



DEV DAY

SAN FRANCISCO

AI Agent - Overview

Uma Ramadoss

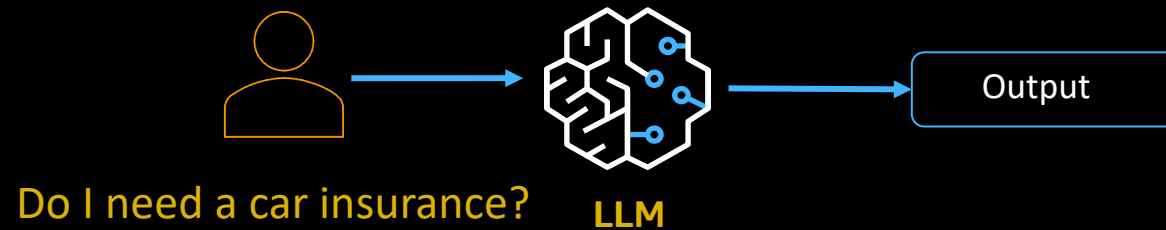
Principal Solutions Architect @AWS



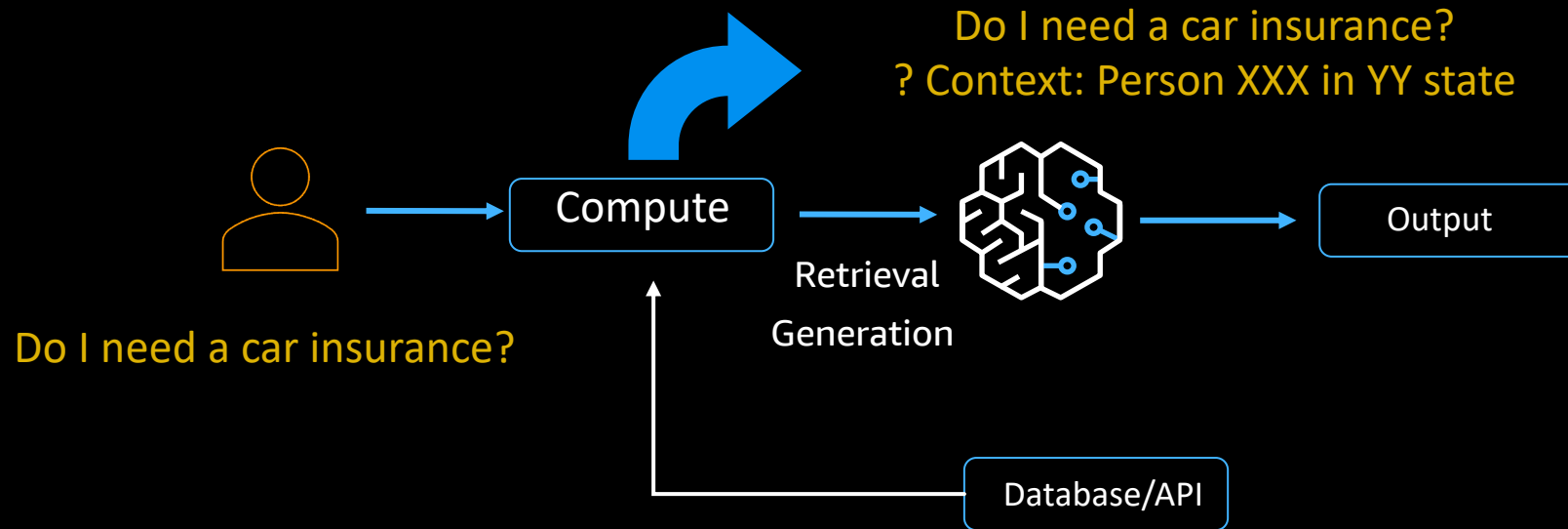
Agenda

- What is an AI Agent?
- Types of AI Agents
- Building Agentic AI on AWS

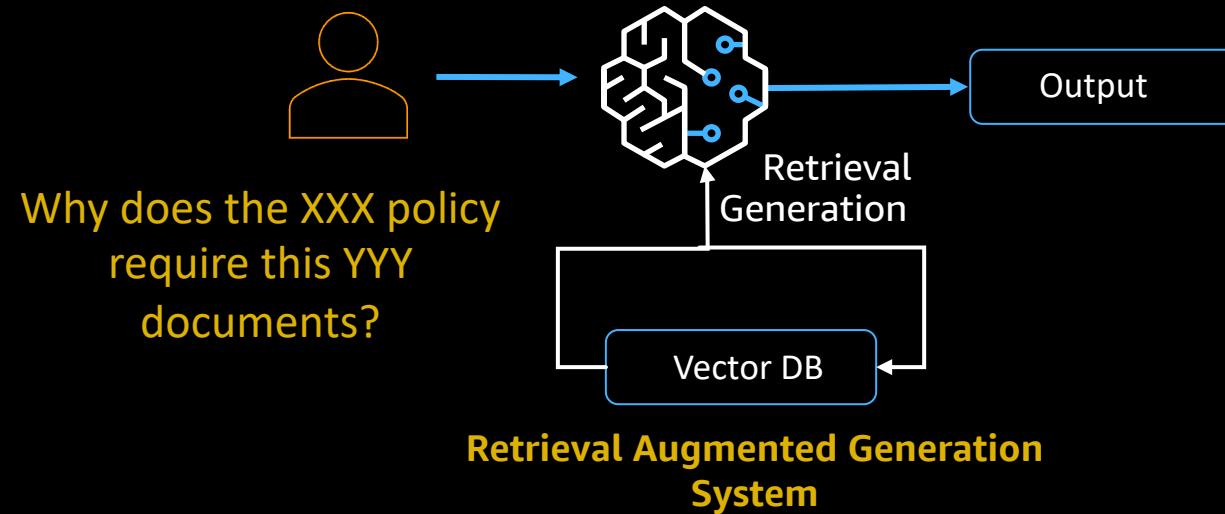
Rise of Agentic AI Systems – Simple invocation



Rise of Agentic AI Systems – Domain specific invocation



