

Agentic AI on AWS

Amazon Bedrock AgentCore Workshop

Daniel Pruessner (dpkey@amazon.com), Solution Architect covering BNZ

Rob van der Harst (rhurst@amazon.com), Account Director covering BNZ



Whātuia te rangi e tū nei
Whātuia te papa e takoto nei
Korowaitia ki te kahu ora
Korowaitia ngā pito mata, ngā kura huna
Kia toitū
Kia toiroa
Kia toi kairangi
Kia puta, kia ora!

Hui e, tāiki e!

Tie together all that is above
Tie together all that is from below
Nurture with the cloak of life
Nurture, All Potential and aspiration, knowledge
and x-factor
To be permanent
To be sustainable
To be excellent
Release and thrive!
Bind and confirm!

wh = fa
ng = like the sound in sing
r = rolled r, like a soft d

a – as in but
e – as in vet
i – as in beat

o – as in walk
u – as in to

ā – as in bar
ē – as in dairy
ī – as in peel

ō – as in bore
ū – as in too



Agenda

Topic	Time	Duration(min)	Type
Welcome and Introduction (people arrive and have coffee)	9am	30	
Overview of Agentic AI	9:30am	15	Presentation
Use Case – Mortgage Assistant and Introduction to Strands	9:45am	15	Presentation
Running Strands Agent on local environment	10am	30	Lab
Morning Tea	10:30am	20	
Introduction to AgentCore Runtime	10:50am	15	Presentation
AgentCore Runtime Lab	11:05am	25	Lab
Introduction to AgentCore Identity	11:30am	15	Presentation
Introduction to AgentCore Gateway	11:45am	15	Presentation
Lunch	12pm	60	
AgentCore Gateway and Identity Lab	1pm	25	Lab
Introduction to AgentCore Memory	1:45pm	20	Presentation
AgentCore Memory Lab	2:05pm	20	Lab
Introduction to AgentCore Observability	2:40pm	15	Presentation
AgentCore Observability Lab	2:55pm	15	Lab
Introduction to AgentCore Tools	3:10pm	10	Presentation
AgentCore Tools Lab	3:25pm	20	Lab
Trivia/Quiz	3:45pm	10	

Organizations are creating value with agentic AI

Workplace productivity



Ex. Knowledge worker productivity,
Software development

Business workflows

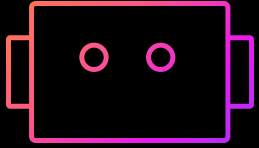


Ex. Customer experience, incident
management, demand forecasting

Innovation and research



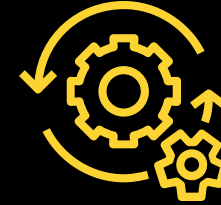
Ex. Automate complex data analysis
and simulations



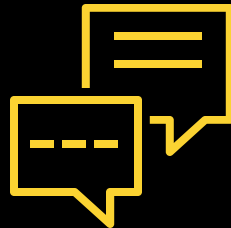
What is Agentic AI?



Intelligent,
autonomous
systems



Plan, reason,
and act

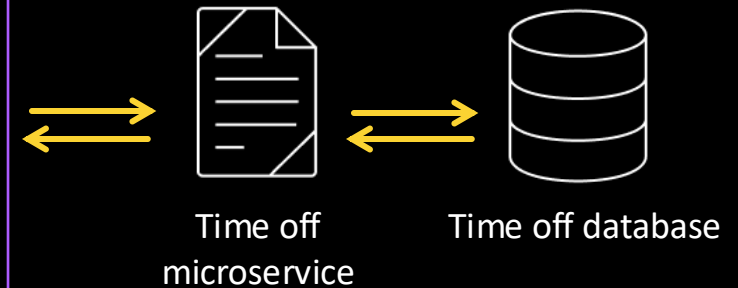
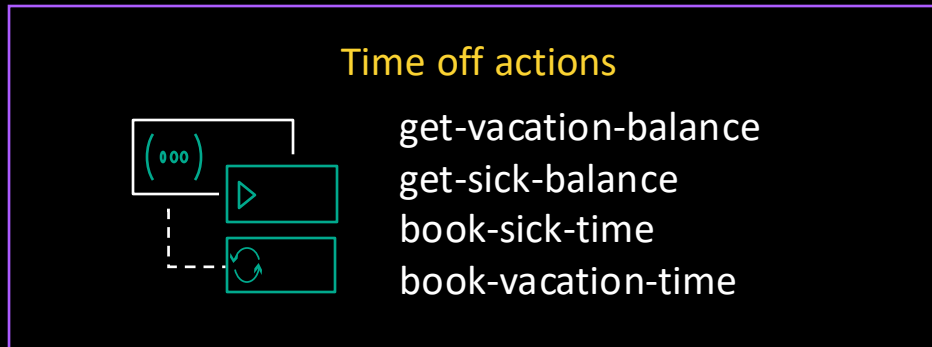


Access to
enterprise data

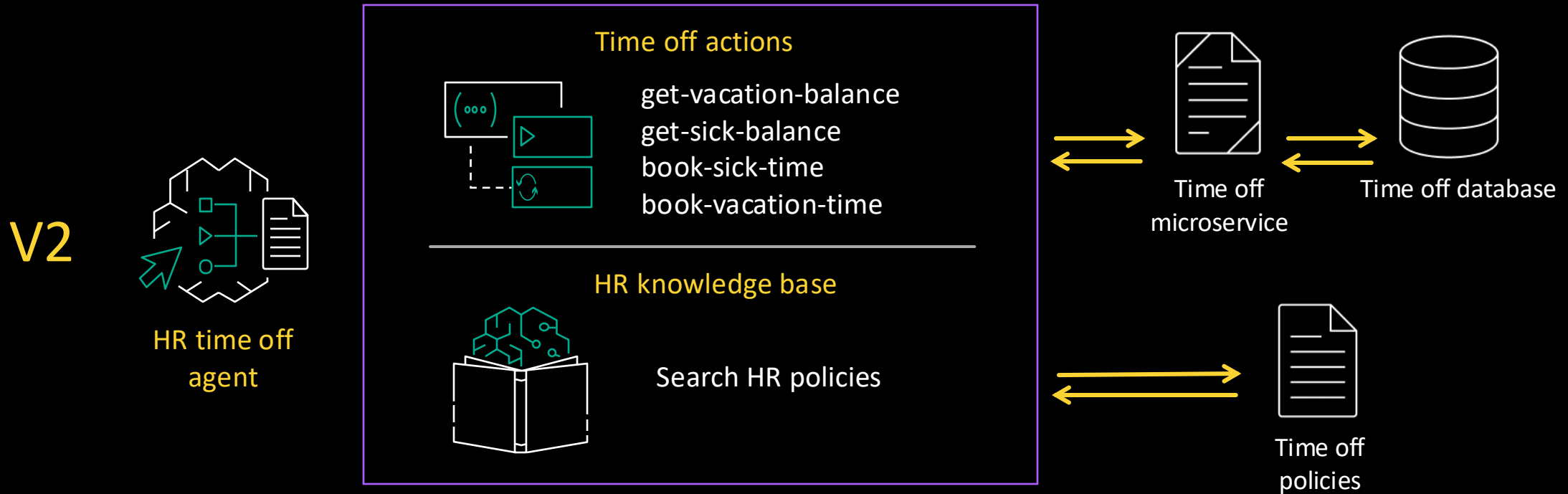


Ability to use
tools

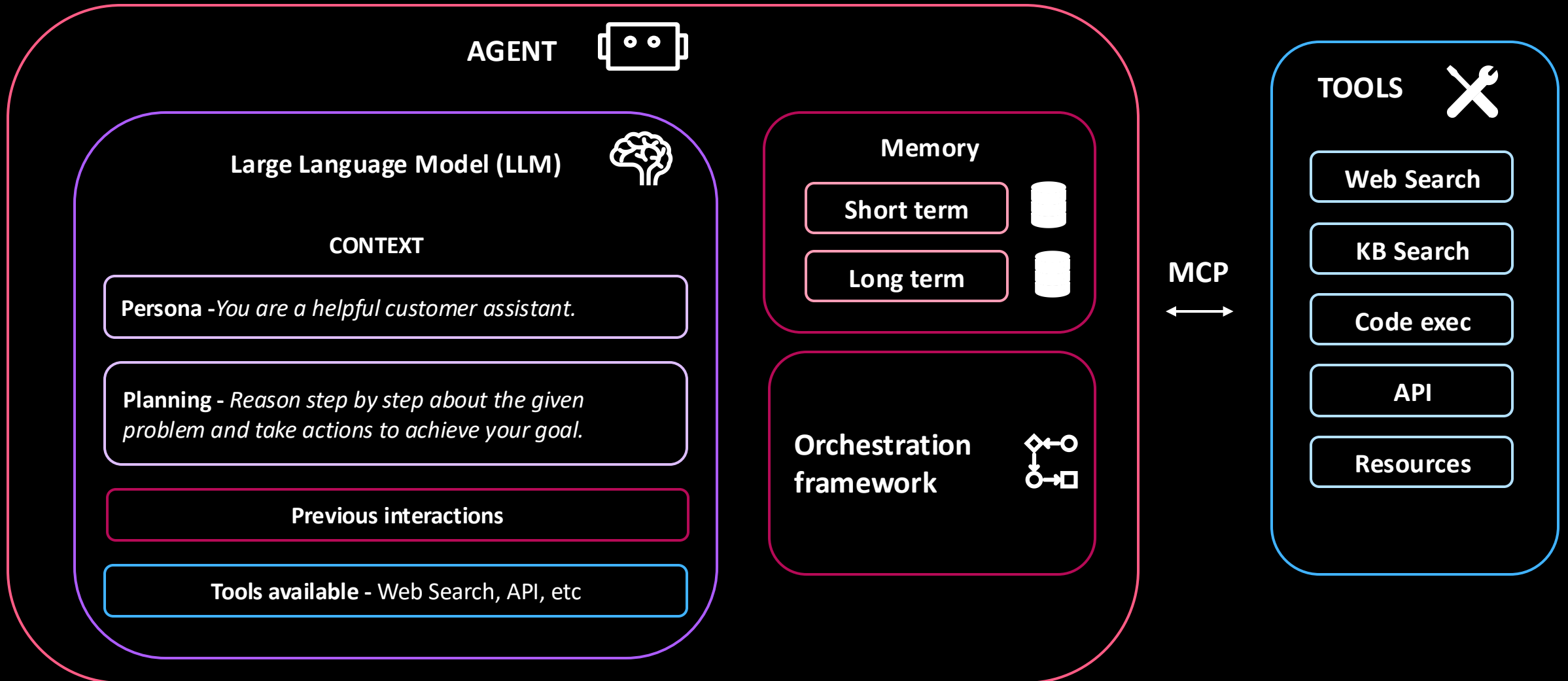
Agents can start small and focused . . .



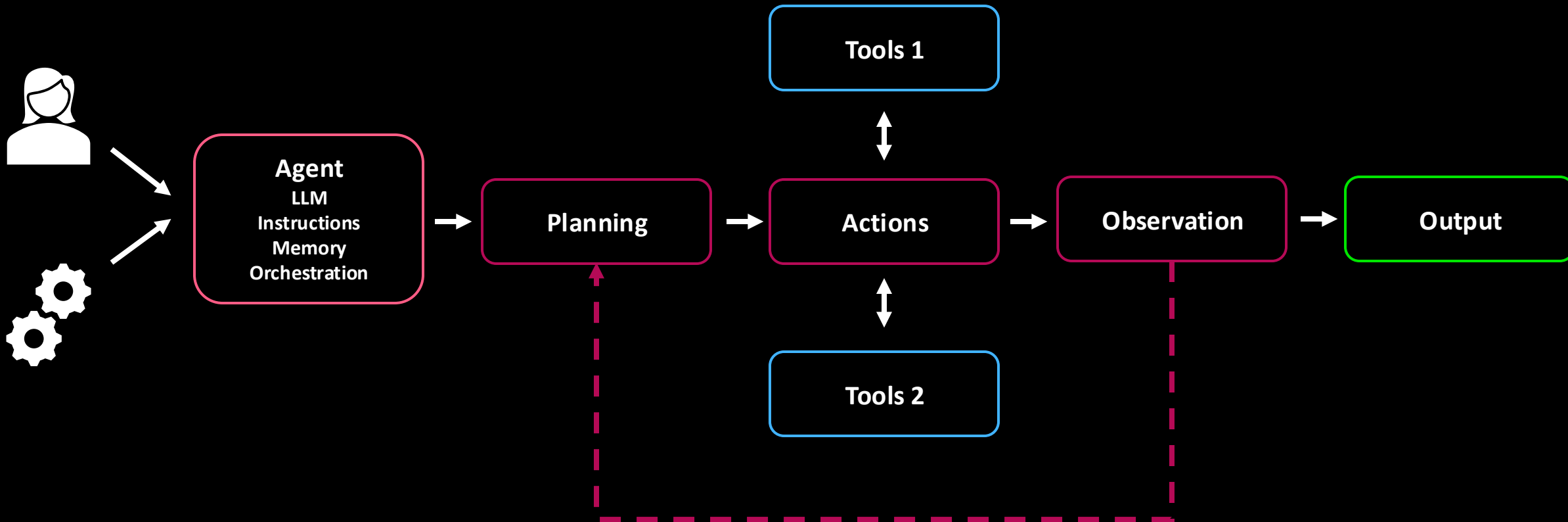
... and can be easily expanded



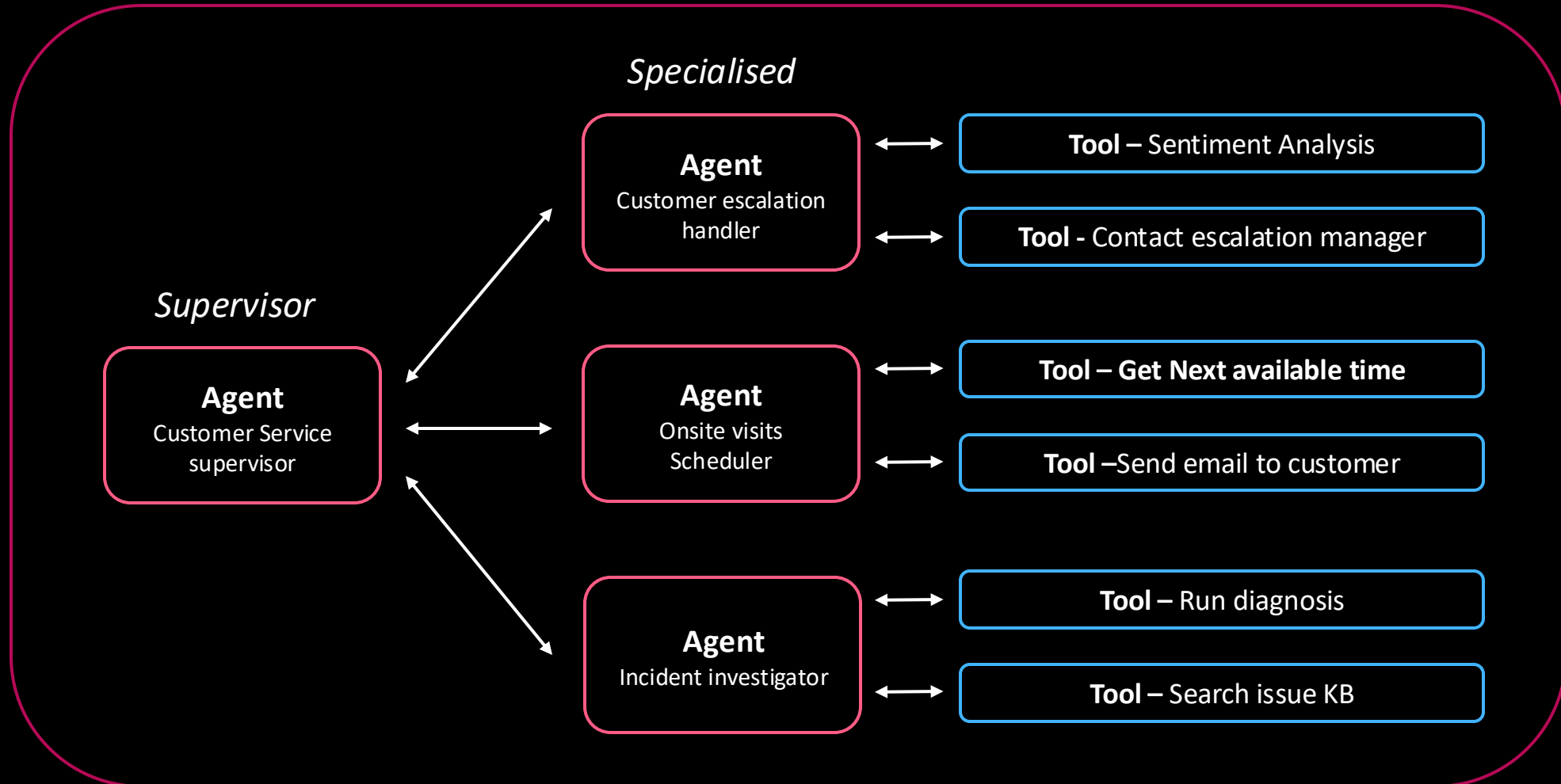
Inside view of an agentic system



Simple execution of a single agent

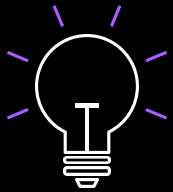


Multi-Agents system examples



The prototype to production “chasm”

Excitement
and potential



POC

Challenges on the path to production



Performance



Scalability

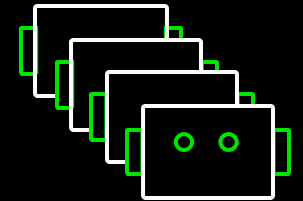


Security



Governance

Meaningful
business value



AI production
agents



Agentic AI Primitives



Open source

Protocols:

