Infrastructure for Agentic AI: Architecting Scalable Secure Multi-Model Systems

Brian Shlisky,
Distinguished Solutions Engineer, GES

John Cuneo,
Worldwide AI Solutions Engineer

BRKCOM-1126

Cisco Webex App

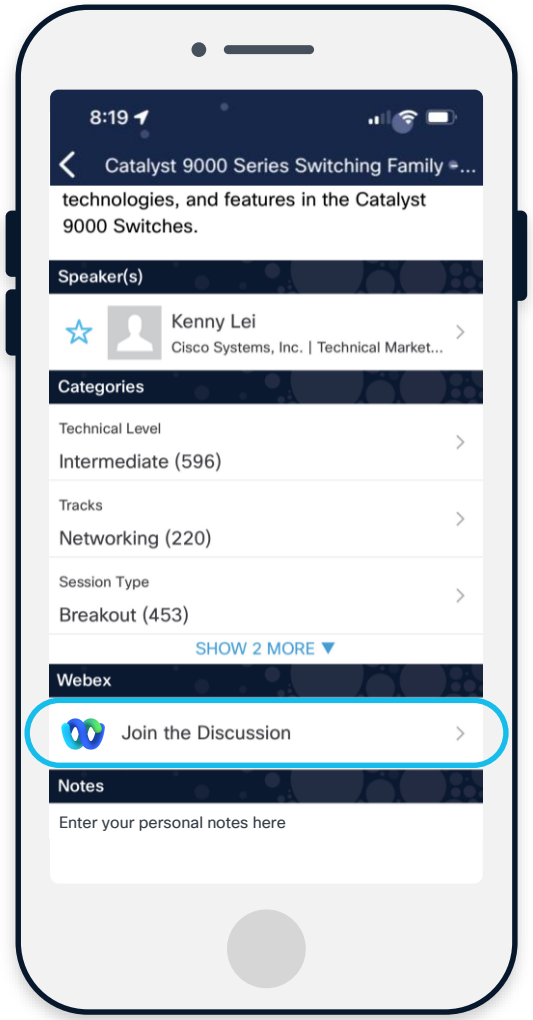
Questions?

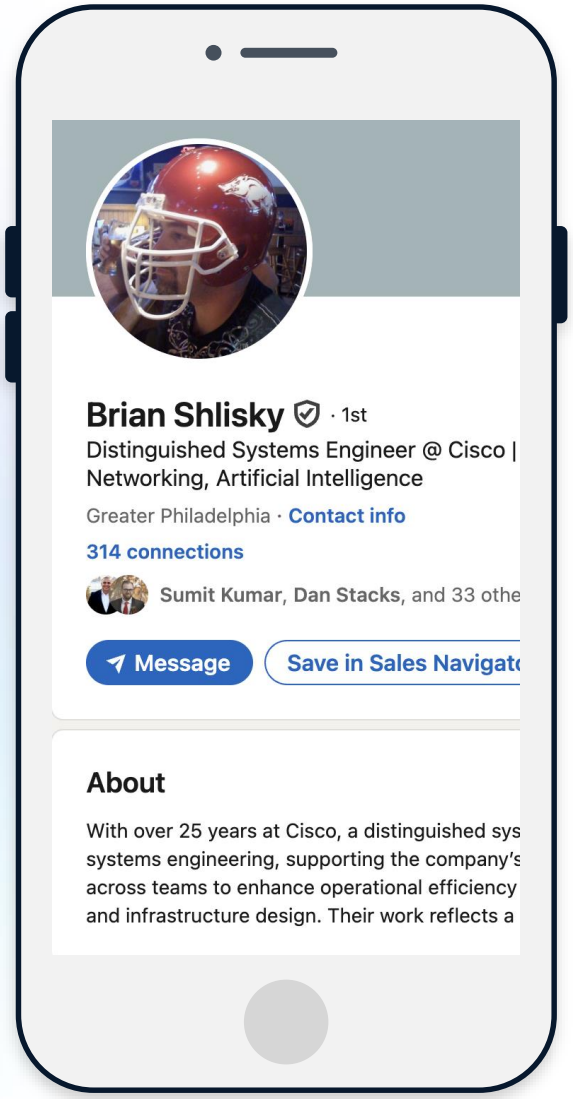
Use Cisco Webex App to chat
with the speaker after the session

How

1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until June 13, 2025.



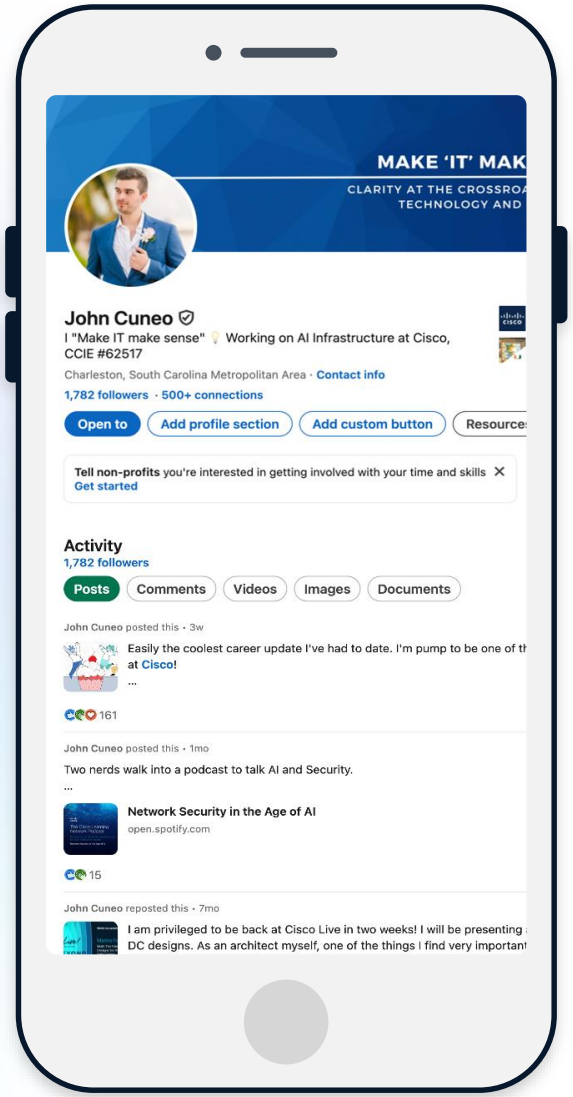


Who's Brian
What I do...

- Distinguished Solutions Engineer
- Focused on Workload Solutions
- Focus on all thing AI – Not just Gen-AI
- Tattoo enthusiast

Connect on LinkedIn: www.linkedin.com/in/brian-shlisky

Contact me at: bshlisky@cisco.com



Who's John
What I do...

- Global technical AI go-to-market
- Strategic customer engagement
- Executive briefing
- Eat, breathe, and dream GenAI for 2 years
- Lifelong fantasy nerd & gamer

Connect on LinkedIn: https://www.linkedin.com/in/cuneojohn/

Contact me at: johcuneo@cisco.com

Agenda

01 What is Agentic AI

02 Agentic System Architecture

03 Mapping Agentic AI to Infrastructure

04 Getting Started

Learning objectives

By the end of this session, you'll be able to:

Understand the difference between basic and agentic AI

Describe Agentic AI architecture

Architect the building blocks for Agentic AI

1. What is Agentic AI

Generative to Agentic AI Capabilities

Basic text generation

- Basic tasks with “emergent” results
- Writing and coding
- Text summarization
- Translation

Augmented generation

- Integrated store of information to enhanced prompt
- Then execute basic text generation

Tool calling

- Ability to execute custom functions / API calls
- Query for structured source of truth
- Interact with existing applications

Advanced reasoning

- Simulate human thought process via recursive prompting “chain of thought”
- Deconstruct request
- Plan the approach
- Execute based on the plan
- Validate if response makes sense

What Is Agentic AI?

AI systems that **act autonomously** and **adapt in real time** to changing environments.

They solve multi-step, complex problems **without constant human guidance**.

Agentic AI relies on autonomous agents leveraging large language models and machine learning.

AI generated image

How Agentic AI Differs from Traditional AI

Traditional AI

Rule-based systems that automate simple tasks

Requires human intervention for adjustments

Limited adaptability to new scenarios

Agentic AI

Independently executes tasks toward specific outcomes

Self-adapts to changing conditions

Operates through networks of collaborating agents

LLMs: Word-by-Word Prediction

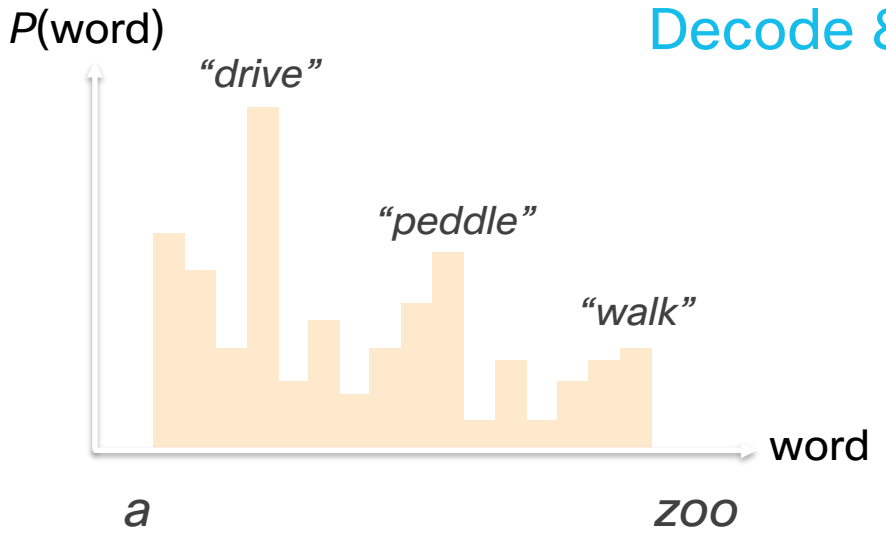
Language models perform word-by-word prediction

Transformers / Attention Mechanism →
Use prompt and previously-generated text to predict the next word

"I got into the car and started to ..." → LLM → "drive"

Input & Prefill

Decode & Output



Word-by-word predictions based on patterns they have **learned** from training on enormous corpora of text

Agentic Flows Origin Story: Prompting Techniques

Early-to-mid 2023: Best practices for prompting and code generation

A **PROMPT** is a set of instructions provided to an LLM for it to execute.

PROMPT ENGINEERING is the art and science of designing prompts to give optimal results.

Prompt Engineering

A well-engineered prompt can contain one or more of the following elements:

- Persona
- Task / objective / instruction
- Tone / style
- Target audience
- Context
- Output format

Prompts: powerful "knobs" to get the most out of LLMs
Engineering prompts → Effective results

Generative AI vs. Agentic AI Token Generation

Basic Generative AI

Input prompt: 10 tokens

Direct processing path

Output completion: 50 tokens

Total: 60 tokens

Agentic AI with Reasoning

Input prompt: 10 tokens

Internal reasoning: 5,000 tokens

Multiple solution pathways explored

Output completion: 100 tokens

Total: 5,110 tokens

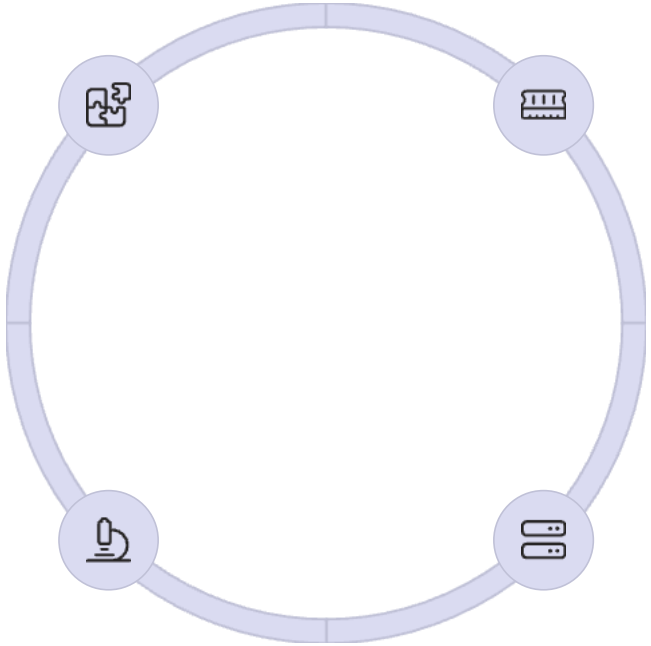
Implications of Large-Scale Token Generation

Enhanced Problem-Solving

More tokens enable exploration of multiple solution paths simultaneously, yielding better results.

