-1:50pm *tentative*

Your Model is Private. Your System Isn't.

Joshua Mo

Privacy in AI isn't just about choosing the right model. Data leaks rarely happen inside the LLM itself - they happen in the systems surrounding it. Observability pipelines, analytics platforms, prompts, agents, and infrastructure often become accidental channels for exposing user data. In this session, Joshua Mo, Lead DevRel Engineer at Venice AI, explores why private models alone are not enough and shares practical privacy-preserving patterns that AI engineers can adopt today. From revocable handles and hashed identifiers to agent boundaries and confidential computing, attendees will leave with concrete ideas for building AI systems that protect user data by design.

SESSION **Expo Stage 4** 1:30pm-1:50pm *tentative*

Bright Data Expo Session 1

1:55pm

SESSION **Harness Engineering · Main Stage** 1:55pm-2:15pm

🎵 Every step you take, every call you make - the reliable agent stack

Giselle van Dongen — Restate

SESSION **Generative Media · Track 1** 1:55pm-2:15pm

Voice agents with Realtime Video

Lina Colucci — Co-founder and CEO, Lemon Slice

SPONSOR **Agentic Commerce · Track 2** 1:55pm-2:15pm

Teaching agents to pay

Anna Spysz — Stripe

SESSION **AI in Finance · Track 3** 1:55pm-2:15pm

We Vetted 2,000 AI Skills Before They Reached Developers

Lucas Palma — Information Security Manager, Nubank

AI skills and plugins are becoming part of the software supply chain. They steer agent behavior, describe tools, run commands, access files, and shape how developers build with AI. Treating them as harmless configuration is a mistake. This talk shares what we learned from building an automated security review system for more than 2,000 internal AI skills before they reached a company wide plugin marketplace. I will walk through the risks we found, the checks that worked, the checks that created noise, and how we turned skill review into something developers could run locally and in CI. We will cover practical patterns for reviewing unsafe instructions, destructive commands, sensitive data exposure, risky tool use, credential handling, external calls, and agent behavior drift. The goal is to help AI engineers think about skills, plugins, and agent instructions as production dependencies that deserve review before they reach real users.

SESSION **Agentic Engineering · Track 4** 1:55pm-2:15pm

Multiplayer agentic engineering: enabling your whole team and your best agents to work together

Arjun Singh — Co-founder and CEO, Superconductor

For a solo developer, coding agents are a superpower. For a team, they surface new kinds of bottlenecks: coordination, visibility, review, and shared context. We wanted our whole team and our best agents to work together, with no work or context trapped on any one developer's machine. So we pressed pause on the product we were building to create a multiplayer cloud workspace for agentic engineering. This talk shares five key practices we've learned from building and using our platform: Turn every surface the team uses into an agent interface. Kick off sessions from Slack, review via iOS app, iterate in GitHub comments, ship from web. Agents run in the cloud, so work keeps moving even when your laptop is closed. Make agent work visible and collaborative across the whole team. Every agent session is shared, has a live app preview, and an agent-guided code review. This allows engineers, PMs, and designers to steer and evaluate agent work collaboratively. Turn every external signal into shipped code your team can quickly evaluate. Automatically turn customer emails, meeting action items, and bug reports into agent implementations that the whole team can review. Set up shared cloud dev environments so agents aren't siloed to individual machines. Secrets, role-based access, and network controls shared across the whole team. Fast environment startup, so you're not giving up speed by moving off local. Benchmark agents on your own codebase. Claude Code, Codex, Gemini, Amp, OpenCode — how do you know which is actually better on your stack? We'll cover using your merged PRs as ground truth to build a "Personal SWE-Bench" for your codebase. Agentic engineering is going multiplayer. This is how your team gets there.

SPONSOR **Graphs · Track 5** 1:55pm-2:15pm *tentative*

Video Has No Memory. Here's How We Built One.

James Le — Head of Developer Experience, TwelveLabs

Every video AI query today starts from scratch. There's no durable state, no entity continuity, no way to ask "what does this corpus know?" instead of "find me something like this." This talk is about fixing that by engineering a proper memory layer for video intelligence, grounded in what we shipped at TwelveLabs with Jockey. What this talk covers: 1 - Why video memory is categorically different from text memory: Video is temporal, multimodal, dense, ambiguous, and evidence-sensitive. Larger context windows don't solve this. The problem isn't retrieval bandwidth, it's that there's no durable representation to retrieve into. 2 - The context graph as a systems concept, not a database choice: I'll define what "context graph" actually means in practice: time-bounded moments, cross-video entity resolution, appearance tracking, and relationship mapping. This is infrastructure-level thinking, not a graph DB sales pitch. 3 - Five design principles that determine whether video intelligence is reusable infrastructure or a search wrapper with extra steps: + Ingest once, reason many times (move expensive understanding work into preparation) + Store primitives, not just answers (moments, entities, appearances, relationships) + Ground every claim to source video (a timestamp is a product requirement, not a safety footnote) + Let intent shape memory (brand safety and sports highlights need different primitives from the same footage) + Keep the memory layer composable and API-first 4 - What this unlocks for builders. Corpus digest, agentic search with grounded references, entity-centric workflows, timeline reconstruction, and compliance tooling, all built on the same durable substrate. The talk is concrete and demo-grounded. You'll leave with a specific mental model for memory architecture, actionable decisions for ingestion pipeline design and entity resolution, and a clear line between "search with extra steps" and actual video intelligence infrastructure.

SESSION **AI in GTM · Track 6** 1:55pm-2:15pm

How We Got LLMs to Recommend Our Open Source Library (Without Paying or Plug-ins)

Christopher Burns — Founder & CEO, Inth

SESSION **AI in Healthcare · Track 7** 1:55pm-2:15pm

Healthcare's Agent Bytecode: X12 as the Harness for AI Agents

Vasant Kearney — CEO, Onlay AI

LLMs made old languages newly useful: COBOL for mainframes, Fortran for scientific code, and Rust, SQL, and Prolog as strict substrates for agentic systems. Healthcare has its own old language hiding in plain sight: X12. Before LLMs, X12 was mostly treated as ugly plumbing: loops, delimiters, companion guides, clearinghouse edits, payer-specific quirks, rejections, and acknowledgments. In an agentic workflow, those constraints become the feature. They give stochastic agents a deterministic target. This talk shows how healthcare agents can compile messy operational evidence into X12-shaped workflows: chairside audio into 837D claim narratives, imaging systems into 275/PWK attachment flows, payer portals and phone calls into 270/271 eligibility and 276/277 claim status, preauth evidence into 278 workflows, and EOBs, scanned mail, and bank data into 835/820 payment reconciliation. The core pattern is simple: LLMs reason over ambiguity; X12 provides the syntactic and semantic harness for validation, auditability, acknowledgments, rejections, human review, and high-volume automation. This is not an EDI nostalgia talk. It is a production architecture talk about building reliable agents in one of the messiest enterprise domains.