MCP, A2A (coming soon)

Frameworks:

CrewAI, LangGraph,
Strands Agents



Amazon Bedrock

AgentCore



Runtime



Identity



Gateway



Code interpreter



Memory



Browser tool



Observability

Model Capabilities



Model access



Cost and performance
optimization



Customization



Guardrails

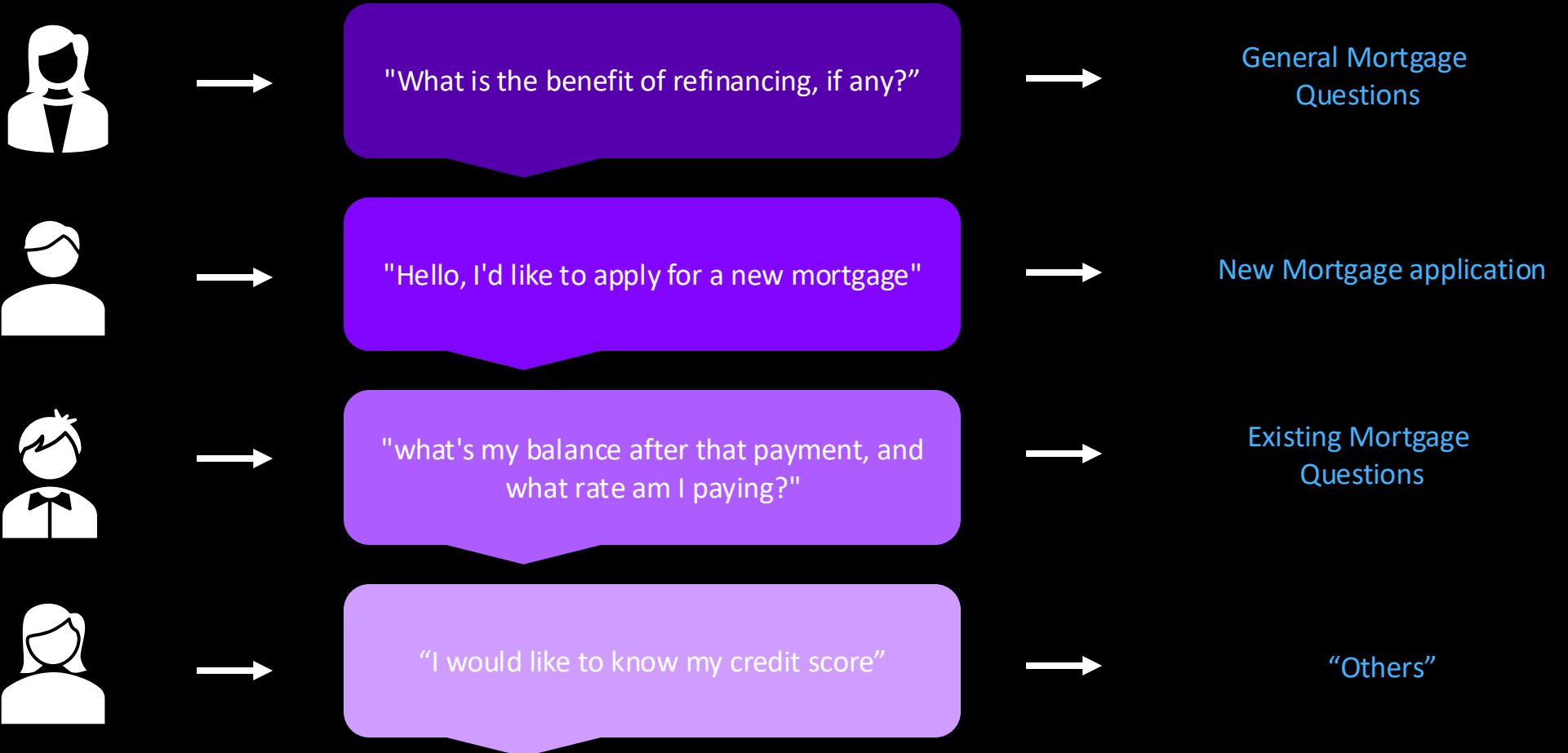
Knowledge Bases



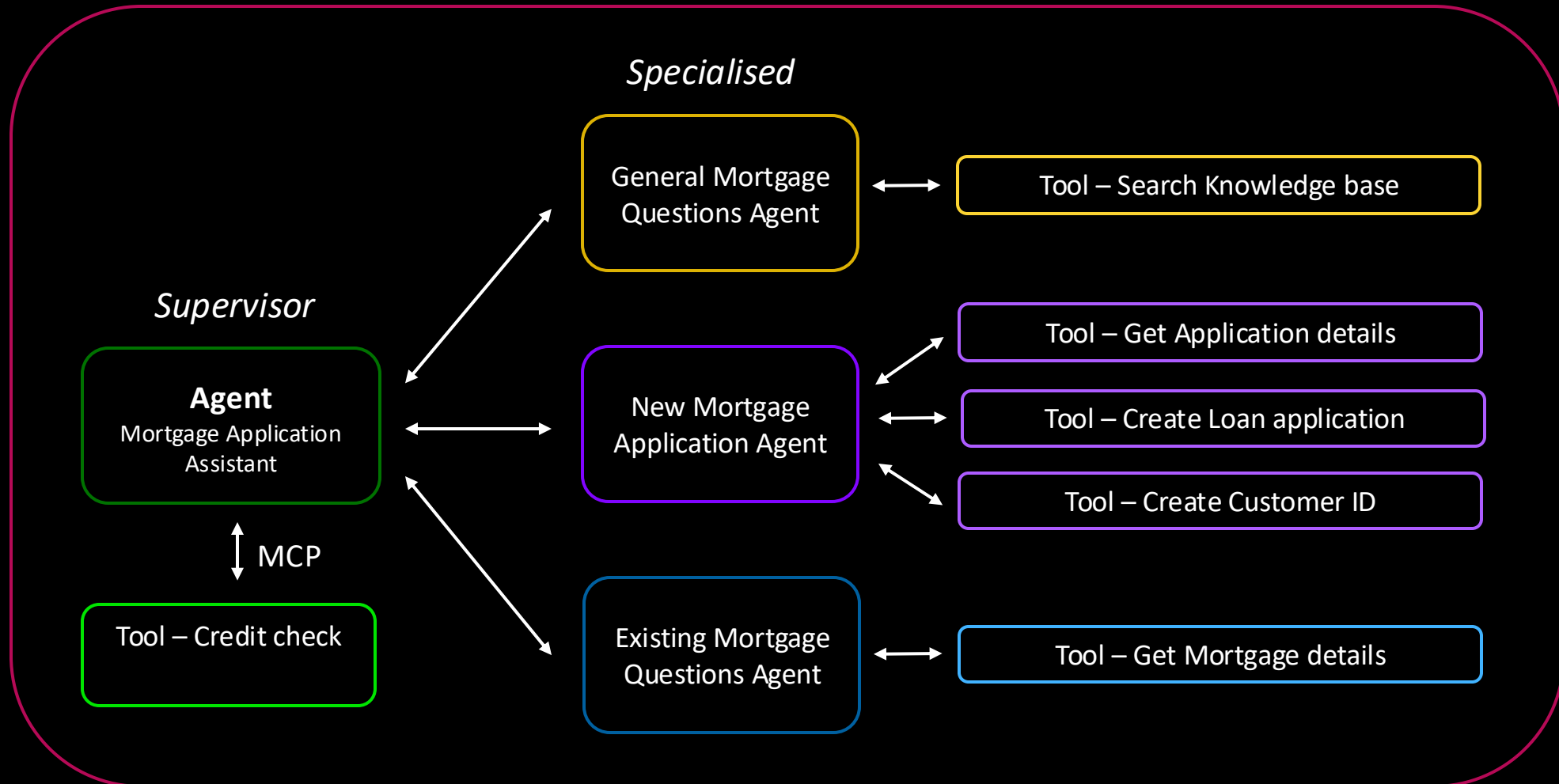
The use case we're solving for

Mortgage Chatbot Assistant

The bank's Mortgage department receives a flood of questions daily.

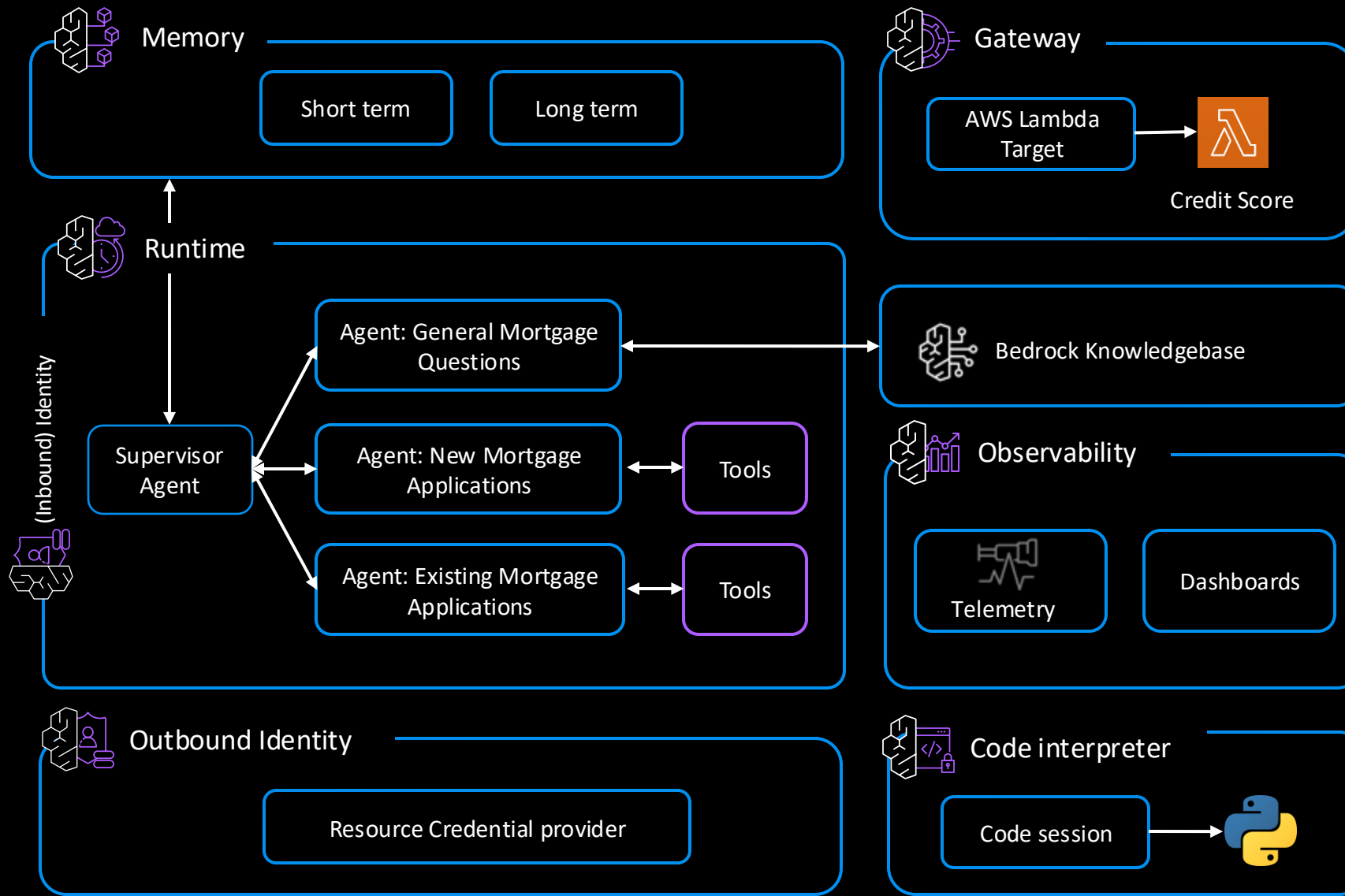


Our Multi-Agents design – Agents as Tools pattern

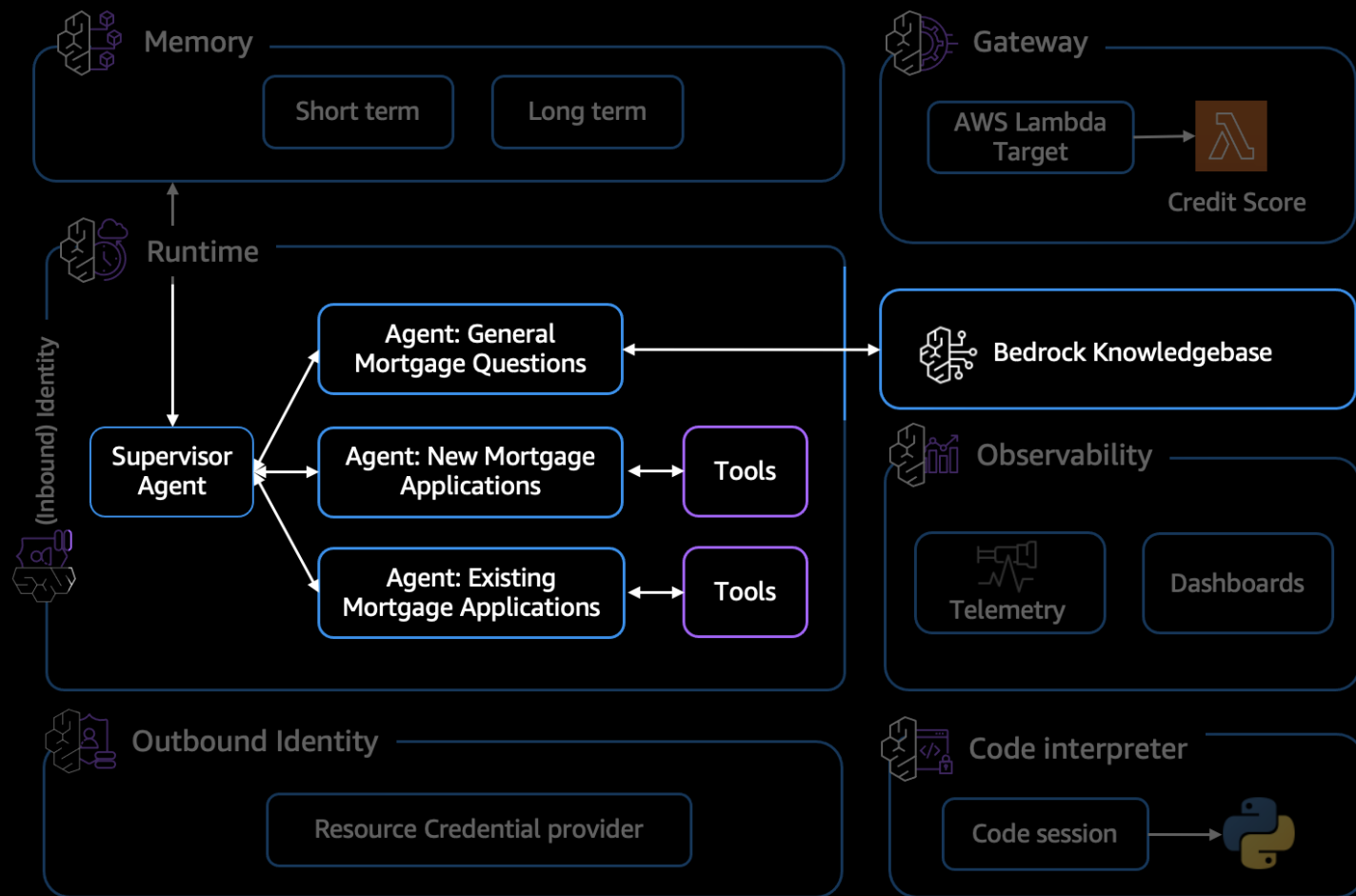




What we're building – step by step



- Step 1 – Set up, KB
- Step 2 – Strands Agent
- Step 3 – Runtime
- Step 4 – Gateway
- Step 5 - Identity
- Step 6 – Memory
- Step 7 – Observability
- Step 8 – Tools



