Agentic AI

Surendra Panpaliya

Agenda

MULTI-AGENT ORCHESTRATION WITH LANGGRAPH

SECURITY, GOVERNANCE, RESPONSIBLE AI

COST OPTIMIZATION & MONITORING GPT-5 APPS

FUTURE OUTLOOK: AGENTIC AI, MCP, ENTERPRISE COPILOTS

Agentic AI & Task Chaining

What is Agentic AI?

Core components:

Agents

Tools

Planner

Executor

What is Agentic AI?

Agentic AI = Smart AI agents that

Understand goals

Plan steps

Execute tasks

Work together (like a team)

What is Agentic AI?

New way of using AI

to **plan, execute, and manage tasks**

independently

like an intelligent assistant

that thinks and acts.

What is Agentic AI?

ACT LIKE A RESPONSIBLE ASSISTANT,

DOING TASKS **INDEPENDENTLY**

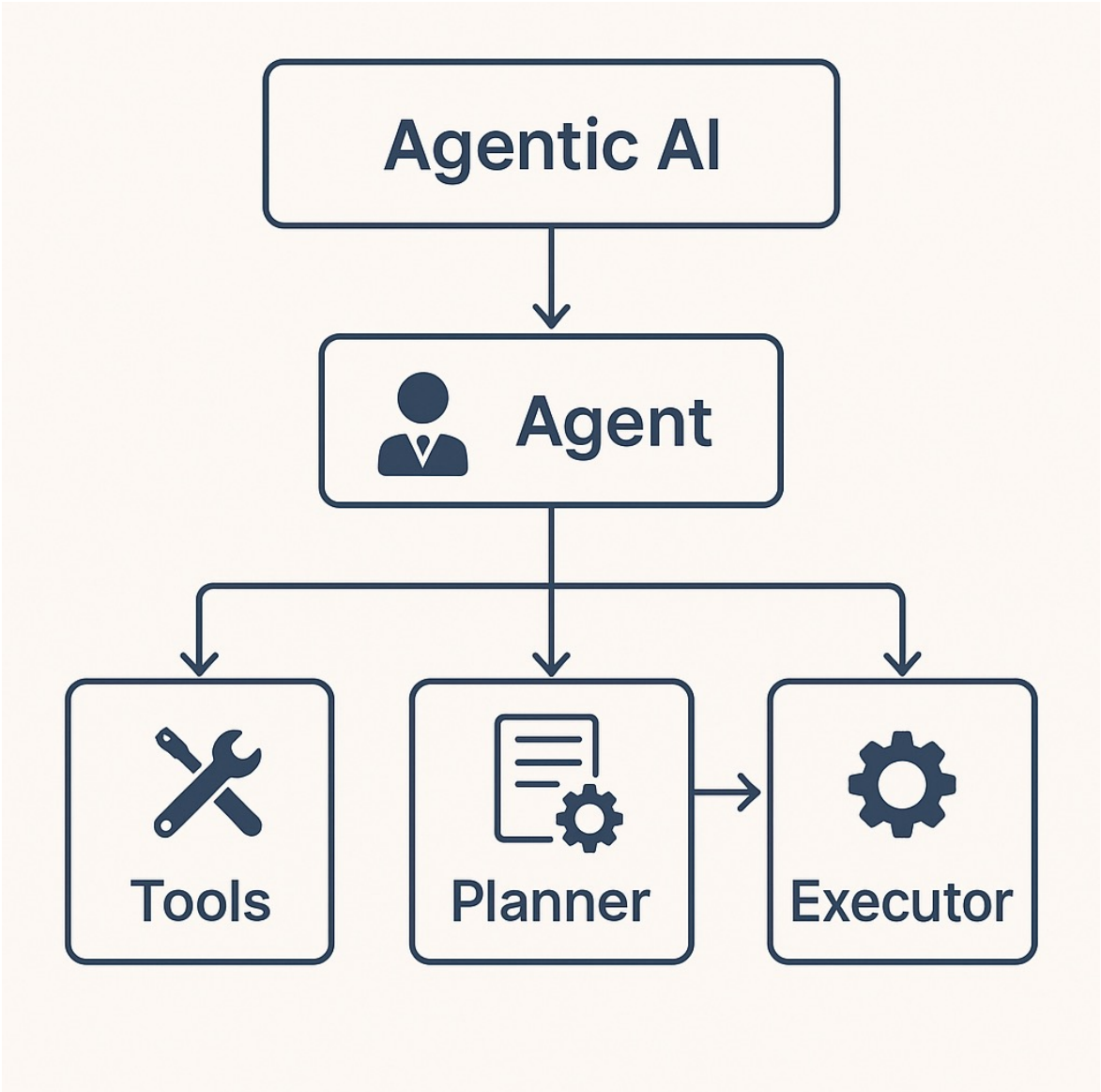
BASED ON GOALS YOU GIVE IT.

