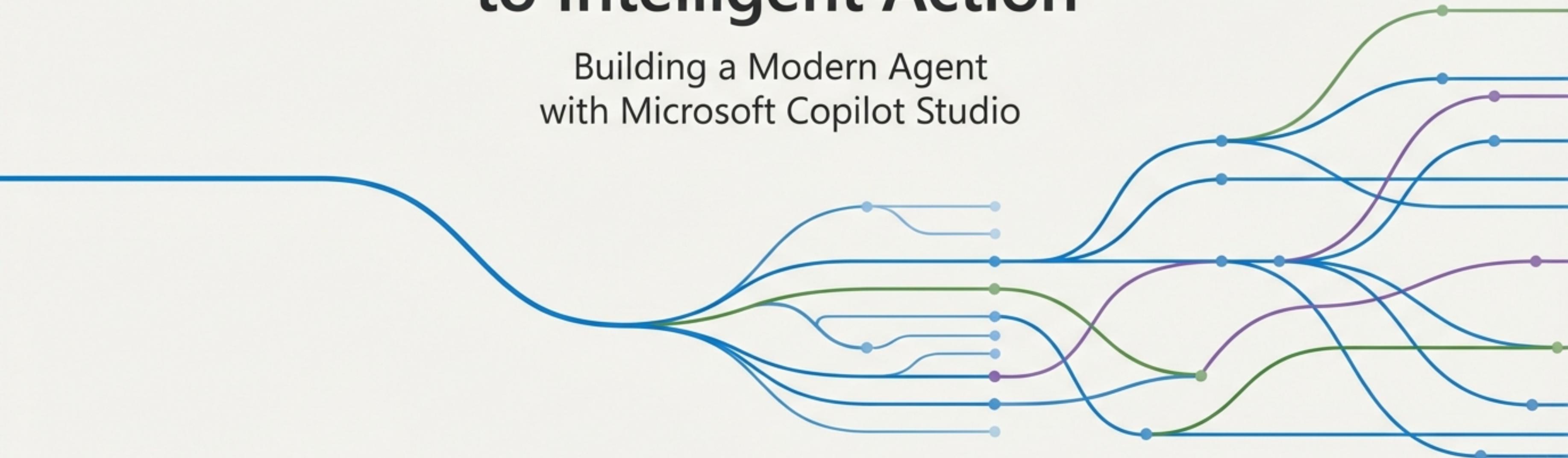


# From Simple Answers to Intelligent Action

Building a Modern Agent  
with Microsoft Copilot Studio



# The Need for Instant, Accurate Information

Employees who have quick access to information make better decisions, reduce errors, and save money.

## Our Challenge: The 'Event Contacts' Agent

For our upcoming sales events, we need a way to quickly find the correct event contact for a particular country/region. Manually searching documents or asking colleagues is slow and inefficient.



# A Low-Code Platform for Building Intelligent Agents

Microsoft Copilot Studio empowers your teams to create agents through a guided, no-code graphical interface, without needing to rely on data scientists or developers.

 **Empower Your Teams**  
Build agents without coding or AI expertise.

 **Reduce Costs**  
Automate common inquiries to free up human agents for complex issues.

 **Improve Satisfaction**  
Provide 24/7 self-help through personalized conversations.

Copilot Studio

Home Topics Entities Analytics Settings

Trigger → Message

Message

Question

Condition Action

Test

Welcome to the Copilot Studio. We coors commanion at the agent. Ask the omanence: are you to qunvilmnd fcnx your queries?

Type a queries...

NotebookLM

## Chapter 1: The Foundation

# Creating Our First Topic: The Static Q&A

### Core Concepts:

**Topics:** Define how agent conversations evolve.  
Each topic addresses a specific issue or task.

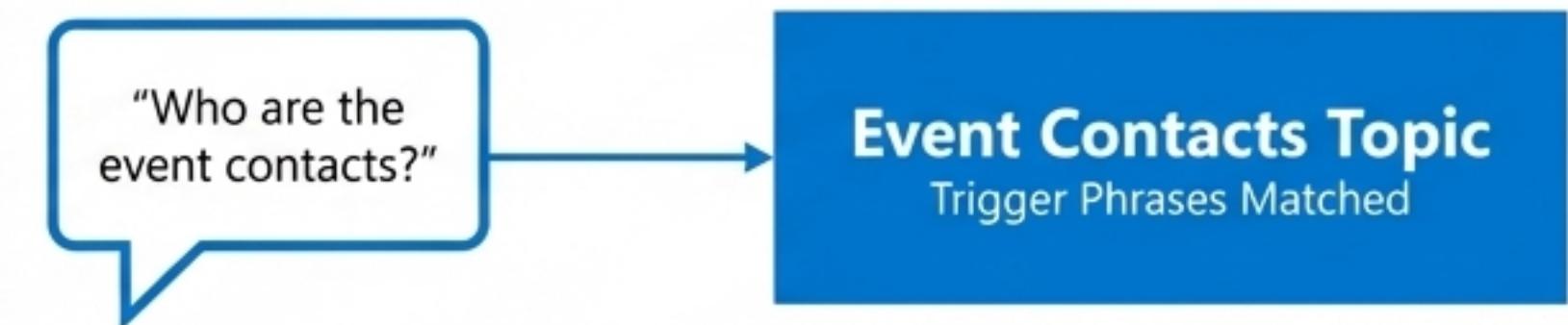
**Trigger Phrases:** Phrases or keywords that teach  
the agent when to use a specific topic. We need  
5-10 for good coverage.