Step 1 – Set up, KB

Step 2 – Strands Agent

Step 3 – Runtime

Step 4 – Gateway

Step 5 (Optional)– Identity

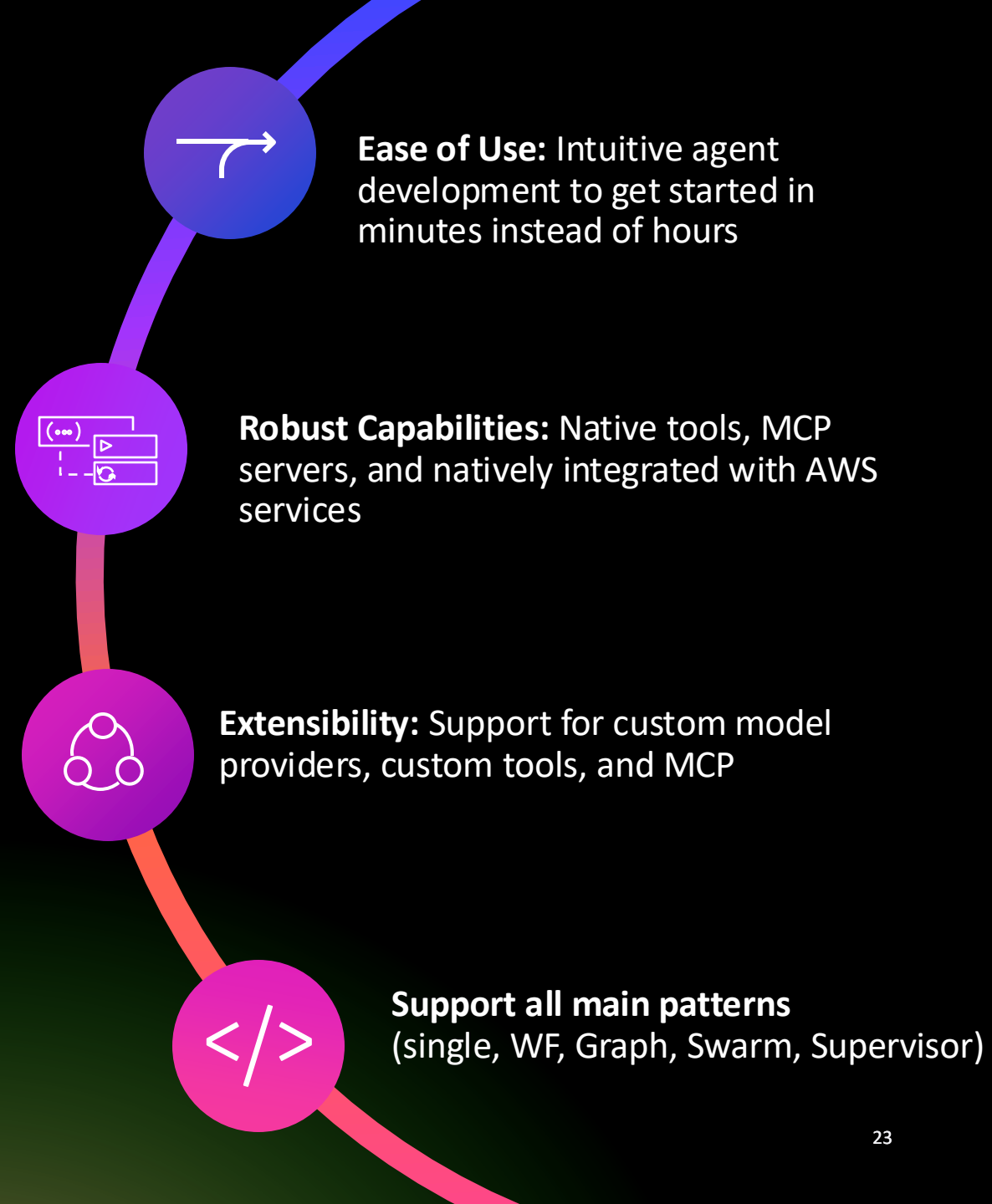
Step 6 – Memory

Step 7 – Observability

Step 8 (Optional) – Tool

Strands Agents

Strands Agents is an open source python SDK for building agents using just a few lines of code



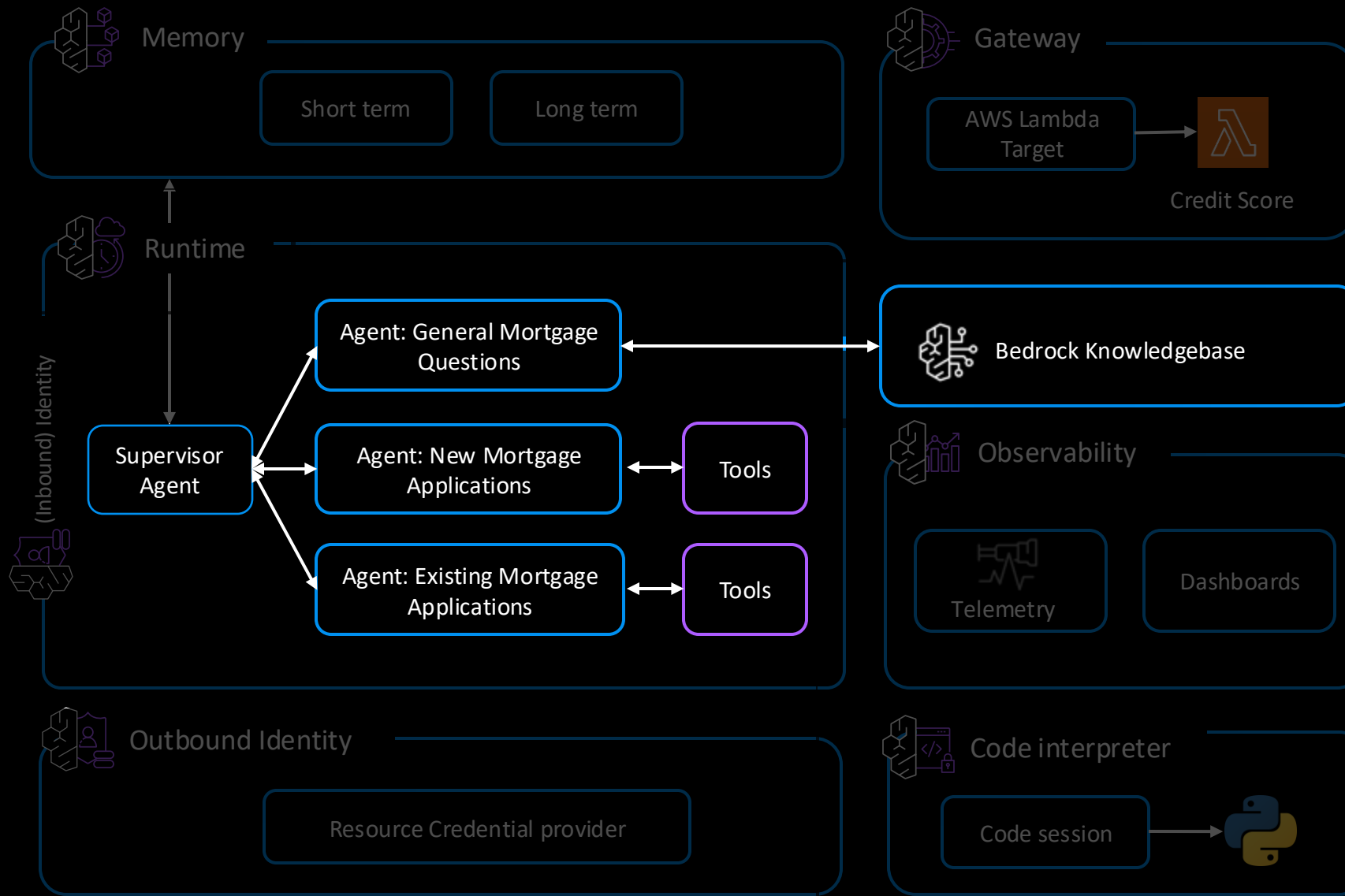


Lab 01 and Lab 02

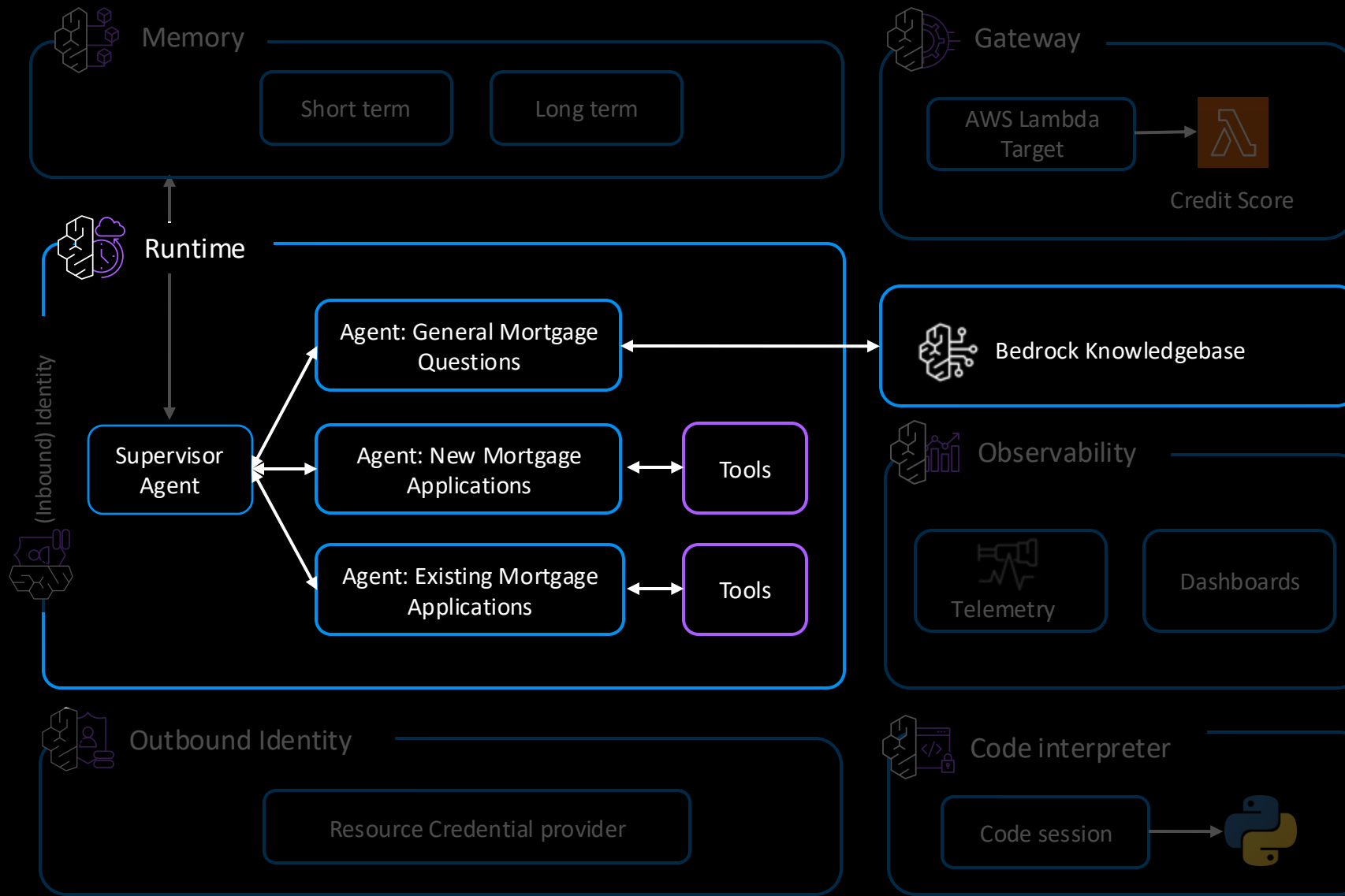


AgentCore Runtime





- Step 1 – Set up, KB
- Step 2 – Strands Agent
- Step 3 – Runtime
- Step 4 – Gateway
- Step 5 - Identity
- Step 6 – Memory
- Step 7 – Observability
- Step 8 – Tools

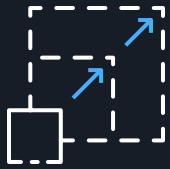


- Step 1 – Set up, KB
- Step 2 – Strands Agent
- Step 3 – Runtime
- Step 4 – Gateway
- Step 5 - Identity
- Step 6 – Memory
- Step 7 – Observability
- Step 8 – Tools



AgentCore Runtime

Scale from real-time to multi-hour workloads



- Multi-modal
- Real time & long running (8h)
- Auto-scaling

Accelerate time to market



- Any frameworks
- Any models
- Starter toolkit to speed up deployment

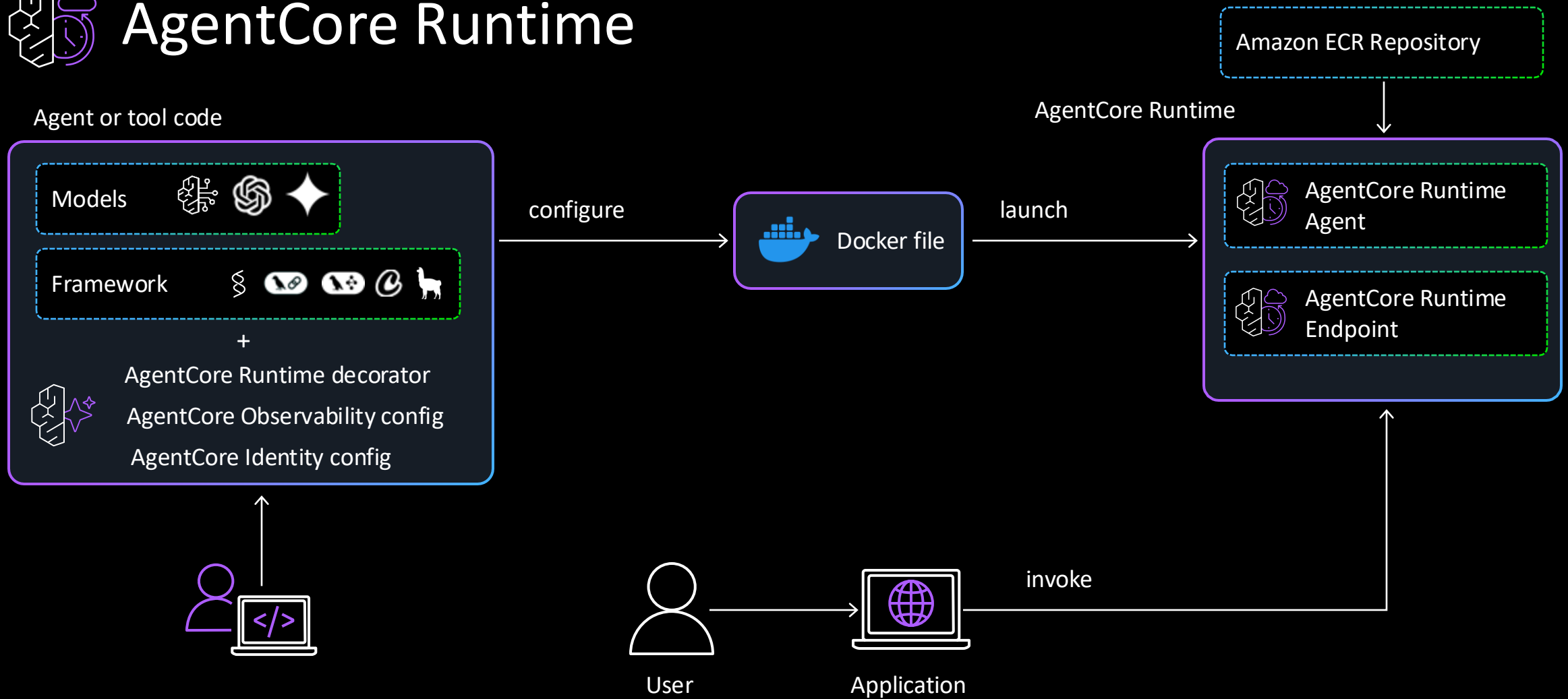
Secure workload with enterprise-grade isolation



- MicroVMs
- True session isolation to protect your data
- Integrates with existing identity providers



AgentCore Runtime





AgentCore Runtime

```
● ● ●

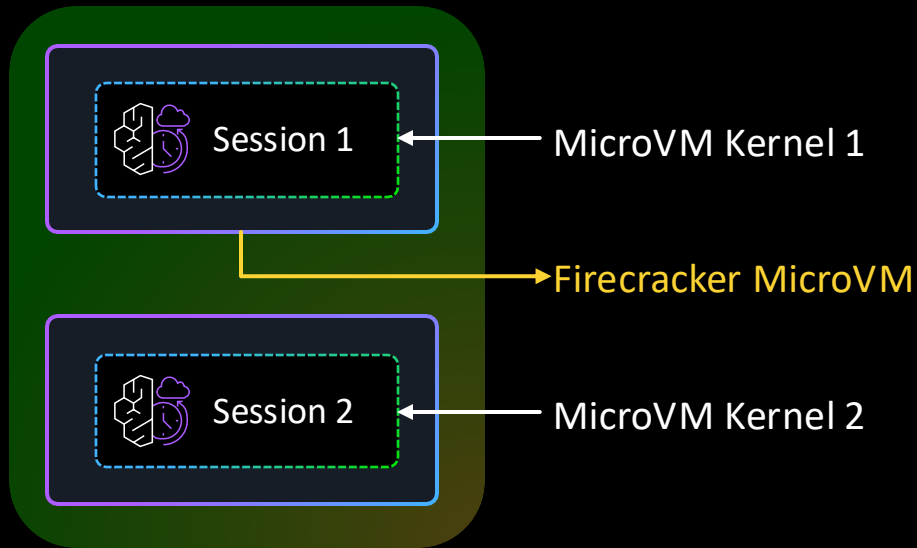
# Build your agent with the SDK
from bedrock_agentcore import BedrockAgentCoreApp

app = BedrockAgentCoreApp()

@app.entrypoint
def my_agent(request):
    # Implement agent logic
    return response

app.run()
```

