AKKA

Webinar

Creating Certainty in Age of Agentic AI

Build, run, and evaluate with certainty, scale and resilience.

Tyler Jewell, CEO
Alan Klikic, Principal Solution Architect

Today’s agenda

AKKA

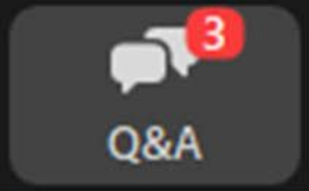
01 Welcome and introduction
Darin Bartik, CMO

02 The Akka Agentic Platform
Tyler Jewell, CEO

03 Demo: build and deploy multi-agent system
Alan Klikic, Principal Solution Architect

04 Live Q&A
All

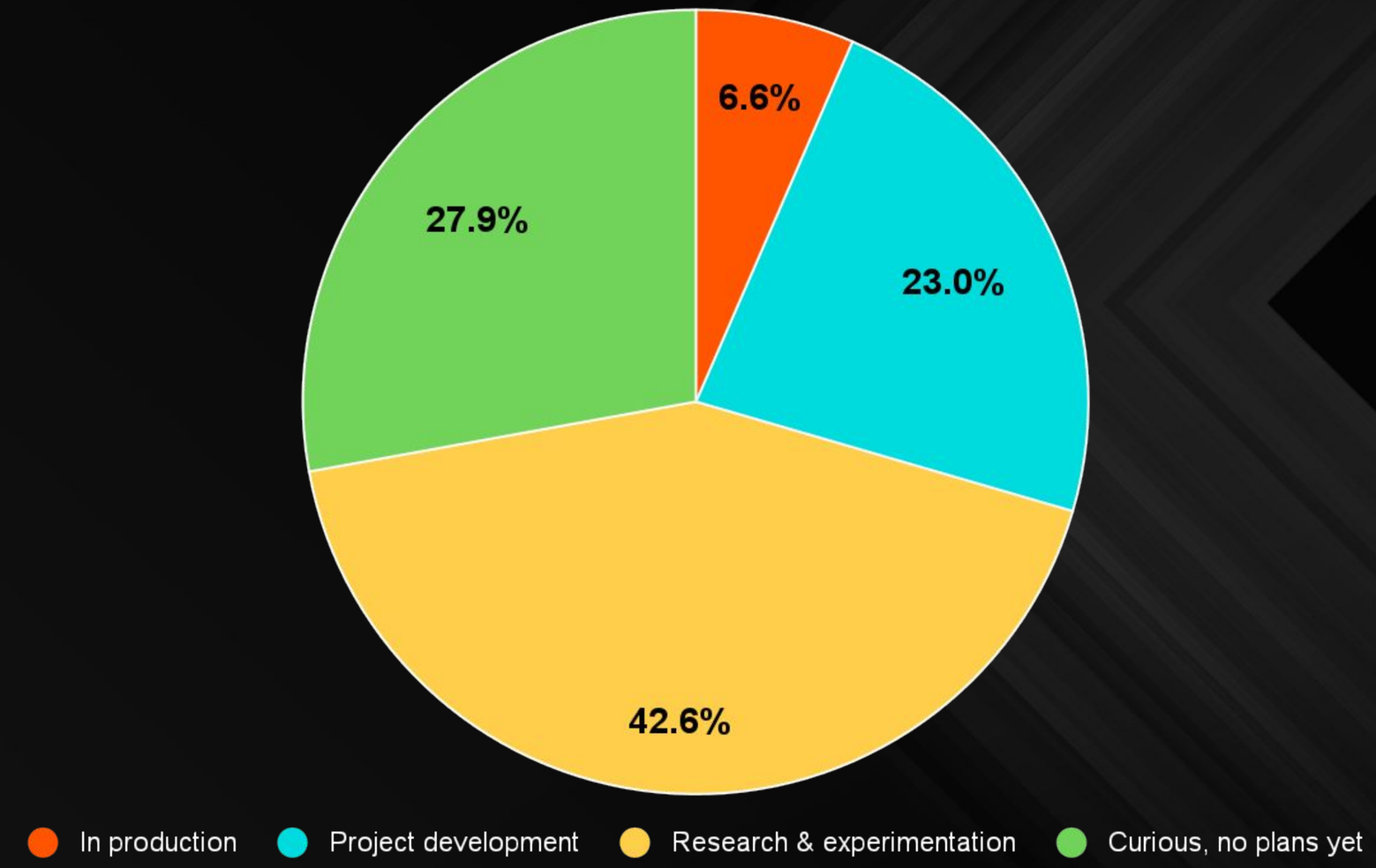
05 Next steps
Darin Bartik, CMO



Use the Q&A, not the chat function for questions

Pollinq question results:

Where are you in your agentic journey?



The market is unlocking at stunning speed

AKKA

Increased agency, innovation, and impact as orgs evolve from agents to agentic systems

Agents

for business-driven automation

e.g., data entry, chatbots, batch processing, integration, process automation

Agentic Systems

for intelligent automation

e.g., collaborative content creation, analysis and processing pipelines, analytics & transactions

Autonomous AI

for role augmentation

e.g., autonomous financial advisors, AI-driven R&D assistants, self-healing IT infrastructure

Low code systems

rapid prototyping,
democratizing agent creation,
well-defined task automation

mendix

boomi

+ every low code
+ every iPaaS
+ every ERP SaaS
+ every RPA
+ every BPM
+ every data provider

Pro code platform

→ complex reasoning
→ performance-critical systems
→ highly customized behavior
→ regulatory protocols
→ edge & real-time
→ digital twins

AKKA

Challenges we observe with Agentic

Two emerging problems: infra + token costs & DevEx productivity

AKKA

Agentic systems get expensive, quickly

➔ 1K TPS token costs:
- $.20 - $20 / second
- $6M - $600M / year

➔ 1K TPS agent infra costs:
- $40K / year
- Orchestration
- Memory
- Agent execution
- API serving
- AI gateways

Agentic DevEx is multi-dimensional

➔ Multi-agent systems are distributed systems:
- Coordination
- Shared state
- Evaluation
- Adaptation

➔ New dimensions of expertise beyond dev:
- Context engineering
- Systems engineering
- MLOps

Akka's approach to agentic

AKKA

1. Leverage Akka's actor concurrency and async event model to minimize token and infra costs.

➔ **Reactive.** Non-blocking, asynchronous, event-driven interfaces to LLMs, semantic stores, tool calls, and memory.

➔ **Inline intelligence.** Evaluation, context mgmt, goal-targeting, summarization, cost::perf decisioning, dynamic prompting, MCP invocations, and exceptioning through "effects".

2. Maximize multi-team productivity with a structured SDK designed to be accelerated by AI-assist.

➔ **Systems.** Single abstraction for programming agents, long-running orchestration, human-in-the-loop interactions, memory, and streaming.

➔ **Implicit memory.** Durable execution of workflows and session memory intrinsic to agents.

Akka Agentic Platform

Build, run, and evaluate with certainty, scale, and resilience.

Sensors Metrics
Humans Agents

Akka Orchestration

Guide, moderate, and control long-running systems.

Akka Memory

Durable, in-memory, and sharded data.