Agentic AI uses Agents that can

- Understand goals
- Break them into tasks
- Use tools or APIs to complete tasks
- Adjust actions based on results

Core Components of Agentic AI

1. Agents
2. Tools
3. Planner
4. Executor


Agentic AI
Agent
Tools
Planner
Executor

1. Agents

Individual units with specific roles or goals.

Can reason, learn, and make decisions

based on their goals.

2. Tools

Resources or capabilities available to agents.

For example: APIs, databases,

web search engines, or AI models.

3. Planner

Determines how to achieve the goal.

Decides which agents

should collaborate, and in what order.

4. Executor

Executes the planned tasks.

Coordinates interaction

between agents and tools

to complete tasks.

Core Components of Agentic AI

Component	Purpose
👨‍💼 **Agent**	An intelligent entity that receives instructions and decides what to do.
🧰 **Tool**	A function or API the agent can use (like a search engine, database query, code interpreter, etc.)
📋 **Planner**	Breaks down a big problem into smaller tasks.
⚙️ **Executor**	Executes tasks one by one, using tools, and adjusts based on feedback.

What Are Agent Design Patterns?

AGENT DESIGN PATTERNS DEFINE HOW AN AI AGENT THINKS, PLANS, AND EXECUTES TASKS.

What Are Agent Design Patterns?

These patterns guide:

- How the agent reasons
- How tasks are broken down
- How tools are used
- How the agent adapts to feedback

Key Agent Design Patterns

ReAct (Reasoning and Acting)

Chain-of-Thought (CoT)

Plan-and-Execute (PnE)

ReAct (Reasoning and Acting)

The agent **reasons**

step-by-step and then

decides an action,

repeating until

the final answer is found.

Pattern

Thought → Action → Observation →

Thought → Action → … → Final Answer

Chain-of-Thought (CoT)

The agent is encouraged

to **explain its reasoning process**

before giving an answer.

It doesn’t act with tools

but reasons clearly.

What is Chain-of-Thought (CoT)?

Instead of jumping to the final answer,

the model is instructed to **think aloud**

by breaking the problem into

logical reasoning steps.

What is Chain-of-Thought (CoT)?

Improves accuracy, interpretability

often factual correctness,

especially for complex or

multi-step questions.

Plan-and-Execute (PnE)

The agent **first plans**

the full sequence of tasks,

then executes each step.

What is Plan-and-Execute (PnE)?

Powerful agent pattern that:

First plans a sequence of subtasks

based on the user request.

Then **executes each step**,

often using tools, APIs, or functions.

What is Plan-and-Execute (PnE)?

This improves

reliability,

transparency, and

modularity in AI task solving.

Multi-agent orchestration

- Instead of one “do-everything” agent, use **specialized agents** that collaborate:
- **Planner**: decomposes goals into steps
- **Researcher**: searches/reads knowledge base & tools
- **Coder**: generates/fixes code & tests
- **Reviewer**: checks quality/safety
- **Supervisor**: routes work among agents, decides “done/hand back to user”

Multi-agent orchestration

- Benefits:
- separation of concerns,
- easier testing/guardrails,
- better reuse.

