AKKA

Webinar Q&A: A Blueprint for Agentic AI Services

6 March 2025

Q: Will agnetic AI services replace SAAS in the future?

A: Answer provided in live Q&A

Q: Could this be used for the low-latency internal factored cognition/chain of thought, or are you focused on user-user and user-agent interaction?

A: The challenges with CoT etc type applications are very similar to user-agent interaction — and having business logic to help guide CoT can be a very powerful technique to improve accuracy. (So, in general, the approaches we're discussing apply equally to user-agent as well as CoT-type use cases.)

Q: It can't be 100% for a fundamental reason. Not all use cases accept stochastic outcomes.

A: Answer provided in live Q&A

Q: What is the cost model for operating Agentic AI services? In one of the slides, you mention a TPS in millions while having a near 1-1 ratio between users and AI Agents. Will this lead to prohibitive operating costs?

A: Running LLMs (whether you use OpenAI or host internally) is a new operating expense. The key to managing this expense is to be efficient about using LLMs, pre-processing the information, calling the right size of LLM for a given query, balancing between different LLMs that have perf: cost profiles, and managing rate limits are some techniques in an agentic platform that help optimize efficiency. You also need an agentic compute platform that is event-driven and non-blocking to maximize LLM utilization and avoid wasting blocking memory during LLM calls.

Q: Why wouldn't I ensemble SLM 'mixture of experts' to overcome LLM limitations?

A: Answer provided in live Q&A

Q: Routing across multiple LLMs may be unavoidable for production use because individual LLM vendors have outages.

A: Yes, most situations are multi-model - ranging across performance, functionality, accuracy, cost, and resilience.

Q: SLMs should - in theory - have lower latencies and fewer 'hallucinations' . I wonder if SLMs are more compatible technically with agentic flows.

A: Answer provided in live Q&A

Q: Wow! So this is all about latency management?

A: Capacity planning and performance tuning go hand-in-hand. An optimized agentic platform maximizes memory and compute efficiency through high concurrency and non-blocking execution. This improves resource utilization, making vector databases, context databases, and model invocations more efficient. The more efficient the agentic platform, the lower the latency and model inference costs.

Q: One of the really important questions working with LLM is the prompt caching, and integrating this with the "Agentic Tier" is *the* highest reduction in latency you can get in these GPU compute-dominated LLM executions.

A: Yes, an in-memory context database for keeping the prompt and its history is the fastest and most performant way to achieve this.

Q: Pardon my naivety, but if the margin of error is 1% in each LLM data set, how does that affect an agentic system where agents are relying on each other? Does this increase the margin of error?

A: In practice, agentic systems do not rely exclusively on LLMs and are not as stochastic as they may seem. Here is a simple example: an LLM may be asked to interpret a natural language conversation, and based on the query, choose the most appropriate API call to execute based on the conversation. In other words, there may not be a path to eliminating all potential errors within an agentic system. An agent that depends upon multiple models, each with a margin of error, increases the overall potential error rate.

Q: What is the factor for agentic AI pricing going down? Is it in LLM network design change?

A: So far, improvements in LLM efficiency have been the main factor. In the future, leveraging compute beyond GPUs will be critical for achieving generational leaps in price-to-performance efficiency.

Q: I really like the thoughtful and intentional design approach here. Let me ask a trick question: Do you think of this as a 'stack' or an 'ecosystem'? Is it more about 'infrastructure' or—forgive me—a 'platform'?

A: Answer provided in live Q&A

Q: Does your agent platform allow LLMS to generate and run Python code? If so, how do you protect your platform from prompt injection attacks?

A: If you created an agent that was using an LLM to generate Python code, then that generated Python code would need a "target" platform to execute within. This would be external to the Akka environment and need to be specific to an environment for executing foreign Python code securely.

Q: Would you say "Agents" are Akka "Actors"?

A: I think the most correct definition is that the agent is a workflow loop. It's the iterative, orchestrated cycle. However, each step in the workflow needs to be a concurrent service. We just so happen to implement all of those steps as actors under the covers, so yeah, an agent is a composition of many actors coordinated and coordinating together.

