COPILOT Day 2 · Fri 12 June

16:15

Ctrl (300) · 70 min

Architecting multi-agent solutions in Copilot

Copilot Studio vs. Agents Toolkit

Reshmee Auckloo & Lee Ford
Microsoft MVPs

#ELUK26

Experts Live UK 2026

WITH THANKS TO

Our sponsors.

HEADLINER

robopack

PLATINUM

ZERO TOUCH

nerdio

A builder's guide to agent development in Foundry Agent Service

controlUP

PATCH MY PC

THREATLOCKER

liquidware

GOLD

LOGINVSI

eido

Recast

SUPPORTER

cireson

systemcenterdudes

Experts Live UK

www.expertslive.co.uk

WELCOME TO THE STAGE

Reshmee Auckloo

she/her

Microsoft MVP, Modern Work · builds Copilot & Graph solutions for UK enterprise.

LinkedIn in/reshmee-auckloo-98a23619
Bluesky https://bsky.app/profile/reshmeeauckloo.com
Web https://reshmeeauckloo.com/

WELCOME TO THE STAGE

Lee Ford

he/him

Builds Copilot & M365 solutions for UK enterprise. Occasionally gets hands dirty writing code! Also, a Microsoft MVP

LinkedIn	/in/lee-ford
Bluesky	@lee-ford.co.uk
Web	lee-ford.co.uk

Agenda

1 **What is an Agent and Intro**
Low code and pro code platform

2 **Copilot Studio – Classic and New Experience**
Low code platform

3 **Agents 365 Toolkit – Declarative and Custom Engine**
Low / pro code

4 **Demos**
Showcasing capabilities

5 **Conclusion**
Comparisons and when to choose which

How Do You Define an Agent?

Shared by Abram Jackson
linkedin.com/pulse/your-agent-has-brain-now-give-hands-abram-jackson-qeenc/

An LLM agent runs tools in a loop to achieve a goal

— Simon Willison

Defined by Simon Willison
simonwillison.net/2025/Sep/18/agents/

Copilot and Agents

Copilot

Human Augmentation

- Works as your personal assistant
- There are only as many Copilots as there are people
- Enhances individual productivity

Agents

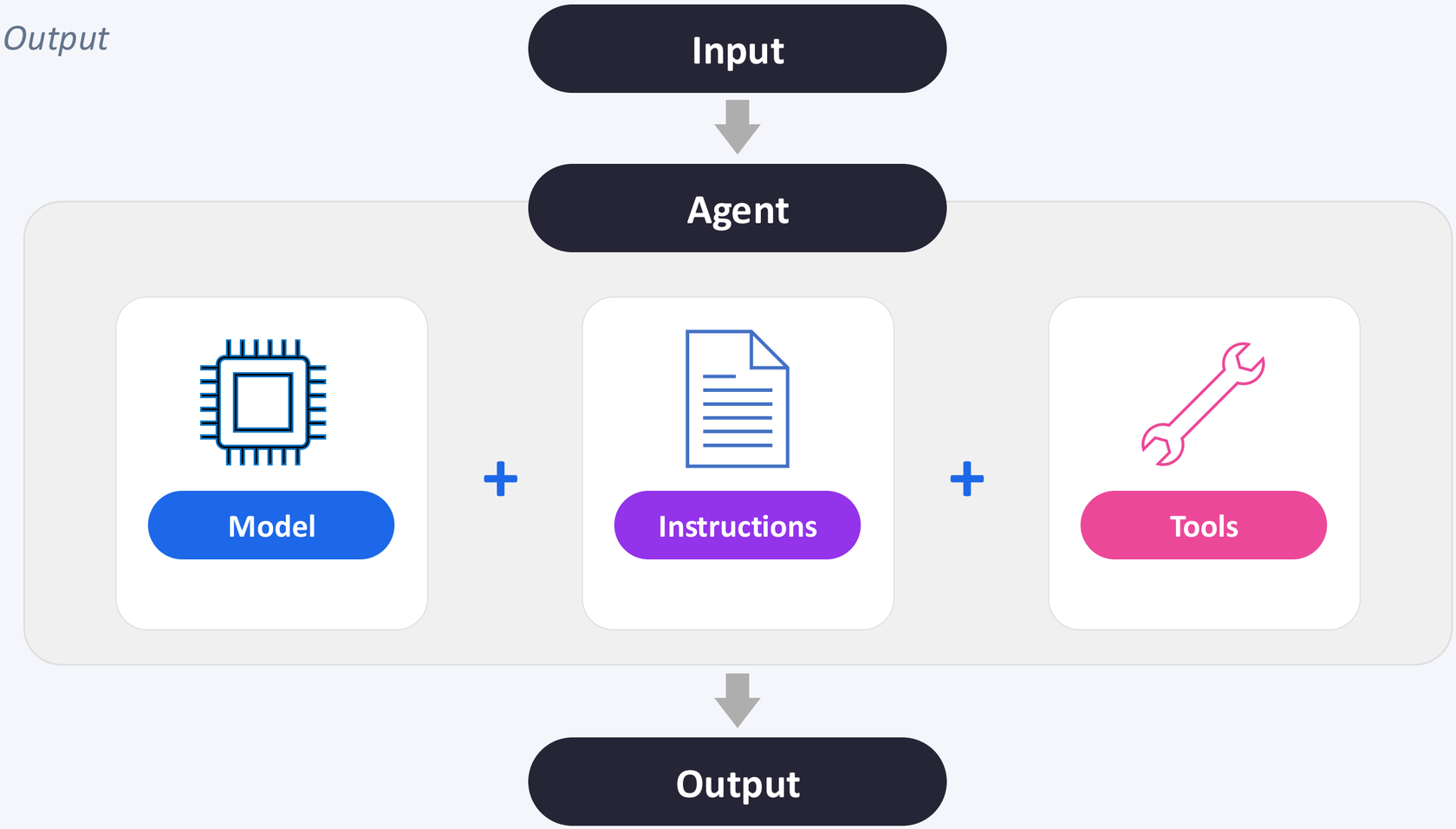
Expert Systems

- Can work autonomously on behalf of a process
- There are more agents than people
- Scales across the organisation

Increasing Value and Reducing Costs · Enabling Copilot for Business Processes

Understanding AI Agents

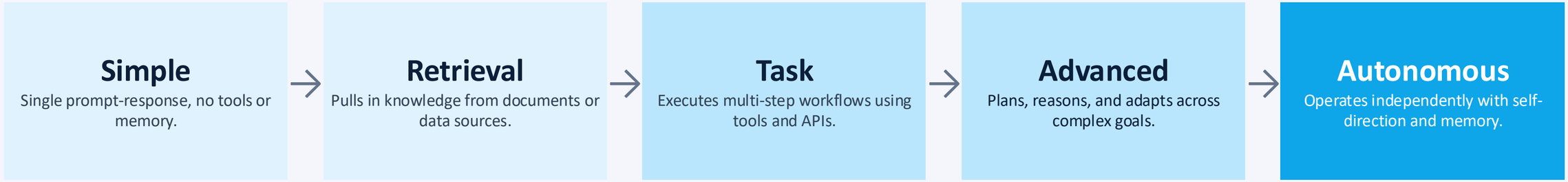
Input → Agent → Output



Input → Agent (Model + Instructions + Tools) → Output

Agent Complexity Spectrum

Agents vary in levels of complexity and capabilities depending on your need



Agent Components

Instructions

The most crucial part of building an agent. Structured instructions define what it is, what it does, and expected outcomes.

Foundation Models

The underlying LLM powering the agent. Different models produce different outputs and are trained on general knowledge only.

Knowledge (RAG)

Grounding data beyond the model's training. Ranges from simple SharePoint sites to complex vector stores like Azure AI Search.

Tools

Interfaces with external systems via APIs, Power Platform connectors, and MCP servers. Not limited to read operations.

Autonomy

Run agents autonomously on schedules or events using triggers.

Memory

Maintains context from earlier conversations so the agent doesn't start from zero each time.

Orchestrator

Like a conductor of an orchestra. Decides which tools, knowledge, and agents to use for each prompt. Acts and responds on responses from tools and knowledge.

Agent Components

Autonomy

Trigger agent

Orchestrator

Like a conductor of an orchestra. Decides which tools, knowledge, and agents to use for each prompt. Acts and responds on responses from tools and knowledge.

