

Agentic AI Infrastructure

Company perspectives from the 2026 Berkeley summit

Presented by Fatemeh Nourilenjan Nokabadi

14/09/2026



Content

Overview

Amazon / AWS: Constraint-driven systems

NVIDIA: Accelerated computing as an AI factory

Google / DeepMind: From models to the full stack

IBM: Enterprise control plane beyond the harness

Microsoft: Agent runtime to Zero ops

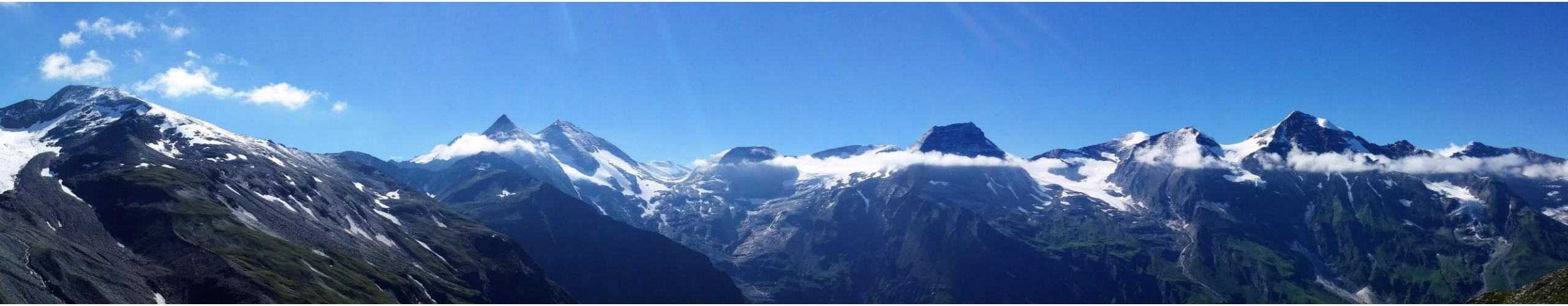
OpenAI: Long-running agent intelligence and runtime

Anthropic: Safe tool use and computer-using agents

Open inference serving: Inferact vLLM and RadixArk SGLang

Postman: Reliable multi-agent infrastructure

Source guide and videos



Center for Responsible Decentralized Intelligence (RDI)

UC Berkeley

1 → 2 → 3

LLM Agents 2024

- The initiative explored how large language models could reason, plan, use tools, retain memory, and collaborate.

Agentic AI Summit 2025

- The Agentic AI Summit 2025 expanded the conversation toward practical agent frameworks, applications, evaluation, and deployment.

Agentic AI Summit 2026

- Agentic AI Summit 2026 focused on the infrastructure, evaluation, safety, and deployment practices that enable reliable agentic AI at scale.



Agentic AI Summit 2026

Meta-Themes

Intelligence → Agency

How foundation models evolve into systems that can reason, plan, and act.

Automation → Autonomous Organizations

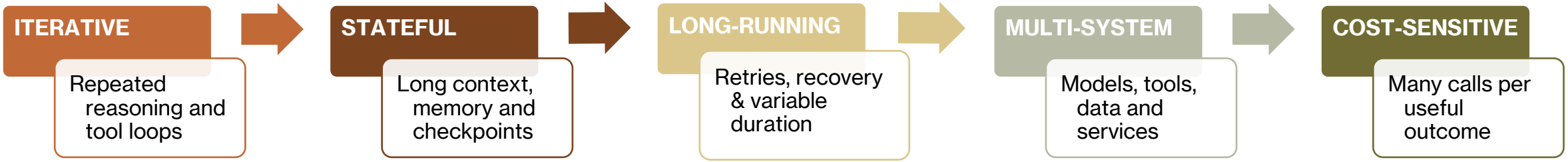
How enterprises restructure around continuously operating AI agents.

Prototype → Production

How we build the infrastructure, architecture, evaluation, reliability, and scalability necessary for agents to work in the real world.

Why do agents force an infrastructure reset?