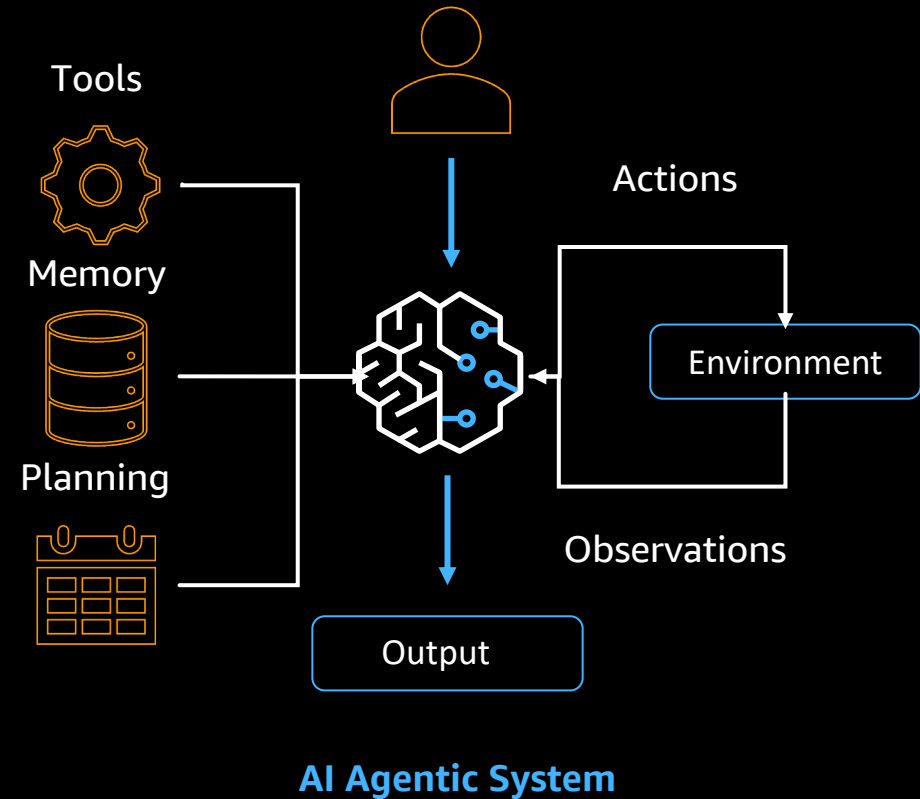
Rise of Agentic AI Systems – Domain specific invocation



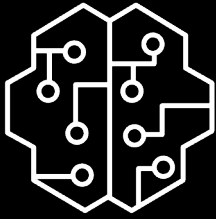
Rise of Agentic AI Systems – AI agents

Finance Analyst

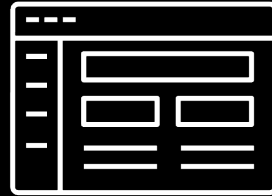
How does the Amazon's revenue, profit margins, and cash flow look in 2025? Are there any trends or significant changes year-over-year?



