#Milano

Rendere un MCP server enterprise-ready: autenticazione, Azure API Management e Microsoft 365 Copilot

Matteo Pagani

Cloud Solution Architect @ Microsoft

#GLOBALAZURE 2026

#Milano

improove

TD SYNNEX

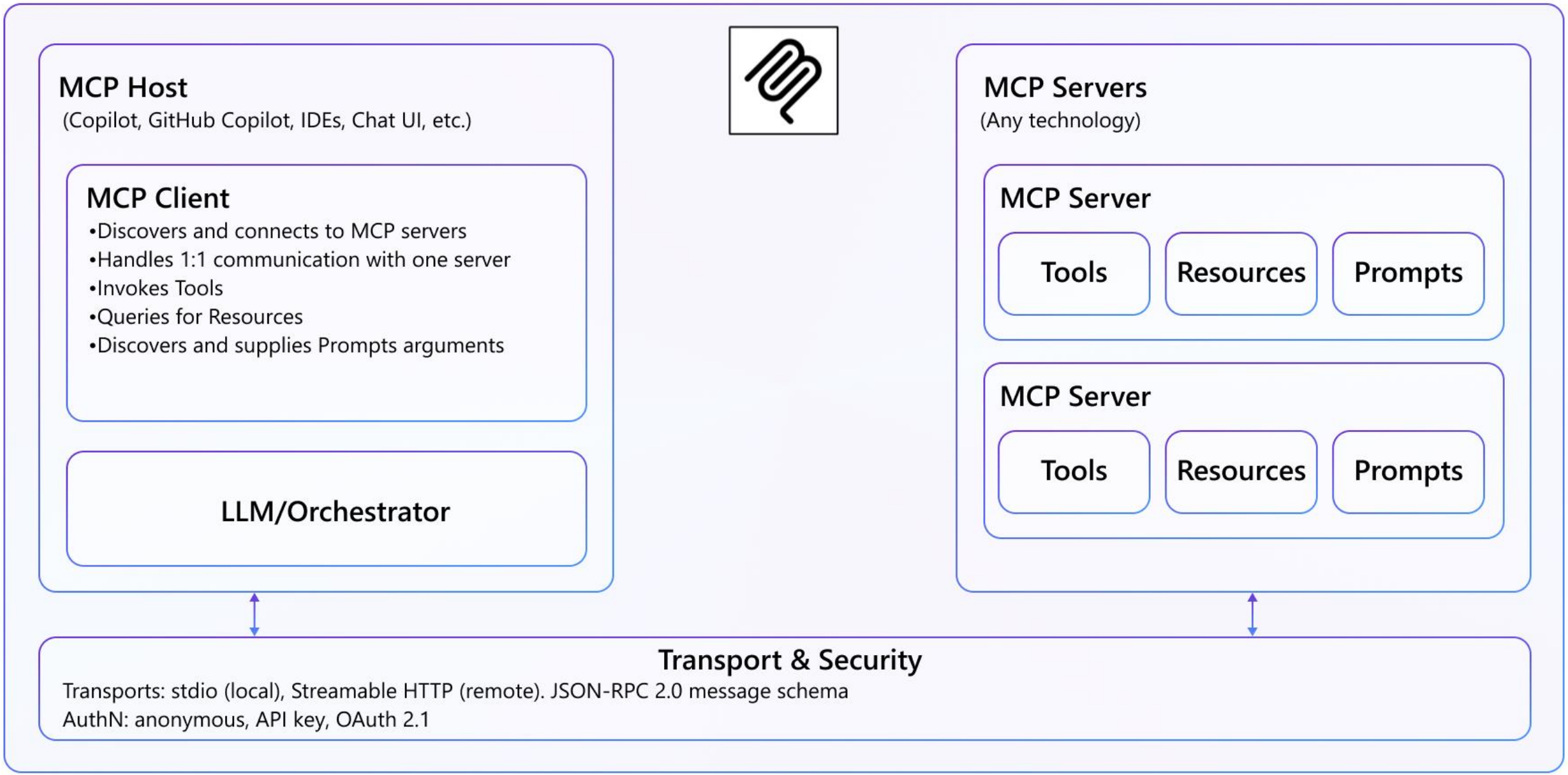
Grazie ai nostri sponsor 🙏

#GLOBALAZURE
2026
#Milano

Understanding Model Context Protocol

What is MCP?

- Model Context Protocol
  - Open-source standard for connecting AI applications to external systems
- MCP is like a “universal plug”
  - Allows AI models connecting to tools, apps, and data
  - Just like USB-C connects your phone to anything else
- Helps AI agents do more
  - Calling APIs, reading files, sending messages, etc.
  - Without needing custom code for each task
- Developers save time
  - Making AI solutions more powerful, flexible, and easier to maintain


MCP Host
(Copilot, GitHub Copilot, IDEs, Chat UI, etc.)
MCP Client
•Discovers and connects to MCP servers
•Handles 1:1 communication with one server
•Invokes Tools
•Queries for Resources
•Discovers and supplies Prompts arguments
LLM/Orchestrator
MCP Servers
(Any technology)
MCP Server
Tools
Resources
Prompts
MCP Server
Tools
Resources
Prompts
Transport & Security
Transports: stdio (local), Streamable HTTP (remote). JSON-RPC 2.0 message schema
AuthN: anonymous, API key, OAuth 2.1