Akka Agents

Create agents, MCP tools, and HTTP/gRPC APIs.

Akka Streaming

High performance stream processing.

Agents Vector DB
OLAP Transactional DB
Models Message Brokers
LLMs APIs & Tools

Composable

Akka SDK

Composable components that do not require prior developer knowledge of events, actors, persistence or asynchrony.

Reasoning

Agents
Build complex enrichment loops that integrate memory and tools with an effects builder.

Memory
In-process, durable short-term and long-term history continuously updated by events.

Tools
Create functions that can be requested by agents and custom MCP servers as Endpoints.

Systems
Create swarms and teams of orchestrated agents that comm through events or A2A protocols.

Transacting

Workflows
Execute durable, long-running processes with point-in-time recovery.

Entities
In-memory, durable, and replicated DB.

Streaming
Streaming producers and consumers to enable real-time data integration.

Endpoints
Build HTTP and gRPC APIs.

Views
Create read-only projections of complex data spanning across distributed system.

Timers
Execute actions with a reliability guarantee.

Deploy anywhere

Akka Runtime

Your services are packed into a single binary. Deploy instances on any infrastructure.

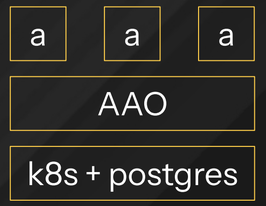
1. Create any number of instances...

2. ... which deploy onto any existing infrastructure ...

bare metal | VMs | k8s | PaaS | edge

3. ... that securely self-cluster without a service mesh.

4. Optionally, add Akka Automated Operations to gain:



- ➔ No downtime updates
- ➔ Multi-region failover
- ➔ Auto-elasticity
- ➔ Persistence oversight
- ➔ Multi-tenant services
- ➔ Deploy in our serverless cloud
- ➔ Or, deploy in your VPC
- ➔ Observability via control tower
- ➔ Multi-org access controls
- ➔ Certificate and key rotation

An AI ecosystem that accelerates delivery

AKKA

AI DevEx for every AI tool, event-driven integration that unifies, and operational EvalOps tie-ins

AI inference

OpenAI · vLLM · ANTHROP\C · Gemini · Hugging Face

Tools

Model Context Protocol · OPENAPI INITIATIVE · Browserbase · E2B

Evaluation

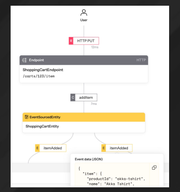
arize · comet

AKKA

Foundation models

deepseek · LLaMA by Meta

Observability



Semantic data stores

Weaviate · neo4j · Pinecone · milvus

Akka - Agentic Cost Savings

AKKA

Increase certainty while accelerating velocity and lowering costs.

Akka Total Savings

$1K / billion tokens

- ➔ Akka memory and context mgmt (sliced windows, compaction) sends LLMs ~5% fewer tokens vs. OSS.
- ➔ Intelligently short-circuit LLM execution (ie tokens) with reactive, non-blocking architecture.

** Per billion token rates vary from $50 - $2K. Pricing depends upon model choice, context window size, throughput, latency, egress, and infrastructure fees.

$1.4K / year / core

- ➔ Langchain - orchestration, agents, APIs, memory and streaming on separate compute.
- ➔ Actor-based concurrency model drives 70% higher per core execution density.

** Per core pricing will vary from $500 - $2K / year depending upon many factors: hyperscaler, RAM configuration, I/O configuration, regions, OS, and payment options.

$2K / day / HC

- ➔ Langchain - more HC to absorb diff DevEx for orchestration, agents, APIs, memory, streaming.
- ➔ Akka - Uniform DevEx, multi-team envs, & agent self-clustering for no-effort day 2 ops.

** Demonstrated productivity of 1 Akka HC is equivalent to 3 Langchain devs at per diem rate of $1K / HC.

A vibe-to-power DevEx

Maximum dev productivity for any way you choose to work

	Vibe ideation	AI-assisted coding	Power coding
Dev tool	ChatGPT Gemini Claude Desktop	Cursor Windsurf + Akka AI Coding Kit Qudo	Any IDE maven

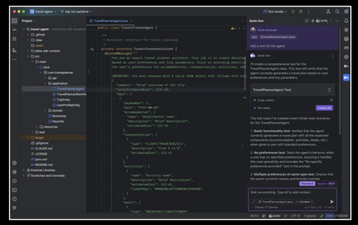
Dev UX	Agentic prompts
	"Generate an AI agent definition for an agent that uses the gpt-4o model to evaluate a combat turn in a tabletop RPG game. This agent will use a function tool called `roll_dice` whenever it needs to obtain random results."
		Akka prompts
		"I'm developing a multi-player game for rock-paper-scissors. Use the Akka SDK to define entities representing players and a game. Use `EventSourcedEntity` in the Akka 3 SDK. Use the same code structure as in `template/src` directory."
			IDE + terminal
			`project/`
			`  api/`
			`  application/`
			`  domain/`

Dev output	Agent DL + Agent orchestration DL + Akka agent code	Akka SDK components + Akka agents + Test suite	Akka SDK components + Akka Libraries + Test suite

```
agent combat-evaluator v0.0.1
  model gpt-4o
  emit function-events
  prompt
    "Generated prompt"

  function roll_dice
    param expression string
```

```
class Agent {
  public Effect<String> suggest(String question) {
    return effects()
      // retrieve system message from the SDK
      // define agent behavior by chaining calls
      .systemMessageFromTemplate("activitySytemPrompt")
      .includeSessionMemory()
      .userMessage(question);
  }
}
```



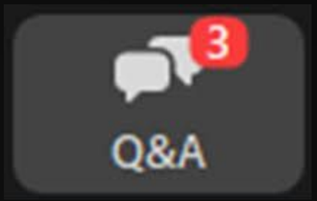
Get started

Over a dozen examples: real-time, adaptive, IOT, RAG, & more

```
akka code init <new_project>
```

Demo

Live Q&A



Use the Q&A, not the chat function for questions

Next steps

Discuss

Let’s start a conversation
We’ll connect you with the right Akkatects to address technical and commercial questions.

Do

Dig deeper on your own
Whether you want to read more about Akka, get into the SDK, you can get started on your own now.

See

Concept to proof in 48 hours
Given the power and simplicity of Akka, we can prove out your use case quickly. Let’s discuss – we’ll show you real results.