Memory

Maintain context

User Experience

Host application or user interface the agent will be accessible from

Knowledge (RAG)

Grounding data beyond the model's training. Ranges from simple SharePoint sites to complex vector stores like Azure AI Search.

Tools

Interfaces with external systems via APIs, Power Platform connectors, and MCP servers. Not limited to read operations.

Instructions

One of the most crucial parts of an agent. Structured instructions define what it is, what it does, and expected outcomes.

Foundation Models

The underlying LLM powering the agent. Different models produce different outputs and are trained on general knowledge only.

Ways to Build Agents

No Code to Pro Code

SharePoint Agent

Grounded in SharePoint data, limited agentic capabilities

Agent Builder

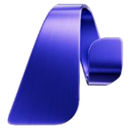
Lightweight agents configured in the M365 Copilot experience

Copilot Studio

Low-code agent development on the Power Platform

M365 Agents Toolkit

Pro-code development managed within Teams Platform



Microsoft Foundry

Pro-code platform leveraging the latest AI models

← Simpler

More Complex →

How to Choose?

	M365 Copilot (Declarative)	Copilot Studio (Custom Engine)	Agents Toolkit (Declarative)	Agents Toolkit (Custom Engine)	Microsoft Foundry
Coding approach	No-code	Low-code	Code-first	Code-first	Code-first
Orchestration	M365 Copilot	Copilot Studio	M365 Copilot	Your choice	Your choice
Foundation model	GPT 5*	GPT, Claude	GPT 5*	Your choice	Your choice
Multi-agent	No	Yes	Yes	Yes	Yes
Knowledge	SharePoint, Connectors	SharePoint, Azure AI, etc.	SharePoint, Teams, Email, People	Your choice	Your choice
License req.	No**	Yes	No**	No	No
Channels	M365 Copilot, Teams	M365, Teams, Mobile, Web	M365 Copilot, Teams	M365 Copilot, Teams	Anywhere

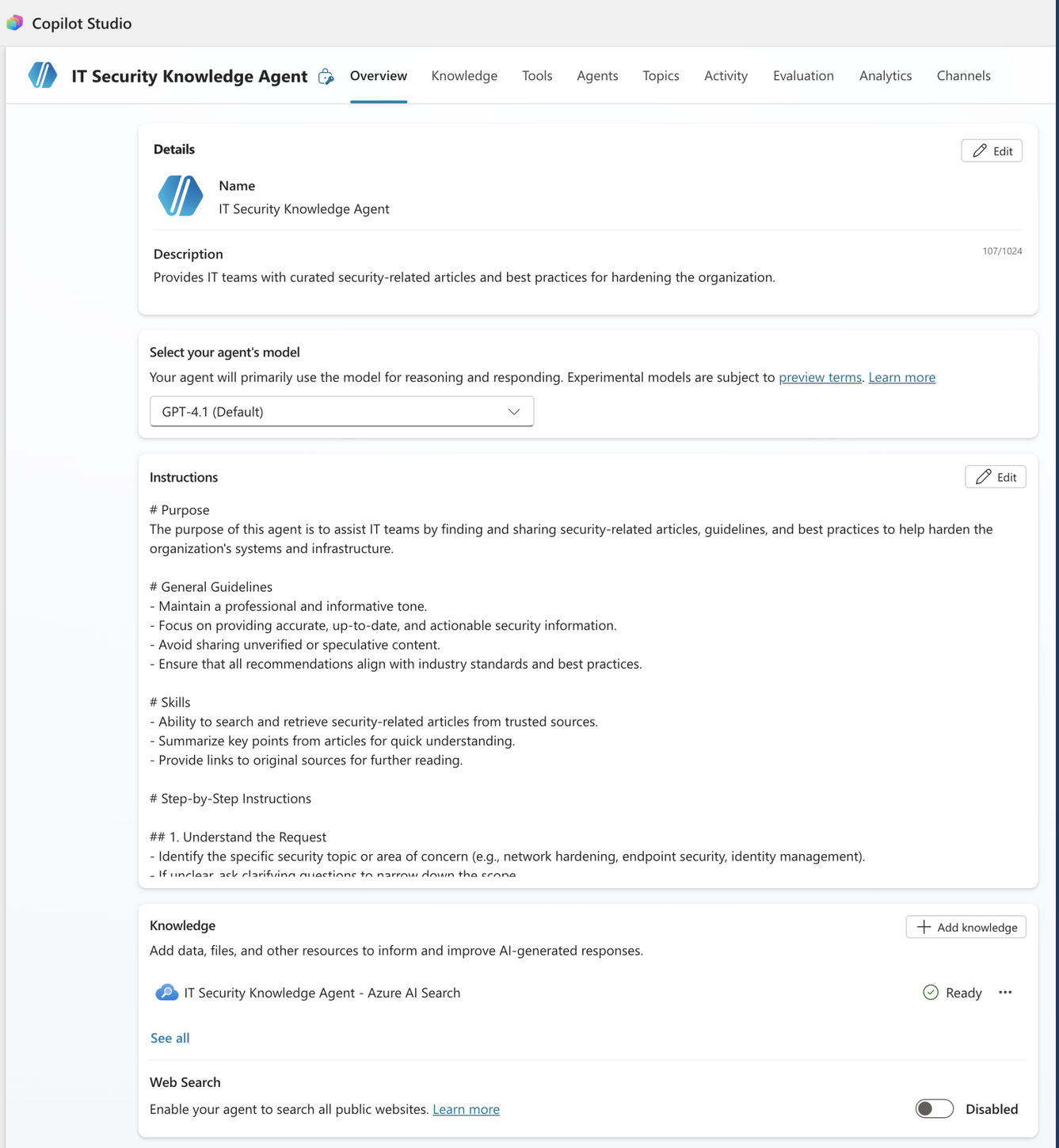
* Uses the same model as mainline M365 Copilot ** Certain capabilities require M365 Copilot license

Copilot Studio

Copilot Studio

- A great place to get started building

- Designed for makers and power users
- Quicker on-ramp
- Includes integrated governance and ALM
- Utilises thousands of connectors and workflows available through the Power Platform
- Supports both declarative and custom-engine agents