Microsoft ❤️ MCP

MCP at Microsoft

MCP servers across Microsoft

- Azure
- Foundry Tools
- GitHub
- Azure Logic Apps
- Agent 365

Build your own and publish

- Logic Apps
- Azure API Center
- Azure Functions
- Azure App Service
- Azure Container Apps

Call from MCP clients anywhere

- Foundry agents
- Copilot Studio
- VS Code
- GitHub Copilot

Secure access and governance

- Azure API Management
- Entra ID

What we'll see today

Authentication

Protecting the MCP server using Microsoft Entra and the OAuth 2.0 protocol

Support for enterprise agentic platforms

Building agents connected to a protected MCP server with:

- Microsoft 365 Agents Toolkit
- Copilot Studio
- Microsoft Foundry

Building an MCP server

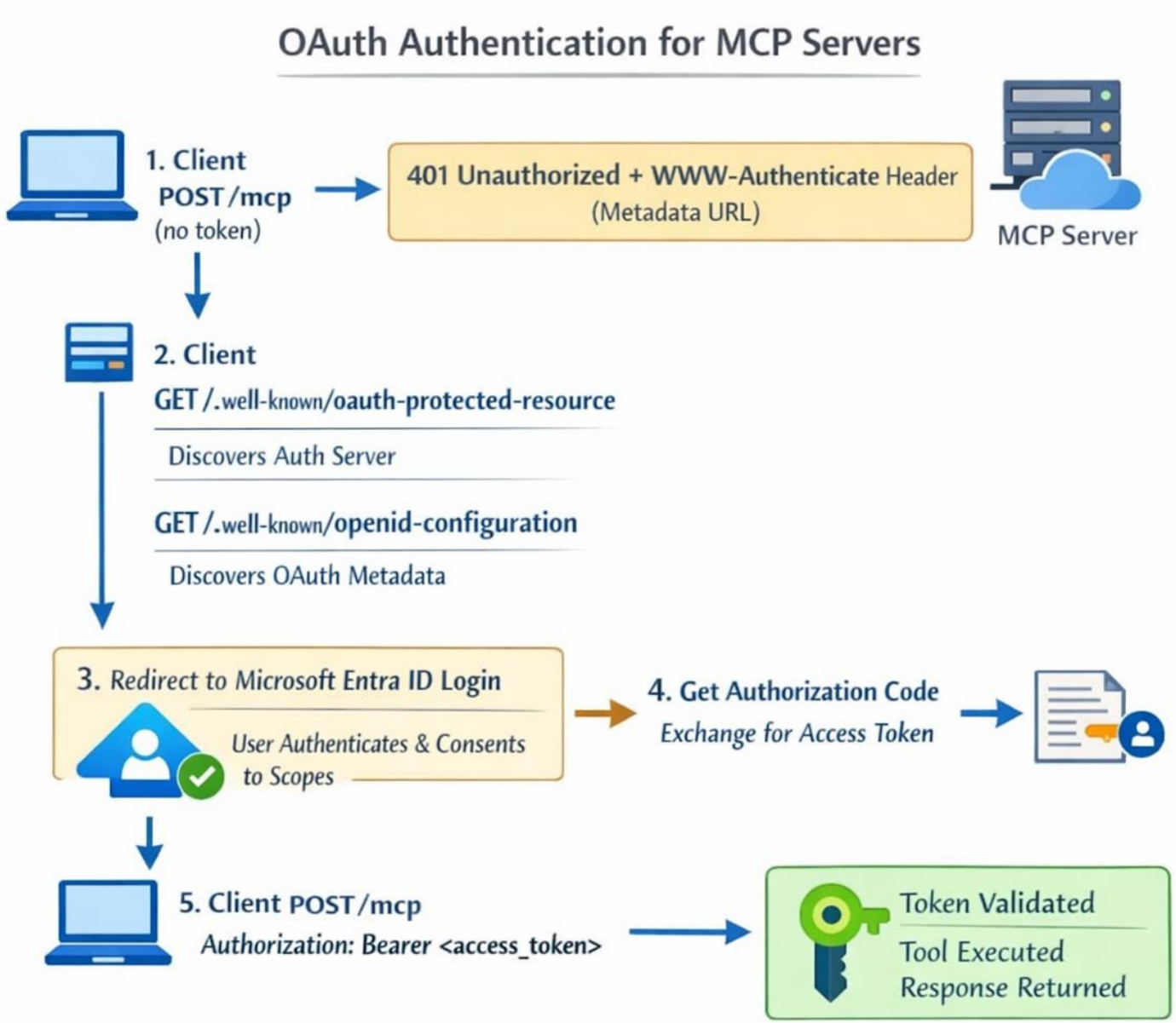
- If you already have an API, the hard part is already done ☺
- You expose your API endpoints as MCP tools
- Key difference with APIs: you must describe the purpose of the functions and the input parameters, so that the agent knows when and how to use it