SESSION **SemiAnalysis · Track 8** 1:55pm-2:15pm *tentative*

To be announced

SESSION **Inference · Track 9** 1:55pm-2:15pm

Gavin Uberti — transformer-only ASICs for inference

Gavin Uberti — Co-Founder & CEO, Etched

Etched's Sohu approach to transformer inference on custom silicon.

SPONSOR **Track M** 1:55pm-2:15pm

M6

SESSION **AI-Native Enterprises · Leadership 1** 1:55pm-2:15pm

Which AI startups actually land enterprise contracts? Lessons from evaluating 100+ AI startups at Millennium Management

Brian Lewis — AI Product Lead, Millennium

SESSION **AI Architects: AI Factories · Leadership 2** 1:55pm-2:15pm

Why your LLM is slow and expensive: lessons learned from running models in production

Zach Bratun-Glennon — General Partner, Gradient

SESSION **Expo Stage 1** 1:55pm-2:15pm *tentative*

Dash0 Add-On Expo Session

SESSION **Expo Stage 2** 1:55pm-2:15pm *tentative*

Apify Expo Session

SESSION **Expo Stage 3** 1:55pm-2:15pm *tentative*

Neo4j Expo Session 1

SESSION **Expo Stage 4** 1:55pm-2:15pm *tentative*

LlamaIndex Expo Session

2:25pm

SESSION **Harness Engineering · Main Stage** 2:25pm-2:45pm

We let an AI agent execute Bash and lived to talk about it

Sarah Sanders — PostHog

SESSION **Generative Media · Track 1** 2:25pm-2:45pm

Generative Video at the Speed of Light

Keegan McCallum — uRun

SPONSOR **Agentic Commerce · Track 2** 2:25pm-2:45pm

The Agentic Commerce Stack

Ahnaf Prio — Senior Engineering Manager, Best Buy

Agents are already handling product discovery, cart building, and checkout — no human clicking required. But what's the protocol stack actually making this work? This talk maps the real infrastructure: MCP for tool access, A2A for agent coordination, the ACP spec (backed by OpenAI) and the UCP spec (backed by Google) — two competing approaches to standardizing the full agentic commerce lifecycle — and AP2 for agentic payments. We'll cover what each does, how they compose, and where they're still forming. Then we'll see it live — a working demo with a protocol inspector showing every tool call, task transition, and checkout event in real time. You'll leave with a clear mental model of the agentic commerce landscape and a reference implementation you can use.

SESSION **AI in Finance · Track 3** 2:25pm-2:45pm

Your Finance Agent's Bottleneck Is You

Ramana Siddanth Emani — Data Scientist, Auditoria AI

Most "AI for Finance" demos look great and almost none survive past pilot. If you've pushed an agent past one workflow, one tenant, or one Workday schema, you know the bottleneck isn't the model - it's the engineer behind the agent, who can't iterate fast enough to keep up with real AP data, real RBAC, and real query volume. What if you built your dev loop with the same primitives you're shipping to the finance team? In this talk, I'll show the subagent + skills + MCP stack - a production multi-agent system over AP, PO, vendor, and multi ERP systems, a LangGraph pattern that survives production, and the three failure modes that kill finance pilots before they ship.

SESSION **Agentic Engineering · Track 4** 2:25pm-2:45pm

What it takes to trust AI that runs production software

TBD — Resolve AI — TBD, Resolve AI

Diagnosing a production incident isn't a question you ask a model, it's an investigation: forming hypotheses, revising them as evidence arrives, and knowing when you're confident enough to act. All this under time pressure with incomplete data. This is why frontier models pointed at production hit a wall, and why most AI for production breaks on a real incident. Closing this gap is an open research problem across models, agent architecture, the context, and how you even judge whether an investigation was right when engineers themselves disagree on root cause. Join us to understand what Resolve AI Labs in building in each domain. After this session, you'll be able to tell what it takes from an AI that you can trust in mission critical production systems.

SPONSOR **Graphs · Track 5** 2:25pm-2:45pm *tentative*

On-Device Agentic AI for the New York Times Games

Shafik Quoraishee · Joanne Song

Most AI features in consumer apps follow the same architecture: user action → API call → model response → render result. That model is fine for chatbots. It's wrong for games. Game AI needs to be fast (no round-trip latency between a player's hesitation and a hint), private (a player's solve history is intimate data that shouldn't leave their device), and persistent (the agent that helps you today should remember what it learned about you yesterday). This talk is about a different architecture: a fully on-device agent, built with JetBrains' Koog framework in Kotlin, running against a local quantized model, that powers five distinct AI features across the NYT Games app — all without a single call to a remote model endpoint. We'll walk through how we used Koog's agent DSL to give the model a tool set grounded in local game data: a tool that queries your personal solve history from local storage, a tool that retrieves puzzle metadata, a tool that generates a hint at a specified level of abstraction, and a tool that decides whether to intervene at all. The agent's job isn't to answer questions — it's to reason across those tools in sequence, the same way a human coach would before opening their mouth. The five features the agent powers: 1. Crossword Hint That Earns Its Words — The agent reconstructs your current solve path (which squares are filled, which are blank, how long you've spent in each section) and reasons about where you're stuck before generating a hint. Critically, "stay silent" is a first-class tool output — the agent is allowed to decide you don't need help yet. 2. Spelling Bee Vocabulary Coach — The agent holds a personal vocabulary model built from every Spelling Bee session. It doesn't tell you what words are left. It surfaces the morphological pattern you're missing — "you haven't found any words with this suffix yet" — keeping the discovery intact. 3. Connections Misdirection Detector — Connections is designed to trick you. The agent tracks your grouping attempts and, when it detects you've been placing a "trap" word incorrectly for multiple attempts, asks the one question that reframes it without naming the category. 4. Cross-Game Session Recommender — After you finish Wordle, the agent infers your current cognitive state from how the solve went and recommends the next game. No server. No recommendation API. Just a local model reasoning over your last 10 minutes of play. 5. Archive Puzzle Matchmaker — A retrieval agent that takes natural language intent ("I want something tricky but not trivia-heavy") and searches 20,000+ archived crosswords using ocal embeddings over puzzle metadata. Your personal solve history re-ranks the results. We'll close on the architectural argument: on-device agents aren't just a privacy story. They're a capability story. The features above don't work as well — or don't work at all — when the model doesn't have persistent, intimate access to who you are as a player. Koog running on Android, against a local model, is what makes that possible today.