Q: Response times and latency are not important for all kinds of applications and this would become a concern only after a certain scale. What kind of application areas are you seeing Enterprises being highly concerned about latency and seeing it as a major challenge

A: Latency is critical in any system where digital or data experience is critical: gaming, retail, media, fintech, mobile, edge, IOT --- anywhere there is a large number of concurrent users, devices, or real-time streams of data. Latency also becomes critical when you are trying to lower the cost of compute by ensuring that a system is as efficient as possible with consumption. Compute systems that are higher utilization make more efficient use of their downstream dependencies including databases, storage, and integration systems.

Q: Regarding life-cycle management, do you consider the evolution of the users' knowledge and understanding of the AI capabilities as part of the life-cycle? In other words, how do you account for the users getting smarter?

A: Answer provided in live Q&A

Q: How does the new AI Akka stack fit into the Business Source License (BSL) v1.1 usage?

A: Answer provided in live Q&A

Q: Within a SCADA system the ETL processes need to be real-time and non-disruptive. Would Akka agents be conducive to this while allowing for real-time monitoring of various communication protocols such as I2C, SPI, PCIe, UART, and GPIO?

A: Akka takes an ETL process and breaks it into a pipeline of non-blocking, and in some cases parallel execution tasks. The same extends into monitoring of various communication protocols such that the pipeline and monitoring are all executed in efficient, non-blocking manners, and then structured to make the appropriate batch or streaming invocations to the models that are part of the agent.

Q: Are you going to be adding Akkademy agentic topics?

A: Yes. We do not have a specific timeline, yet. We'd love to hear from the community about what would be most helpful.

Q: So, if we currently start with Bedrock for model catalog, orchestration, deployment, and then workflows, I can imagine the reference architecture for integrating Akka, but curious if you have one or few patterns. Thanks!

A: Given the specifics here, it's best if we follow up in an email discussion.

Q: Are we also trying to build some orchestrator like Lang-Chain/Lang_Graph within the Akka ecosystem, or are we saying to use the available orchestrator from around the world and stream it for low latency by using Akka?

A: Akka has its own orchestration. You write orchestration workflows as part of your service in a simple DSL. Your orchestration logic is compiled and packed into the services that Akka then runs for you. You are effectively building a distributed system that behaves as the agent for you. See the Akka SDK's workflow component.

Q: Managing GPU clusters for any self-hosted/self-trained models is definitely a learning curve and infrastructure ramp-up.

A: No kidding!

Q: How do you manage the human-in-the-loop and user-centric AI approach?

A: The Akka orchestration engine is governed by workflow which is defined in various stages and steps. Those steps can require human involvement or interaction before continuing. We broker the feedback from humans through messaging, APIs, and browser interfaces.

Q: Achieving strict adherence to a predefined response format from an LLM is inherently difficult. Even with structured prompts, explicit JSON schemas, and deterministic parameters (temperature = 0), models may introduce inconsistencies in property ordering, formatting, or interpretation. While enforcing constraints improves reliability, absolute determinism is challenging due to the model's probabilistic nature and evolving response generation behavior. What do you think?

A: If you're not familiar with libraries such as Outlines or OpenAI's Structured Outputs, take a look. They can guarantee that outputs conform to an explicit schema, and they work very well.

Q: Do you see the agents querying the business database directly anytime soon? Or is it going to be through existing APIs for the most part?

A: It'll be both, but it will take a while for agents to get to that level of control. While not suitable for agents, the MCP from Anthropic is starting to show how user-based systems that are using tools (like in your browser) can instruct LLMs about how to query or interact with other systems. MCP is intended for user space and not agentic (backend systems) space.

Q: Is it ok to say or look into Akka as a "developer platform" for agentic AI development/deployment? - like a backstage

A: We are a developer platform that can be a part of an Internal Developer Platform. IDPs are governance and sequencing systems that rely upon a runtime orchestrator like Akka.

Q: You guys really know your stuff. Thanks for sharing. You have given me a better idea of what I don't know, but should.

A: Glad you liked it. Stay tuned to our newsletter for more information.

Q: Will you be able to share any docs w.r.t Akka Aggentic AI to read further and build something on my own? When I search for Akka 3 on Google I am not able to find anything.

A: Yes. Coming soon!