Scientific Applications

Complex domains benefit from exhaustive reasoning capabilities enabled by large token counts.



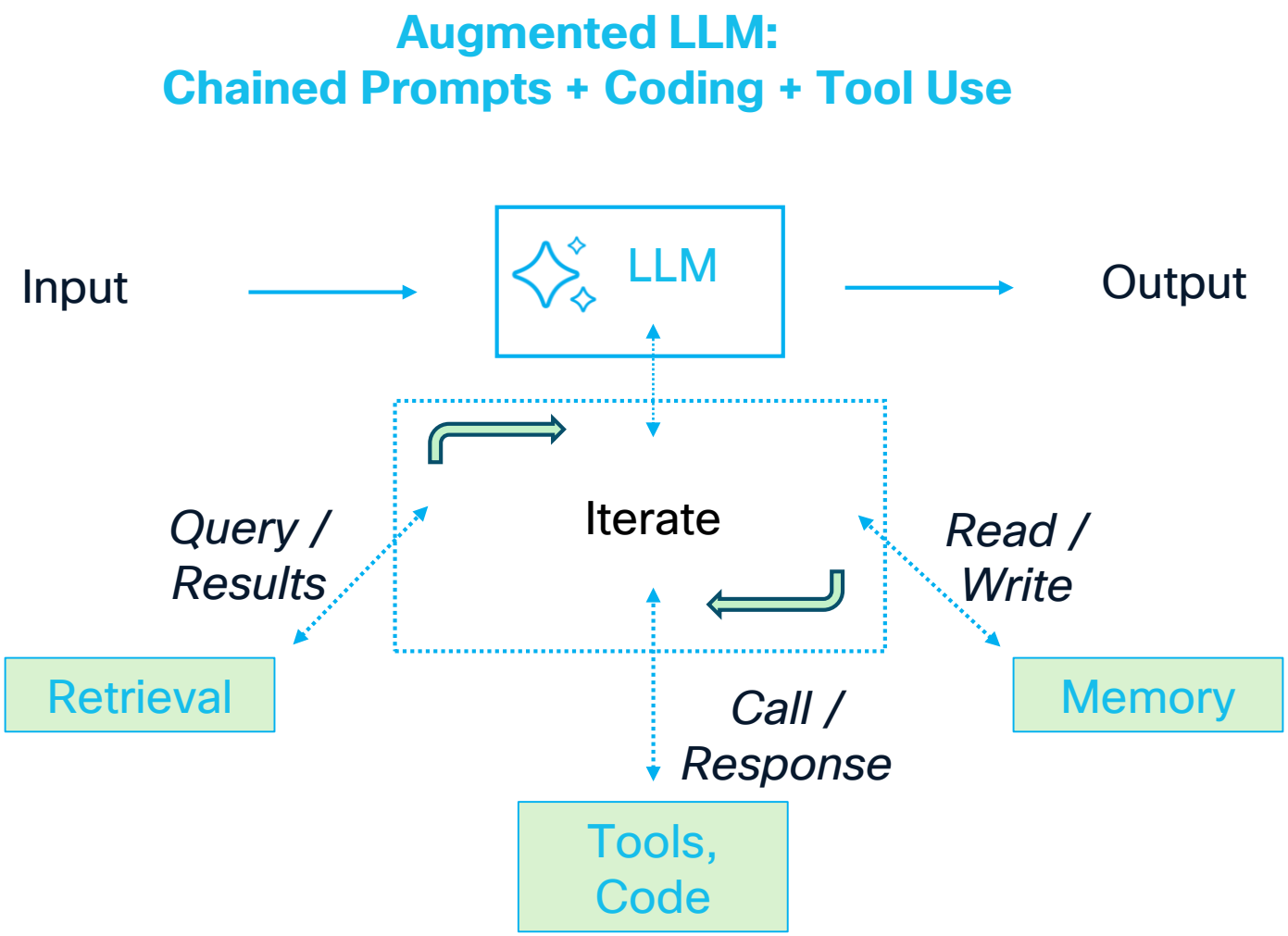
Context Retention

Expanded token budgets allow systems to maintain awareness of entire problem spaces.

Resource Challenges

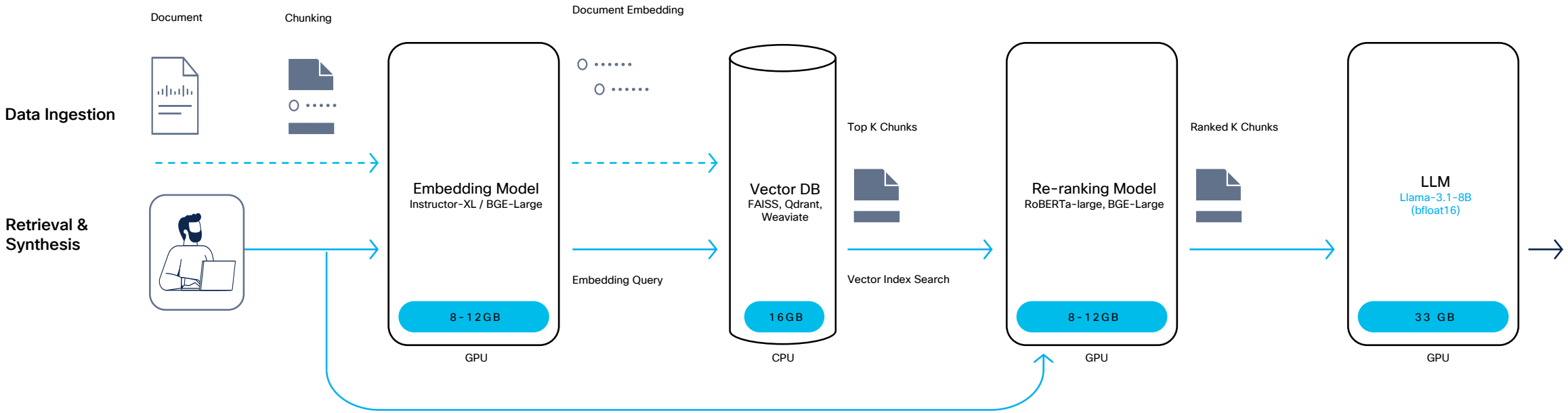
Generating thousands of tokens requires significant computational resources and costs.

Agentic Building Block: The Augmented LLM



Building Block Example – Retrieval Augmented Generation

Retrieval



Building Block Example – Tool Usage

Tools, Code

Natural Language ⟷ APIs, Function Calls

API Call | Function Call | Web Search

Prompt:
Get the weather forecast for San Francisco using OpenWeatherMap.

```python
import requests

api_key = "YOUR_API_KEY"
city = "San Francisco"
url = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={api

response = requests.get(url)
data = response.json()

print(data["weather"][0]["description"])
```

Prompt:
Schedule a meeting with John for Tuesday at 3pm.

```
{
  "function_call": {
    "name": "scheduleMeeting",
    "arguments": {
      "time": "2023-05-09T15:00:00",
      "invitees": ["John"]
    }
  }
}
```

Prompt:
Search the web for recent news about CSCO earnings report.

```python
from some_search_tool import search_web

results = search_web("Cisco CSCO Q2 2023 earnings report")
print(results[0]['title'], results[0]['snippet'])
```

Agents vs. Agentic Flow

Agents

A system or software entity that can independently perform tasks, typically guided by goals or objectives, interacting dynamically with an environment.

Key features:

- **Autonomy:** Can take actions independently.
- **Goal orientation:** Operates based on specific objectives or outcomes.
- **Environment interaction:** Interacts with external systems or resources to accomplish tasks.

Agentic Flow

A structured or semi-structured workflow or series of tasks leveraging multiple agents (or agent-like behaviors), LLMs, and external tools or APIs.

Key features:

- **Coordination:** Multiple tasks, steps, or processes orchestrated together.
- **Structured sequences:** Tasks are broken down into discrete, logical steps.
- **Use of specialized sub-components:** Different models, tools, or APIs may each handle specific tasks.

Agents vs. Agentic Flow

Feature	(Single) Agent	Agentic Flow
Autonomy	High Makes decisions independently	Medium Structured but with some autonomy per step
Coordination	Low/None Standalone tasks	High Multiple tasks coordinated
Complexity	(Typically) singular goal-oriented tasks	Multiple steps, tools, and integrations
Scalability	Limited	Highly scalable, flexible (modular approach)

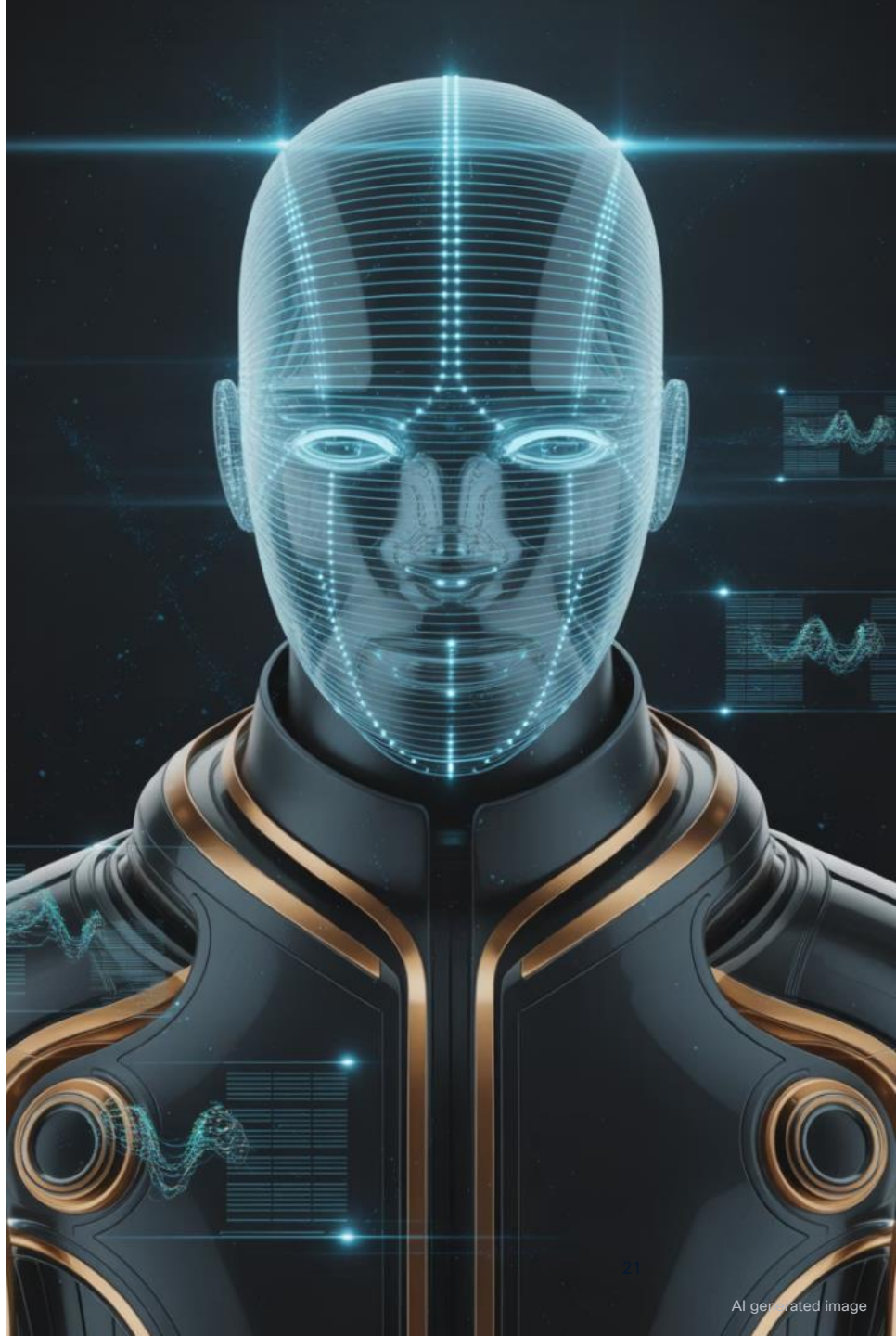
2. Agentic System Architecture

Tools in Agentic AI Applications

Agentic AI systems observe, decide, and act autonomously. Their capabilities extend beyond language generation through specialized tools.