Why LangGraph for orchestration?

- **State graphs** with nodes
- (functions/LLMs/tools) and
- **conditional edges**
- **Checkpoints/memory**
- per session (resume, replay)

Why LangGraph for orchestration?

- Built-in **tool execution** nodes
- Guardrails (insert moderation/validators as nodes)
- Deterministic control flow you can test

Security, governance, responsible AI GPT-5 Applications

Security for GPT-5 Applications

- **a) Data Security**
- **Encryption**: Use TLS in transit, AES-256 at rest for embeddings, chat logs, and documents.
- **Access Control**: Enforce **RBAC/ABAC** (Role/Attribute-Based Access Control) so only authorized users/agents see specific data.
- **Secrets Management**: Store API keys, tokens, and DB credentials in **vaults** (HashiCorp Vault, AWS Secrets Manager), never in code.
- **Secure Vector DBs**: When using Milvus/Qdrant/pgvector, enable TLS + auth, and restrict network access.

b) Application Security

- **Input Sanitization**: Guard against prompt injection or malicious tool calls (e.g., “delete all records” disguised in user input).
- **Sandboxing**: Run code-generation/execution in **isolated containers** with resource limits.
- **API Gateway**: Rate limiting + WAF rules to block abuse and DOS attacks.

c) Monitoring & Incident Response

- **Audit Logs**: Record queries, model outputs, and tool invocations.
- **Anomaly Detection**: Flag unusual query volumes or PII exposure attempts.
- **Alerts**: Integrate with SIEM (Splunk, Azure Sentinel) for real-time monitoring.

2. Governance in GPT-5 Applications

a) Policy & Compliance

- **Data Retention Policies**: Define how long prompts/responses are stored.
- **Right to be Forgotten**: Implement deletion workflows for user-specific embeddings (GDPR/CCPA).
- **Cross-Border Data**: Ensure embeddings & logs stay in compliant regions (EU/India/US).

b) Lifecycle Governance

- **Model Registry**: Track which GPT-5 versions or fine-tunes are in use.
- **Prompt/Template Management**: Centralize approved prompts, enforce version control (like code).
- **Change Control**: Any update to prompts/tools should go through review → test → approval.

c) Oversight Structures

- **AI Governance Board**: Cross-functional (IT, Legal, Compliance, Business).
- **Audit Trails**: Full traceability — which model, which embeddings, which vector search, which answer.
- **KPIs**: Track **accuracy, latency, cost, user satisfaction, compliance incidents**.

Responsible AI in GPT-5 Applications

a) Fairness & Bias

Bias Audits: Regularly test outputs for demographic, geographic, or gender bias.

Balanced Training/Evaluation Data: Especially for fine-tuned GPT-5 models.

Human Review Loops: For sensitive domains (finance, healthcare, hiring).

b) Transparency

- **Explainability**: Provide citations to retrieved docs (RAG) so users know “where the answer came from.”
- **Disclaimers**: Label AI-generated output vs. human content.
- **Confidence Scores**: Share retrieval confidence, not just polished text.

c) Accountability

- **Human-in-the-Loop**: Approval workflows for high-risk actions (contracts, hiring, medical advice).
- **Incident Reporting**: Allow users to flag “incorrect / unsafe output.”
- **Escalation Paths**: Define who in the organization is accountable for AI decisions

d) Safety Guardrails

- **Toxicity Filters**: Pre/post-process outputs through moderation models.
- **Domain Guardrails**: Restrict LLM to specific knowledge bases (via RAG), disallow free hallucination.
- **Evaluation Benchmarks**: Continuously test with red-team prompts (prompt injection, jailbreaks, policy violation tests).

What is GenAI Governance?

Set of practices ensuring that

AI solutions comply with organizational policies,

legal frameworks, ethical standards,

and business objectives.

What is GenAI Evaluation?

Systematic assessment

to ensure AI models are accurate,

fair, reliable, robust, and

aligned with business requirements.

What is GenAI Evaluation?

