Lecture 9: Agentic Protocols

Mia Mohammad Imran

So far we talked about

- Prompt engineering
- RAG-based grounding
- CLI Agents
- Agentic Design Patterns
- Personalized AI Assistant
- Agent Orchestration

Now the question becomes:

> *How do multiple agents, tools, and systems communicate in a structured and reliable way?*

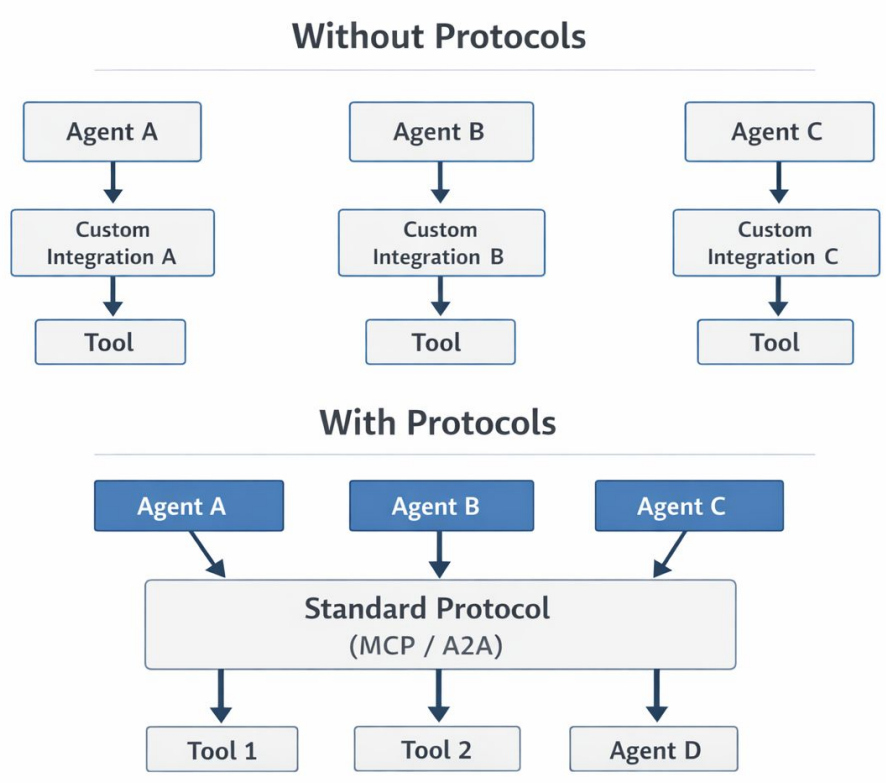
This is the role of agentic protocols.

What Are Agentic Protocols?

Agentic protocols are **standardized communication interfaces that allow AI agents to interact with tools, data sources, and other agents**

They define:

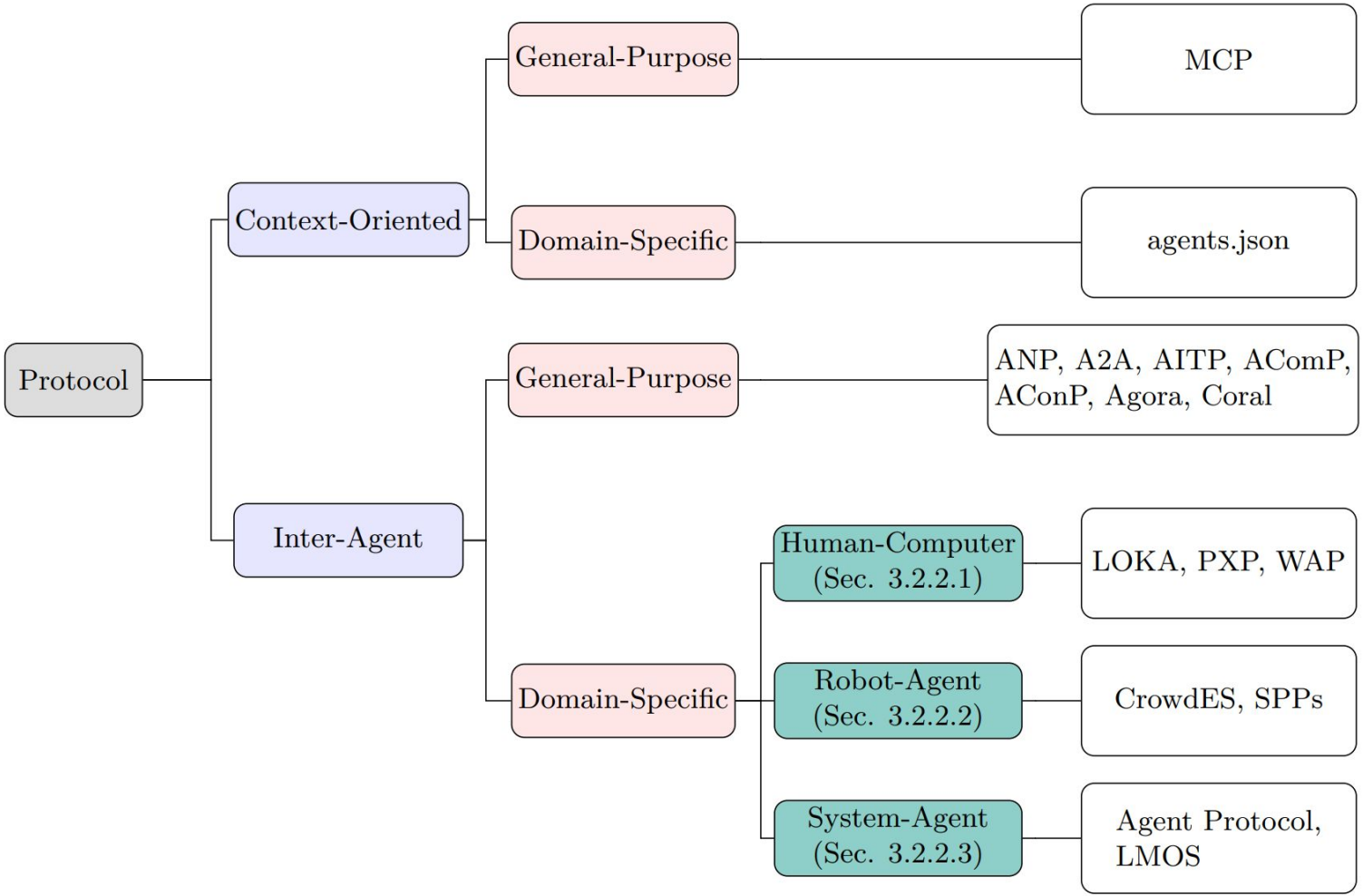
- How agents exchange messages
- How agents access tools
- How context is shared
- How tasks are delegated
- How results are returned



Types of Agentic Protocols

Protocol Type	**Purpose**
Tool Protocols	Agent ↔ Tool interaction
Agent Communication Protocols	Agent ↔ Agent interaction

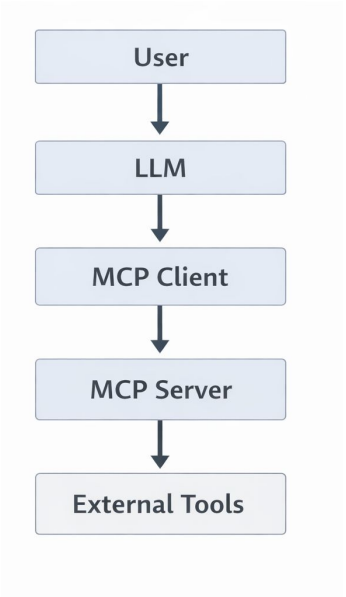
Protocol	**Role**
Model Context Protocol (MCP)	Tool interaction
Agent-to-Agent (A2A)	Agent communication


Protocol
Context-Oriented
General-Purpose
MCP
Domain-Specific
agents.json
Inter-Agent
General-Purpose
ANP, A2A, AITP, AComP, AConP, Agora, Coral
Domain-Specific
Human-Computer (Sec. 3.2.2.1)
LOKA, PXP, WAP
Robot-Agent (Sec. 3.2.2.2)
CrowdES, SPPs
System-Agent (Sec. 3.2.2.3)
Agent Protocol, LMOS

Model Context Protocol (MCP) [Claude]

Model Context Protocol standardizes how models access external capabilities such as tools, APIs, files, and databases

Instead of each tool using a custom interface, MCP provides a common interface for exposing capabilities to agents



MCP Components

MCP Client: The application using the LLM

Examples:

- IDE assistants
- CLI agents
- Coding copilots

The client interprets the model's requests and communicates with MCP servers

MCP Components

MCP Server: Exposes tools to agents

Example tools:

- list_files
- read_file
- run_tests
- search_repo
- edit_file

Each tool has a structured description: called ***tool_schema***

Tool Schema

Tools must describe:

- name
- description
- parameters
- output format

Tool: **search_repo**

Description:
Search repository files for a query.