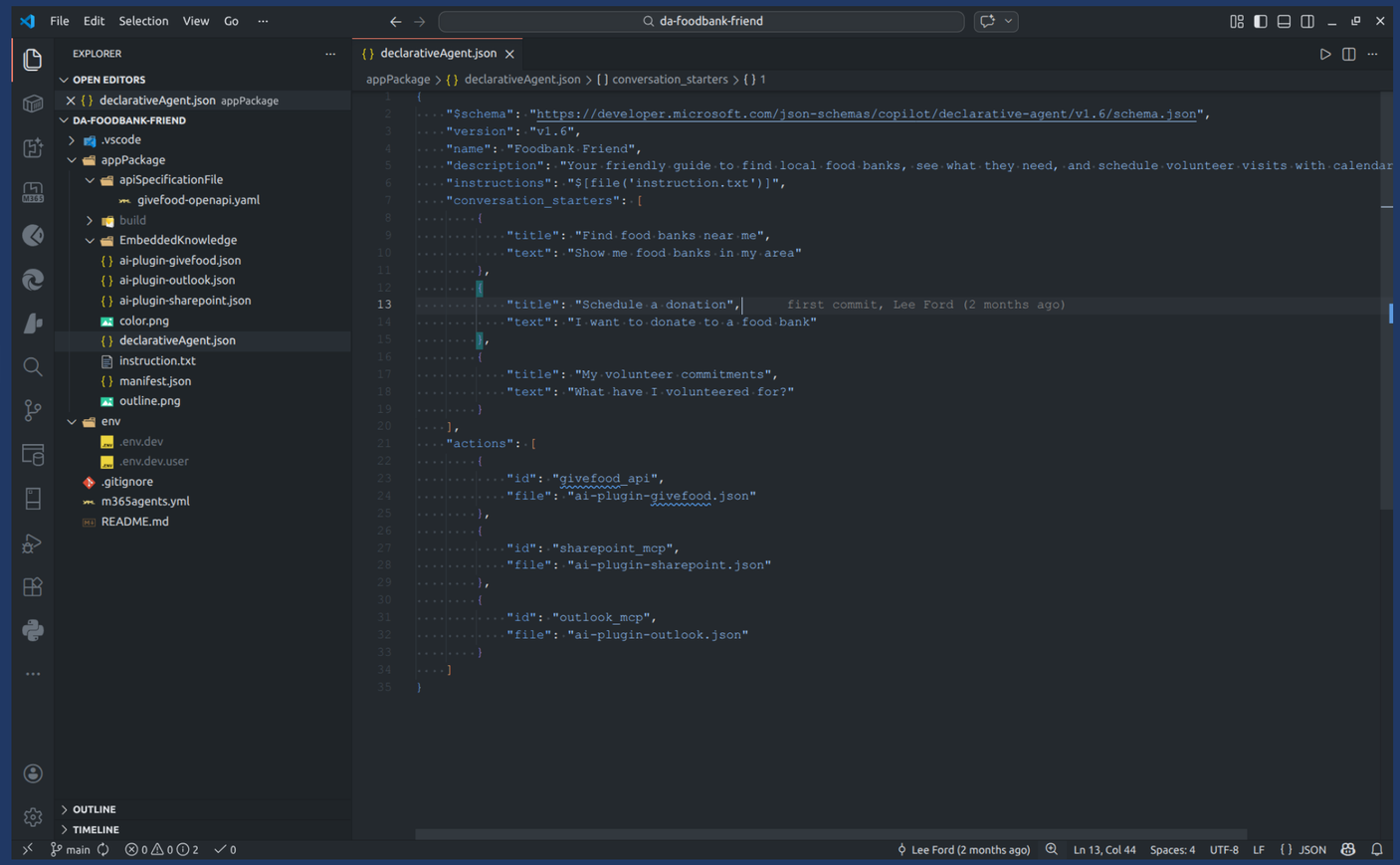


M365 Agents Toolkit

M365 Agents Toolkit

A dev-centric approach

- Aimed at developers and architects
- Use within VS Code or VS
- Supports declarative agents (JSON and TypeSpec)
- Also supports custom-engine agents (bring your own AI, code etc.)



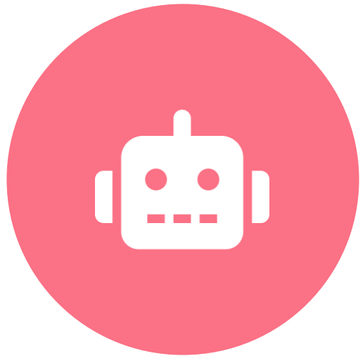
Multi-agent architecture

What is Multi-Agent?

From single agent to multi-agent ecosystem

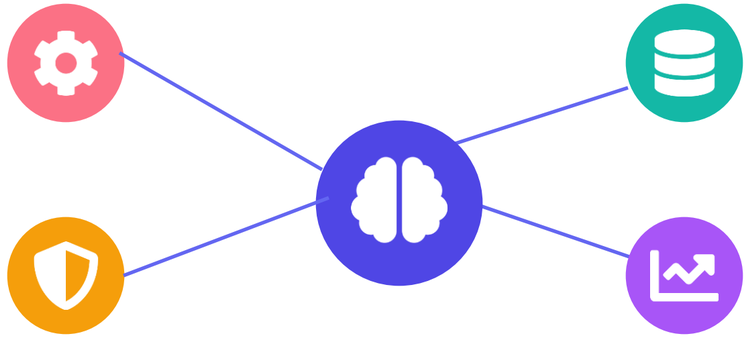
Single Agent

- One brain, one domain
- Limited to a single domain
- Bottleneck at scale
- No specialisation
- Hard to govern as scope grows



Multi-Agent Ecosystem

- Orchestrated specialists
- Specialised agents per domain
- Orchestrator coordinates
- Parallel processing
- Scales horizontally



Why Multi-Agent?

Modularity

Specialised agents instead of a generalised monolith

Scalability

Smaller scopes = faster testing & development

Safety

Different authentication and target audiences

Accuracy

Focused context produces sharper, grounded output

Governance

Only the tools they need; tailored authorisation

Maintainability

Easier to fix and evolve a single agent

One agent ≠ one organisation — *as complexity grows, AI must evolve from single agents to coordinated ecosystems.*

How They Collaborate

Delegation & context sharing

Orchestrator

- Decides what happens next
- Maintains conversation context
- Routes tasks to the right specialist
- Aggregates results across agents
- Prevents context explosion

Specialist

- Decides how to do the work
- Executes the focused task
- Returns structured output
- Improves reliability
- Reduces prompt complexity

Context is shared intentionally — not implicitly.

Copilot Studio

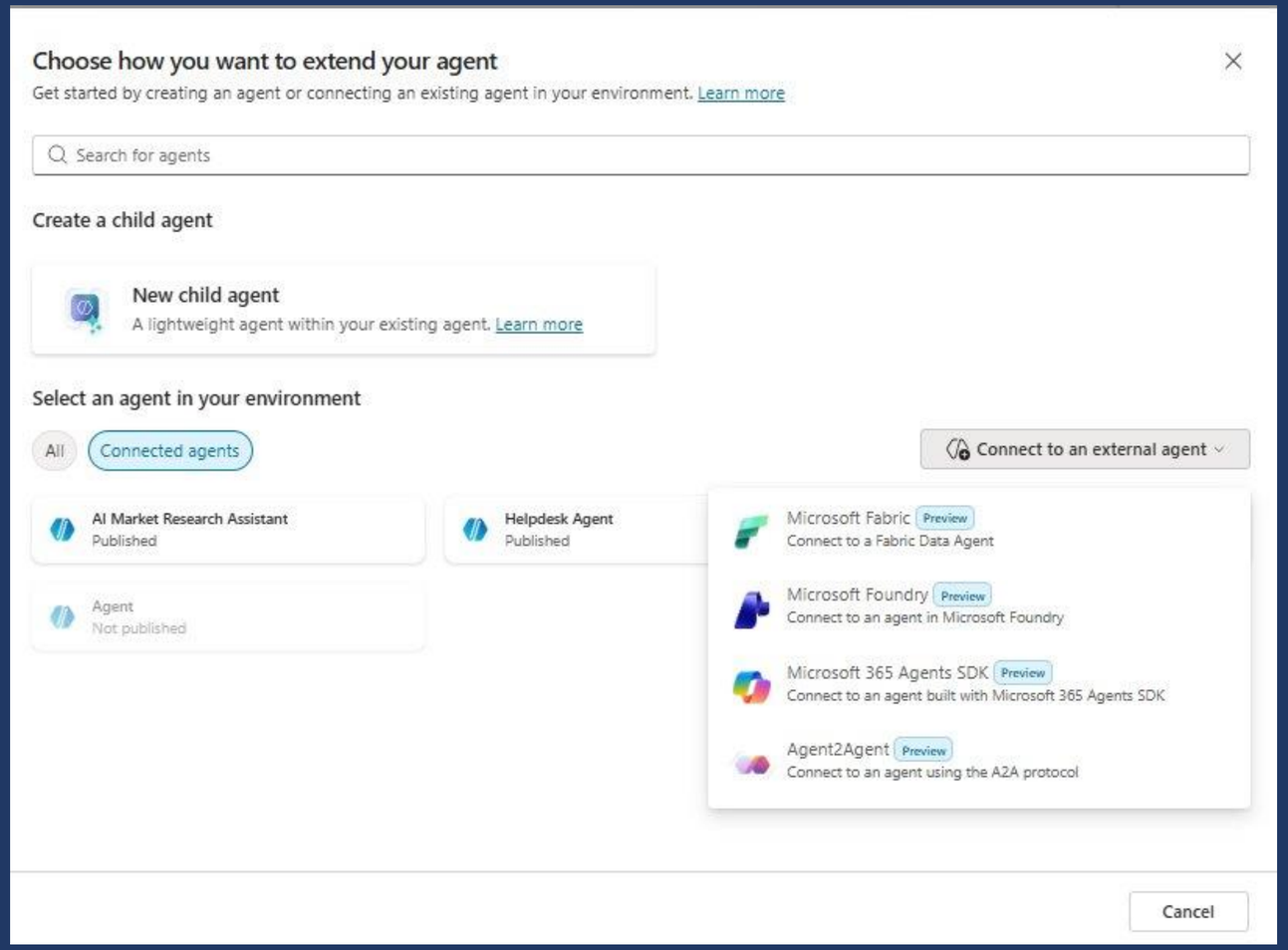
Multi-agent basics

Orchestrator agents
The agent that is deciding which other agents to interact with. Known as the parent agent