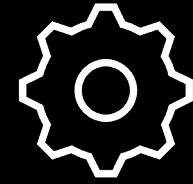
AI Agentic Systems



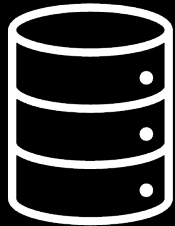
Plan and sequence **actions**
to achieve a **goal**



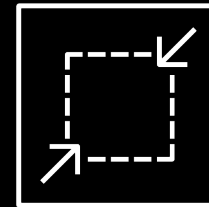
Perceive and process **context**
aware information



Use **tools** to perform tasks
efficiently



Remember past **interactions**
and **behaviors**



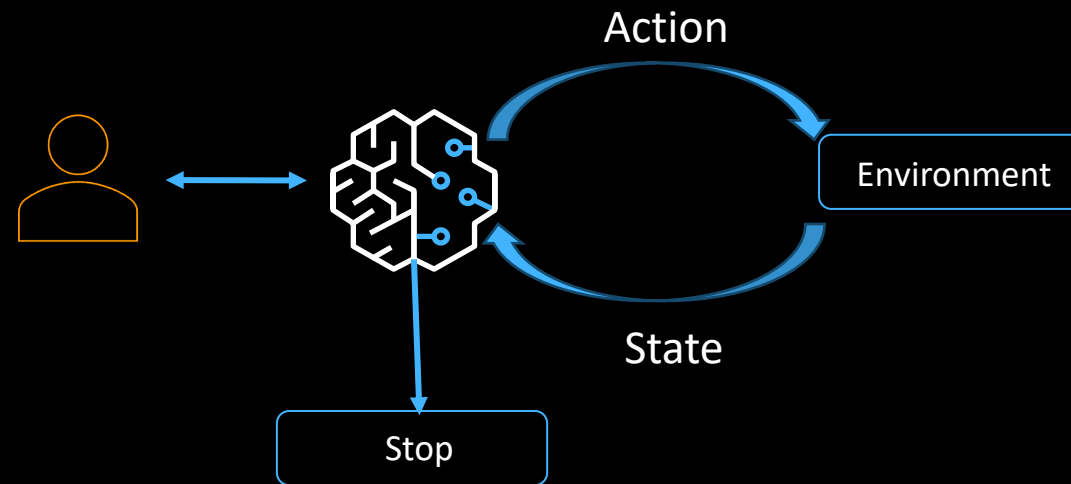
Makes decisions and **execute**
actions based on goals

“

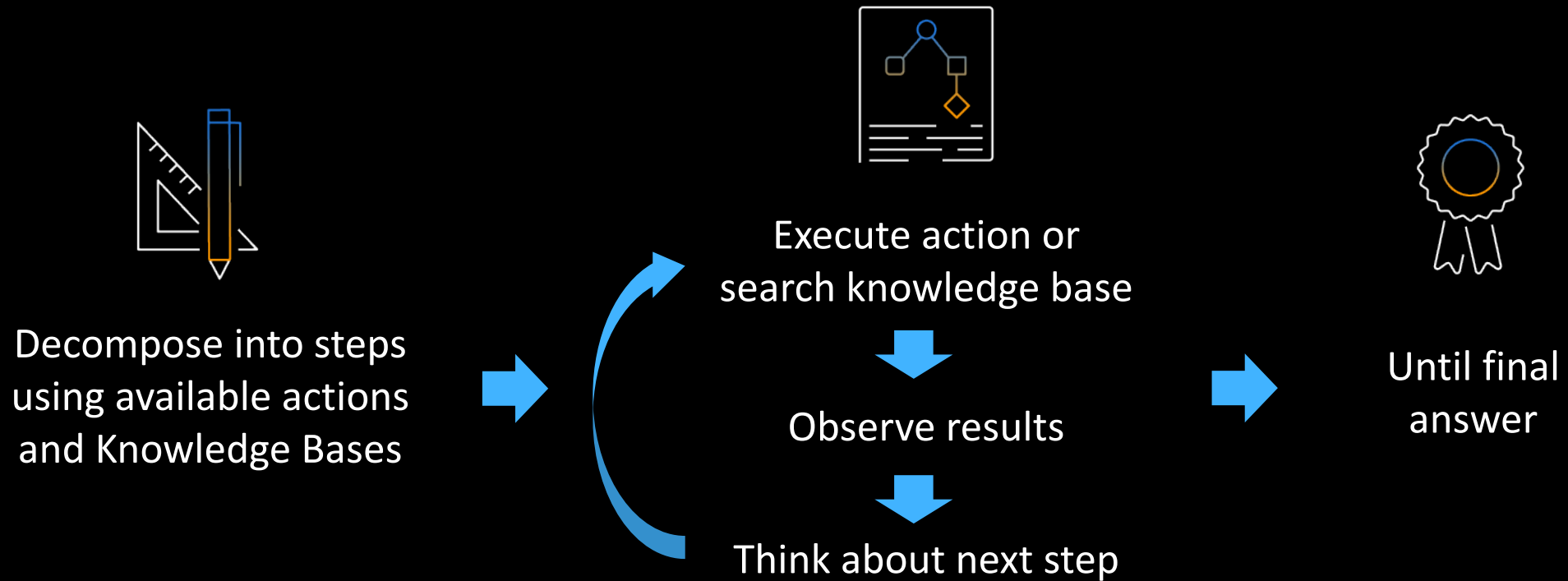
Agents are essentially models
using tools in a loop.