An MCP server to manage flights

Demo

The MCP authorization chain



The MCP authorization chain

Demo

Three ways to build this infrastructure

Code

Most flexible, but more work and harder to maintain

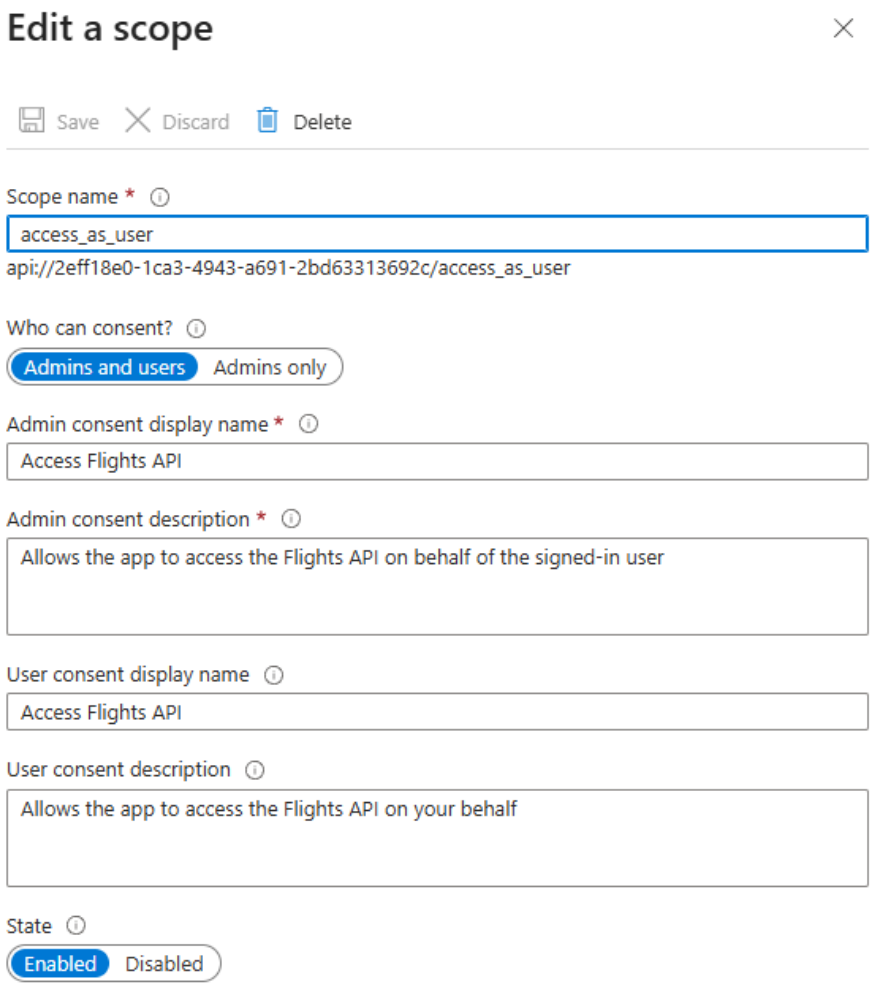
Azure App Service

Simple to implement, but least flexible option

API Management

In between the other options: more work to implement it than App Service, but more flexible

You need an app registration in Entra (server)



- It’s used only to protect the MCP server
- No need to configure secrets os redirect URI
- You must expose a scope to enable clients to access to the MCP server

Azure App Service

1. You deploy your MCP server (with no authentication code) to the App Service
2. You enable Easy Auth
3. You set an environment variable called **WEBSITE_AUTH_PRM_DEFAULT_WITH_SCOPES** with the scope exposed by the app registration used to protect the App Service (e.g. api://5643e255-1c40-4eb5-98fd-255148bad97b/user_impersonation)

Your app service will automatically expose the .well-known/oauth-protected-resource endpoint with the required information

Azure App Service and MCP

Demo

API Management

1. You deploy your MCP server (with no authentication code) to an App Service or Azure Container Apps
2. You restrict traffic so that only APIM can access to the service
3. You use the validate-azure-ad-token policy to restrict access to clients authenticated with Entra
4. If the authentication fails, you return a 401 and the WWW-Authenticate header with the .well-known/oauth-protected-resource URL
5. You manually implement the .well-known/oauth-protected-resource endpoint using policies