AgentCore Runtime *True* Session Isolation



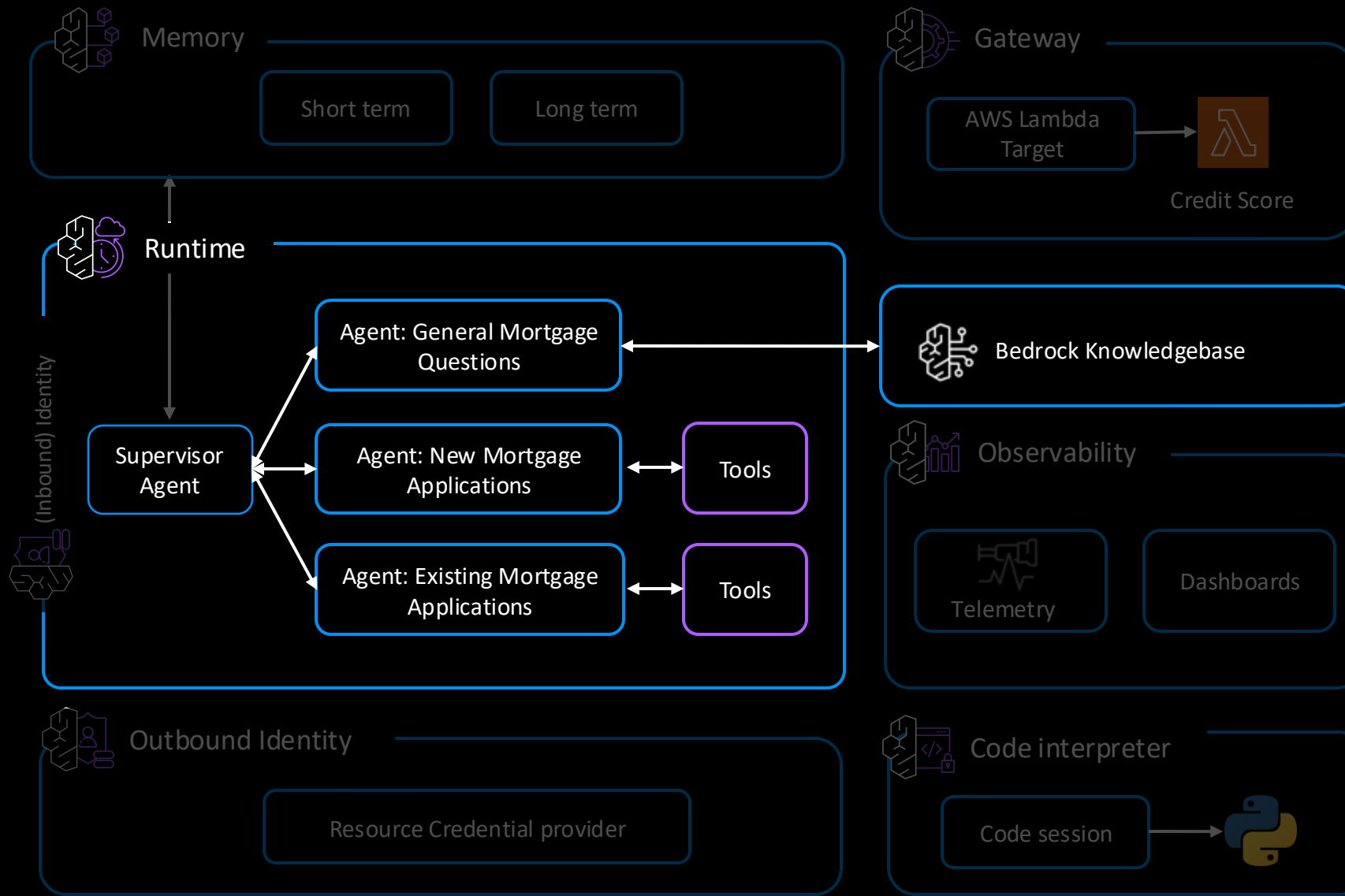
- Each session runs in a completely isolated microVM (compute + memory + filesystem)
- With other serverless offerings, multiple sessions *may* execute in the same microVM
 - ⚠ Without session isolation, local files and state could be accessed across sessions
- Stateful - preserves state securely within a session
- Use Agentcore Memory for out-of-session state, short and long term



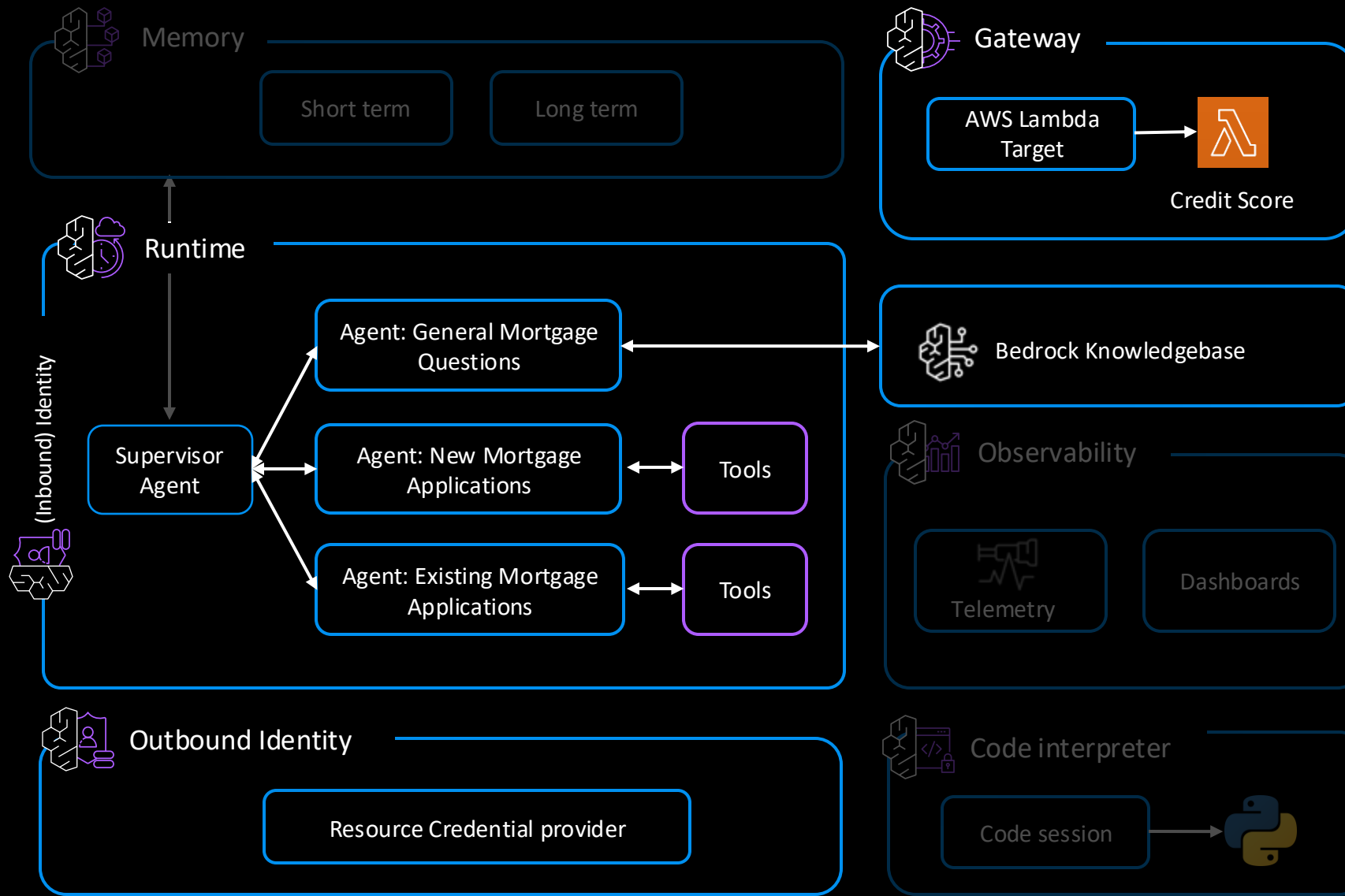
LAB #03-agentcore-runtime



AgentCore Identity

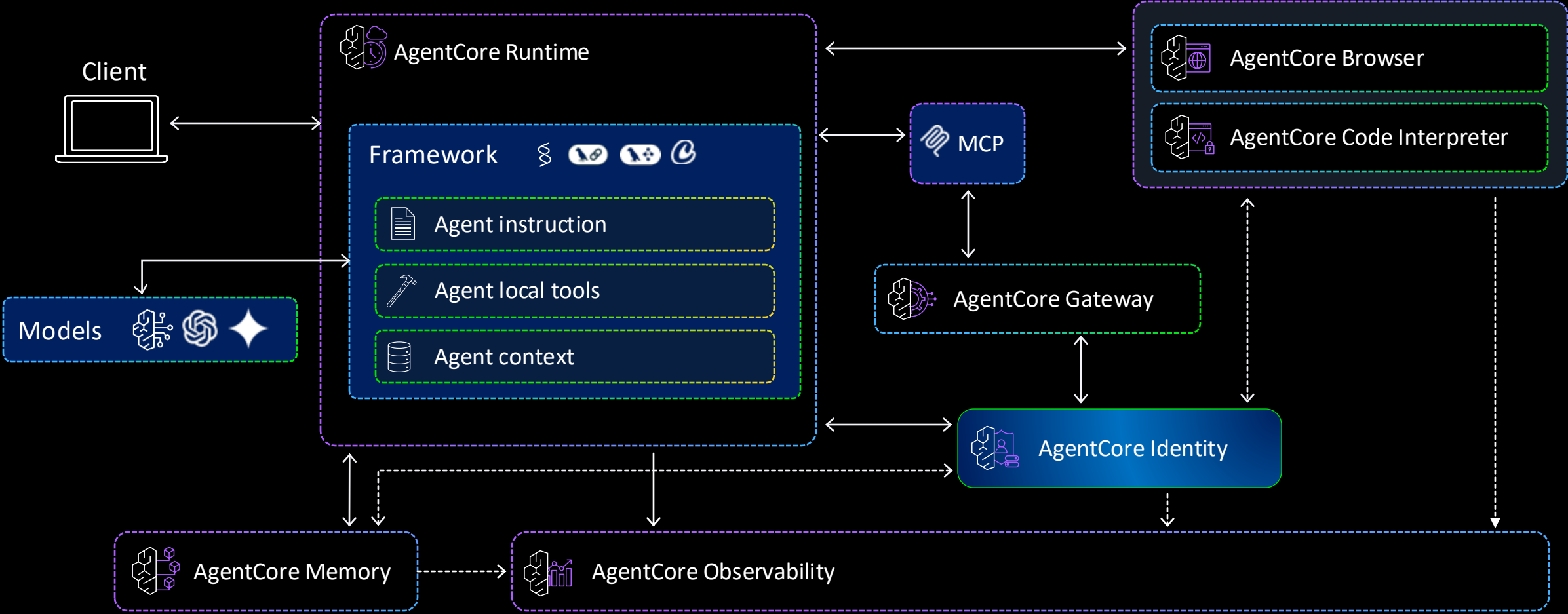


- Step 1 – Set up, KB
- Step 2 – Strands Agent
- Step 3 – Runtime
- Step 4 – Gateway
- Step 5 - Identity
- Step 6 – Memory
- Step 7 – Observability
- Step 8 – Tools



- Step 1 – Set up, KB
- Step 2 – Strands Agent
- Step 3 – Runtime
- Step 4 – Gateway
- Step 5 - Identity
- Step 6 – Memory
- Step 7 – Observability
- Step 8 – Tools

Amazon Bedrock AgentCore



A recap of concepts

Authentication

Proving the user/agent/entity is who they say they are

Authorisation

Verifying the user/agent/entity is allowed to perform an action

OAuth

An open standard for access delegation

JWT

JSON Web Token – contains information about identity and other attributes

IdP

Identity Provider – issues tokens

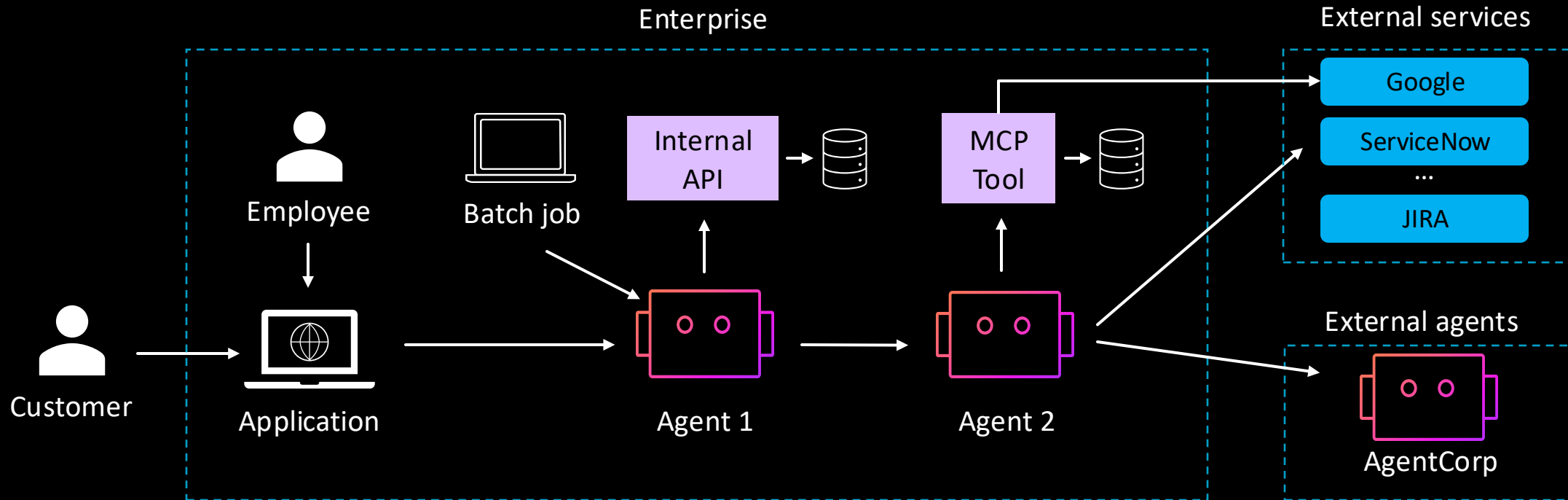
2LO

Client Credentials Flow - machine to machine auth.

3LO

Authorization Code Grant – used to auth with user context

Agents Present Identity Challenges



Access control,
enforcement

Authorization,
delegation

Manage identity
lifecycle, scale

SaaS, third-party
integration

Governance,
compliance



AgentCore Identity

Managed agent identity and access management

Secure, delegated access for AI agents



- Assigns distinct agent identities for secure agent operations at scale
- Enables AI agents to securely access AWS resources and third-party tools and services such as GitHub, Google, Salesforce and Slack

Build streamlined AI agent experiences



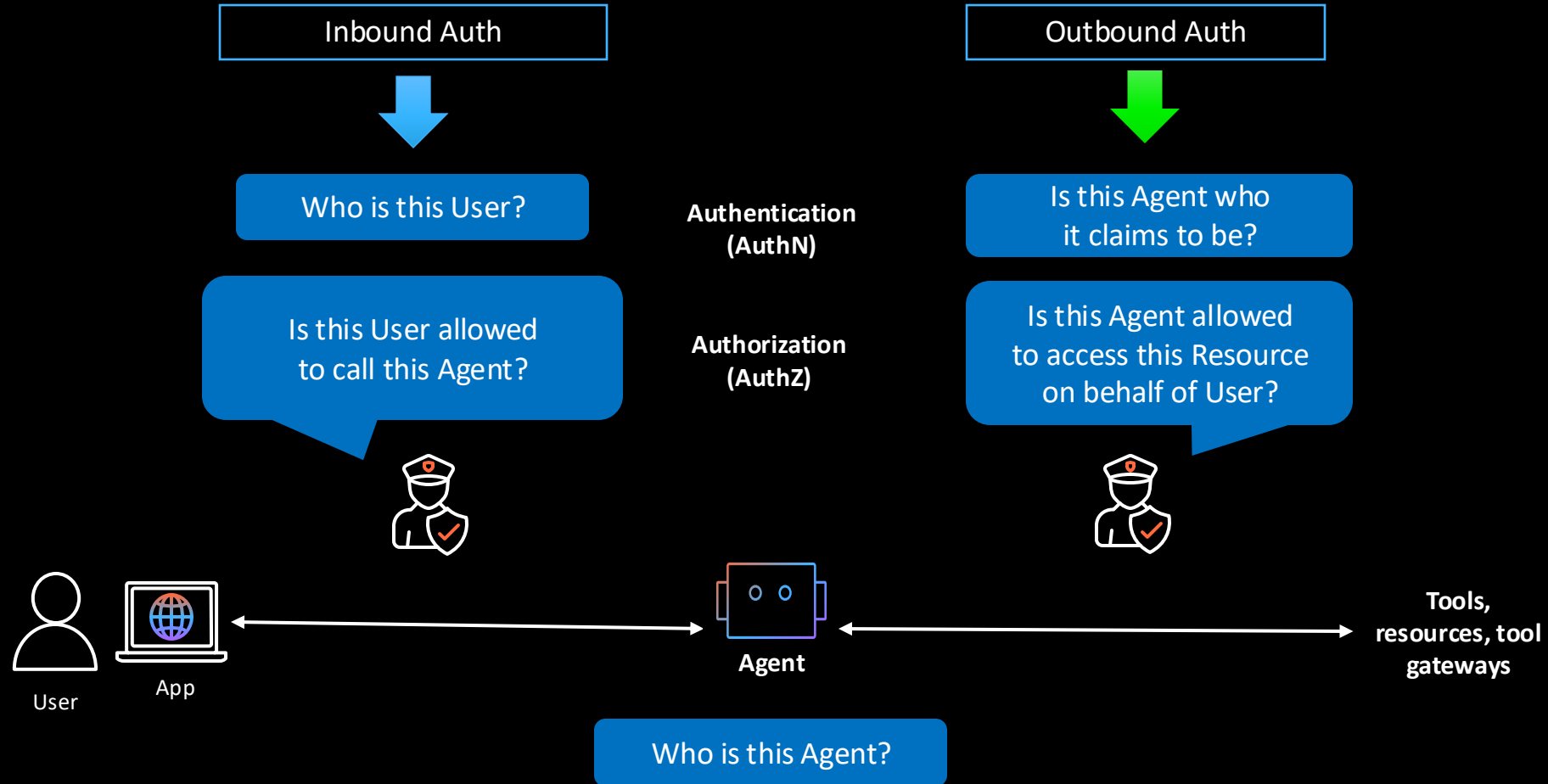
- Minimizes consent fatigue with a secure token vault
- Streamlines authentication flows and secure credential management
- Offers complete visibility through comprehensive audit trails

