Enterprise AI Agent Development with Azure AI Foundry, Microsoft Agent Framework, Python, MCP & RAG

Training Objective

The objective of this training is to equip technical leaders, architects, and senior developers with the practical knowledge required to design, develop, integrate, secure, deploy, and govern enterprise AI Agents.

The training focuses on Microsoft Agent Framework, Azure AI Foundry, Azure OpenAI, Python, FastAPI, Model Context Protocol, Retrieval-Augmented Generation, contextual memory, enterprise workflows, security, deployment, and observability.

The course combines instructor-led explanations, live demonstrations, guided hands-on exercises, and an integrated enterprise capstone project.

Target Audience

- Solution Architects
- Technical Leads
- Engineering Managers
- Senior Developers
- AI Platform Owners
- AI and Integration Architects

Duration: 4 Days (32 Hours)

Course Curriculum

Day 1 – Module 1 – Foundations of Enterprise AI Agents

- Evolution from chatbots and AI assistants to autonomous AI Agents
- Core Agent components: model, instructions, tools, knowledge, context, memory, and orchestration
- Deterministic applications versus AI Agents and controlled workflows
- Single-agent, agent-as-tool, and multi-agent architecture patterns
- Enterprise AI use cases, adoption considerations, and value identification
- Responsible AI, transparency, reliability, and human oversight principles

Day 1 – Module 2 – Azure AI Foundry Fundamentals

- Azure AI Foundry projects, models, deployments, connections, and environments
- Azure OpenAI and Foundry model endpoints
- Direct model inference versus service-managed Foundry Agents
- Model selection, configuration, token usage, latency, and cost considerations
- Authentication using API keys, Azure credentials, and Managed Identity
- Demonstration of Foundry project configuration and model deployment

Day 1 – Module 3 – Microsoft Agent Framework Fundamentals

- Microsoft Agent Framework architecture and core capabilities
- Agents, Agent Harnesses, and Workflows
- Chat clients, Agent sessions, tools, Context Providers, and middleware
- FoundryChatClient and FoundryAgent implementation patterns
- Agent runtime, tool-calling loop, streaming, and structured responses
- Selecting between an Agent, Workflow, Skill, Harness, or deterministic function

Day 1 – Module 4 – Building AI Agents with Python and Foundry SDK

- Configuring the Python environment and Microsoft Agent Framework packages
- Creating an Agent with name, instructions, description, model client, and behaviour
- Implementing streaming and non-streaming Agent responses
- Generating structured output using Pydantic and JSON Schema
- Managing multi-turn conversations using AgentSession
- Hands-on exercise to build and test a Foundry-connected enterprise Agent

Day 2 – Module 5 – Prompt Engineering and Agent Instructions with Foundry Workflows

- Designing system instructions and defining Agent responsibilities
- Establishing business rules, task boundaries, and response limitations
- Creating structured and consistent response formats
- Grounding Agent responses in provided enterprise context
- Applying guardrails and reducing unsupported or hallucinated responses
- Demonstration and exercise to improve Agent instructions through iterative testing

Day 2 – Module 6 – Python and FastAPI for Agent Applications

- Recommended Python project structure for enterprise AI applications
- Asynchronous programming and configuration management
- FastAPI fundamentals for exposing AI Agent services
- Request validation, response models, exception handling, and logging
- Separation of API, Agent, tool, retrieval, and security layers
- Hands-on exercise to expose an Agent through a FastAPI endpoint

Day 2 – Module 7 – Enterprise Tools and Application Integration

- Creating typed function tools using Python functions and annotations
- Automatic tool selection, execution, validation, and structured results
- Integration with REST APIs, relational databases, and enterprise services
- Tool error handling, retries, timeouts, and fallback responses
- Tool approval and access control for sensitive operations
- Hands-on exercise to build API and database tools for an Agent

Day 2 – Module 8 – Model Context Protocol

- MCP architecture, clients, servers, tools, resources, and prompts
- Local and remote MCP servers using STDIO and Streamable HTTP
- Building reusable enterprise functions as MCP tools
- Connecting Microsoft Agent Framework Agents to MCP servers
- MCP authentication, authorization, approval, trust, and versioning
- Demonstration and hands-on exercise to build and consume an MCP tool

Day 3 – Module 9 – Retrieval-Augmented Generation

- Enterprise RAG architecture, document ingestion, chunking, and embeddings
- Keyword, vector, semantic, and hybrid search techniques
- Azure AI Search integration with Microsoft Agent Framework
- Metadata filtering, document-level security, and contextual retrieval
- Grounded response generation, source attribution, and citations
- Hands-on exercise to build and test an Azure AI Search-based RAG Agent

Day 3 – Module 10 – Contextual Memory and Conversation Management

- Stateless versus stateful Agent execution
- AgentSession and conversation-history management
- Context Providers for user, tenant, role, and business information
- Short-term context versus long-term memory
- Conversation summarization, context compaction, and token management
- Hands-on exercise to add user-specific and business-specific context

Day 3 – Module 11 – Agent Framework Middleware

- Agent, chat-client, and function middleware concepts
- Logging, diagnostics, and correlation information
- Authentication, authorization, and tenant-context injection
- Input validation, output validation, and content filtering
- Exception handling, rate limiting, and runtime policy enforcement
- Demonstration of reusable middleware for security and observability

Day 3 – Module 12 – Agent Framework Workflows

- Difference between dynamic Agent execution and controlled workflows
- WorkflowBuilder, executors, edges, events, and message routing
- Sequential, conditional, and parallel workflow patterns
- Integration of Agents, Python functions, APIs, and MCP tools
- Human-in-the-loop approval, checkpointing, interruption, and resumption
- Hands-on exercise to build a controlled approval-based workflow

Day 4 – Module 13 – Multi-Agent Architecture Patterns

- Selecting single-agent versus multi-agent architecture
- Agent-as-tool and specialist Agent patterns
- Router, supervisor, reviewer, and aggregator Agent roles
- Sequential and concurrent Agent coordination
- Demonstration of a specialist Agent workflow with controlled routing

Day 4 – Module 14 – Enterprise Security, Governance and Compliance

- Microsoft Entra ID, OAuth, OpenID Connect, and Managed Identity
- Role-based access control for users, tools, data, and enterprise knowledge
- Azure Key Vault and secure credential management
- Prompt injection, indirect prompt injection, data exfiltration, and tool poisoning
- MCP trust boundaries, human approvals, and audit requirements

Day 4 – Module 15 – Deployment and Enterprise Integration

- Managed Foundry Agents versus self-hosted Agent applications
- FastAPI, Docker, Azure App Service, and Azure Container Apps
- Environment configuration, Managed Identity, and secret management
- CI/CD, application versioning, rollback, scalability, and cost considerations
- Integration with web, mobile, .NET, Angular, and collaboration applications
- Demonstration of containerizing and deploying an Agent application

Day 4 – Module 16 – Observability and Evaluation

- OpenTelemetry support in Microsoft Agent Framework
- Model-call, tool-call, MCP-call, and workflow tracing
- Logging, correlation IDs, Azure Monitor, and Application Insights
- Monitoring token usage, latency, failures, tool behaviour, and cost
- Evaluating groundedness, relevance, completeness, safety, and tool correctness
- Demonstration of Agent traces and diagnostics using observability tools