SESSION **AI in GTM · Track 6** 2:25pm-2:45pm *tentative*

Exa AI in GTM

Jeff Wang — CTO, Exa

SESSION **AI in Healthcare · Track 7** 2:25pm-2:45pm

Trading Desks to Clinical Trials: Parallels in Applied Vertical AI

Ayush Bhardwaj — Allos AI

SESSION **SemiAnalysis · Track 8** 2:25pm-2:45pm *tentative*

To be announced

SESSION **Inference · Track 9** 2:25pm-2:45pm

KV Cache-Aware Routing and P/D Disaggregation on Kubernetes: The Parts Public Benchmarks Don't Show

Yuchen Fama — Software engineer, Red Hat / vLLM · Michey Mehta · Ashish Kamra

SPONSOR **Track M** 2:25pm-2:45pm

M7

SESSION **AI-Native Enterprises · Leadership 1** 2:25pm-2:45pm

Your Hero Agent Needs a Party

Kunal Lanjewar — Riot Games

A front-door persona, a party of deterministic specialist agents, A2A between. Your support bot deflects half its tickets, then, soloing a problem it was never built for, confidently runs the wrong `kubectl` command. Most teams respond by rewriting the prompt. The real fix is a multi-agent party of specialists. This talk gives you a production pattern that turns one over-leveled hero agent into a coordinated party of specialists you can trust on tier-zero infrastructure. Persona and ReAct agents make great heroes at the front door. Any team can copy one, paste it into their stack, and adjust the behavior in plain English. But if you send a lone hero to clear the dungeon, whether it is a deploy or an incident, a non-deterministic Reason-Act loop tends to loop, over-act, or punt back to a human. More prompts and more skills do not reliably level it up. Instead of soloing, keep the persona as the front-door face and give it a party: deterministic DAG specialists where the graph is fixed and the LLM is called only at decision points. For example, a deployment specialist can list rolling pods, choose the next tool, run it, read logs, and then diagnose the result. Each specialist is a class with one job and a narrow set of tools, and they coordinate over A2A for capability discovery and delegation across frameworks. Reliability and tighter least-privilege access become properties of the design, not something you try to bolt onto a prompt. You'll leave with the pattern: where to draw the line between the hero and its specialists, how to shape a DAG specialist so it decides instead of flails, and where A2A fits as the seam between them, grounded in lessons from a tier-zero fleet.

SESSION **Inference · Leadership 2** 2:25pm-2:45pm

Scaling AI systems: where theory meets constraint

Zach Bratun-Glennon — General Partner, Gradient · Stephen Balaban — Co-founder / CTO, Lambda

SESSION **Expo Stage 1** 2:25pm-2:45pm *tentative*

Nebius Expo Session

SESSION **Expo Stage 2** 2:25pm-2:45pm *tentative*

How Reducto parsed the Epstein Files for the Viral JMail Project: The Secret Complexities of Documents

When the Epstein files dropped, a team of indie hackers built JMail: a duplicate of Gmail that was logged in as Jeffrey himself. It went viral. But the parsing problem underneath it was brutal. Court documents are some of the nastiest inputs a parser can face. Scanned exhibits with varying resolution, redactions sitting directly over key text, inconsistent formatting across decades of filings, handwritten annotations mixed into typed pages, documents photocopied from a photocopy of a photocopy. But legal is just one flavor of hard. In finance, you're dealing with tables nested inside tables, footnotes that span pages, and numbers that mean different things depending on which section of the filing you're in. In healthcare, it's mixed handwritten and typed content, inconsistent date formats, and forms that were designed in 1987 and never updated. In government records, it's degraded scans, stamps overlapping text, and documents where a key field is missing on half the corpus. Every industry has its own specific ways documents break parsers. This session walks through the failure modes we've actually hit across these corpora, what causes them, and how to build pipelines that hold up when the documents stop cooperating.

SESSION **Expo Stage 3** 2:25pm-2:45pm *tentative*

Neo4j Expo Session 2

SESSION **Expo Stage 4** 2:25pm-2:45pm *tentative*

The Human Is an Async API Subtitle: Designing Durable Human-in-the-Loop Agents

Melanie Warrick

Production agent systems need humans in the loop. So why do they keep getting modeled as synchronous tool calls? The agent ecosystem is focused on autonomy, but in reality, especially for high-stakes or regulated workflows, humans are a critical feature, not an afterthought. This demo-driven talk shows how to stop bolting on humans and start treating them as async-by-default endpoints with proper durability, retry, and escalation semantics. We will walk through two live, multi-agent patterns built with LangGraph and Google ADK, on Temporal for durable execution: The Agent Calls the Human. A fleet dispatch system escalates a disruption to an approver. We will intentionally kill the worker process mid-wait. Hours later, the human responds. State survives, and the agent resumes. The Human Calls the Agent. An operator interrupts a long-running task mid-flight to redirect it. The agent halts gracefully, surfaces state, accepts the override, and continues. You will leave with two production-ready architectural designs you can apply this week: agent-initiated approval gates with timeout and escalation semantics, and human-initiated interrupts with graceful agent halt and resumption.