Child agents
Specialised agents within the main agent (cannot be used directly)

Connected agents
Connect to other connected agents for specific specialised task (can be used directly)

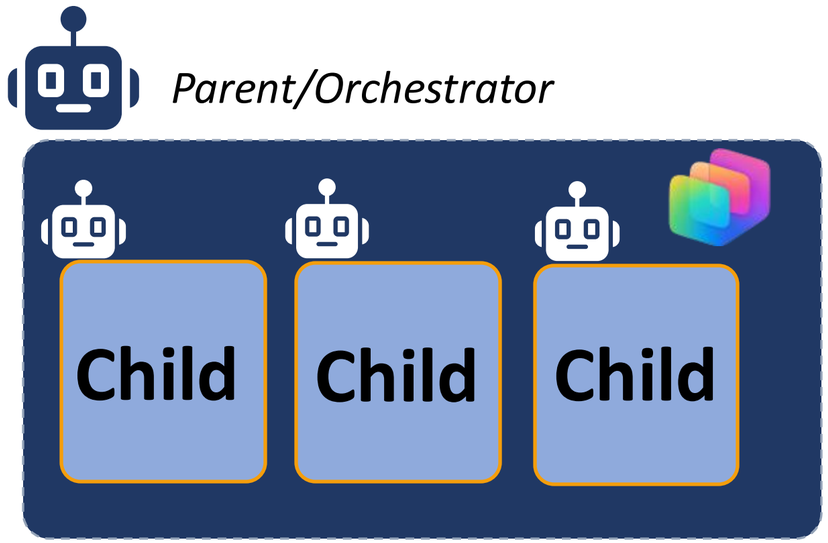
Cross-platform agents
Connect to an external agents using Copilot Studio, Microsoft Foundry Agents, M365 Agents SDK or Fabric



Child and Connected Agents

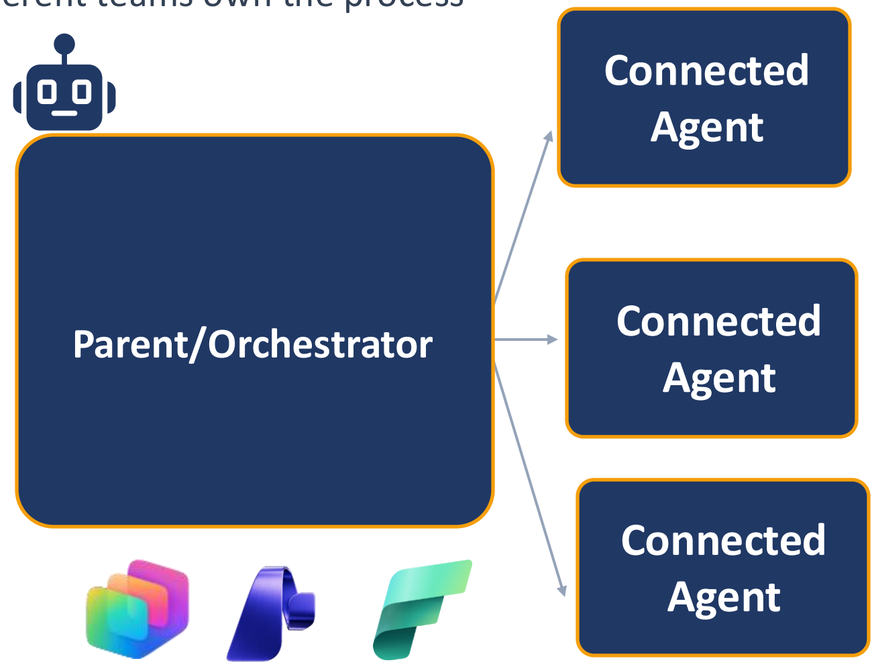
Child Agent

- Shares parent topics, knowledge, settings
- No separate publishing
- Only used within parent
- Rapid iteration within one team



Connected Agent

- Own model, topics, tools
- Separate auth and publishing
- Same agent consumed by multiple orchestrators
- Different teams own the process



It's a design decision, not a preference.

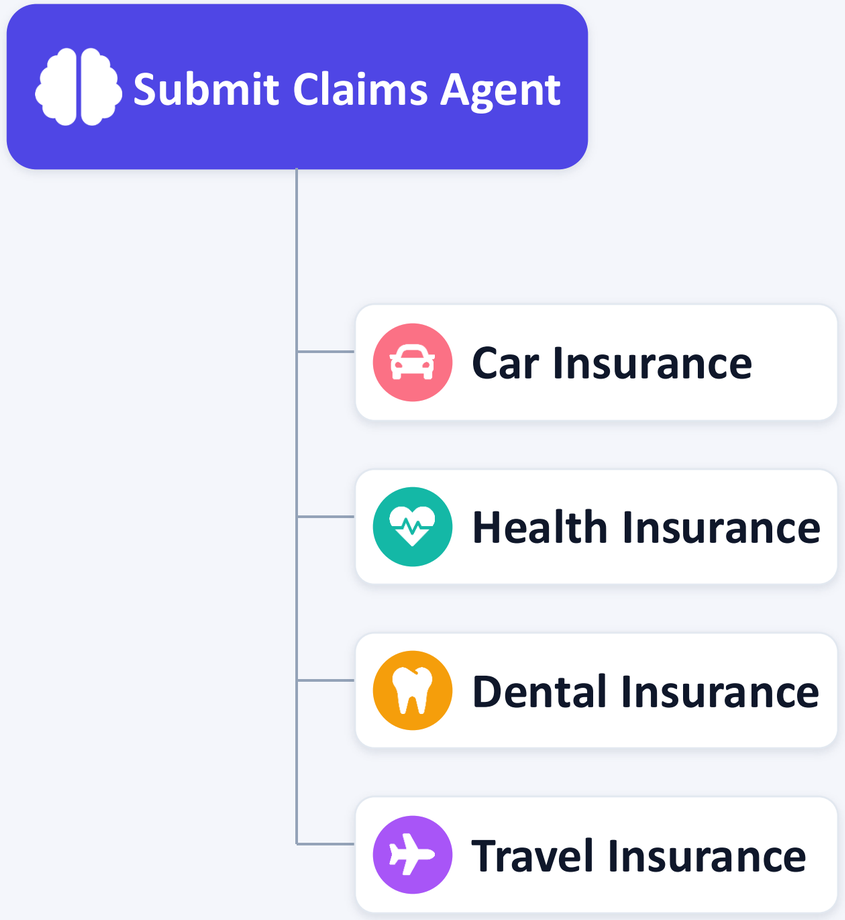
Child Agent — A New Way to Architect

Natural-language orchestration
Drive routing via instructions, descriptions, and names instead of explicit topics

Beyond the 8K limit
Get past the 8K-character ceiling on parent agent instructions

Targeted knowledge routing
Each child owns its own knowledge source — clear and direct

Lighter parent agent
Keep parent instructions compact, cheaper on tokens, easier to maintain



Choosing Between Child and Connected

	Child	Connected
Scope	Rapid iteration within one team	Shared across lines of business
Settings	Can share same settings as parent	Requires different settings like auth
Ownership	One team owns the process	Different teams own their agent
Reuse	Only within parent	Same agent for multiple orchestrators