Anonymous

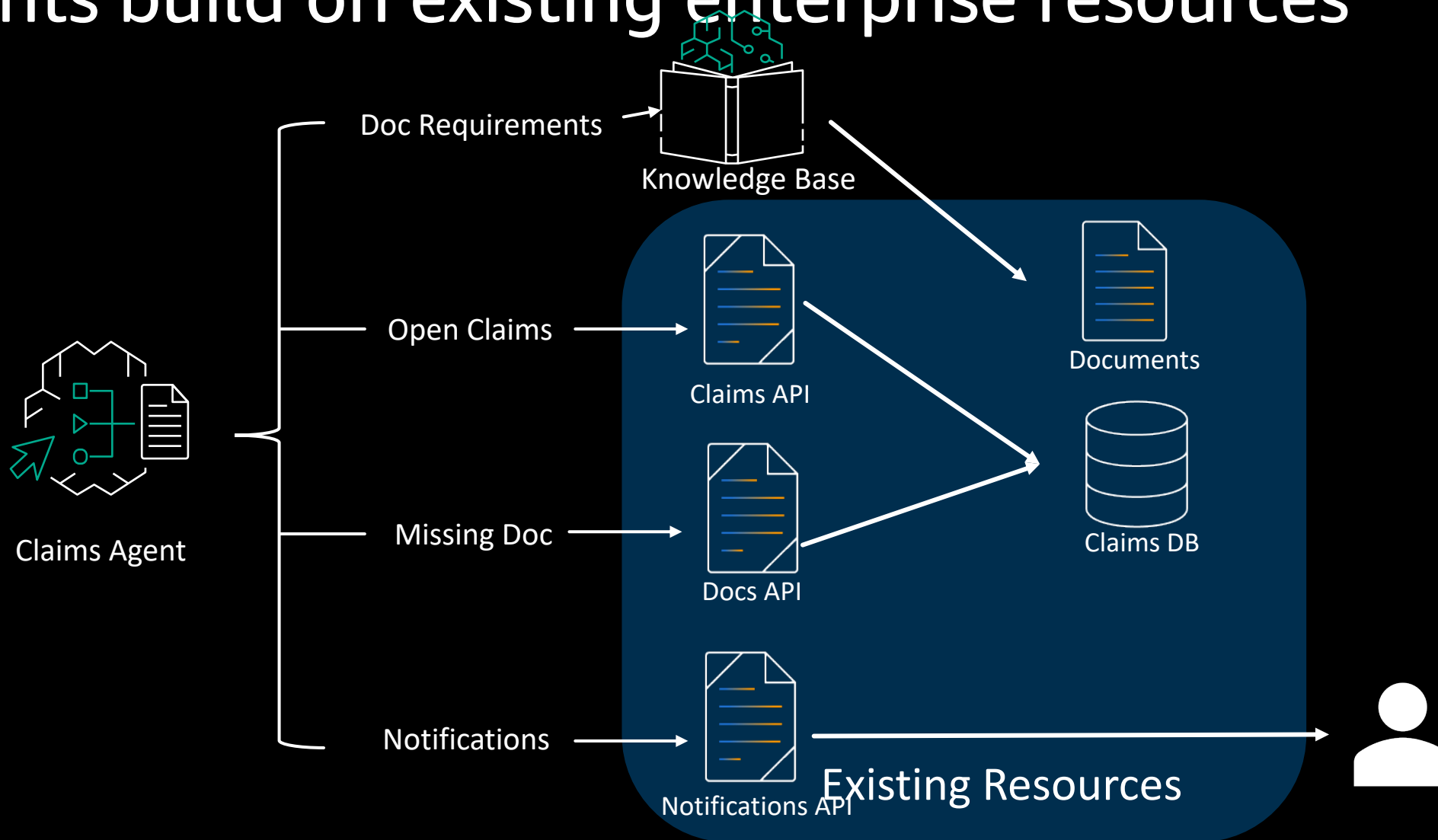
Agentic systems rely on action and state