Parameters:
query: string
path: optional string

Returns:
list of matching files

Example MCP Workflow

User request: *Fix failing unit test in repository*

Agent Workflow:

- Planner Agent
- Code Agent
  - run_tests tool
- Review Agent

Tool execution occurs through MCP

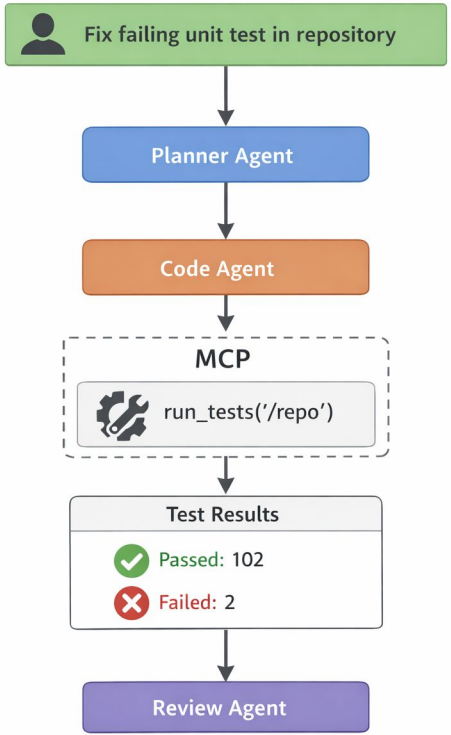
Example call: run_tests(path="/repo")

Result returned:

passed: 102,

failed: 2

The agent continues reasoning using this information



MCP vs RAG

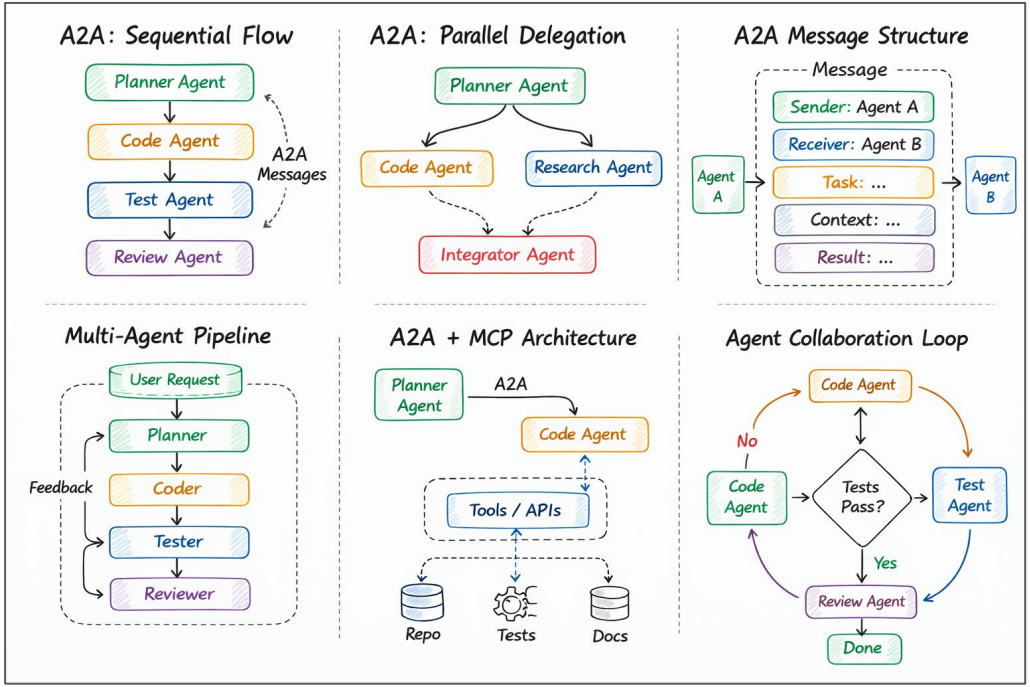
Often People confuse between MCP and RAG

- RAG retrieves documents to ground answers in external information
- MCP enables agents to execute actions in external systems

