training code: AI-500 / ENG DL 4d / EN

Designing and Implementing Multi-Agent AI Solutions

The AI-500: Designing and Implementing multi-agent AI Solutions training is an advanced, expert-level course designed for practitioners who architect and develop scalable, production-ready multi-agent AI systems and workflows

Training recipients

- Expert-level practitioners who have subject matter expertise in designing, building, and optimizing scalable, production-ready, multi-agent AI systems, solutions, and workflows
- Professionals who manage the development from design to production and work closely with developers, machine learning engineers, platform engineers, data scientists, and business stakeholders
- Architects and engineers with experience developing AI and machine learning solutions, deploying agentic systems in production environments, and orchestrating agent logic using Microsoft Foundry

Benefits

- Multi-agent architecture mastery – You'll learn to design logical architectures for multi-agent solutions, distinguishing between agentic AI and multi-agent AI architectures, and implementing advanced orchestration patterns including hub-and-spoke and parallel agent spawning.
- Production-ready implementation skills – You'll gain hands-on experience building and integrating tool ecosystems with MCP, implementing stateful agentic loops with Microsoft Foundry agent service, and designing enterprise-scale agent communication with A2A in Azure.
- Advanced orchestration expertise – You'll master task decomposition and agent collaboration strategies, implement dynamic adaptive task decomposition, design agent handoff message schemas, and ensure handoff reliability and context preservation.
- Security and governance proficiency – You'll learn to secure multi-agent systems with Azure zero-trust

architecture, apply zero-trust identity to agent networks, implement JIT and workload identity access controls, and configure compliance controls for regulated deployments.

- Enterprise-scale optimization – You'll develop skills in designing distributed observability with OpenTelemetry, implementing evaluation frameworks with Microsoft Foundry, optimizing multi-agent performance and cost, and establishing audit and accountability frameworks.
- Real-world application through scenarios – You'll work with four distinct fictional customer scenarios spanning different industries, providing practical context for implementing multi-agent solutions in diverse organizational settings.

Training program

1. Design stateful agentic loops with Microsoft Foundry agent service
   - Introduction
   - Examine production agentic loop architecture
   - Examine the Foundry responses API and agents v2 model
   - Implement agent reflection and planning cycles
   - Design session state and context management
   - Implement fork-based sessions and conversation resumption
   - Migrate stateful agentic loops from agents v1 to agents v2
2. Implement advanced multi-agent orchestration patterns in Microsoft Foundry
   - Introduction
   - Differentiate agentic AI from multi-agent AI architectures
   - Examine advanced orchestration architectures
   - Implement hub-and-spoke orchestration
   - Design parallel agent spawning and synchronization
   - Compare orchestration frameworks
3. Apply task decomposition and agent collaboration strategies in Microsoft Foundry
   - Introduction
   - Design prompt chaining workflows
   - Implement dynamic adaptive task decomposition
   - Design agent handoff message schemas
   - Ensure handoff reliability and context preservation
   - Optimize decomposition granularity
4. Design enterprise-scale agent communication with A2A in Azure
   - Introduction
   - Design A2A agent ecosystems at scale
   - Implement distributed shared state management
   - Design context isolation and sharing strategies
   - Build conflict detection and resolution mechanisms