Accelerated AI agent development



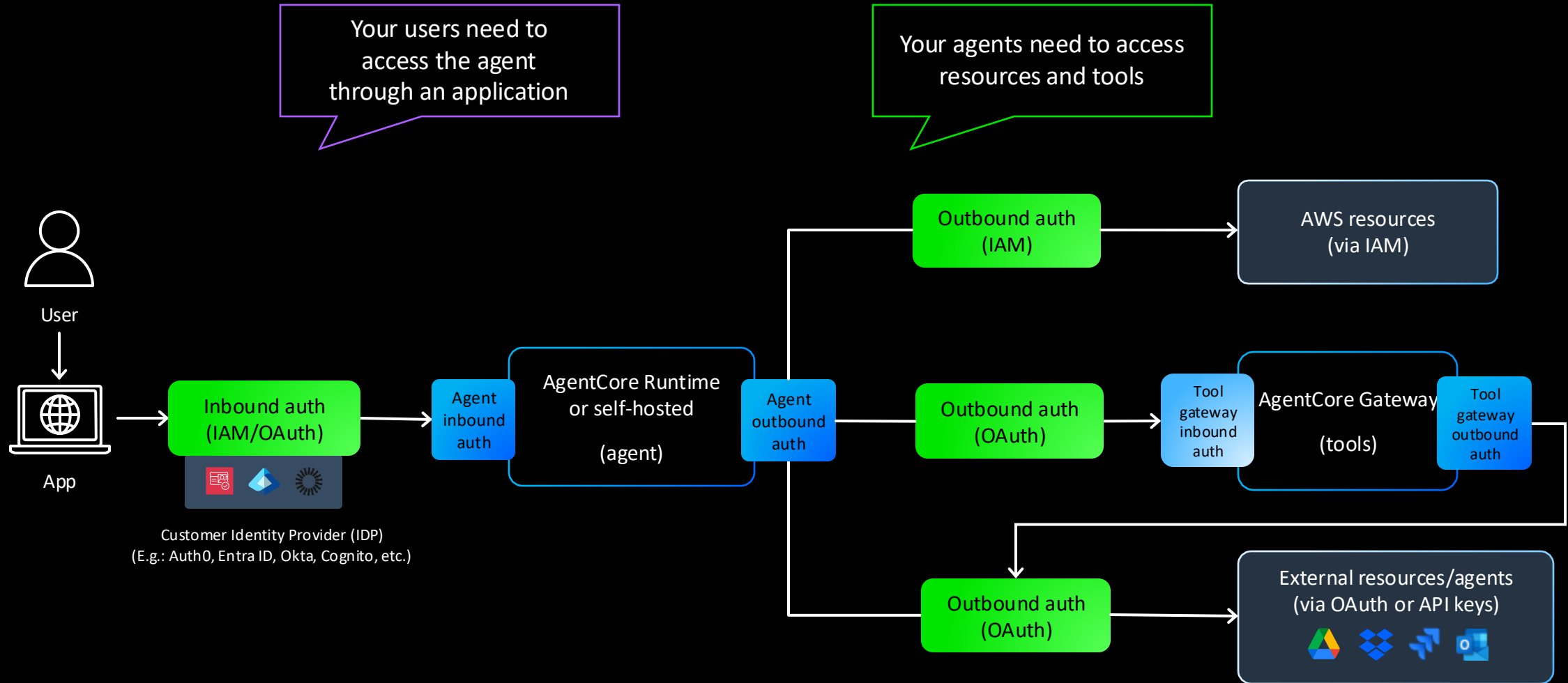
- Seamless integration with existing identity systems such as Okta, Azure Entra ID, or Amazon Cognito
- Lowers custom development efforts without need for migrating users or rebuilding authentication flows

Agentic AI Auth Basics





Auth with AgentCore Identity



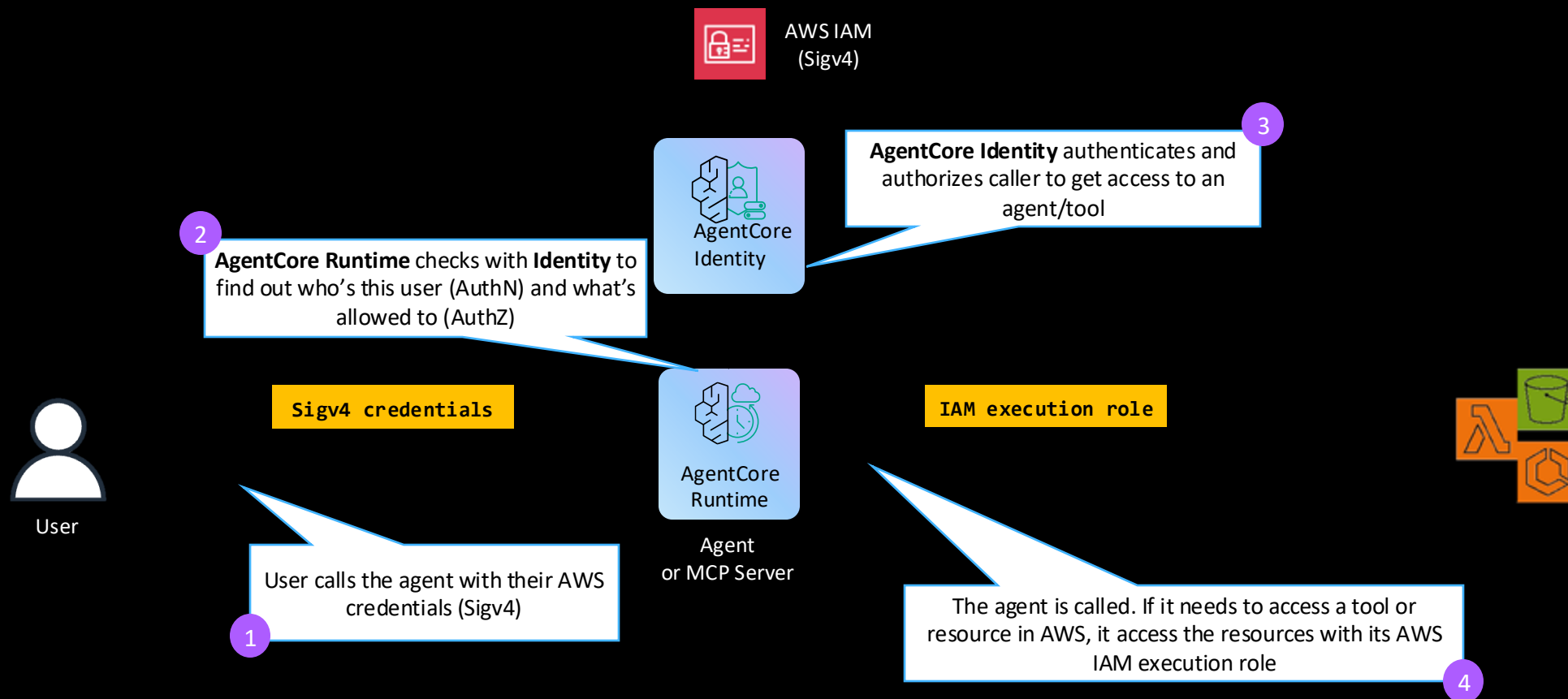
How it works – AgentCore Identity Scenario 1



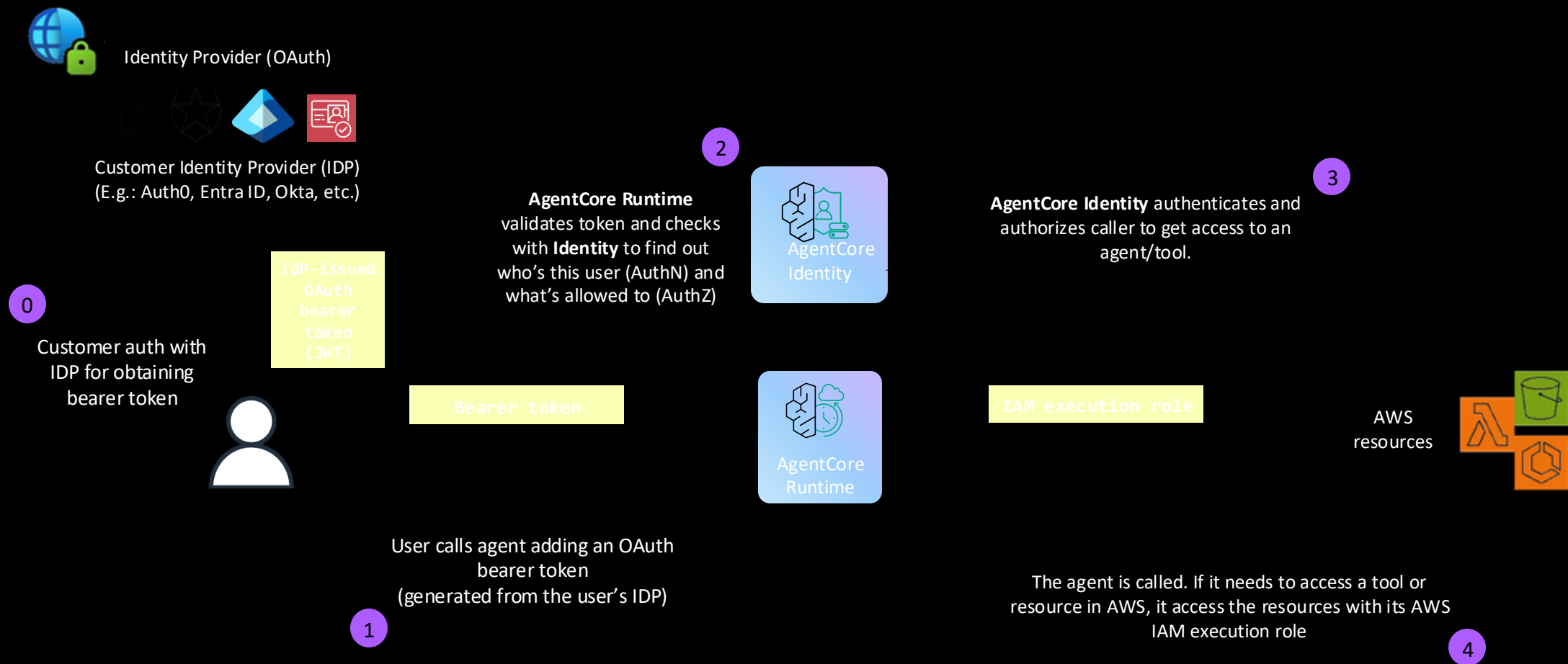
Agent/tool hosted in AgentCore Runtime accessing resources in AWS

*Sigv4 = [AWS Signature Version 4](#)

- **Inbound auth method:** User identity authorization method AWS IAM (Sigv4 credentials)
- **Outbound auth method:** Resources access via AWS IAM role



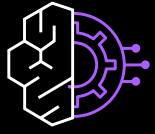
How it works – AgentCore Identity scenario 2





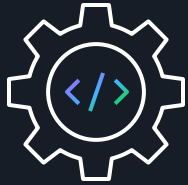
AgentCore Gateway





AgentCore Gateway

Simplified tool development & integration



- Turn APIs, Lambda functions, and existing services into MCP-compatible tools
- Access thousands of tools through a standardized interface

Secure and unified access



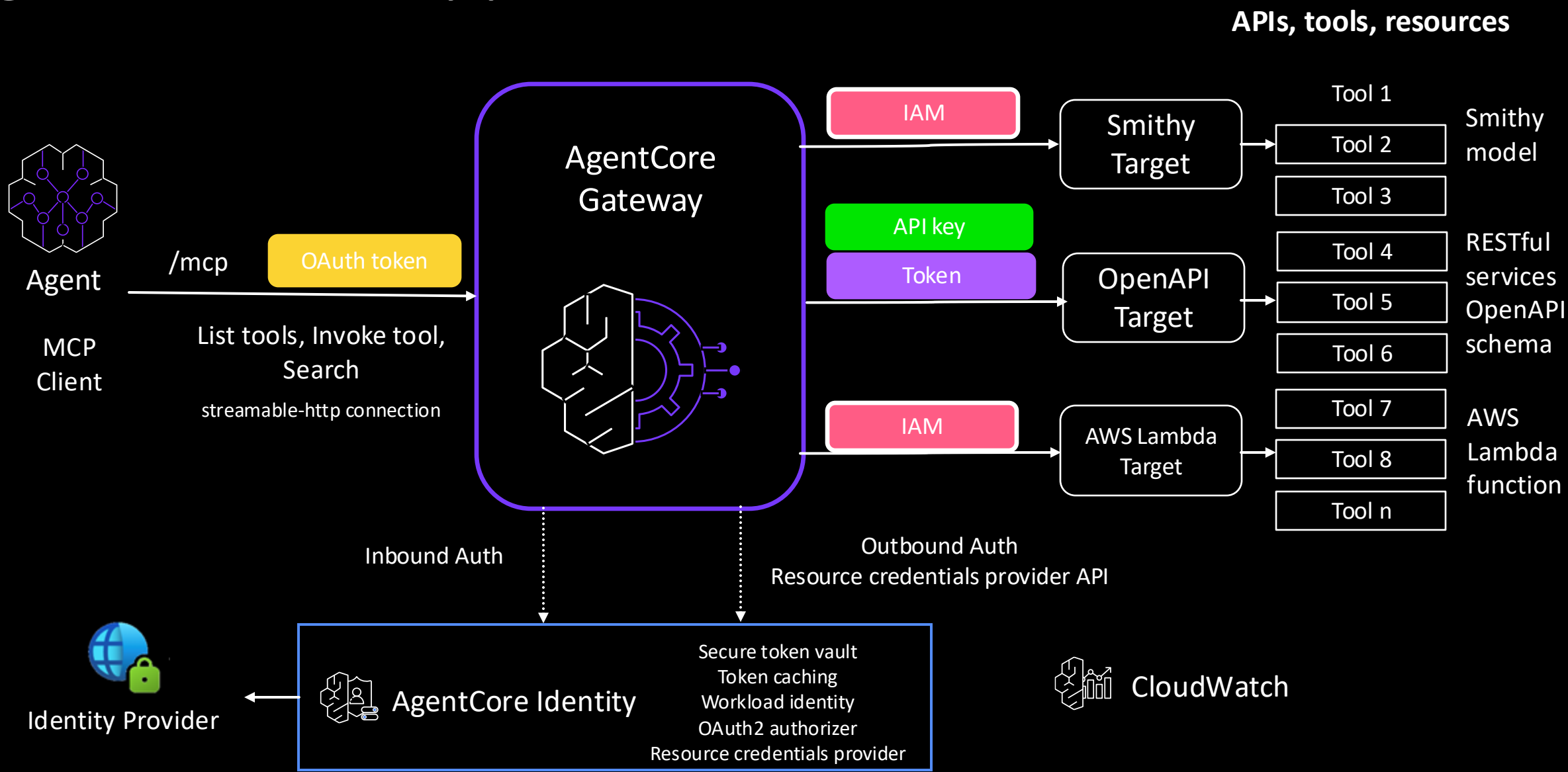
- Discover and use tools through a single, secure endpoint
- Combine multiple tools sources into one unified interface

Intelligent tool discovery

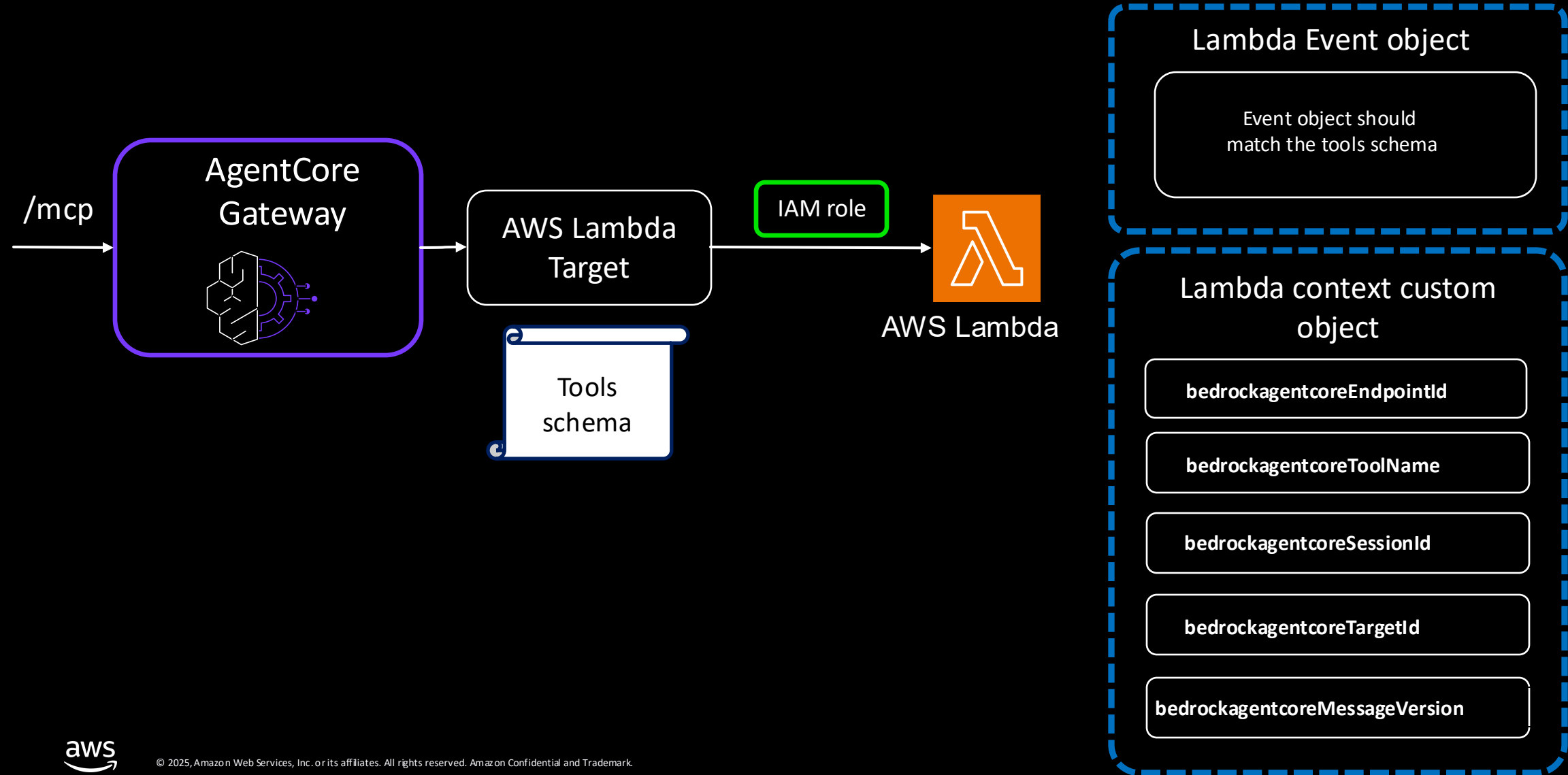


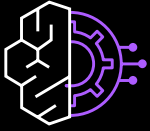
- Enable agents to find the right tools with context aware discovery
- Curated tool collections with granular permissions

