Concurrency

Copilot Studio

Multi-Agent Design

Corey Milliman

Copilot Studio & Multi-Agent Systems – Complimentary Session

- 30 minutes with our experts to learn how to build, orchestrate, and manage AI agents effectively, with live demos showcasing multi-agent collaboration.
- Get practical insights and answers to your specific questions.
- Schedule a follow-up — **we'll drop the bookings link in the chat during the session.**

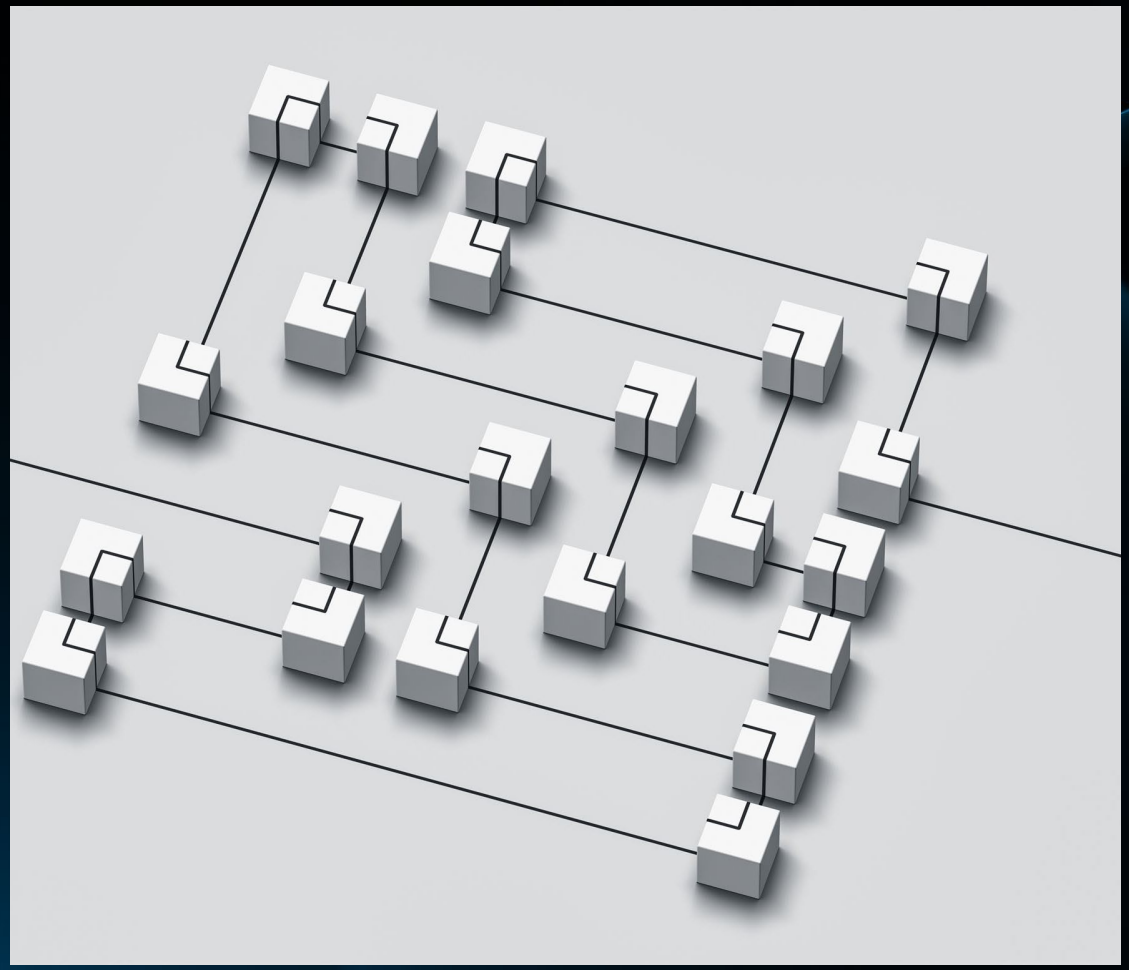
Concurrency

Agenda

- Introduction to Multi-Agent Copilot Studio
- Configuring and Managing Agents in Copilot Studio
- Parent-Child Model: Child Agents
- Agent-to-Agent Model: Connected Agents
- Comparing Child and Connected Agents
- Practical Demonstration: Multi-Agent Copilot Scenario
- Best Practices for Multi-Agent Copilot Design
- Limitations and Considerations

Concurrency

Multi Agent Capabilities



- **Collaborative Copilot Design**
  - Copilot Studio enables designing copilots that collaborate instead of working in isolation for enhanced efficiency.
- **Parent-Child Hierarchy**
  - Parent copilots delegate tasks to child copilots, creating a structured workflow for task completion.
- **Peer Collaboration**
  - Connected peer copilots collaborate using orchestration to provide seamless user experiences.
- **Modular Team of Specialists**
- Modular copilots act as specialist teams, each focused on their role but coordinated for effectiveness.

Concurrency

Benefits of Multi-Agent Systems

Concurrency

Simplify Complexity

- Splitting responsibilities into multiple agents keeps logic simpler and easier to manage and test.