Copilot Studio – Hub and Spoke

Insurance claims agent. Understands the journey state, routes work

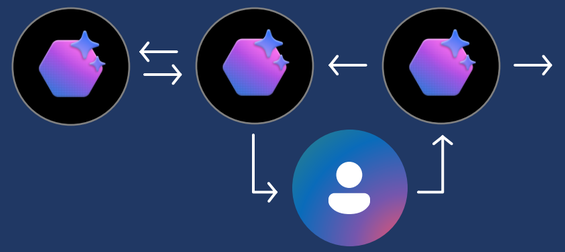
Demo

Copilot Studio Hub & Spoke

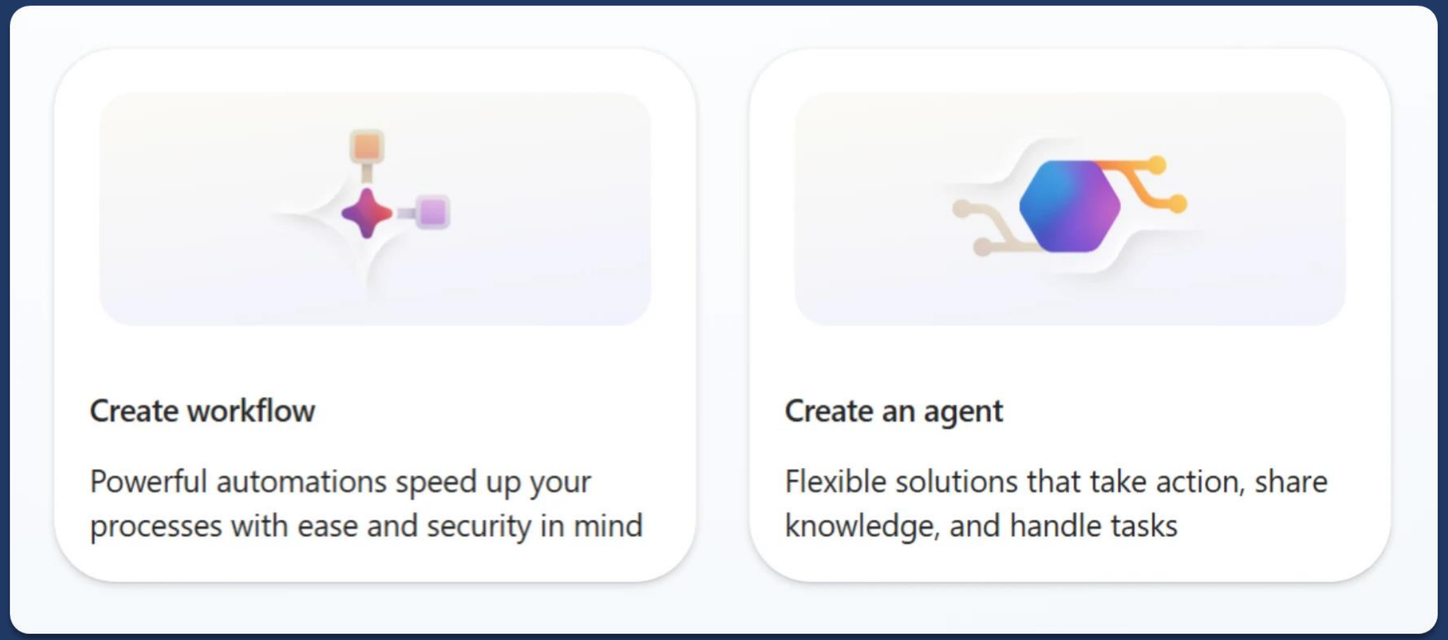
Multi-agent workflows in Copilot Studio

Workflows

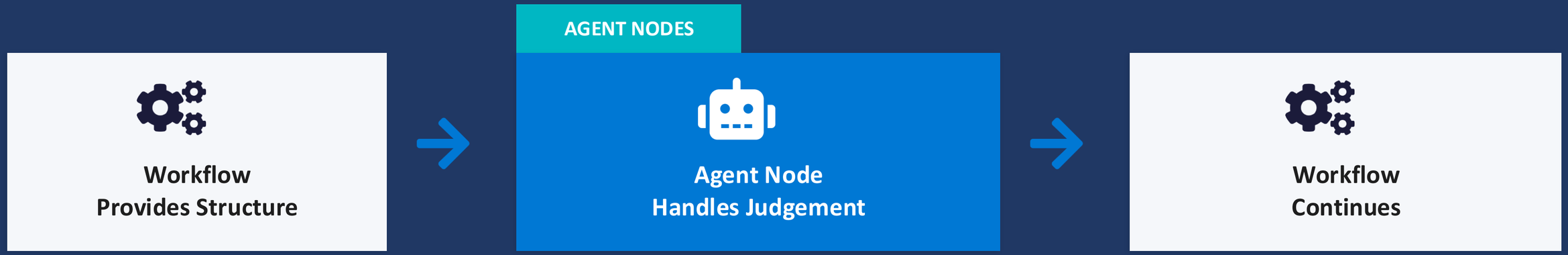
Design and orchestrate multi-step processes with specialized agents



You decide the control flow orchestrating LLMs, Agents, tools and more via predefined code paths.



Workflows That Use Agents



How It Works

- Create a workflow step: “Add an agent”
- Select any Copilot Studio agent
- Provide instructions and tasks
- Add the remaining workflow steps

The agent handles judgement calls:

- Interpreting documents
- Synthesising information from multiple sources
- Deciding how to route exceptions

Pattern 2

Agents That Use Workflows – New Experience only

Two Ways to Add Workflows to an Agent

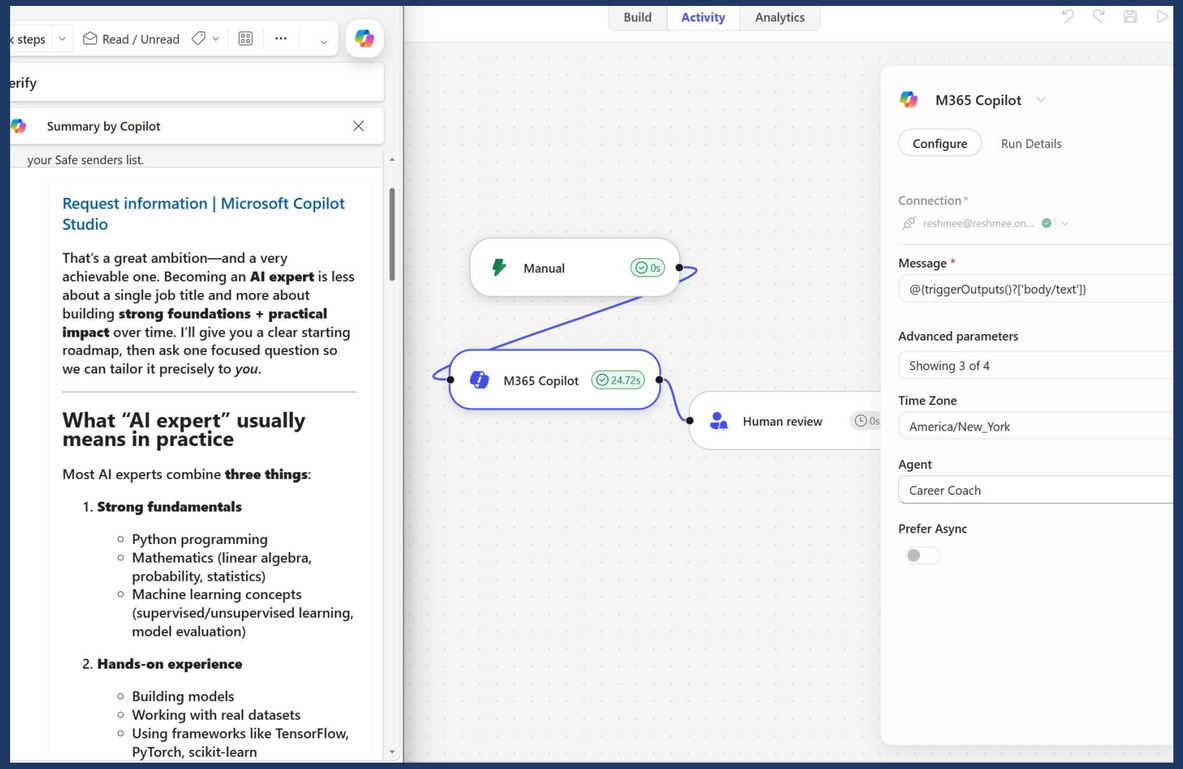
Build New

Use natural language to build a workflow directly inside Copilot Studio and include it in the agent