- The workload is no longer a single stateless model request!



Infrastructure optimization moves from “fast model call” to “**reliable, economical agent outcome”**.

We must ask not only how many tokens per second the model generates, but whether the complete task succeeds, how much it costs, how safely it operates, and whether it can recover from failure.

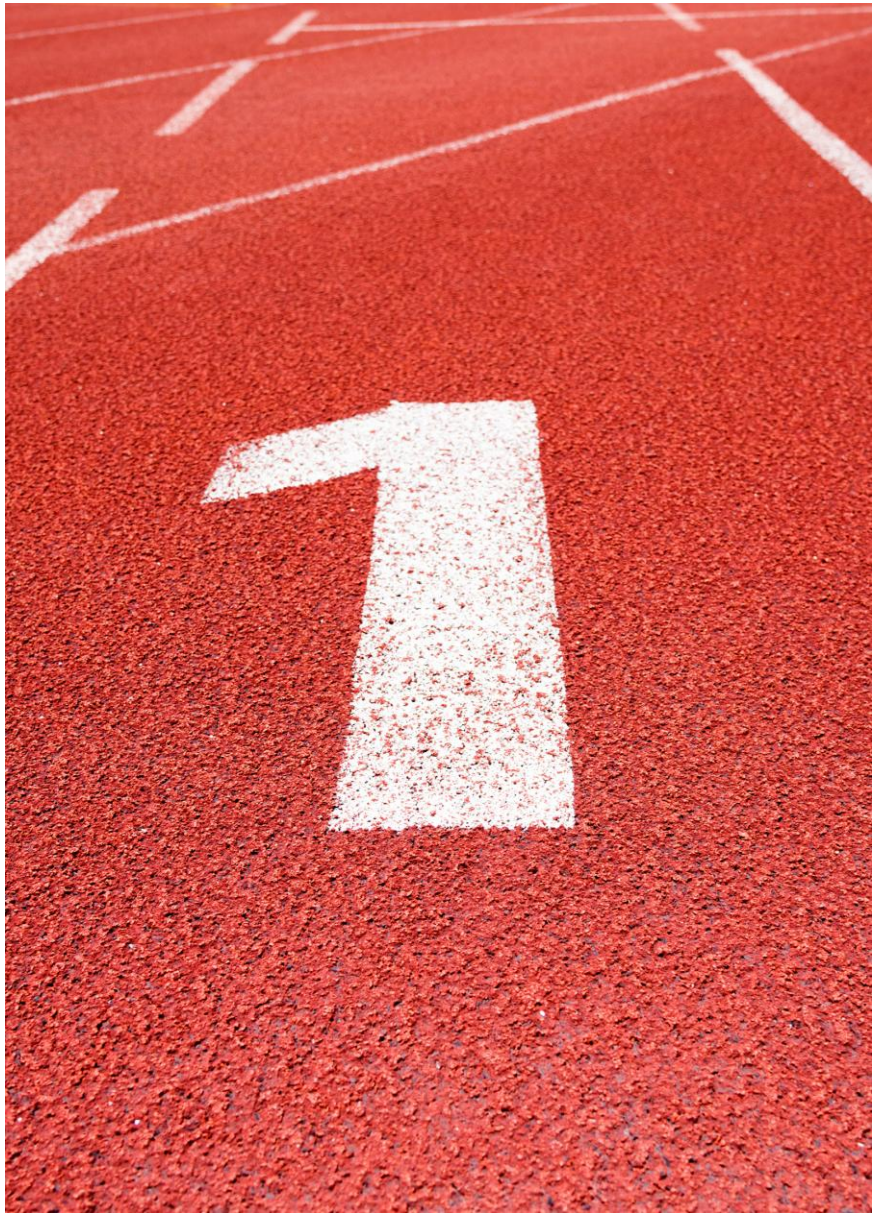
Amazon / AWS | Constraint-driven systems

- Moving from exclusive reliance on general-purpose GPUs toward workload-specific accelerators such as **Trainium** and **Inferentia**.