AKKA

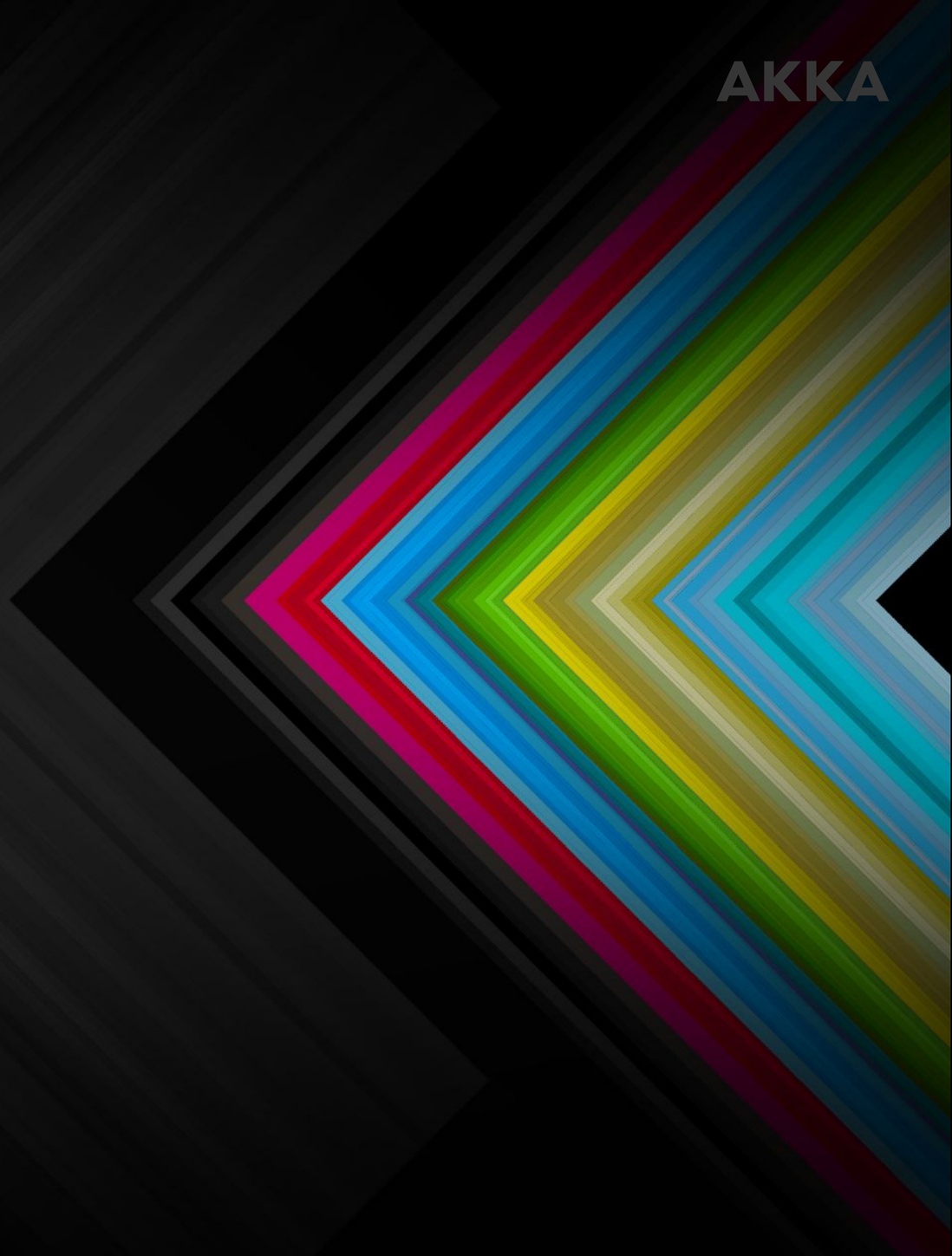
Thank you!

AKKA

Additional resources

- Prompts
- Background slides
- Akka product detail slides
- Case studies

Prompts used in the webinar demo



Add an agent

AKKA

Create an Akka SDK agent named `GreetingAgent` that assists users in learning greetings in different languages.
- **Input:** A string question.
- **Output:** Returns an `Effect<String>`.
- method with name `ask`
- Do not perform any environment variables check.
- Do not add JavaDoc annotations
- place it into com.example.application package
- system message should be a static variable
- **Guidelines for system prompt:**
    - The user must provide a language.
    - Always append the language used in parentheses in English (e.g., "Bonjour (French)").
    - At the end of each response, append a list of previous greetings used in the current session.

Test an agent

AKKA

Develop only one integration test for `GreetingAgent` in the `com.example` package, named `IntegrationTest`.
- Include a `TestModelProvider` to mock the model with a fixed response.
- Ensure all necessary classes are properly imported.

Add an endpoint

Implement an HTTP endpoint `GreetingAgentEndpoint` with the following specifications:
- Include a `TestModelProvider` to mock the model with a fixed response.
- Ensure all necessary classes are properly imported.

Add an MCP endpoint

AKKA

Create an MCP server endpoint named `UserNameMcpEndpoint` for retrieving user's name based on `userId`.
- Place the implementation in the `com.example.api` package.
- Provide a mock implementation that returns a random name for each query.
- add all needed annotations (class and method)
- use MCP Tool annotations with all required parameters including the McpEndpoint annotation

Add an MCP tool to an agent

AKKA

Update the `GreetingAgent` to display the user's name during the first interaction, for example: "Hello <user name>!".
- Configure GreetingAgent to use `UserNameMcpEndpoint` as an MCP tool. No need to use AllowedToolNames. Use the service name specified by the `artifactId` in `pom.xml`.
- set `userId` from context().sessionId().
- Add the `userId` to the user message in the following format: "userId:<userId>;question:
- update system prompt accordingly

Add streaming support

AKKA

Update the `GreetingAgent` to display the user's name during the first interaction, for example: "Hello <user name>!".
- Configure GreetingAgent to use `UserNameMcpEndpoint` as an MCP tool. No need to use AllowedToolNames. Use the service name specified by the `artifactId` in `pom.xml`.
- set `userId` from context().sessionId().
- Add the `userId` to the user message in the following format: "userId:<userId>;question:
- update system prompt accordingly

Background

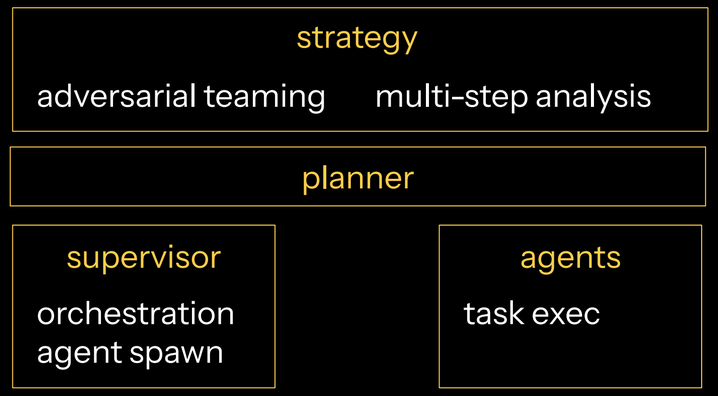
Agentic systems are distributed systems that must deliver reliable outcomes while depending upon unreliable LLMs.