2:50pm

SESSION **Harness Engineering · Main Stage** 2:50pm-3:10pm

No Memory, No Harness: Why the Database Is the Last Line of Defense

Kay Malcolm — Oracle

AI has made implementation faster, cheaper, and more widely available. That changes the real bottleneck in software. When every team can generate code, spin up agents, prototype workflows, and ship demos faster than ever, the advantage moves to a different layer: knowing what is worth building, who it is for, how it fits into the system, and what keeps it reliable in production. The model is the easy part. Everything that makes an agent survive contact with production lives in the harness around it: orchestration, tooling, governance, and the memory core that keeps the system grounded when the model itself is probabilistic, forgetful, and non-deterministic. This talk walks the surface areas of an agent harness and consolidates the lessons we're learning as we ship them, from agentic applications in their current form (autonomous systems that now build their own automations) to the continual-learning loops that let agents improve from their own experience. We'll look at how the discipline is segmenting. AI application development is no longer one role but several: agent engineers, memory engineers, and platform engineers. We'll map Oracle's primitives onto each as the current state of harness engineering takes shape. We'll also examine the two populations betting on this stack at once, enterprise customers who need governance, reliability, and scale, alongside the cracked developers who need fast, composable primitives, and why a well-engineered harness serves both. And we'll make the case that has held through every shift in the stack: memory isn't a feature you bolt on, it's the foundation the rest of the harness stands on. The database remains the memory core, and when everything above it is probabilistic, it's the last line of defense.

SESSION **Generative Media · Track 1** 2:50pm-3:10pm

The Next Medium: Why Real-Time Interactive Video Changes Everything for Developers

Ahmed Ahres — Reactor

SPONSOR **Agentic Commerce · Track 2** 2:50pm-3:10pm

Your Agent Just Authorized What?!

Jay Mok — PayPal

The nightmare scenario writes itself: your agent just ran off with your credit card and maxed it out on concert tickets, crypto, and a questionable NFT collection. Relax — we're building the guardrails. When an agent acts on your behalf, three questions must always be answerable: Did the human authorize this? Did they authorize this, now, in this scope? And can we prove it later? This talk maps three permissioning layers onto a stakes ladder: OAuth scopes at the bottom (broad capability, weak per-action proof, fine when reversible), Claude Code's tool-scoped allow/ask/deny model in the middle (brilliant for developer tooling, but no cryptographic evidence), and signed payment mandates at the top — where FIDO's Agentic Payments Working Group is building toward cryptographically-bound, constraint-carrying credentials. We'll share artifacts from Agent to Agent payments using our Shared Vault and Oauth to our constraint carrying Approval token leveraging our pillars of Identity and Buyer and Seller protection. You leave with a stakes × evidence matrix and a mental model that applies beyond payments: medical orders, e-signatures, securities trading, activities where you want you want to be more careful with your agent.

SESSION **AI in Finance · Track 3** 2:50pm-3:10pm

Autonomous Finance in Retail

Anant Arora — Senior Director of Product Management and Technology - Digital, Lowe's

SESSION **Agentic Engineering · Track 4** 2:50pm-3:10pm

Realtime multiplayer, automation, and you!

Idan Gazit — Senior Director of Research, GitHub Next, GitHub

Now that the models are powerful and the agents are capable, why are we still approaching software development as if it's the same activity that it used to be, but "faster"? GitHub Next thinks about what this future wants to be through two lenses: - Automation: intelligence allows us to automate much more than we could with heuristics alone. How should that automation work? What guardrails do we have to put in place so that our CISOs allow us to do that? - Collaboration: agents can understand anything in your codebase, but what about all the facts that are in the heads of your teammates? Whether it's corporate politics or taste, how do we get the humans to leak that context where agents can see it and use it to produce better outcomes? Realtime multiplayer tools have displaced every turn-based tool out there. What should that look like for code? It's not going to be as simple as multiple cursors. Come by to hear more about what GitHub Next is learning about the changing shape of software creation — one that allows us to build better, not merely faster. One that allows us to scale up teams, not only individuals. And one where automations buy us time for craft and polish, not slop. We were promised flying cars, instead we have fifteen terminals. Let's have a nicer future than that.

SPONSOR **Graphs · Track 5** 2:50pm-3:10pm *tentative*

Quill: Event-Driven AI That Keeps Documentation Alive at Enterprise Scale

Arjun Harbhajanka

Documentation is where good intentions go to die. Teams write it once, then it decays silently as code evolves. What if documentation could maintain itself? Quill is an event-driven platform that uses LLMs to automatically generate, update, validate, and summarize software documentation, triggered by the same git events teams already produce. When a PR merges, Quill reads the code changes, identifies affected docs, and opens PRs with precise updates. When a release ships, it synthesizes commit history and deployment data into stakeholder-ready release notes. When docs violate quality standards, it auto-fixes what it can and flags what it can't. This talk dives into the AI engineering decisions that made Quill work in production at scale. We cover how we designed mode-aware prompts using the Diataxis documentation framework to give the LLM a structured target instead of open-ended prose. We explore the no change narration principle, why AI-updated docs must never say previously or was changed to, and how a single prompting constraint transformed output quality. We walk through the multi-pass generation pipeline: per-file LLM calls, aggregation into a cohesive corpus, and information architecture restructuring, and how we handle repos with hundreds of source files without blowing up context windows or costs. Beyond prompting, we cover the system design: an event gateway that fans out git webhooks and package lifecycle events to specialized microservices, a Neo4j knowledge graph that models relationships between repositories, packages, and deployments, and a per-repository configuration system that lets teams opt into exactly the AI actions they trust. Attendees will leave with concrete patterns for building reliable, controllable AI-powered developer tools, not demos, but production systems.