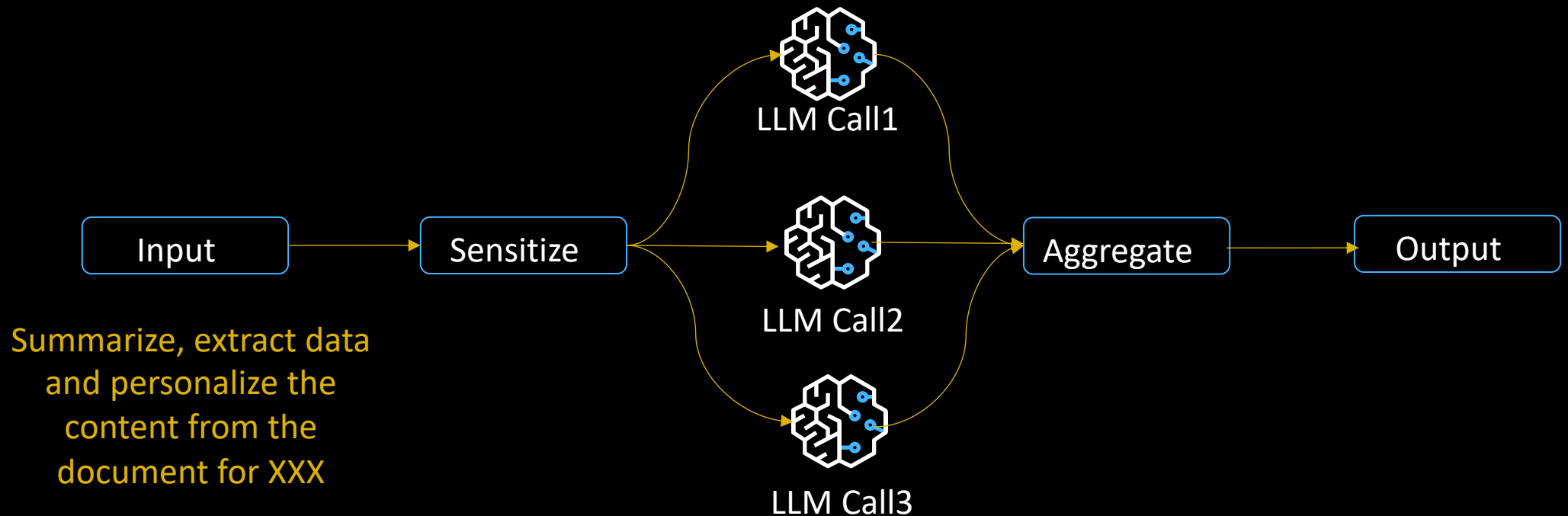
Agent is an orchestrator



Agents build on existing enterprise resources



AI workflows are different from agentic AI



Implementing Agentic AI on AWS



Different approaches to implement Agentic AI

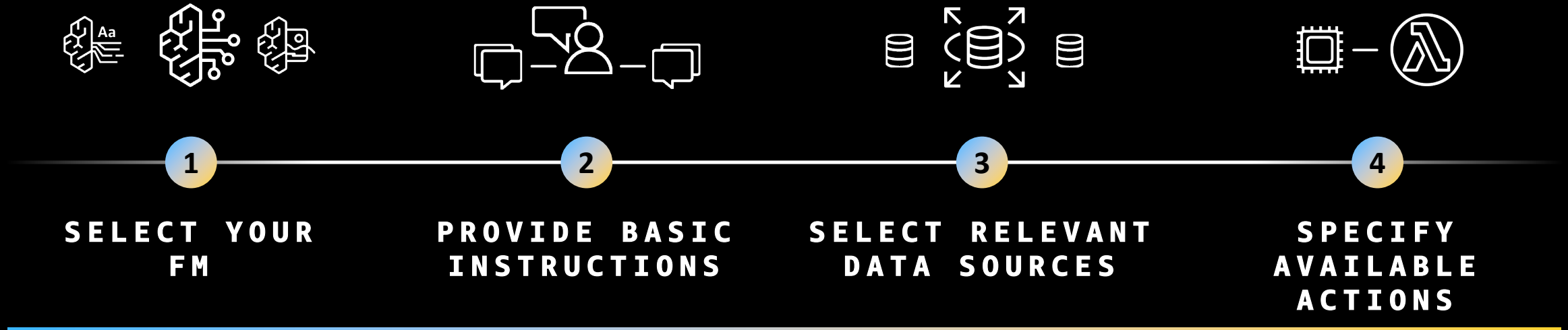
- Amazon Bedrock Agents
- Containerize your favorite Agent framework
- Use Step Functions as agent