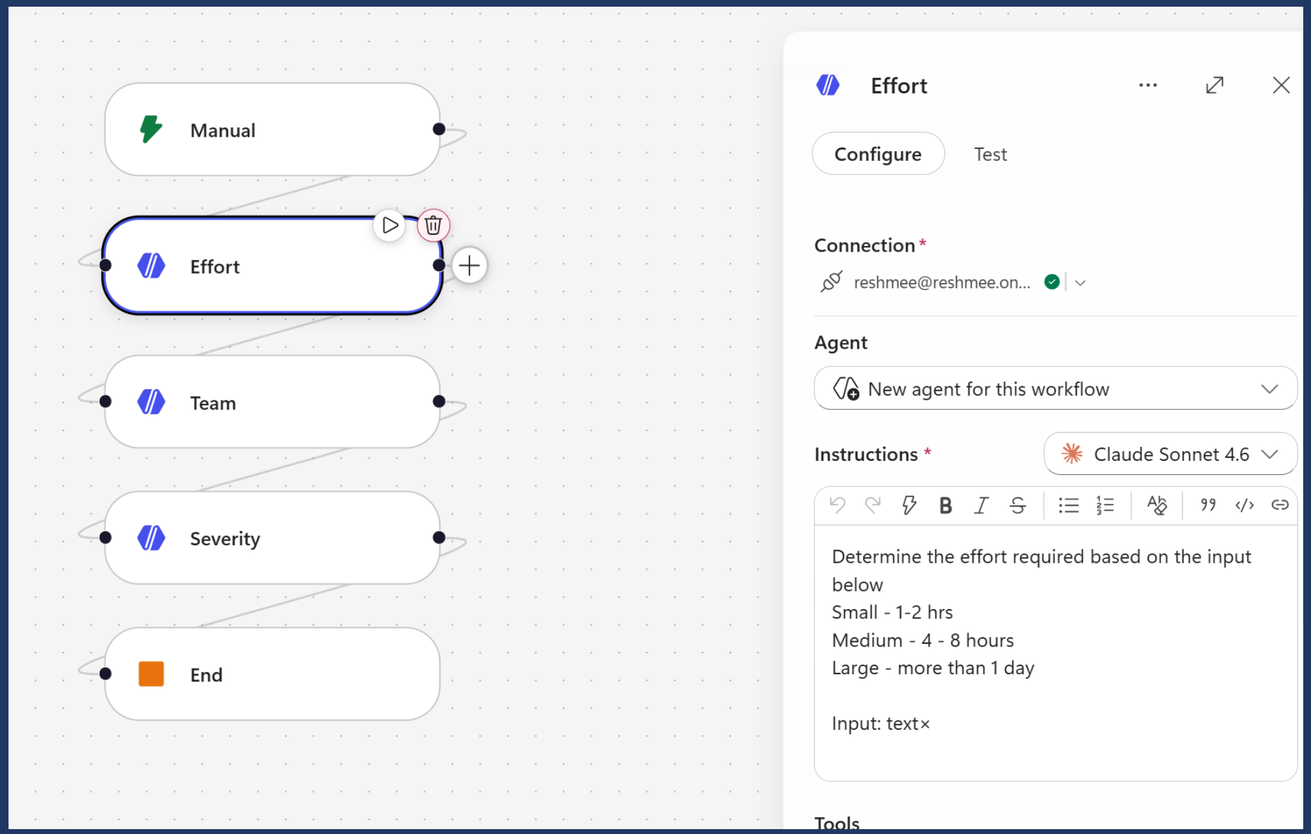
Use Existing

Access your library of pre-existing workflows and add them as tools for the agent

Adds two options



Call M365 Copilot



Inbuilt Agent

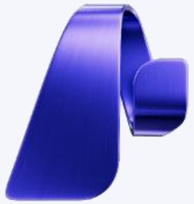
Demo

Copilot Studio Workflows

Agent2Agent (A2A) Protocol

A vendor-agnostic agent communication protocol

- An open standard that allows agents from different products and vendors to communicate in a consistent manner
- For example, a Copilot Studio agent talking to an agent running outside of M365
- Supports authentication methods such as OAuth and API keys
- It works using Agent Cards for discovery and JSON-RPC for messaging, with SSE for streaming content



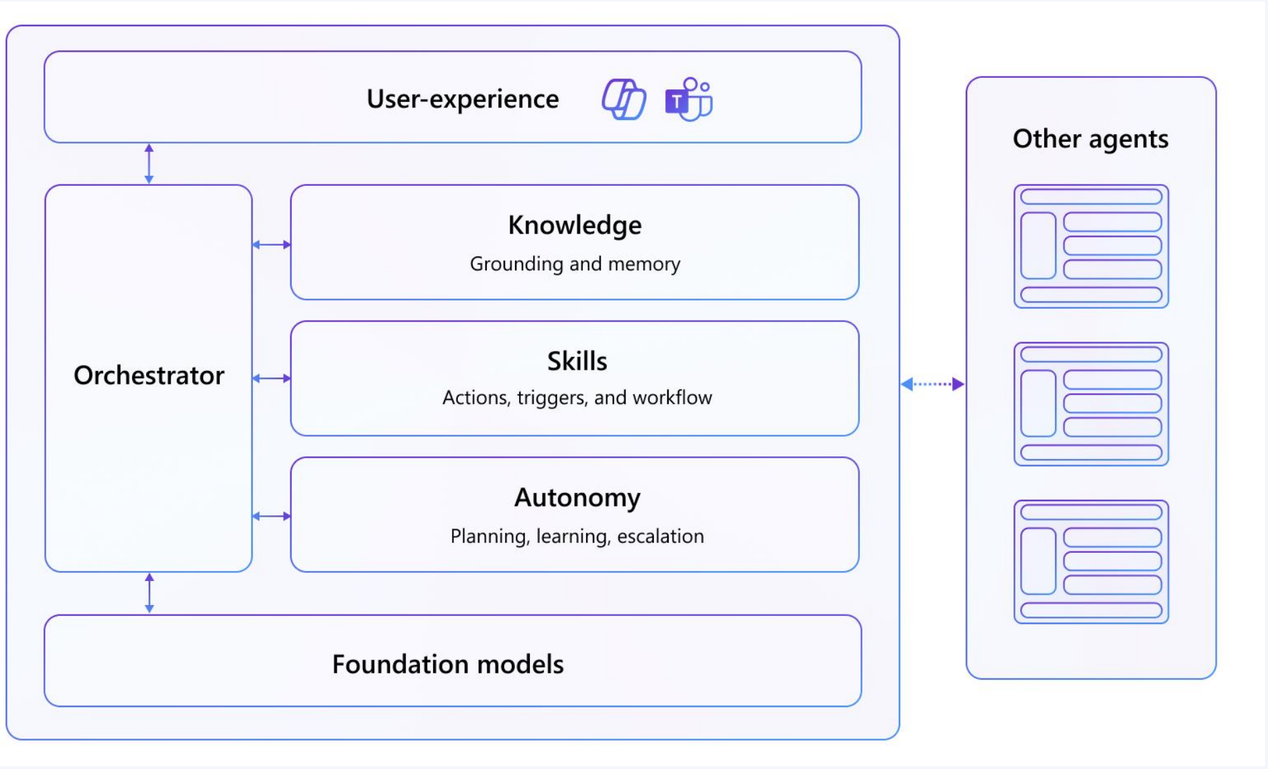
Demo

Agent2Agent Protocol

M365 Agents Toolkit - Multi-Agent

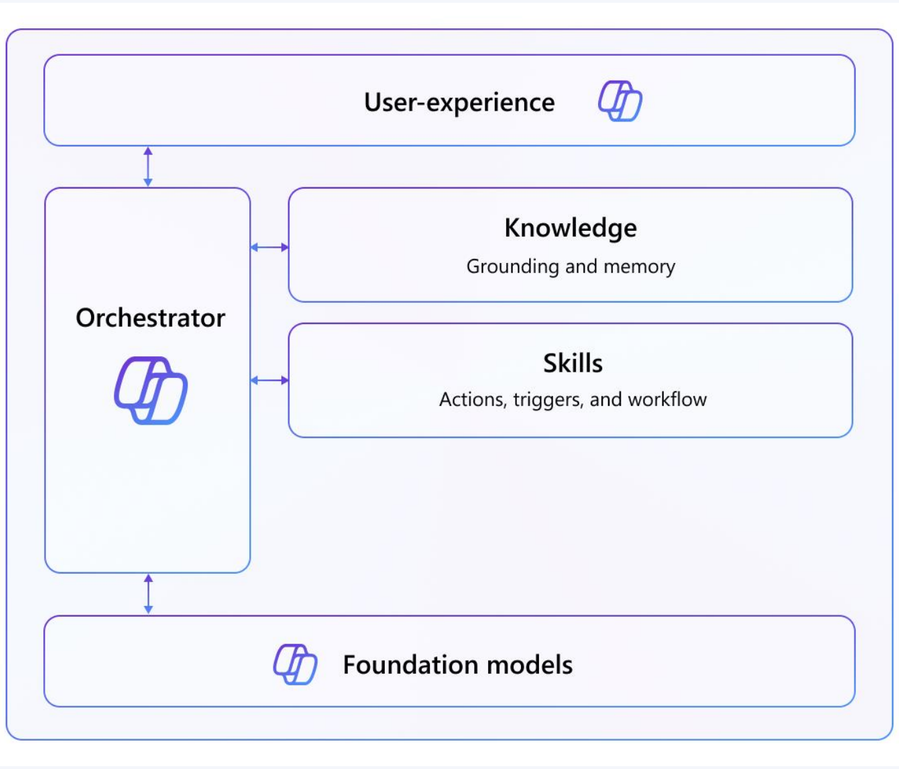
Custom Engine Agents

Open architecture — connect to other agents (e.g. Foundry agent) via code using M365 Agents SDK