SESSION **AI in GTM · Track 6** 2:50pm-3:10pm

Cloudflare AI in GTM with Justin Joyce

Justin Joyce — Cloudflare

SESSION **AI in Healthcare · Track 7** 2:50pm-3:10pm

How to build an AI-Native Health Company

Dan Feng — Maven Clinic

SESSION **SemiAnalysis · Track 8** 2:50pm-3:10pm *tentative*

To be announced

SESSION **Inference · Track 9** 2:50pm-3:10pm

Two Bugs That Hid in Plain Sight: A vLLM Debugging Detective Story

Asaf Gardin — Inference Engineer, AI21 · Yuval Belfer — Senior Developer Advocate, AI21 Labs

SPONSOR **Track M** 2:50pm-3:10pm

M8

SESSION **AI-Native Enterprises · Leadership 1** 2:50pm-3:10pm

AI Agents Are Just Distributed Systems Now

Salman Munaf — Lead Site Reliability Engineer, TikTok

AI agents are often described as a new kind of software, but once they move beyond chat and start calling tools, reading data, making decisions, retrying tasks, and coordinating workflows, they begin to look a lot like distributed systems. They have state. They call external services. They depend on APIs. They fail partially. They retry. They time out. They can loop. They can act on stale context. They can produce inconsistent results. And when something goes wrong, teams need logs, traces, permissions, ownership, and rollback paths just like they do with any other production system. This session will give engineers a practical way to reason about AI agents using familiar distributed systems concepts. We will break down the agent loop: planning, tool use, observation, memory, and retries. Then we will map common agent failure modes to engineering patterns teams already know, including timeouts, circuit breakers, idempotency, rate limits, least privilege, observability, and human approval. The goal is to move past the hype and treat agents like real production systems. Attendees will leave with a clear mental model for designing, debugging, and operating agents safely, especially as they become part of customer-facing products, internal developer tools, and business workflows.

SESSION **AI Architects: AI Factories · Leadership 2** 2:50pm-3:10pm *tentative*

Inside 847 Production Clinical AI Notes

Sebastian Fox — Composo

SESSION **Expo Stage 1** 2:50pm-3:10pm *tentative*

Harness Engineering: The New Core Skill for Agentic Developers

Dru Knox

Harness engineering is emerging as a new core competency for agentic engineers. Your job isn't writing good code, it's upgrading your codebase so that agents reliably succeed. This talk covers the core loop of harness engineering, the most common codebase modifications you'll make, and how to 10x your harness engineering efforts with Tessl's harness engineering agent.

SESSION **Expo Stage 2** 2:50pm-3:10pm *tentative*

Designing Evals That Earn User Trust

Felipe Blanes

Most teams measure their agent against a benchmark, ship it, and hope. But when your agent serves real users, a benchmark won't tell you if it's actually working. This session is about building an eval suite that captures what success looks like in production, runs against real user workflows, and feeds back into product decisions. Here's the flywheel we use in practice: start with what success looks like from the user's perspective, instrument production workflows to capture those signals, diagnose where the agent falls short, and feed those insights into the next thing you build. You'll see how it shaped concrete product bets, turning eval results from a report card into a discovery tool.

SESSION **Expo Stage 3** 2:50pm-3:10pm *tentative*

Snyk Expo Session

SESSION **Expo Stage 4** 2:50pm-3:10pm *tentative*

The Software Factory

In the leading engineering organizations, a single engineer now supervises teams of agents, migrations scoped for years close in weeks, and code review has become the tightest constraint in the system. The teams pulling ahead are operating a software factory: an integrated system of agents that share context across the entire SDLC. In this session, Factory Co-Founder and CTO Eno Reyes offers a field guide to that operating model and how it runs at scale: what each stage looks like in practice, what shifts for engineers as they move from writing code to stewarding the system, and the hard truths that decide whether a factory compounds, starting with why the infrastructure you built for humans sets the ceiling on what agents can do.

3:20pm

SESSION **Harness Engineering · Main Stage** 3:20pm-3:40pm

We Solved Agent Building - The Evolution of Building A Successful Data Science Agent

Andrew Qu — Vercel

SESSION **Generative Media · Track 1** 3:20pm-3:40pm

Beyond Prompts: Building a Multi-Agent Creative Computer That Orchestrates 5+ AI Models in Real-Time

Brennan Erbz — CEO & Co-Founder, Flik

Flik is a production multi-agent system that generates complete work, not pieces. This talk demonstrates how the system orchestrates Claude, Gemini, Nano, Seedance, and Eleven Labs in a single workspace across text, image, video, and audio; shows an end-to-end workflow from prompt to finished output; explains coordination across modalities; and covers built-in likeness/IP safety plus real customer examples.

SPONSOR **Agentic Commerce · Track 2** 3:20pm-3:40pm

The End of the Static Screen: Architecting Intent-Driven UX with Agentic Orchestration

Gus Iwanaga — Product, UX, and Engineering lead for mosAIc, commercetools

For 30 years, interfaces were designed ahead: wireframes, fixed flows, pre-built dashboards - because we couldn't make them otherwise. Three shifts changed the constraint: LLMs that reason over business context, agentic frameworks that work at production grade, and composable backends that expose a real tool surface. With all three in place, the interface stops being something you design and ships as the output of an orchestrator composing it per intent. I'll walk through the hypothesis, the architecture we're running in production for enterprise commerce, and a live demo where it all moves.

SESSION **AI in Finance · Track 3** 3:20pm-3:40pm

It's a Skill Issue : Best practices in building skills that work

Yogendra Miraje — Lead Machine Learning Engineer, FactSet

SESSION **Agentic Engineering · Track 4** 3:20pm-3:40pm

Velocity Sickness: What Happens When Your Whole Team Gets 10x Faster

Matt Dailey — Ref.

Learn more about Ref: https://ref.tools/ AI made writing code nearly free, and on most teams, that's quietly breaking how the team works. Individually, everyone feels ten times faster. Together, the signals point the other way: too many PRs moving in too many directions, engineers throwing away whole agent sessions and starting over ("declaring agent bankruptcy"), and critical decisions getting made inside agent chats that no one will ever see or review. There's a lot of energy, and it's all going somewhere different. I call this velocity sickness: the organizational pain that comes from individual speed. It's the engineering version of an author who ships a book a week: prolific, productive, and completely unreadable by the team that's supposed to build on it. Almost every conversation about AI coding is about making one engineer faster. This talk is about what happens to the team when all of them are. Once implementation stops being the bottleneck, the hard part isn't writing the code. It's tracking it, reviewing it, and keeping a hundred parallel decisions coherent. That's the problem eng leaders are actually being handed, and it's the one this session takes on directly. Engineering has always had three phases: plan, implement, polish. AI collapsed the middle one to almost nothing, so the leverage, and the real work, move to the decision-heavy ends. The fix isn't better prompts; it's changing what our tools treat as first-class. We have to split the decision layer from the implementation layer: humans spend their time at the decision layer, reviewing and making the choices that matter, while agents handle the implementation. That means durable, reviewable plans, not ephemeral chats. Review the decisions before you review the diff.