- Trainium uses a systolic-array architecture to efficiently execute the predictable matrix operations common in AI workloads.

- Specialization reduces unnecessary flexibility and data movement, improving performance, power efficiency, and cost at scale.

- "**Day 1 in AI**": AI infrastructure is still evolving, and future workloads will require a diverse ecosystem of specialized hardware.



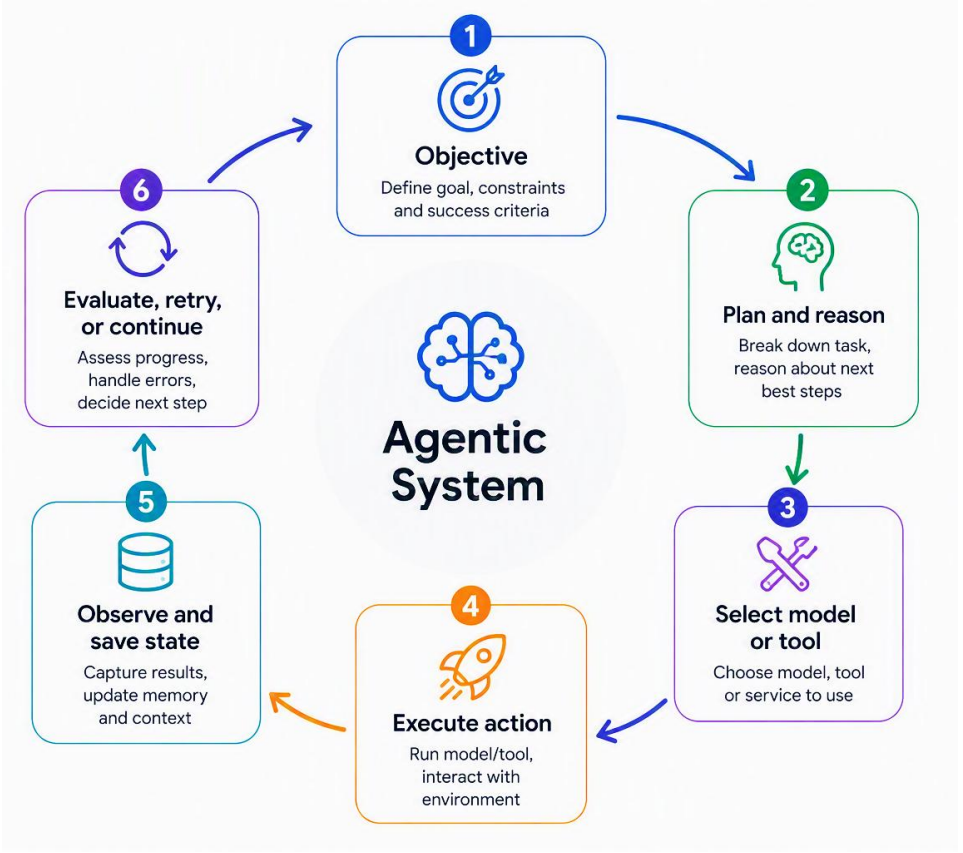
NVIDIA | Accelerated computing as an AI factory



- An AI factory is infrastructure designed to continuously transform **data** and **energy** into **intelligence**, with tokens representing its primary production output.
- Unlike a conventional data centre, an AI factory is optimized specifically for training models and running large-scale, always-on inference.
- Agentic AI increases the workload because agents repeatedly reason, plan, retrieve information, use tools, and coordinate with other agents.
- The factory integrates GPUs, high-speed networking, memory, storage, AI software, power, and cooling as one coordinated production system.
- Its performance depends on the efficiency of the entire workflow, not simply the speed of an individual GPU.
- Success is measured through metrics such as **tokens per second**, **tokens per watt**, **cost per token**, **utilization**, and **uptime**.