Two options for APIM

Expose MCP tools in code

You use APIM only to enforce the authentication layer

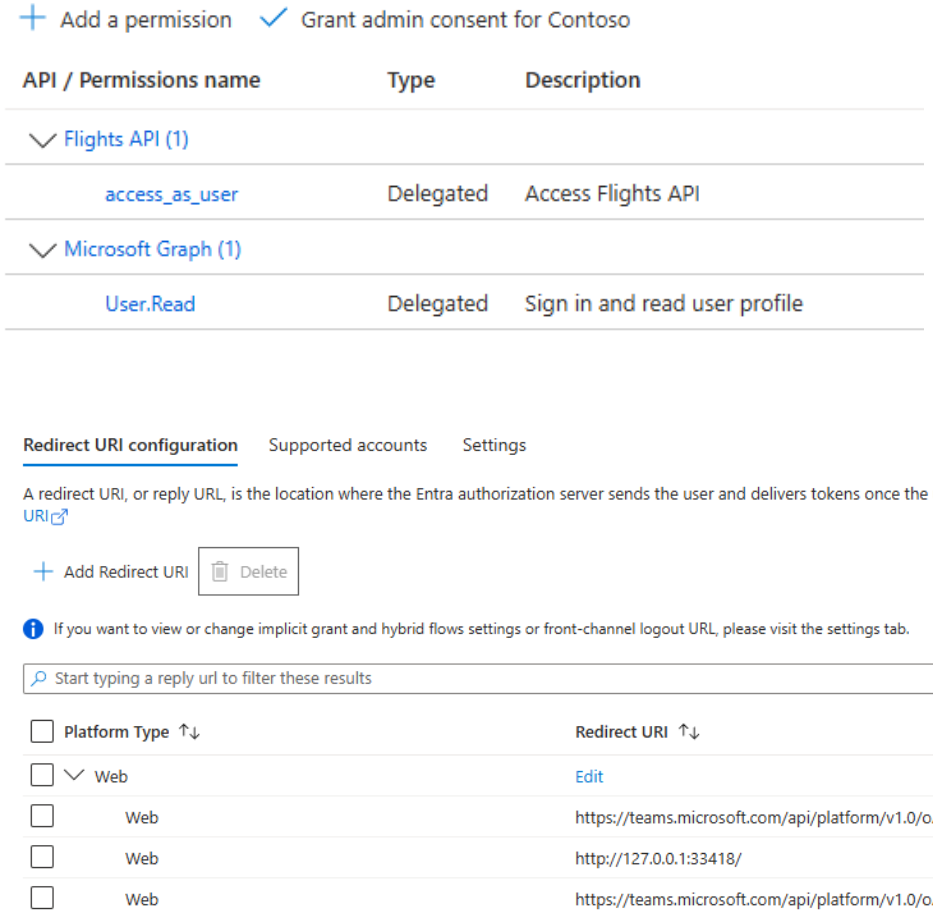
Turn APIs into MCP tools

You use APIM to turn your APIs into MCP tools without changing the code

API Management and MCP

Demo

You need an app registration in Entra (client)



- It’s used to authenticate against the MCP server
- It’s a proper OAuth registration
- You expose a client id and client secret
- You set up the redirect URIs for your calling platform
- You add permission for the scope you created for the MCP server

Consuming MCP from Declarative Agents

Empowering every developer to build agents

Build on the M365 Copilot Stack

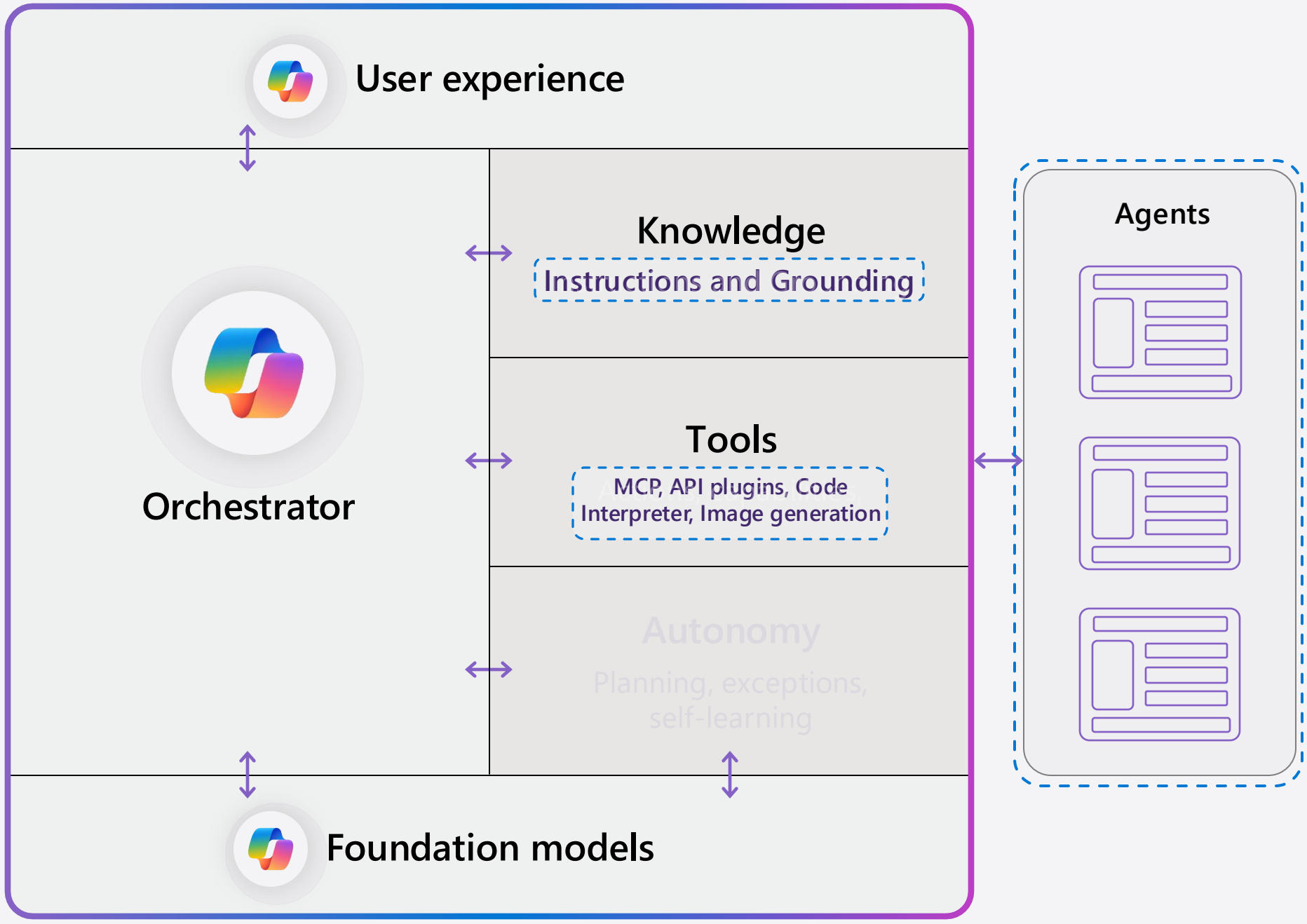
Build on your own AI Stack

Build for your own app

Declarative agents

Deliver powerful solutions on the Copilot AI stack.