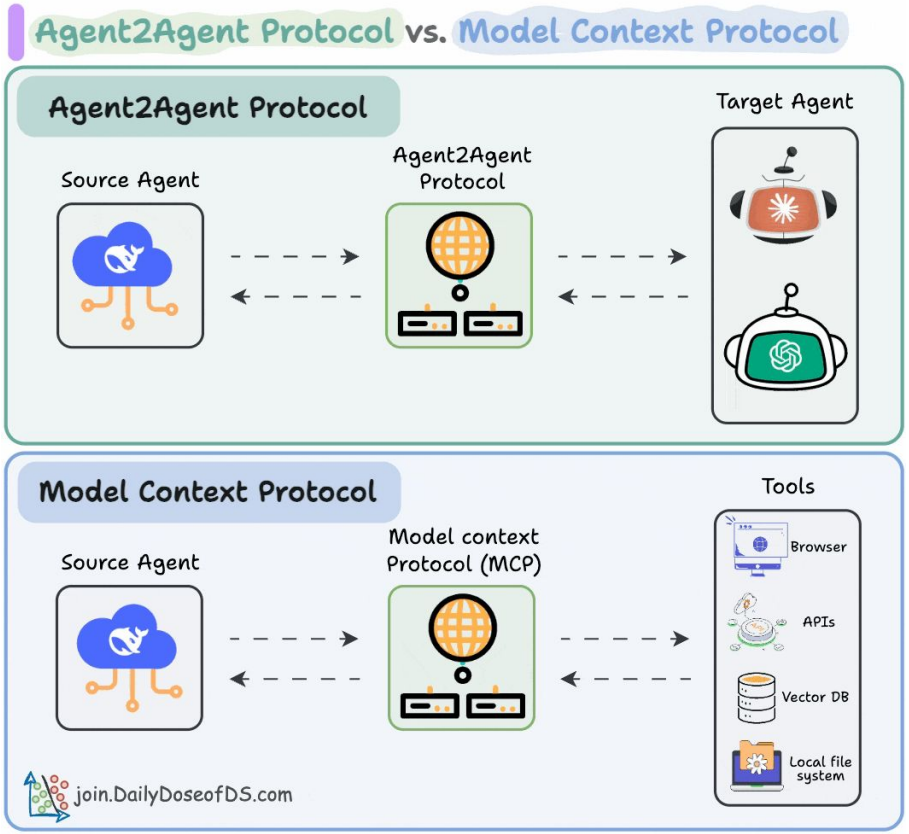
Concept	**Purpose**
Prompting	guide reasoning
RAG	retrieve external knowledge
Tool calling	perform external actions
MCP	standardize tool interface

Agent-to-Agent Communication (A2A) [Google]

- MCP handles **tool access**, but agents must also communicate with each other
- This is where **Agent-to-Agent communication protocols (A2A)** are used
- A2A enables agents to exchange structured messages



A2A vs MCP


Agent2Agent Protocol vs. Model Context Protocol
Agent2Agent Protocol
Source Agent
Agent2Agent Protocol
Target Agent
Model Context Protocol
Source Agent
Model context Protocol (MCP)
Tools
Browser
APIs
Vector DB
Local file system
join.DailyDoseofDS.com