Specialization of Roles

- Each agent focuses on a specific domain, improving accuracy, tone, and depth in that area.

Maintainability

- Updating one agent does not disrupt others, reducing maintenance complexity and cost.

Scalability and Teamwork

- Adding new agents expands capabilities easily, enabling modular growth like a coordinated team.

CONNECTING AND PUBLISHING AGENTS

Concurrency

Environment and Publishing

- Only agents in the same environment that are published can be connected to ensure stability and consistency.

Allow Connections Control

- Agents must explicitly allow incoming connections to control reuse and prevent accidental linking.

History Passing Toggle

- Toggle conversation history passing to allow continuity or isolated interaction in connected agents.

Manage Connections Flexibly

- Agents can be enabled, disabled, or disconnected without deletion to support testing and staged rollouts.

DESIGN AND FUNCTIONALITY OF CHILD AGENTS

Hierarchical Design

Child agents operate inside a parent agent, ensuring a coherent single conversation experience for users.

Invisible to Users

Child agents work behind the scenes and do not interact with users directly, simplifying user experience.

Event-Triggered Actions

Child agents can activate on events like inactivity, invocation, or workflow completion, expanding their functionality.

Priority Management

Priorities determine which child agent responds first when multiple agents could handle the same event

Concurrency

PEER COLLABORATION AND ORCHESTRATION

Concurrency

Connected agents collaborate as independent peers usable across multiple enterprise solutions.

Each agent manages its own lifecycle and governance, enabling departmental ownership with shared access.

Generative orchestration calls one or multiple agents, merging results for seamless workflow solutions.

Distinct domains for agents prevent confusion and ensure accurate orchestration across tasks.

KEY DIFFERENCES AND USE CASE SELECTION

Concurrency

Hierarchy vs Peers	• Child agents depend on a parent for visibility, while connected agents operate as independent peers.
Visibility Differences	• Child agents are hidden from users, whereas connected agents may be visible in conversations depending on configuration.
Management and Scope	• Child agents managed within parents are lightweight; connected agents are independently managed and scalable.
Triggering and Flexibility	• Child agents rely on fixed triggers and priorities; connected agents are dynamically orchestrated at runtime.

COPILOT DEMO: CHILD AND CONNECTED AGENTS IN ACTION

Parent-Child Agent Structure

The Company Copilot acts as the parent agent managing a hidden child agent for content queries.

Peer Agent Connection

Policy Agent operates as a peer agent, enabling seamless query delegation.

Seamless User Experience

Employees experience one unified conversation while multiple agents collaborate behind the scenes to answer questions.

Concurrency

GUIDELINES FOR EFFECTIVE MULTI-AGENT IMPLEMENTATION

Define	**Define Clear Roles** • Each agent should have a clear domain to avoid overlaps and reduce orchestration confusion.
Write	**Write Strong Descriptions** • Precise and unique descriptions help orchestration select the appropriate agent effectively.
Limit	**Limit Child Agents** • Use child agents selectively to prevent conflicts and maintain parent agent efficiency.
Plan	**Plan for Fallback** • Fallback responses prevent dead ends by guiding users or escalating issues when needed.

Concurrency

Risks, Limitations, and Preview Status

Redirect Node Limitation	Redirect nodes cannot directly hand off to child agents, requiring alternative flow designs.
Citation Loss Risk	Child agents may lose citations when responding to parents; explicit citation inclusion is necessary.
Knowledge Leakage Concern	Child agents might invoke model knowledge even if disabled in parents; sensitive domains need monitoring.
Connection Limits	Agents cannot be connected in multiple places, restricting complex multi-hop chaining scenarios.
Preview Status Notice	Multi-agent features are in preview; expect changes requiring retesting and adjustments.

RECOMMENDATIONS FOR MULTI-AGENT ADOPTION

Concurrency

Child Agents for Specific Tasks

- Child agents handle narrow, hidden tasks, keeping the parent agent simple and focused.

Connected Agents for Collaboration

- Connected agents serve entry points for complex scenarios, external system.

Combining Models for Scalability

- Combining child and connected agents creates a modular, flexible system that scales like a specialist team.

Future-Proof and Adaptable Copilots

- This approach makes copilots easier to maintain, adapt, and expand as business needs evolve.