Declarative agents



#GLOBALAZURE
2026
#Milano

Consuming MCP from Copilot Studio

Empowering every developer to build agents

Build on the M365 Copilot Stack

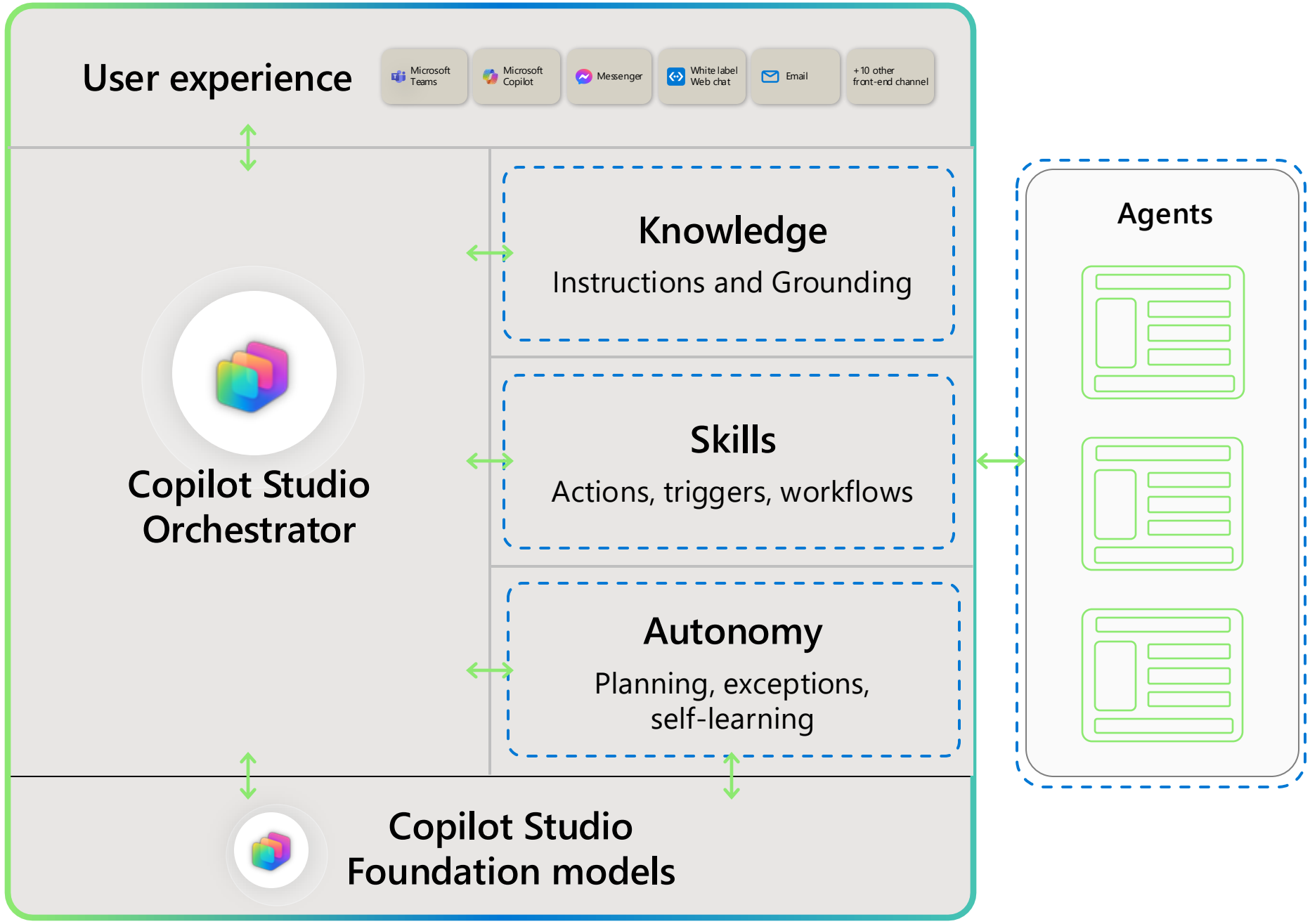
Build on your own AI Stack

Build for your own app

Custom Engine