Evaluation is a structured approach

to measure and improve the effectiveness,

accuracy, fairness, and safety

of Generative AI (GenAI) models.

Why Is GenAI Evaluation Essential at Walmart?

- Ensuring Accuracy & Reliability
- Mitigating Risks & Biases
- Enhancing Customer Trust
- Regulatory Compliance
- Continuous Improvement

1️⃣ Ensuring Accuracy & Reliability

Confirms the AI-generated responses and

actions align with Walmart's business requirements.

Maintains trust by providing consistently accurate results.

2️⃣ Mitigating Risks & Biases

Detects and reduces unwanted biases

that could harm Walmart's reputation.

Prevents incorrect decisions

that could lead to financial or operational risks

3️⃣ Enhancing Customer Trust

Customers interact confidently

with reliable, transparent AI solutions.

Strengthens Walmart’s brand value

by ensuring fairness and trustworthiness

in automated interactions.

4️⃣ Regulatory Compliance

Ensures that Walmart’s AI solutions

comply with global regulations and

standards, avoiding legal issues.

5️⃣ Continuous Improvement

Provides insights into

model performance,

highlighting areas

for further development.

5️⃣ Continuous Improvement

Enables Walmart to maintain competitive advantage through adaptive and improved GenAI solutions.

Key Metrics in GenAI Evaluation

Accuracy & Precision:

Correctness of AI responses.

Fairness & Bias:

AI decisions equitable across diverse user groups.

Key Metrics in GenAI Evaluation

Robustness:

Stability of AI under various conditions.

Safety & Ethics:

AI adherence to ethical guidelines and policies.

Potential Risks Without Effective Evaluation

- Misleading customer interactions.
- Financial losses from incorrect AI decisions.
- Reputational damage from
- biased or inappropriate outputs.
- Legal and compliance risks.

Practical Steps for Walmart GenAI Evaluators

Set clear, measurable criteria aligned with business objectives.

Implement standardized evaluation frameworks and tools.

Conduct regular audits and reviews.

Use feedback loops for continuous refinement

MCP Server using LangChain

Surendra Panpaliya
GKTCS Innovations
https://www.gktcs.com

Model Context Protocol (MCP)

Standardized framework

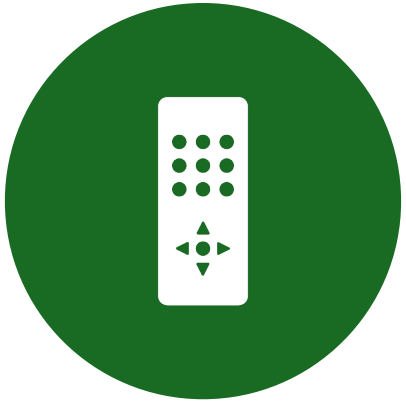
Allows **AI models**

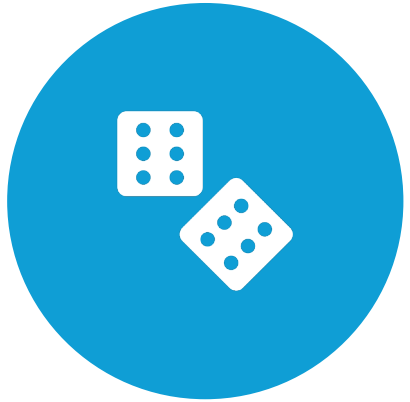
To **interact with real-world tools,**

APIs, and data sources

in a **safe, modular, and controlled way**.

What is MCP (Model Context Protocol) Server?





LIKE A **TOOLBOX AND** **REMOTE CONTROL** FOR AI MODELS.

What is MCP (Model Context Protocol) Server?

Lets AI systems not just think and answer

but also **take real-world actions**

by calling APIs, tools, or databases

in a safe, modular way.