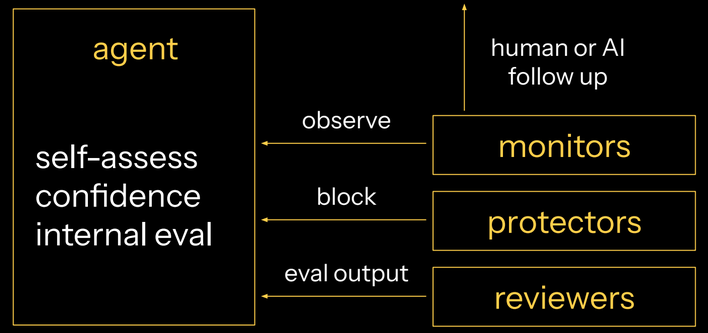
Anatomy of an Agentic System

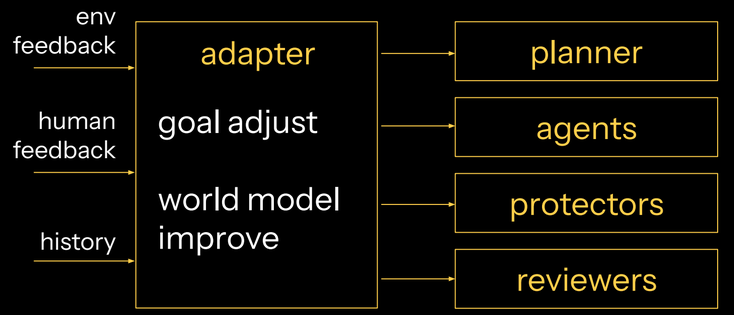
Goals
Create a plan and execute steps that work toward achieving a goal.

Guardians
Oversee agent capabilities, balancing runtime decision making with risk mgmt.

Adaptation
A system that continuously alters goals and behaviors as its world model improves.







Coordination
Participants are coordinated through dynamic orchestration adjusted by the supervisor.

State
Evolving goals require shared data and memory:

1. Goals & sub-goals
2. World model
3. Plans and execution progress
4. Communication logs & history
5. Learning & experience
6. Resource usage

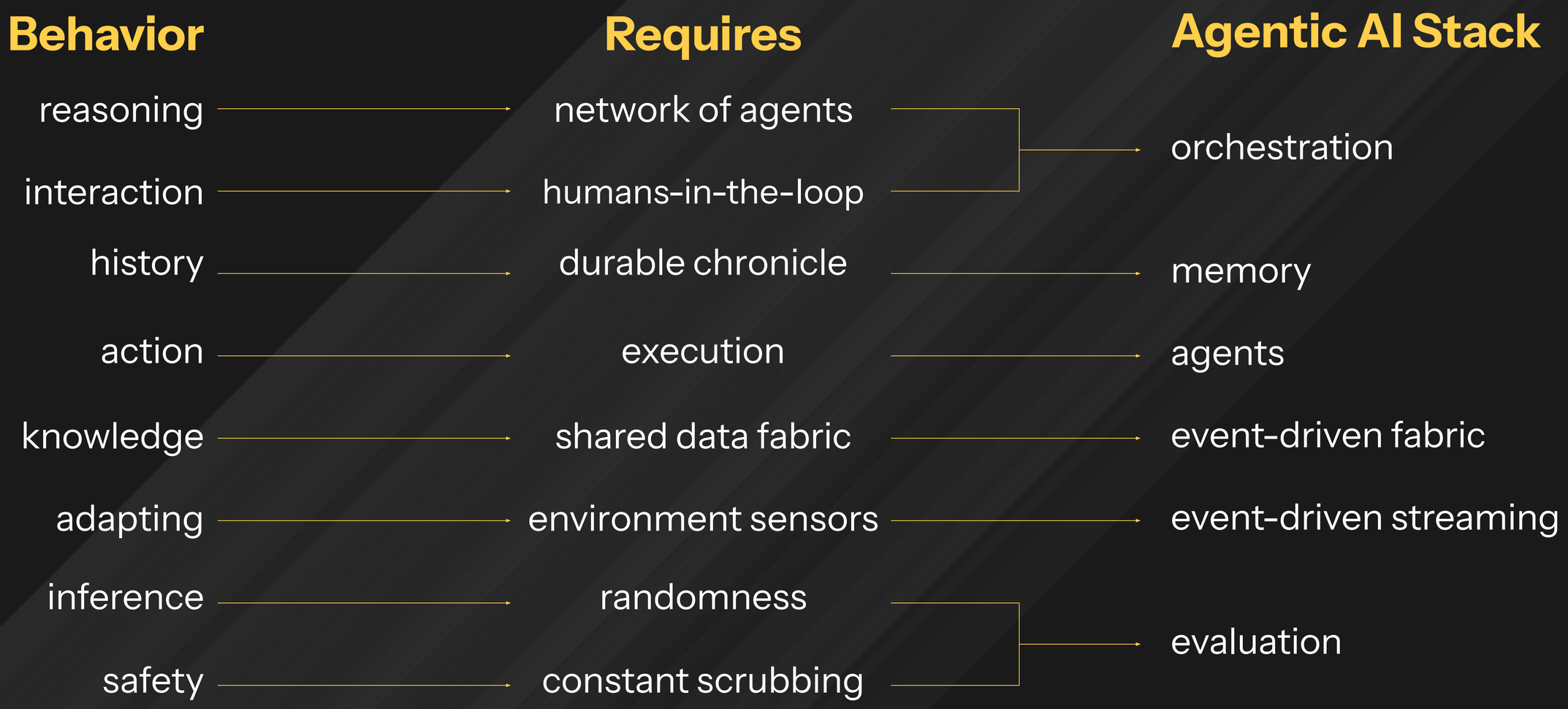
Registry
List available resources, tools, actions, and access controls:

1. Action & tasks
2. Verification criteria & results
3. Resource allocation & availability

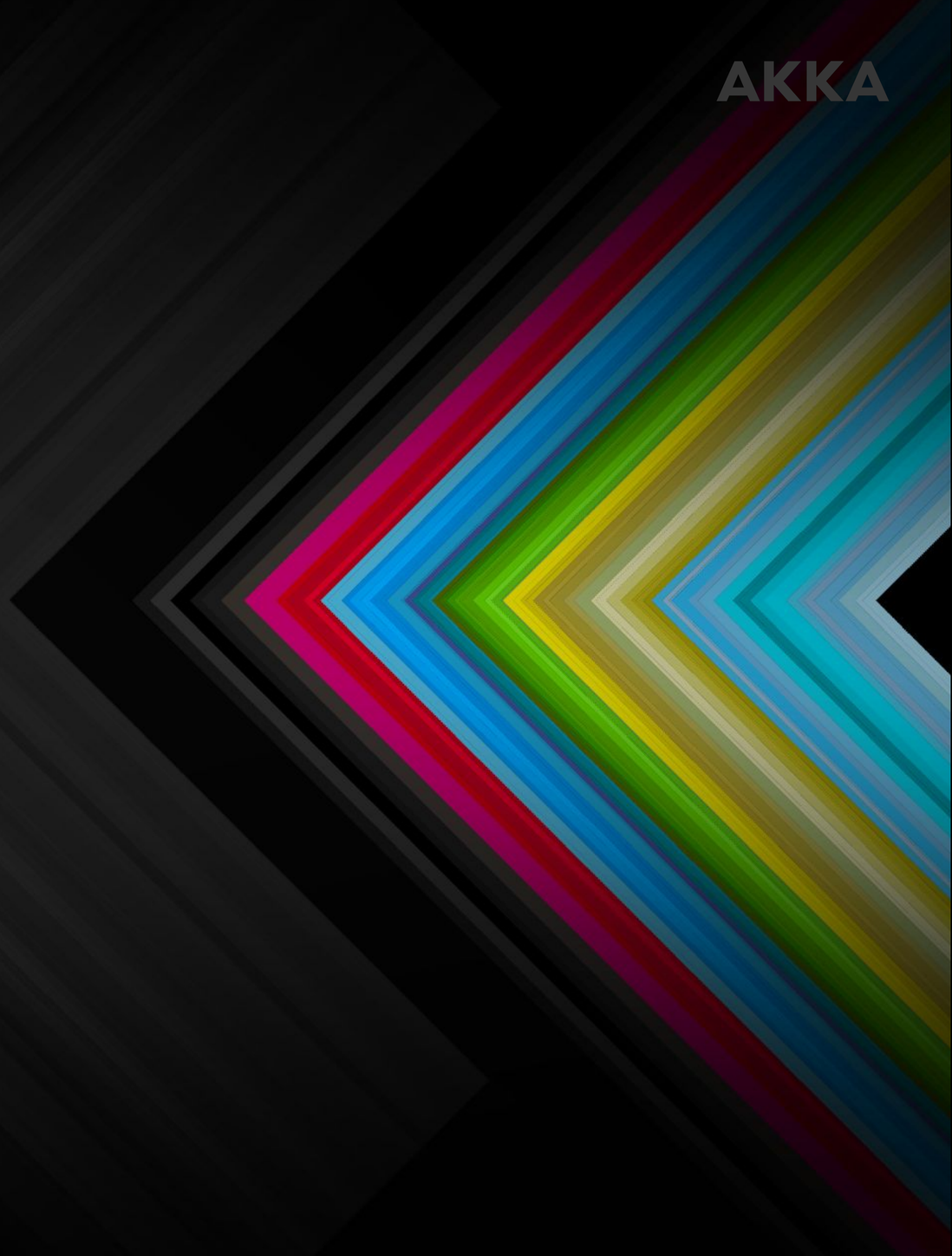
Emerging agentic AI stack for IT

orchestration + memory + event-driven + streaming + evaluation

AKKA



Product details



Akka Agentic Platform

Sensors Metrics
Humans Agents

AKKA

Akka Orchestration

Guide, moderate, and control long-running, multi-agent systems

Akka Agents

Create agents, MCP tools, and HTTP/gRPC APIs

Akka Memory

Durable, in-memory and sharded data

Akka Streaming

High performance stream processing

Development

AI DevEx with composable components

Runtime

Award-winning actor runtime for infinite scale and resilience

Operations

Any infrastructure, cloud, or tool

Security

Zero trust, guardians, and evaluation

Agents Vector DB
OLAP Transactional DB
Models Message Brokers
LLMs APIs & Tools

Akka Agentic Platform

Env sensors | IOT metrics | Human feedback

AKKA

Akka Orchestration

Guide, moderate, and control long-running, multi-agent systems across disruptions, crashes, delays, and infrastructure failures, with sequential, parallel, hierarchical, and human-in-the-loop workflows.

Akka Agents

Create goal-directed agents, MCP tools, and HTTP/gRPC APIs that reason, act, and analyze. Integrate any 3rd party broker, agent or system.

Akka Memory

Durable, in-memory, and sharded data for agent context, history retention and personalized behavior. Nano-second writes and replication for failover.

Akka Streaming

High performance stream processing for ambient, adaptive, and real-time AI. Continuous processing, aggregation and augmentation of live data, metrics, audio and video.

Development

Build and test services with an AI-assisted DevEx that uses event-driven, component composition.

Runtime

Our award-winning actor runtime that clusters your services from within for infinite scale and resilience.

Operations

Deploy and operate on any infra, anywhere.

Development	Local, offline build and test of agentic systems
Self-Managed Operations	DIY ops on bare metal, k8s, PaaS, or edge
Akka Automated Operations	Deploy in our serverless cloud or your VPC

Security

Zero trust architecture, guardians, evaluation, and 19 compliance certifications for ironclad safety.

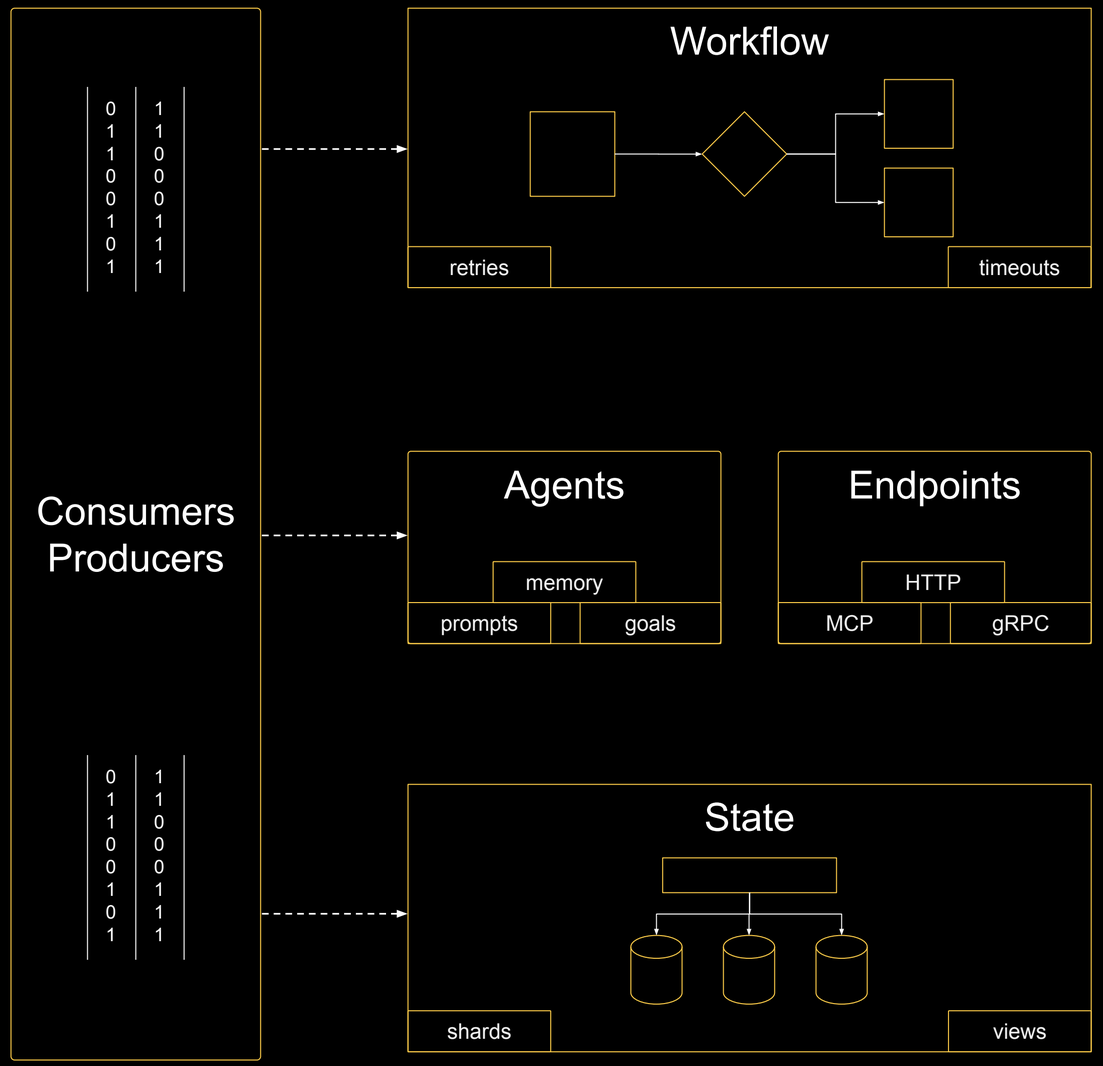
Transact DB | OLAP | Models | Vector DB | Remote agents | Message brokers | APIs & tools