Bring your own agent, custom models, orchestration and logic

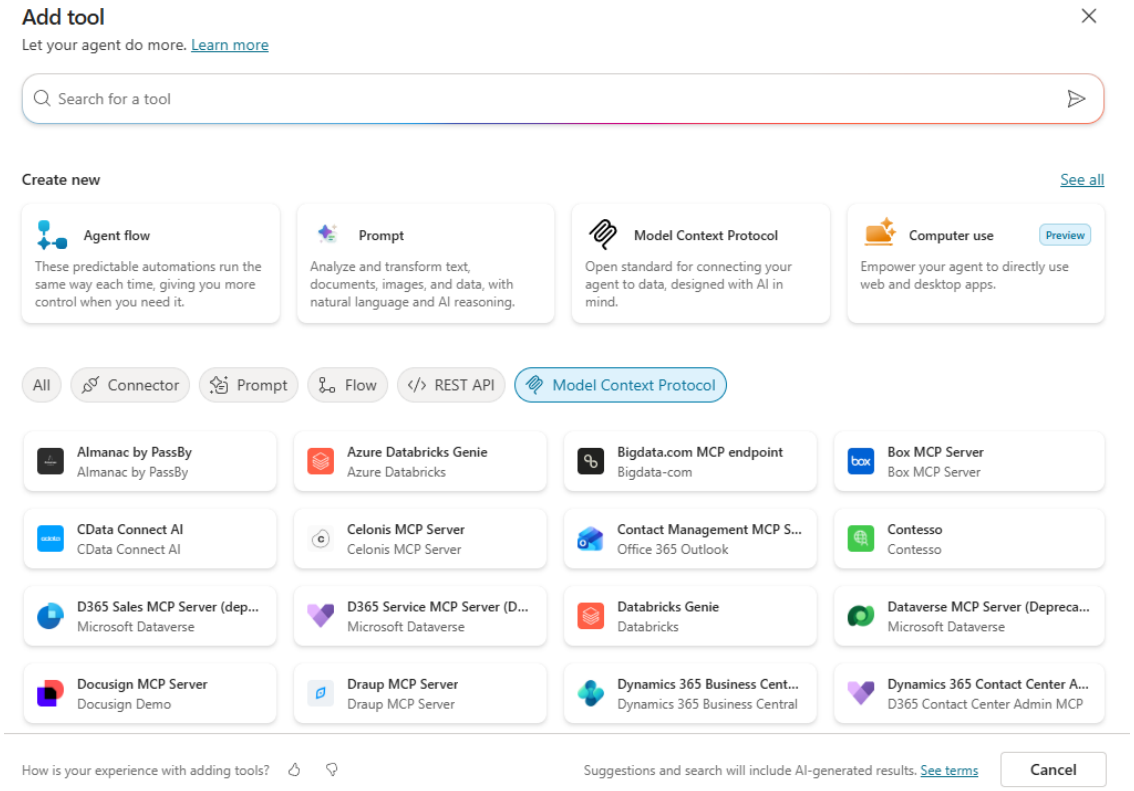
Copilot Studio Full agents

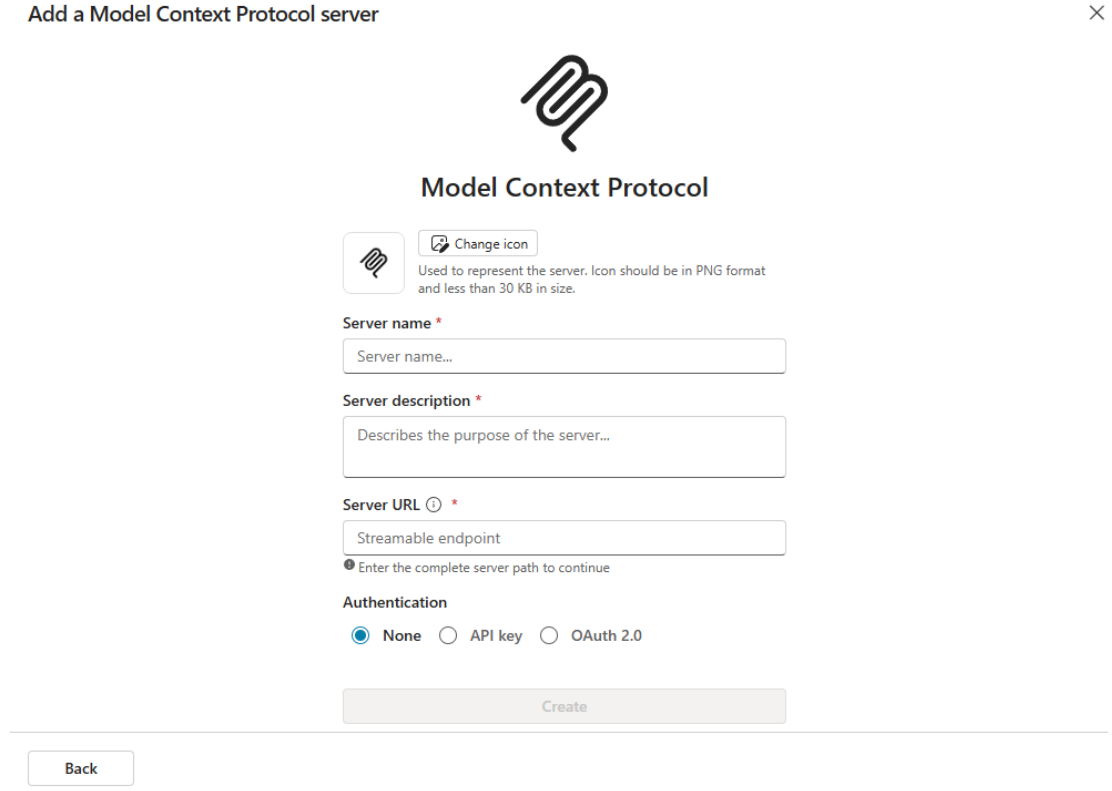


Microsoft provided

Maker designed

Adding an MCP server to a Copilot Studio agent





#GLOBALAZURE
2026
#Milano

Consuming MCP from Foundry Agent Service