### Our "Event Contacts" Topic - Version 1.0

**Topic Name:** 'Event Contacts'

#### Trigger Phrases:

- who are the event contacts
- event contact information
- who to contact for the event



# Version 1.0: A Simple, Hardcoded Response

The topic's conversation flow is built on the authoring canvas using Conversation Nodes.

Our first version uses just two nodes:

1. **Trigger Phrases:** The entry point.
2. **Show a message:** A static response.

The event team contacts for North America are:

## **USA**

Lynne Robbins [LynneR@contoso.com](mailto:LynneR@contoso.com)

## **Canada**

Lidia Holloway [Lidia@contoso.com](mailto:Lidia@contoso.com)

Trigger Phrases

Message

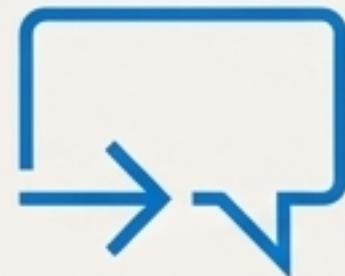
End of Conversation

# Moving from Static Text to a Dynamic Conversation

**The ‘Why’:** Our first agent is too rigid. It lists all contacts at once.  
We need to ask the user which country they need information for.

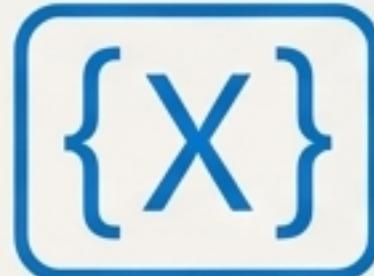
---

## The ‘What’ – Our New Tools



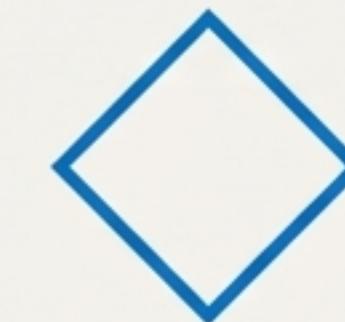
### Inputs

The user's response collected via an "Ask a question" node.



### Variables

A place to store the user's input for later use.



### Conditions

Branching logic that defines different paths based on a variable's value.

# How to Ask a Question and Branch the Logic

## Step-by-Step Implementation

### 1. Add an 'Ask a question' node

Replace the old message node.

- **Question Text:** 'I can help with event contacts. Please tell me which country/region the event is scheduled for.'
- **Identify:** Set to Multiple choice options.

### 2. Define Options

Add 'USA', 'Canada', 'France', and 'Spain' as user options.

### 3. Create a Variable

Under 'Save response as', create a new variable named 'VarCountry'. This stores the user's selection.

## The Result

Copilot Studio automatically creates '**Condition**' nodes for each option, branching the conversation path based on the value of '**VarCountry**'.

### Ask a question

I can help with event contacts. Please tell me which country/region the event is scheduled for.