AgentCore Gateway provides secure access

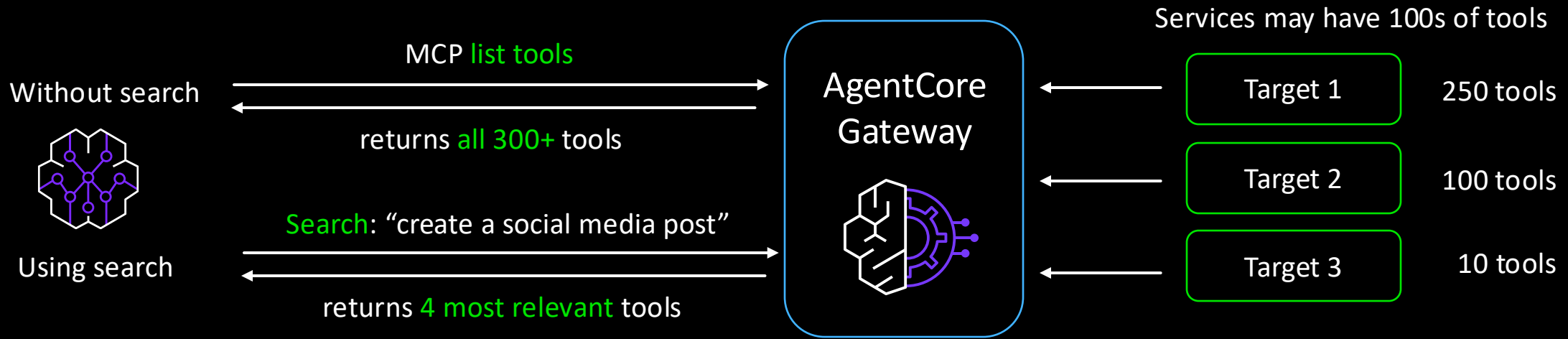


AWS Lambda function tools for Bedrock AgentCore Gateway





AgentCore Gateway semantic search



Benefits

- AgentCore Gateway automatically indexes tools, and gives serverless semantic search
- Reduces context passed to the agent's LLM, improving accuracy, speed, and cost
- Lets agent focus on tools relevant for a given task



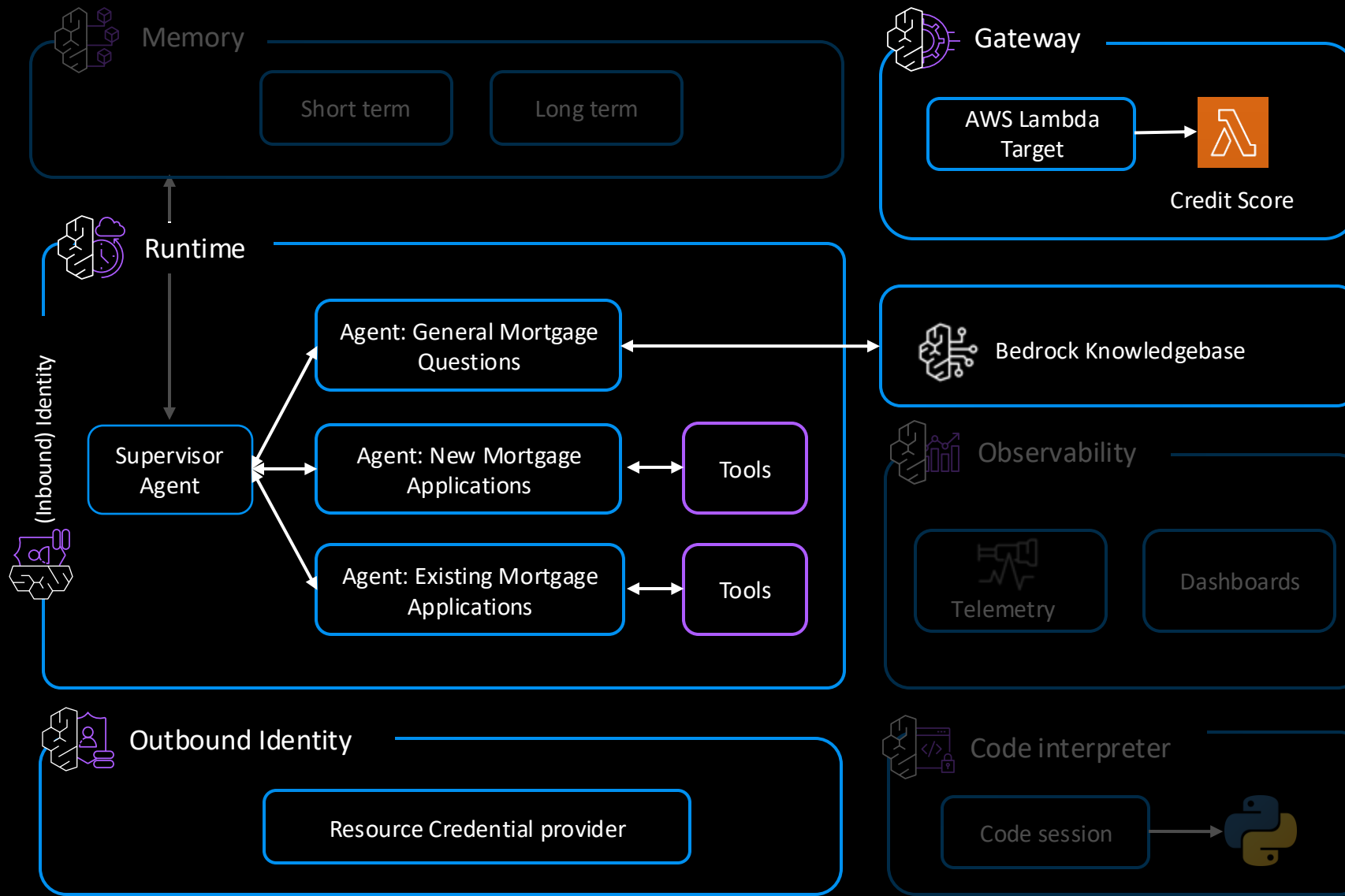
LAB #04-agentcore-gateway



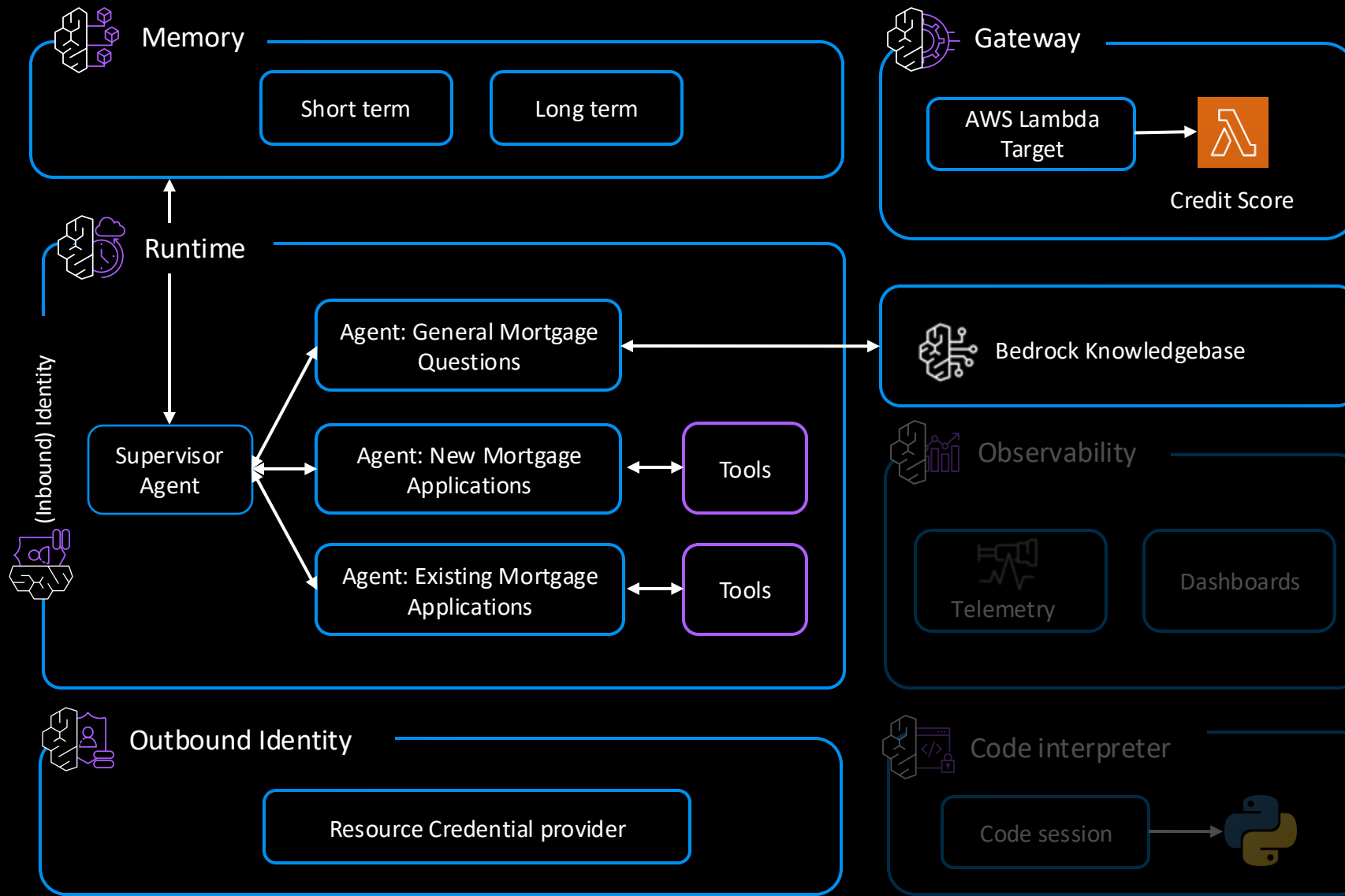
LAB #05-agentcore-identity



AgentCore Memory



- Step 1 – Set up, KB
- Step 2 – Strands Agent
- Step 3 – Runtime
- Step 4 – Gateway
- Step 5 - Identity
- Step 6 – Memory
- Step 7 – Observability
- Step 8 – Tools



Agent experience without memory

User: “Hey, I’m trying to book a flight”

AI: “Sure I can help, what is your destination and dates?”

User: “From New York to London, 8 December”

AI: “The flight time from New York to London is 7 hours. December is a wonderful time to visit!”

User: “No, I want to *book* the flight, not get facts about it.”

AI: “Sure I can help, what is your destination and dates?”



AgentCore Memory

Simplify memory management



- Abstracts memory infrastructure
- Scales automatically with serverless architecture
- Automatically stores and manages agent context across sessions

Enterprise-grade



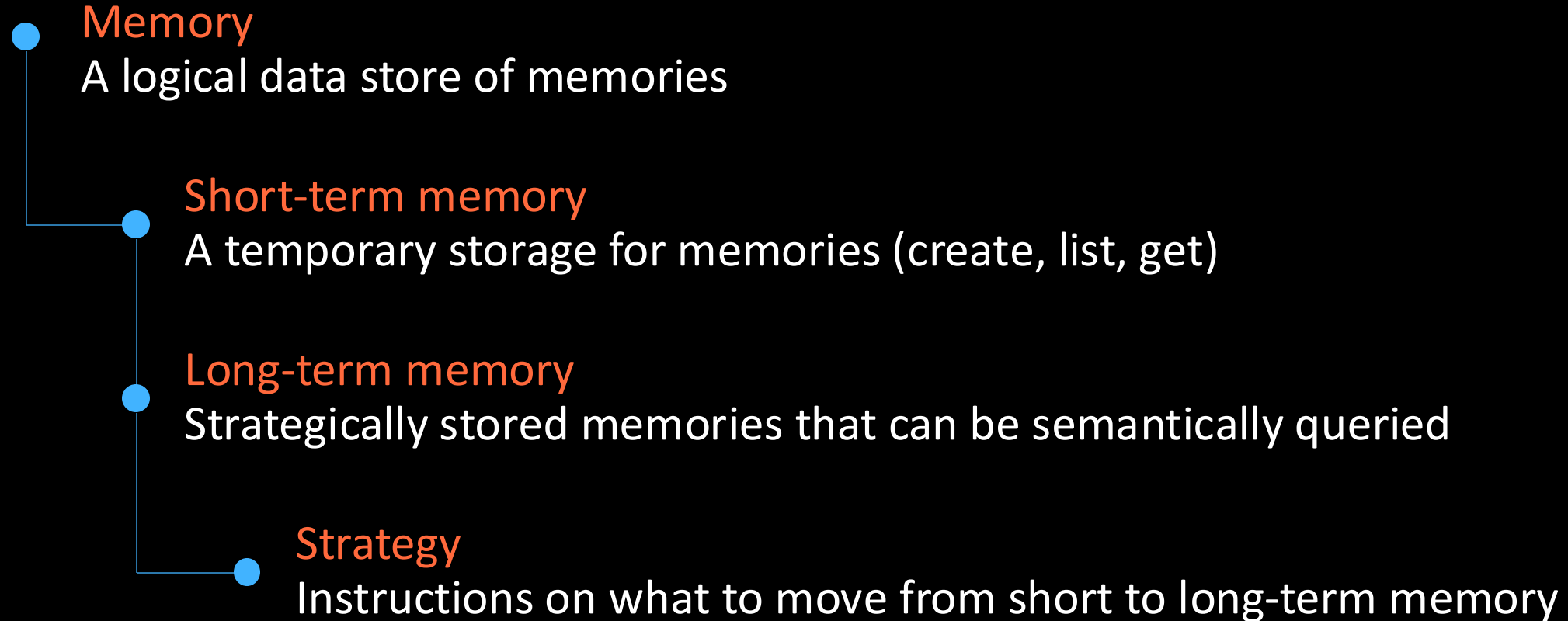
- Complete **data privacy** with **dedicated storage** for each customer
- Enterprise security with **encryption** and regional data storage options

Deep customization



- Define memory patterns based on your use case
- Configure extraction rules
- Choose models and customize prompts for memory extraction

AgentCore Memory - key concepts



Logical isolation of memory data: short-term memory

memory_id – the memory resource in your AWS account

actor_id – entities in your system (users, agents, project, or combinations),

session_id – a session of related events

Logical isolation of memory data: long-term memory

Namespaces

A user-defined path where memories are organised and stored **per strategy**.

Use built-in variables for dynamic namespaces:

- {actorId}
- {sessionId}
- {memoryStrategyId}

Examples:

Store everything under actorId → “/{actorId}”

Store records per actor and session → “/{actorId}/{sessionId}”

Store records food preferences per actor → “/{actorId}/food_preferences”

Logical isolation of memory data: long-term memory

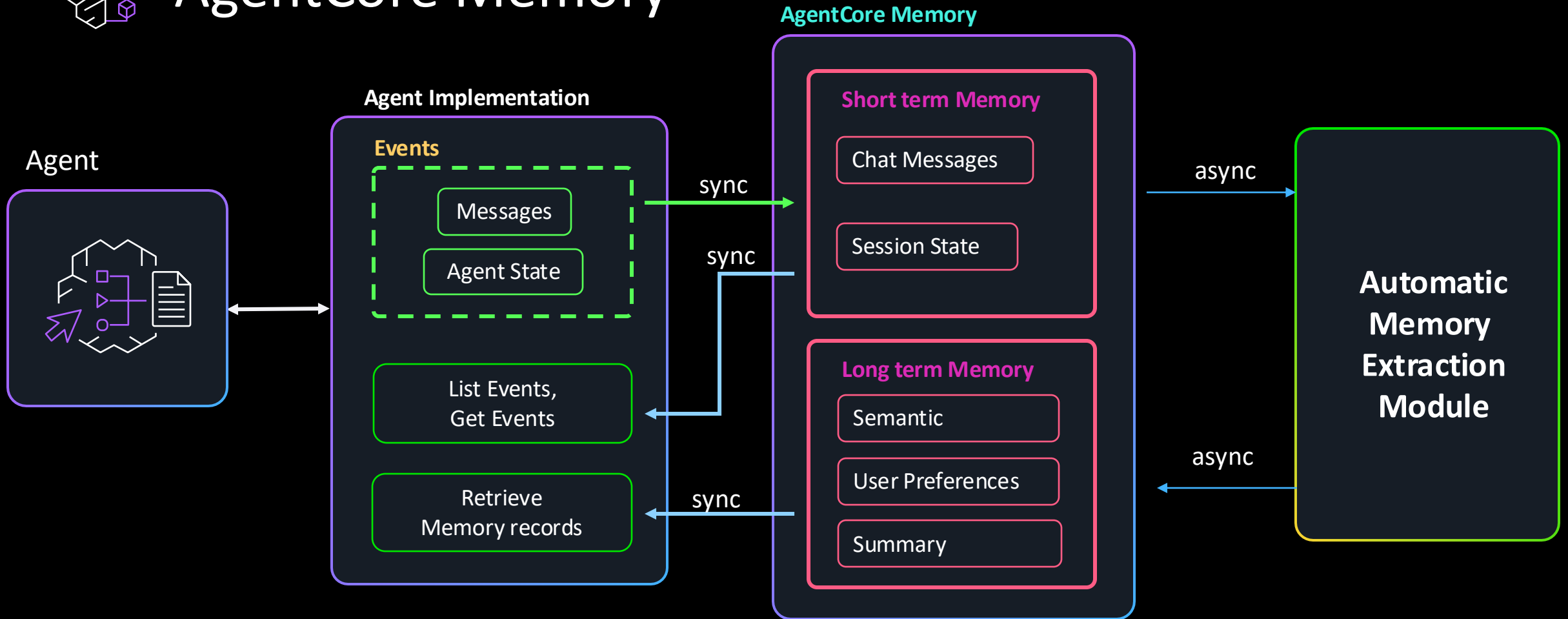
Example: retrieve food preferences for actor (user)

```
client = boto3.client("bedrock-agentcore")

client.retrieve_memory_records(
    memoryId="abcd1234",
    namespace="/anthonyk/food_preferences",
    searchCriteria={
        "searchQuery": "food preferences for anthony",
        "memoryStrategyId": "food-pref-strat",
    },
)
```