Empowering every developer to build agents

Build on the M365 Copilot Stack

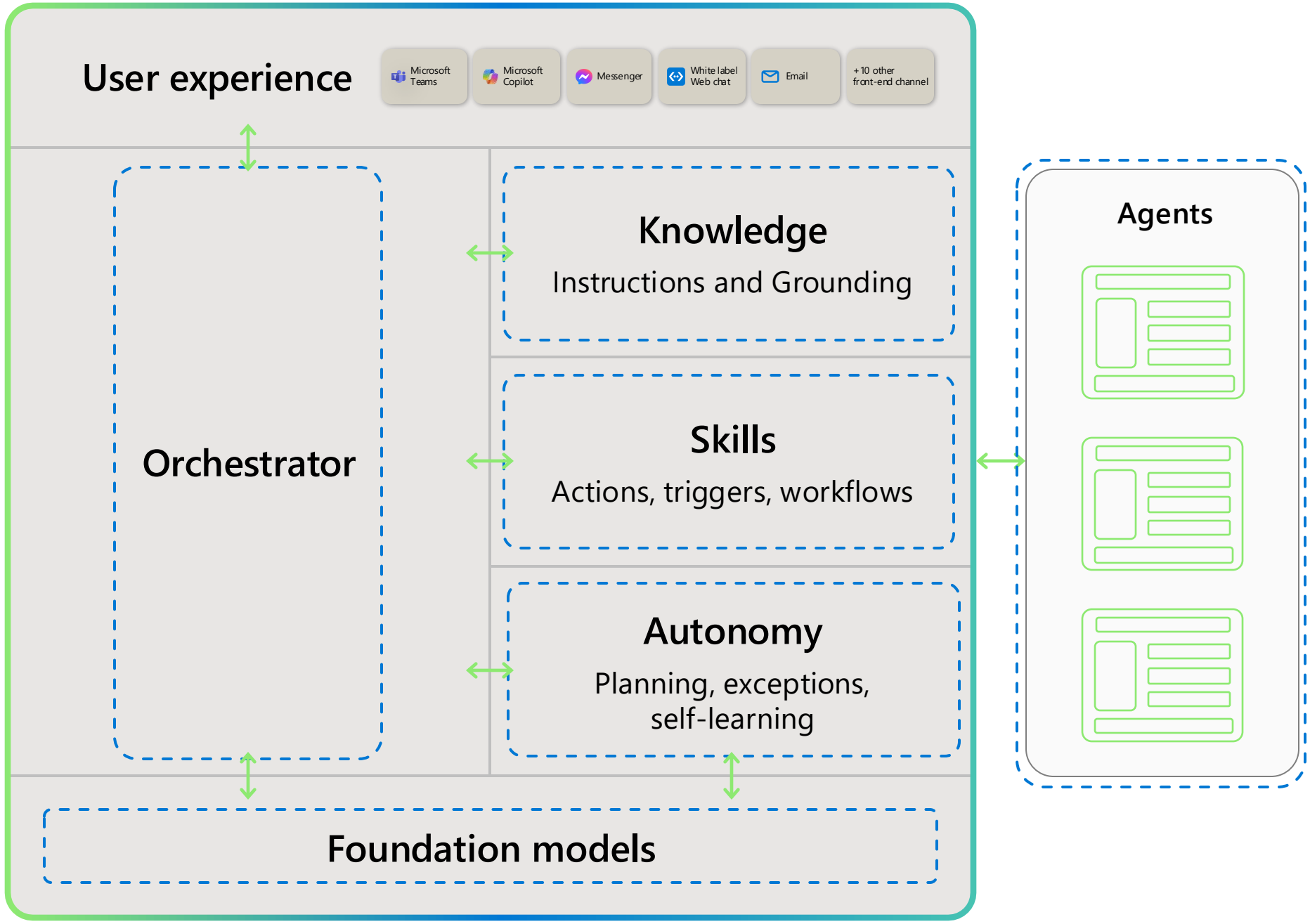
Build on your own AI Stack

Build for your own app

Custom Engine

Bring your own agent, custom models, orchestration and logic

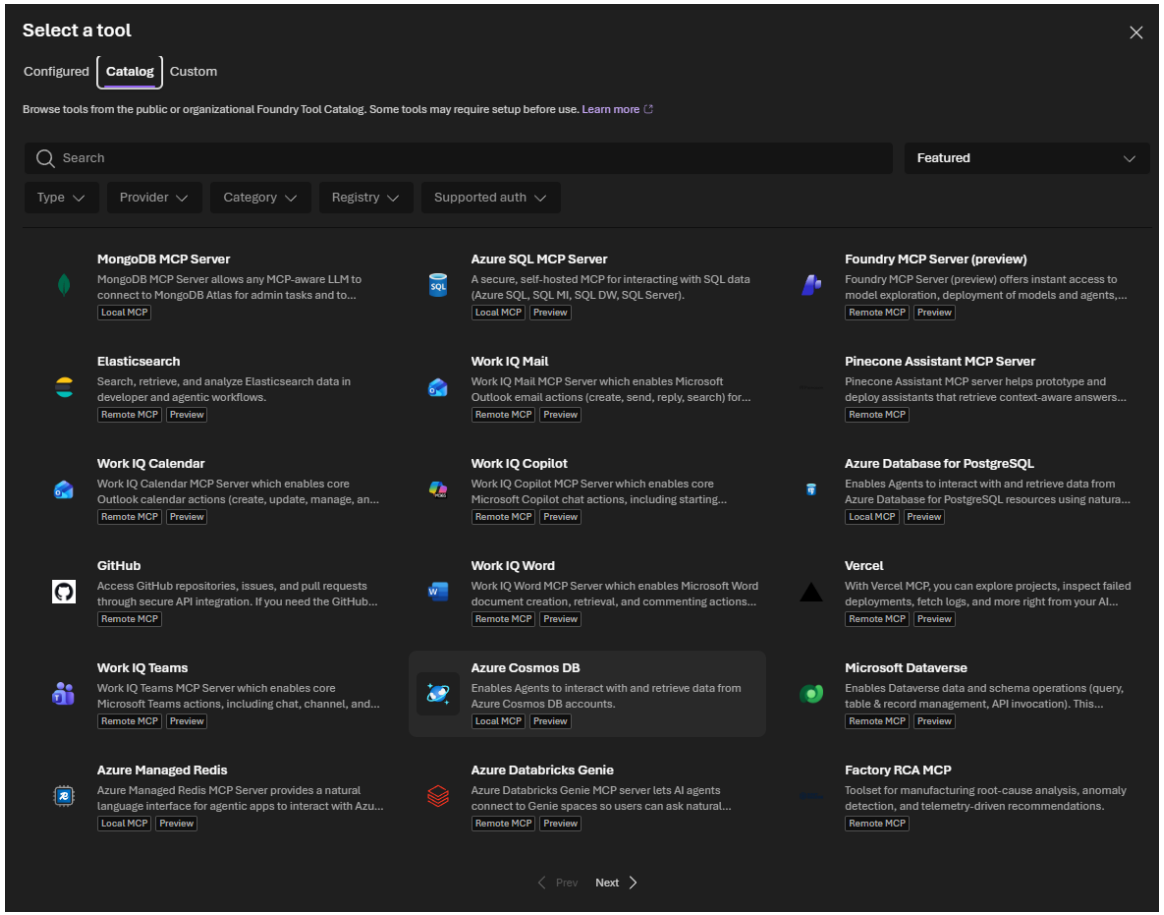
