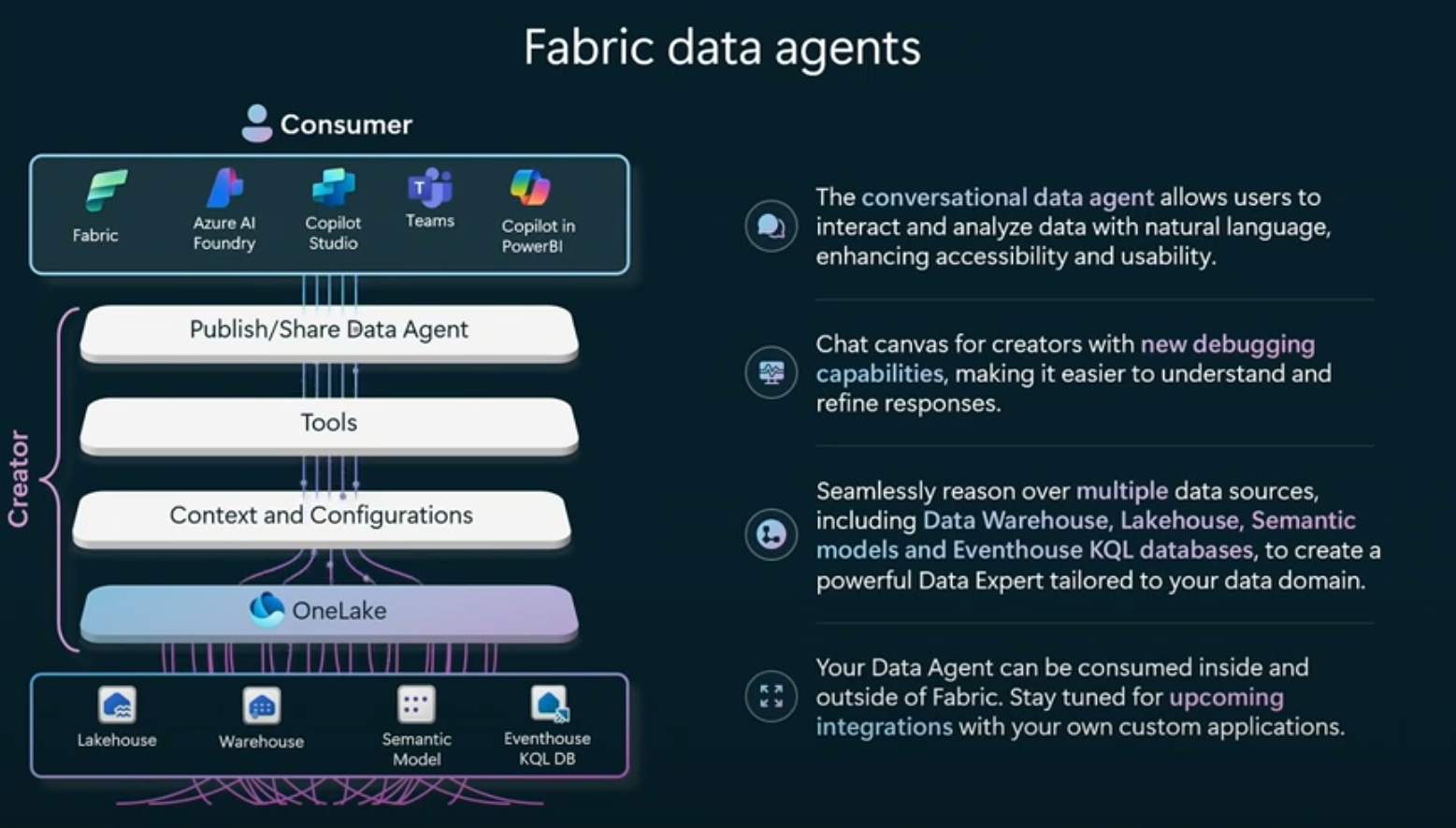
In this Blog, we are going to cover, Following

1. What is the Microsoft Fabric Data Agent?

The Microsoft Fabric Data Agent is a powerful tool that enables you to interact with structured enterprise data using natural language queries. Instead of writing complex SQL, DAX, or KQL statements, you can simply ask a question and get an instant, data-driven answer—whether your data lives in a Lakehouse, Warehouse, Power BI model, or KQL database. This not only speeds up analysis but also makes data insights accessible to a wider audience, from business analysts to executives, without requiring deep technical expertise. By configuring and tailoring the Data Agent to your organization’s needs, you can unlock self-service analytics while maintaining governance and security.