Question Text

### Options for user

USA

Canada

France

Spain

Multiple Choice Options

### Save response as

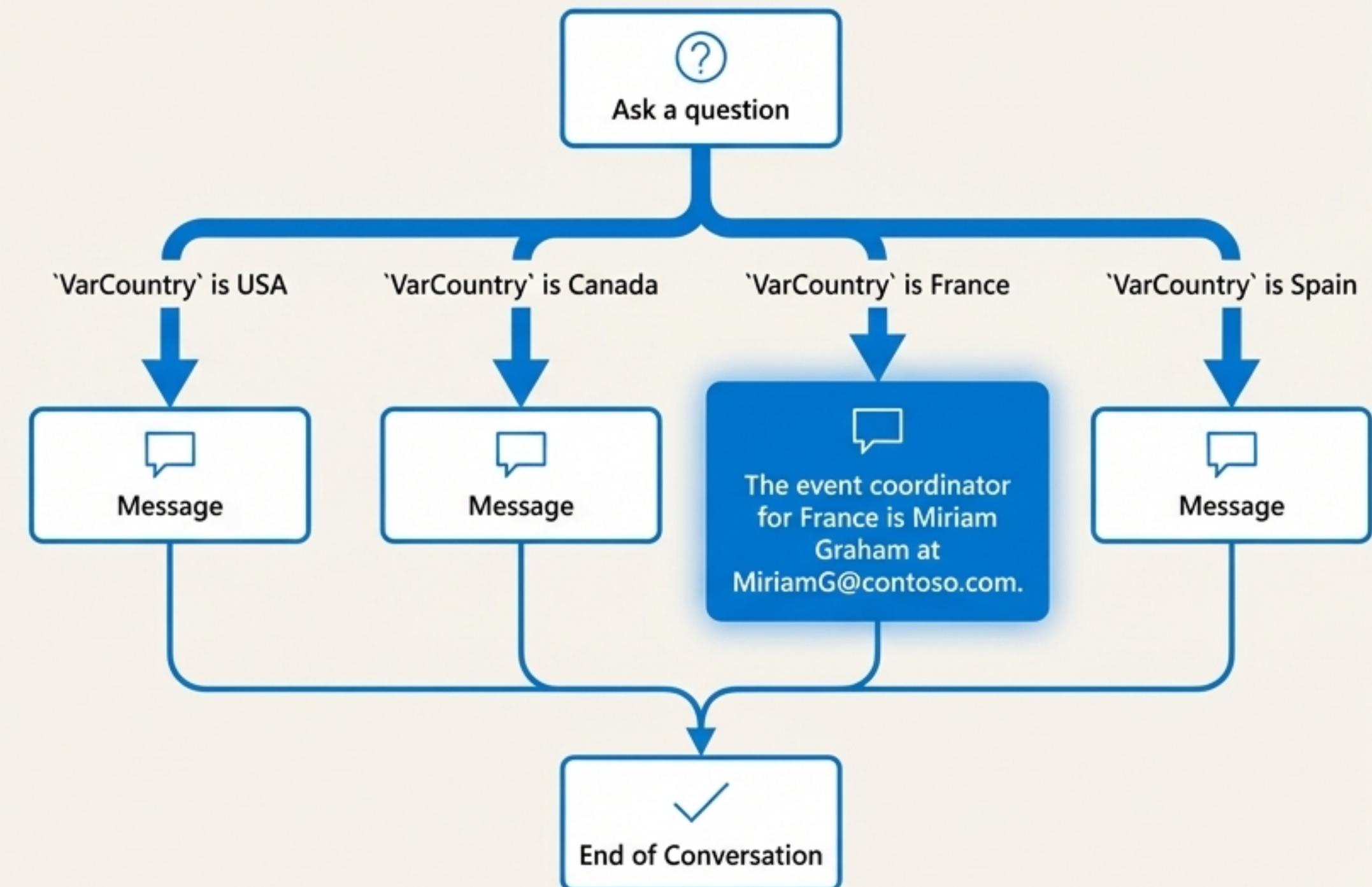
VarCountry

Save response as  
'VarCountry'

# Version 2.0: The Interactive, Branching Agent

## The New Conversation Flow

- The agent now asks for the user's country.
- Based on the 'VarCountry' variable, it follows a specific condition path.
- Each path has its own 'Message' node with the correct contact information.
- Example Branch for "France": The message node contains "The event coordinator for France is Miriam Graham at MiriamG@contoso.com."



# Chapter 3: Unlocking External Data From Hardcoded Logic to a Live, Scalable Data Source

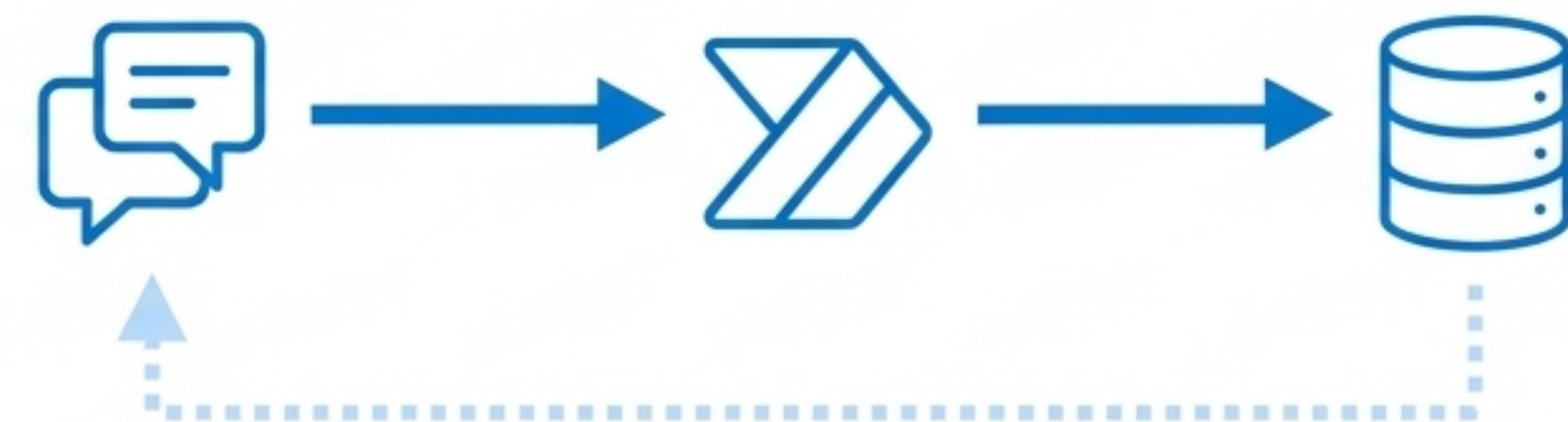
## The "Why"

Our agent is smarter, but the contact info is still manually entered inside the conversation logic. If a contact changes, a developer has to edit and republish the agent. This isn't scalable.