The market is booming. Projections show Agentic AI reaching $38.2B by 2030.

Leading examples include AutoGPT, BabyAGI, Microsoft Copilot, and Anthropic Claude.



AI generated image

Agentic AI System Capabilities

Perception

Data input processing from various sources

Collaboration

Multi-agent coordination and communication

Reasoning

LLM-powered analysis and decision making

Action

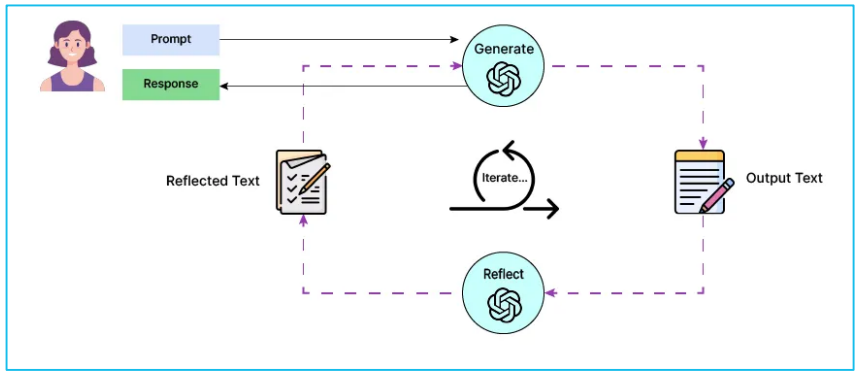
Task execution across connected systems

Learning

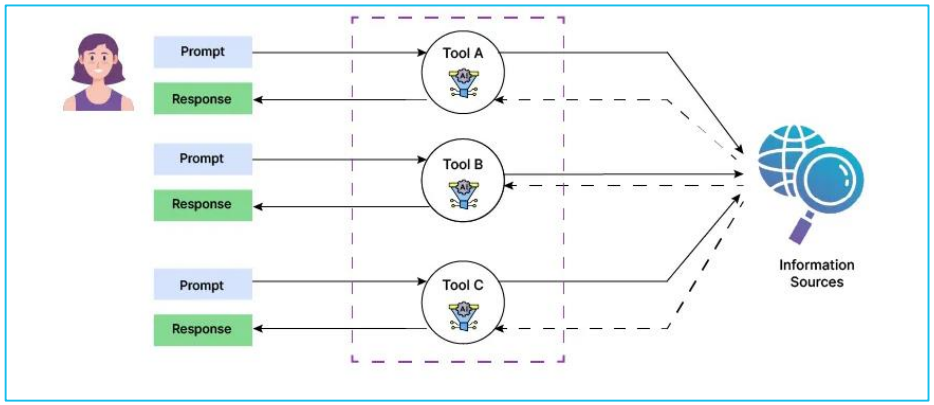
Continuous improvement from feedback

Agentic Reasoning: Design Patterns

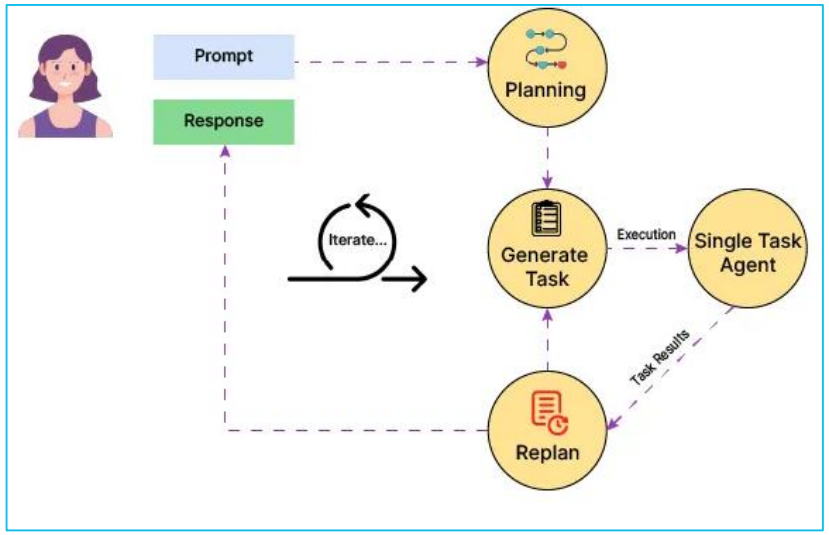
Reflection



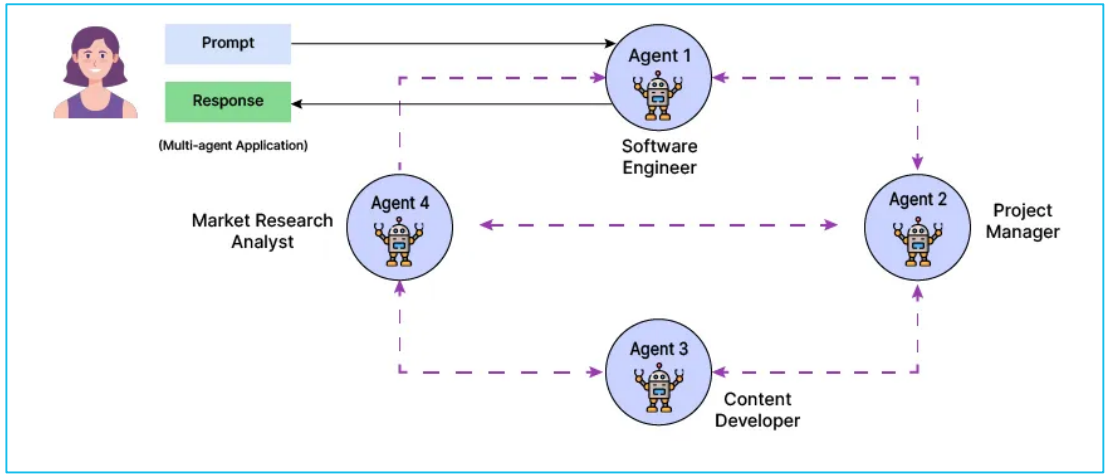
Tool Use



Planning



Multi-Agent Collaboration



Source(s):
Deeplearning.ai: https://www.deeplearning.ai/the-batch/llms-evolve-with-agentic-workflows-enabling-autonomous-reasoning-and-collaboration/
Analytics Vidhya: https://www.analyticsvidhya.com/blog/2024/10/agentic-design-patterns/