Akka Agentic Platform

AKKA

Akka Streaming

High performance stream processing



Akka Orchestration

Guide, moderate, and control long-running systems

Akka Agents

Create agents, MCP tools, and HTTP/gRPC APIs

Akka Memory

Durable, in-memory and sharded data

Akka Orchestration

Agents that complete the mission – no matter what

Workflows

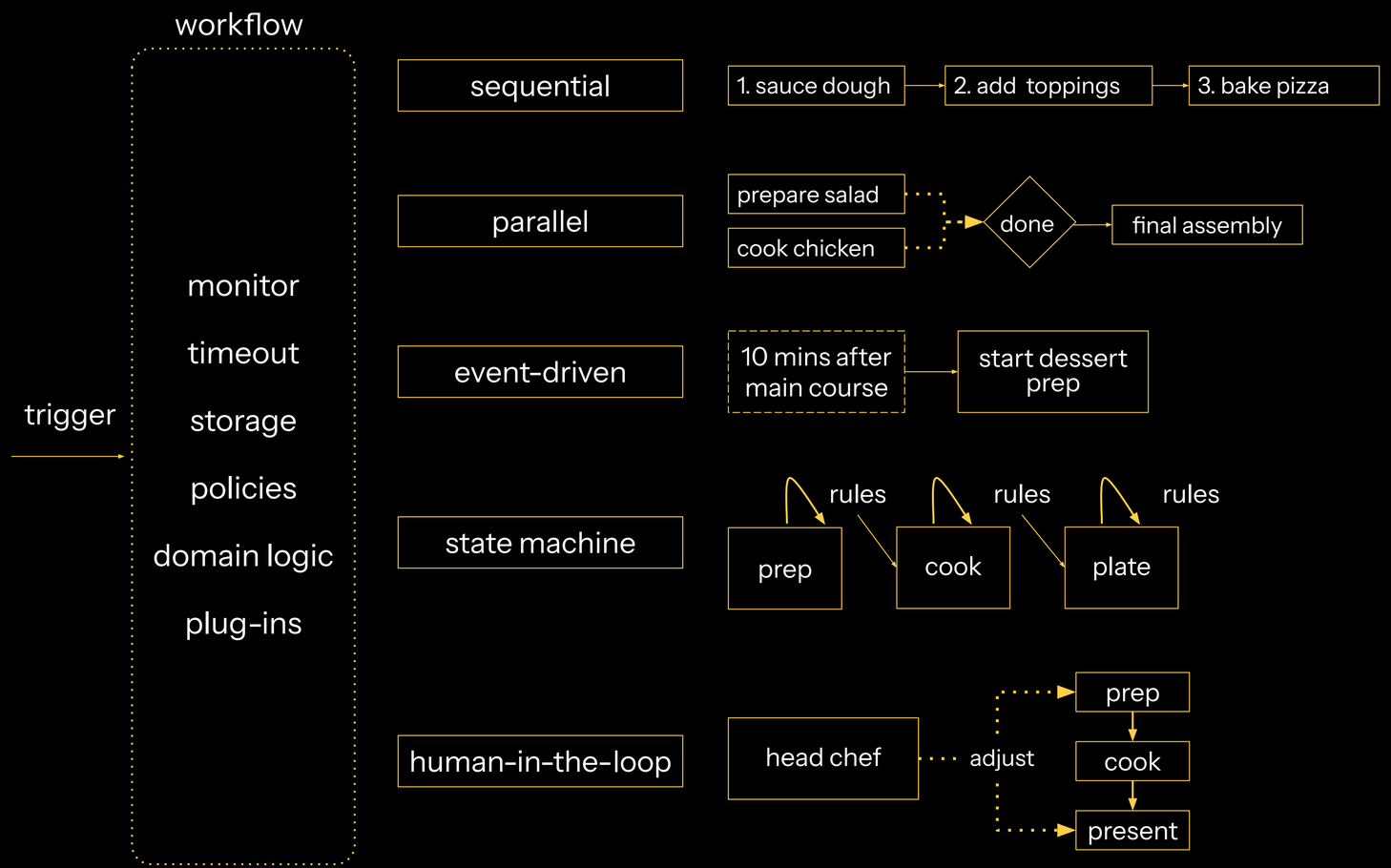
Sequential, parallel, and event-driven with exactly-once execution.

Long running

Persist through crashes, restarts, and extended interruptions.

Visual tracing

Inspect workflow state, trace execution paths, and debug failures.



Akka Agents

Goal-directed agents with purpose: reason, act, and transact

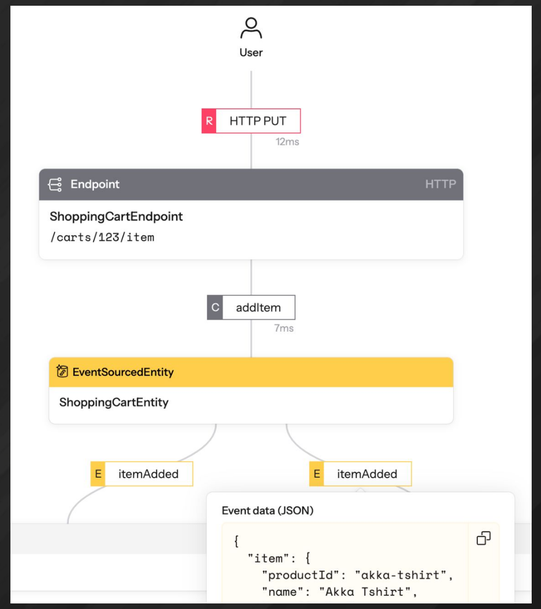
Agents & Endpoints

Create agents with memory, tools, prompts, and context then attach them to HTTP, gRPC, or MCP Endpoints.

```
effects()
  .systemMessage("")
  .memory(<<memory_provider>>)
  .tools(<<local_or_remote_MCP>>)
  .userMessage(question)
  .thenReply()
```

Flow Tracking

Visualize, track, and understand event flows with agent evaluation for performance and accuracy.



