COURSE GLOSSARY

Building Scalable Agentic Systems

Agent Card: A metadata descriptor that advertises an agent's capabilities, supported features, and usage details to other agents or clients to enable discovery and standardized interaction

Agent Executor: A standardized component or protocol mechanism that forwards user context and execution requests from a client agent to a remote agent and manages the execution lifecycle

Agent-to-Agent (A2A): A protocol or framework for enabling agents to discover, communicate, and delegate tasks to other agents, including mechanisms like agent discovery, event queues, and standardized executors

Agentic System: A complete application composed of one or more AI agents plus surrounding components like user interfaces, storage, tool integrations, and orchestration layers that enable agents to operate in production

AI Agent: A software system that receives user inputs, uses a model (often an LLM) to interpret tasks, optionally calls external tools, and returns actions or responses to accomplish goals

Artifact (in A2A): A packaged result produced by a remote agent that contains the outcome, description of work performed, and contextual text used to close the interaction and provide reusable outputs

Caching: The strategy of storing prior tool or model responses to reuse for repeated or common queries, reducing latency, load, and cost when the underlying data is sufficiently stable

Fine-tuning: The process of continuing to train a pre-trained model on a task-specific dataset to adapt and improve its performance for that task

Facet (Facet layer): A technique that splits data into subsets and draws the same plot for each subset in a grid or wrap layout, enabling comparison across levels of a categorical variable

Guardrails: Explicit scope limits and safety constraints (policy rules, filters, or refusal behaviors) that prevent agents from attempting tasks outside their intended authority or producing unsafe outputs

Human-in-the-Loop: A deployment pattern that keeps humans involved for oversight, approval, or intervention in high-risk or ambiguous situations to ensure safety, compliance, and quality

Large Language Model (LLM): A neural network trained on large text corpora that generates or interprets natural language and is typically used by agents to reason, plan, and produce responses

Model Context Protocol (MCP): An open standard developed to provide a uniform protocol for connecting AI hosts with data and tool servers, enabling dynamic tool discovery, standardized authentication, and adaptable integrations

Modularity: The design principle of building systems as independent, interchangeable components (e.g., UI, agents, storage, tools) so parts can be updated or replaced with minimal impact on others

Multi-Agent System: A design where multiple specialized agents collaborate or coordinate, each handling a defined domain or set of tools, to solve complex or multi-step problems more robustly than a single agent

Network (swarm) Architecture: A decentralized multi-agent pattern in which agents independently process inputs and hand off tasks to peers as needed, enabling flexible collaboration without a central controller

Observability: The practice of logging, monitoring, and tracking system metrics (such as success rates, latencies, and errors) and user interactions to detect, diagnose, and improve agent behavior

Orchestration Layer: The software component that coordinates model calls, tool invocation, memory, logging, and multi-step workflows to manage the agent's end-to-end task execution

Prompt (system/user): The combined input messages given to an LLM—system prompts guide behavior and constraints, while user prompts contain the task or query—used to shape model outputs

Retry with Backoff (Backoff Strategy): A fault-tolerance pattern that retries failed external calls (e.g., API requests) with gradually increasing delays to avoid overwhelming services and improve the chance of eventual success

Shadow Mode: A testing technique where an application processes real user inputs and logs outputs for review but does not expose those outputs to users, enabling safe validation on live traffic

Supervisor Architecture: A hierarchical multi-agent pattern where a supervisor agent receives user input, delegates subtasks to worker agents, aggregates results, and is solely responsible for final responses

Tool (in agent context): A callable external function, API, or service that an agent can invoke to perform actions or retrieve data, such as databases, web APIs, code execution, or system commands

Vector Store: A specialized data store that saves high-dimensional vector embeddings (representations of text, images, etc.) to support fast semantic retrieval during agent reasoning and tool use