## The "What" - The Solution

We will connect our agent to a central data source.

- Dataverse for Teams:** A low-code database to store our contact list.
- Power Automate:** A workflow to fetch the data from Dataverse and pass it back to the agent.
- Call an Action Node:** The component in Copilot Studio that runs the Power Automate flow.



Live Data Request

# How to Connect the Agent to a Dataverse Table

## Step 1: Create the Data Source in Power Apps for Teams

- Create a new Dataverse 'Table' named 'Sales Project Team'.
- Add two columns: 'Name' (Text) and 'Email Address' (Email).
- Add rows for each contact (Joseph Price, Nathan Rigby, etc.).

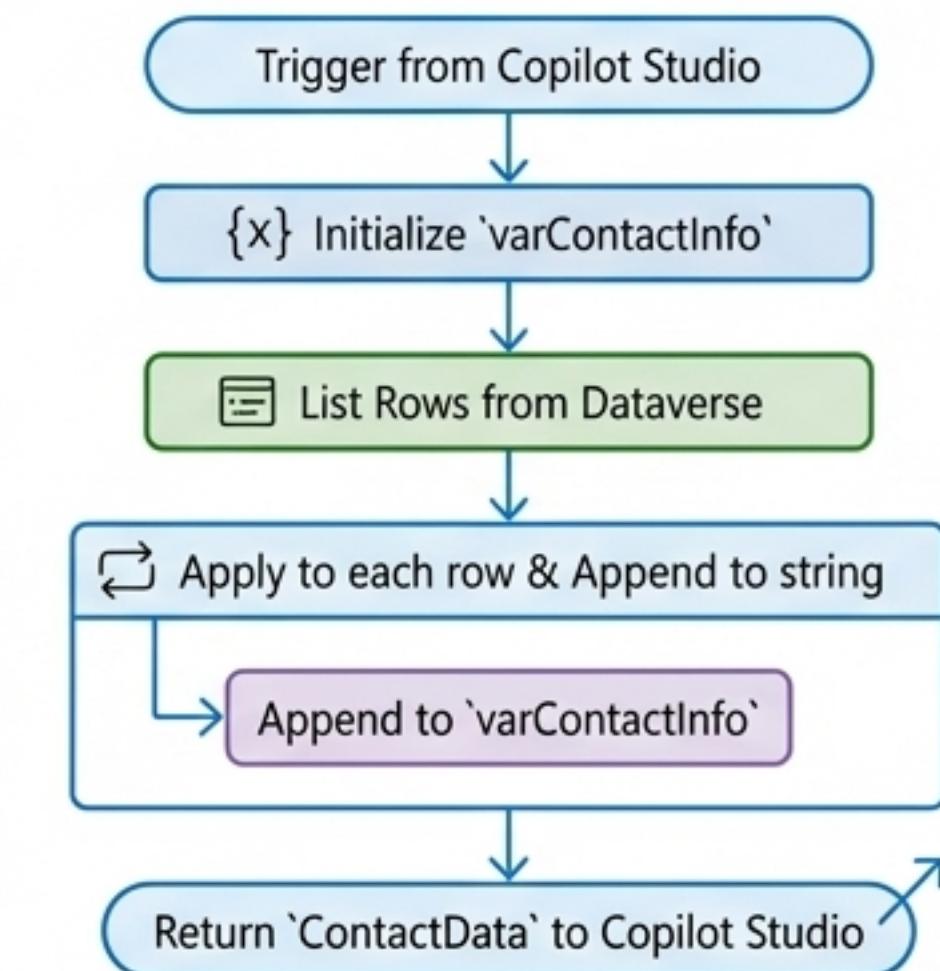
## Step 2: Build the Power Automate Flow

1. From a new topic, add a 'Call an action' node and select 'Create a flow'.
2. **Initialize Variable:** Create a string variable 'varContactInfo' to hold the results.
3. **List Rows:** Use the Microsoft Dataverse connector to get all rows from the 'Sales Project Team' table.
4. **Apply to each:** Loop through each row and 'Append' the Name and Email to the 'varContactInfo' string.
5. **Return value(s) to Copilot Studio:** Create a text output called 'ContactData' and assign it the value of 'varContactInfo'.

Sales Project Team

Name	Email Address
Joseph Price	JosephP@contoso.com
Nathan Rigby	NathanR@contoso.com
...	...