External API Integrations

Weather APIs

OpenWeatherMap: 1B+ daily requests

E-commerce

Shopify, Amazon, Stripe APIs

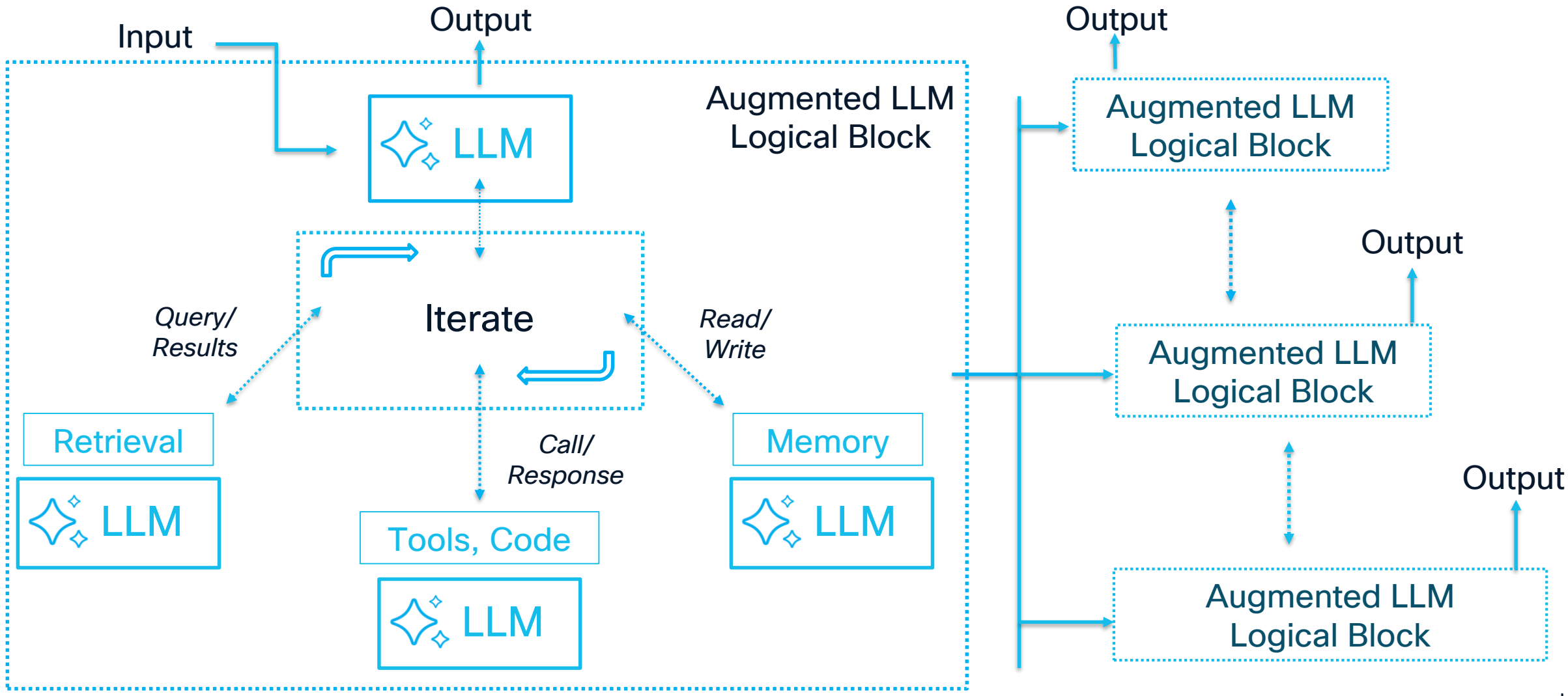
Productivity

Google Workspace, Microsoft 365 Graph API

Communication

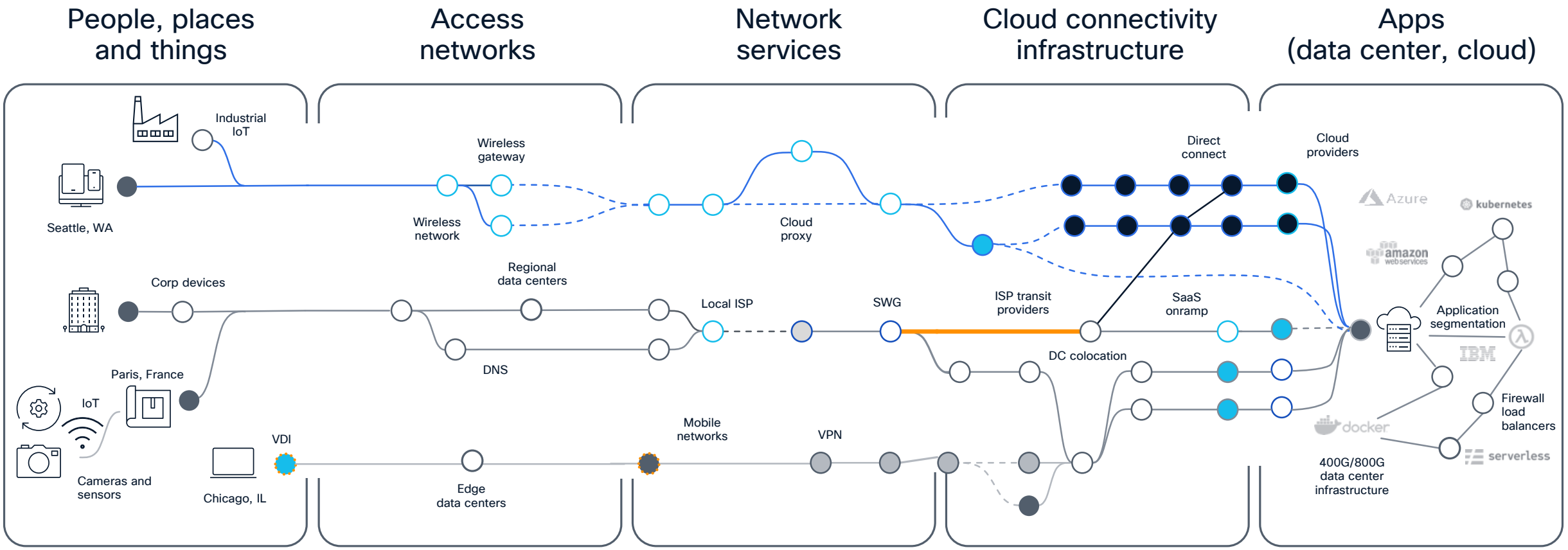
Twilio, SendGrid: 100B+ emails yearly

Agentic Workflow Building Blocks

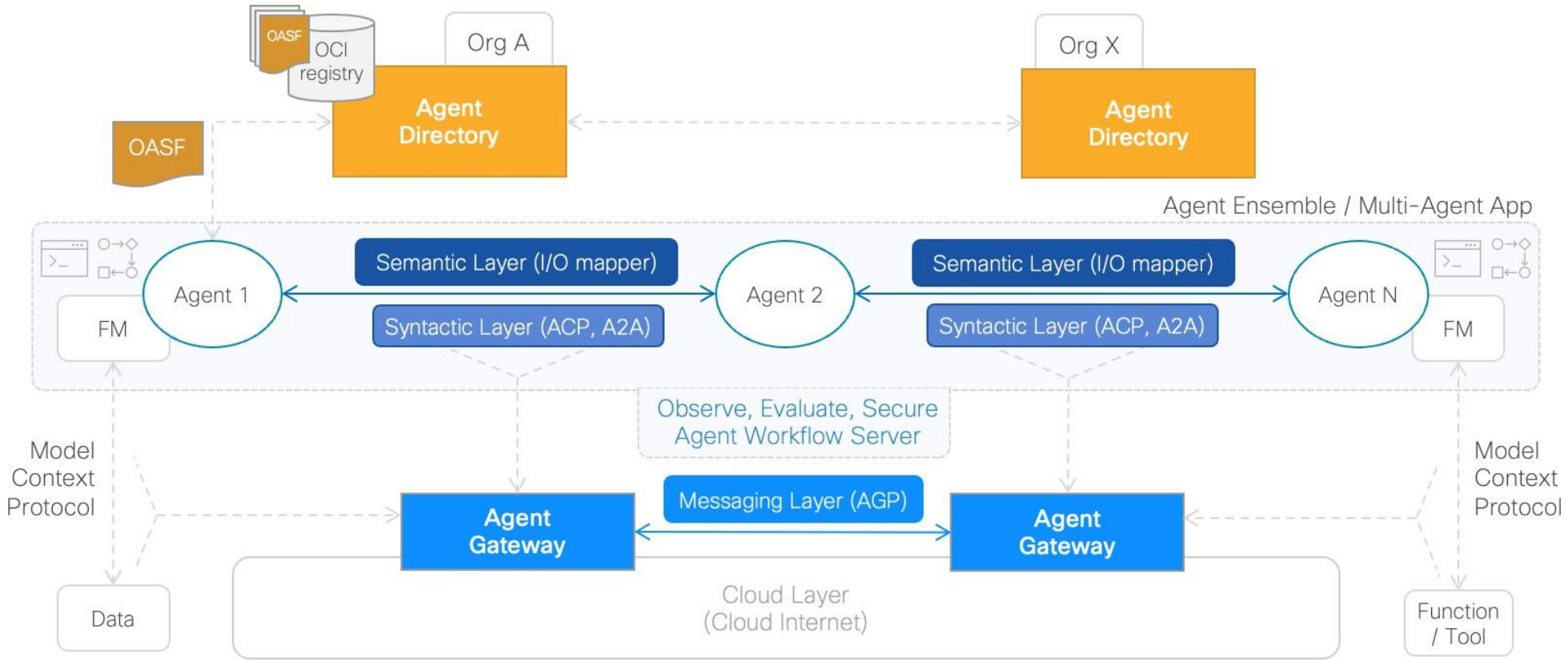


Agent Communication & Security

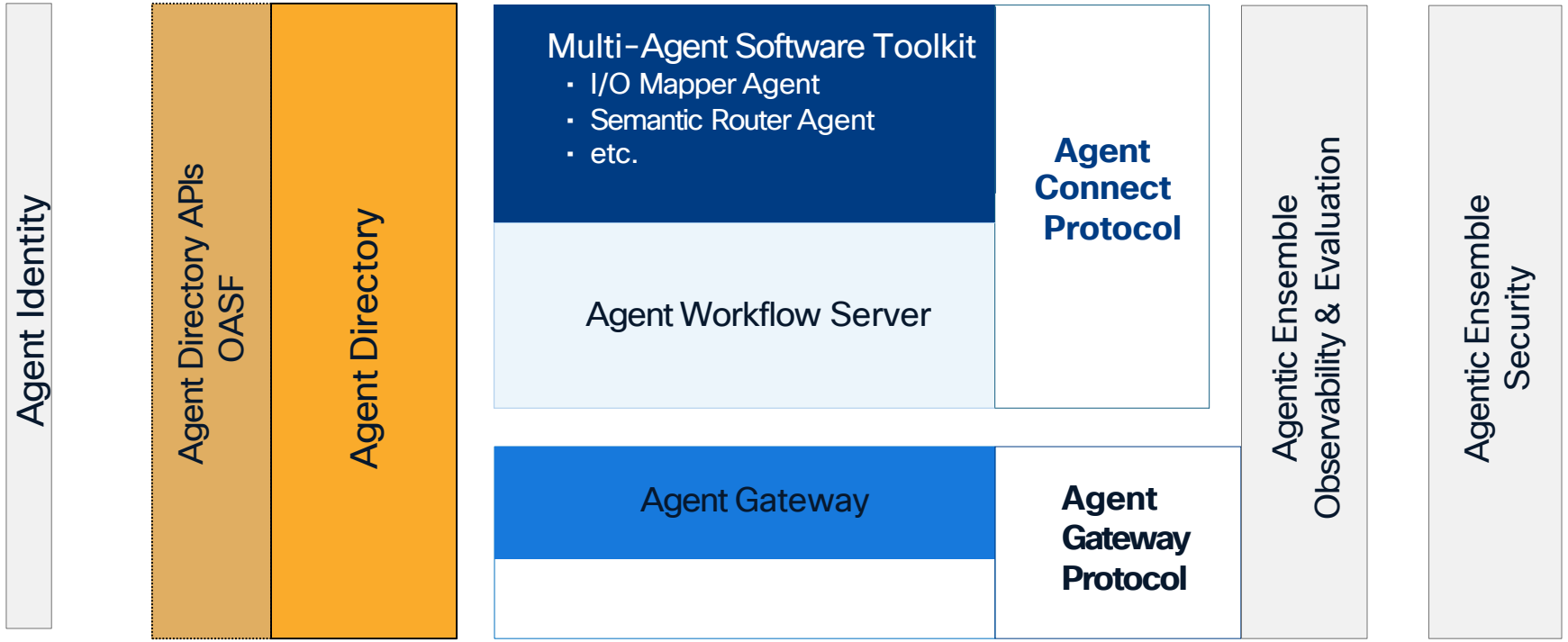
Moving the data



Internet of Agents Architecture



Simplified Internet of Agents stack



Simplified AGNTCY stack: description

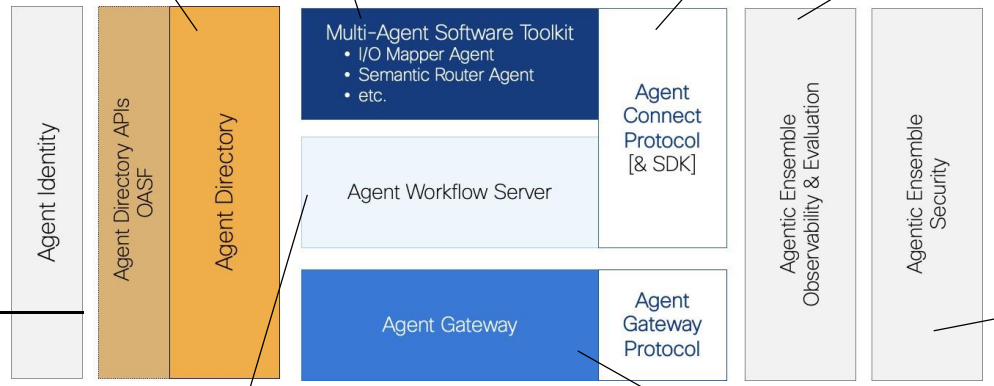
This toolkit contains a set of agents that support seamless creation of Multi-Agent workflows. Few examples:

- I/O Mapper: Handles semantic data adaptations.
- Semantic Router: Directs workflows via semantic matches.
- Human in the Loop: interfaces with human input/output seamlessly.

Allows to announce and discover agents or multi-agent applications. Any organization can run its directory and keep it in sync with others, forming the Internet of Agents inventory

Standard interface (+ client/server SDK) to invoke agents (or agentic applications), provide input, retrieve output, retrieve supported schemas, graph topology and other useful information

Telemetry collectors, tools and services to enable multi-agent application observability and evaluation



Open Agent Schema Framework is an OCI based extensible data model allowing to describe agents' attributes and ensuring unique identification of agents

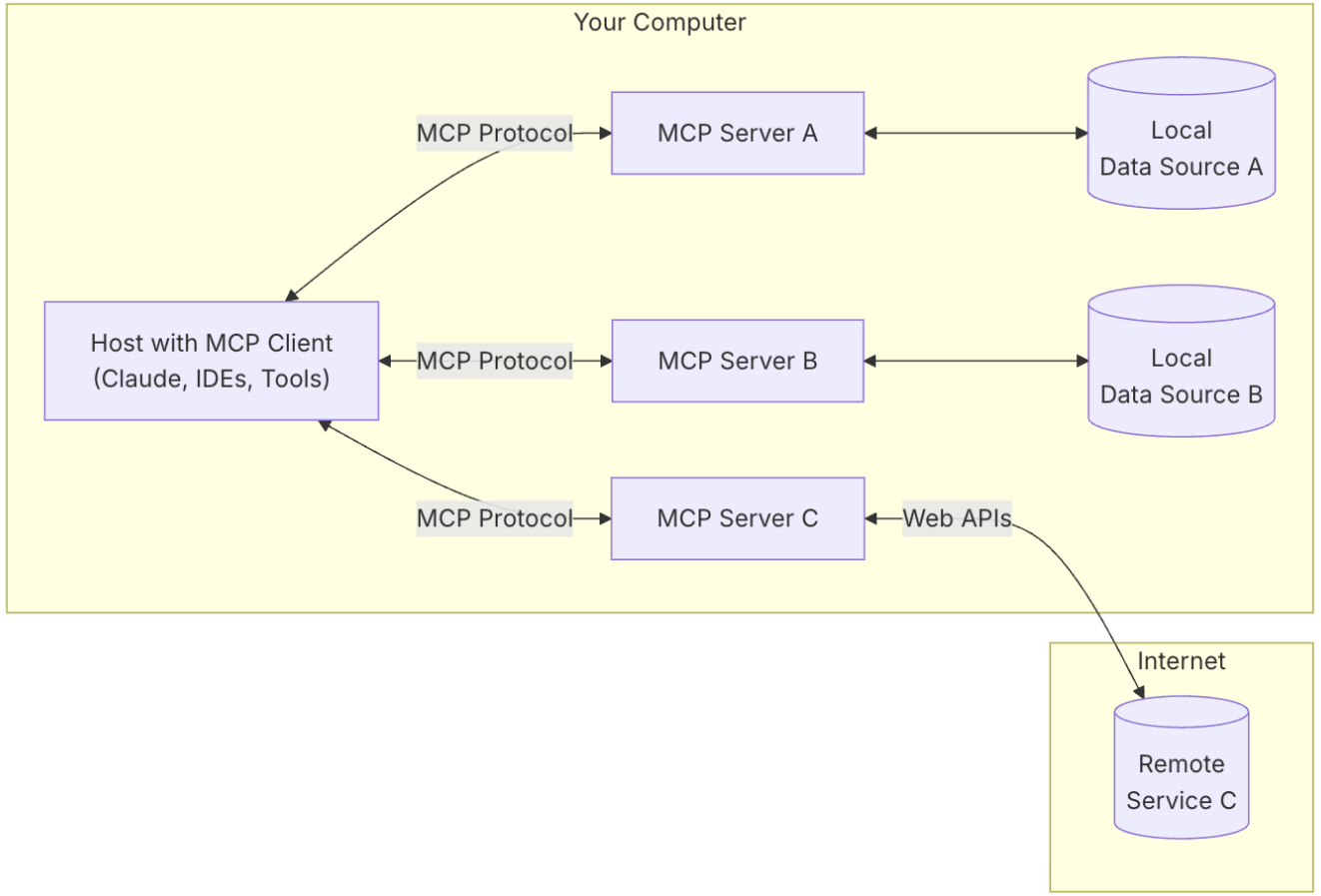
Tools and services to trust and protect multi- agent applications

This component deploys and supervises agent workflows written in various frameworks and makes them available through the Agent Connect Protocol. Such workflows could be multi-agent applications including a mix of toolkit agents, local and remote agents.