MCP Workflow

[AI Model / LLM]

↓

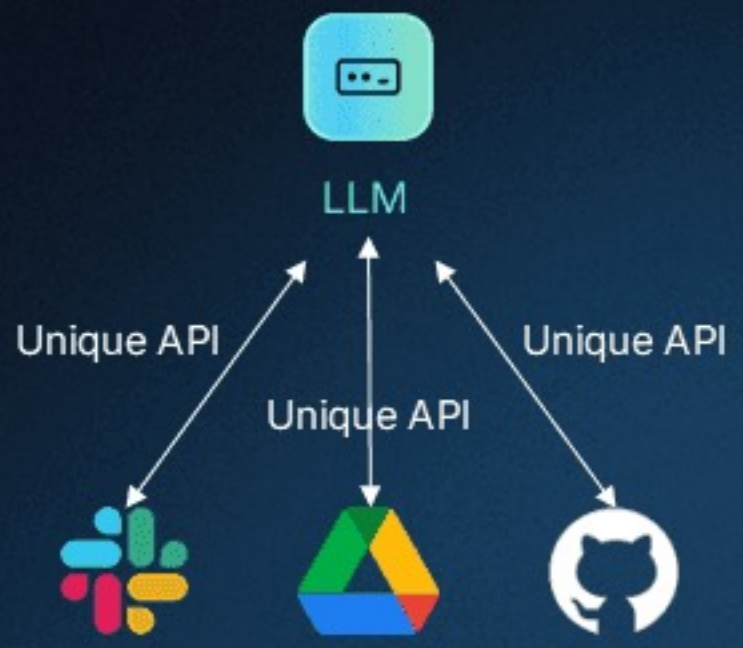
[MCP Protocol Layer]

↓

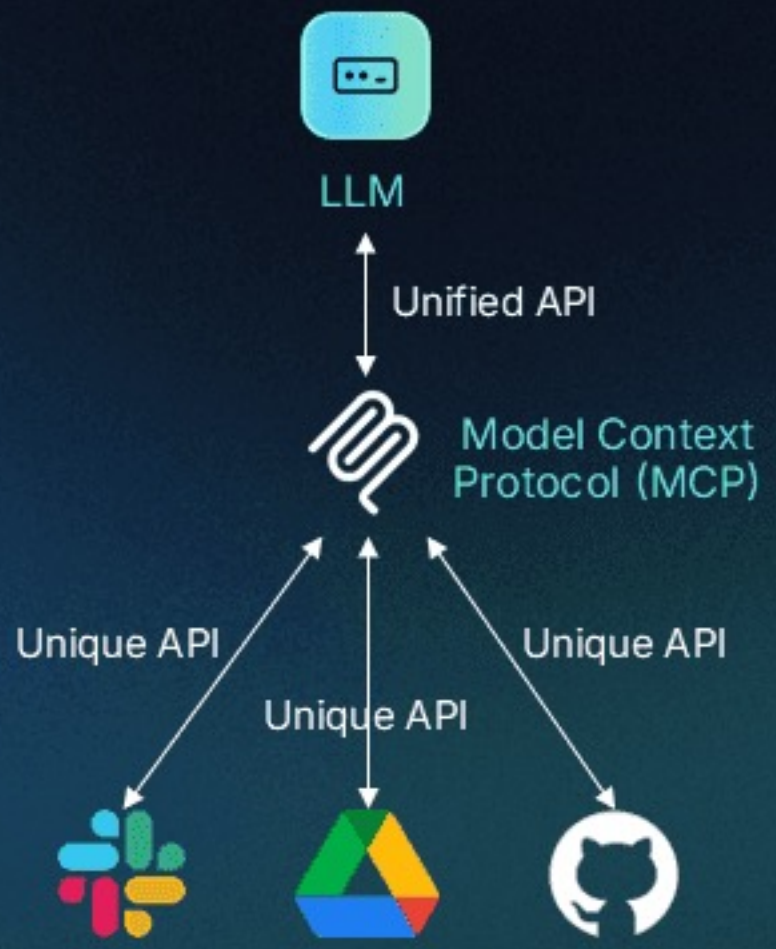
[Tools / APIs / Databases / Systems]