AgentCore Memory

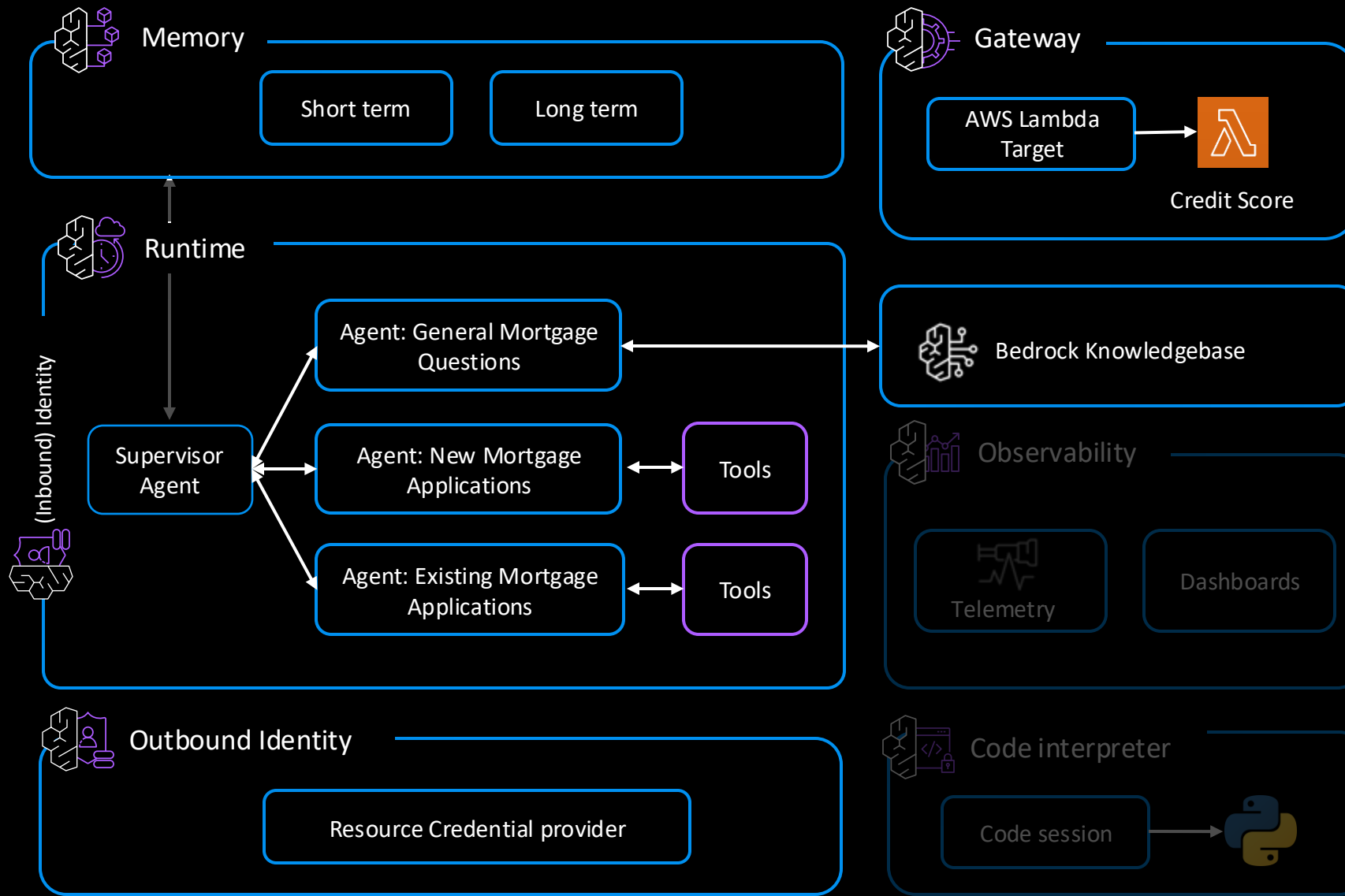




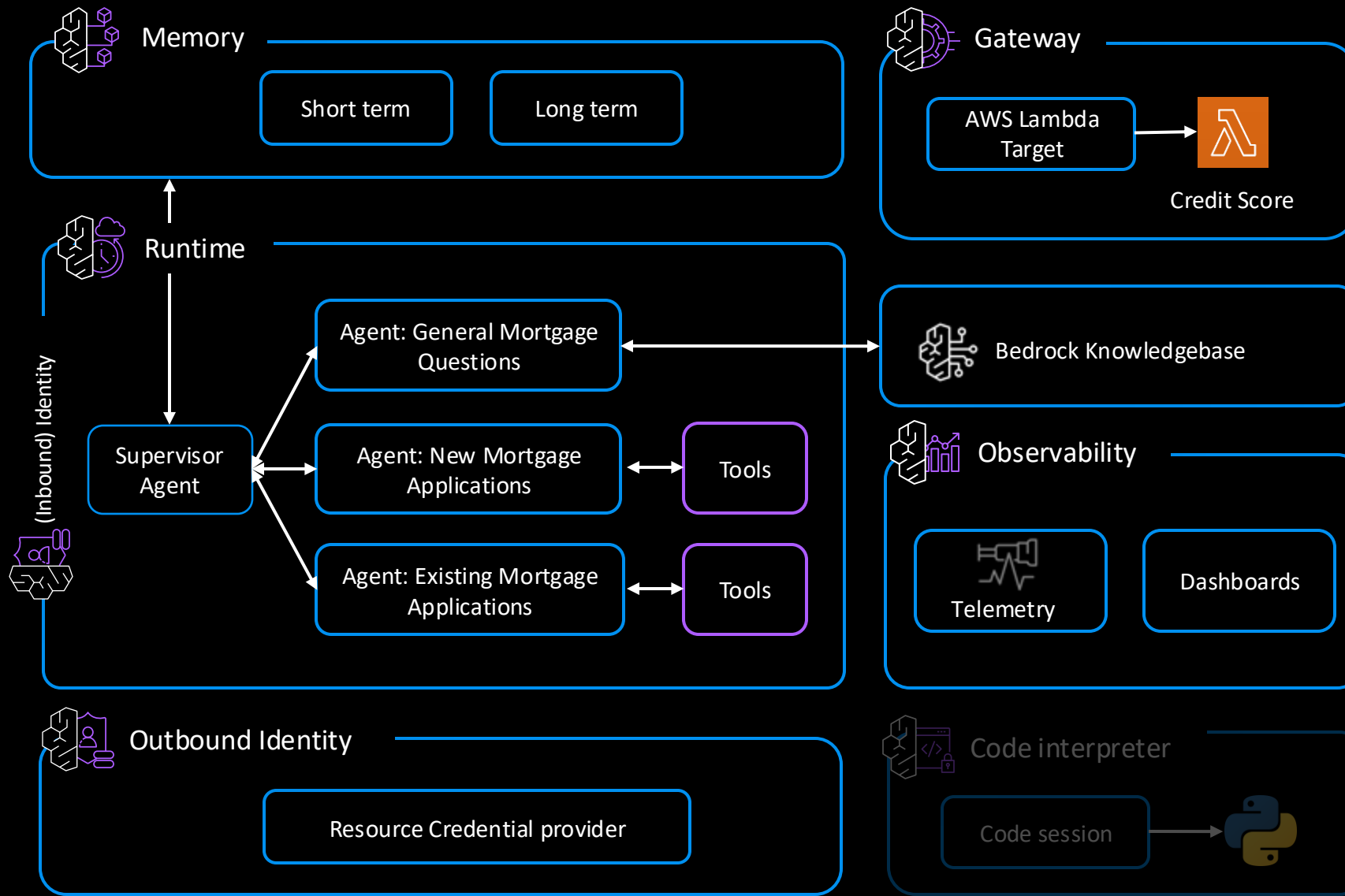
LAB #06-agentcore-memory



AgentCore Observability



- Step 1 – Set up, KB
- Step 2 – Strands Agent
- Step 3 – Runtime
- Step 4 – Gateway
- Step 5 - Identity
- Step 6 – Memory
- Step 7 – Observability
- Step 8 – Tools



- Step 1 – Set up, KB
- Step 2 – Strands Agent
- Step 3 – Runtime
- Step 4 – Gateway
- Step 5 - Identity
- Step 6 – Memory
- Step 7 – Observability
- Step 8 – Tools



AgentCore Observability

Maintain quality and trust



- Comprehensive end-to-end visibility into agent behavior
- Accelerated debugging and quality audits
- Quickly detect issues and assess performance trends

Accelerate time to market



- Dashboards powered by CloudWatch save developers time
- Single-pane-of-glass view into agents' operational health
- Eliminate the need to manually integrate data from multiple sources

Integrate with 3P observability tools



- Integration with a wide range of monitoring and observability tools, including CloudWatch
- Flexibility to leverage your existing observability stack

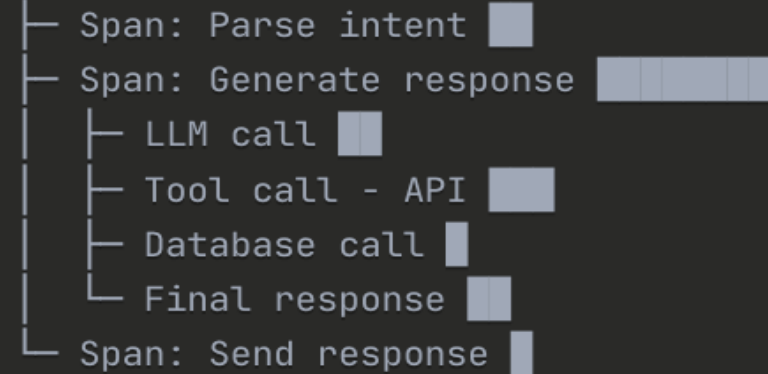


Key Concepts

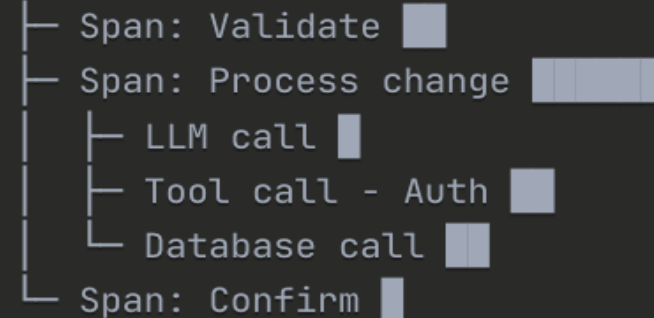
- Session
- Trace
- Span
- Sub-Span

Session: AI Agent Chat

Trace: "Help me track my order"



Trace: "Change address"



Session End

How it Works

Visibility to
operate agents
you can trust

The screenshot displays the AWS CloudWatch GenAI Observability console for the Bedrock AgentCore service. The interface includes a left-hand navigation menu with options like Dashboards, AI Operations, Alarms, Logs, Metrics, Application Signals (APM), Network Monitoring, and Insights. The main content area is titled 'GenAI Observability' and features a 'How it works' section with links to 'Enable & Configure' and 'View Analytics'. Below this, there's a 'Model Invocations' section for 'Bedrock AgentCore' with tabs for 'Agents view', 'Sessions view', and 'Traces view'. The 'Agents view' is active, showing an 'Overview' section with a table of metrics for agents/aliases, sessions, traces, error rate, and throttle rate. The 'Agents view' also includes a 'Runtime metrics' section and a table of agents with columns for Name, Environment, Sessions, Traces, Errors, Throttles, and P95 span latency (ms).

CloudWatch < GenAI Observability > Bedrock AgentCore

5m 30m 1h 3h 12h Custom (4w) Local timezone

How it works

Enable & Configure

- Learn about [Bedrock Model Invocation observability](#) and [Bedrock Agent Core observability](#).
- To protect and mask your sensitive data, turn on CloudWatch [Data protection at account level](#).

View Analytics

Access real-time dashboards for operational metrics (latency, errors, throttles).

Troubleshoot & Analyze

Use trace data and request flows to investigate issues, optimize performance, and identify bottlenecks.

Model Invocations **Bedrock AgentCore**

Agents view Sessions view Traces view

Overview

The following metrics provide insights derived from sampled spans for observability enabled agents

Agents/Aliases	Sessions	Traces	Error rate	Throttle Rate
55/31	275	864.8K	0%	0%

View details

Runtime metrics

These metrics provide insights into all agents deployed on Runtime

Agents (0/55)

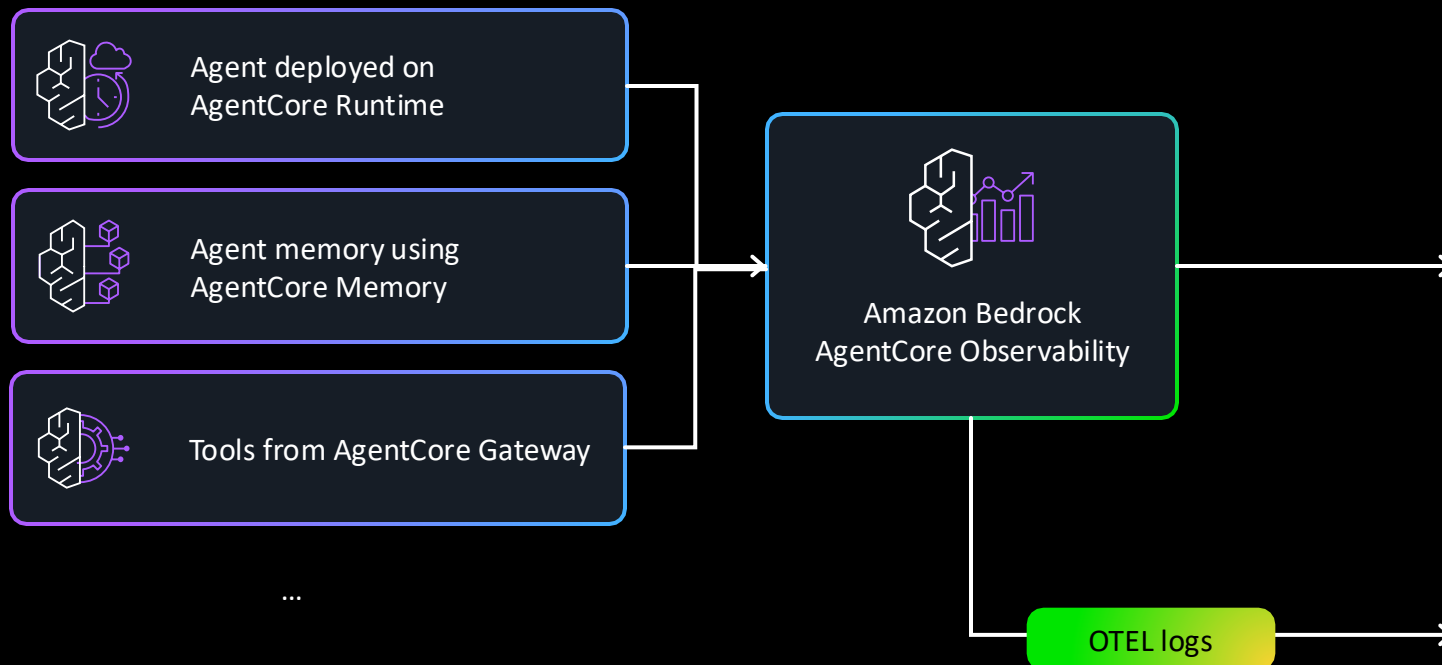
List of agents instrumented to send spans. Select an agent alias to view agent details.