Custom engine agents

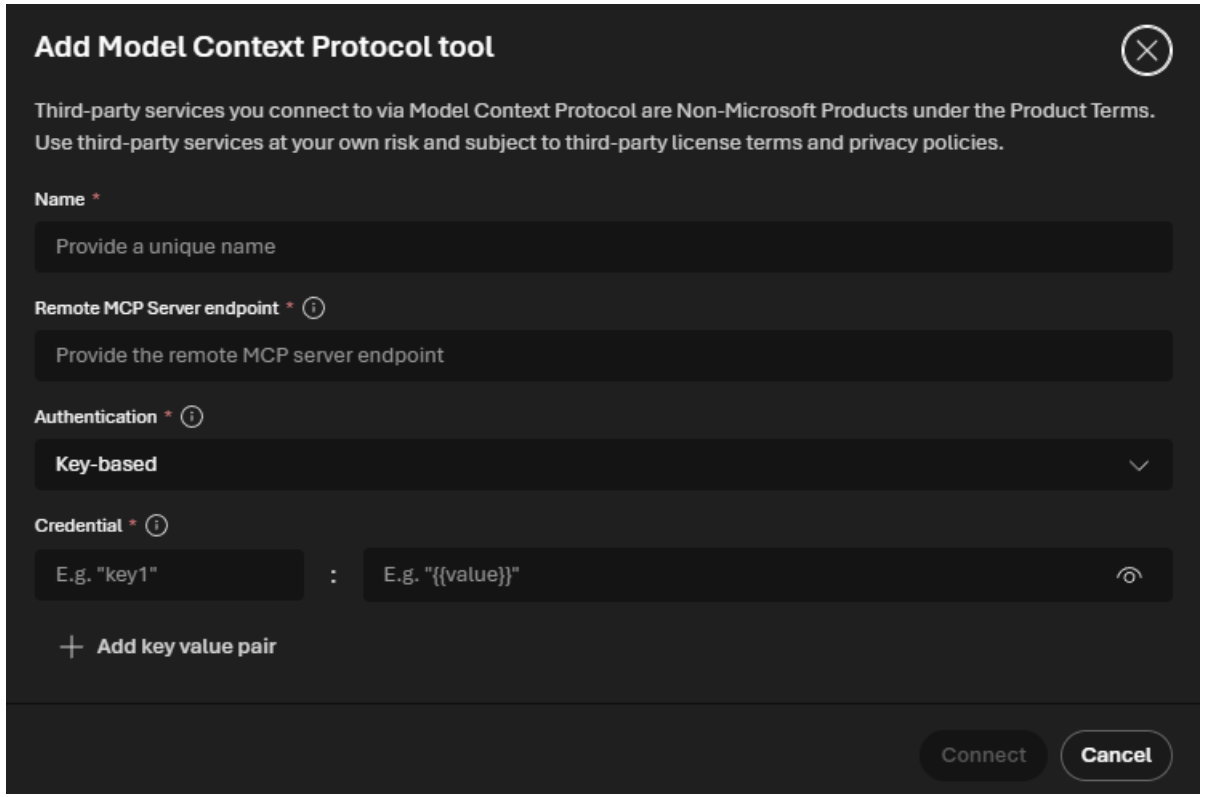


Microsoft provided

Developer provided

Adding an MCP server to Foundry agent





#Milano

Copilot Studio and the Flight MCP server

Demo

#GLOBALAZURE
2026
#Milano

Slide e video:
https://www.globalazuremilano.it