Google / DeepMind | From models to the full agent stack

- Agentic AI expands the system from a standalone model to a complete stack connecting **models, agents, tools, data, and environments.**
- Models provide intelligence, while the **agent layer** manages planning, tool selection, execution, and feedback.
- Agents require infrastructure for persistent state, memory, context, and intermediate results across multiple steps.
- Agent workloads may be **long-running** and **dynamic**, requiring scheduling, recovery, retries, and changing compute resources.
- Traditional orchestration platforms such as Kubernetes were designed largely around services and containers, so they must adapt to **agent-shaped workloads**.
- The full stack should support the entire agent lifecycle, from reasoning and execution to **monitoring, governance,** and **discovery.**



IBM | Enterprise control plane beyond the agent harness

- An agent harness manages the agent loop, including **model calls**, **tool use**, **planning**, and **execution**.

- Enterprise deployment requires a broader platform layer beneath the harness to **manage agents in production**.

- The platform must address the agents' variable outcomes, autonomy, new failure modes, and security risks.

- Agents require persistent state management to **preserve context, results,** and **progress** across multiple steps.

- Platform services should provide identity, access control, isolation, monitoring, recovery, and resource management.

- The objective is to move agents from controlled demonstrations to systems that operate **reliably**, **securely**, and **efficiently** at enterprise scale.



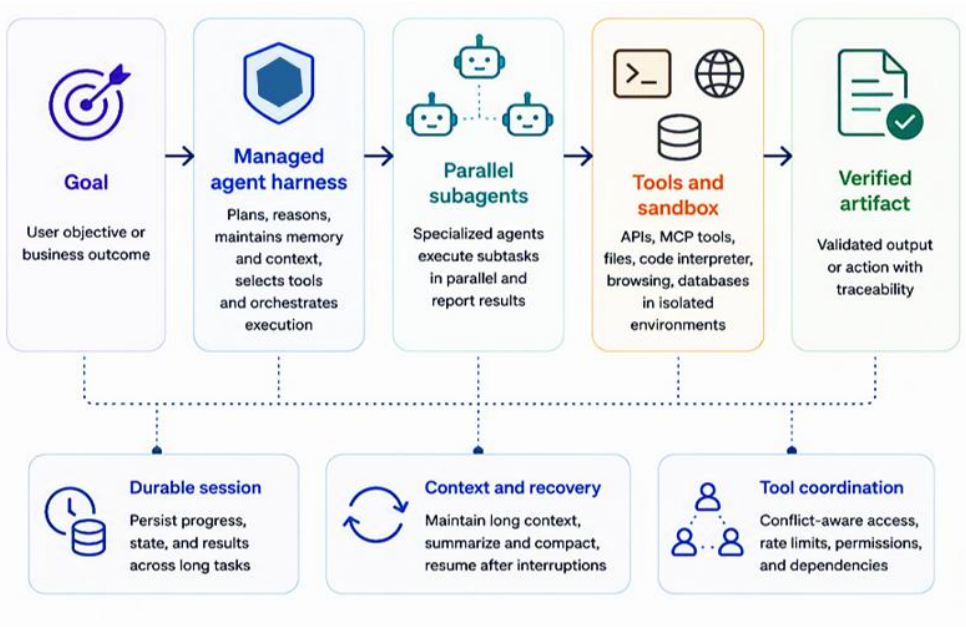
Microsoft | Agent runtime to Zero Ops

- **Continuous improvement**: Production feedback and evaluation create an ongoing model-improvement loop.
- **Internalized harnesses**: Planning, memory, tool use, and feedback may become more deeply integrated into the model-system architecture.
- **Agentic operations**: Agents can monitor, diagnose, and remediate infrastructure issues.
- **Zero Ops**: The operational model shifts from humans operating systems to agents operating under human governance.
- **Production controls**: Identity, permissions, observability, approvals, rollback, and auditability constrain agent actions.
- **Human responsibility** remains: People define intent, policy boundaries, and escalation conditions.



