Designing & Implementing Agents and Pro Code Copilots using Microsoft Agent Framework and Azure AI Agent Service (AZAGENTS)

ID AZAGENTS **Preis** CHF 2'490.– (exkl. MwSt.) **Dauer** 4 Tage

Zielgruppe

- Software Architects & Engineers for Agentic AI Solutions
- Microsoft 365 & AI Pro-Code Developers

Voraussetzungen

- Python, C#, Typescript
- GitHub Account
- Microsoft 365 Development & Azure Development Skills helpful

Kursinhalt

Module 1: Copilot & Agent Extensibility Fundamentals

Microsoft Foundry Essentials

- Copilots & Agent Frameworks in the Microsoft Ecosystem
- Deploying LLMs in Microsoft Foundry
- Model Router: Smart Model Selection
- Microsoft Foundry SDK & Microsoft.Extensions.AI
- Infrastructure as Code (IaC) using Azure Developer CLI Agentic Mode

Agentic AI Fundamentals

- What Are Agents & Agentic AI?
- Prompt Engineering vs Context Engineering
- Knowledge Integration & Agentic RAG
- Deep Reasoning & (Reasoning and Acting)
- Function Calling, REST APIs & MCP Servers
- Evaluating Generative AI Performance
- Governance & Guardrails for Responsible Agents

Implementing Model Context Protocol Servers (MCP)

- MCP Core Concepts & Architecture
- Transports: STDIO vs HTTP Streaming
- Debugging with MCP Inspector
- Authentication & Security Best Practices
- Hosting MCP's in Azure Functions
- Implementing MCP Apps

Module 2: Build Agents using Foundry Agent Service

- Introduction to Foundry Agent Service
- Conversations, Runs & State Management
- Knowledge Integration: Foundry IQ, File Search, Azure AI Search, Agent Memory & Bing Grounding
- Executing Actions with Tools: Code Interpreter, Azure Functions, OpenAPI, MCP & Deep Research
- Automating UI Tasks using Browser Automation and Computer Use
- Voice Agent Integration using Azure Speech Voice Live API
- Tracing, Observability & Performance Evaluation
- Hosted Agents: Containerized Deployments with Hosting Adapter & Agent Identity
- Agent-to-Agent Protocol (A2A) & Connected Agents

Module 3: Orchestrate Agents using Microsoft Agent Framework

Microsoft Agent Framework Basics & Concepts

- Introduction to the Agent Framework
- Chat Clients vs Agents: Key differences
- Agent types and configuration essentials
- Integrating Microsoft Foundry agents
- Threads, Conversation management & persistence
- Implementing long-term memory
- Governance, Middleware & Observability
- Hosting Agents in ASP.NET Core & Python (AddAIAgent, Responses API)

Agent Skills & Knowledge

- Agent Skills: SKILL.md structure, SkillsProvider & Progressive Disclosure
- Code-defined Skills vs. File-based Skills
- Agent Skills vs. Workflows: When to use each

- Built-in tools: Code Interpreter, File Search, Bing Grounding
- Adding custom tools and calling them from agents
- Integrating OpenAPI and MCP tools
- Function-calling middleware for advanced workflows