The Agent Gateway offers handy secure (MLS and quantum safe) network-level communication services to a group of agents (typically those of a given multi-agent application) through SDK/Libraries. It extends gRPC to support pub/sub interactions in addition to request/reply, streaming, fire & forget and more.

Model Context Protocol

- Standard interface for LLM to data & tool connections
- Compatible with AGNTCY stack



Source(s): https://modelcontextprotocol.io/specification/2025-03-26/architecture

Cisco AI Defense Guardrail Categories

Security

- Prompt Injection
- Denial of service
- Cybersecurity and hacking
- Code presence
- Adversarial content
- Malicious URL

Privacy

- IP Theft
- PII
- PCI
- PHI
- Source code

Safety

- Financial harm
- User harm
- Societal harm
- Reputational harm
- Toxic content

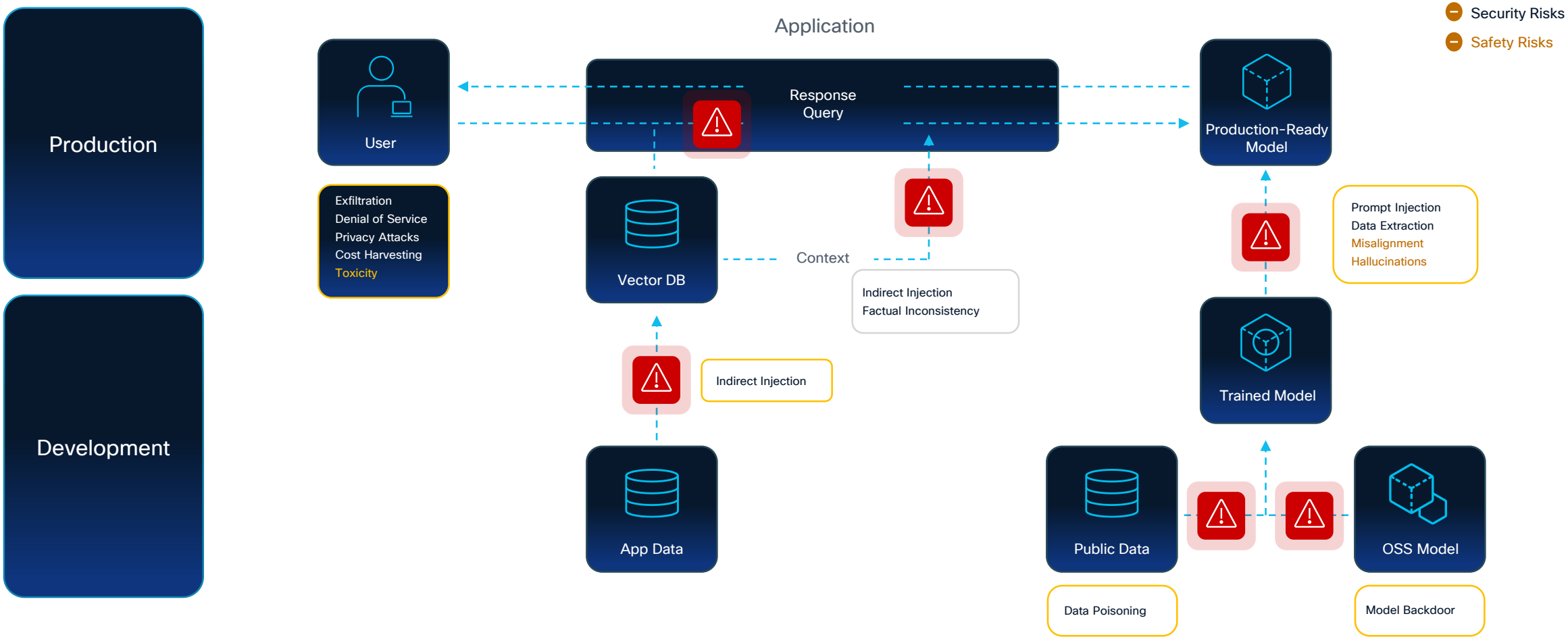
Relevancy

- Content moderation
- Hallucination
- Off-topic content

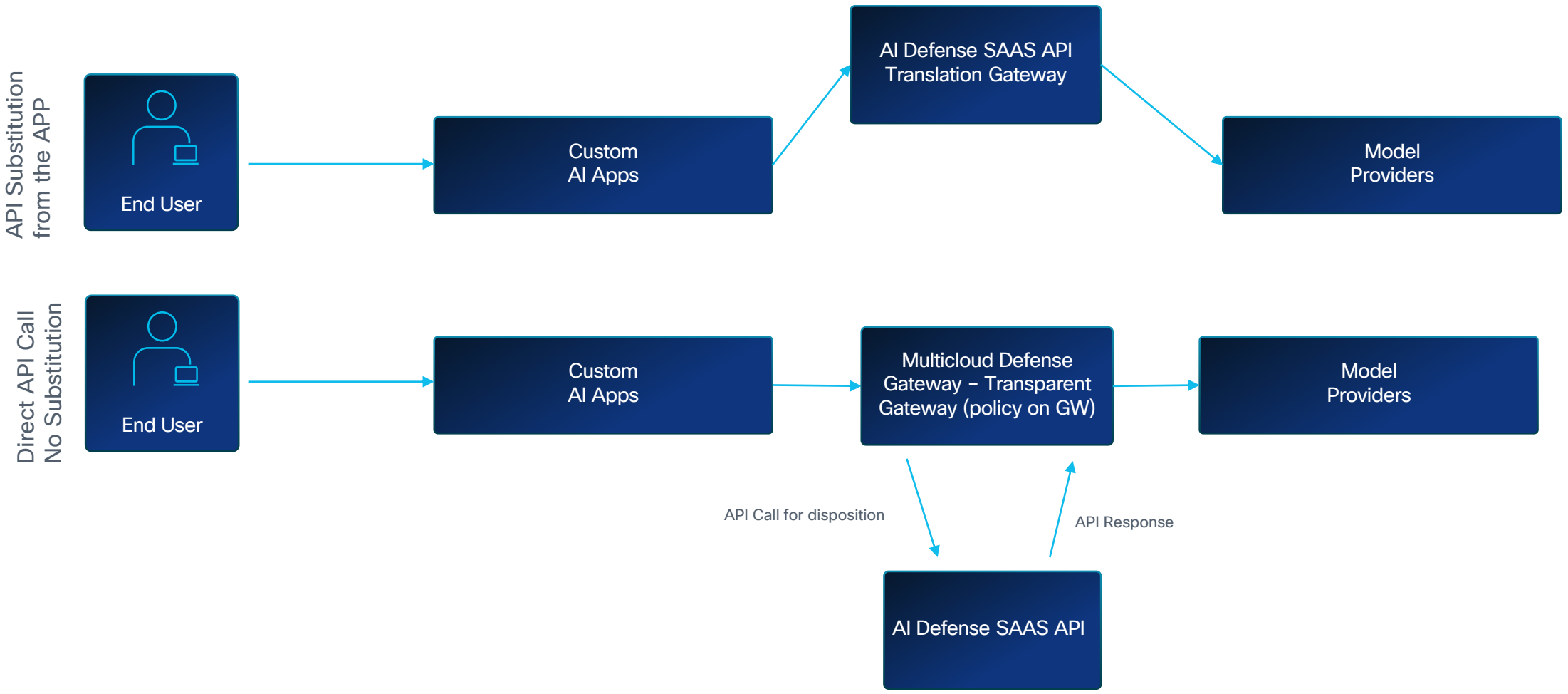
Map guardrails to standards and frameworks like: OWASP MITRE ATLAS

Guardrails can be modified to fit industry, use case, or preferences

How are enterprises using AI applications?