Amazon Bedrock Agents



Amazon Bedrock Agents

Enable generative AI applications to execute MULTISTEP tasks using company systems and data sources



Action Groups



Define Action Group

- Overview
- Relevancy

Define API Schema



- Operation name,
- Input parameters,
- Data types,
- Response details



Define Lambda functions

- Business logic
- Choice of programming language (Python, C#, JavaScript, Java, ...)

Benefits



Simplifies building
and deploying
AI assistants



Provides secure
access to enterprise
data and APIs



Enables
CoT traces for visibility



Provides fully
managed
infrastructure

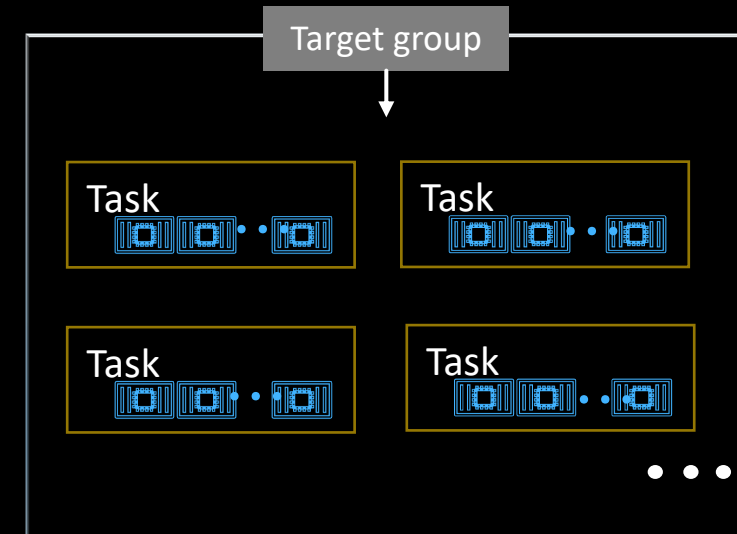
Containerize OSS agents

Building Agents with Open-source

- Build agents with [LangGraph](#), [LlamaIndex](#) using Bedrock Converse API.
- Deploy, scale and manage