Power Automate Flow



# Chapter 4: Supercharging with Generative AI

## What Happens When There Isn't a Topic?

### The “Why”

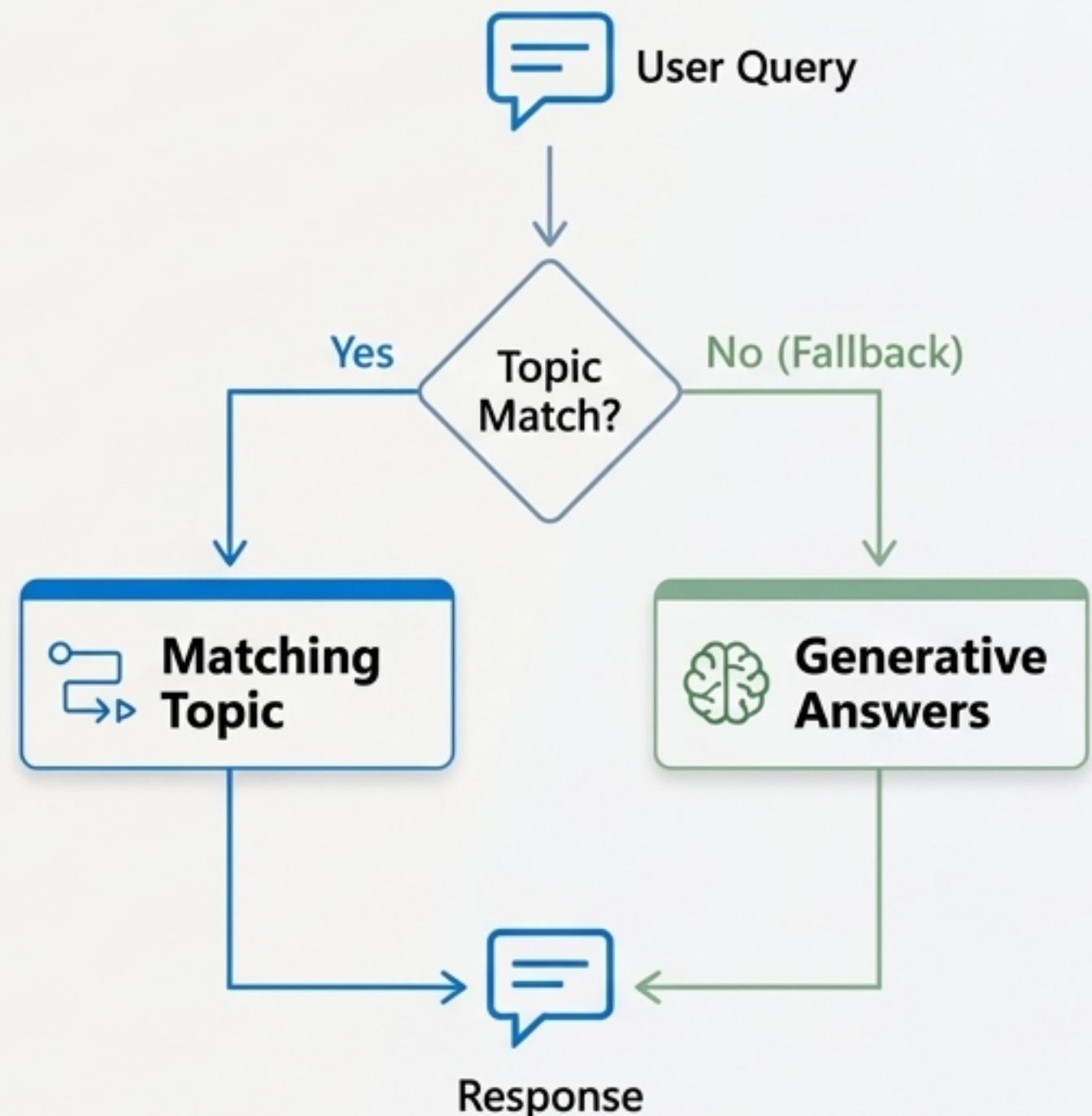
Previously, if an agent couldn't find a relevant topic after a few tries, it would escalate to a live agent. This requires manual intervention for un-authored questions.

### The “What” - Generative Answers

Microsoft Copilot Studio allows your agent to find and present information from multiple sources, internal or external, *without you having to create topics*.

### How it Works

Use Generative Answers as a fallback. When a user's query can't be matched to an existing topic, the agent uses AI to generate a response from your specified knowledge sources.