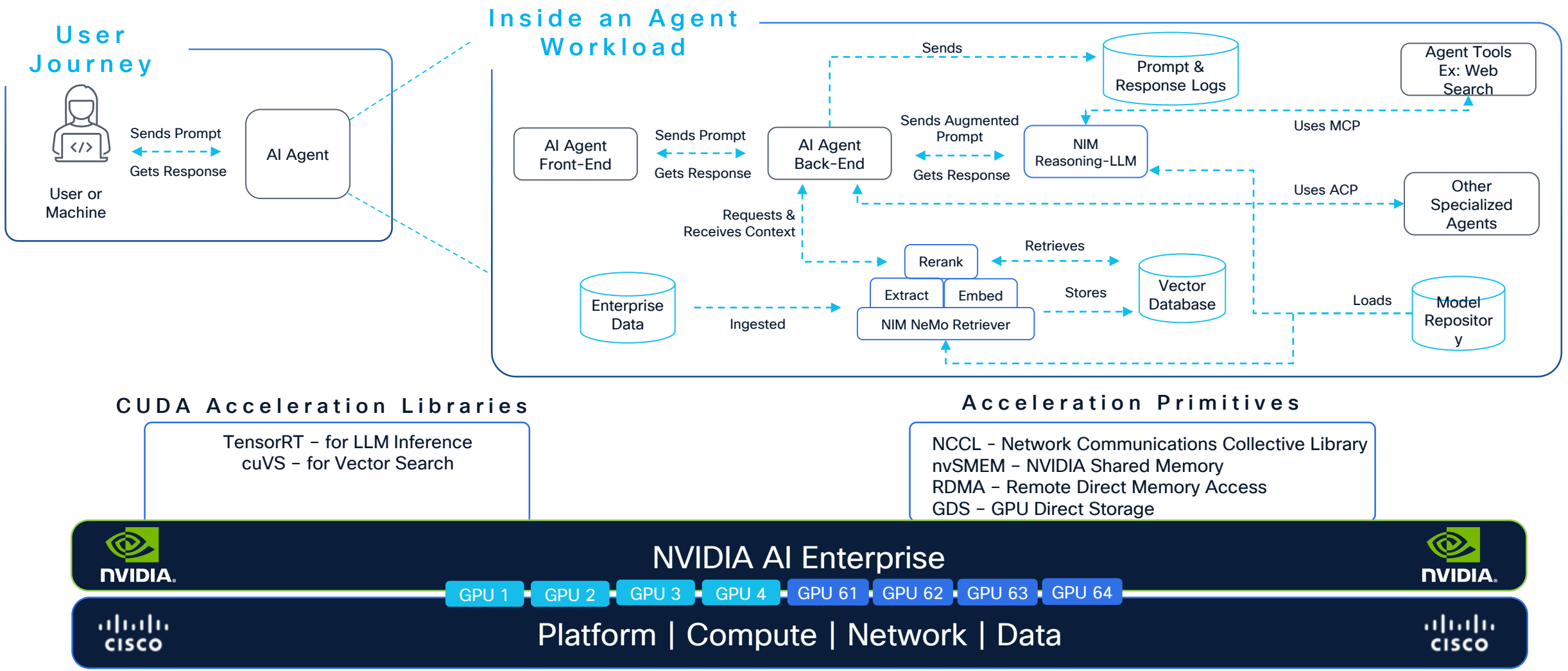


Guardrail Implementation



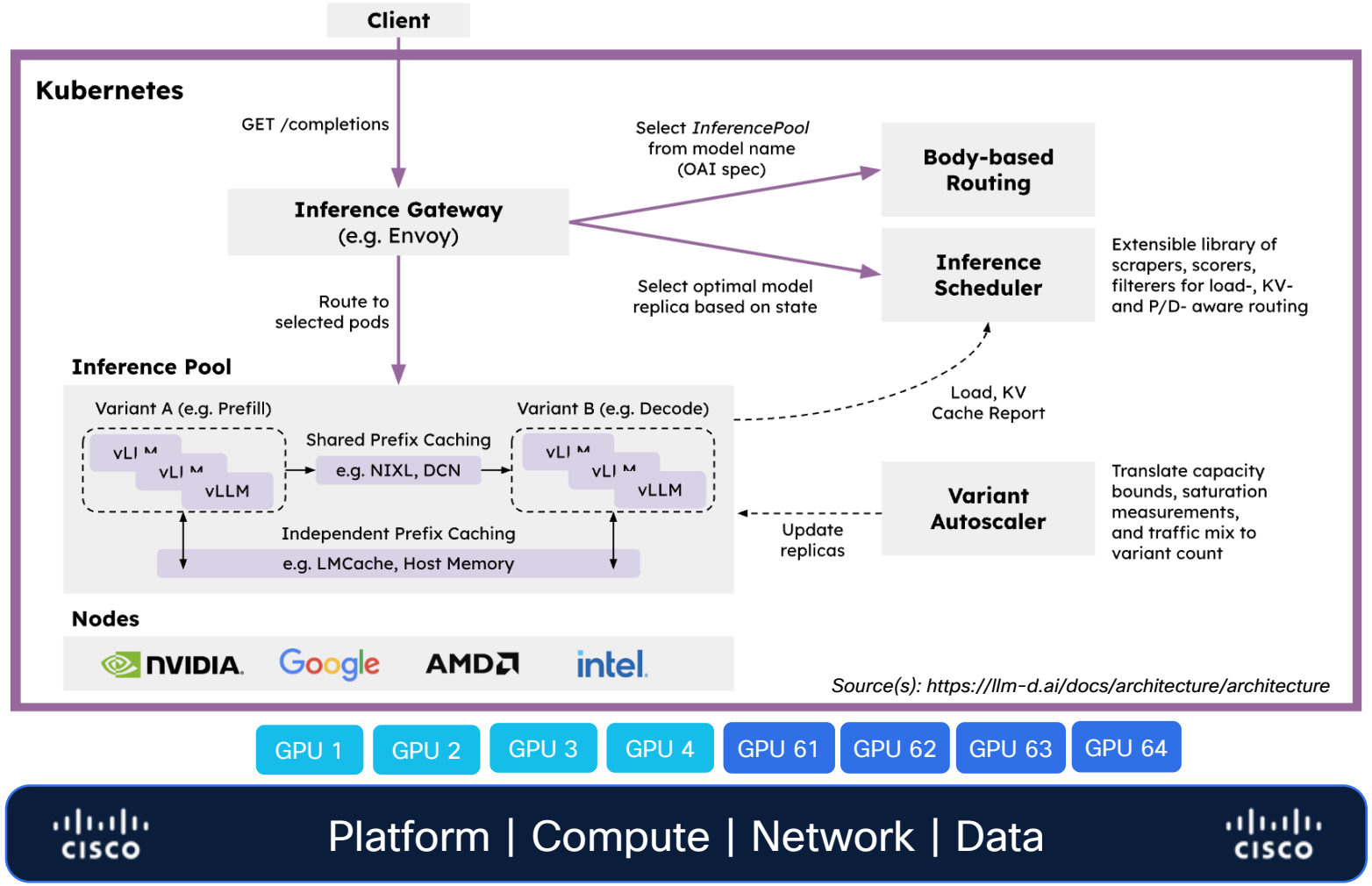
3. Mapping Agentic AI to Infrastructure

Agentic Workload Architecture



llm-d on Red Hat OpenShift

- Optimized Inference Scheduling
- Disaggregated Serving Prefill vs Decode
- Disaggregated Prefix Caching

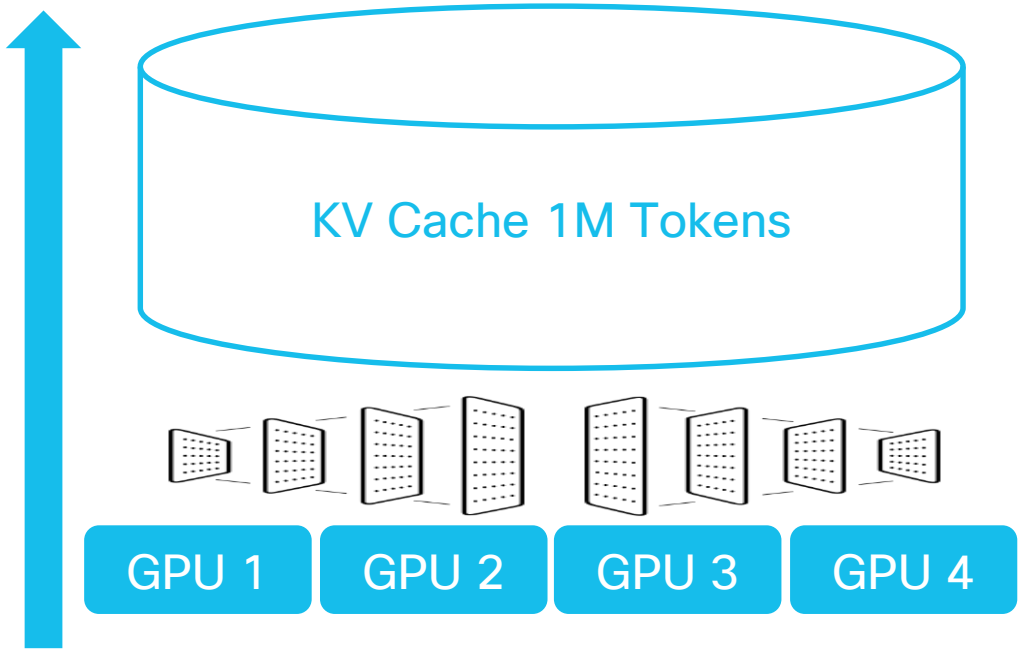


Placing AI Agents

Large Reasoning Model

Better token throughput for long context interactions.

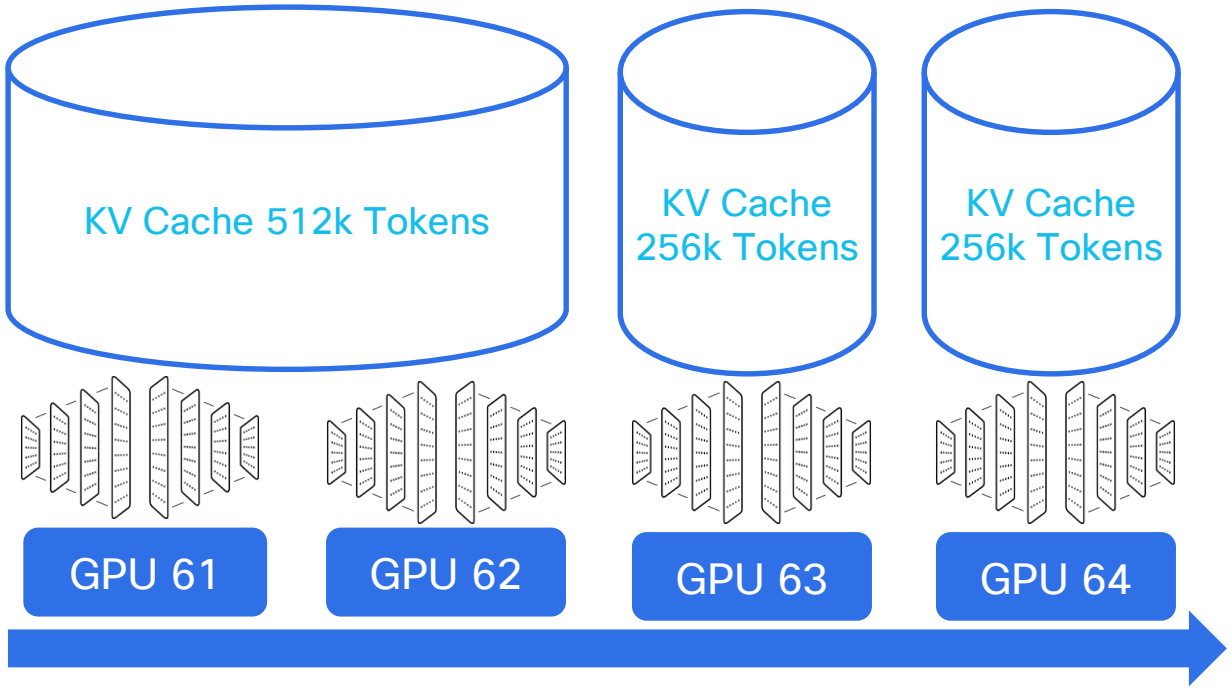
Coordinate and evaluate domain expert models.



Smaller Domain Expert Models

Better performance for short context interactions.

Linear throughput scaling & performance isolation.



Cisco AI Compute Portfolio ”Show me the metal”

For performance heavy use cases
like model training and high scale inferencing

Cisco UCS® C885A M8 Rack Server

NVIDIA HGX platform with

8 NVIDIA H100 NVL/H200 NVL & M300X GPUs

2 AMD 4th Gen/5th Gen EPYC processors

For fine tuning and scalable
inference use cases

Cisco UCS® C845A M8 Rack Server

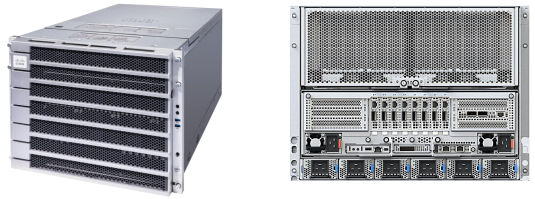
NVIDIA MGX platform with

2/4/6/8 NVIDIA H100 NVL/H200 NVL & L40S/RTX 6000 Pro GPUs

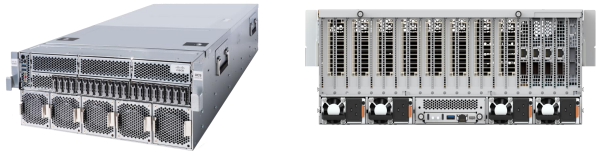
2 AMD 5th Gen EPYC processors

Intra-GPU Connectivity

UCS C885A



UCS C845A

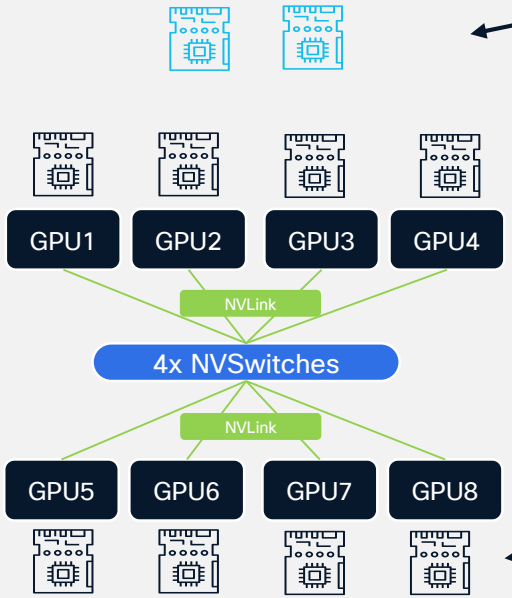


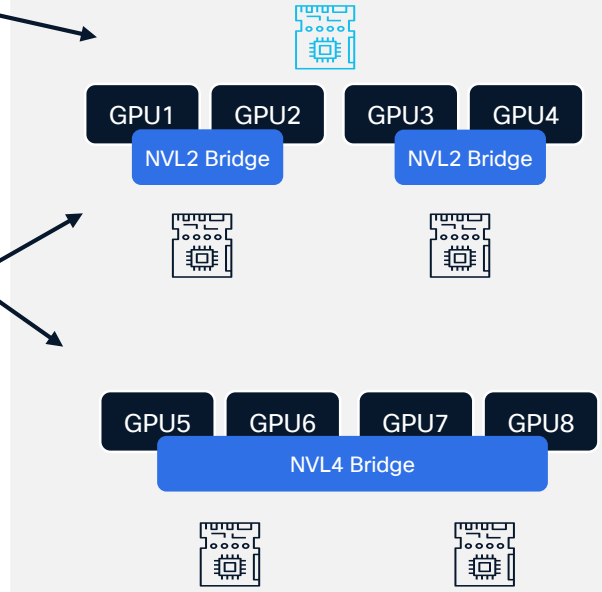
Dedicated NICs for north-south connectivity to front-end networks

Each GPU has multiple NVLink ports and connects to four non-blocking NVSwitches

2 or 4-Way Bridge configurations for intra-GPU connectivity.
Each GPU pair has also dedicated NICs GPU-to-GPU connectivity across nodes