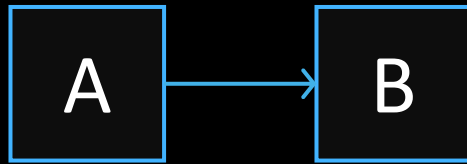
AWS Fargate



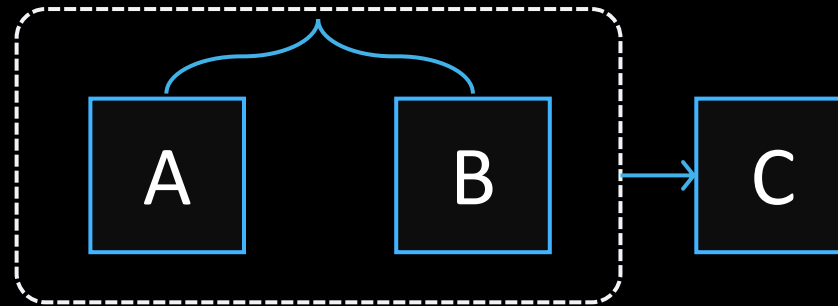
Use an orchestrator

AWS Step Functions is a **Serverless** orchestrator

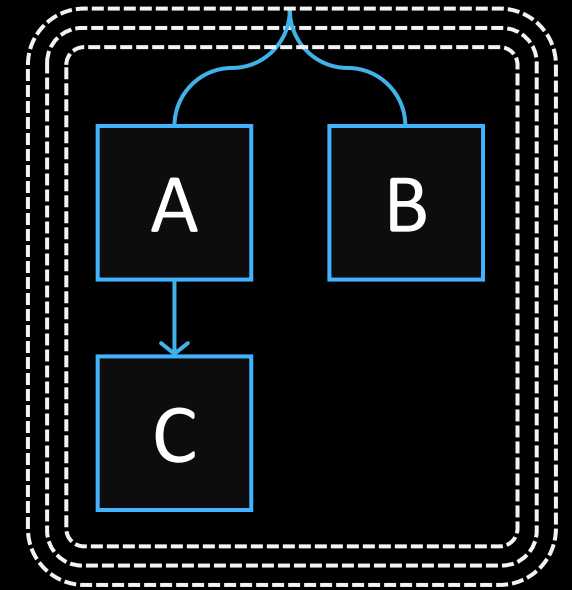
“I want to
sequence tasks”



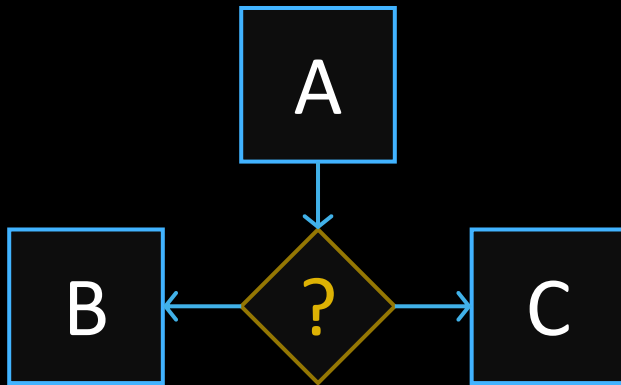
“I want to run tasks
in parallel”



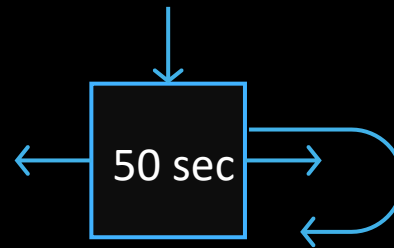
“I want
concurrent and iterative
tasks”



“I want to select tasks
based on data”



“I want wait
(re)try/catch”



“I want
Human review”



Workflow Studio | Integrations sidebar

Flow States

Actions

Flow



Choice
Adds if-then-else logic.



Parallel
Adds parallel branches.



Map
Adds a for-each loop.



Pass
Transforms data or acts as placeholder.



Wait
Delays for a specified time.

Optimized Integrations

Actions

Flow

MOST POPULAR



AWS Lambda
Invoke



Amazon SNS
Publish



Amazon ECS
RunTask



AWS Step Functions
StartExecution



AWS Glue
StartJobRun

AWS SDK Integrations (11,000+)