descope

Before MCP

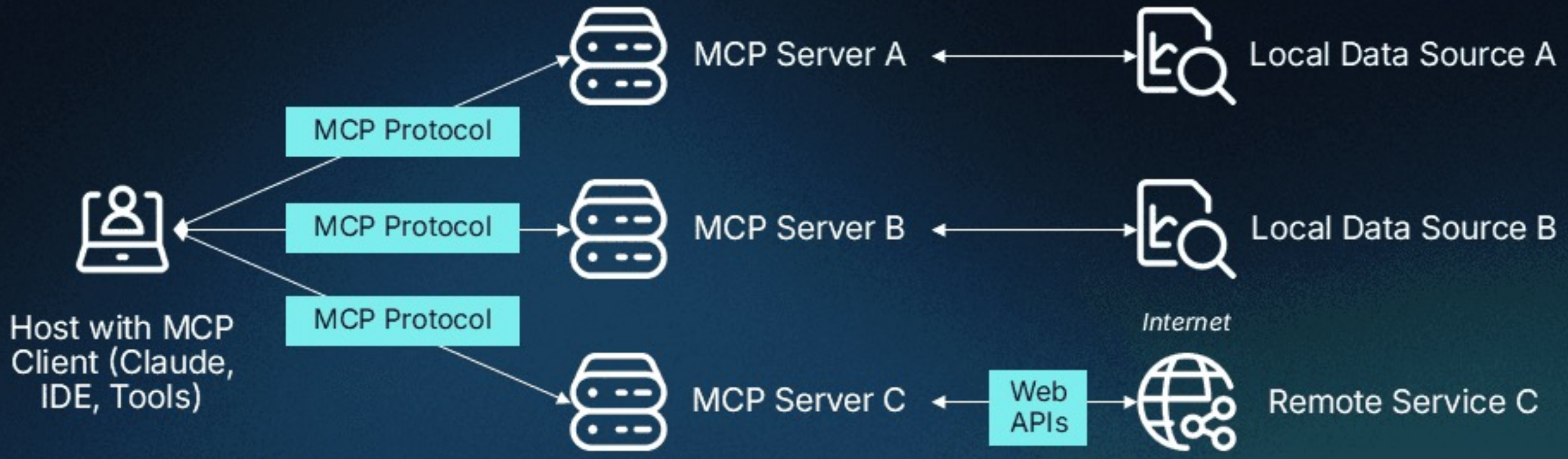


After MCP



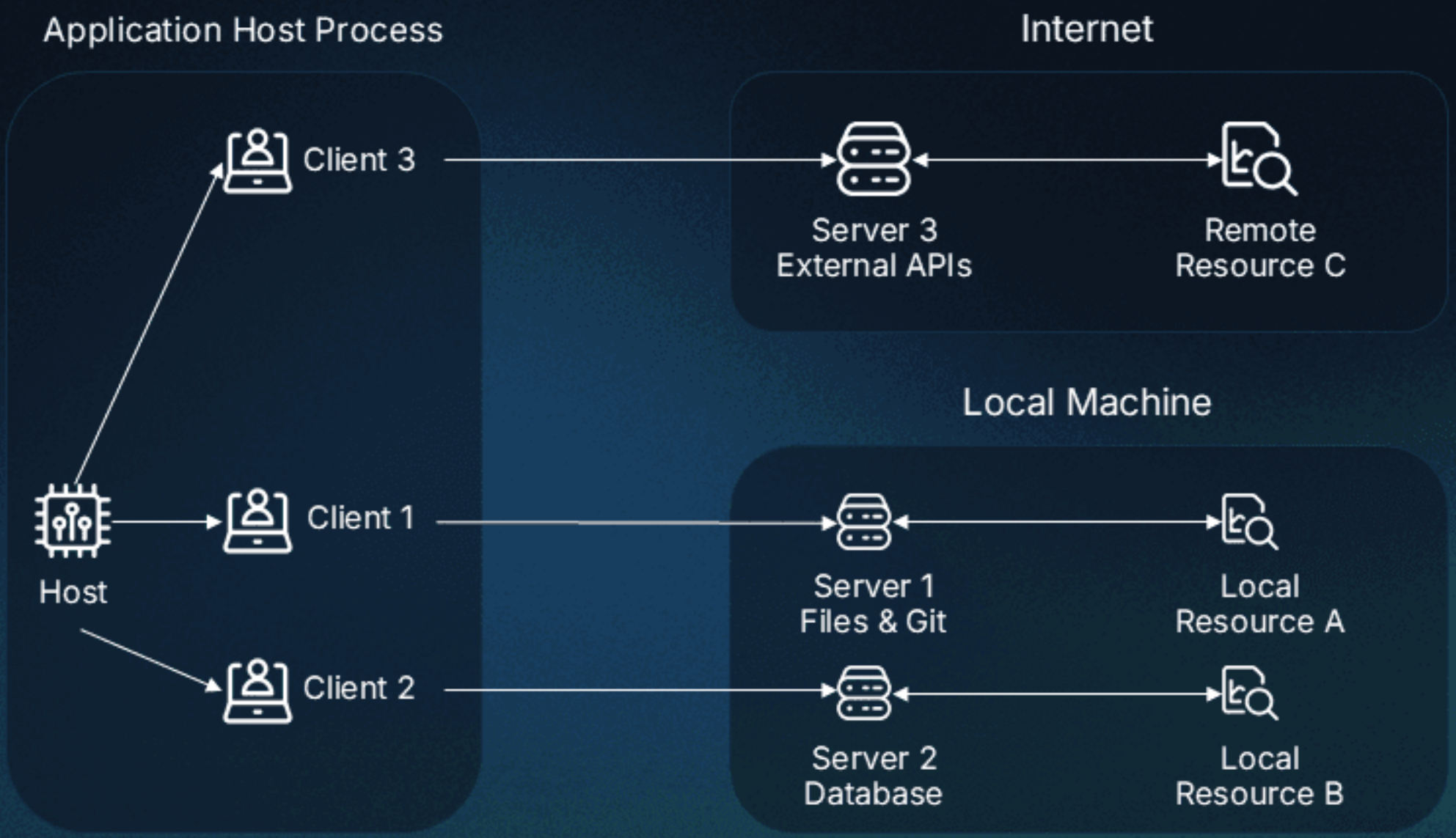
MCP Architecture

descope



MCP Core Components

descope



Why Do We Need MCP Server?

Large Language Models

Can **think**, **reason**, **write**, and **chat**.

But **cannot directly act**

Why Do We Need MCP Server?

MCP Server bridges

this gap by letting AI interact

with the real world

securely and efficiently.

RAG vs Agentic AI vs MCP

Feature / Aspect	RAG	Agentic AI	MCP
Main focus	Retrieval + grounded generation	Planning, reasoning, tool orchestration	Standardized context & tool interface
Handles multi-step tasks	❌	✅	❌ (but can be used by agents)
Requires vector DB	Usually	Optional	Optional

RAG vs Agentic AI vs MCP

Feature / Aspect	RAG	Agentic AI	MCP
Tool/API integration	Minimal	Core feature	Yes, as standardized MCP tools
Interoperability	❌	❌ (custom per agent)	✅ cross-app/LLM
Example in Walmart	Return policy Q&A	Return eligibility + logistics	Serve policy DB & inventory API to any LLM client

References

https://www.descope.com/learn/post/mcp

https://modelcontextprotocol.io/introduction

https://youtu.be/GQDHxlKJe_M

https://codingscape.com/blog/how-model-context-protocol-mcp-works-connect-ai-agents-to-tools

https://github.com/modelcontextprotocol

References

https://github.com/modelcontextprotocol/python-sdk?tab=readme-ov-file#mcp-python-sdk

https://claude.ai/public/artifacts/aed32faf-a9bc-43b8-8fd0-eb104a0cb261

https://claude.ai/public/artifacts/0a8124b7-3e44-4ba4-a159-b29669fcc799

Happy Learning!! Thanks for Your Patience ☺

Surendra Panpaliya

GKTCS Innovations