Testkit

Execute local, integration, and CI/CD tests with a library for asserting behaviors of components and services.

```
client
  .forAgent()
  .inSession(sessionId)
  .method(Agent::ask)
  .invoke()

Assertions.assertNotNull()
```

AKKA

Akka Memory

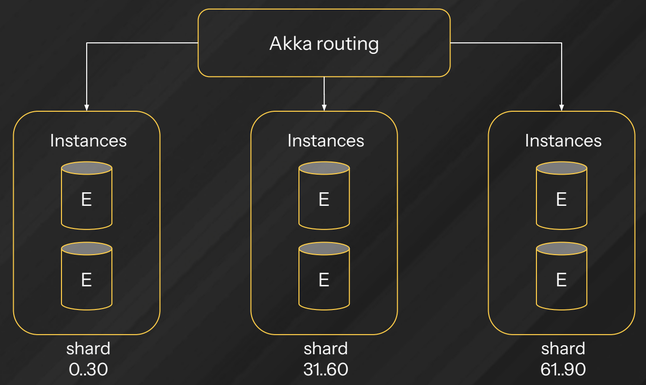
Agents that remember what matters

Event Sourcing

Create long-term state as a sequence of events that can be replayed, replicated, broadcast, or persisted.

Data Sharding

Akka automatically shards data across cluster nodes and replicates across regions for failover and DR.



Delegation Effect

Developers describe application outcomes through Effects. Akka handles the persistence.

```
effects()
  .updateState(data)
  .end()
  .thenReply(Done)
```

Effects express your intent:

1. update state
2. transition state
3. reply to caller
4. raise error
5. persist event
6. include memory
7. pause execution

Akka Streaming

High performance stream processing for agents that act on live data

Continuous Processing

Events, metrics, audio, and video real-time, non-stop from any source to any sink.

Flow Control

Back pressure and throttling that prevents process exhaustion.

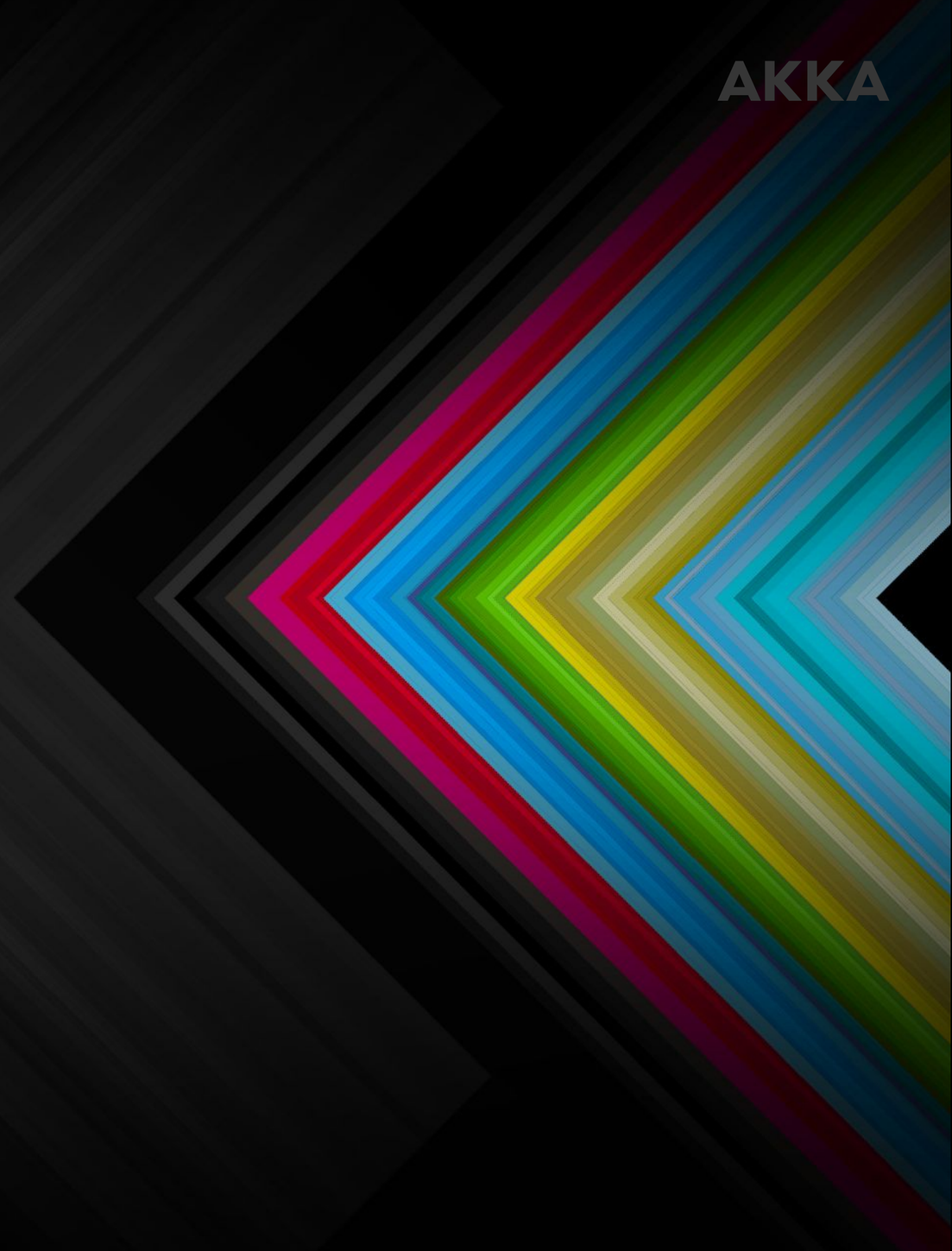
Brokerless Messaging

Flow streams between services and agents without a middle person.

```
public Effect onEvent() {
   return switch (event) {
      case ValueIncreased increased →
        // process event
      case ValueMultiplied multiplied →
        // process event
   };
}


kafkaTopicSource
   .filter(validMessage)
   .map(transformMessage)
   .mapAsync(parallelism = 4, sendEmail)
   .runWith(Sink.commitMessage);
```

Case studies



- Get the **engineering** right, and the AI will work
- Design for **trust** from day one
- **Evolve**, don’t break, your architecture
- Create **layers of certainty** to enable determinism

AKKA

Swiggy 4M recommendations per second

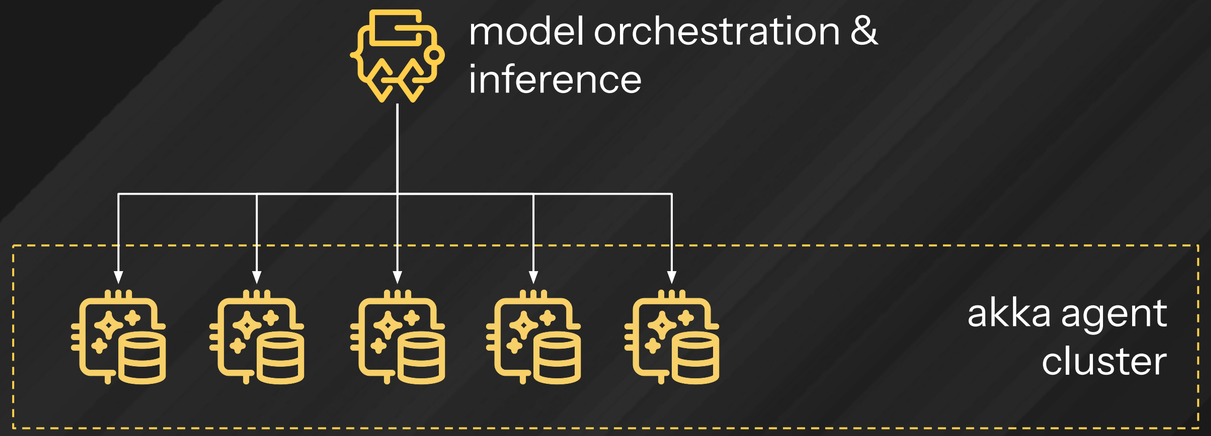
Get the engineering right and the AI works itself out.