SPONSOR **Graphs · Track 5** 3:20pm-3:40pm *tentative*

Why Agentic Systems Need Ontologies

Frank Coyle — Computer Science Educator; Founder, The AI Edge, University California Berkeley

Agentic systems fail in predictable ways: context degradation, brittle tool descriptions, fragile multi-agent handoffs, stop-reason confusion, and the ever-present temptation to fix reliability problems with more natural-language instructions. These anti-patterns aren't bugs to be patched turn by turn — they're symptoms of a missing architectural layer. LLMs reason probabilistically over domains they only partially understand, and no amount of prompt engineering fully closes that gap. This talk argues that the missing layer is an explicit ontology: a formal, shared map of the domain's concepts, relationships, and constraints. The pattern is not new — ontologies have driven commercial success in defense and intelligence systems for over a decade, where probabilistic models must operate over high-stakes enterprise data without drifting into nonsense. Graph databases like Neo4j and Amazon Neptune have made the underlying primitives widely accessible. We'll show how lightweight ontology constructs can surround an agentic system with enforceable logical constraints: typed entities and relationships that tools must respect, cardinality and domain restrictions that catch malformed tool calls before they execute, and a shared vocabulary that keeps coordinators and subagents talking about the same things. The session walks through several agentic applications — a multi-agent research workflow, a tool-heavy customer support agent, a coordinator-subagent delegation pattern — and shows in each case how an ontology layer addresses the kinds of anti-patterns catalogued in Anthropic's Claude Certified Architect exam. The result is a hybrid neurosymbolic architecture: probabilistic reasoning inside, logical guardrails outside. Who should attend: engineers building production agentic systems, architects evaluating reliability strategies beyond prompt engineering, and technical leads who suspect their agents need more structure than another system prompt can provide.

SESSION **AI in GTM · Track 6** 3:20pm-3:40pm

Fin.ai in GTM

Bil Erdenekhuyag — Intercom

SESSION **AI in Healthcare · Track 7** 3:20pm-3:40pm

Don't be data poor

Anuj Iravane — AI Lead, Anterior

What do you do when the data you most need to train and evaluate on is the data you're least allowed to keep? It's a bind for anyone building AI in a high-stakes vertical: the cases that would teach your model the most — the rare, the messy, the sensitive — tend to be the ones wrapped in the tightest constraints. In healthcare it's near-absolute. PHI can't be retained, reused, or transformed, so your long-lived datasets can't contain real patient data at all. Synthetic data is the obvious escape hatch, but it has its own trap: synthetic records tend to look synthetic, and a model that passes on fake-looking data tells you nothing about the real thing. So the bar isn't generating data — it's generating data faithful enough to trust. This talk is how we got there. Ask an LLM for a full case in one shot and you get something generic and averaged-out — models are worse at inventing convincing, specific detail than you'd expect. We present our synthetic generation pipeline (and the process around it) that enabled us to create golden datasets at scale. The pipeline features a coarse-to-fine process that enriches a patients medical history layer by layer, with a human in the loop hooks to steer the narrative at each step. You'll leave with ideas on how to build your own synthetic data generation capabilities and how to build a data pipeline your domain experts actually enjoy owning.

SESSION **SemiAnalysis · Track 8** 3:20pm-3:40pm *tentative*

To be announced

SESSION **Inference · Track 9** 3:20pm-3:40pm

Weight Folding, CUDA Streams, and the Bug That Made My Model Speak Backwards

Filip Makraduli — Machine Learning Engineer, Superlinked

SPONSOR **Track M** 3:20pm-3:40pm

M9

SESSION **AI-Native Enterprises · Leadership 1** 3:20pm-3:40pm *tentative*

To be announced

SESSION **AI Architects: AI Factories · Leadership 2** 3:20pm-3:40pm

Give the Agent a Budget, Not a Token

Sachin Malhotra — Platform Engineer, Anthropic

Every agent demo runs with a god-token. Then it ships, and someone has to explain why the helpful AI just rm -rf'd the staging database "to clean up." I run platform infrastructure at a frontier lab, and for the last year my job has partly been: let coding agents do real work against real systems, without ever having to write the postmortem. This talk is the permission model that fell out of that - not RBAC-with-extra-steps, but primitives designed for an actor that's smart, fast, tireless, and occasionally *confidently wrong*. **The four primitives:** - **Asymmetric verbs** - the agent can `quarantine` but not `delete`, `retry` but not `approve`, `propose` but not `merge`. The verb list *is* the security boundary. Stop thinking in resources, start thinking in reversible vs. irreversible actions. - **Regenerating budgets** - every agent identity gets N disruptive actions per window. Burn the budget, you're benched until it refills. No human-in-the-loop until the budget's gone — which means 95% autonomy with a hard ceiling on blast radius. - **The undo test** - if the agent can't undo it, the agent can't do it without a second key. One line, surprisingly load-bearing. - **Tripwires over allow-lists** - let the agent roam, but instrument the three actions that would actually hurt. Cheaper than enumerating everything safe. I'll show the ~200-line policy layer that implements all four, the failure modes each one exists to catch, and the one design I shipped that turned out to be security theater. Tool-agnostic - works whether your agent is touching CI, a database, a cloud account, or your users' files. If you're shipping an agent that does anything more than read, you'll leave with a threat model and a starting policy you can paste into your repo on the flight home.

SESSION **Expo Stage 1** 3:20pm-3:40pm *tentative*

Agent Memory Is a Solved Problem. Agent Learning Is Not.

Karthik Ranganathan · Heather Downing

The failures that break multi-agent systems are not reasoning failures, they are handoff failures. One agent works something out and the knowledge dies in its private context, because the only thing that crosses the boundary is output. Memory made each agent better in isolation and changed nothing about what the group knows. The missing primitive is supervised promotion: a deliberate decision about which private learning is worth sharing, moved into common knowledge with the reasoning attached, so trust survives the handoff. Today a human makes that call, and promoted knowledge resolves on read, in any tool, with no retrain or reindex. Those calls are also the training signal for what comes next: orchestrator agents, trained on what matters to the people they serve, that promote on their own. This talk covers how our collective knowledge grew as we approached memory promotion, including what the first build got wrong, and a live look at it working between humans and agents.