# Powering Generative Answers with Your Knowledge

## The Process

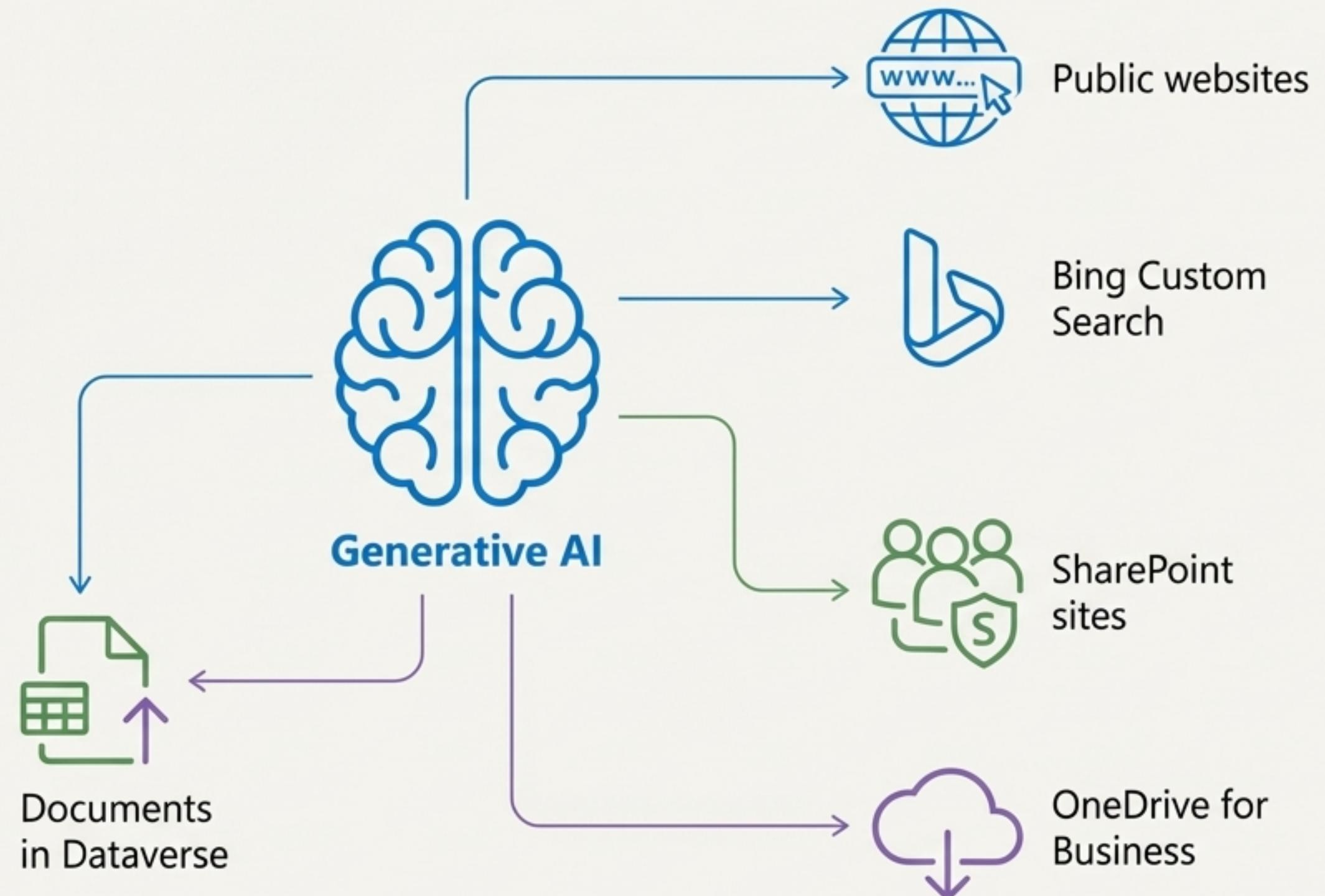
Enable Generative AI in the agent's settings to act as a fallback.