Demand-driven elasticity

Swiggy executes API-driven predictions with multi-model fan-out and ultra-low latency for every user request.

Actor runtime and event-driven

Running agents in a cluster of actor runtimes enables recovery from failures, nearly infinite concurrency, and non-blocking communications.



6 models

72ms p(99) latency

10K cores

4M model hits / sec

Tubi Adaptive Content Recommendations

Evolve, don't break, your architecture

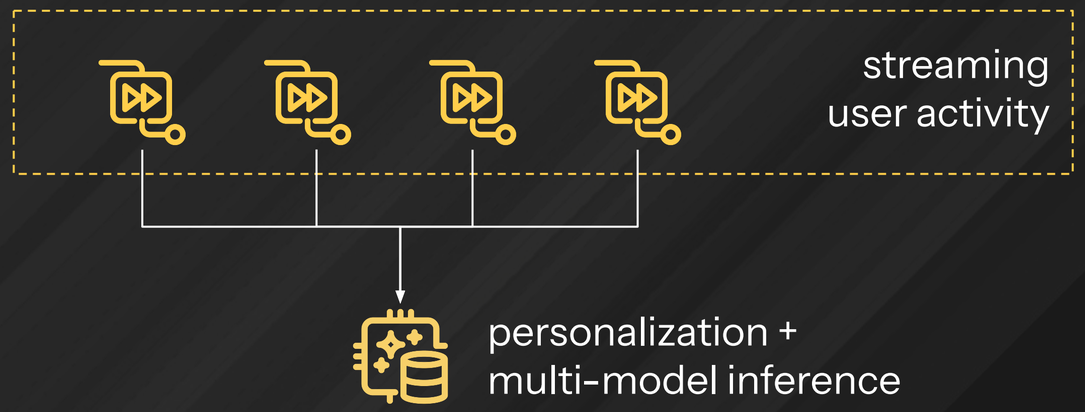
AKKA

Continuous data without overload

Tubi provides users real-time content recommendations influenced by more than a dozen sources of continuously updated user activity - like their browsing history.

Ingest real-time event streams

Apply real-time streams of data (as events) to ML models with in-memory, durable journals tracking history.



2 months delivery

75mm MAUs

8 streams

Llaama AI for Personalized Healthcare

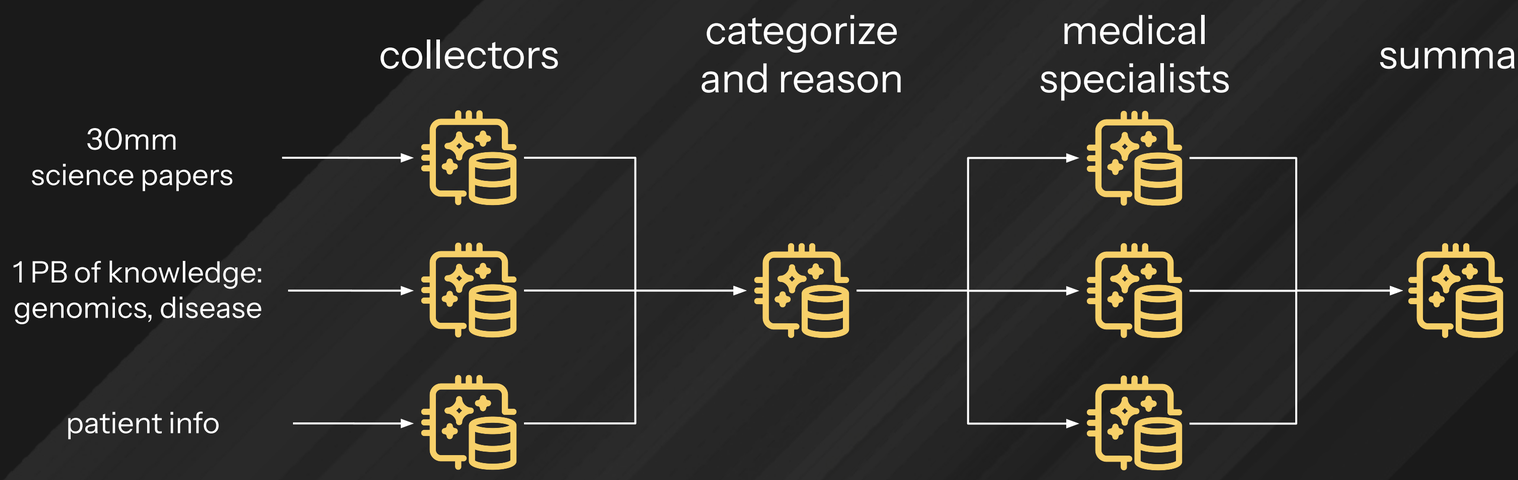
Create layers of certainty to enable determinism

High velocity dev and delivery

LLaama deciphers terabytes of medical research with sourcing, analysis and reasoning.

Orchestration coordinates many agents

Orchestration coordinates the execution of 1000s of concurrent Akka actors into a reasoning system that would take Python-based agents months to complete.



2 devs
2 months

16 agents

Results in hours

Global Top 3 Retailer

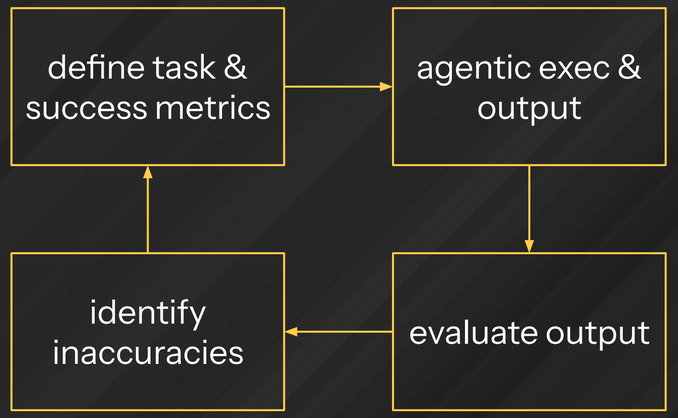
Design for trust from day one

Solve problems faster

Nervous about the randomness of LLMs, a retailer introduced agents into a non-critical system to create ‘how-to-fix’ suggestions on integration errors later reviewed by IT staff.

Eval-driven development

Agents rely upon humans, other models, and self-evaluation to measure and improve AI accuracy, performance and safety.



5 billion msgs / yr

500 errors / yr

72% faster time to fix

10 → 6 fix it HC