Getting started with Data agents - <https://learn.microsoft.com/en-us/fabric/data-agents/overview>



1. Fabric Copilot vs. Fabric Data Agent – Key Differences (comparison table)

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Fabric Copilot** | **Fabric Data Agent** |
| **Configuration** | Preconfigured and embedded | Fully configurable with custom instructions and examples |
| **Customization** | ❌ No control over behavior or prompts | ✅ Full control via prompt engineering and examples |
| **Use Case** | Embedded in Fabric experiences (e.g., reports, notebooks) | Standalone Q&A experience built by the user |
| **Data Source Support** | Uses active context (e.g., current report or notebook) | Supports multiple selected sources (Lakehouse, Warehouse, Power BI model, KQL DB) |
| **Context Injection** | ❌ Not supported | ✅ Supported (rules, definitions, examples) |
| **Query Generation** | Based on Fabric context | Uses NL2SQL, NL2DAX, NL2KQL for structured query generation |
| **Integration Scope** | Internal to Microsoft Fabric | Usable in Copilot Studio, Microsoft Teams, Azure AI Foundry |
| **Security Model** | Inherits user context within Fabric | Respects user credentials and configured data access |
| **Primary Purpose** | Task-specific assistance within Fabric workflows | Organization-wide natural language Q&A on structured data |
| **Prompt Engineering** | ❌ Not available | ✅ Available (add query/question pairs to guide behavior) |
| **Target Audience** | No-code / low-code users | Code-first users (Python SDK available) |

1. Key Highlights & Limitations

**Query Capability**

* Supports **read-only** queries in:
* SQL
* DAX
* KQL
* ❌ No support for **create/update/delete** operations

**🔸 Data Scope**

* Access limited to **configured structured sources**
* ❌ No support for **unstructured data** (e.g., PDF, DOCX)

**🔸 Query Complexity**

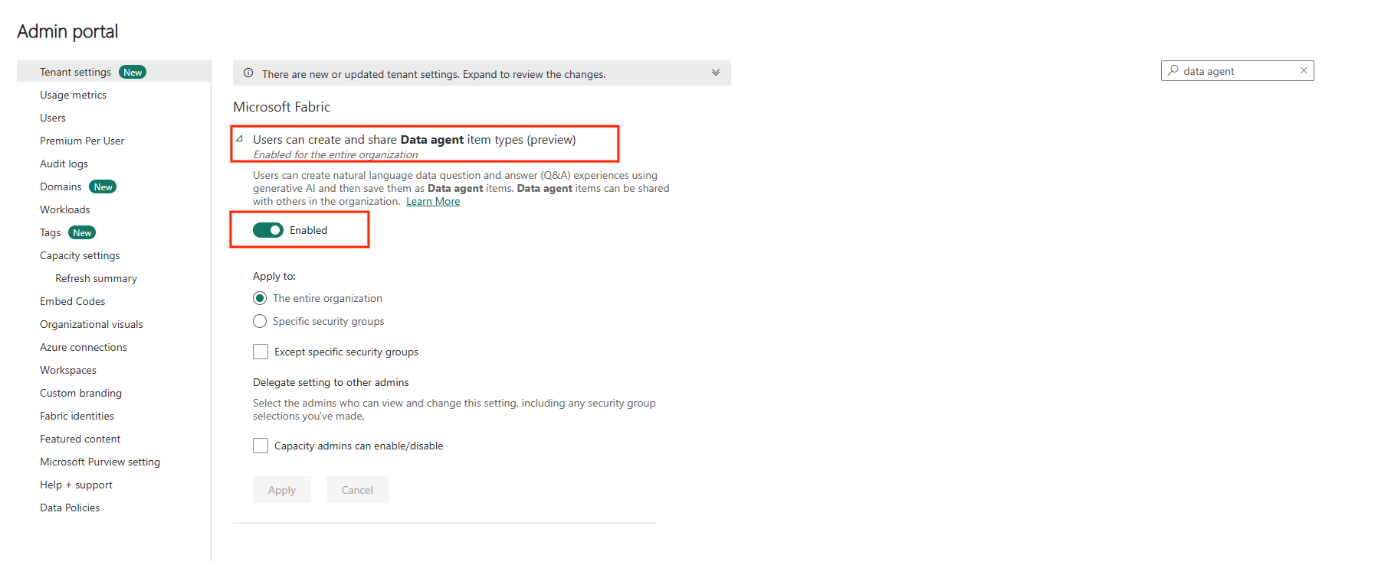
* Optimized for **simple queries**
* ⚠️ Lower reliability with **complex joins or logic**