## Available Knowledge Sources

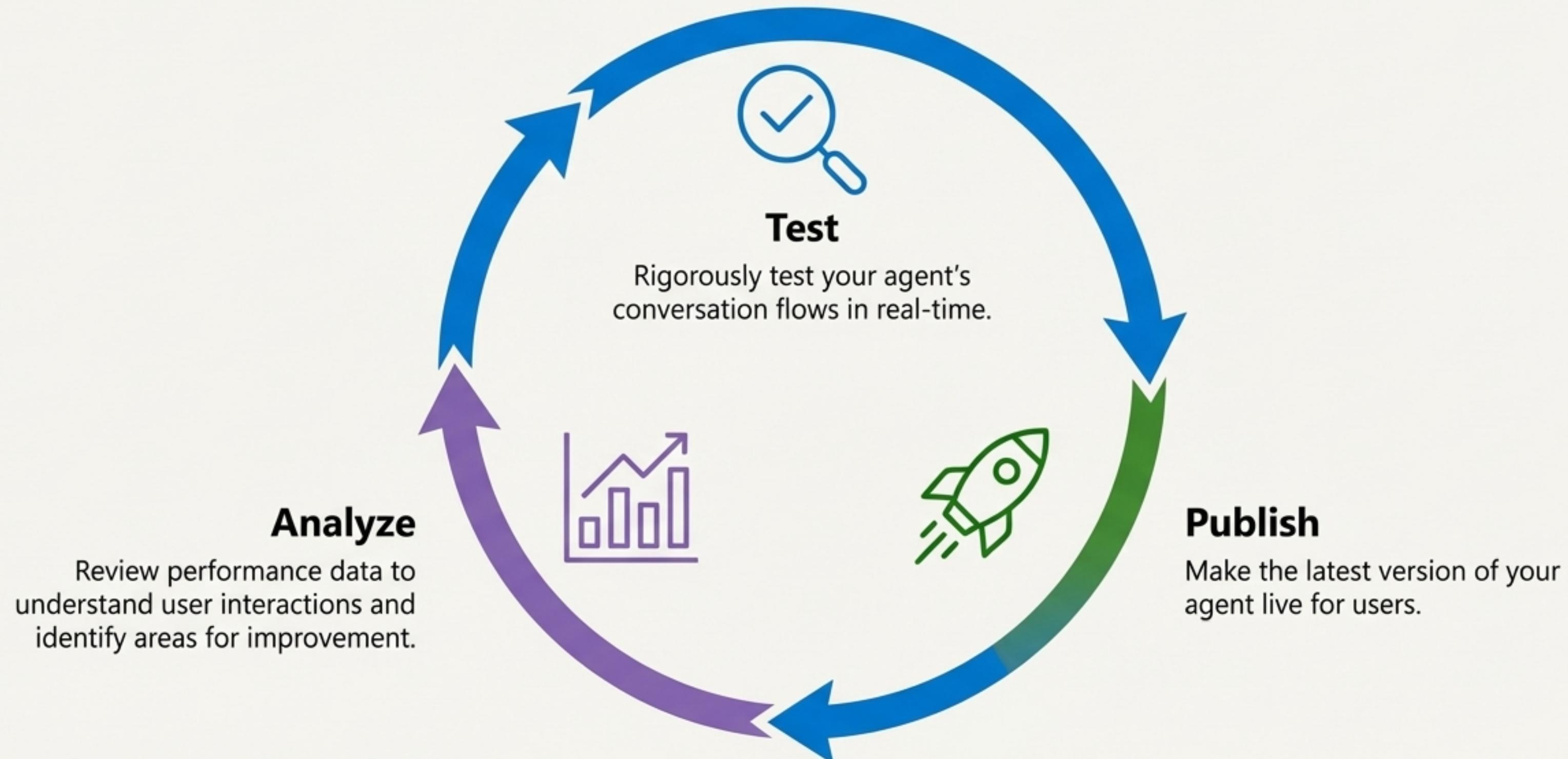
You can ground the AI in your own data to provide relevant, contextual answers.

## Key Consideration

The agent uses this 'grounding data' to provide context. Answers will be based on your data, not generic public information.

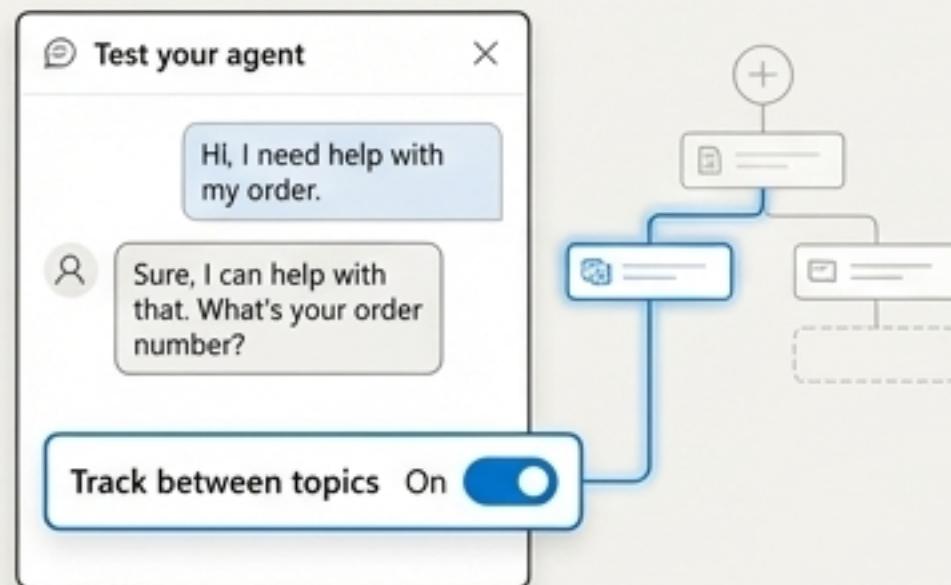


# Testing, Publishing, and Analyzing Performance



# The Go-Live Checklist

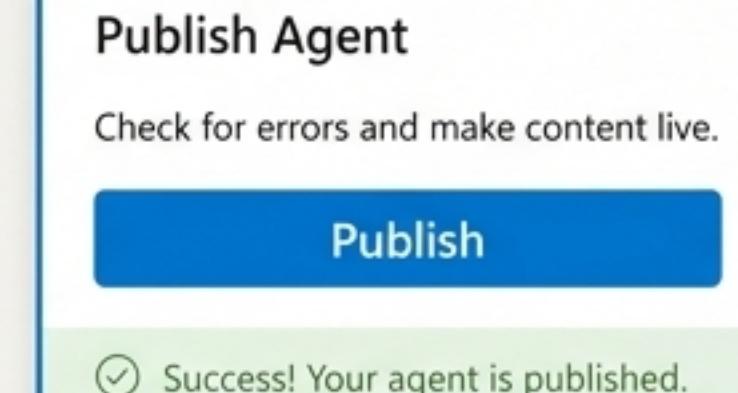
## Use the Test Panel



The built-in '**Test your agent**' panel lets you interact with the agent as a user would.