- Resolve conflicts and maintain audit trails

5. Design advanced prompting strategies for production AI agents
   - Introduction
   - Design multiturn reasoning prompt architectures
   - Implement prompt injection defenses
   - Build system prompt frameworks for agent control
   - Design multi-intervention guardrail architectures
   - Implement prompt versioning and optimization
   - Automate prompt regression and optimization
   - Design fine-tuning strategy and data pipelines
6. Build enterprise-grade tool ecosystems with MCP and Microsoft Foundry
   - Introduction
   - Design production MCP server architecture
   - Build MCP servers with error handling and fallback
   - Implement tool selection and routing logic
   - Govern tool dependencies and versioning
7. Implement advanced RAG pipelines with Azure AI Search and Microsoft Foundry
   - Introduction
   - Design hybrid search architectures
   - Implement reranking and context ranking
   - Design dynamic knowledge source routing
   - Optimize chunking and embedding strategies
8. Design multi-agent memory architectures with Azure Cosmos DB
   - Introduction
   - Examine memory architecture patterns
   - Implement semantic memory with vector storage
   - Optimize memory retrieval and context injection
   - Configure context window optimization
   - Design memory retention and consolidation
   - Enforce memory privacy and audit compliance
9. Implement CI/CD pipelines for multi-agent systems with GitHub Actions
   - Introduction
   - Design multi-agent deployment pipelines
   - Implement progressive deployment strategies
   - Configure multi-environment agent deployment strategies
   - Automate rollback procedures
10. Secure multi-agent systems with Azure zero-trust architecture
    - Introduction
    - Apply zero-trust identity to agent networks
    - Secure agent access with JIT and workload identity
    - Design authentication flows and secrets lifecycle

- Prevent lateral movement in agent networks
- Implement tenant context propagation and data isolation
- Validate tenant boundaries and enforce encryption
- Configure compliance controls for regulated agent deployments

11. Scale responsible AI governance with Azure AI Content Safety and Microsoft Foundry
    - Introduction
    - Design fairness and bias monitoring
    - Implement transparency and explainability
    - Configure privacy protection in multi-agent workflows
    - Establish audit and accountability frameworks
12. Govern the enterprise agent lifecycle in Microsoft Foundry
    - Introduction
    - Design agent versioning and approval workflows
    - Implement usage quotas and rate limiting
    - Design cost allocation and chargeback models
    - Establish agent retirement and deprecation processes
13. Implement distributed observability for multi-agent solutions with OpenTelemetry
    - Introduction
    - Design distributed tracing for multi-agent solutions
    - Implement structured logging for agent decisions
    - Configure telemetry aggregation and dashboards
    - Build anomaly detection for agent behavior
14. Design evaluation frameworks for multi-agent solutions with Microsoft Foundry
    - Introduction
    - Define multi-agent success metrics
    - Implement LLM-as-judge evaluation for multi-agent systems
    - Design synthetic test datasets for multi-agent evaluation
    - Build regression testing pipelines to detect agent drift
15. Optimize multi-agent performance and cost in Microsoft Foundry
    - Introduction
    - Design model routing for agent ecosystems
    - Implement multi-level caching strategies
    - Optimize token usage and context management
    - Balance quality, cost, and latency tradeoffs
16. Design human-in-the-loop approval workflows with Power Automate and Microsoft Teams
    - Introduction
    - Design confidence-based escalation for human intervention
    - Implement approval workflows for agent-initiated actions
    - Build active learning from human feedback
    - Configure audit workflows for regulated decisions
17. Debug and respond to production multi-agent incidents in Azure

- Introduction
- Implement agent replay for production debugging
- Design root cause analysis for agent failures
- Configure automated incident detection and remediation
- Establish incident response and post-mortem processes

Expected preparation of the participant

- Experience developing AI and machine learning solutions
- Experience deploying agentic systems in production environments
- Experience orchestrating agent logic using Microsoft Foundry
- Proficiency in Python programming
- Knowledge of basic AI concepts and services on Azure platform
- Experience with Azure compute, network, storage, and data services (recommended)

To increase the comfort of work and training's effectiveness we suggest using an additional monitor. The lack of additional monitor does not exclude participation in the training, however, it significantly influences the comfort of work during classes.
You can find information and requirements of participation in Distance Learning trainings at:
https://www.altkomakademia.pl/en/how-it-works/about-courses/

Training Includes

* electronic handbook available at:https://learn.microsoft.com/pl-pl/training/
* access to Altkom Akademia student portal

Duration

4 days / 28 hours

Language

- Training: English
- Materials: English