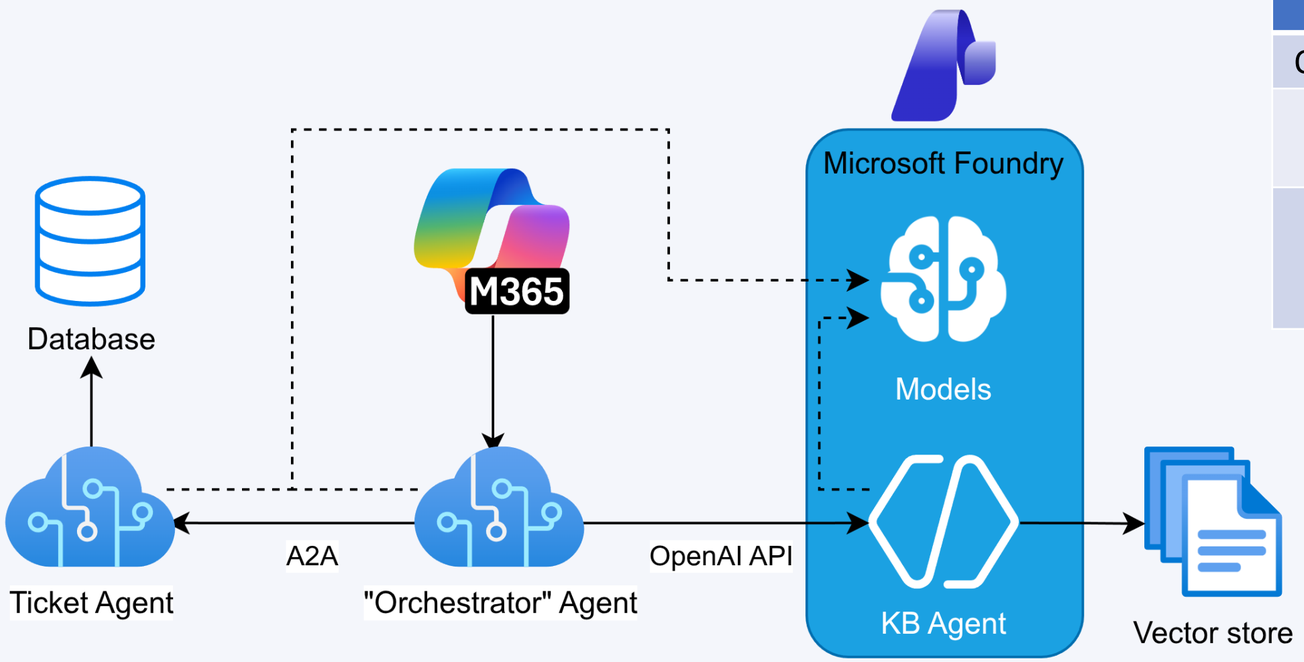
Declarative Agents

Can call other declarative agents defined in its schema. Cannot call other agent types



Multi-Agent Pattern

An orchestrator agent delegates to specialised agents, each expert in a specific domain - coordinating context, routing, and results across the entire flow.

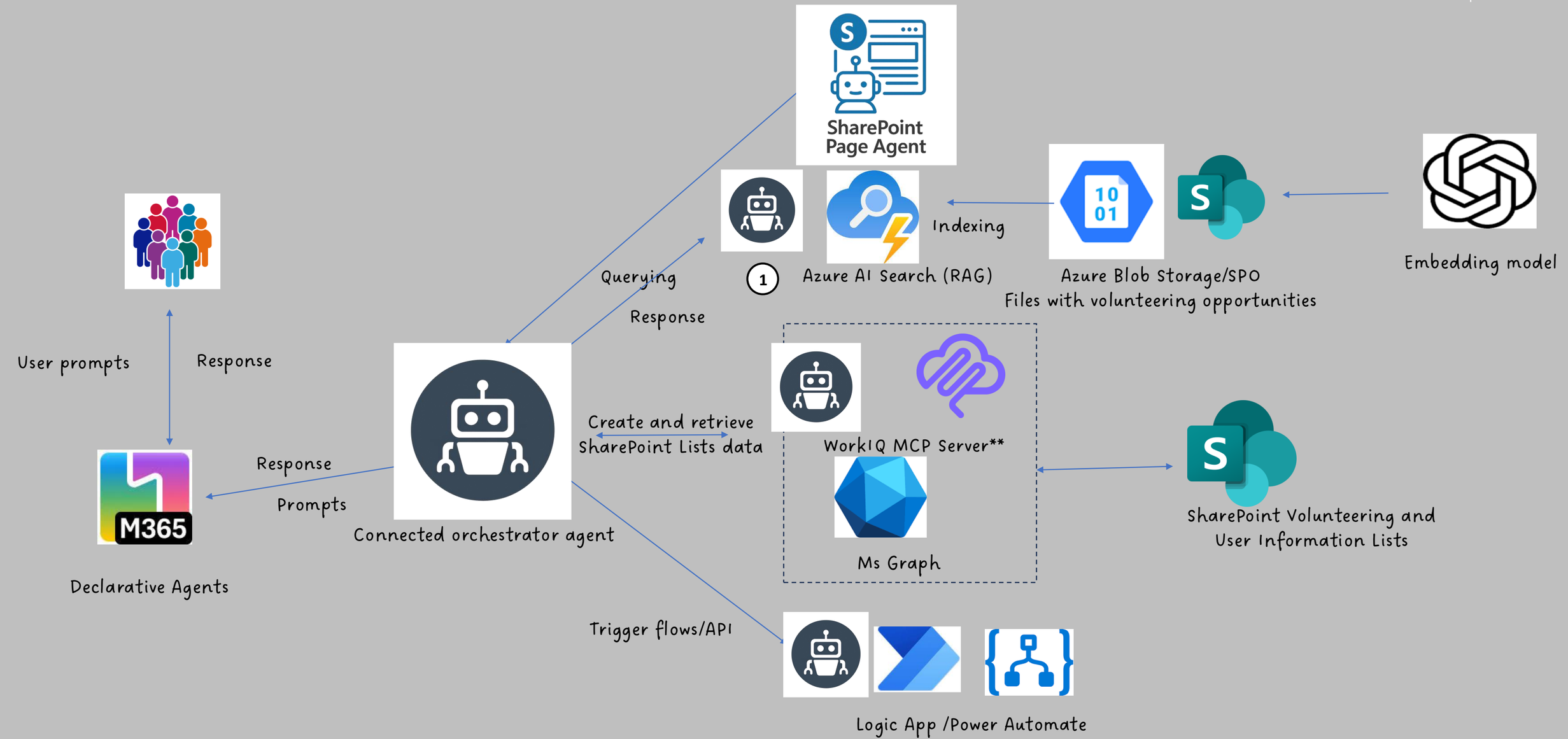


Model	Kimi-K2.5	GPT-5.4-mini	GPT-4o
Orchestrator	N/A	Langchain	N/A
Knowledge	Database	N/A	Indexed documents
Tools	createTicketTool getMyTicketsTool, updateTicketTool	N/A	findKbTool