SESSION **Expo Stage 2** 3:20pm-3:40pm *tentative*

Deepmind Expo Session 2

SESSION **Expo Stage 3** 3:20pm-3:40pm *tentative*

Orkes Expo Session

SESSION **Expo Stage 4** 3:20pm-3:40pm *tentative*

An AI Future Without the Lock-In

Remy Guercio

Every organization navigating AI adoption faces the same trap: the market moves faster than any procurement cycle, no single vendor leads across model quality, interface, sandbox, and data access for more than a few months at a time, and the obvious answer of consolidating behind one platform trades short-term control for long-term lock-in. This session makes the case that the winning strategy is not picking the best walled garden. It is building a connective layer underneath all of them. Tailscale's Remy Guercio walks through the four components required for transformative AI, why vertically integrated stacks are structurally fragile, and how organizations can maintain visibility and control without betting on a single vendor's continued dominance. The second half of the session covers three new capabilities in Aperture, Tailscale's identity-aware AI gateway: Identity-Aware Universal Data Connectors (Public Alpha), which translate Tailscale network identity into scoped access to internal data sources via MCP and API endpoints; a Responsive Chat UI (Public Alpha) that gives non-technical users a mobile-friendly interface to every LLM configured in Aperture; and Sandbox Support (Private Alpha), bringing ephemeral and persistent compute environments into the same identity model. Attendees leave with a framework for evaluating AI platforms that does not depend on picking a winner, and a concrete path to deploying provider-agnostic AI tooling on infrastructure they already run.

3:45pm

SESSION **Harness Engineering · Main Stage** 3:45pm-4:05pm

Agents Without Code: How Skills, YAML, and Filesystems Replaced Python

Philipp Schmid — Staff Engineer, Google DeepMind

Six months ago, building an agent meant writing a Python class with a `while` loop, tool definitions in dicts, manual state management or writing custom python functions. Today, you define an agent in a YAML file, drop a `SKILL.md` into a folder, and deploy. This talk traces the arc from "Agent in Python" to "Agent as filesystem". You'll learn the same agent built three ways: the hard way (Jan 2025), the simple way (Oct 2025), and the zero-code way (today).

SESSION **Generative Media · Track 1** 3:45pm-4:05pm

The Next Game Engine Won't Have a Manual

Arturo Nereu — AI & Game Development, MongoDB

Game development needs to change for the agent era rather than simply dropping an LLM into existing engines. This talk shows the AI systems behind Veselka, using Claude plus Three.js to turn AI into a practical game-development partner and lower the barrier for people who want to build their dream game.

SPONSOR **Agentic Commerce · Track 2** 3:45pm-4:05pm

Beyond the Lethal Trifecta: Agentic Commerce on the Open Internet at Machine Speed

David Levine — Founder & CEO, Kiduna Club

For decades, the internet has had protocols for routing, identity, encryption, payments, and commerce between people and organizations. It has never had a native way for autonomous agents to possess authority, accountability, or legal standing. On July 1, 2026 that changes. A little known law will take effect that changes the world as we know it. As AI agents move beyond the enterprise firewall, a new form of commerce is emerging. Agents can already search, negotiate, schedule, purchase, settle payments, and coordinate work across networks. But the moment they begin acting independently on behalf of people, businesses, and online organizations, fundamental questions appear: Who does this agent represent? What authority does it possess? Who is responsible when something goes wrong? How do counterparties know they can trust it? This talk explores the "Lethal Trifecta" of agentic systems: access to systems, access to networks, and autonomy. Together they create extraordinary capabilities, but they also expose a missing layer in the architecture of the internet itself. Without identity, accountability, governance, and legal standing, agentic commerce remains trapped inside enterprise walls, limited to productivity gains rather than participation in open markets. On the same day as this conference, a new legal framework takes effect that gives autonomous online organizations a registered legal existence, allowing them to hold assets, enter agreements, govern themselves through software, and operate through fleets of agents. Whether you're building agents, agent platforms, autonomous organizations, payment systems, governance systems, or the next generation of internet infrastructure, this shift has global implications, and you'll be the first to know. We'll examine the emerging trust stack for agentic commerce—identity, authority, governance, settlement, and standing—and explore what happens when agents stop acting merely as tools and begin participating as economic actors on the open internet at machine speed.

SESSION **AI in Finance · Track 3** 3:45pm-4:05pm

Wearing the Agent: Engineering a Family-and-Friends Personal Agent, from Group Chats to Glasses

Sai Krishna Rallabandi — Director, Data Science, Fidelity Investments

Judith is a personal AI agent that has run in daily production for a year, used by more than a dozen family and friends across WhatsApp group chats, Telegram, and Discord. This talk covers the engineering for a safe multi-tenant personal agent: permissioning, long-lived memory across FAISS + Neo4j + curated notes, scheduled subagents, and message-time guardrails for privacy, recipient safety, and prompt-injection defense. It then shows how the agent moves onto low-cost smart glasses, capturing visual memory, helping with navigation and in-store tasks, and maintaining conversational latency with on-device speech recognition, cloud reasoning, and a custom neural voice. Includes live demos plus practical takeaways on multi-user agent design, durable memory, defensive agent engineering, and wearable ambient interfaces.