51 matches found



Amazon Rekognition
CompareFaces



Amazon Rekognition
CreateCollection



Amazon Rekognition
CreateProject



Amazon Rekognition
CreateProjectVersion



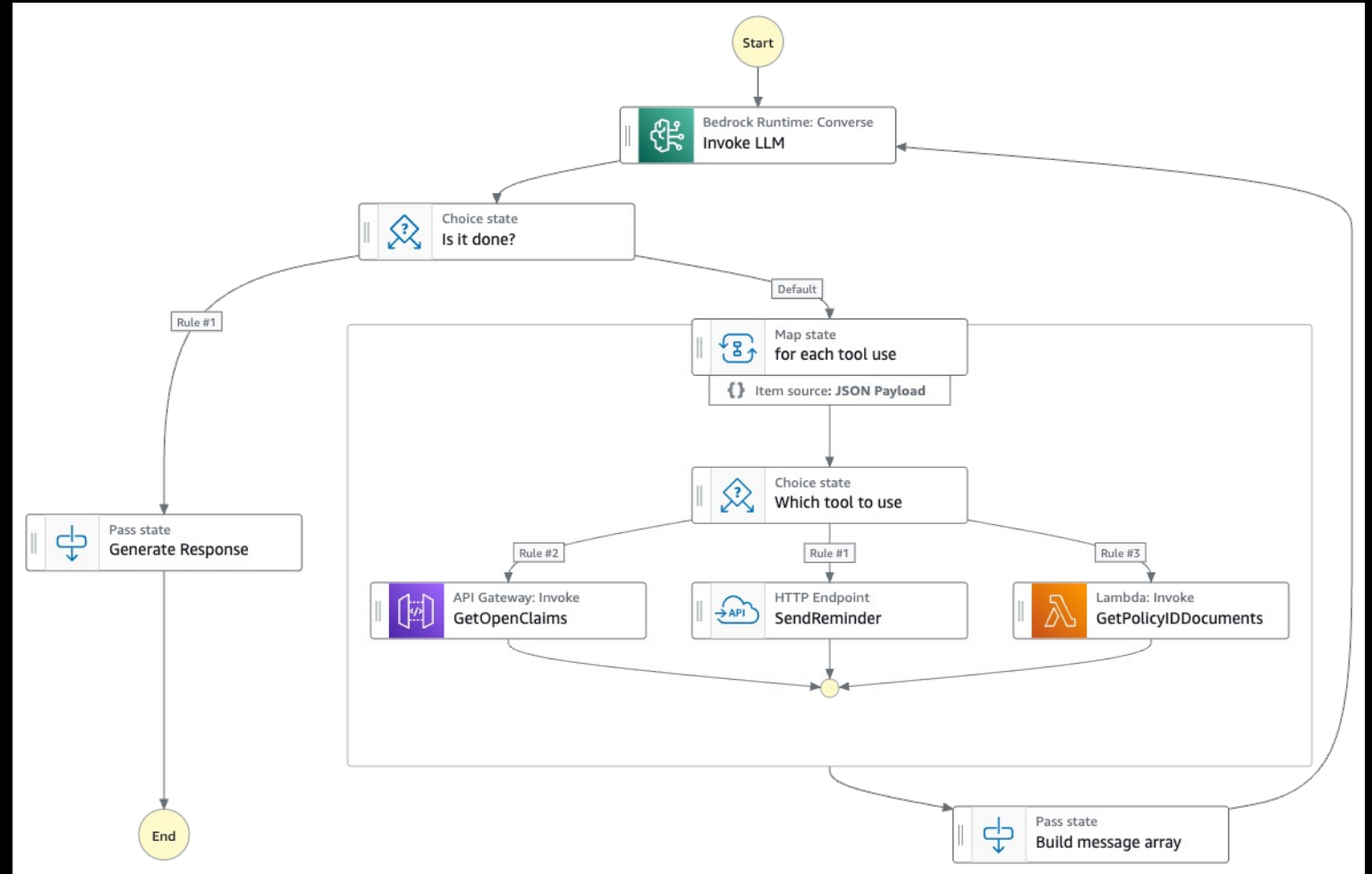
Amazon Rekognition
CreateStreamProcessor



Amazon Rekognition
DeleteCollection

Step Functions with Tool Use

- Invoke LLM to plan and identify tools
- Check if this is no more tools
- If tools, invoke tool
- Keep iterating until no tool



Benefits - Step Functions as agent

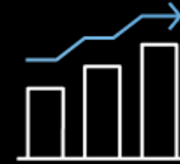


Freedom to choose
the models



Modular Architecture

1. Scaling tools and agents independently
2. Nested workflows for reusability



Scale & Performance

1. Truly Serverless with high scalability
2. Asynchrony
3. Multi-agent and human-in-the-loop



Visual experience with
development and
operation

What Agent architecture works for me?



Which agent works for me?

Criteria	LangGraph	Step Functions as agents	Bedrock Agents
Architecture	Graph-based architecture	Workflow based architecture	LLM-based architecture
Integrations and tool support	Part of LangChain framework, extensive tool support options	11K AWS APIs, Public and private APIs and on-premise systems	Integration with external APIs using Actions, Action Groups powered by AWS Lambda
Memory and Context Management	Long-term, short-term and contextual memory with time travel features for debugging	State management and DynamoDB can be used as memory	Fully managed memory management with APIs, provided control and flexibility
Other Features	Excels in stateful applications	Fit for general purpose AI workflows . static agentic workflows, Agent orchestrations	Fit for single agent workflows
Ease Of Use	Provide granular control, has steep learning curve	Intuitive UI builder, Built-in retry, error handling and observability	Simple and opinionated agentic workflows
Operational responsibilities	Need to manage scale and deployment externally	Serverless	Serverless



Final thoughts on agents

- Agents are not drop-in upgrade for every use case
- Keep it simple
- Models are only as good as the context provided to them

“By using Agents for Amazon Bedrock we turned a labor-intensive, 80-hour manual process

// into a few minutes, freeing valuable resources to work on higher value tasks such as system and business process design.”

Shawn Swaner

CTO of Athene Holdings



Thank you!

Uma Ramadoss

Linkedin: uma-
ramadoss