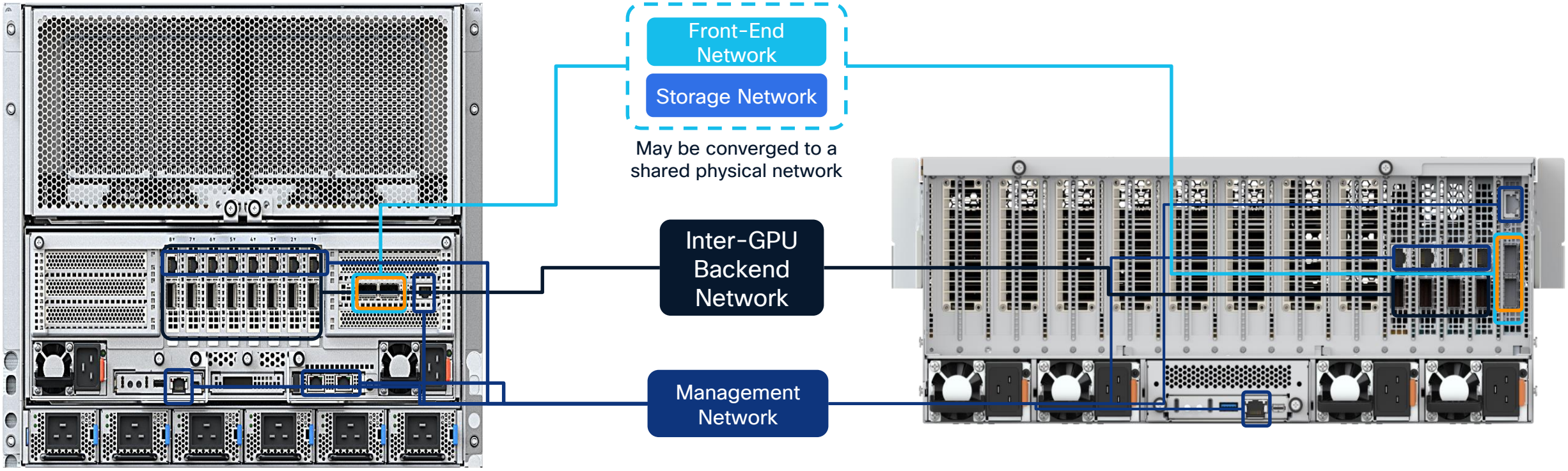
Each GPU has dedicated NICs for GPU-to-GPU connectivity across nodes





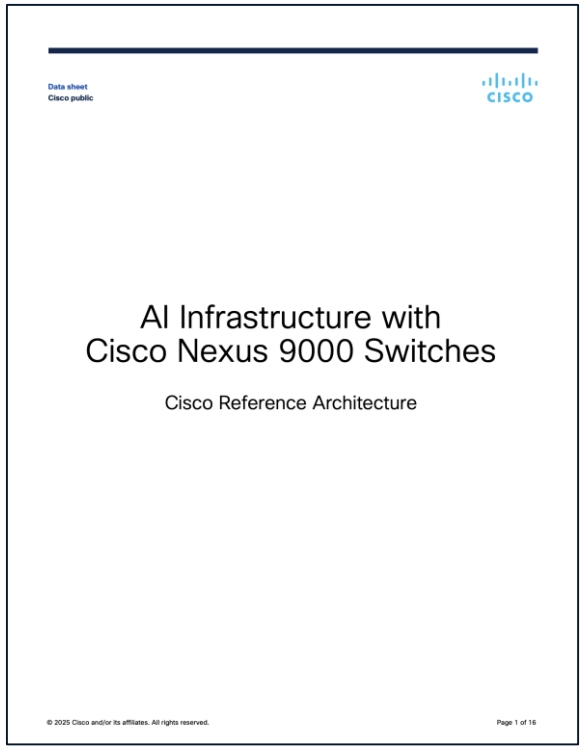
*CPU & PCIe switches omitted for brevity

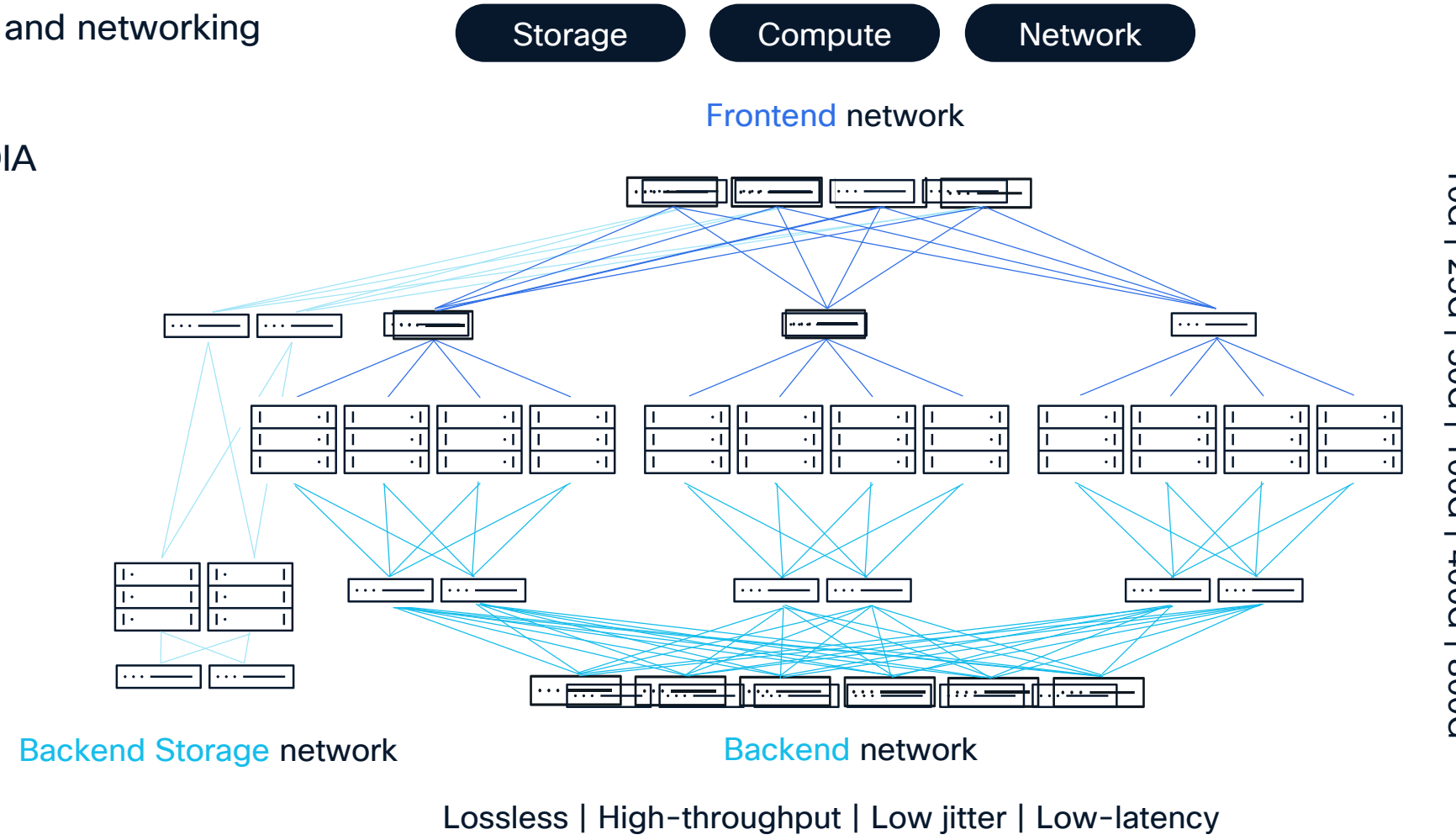
UCS C885A & C845A M8 External Connectivity



Cisco Enterprise Reference Architecture

- AI optimized compute, storage, and networking
- Consistent scaling units
- Validated in partnership of NVIDIA





cisco

Cisco AI PODs

A scalable architecture, built to support any AI workload simply & efficiently

Deploy AI with confidence

Cisco CVD, NVIDIA ERA

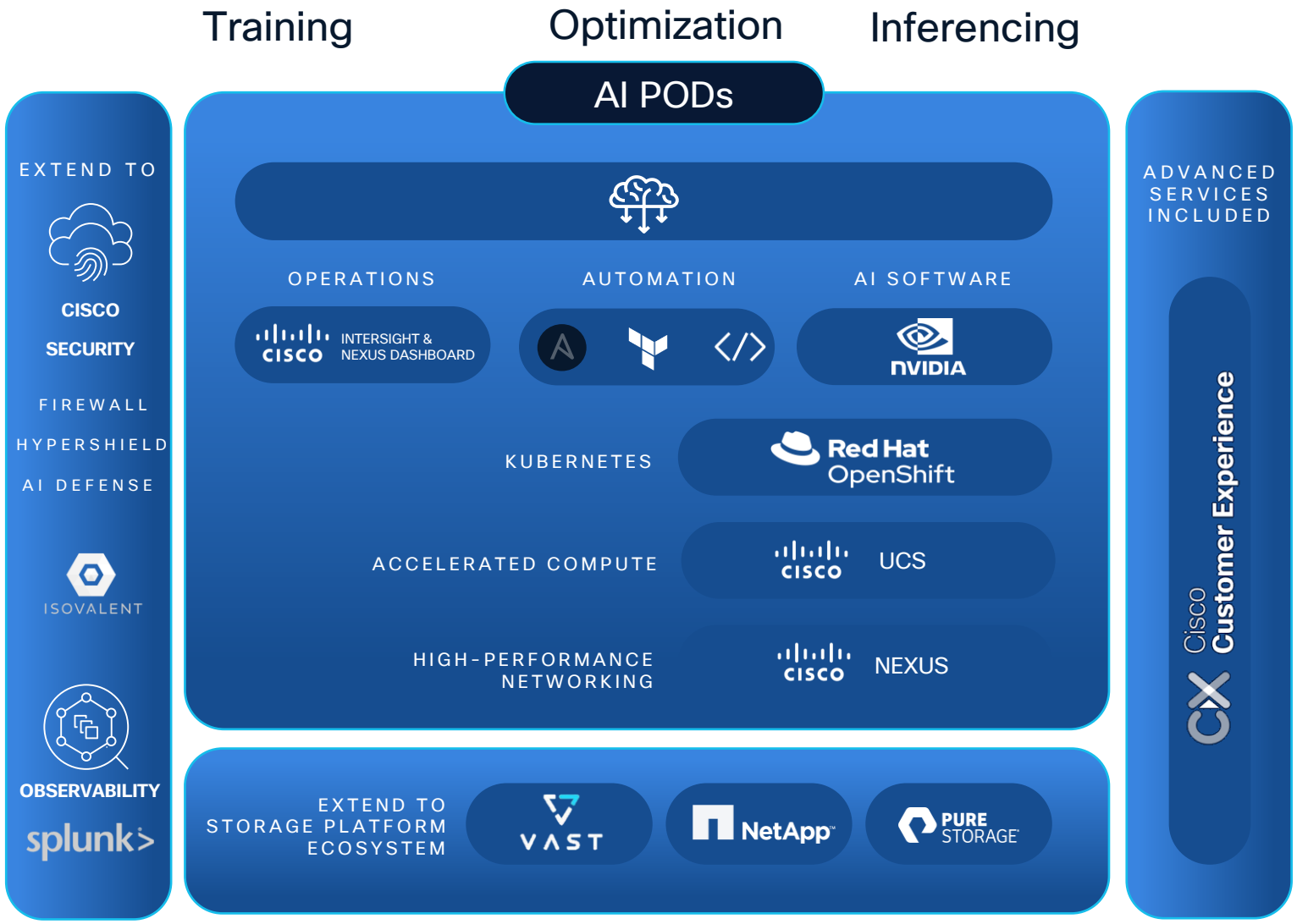
Orderable, use case driven AI-Ready infrastructure stacks

Inferencing. Optimization. Training.