Frameworks Supporting Agentic Protocols

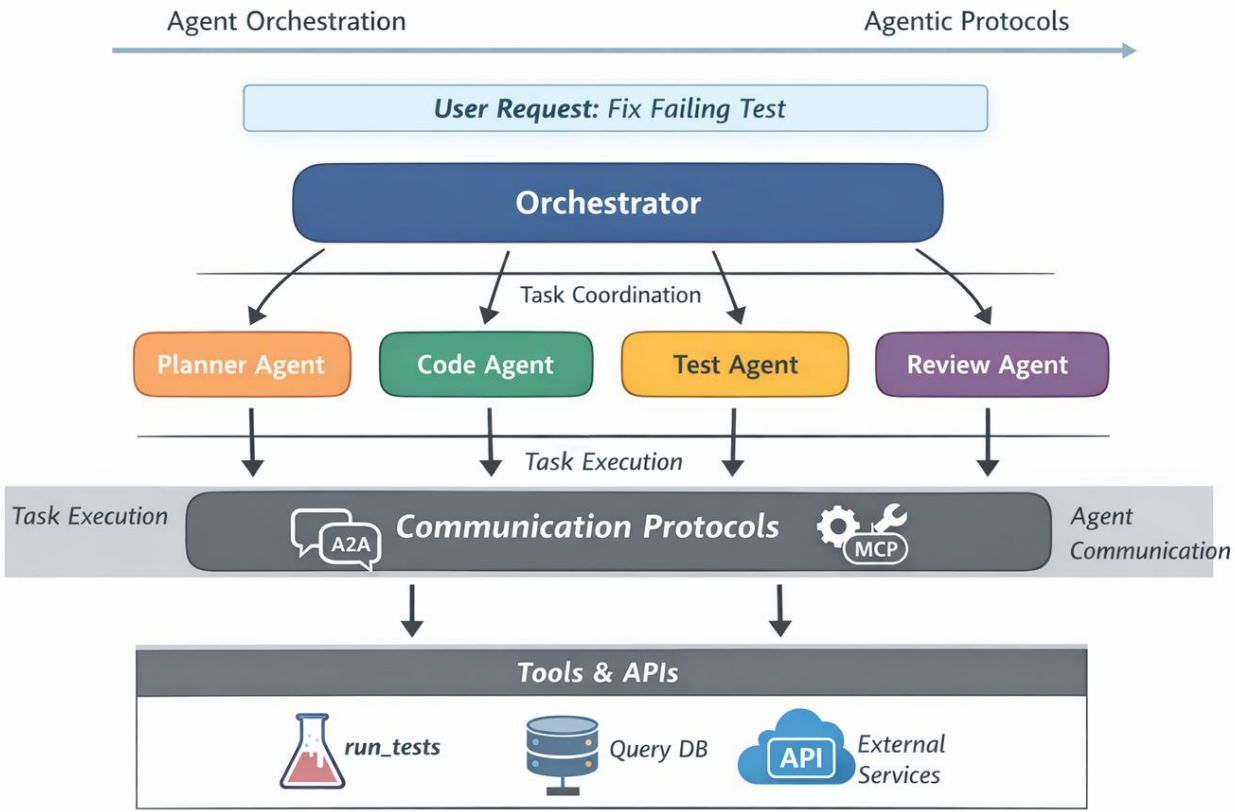
LangChain: Tool-calling abstractions

LangGraph: Graph-based agent workflows

AutoGen: Multi-agent conversational frameworks

CrewAI: Role-based multi-agent collaboration

Agentic Protocols and Agent Orchestration



Other Protocols: https://arxiv.org/pdf/2504.16736

Entity	Scenarios	Protocol	Proposer	Application Scenarios	Key Techniques	Development Stage
Context-Oriented	General-Purpose	MCP Anthropic (2024)	Anthropic	Connecting agents and resources	RPC, OAuth	Factual Standard
	Domain-Specific	agent.json WildCardAI (2025)	Wildcard AI	Offering website information to agents	/.well-known	Drafting
Inter-Agent	Genreal-Purpose	A2A Google (2025)	Google	Inter-agent communication	RPC, OAuth	Landing
		ANP Chang (2024)	ANP Community	Inter-agent communication	JSON-LD, DID	Landing
		AITP NEAR (2025)	NEAR Foundation	Inter-agent communication	Blockchain, HTTP	Drafting
		AComP AI and Data (2025)	IBM	Multi agent system communication	OpenAPI	Drafting
		AConP Cisco (2025)	Langchain	Multi agent system communication	OpenAPI, JSON	Drafting
		Coral citeagentcoralprotocol	The Coral Community	Multi agent system communication	-	Drafting
		Agora Marro et al. (2024)	University of Oxford	Meta protocol between agents	Protocol Document	Concept
	Domain-Specidic	LMOS Eclipse (2025)	Eclipse Foundation	Internet of things and agents	WOT, DID	Landing
		Agent Protocol AlEngineerFoundation (2025)	AI Engineer Foundation	Controller-agent interaction	RESTful API	Landing
		LOKA Ranjan et al. (2025)	CMU	Decentralized agent system	DECP	Concept
		PXP Srinivasan et al. (2024)	BITS Pilani	Human-agent interaction	-	Concept
		CrowdES Bae et al. (2025)	GIST.KR	Robot-agent interaction	-	Concept
		SPPs Gąsieniec et al. (2024)	University of Liverpool	Robot-agent interaction	-	Concept

Key Takeaways

- AI systems are evolving from single models to **agent ecosystems**
- Agentic protocols enable structured communication between components
- **MCP standardizes tool access for agents**
- **A2A enables communication between multiple agents**
- Together they support scalable multi-agent AI systems