Enable '**Track between topics**' to see the conversation path light up on the authoring canvas, which is essential for debugging.

## Publish Your Agent

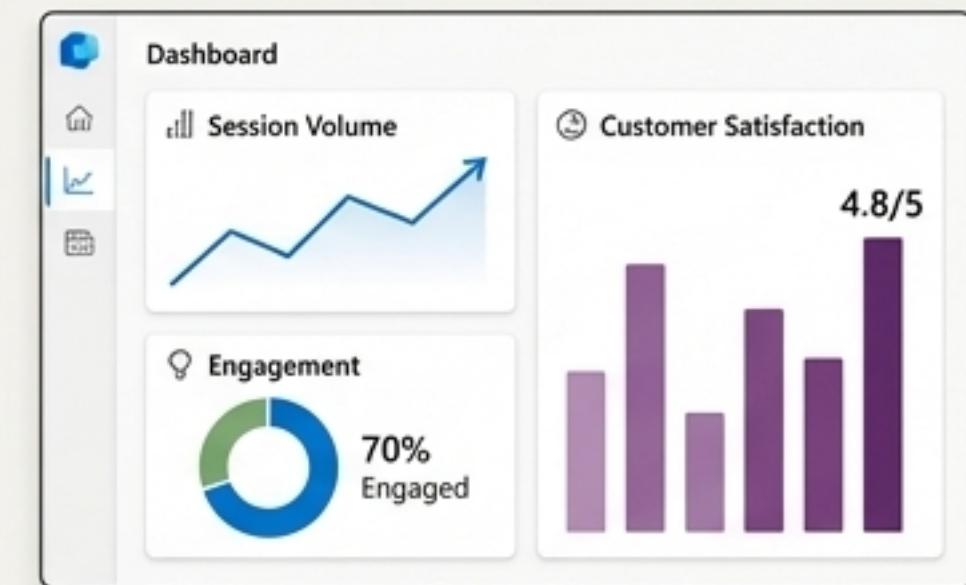


The '**Publish**' button checks the agent for errors and makes the latest content live.

### Share Options:

- **Demo website:** Share a link with stakeholders for feedback.
- **Microsoft Teams:** Publish for your team or submit for admin approval to make it available to the entire organization.

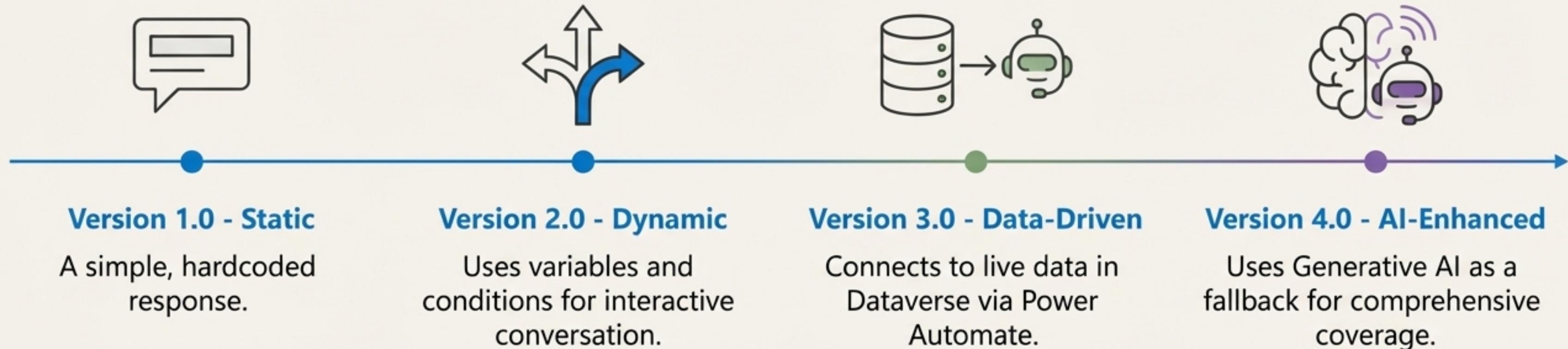
## Analyze Performance



The '**Analytics**' tab provides key performance indicators (KPIs) showing session volume, engagement, escalation rates, and customer satisfaction.

# The Journey Completed: From a Static Bot to an Intelligent Assistant

Using the graphical, low-code interface of Microsoft Copilot Studio, we transformed a simple Q&A agent into a powerful business tool.



You now have the foundational knowledge to start building your own intelligent **agents** to solve **real business** problems.