Filter agents

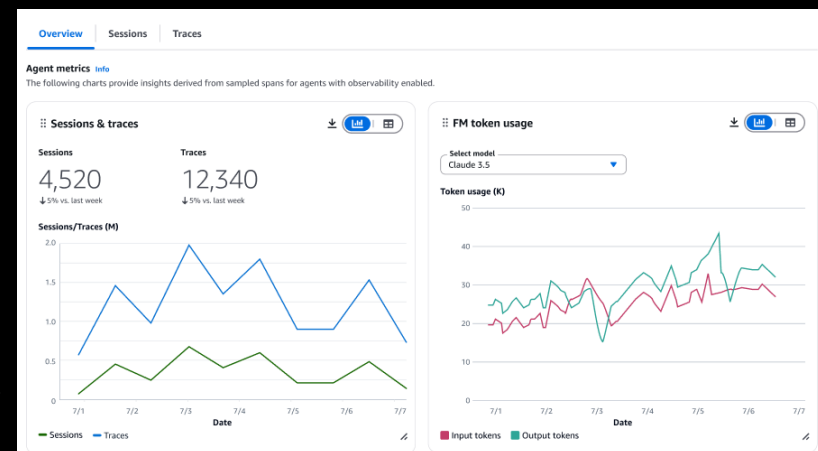
<input type="checkbox"/>	Name	Environment	Sessions	Traces	Errors	Throttles	P95 span latency (ms)
<input type="checkbox"/>	customersupportas	bedrock-agentcore	33	18.2K	46	0	0.85
<input type="checkbox"/>	customersupport	bedrock-agentcore	17	660	31	0	4125.07
<input type="checkbox"/>	customersupport	bedrock-agentcore	48	1.1K	14	0	4499.83
<input type="checkbox"/>	customersupport	bedrock-agentcore	4	837	7	0	1.73
<input type="checkbox"/>	customersupport...	bedrock-agentcore	73	36.7K	2	0	1.29
<input type="checkbox"/>	customersupport	bedrock-agentcore	10	90	2	0	4319.16



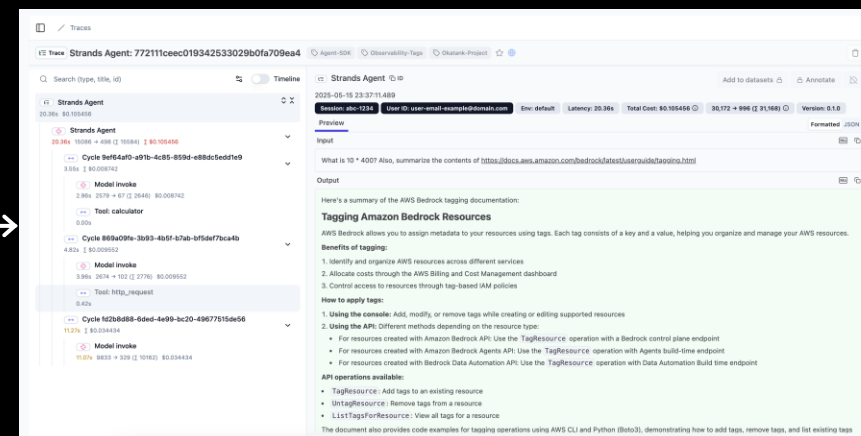
AgentCore Observability



AgentCore Observability dashboards



Third-party observability dashboards

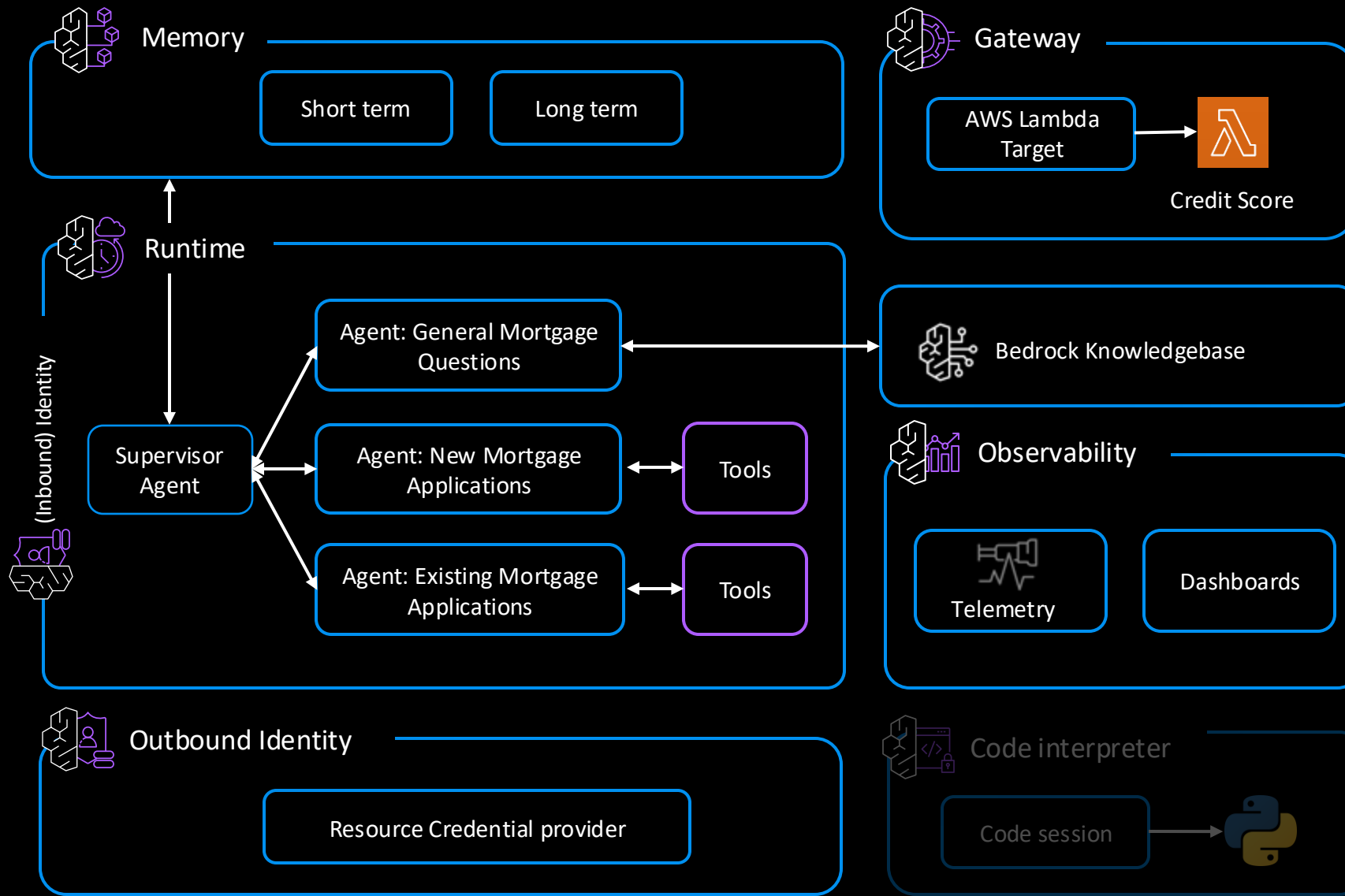




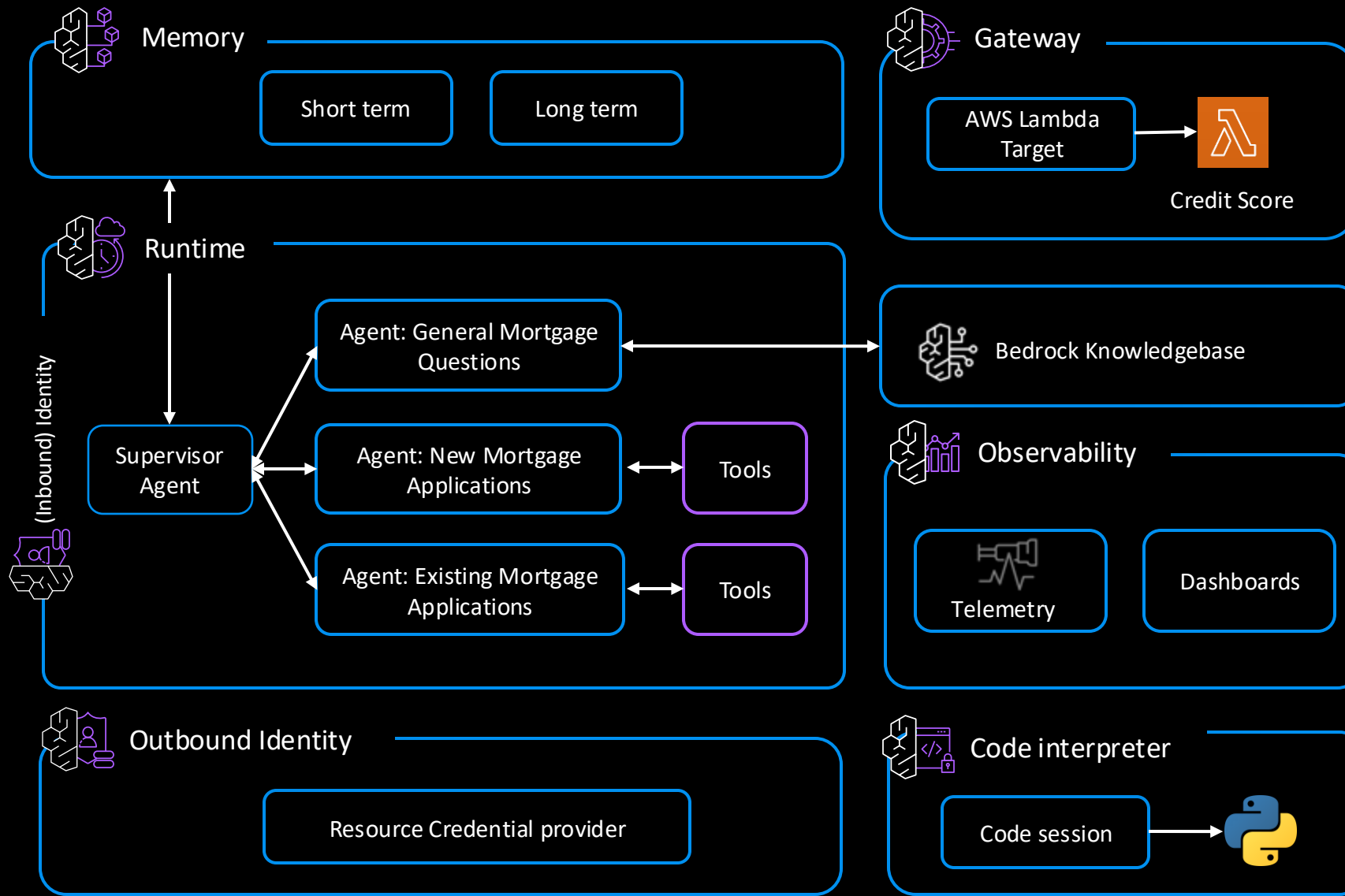
LAB #07-agentcore-observability



AgentCore Built-in Tools



- Step 1 – Set up, KB
- Step 2 – Strands Agent
- Step 3 – Runtime
- Step 4 – Gateway
- Step 5 - Identity
- Step 6 – Memory
- Step 7 – Observability
- Step 8 – Tools



- Step 1 – Set up, KB
- Step 2 – Strands Agent
- Step 3 – Runtime
- Step 4 – Gateway
- Step 5 - Identity
- Step 6 – Memory
- Step 7 – Observability
- Step 8 – Tools



AgentCore Code Interpreter

Execute code securely



- Execute complex workflows and data analysis in isolated sandbox environments
- Access internal data sources securely without exposing sensitive data

Large-scale data processing



- Process gigabyte-scale datasets efficiently using Amazon S3 integration, without API limitations

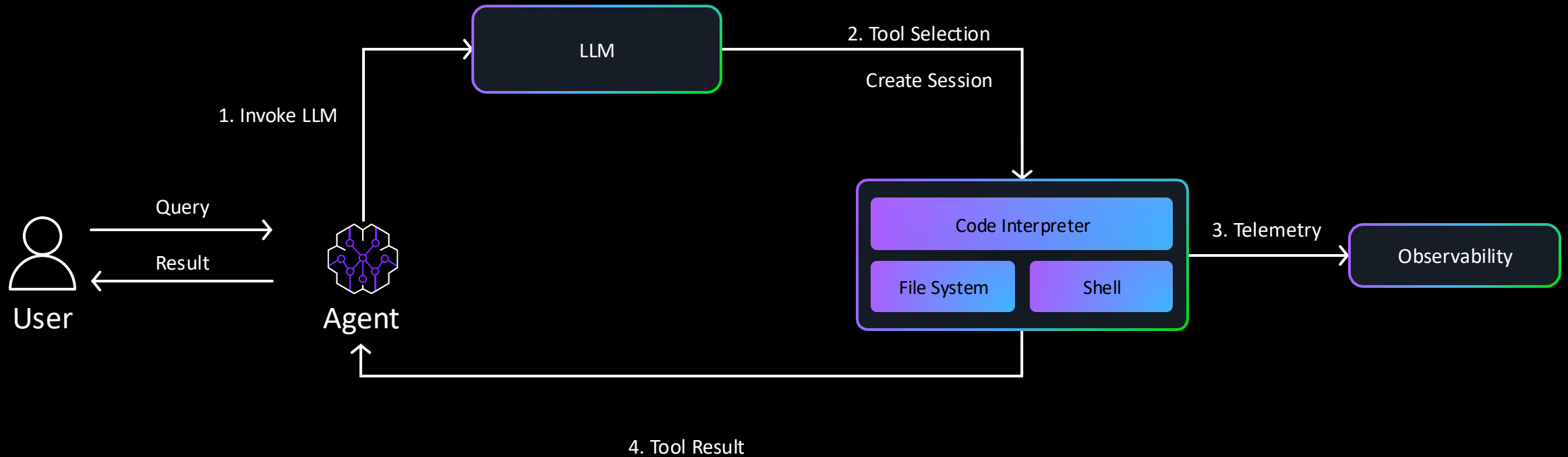
Ease of use



- Quick start with pre-built execution runtimes for JavaScript, TypeScript, and Python with common libraries pre-installed



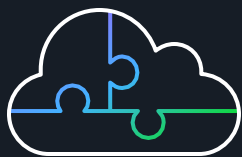
AgentCore Code Interpreter





AgentCore Browser

Serverless browser infrastructure



- Low latency browser sessions
- Auto-scales from 0 to hundreds of concurrent sessions

Enterprise-grade security



- Session isolated compute with VM-level isolation per user
- Secure credential handling

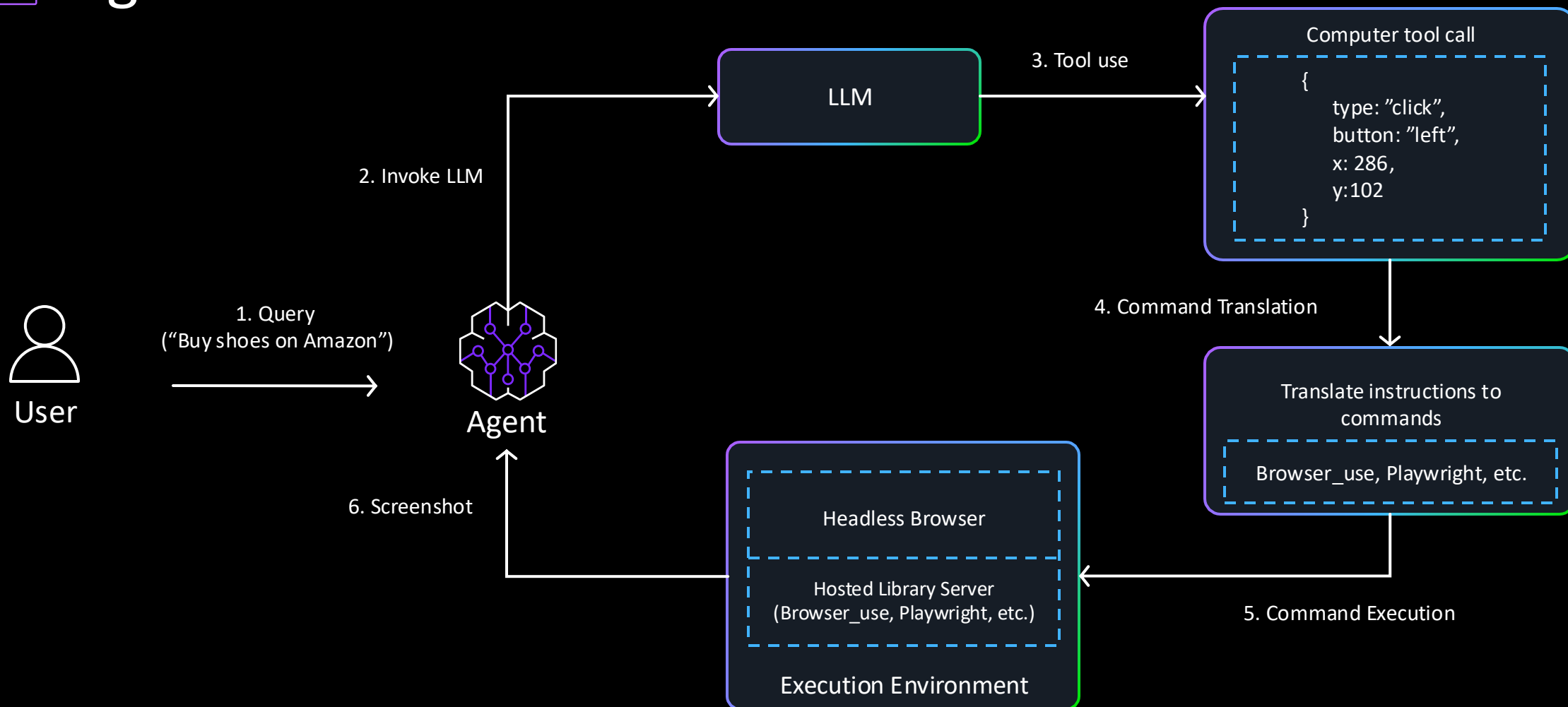
Enterprise observability



- Live streaming URLs for real-time monitoring
- Session replays for debugging
- Extensive logging of all browser commands to CloudTrail



AgentCore Browser





LAB #08-agentcore-tools



Survey



<https://pulse.amazon/survey/BEJSYO2M>





Create your AWS
Builder ID and claim
your community alias
and get access to...



Amazon Q

Access to the most capable generative
AI-powered assistant

600+

AWS Skill Builder Courses

Unlock a world of learning
to build in-demand skills



AWS re:Post

A curated knowledge center
and a vibrant community



Join AWS User Groups in New Zealand

Attend In person regular meetups
organized by the AWS User Groups
enabling peer to peer learning

Sign up today at: bit.ly/anzbuilderid



Join AWS User Groups

User groups are peer-to-peer communities which meet regularly to share ideas, answer questions, and learn about new services and best practices.

17 AWS User Groups in Australia and New Zealand

450+ AWS User Groups worldwide

Monthly/ Quarterly meetups

Find a User Group near you

builder.aws.com/connect/community/user-groups



Scan to join an
AWS User Group
In New Zealand