OpenAI | Long-running agent intelligence and runtime

- Frontier models provide the reasoning and planning capability required for long-horizon agents.
- A managed **agent harness** maintains context, tools, files, execution environments, and task progress.
- Multi-agent execution allows a primary agent to delegate **specialized subtasks to parallel subagents**.
- **Durable sessions**, **context compaction**, and **recovery support** agents that operate for hours or days.
- Codex demonstrates agentic execution through coding, testing, debugging, and software-engineering workflows.
- The infrastructure focus shifts from individual model responses to **reliable completion of extended tasks**.



Anthropic | Safe tool use and computer-using agents



- Claude combines **long-context reasoning** with tool use and agentic coding workflows.
- Claude Code can inspect codebases, edit files, execute commands, and coordinate parallel subagents.
- Computer-use capabilities allow agents to interact with browsers, desktop applications, and development tools.
- Skills, scheduled routines, and connectors extend agents into persistent enterprise workflows.
- Sandboxing, permissions, prompt-injection controls, and human escalation constrain high-risk actions.
- Anthropic’s distinctive priority is capable agent execution with safety and controllability built into the runtime.

Open inference serving | Inferact vLLM and RadixArk SGLang

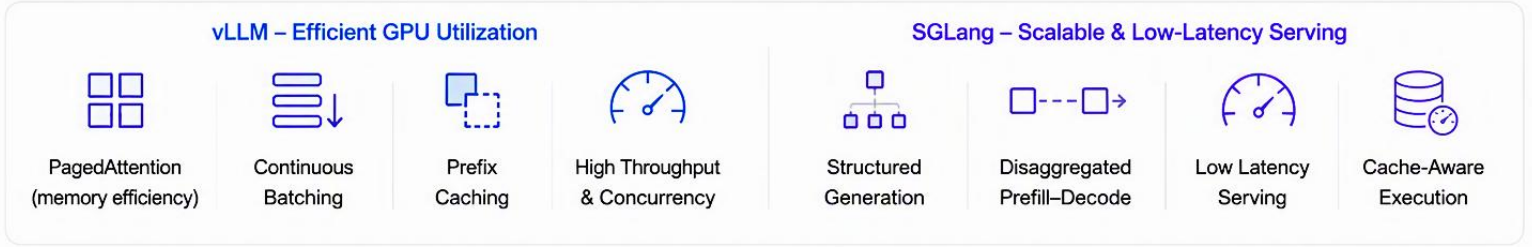
- **vLLM** uses PagedAttention, prefix caching, and continuous batching to improve GPU-memory utilization and inference throughput. Its parallelism strategies distribute models, requests, experts, and long contexts across GPUs and nodes.

- KV-cache transfer and offloading separate prefill from decoding, extend cache capacity beyond GPU memory, and reduce recomputation.

- **SGLang** supports high-throughput, low-latency serving through RadixAttention, structured generation, optimized scheduling, and speculative decoding.

- Both frameworks support distributed inference techniques such as parallel execution and disaggregated prefill and decoding.

- Together, these open-source systems enable longer contexts, higher concurrency, lower latency, and lower cost per agent task.



Postman | Reliable multi-agent infrastructure

- **Context**: Agents need accurate, current, and structured information to reason reliably.
- **Tools**: Agents require discoverable, well-defined, and governed APIs to interact with real systems.
- **Coordination**: Multi-agent workflows need clear roles, controlled handoffs, shared state, and failure handling.
- **Production approach**: Combine agent-ready context, governed APIs, and observable workflows.
- **Key principle**: Adding more agents without strong foundations can multiply complexity and operational risk.



Source guide and videos

Primary research base and presentation links

OPEN OFFICIAL PROGRAM

OFFICIAL PROGRAM

Berkeley RDI Agentic AI Summit 2026 schedule, session descriptions and linked recordings.

OPEN YOUTUBE PLAYLIST

FULL VIDEO COLLECTION

Berkeley RDI playlist with plenary, track and individual talk recordings.

OPEN YOUTUBE PLAYLIST

Agentic AI Infra

Personal playlist of collective talks related to the Agentic AI Infrastructure in Agentic AI Summit 2026





Q&A