Orchestration, Durable & Hosted Agents

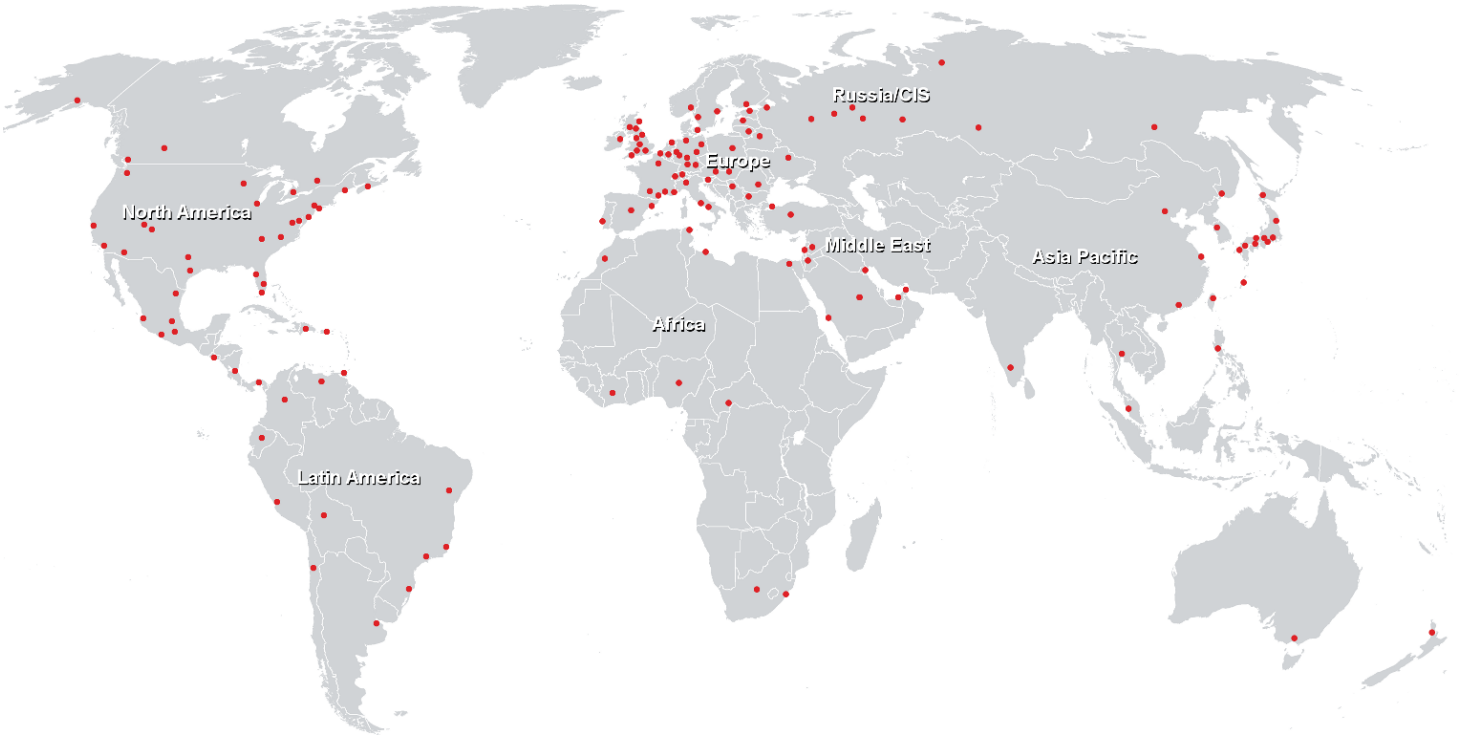
- Introduction to Multi-Agent Orchestration
- Orchestration Patterns (Sequential, Concurrent, Fan-out/Fan-in)
- Durable Agents with Azure Durable Functions (Flex Consumption, auto-scaling)
- Conversation State Persistence & Durable Task Scheduler
- Hosted Agents: Deploying to Azure Foundry (container images, agent identity)
- Branching, Checkpointing & Human-in-the-loop
- Observability & Workflow Visualization

Module 4: Agent Integration using Microsoft Agents SDK

- Overview Microsoft 365 Agents SDK (C#, JavaScript, Python)
- Connecting Copilot Studio & Microsoft Foundry Agents via A2A Protocol
- Orchestrate Multi-Agent Solutions using Microsoft Agent Framework
- Publishing Agentic AI Solutions to Copilot Chat and Teams
- Front-End Integration using Agent–User Interaction (AG-UI) Protocol

Designing & Implementing Agents and Pro Code Copilots using Microsoft Agent Framework and Azure AI Agent Service (AZAGENTS)

Weltweite Trainingscenter





Fast Lane Institute for Knowledge Transfer (Switzerland) AG

Husacherstrasse 3
CH-8304 Wallisellen
Tel. +41 44 832 50 80

info@flane.ch, https://www.flane.ch