SESSION **Agentic Engineering · Track 4** 3:45pm-4:05pm

Making agents you could never make

Ara Khan — Founder, Cline

SPONSOR **Graphs · Track 5** 3:45pm-4:05pm *tentative*

Why We Killed Our Multi-Agent Pipeline: Lessons From Pharma Commercial Intelligence

Subbiah Sethuraman · Abhilash Asokan

Key takeaways: A practical design principle for agentic systems in regulated, high-stakes domains: derive the architecture from agent behavior, don't impose it. Concrete patterns the audience can apply this week — domain knowledge graphs as agent context, deterministic preprocessing as a complement to agentic reasoning, reference-based context management. An honest case study from production: what worked, what didn't, and the open architectural questions we're still working on. Abstract : We lead the architecture and AI engineering org behind ZS Associates' commercial intelligence platform for pharmaceutical brand teams. The product has two surfaces: a proactive alert system that delivers signal-driven intelligence packets when a brand's KPIs move, and a conversational analytics chat where business users ask ad-hoc questions. A year ago we built both surfaces as separate V1 stacks. They broke in different ways. The diagnosis was the same: we had decided on the structure before we knew what the agent actually needed. This talk is about the design principle that came out of rebuilding both — and what it produced. The architecture is derived, not designed. We stopped trying to predict what scaffolding the agent would need and started designing the system around what the agent's behavior, on real production tasks, actually demanded. Tools, context, structure, and guardrails get introduced at the points where the agent's reasoning needs them — and nowhere else. What that produced is an architecture that's smaller than V1, not bigger. A single agent owns each investigation end-to-end across both surfaces, launching parallel sub-agents when the work needs them — not according to a pre-defined topology. A pharmaceutical commercial knowledge graph — HCPs, accounts, payers, territories, brands, KPIs and the relationships between them — gives the agent the domain context it needs without prompt-engineering heroics. Statistical signal detection runs deterministically before the agent wakes up, so the agent's job is to explain signals, not find them. Raw query results stay out of the context window through a reference-pattern that lets the agent reason over data without drowning in it. Each of those decisions came from watching an agent struggle on a real task and asking what does it need here? — not from sketching the architecture in a doc and forcing the agent into it. The patterns generalize. If you're shipping agents over messy enterprise data — finance, supply chain, claims, operations — the failure modes and the fixes will look familiar. We'll close with the open questions and the pieces we haven't solved yet.

SESSION **AI in GTM · Track 6** 3:45pm-4:05pm

Reverse-Engineering the AI Buyer

Aliisa Rosenthal — General Partner, Acrew Capital

SESSION **AI in Healthcare · Track 7** 3:45pm-4:05pm

AI Benchmarks for Vertical Industries: Why we're not measuring what we need to and how to unlock real-world ROI

Christopher Lovejoy — Member of Technical Staff, Anthropic

AI is acing the tests we set for it. So why are so many production deployments falling flat? This talk draws on lessons from building Anterior's internal benchmark for real-world healthcare tasks and how to translate real-world performance into concrete measurement rubrics, use imperfect synthetic data, avoid common pitfalls, and apply the approach to any vertical domain.

SESSION **SemiAnalysis · Track 8** 3:45pm-4:05pm *tentative*

To be announced

SESSION **Inference · Track 9** 3:45pm-4:05pm

The Frontier AI Inference Cloud for Agents

Byung-Gon (Gon) Chun — Founder & CEO, FriendliAI

Agents have changed the economics of AI inference. A chatbot's cost scales roughly linearly with the number of requests; an agent's scales multiplicatively. A single task can fan out into hundreds of model calls, each carrying a repeated context prefix and adding latency that compounds across tool calls and reasoning steps. As open-weight models keep improving and agentic workloads grow, this shift exposes the limits of traditional request-level optimization. Inference infrastructure becomes a first-class concern, one that often shapes performance and cost as much as the model itself. In this talk, we explore what changes when you optimize for the whole task rather than the individual request, and how FriendliAI is rethinking the inference cloud for the era of agentic AI. Description: Agents' inference costs scale multiplicatively. Optimize the whole task, not the request. See how FriendliAI builds the inference cloud for agents.

SPONSOR **Track M** 3:45pm-4:05pm

M10

SESSION **AI Architects: AI Factories · Leadership 1** 3:45pm-4:05pm

The Signal Layer: What to Build When Anything Can Be Built

Lena Hall

AI has made implementation faster, cheaper, and more widely available. That changes the real bottleneck in software. When every team can generate code, spin up agents, prototype workflows, and ship demos faster than ever, the advantage moves to a different layer: knowing what is worth building, who it is for, how people will discover it, and how the product should behave once they do. This talk introduces the Signal Layer: the system of public signals, user intent, agent experience, distribution loops, and product judgment that helps builders decide what deserves to exist before they commit time, infrastructure, and trust to building it. We will look at how AI changes the software lifecycle from "can we build it?" to "should this exist?" and how developers, AI engineers, and technical leaders can design products that earn adoption instead of producing impressive demos that disappear. When anything can be built, the most valuable builders are the ones who can read signal early, shape the right experience, and build the thing users were already moving toward.

SESSION **Harness Engineering · Leadership 2** 3:45pm-4:05pm *tentative*

Agent Frameworks Considered Harmful

Rémi Louf — .txt

SESSION **Expo Stage 1** 3:45pm-4:05pm *tentative*

Modal Add-On Expo Session

SESSION **Expo Stage 2** 3:45pm-4:05pm *tentative*

Microsoft Presenting Expo Session 3

Microsoft — Microsoft

SESSION **Expo Stage 3** 3:45pm-4:05pm *tentative*

Livekit Add-On Expo Session

SESSION **Expo Stage 4** 3:45pm-4:05pm *tentative*

AI agents don't read your policy docs. They hit your APIs.

Every organisation has a policy for what AI should and shouldn't do. But in the era of autonomous agents, who is that document actually for? Odds are no agent has ever read it. It opens a connection and makes a call, and whatever happens at that millisecond is your real policy. So put the control there. This talk is about the gateway as the runtime where AI governance actually executes: per-agent identity and scoped, short-lived credentials instead of a shared god-key. PII and secrets stripped from prompts in flight. Token-aware rate limits so one looping agent can't torch your quota. Semantic caching that cuts spend and latency on requests you've already answered. I'll share the architectural patterns behind each control, what they look like in practice, and what breaks the moment you take them away. Policy states intent. Infrastructure enforces it.

4:30pm

KEYNOTE **Main Stage** 4:30pm-4:50pm

Closing Keynote — Theo Browne

Theo Browne — t3.gg

4:50pm

KEYNOTE **Main Stage** 4:50pm-5:10pm

Closing Keynote: Garry Tan

Garry Tan

5:10pm

KEYNOTE **Main Stage** 5:10pm-5:30pm

Startup Battlefield

TBD

Schedule v2475 · exported June 17, 2026. Tap any talk title to open it on the live interactive schedule. Schedule is subject to change; sessions marked “tentative” are not yet confirmed.