ARTIFICIAL INTELLIGENCE | ADVANCED | 5 DAYS

Agentic AI Bootcamp: Building Production Multi-Agent Systems

A five-day intensive on agentic systems that survive production. Covers agent design patterns, tool and memory architecture, MCP and A2A protocols, orchestration topologies, agent evaluation, guardrails and human oversight, agent security and enterprise governance. Assumes prior LLM application experience or completion of the AI Engineering bootcamp.

Why Agentic AI Bootcamp?

Built on production failure modes

Structured around what actually breaks at scale, not around what demos well.

Memory gets its own day

Context inconsistency is the main cause of multi-agent failure, so it is treated seriously.

Protocols as first-class content

MCP and A2A taught properly as the two-layer backbone of interoperable agent systems.

Governance included, not bolted on

Identity, audit and oversight designed in, because that is what blocks enterprise rollout.

What you'll be able to do

- ✓ Judge when a task genuinely warrants an agent and when it does not
- ✓ Apply the core agentic design patterns and select between them deliberately
- ✓ Design tools and sandboxed execution that agents can use reliably
- ✓ Engineer context for long-running agents through compaction and delegation
- ✓ Architect working, episodic and shared memory with a persistent state layer
- ✓ Build MCP servers exposing internal tools, data and prompts
- ✓ Use A2A and related protocols for cross-vendor agent interoperability
- ✓ Compare LangGraph, CrewAI and vendor agent SDKs and choose on real criteria
- ✓ Design supervisor, peer-to-peer and hierarchical orchestration topologies
- ✓ Diagnose and prevent coordination failures, loops and runaway cost
- ✓ Evaluate agents on trajectory as well as final output
- ✓ Implement guardrails, oversight models and blast-radius containment
- ✓ Defend against prompt injection, tool poisoning and privilege escalation
- ✓ Establish agent identity, inventory, audit and cost accountability at scale

Who this is for

- AI and machine learning engineers building agent systems
- Backend engineers and solution architects

- Platform and security engineers supporting agent deployment
- Technical leads accountable for agents reaching production

Curriculum

01 What Makes a System Agentic

- The observe, reason, act and reflect loop and how it differs from a workflow
- Why agent loops broke before 2026 and what changed to make them hold
- The autonomy spectrum from suggestion through to unsupervised execution
- Choosing the lowest autonomy that solves the problem
- The cases where an agent is strictly worse than a script

02 Agent Design Patterns

- Reason-and-act loops and plan-then-execute designs
- Reflection and self-critique patterns, and their real limits
- Supervisor and worker delegation
- Evaluator and optimiser pairs for quality-sensitive work
- Routing, map-reduce and decomposition patterns
- **Hands-on:** implement the same task with two patterns and compare

03 Tool Design for Agents

- Designing tool schemas an agent invokes correctly first time
- Error surfaces: what a tool should return when it fails
- Idempotency, retries and side-effect safety
- Sandboxed code execution and isolation for untrusted output
- **Hands-on:** harden a badly designed tool until an agent uses it reliably

04 Context Engineering for Long-Running Agents

- Context window management as the dominant quality constraint
- Compaction, summarisation and lossy compression of history
- Subagents as context isolation rather than just parallelism
- Deciding what an agent needs to remember within a single run
- **Hands-on:** extend an agent's effective working session without quality collapse

05 Memory Architecture

- Working, episodic and semantic memory as distinct concerns
- Why agent memory is transient and what that forces you to build
- The shared context layer as persistent state across pipeline steps
- Retrieval into memory, and memory as a tool the agent calls
- Staleness, contradiction and pruning strategies
- **Hands-on:** add a persistent memory layer to a multi-step agent

06 Agentic Retrieval

- Retrieval as a tool the agent decides to use rather than a fixed pipeline stage
- Query planning, decomposition and iterative retrieval
- Knowing when to stop searching and answer
- Grounding and citation in an agentic flow
- **Hands-on:** convert a static RAG pipeline into an agentic one and compare

07 MCP: The Tool and Data Layer

- MCP primitives: tools, resources and prompts
- Building a server that exposes internal capability cleanly
- Transports, clients and deployment options
- Registry, discovery and governing which servers are approved
- **Hands-on:** build and register an MCP server against an internal API

08 A2A and Agent Interoperability

- How agent-to-agent communication differs from tool access
- Agent discovery, capability advertisement and task delegation
- Transport, authentication and role-based access in cross-agent calls
- The connectivity explosion problem as agent counts grow
- Interface protocols for streaming agent state to a user interface
- **Hands-on:** expose an agent over a delegation protocol and call it from another

09 The Framework Landscape

- Graph-based orchestration and why an explicit state machine prevents runaway loops
- Role-based multi-agent frameworks and where they fit
- Vendor agent SDKs and the lock-in trade-off
- Building without a framework, and when that is the right call
- **Hands-on:** implement one system in two frameworks and assess the difference

10 Orchestration Topologies

- Supervisor and worker: central routing, easiest debugging, bottleneck risk
- Peer-to-peer coordination and shared protocols of conduct
- Hierarchical orchestration for deep task trees
- The centralised control plane as the dominant enterprise pattern
- Single points of failure and multi-region considerations
- Choosing a topology from task structure rather than preference

11 Coordination Failure Modes

- Context inconsistency as the primary cause of production failure
- Deciding which agent holds authority over shared definitions
- Conflict resolution between specialist agents
- Loop detection, maximum depth, token budgets and timeouts
- Cost explosions and how they usually happen
- **Hands-on:** break a working multi-agent system deliberately, then fix it

12 Agent Evaluation

- Evaluating the trajectory, not only the final answer
- Rubric grading for multi-step reasoning and tool selection
- Simulation and scripted environments for repeatable testing
- Regression suites for agent and prompt changes
- Measuring cost and latency per completed task as quality dimensions
- **Hands-on:** build a trajectory evaluation suite

13 Guardrails and Human Oversight

- Human-in-the-loop against human-on-the-loop supervision models
- Approval gates and where in a flow to place them
- Blast-radius containment and reversible-by-default design
- Interrupt, cancel and kill-switch mechanics for long-running agents
- Designing for the case where the agent is confidently wrong

14 Agent Security

- Direct and indirect prompt injection through tools, documents and web content
- Tool poisoning and untrusted connector risk
- Excessive permissions and privilege escalation paths
- Data exfiltration routes and how to close them
- Threat modelling an agent before it ships
- **Hands-on:** attack a running agent, then harden it

15 Governance, Identity and Observability at Scale

- Agent identity and applying existing access policy to non-human actors
- Inventory and registry: knowing what agents exist and who owns them
- Delegation tracing: who did what, at what cost, through which chain
- Audit trails and decision traceability for regulated environments
- Ownership, review cadence and safely retiring an agent

16 Capstone Build

- Design and build a multi-agent system against a real business process
- Wire tools through MCP and delegation through an agent protocol
- Attach memory, trajectory evaluation and guardrails
- Threat model the system and demonstrate the mitigations
- Present the architecture, failure analysis and governance model