Fully supported stack including Cisco and 3rd party components

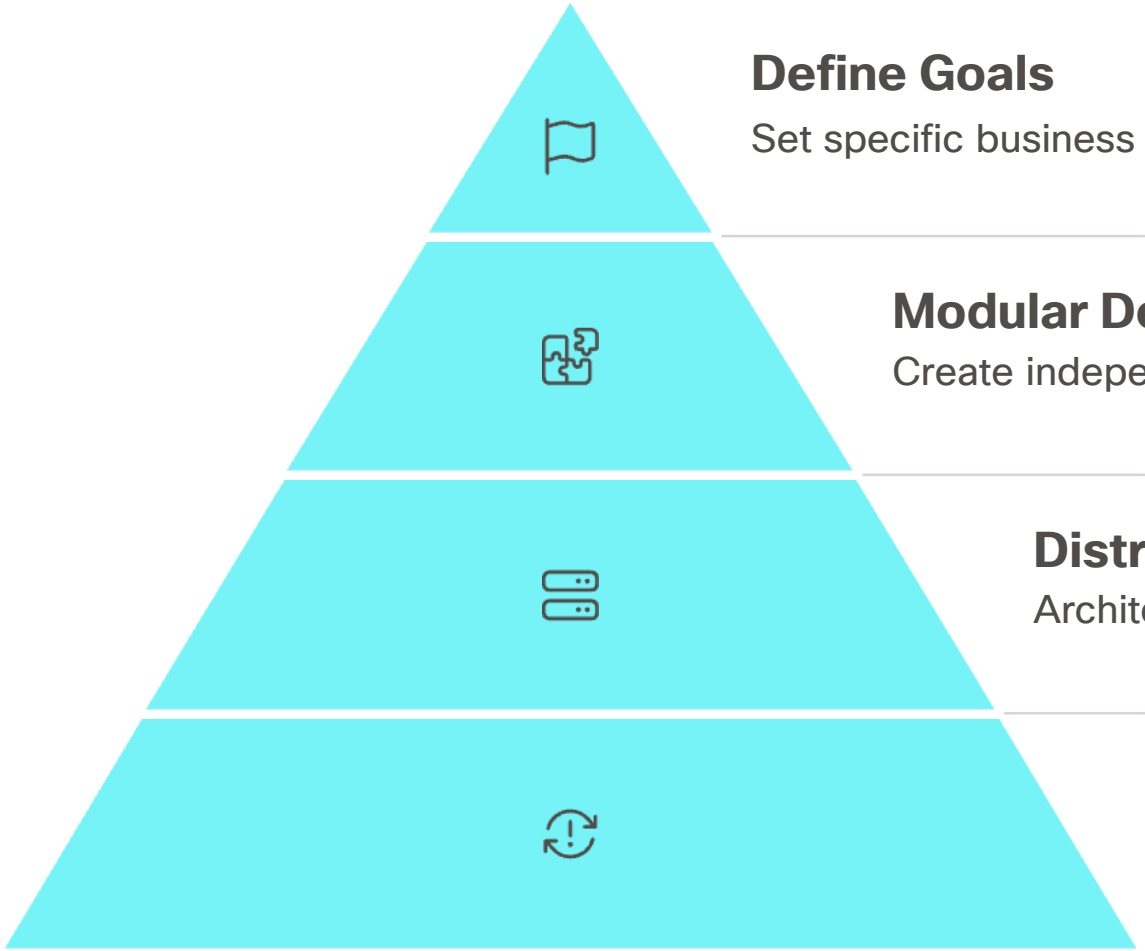
Cisco CX Success Track

Incremental, atomic-level –or- fabric-based cluster scale



4. How can I start the Agentic journey?

The Agentic Journey



Define Goals

Set specific business outcomes

Modular Design

Create independent but collaborative agents

Distributed Systems

Architect for scale and reliability

Feedback Loops

Validate business value

Enable continuous agent learning

Complete your session evaluations

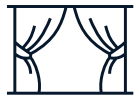
Complete a minimum of 4 session surveys and the Overall Event Survey to be entered in a drawing to win 1 of 5 full conference passes to Cisco Live 2026.

Earn 100 points per survey completed and compete on the Cisco Live Challenge leaderboard.

Level up and earn exclusive prizes!

Complete your surveys in the Cisco Live mobile app.

Continue your education



Visit the Cisco Showcase for related demos.

Book your one-on-one Meet the Engineer meeting.

Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs.

Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

Thank you

cisco Live !

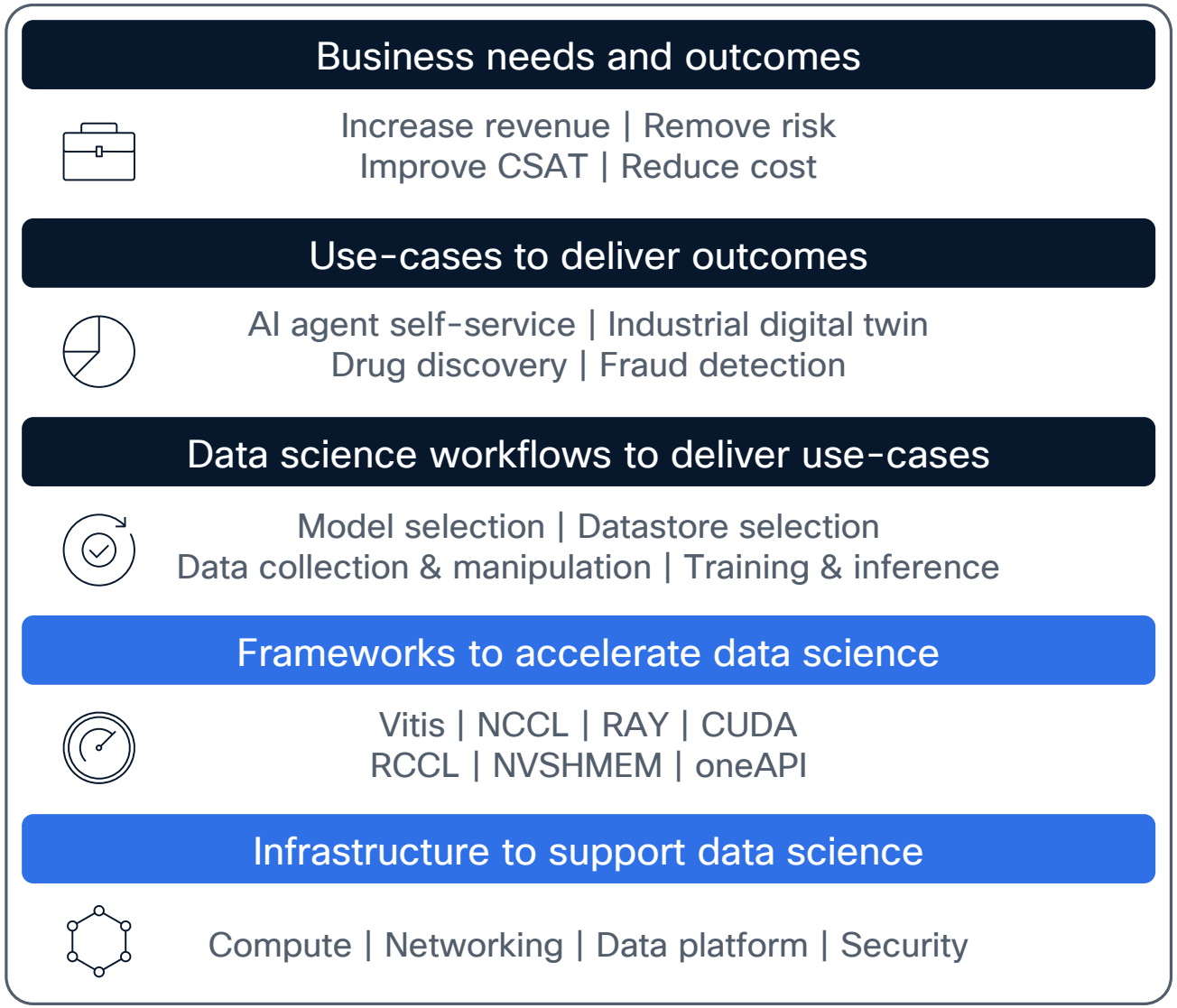
Phases of Inferencing – Why, Where and I/O Profile

Storage Area	Why Needed?	I/O Profile	Recommended Storage Types
Model Storage	Load models for inference	Read-Heavy	NVMe SSD, Local SSD Cache, NAS
Input Data Storage	Ingest and temporarily store inputs	Moderate Read/Write	NVMe SSD, High-Speed NAS
Output Results Storage	Store inference outputs, analytics	Write-Heavy	SSD, Object storage, Databases/Data lakes
KV Cache	Efficiently manage intermediate states	Heavy Read/Write	RAM/VRAM (ideal), NVMe SSD if persisted
RAG Storage	Retrieve relevant contexts/documents	Read-Heavy	SSD/NVMe, In-memory vector DB, Object Storage

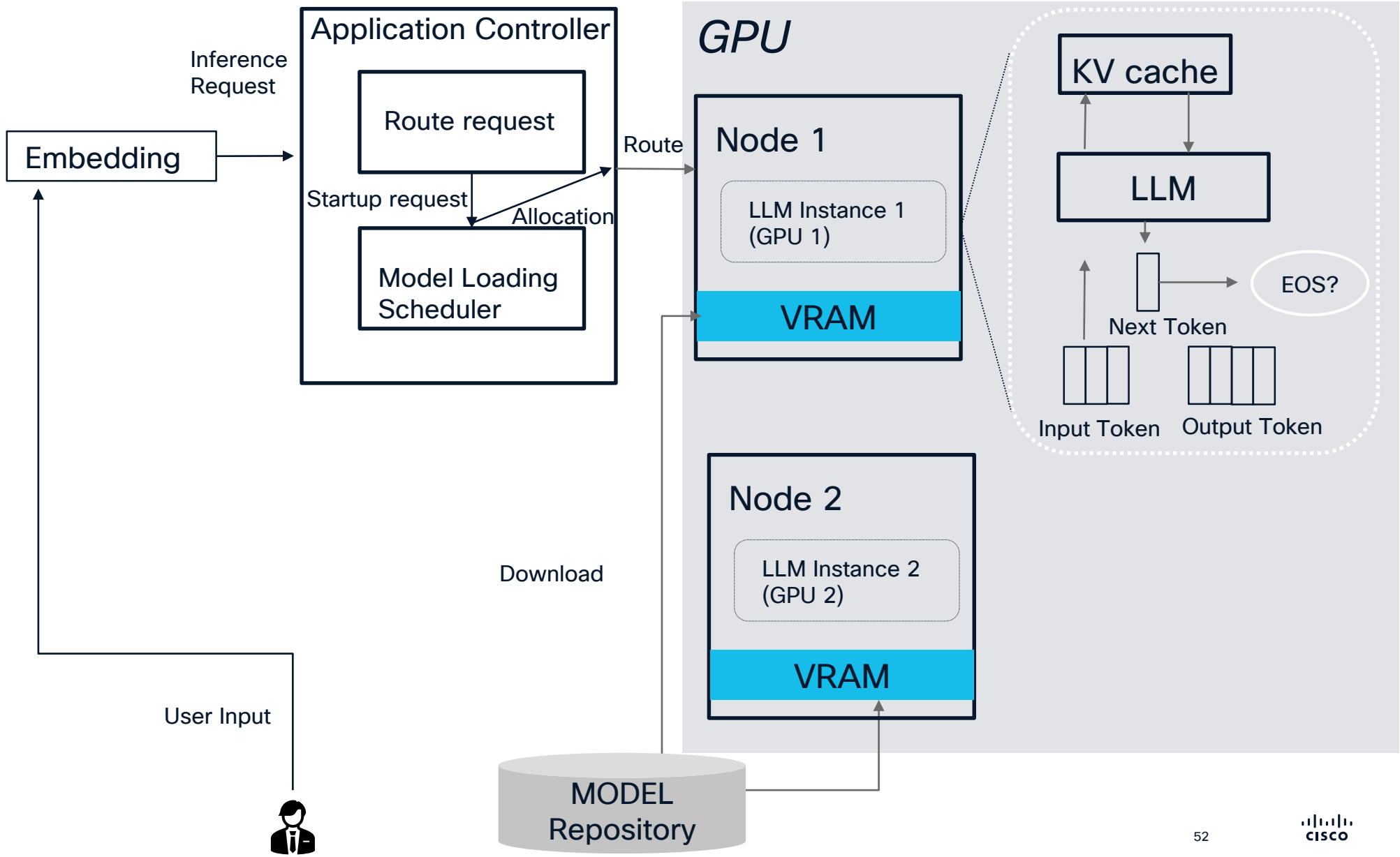

CISCO

Initial questions to define cross-architecture Agentic requirements

- What is the outcome?
- What is the use case?
- What data do I need?
- What model will deliver that use case?
- What is my model size?
- How will I manage resource utilization?
- What security policies will I need to implement?



Inferencing Generative AI Architecture



Information Retrieval Tools

Web Search Tools

- Google Search API
- Bing Search API
- SerpAPI

Vector Databases

- Pinecone (99.9% recall)
- Weaviate
- Milvus

Document Processors

- PyPDF
- Apache Tika
- Supports 1200+ formats

Knowledge Graphs

- Neo4j
- Amazon Neptune
- Billions of relationships