Resources

Video Walkthroughs and Demos

- "One Agent. Controls ALL Your Agents. Build it in Microsoft Copilot": https://www.youtube.com/watch?v=WKKdBC2zM3k
- "How To Use Multi-Agent in Copilot Studio" (Dewain Robinson): https://www.youtube.com/watch?v=eM8BpjyNk_Q
- "Multi-agent orchestration" (official Microsoft demo): https://www.youtube.com/watch?v=AagTqh1ctXU
- "Architecting multi-agent solutions with Copilot Studio and M365 Agents SDK": https://www.youtube.com/watch?v=pG01UDoM3xE
- "Learn to create Multi Agent Setup in Copilot Studio": https://www.youtube.com/watch?v=34GHQszppoU

Official Microsoft Documentation

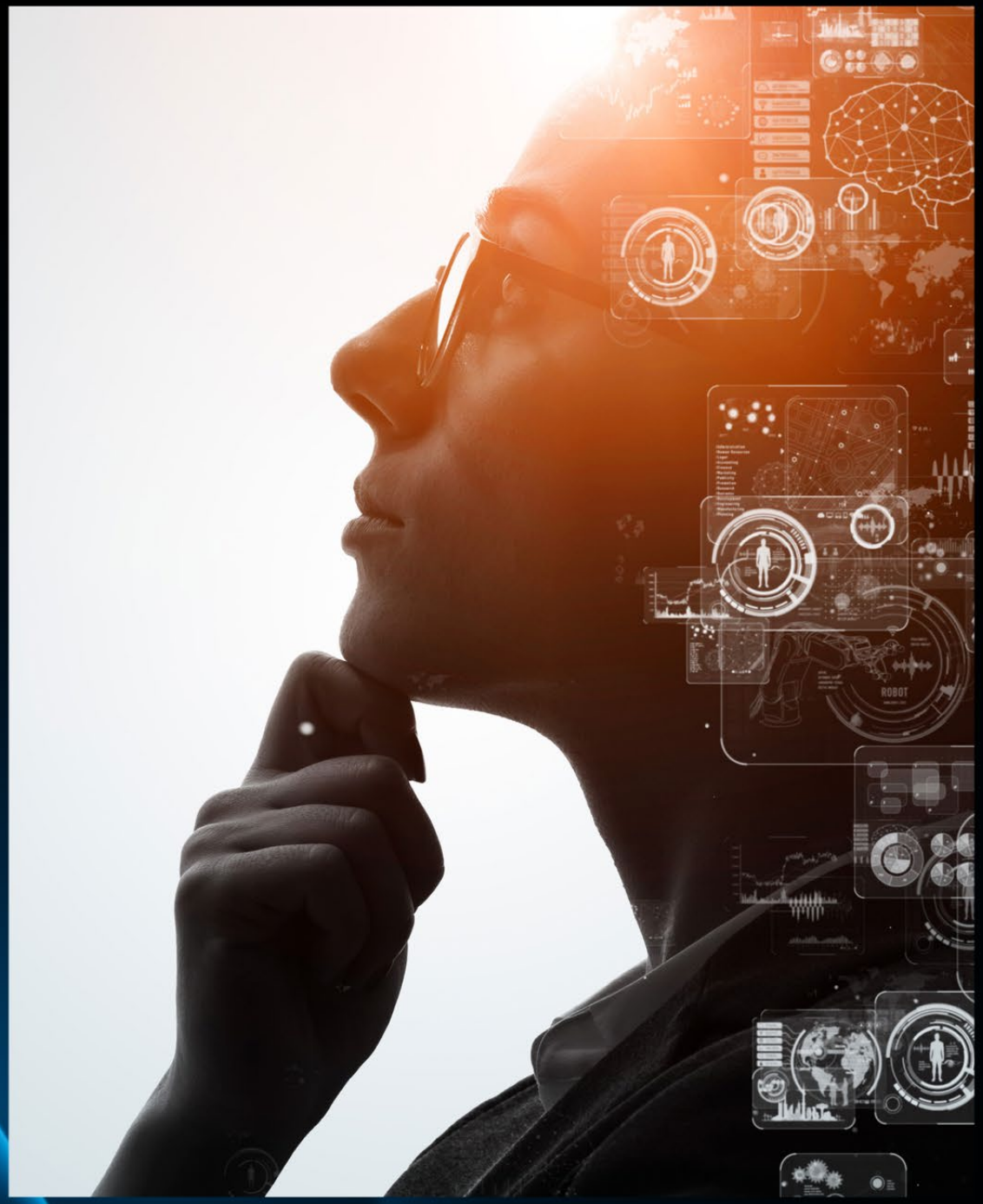
- Add other agents (preview): https://learn.microsoft.com/en-us/microsoft-copilot-studio/authoring-add-other-agents
- Generative orchestration and advanced actions: https://learn.microsoft.com/en-us/microsoft-copilot-studio/advanced-generative-actions
- Key concepts – Authoring copilots: https://learn.microsoft.com/en-us/microsoft-copilot-studio/authoring-fundamentals
- Export and import copilots using solutions: https://learn.microsoft.com/en-us/microsoft-copilot-studio/authoring-solutions-import-export
- Learning Path: Create and extend copilots with Copilot Studio: https://learn.microsoft.com/en-us/training/paths/create-extend-custom-copilots-microsoft-copilot-studio/

Concurrency

Copilot Studio & Multi-Agent Systems – Complimentary Session

- 30 minutes with our experts to learn how to build, orchestrate, and manage AI agents effectively, with live demos showcasing multi-agent collaboration.
- Get practical insights and answers to your specific questions.
- Schedule a follow-up — **we'll drop the bookings link in the chat during the session.**

Concurrency



Concurrency

Thank you!

Follow us on in for the latest IT insights and industry updates! @Concurrency, Inc