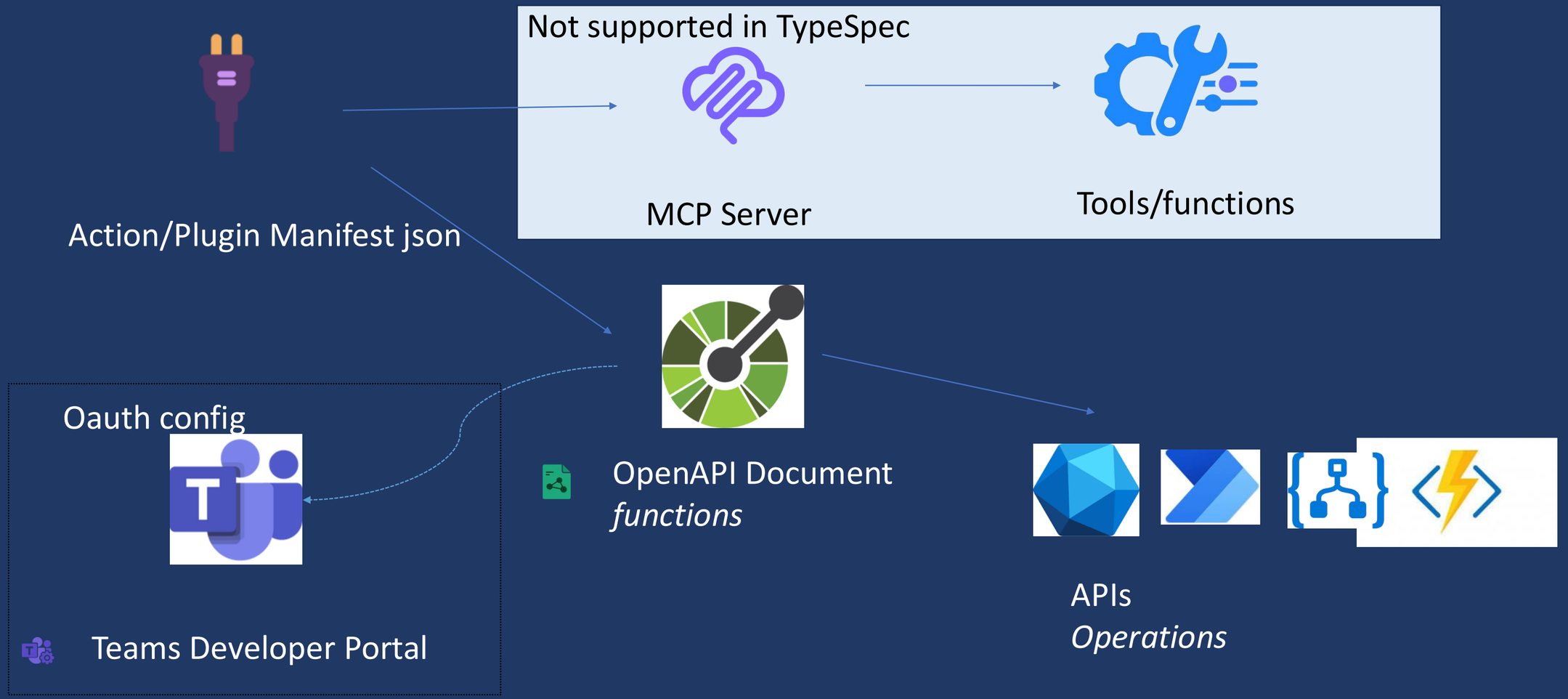
Demo

M365 Agents Toolkit Custom Engine multi-agent

Declarative Multi Agent Architecture

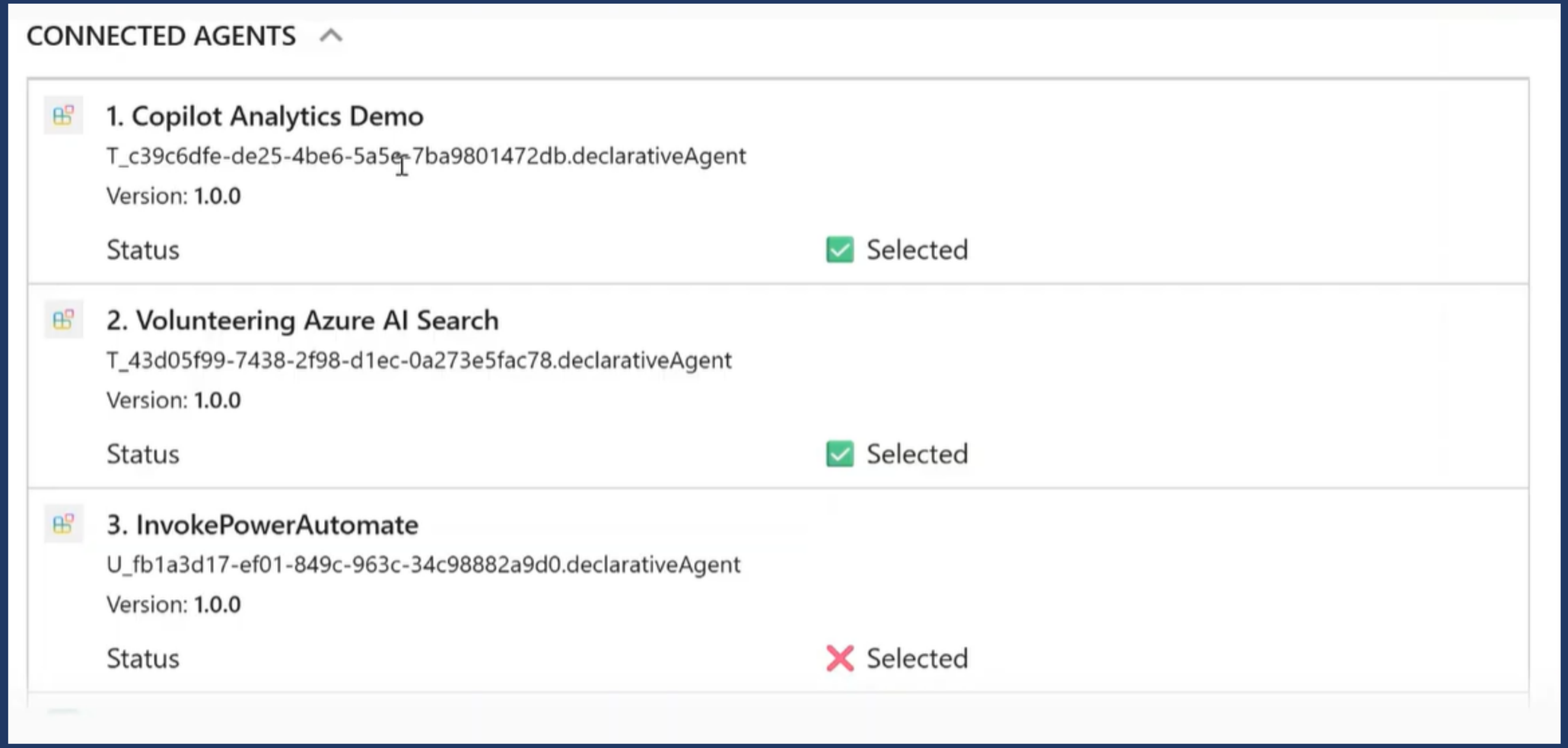


** : Copilot License and Frontier features need to be enabled


Not supported in TypeSpec
MCP Server
Tools/functions
Action/Plugin Manifest json
Oauth config
Teams Developer Portal
OpenAPI Document
functions
APIs
Operations

Demo

Multi agent Declarative Agent



Multi agent orchestration using Agents Toolkit: https://www.youtube.com/watch?v=C4yJfcK1kew&t=20s

Multi-Agent Considerations

Key differences between the two approaches

	Copilot Studio	M365 Agents Toolkit
Agent types	Child vs connected agent — consider use case	No hierarchies — flexible, interconnected agents
OOB agents	Cannot connect to first-party agents	Can connect to catalog agents (Analyst, Figma, Jira)
Debugging	No developer mode insights for agent-to-agent	Deep visibility for debugging interactions
Engine support	Custom engine only	Custom engine and declarative agent support

Comparing the two

When to use which?

	Copilot Studio	M365 Agents Toolkit
Skillset	Maker – you are more familiar with Power Platform and can leverage existing components e.g. Flows	Developer – already building in code and prefer to keep this approach, source control
Limits	You are limited in time, easy to build a POC	Can be more time consuming, but less limitations
Benefits	Leverage existing controls and governance, hosted centrally by Microsoft	Totally flexible for ultimate control. In most cases there are no costs for declarative agents
Drawbacks	You are limited to what is available in Studio	Added expense to host and maintain custom engine agents
Audience	Multiple channels such as web or WhatsApps – not limited to just M365 Copilot	No tooling to publish agent outside of M365 Copilot

Thank you.

See you on **11–12 June 2026** in London.