🔵 **Language & LLM**

* **English only**
* LLM is **fixed** and cannot be changed

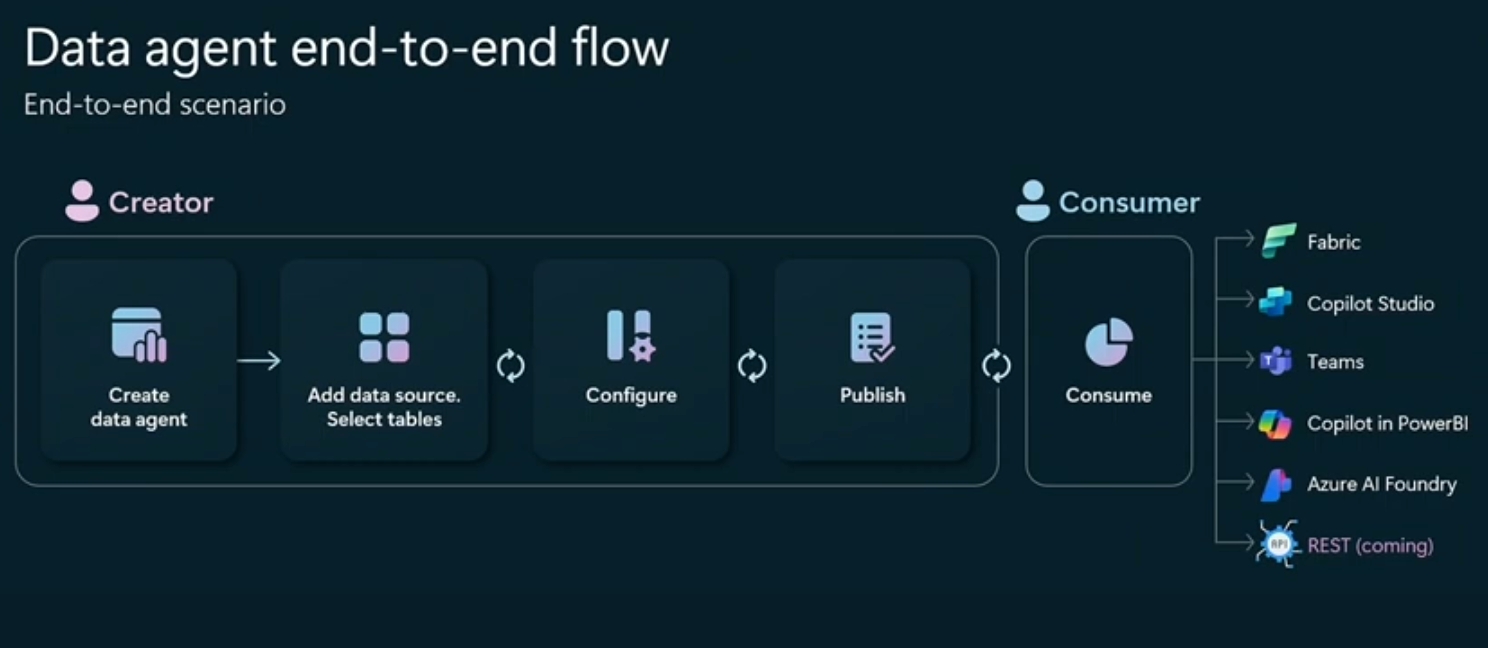
1. Prerequisites

* A paid Fabric capacity (F2 or higher)
* Fabric data agent tenant setting enabled
* Copilot tenant switch enabled
* Cross-geo processing and storing for AI enabled
* At least one of these data sources:
  + Lakehouse
  + Warehouse
  + Power BI semantic model
  + Kusto (KQL) database
* Power BI semantic models via XMLA endpoints switch enabled

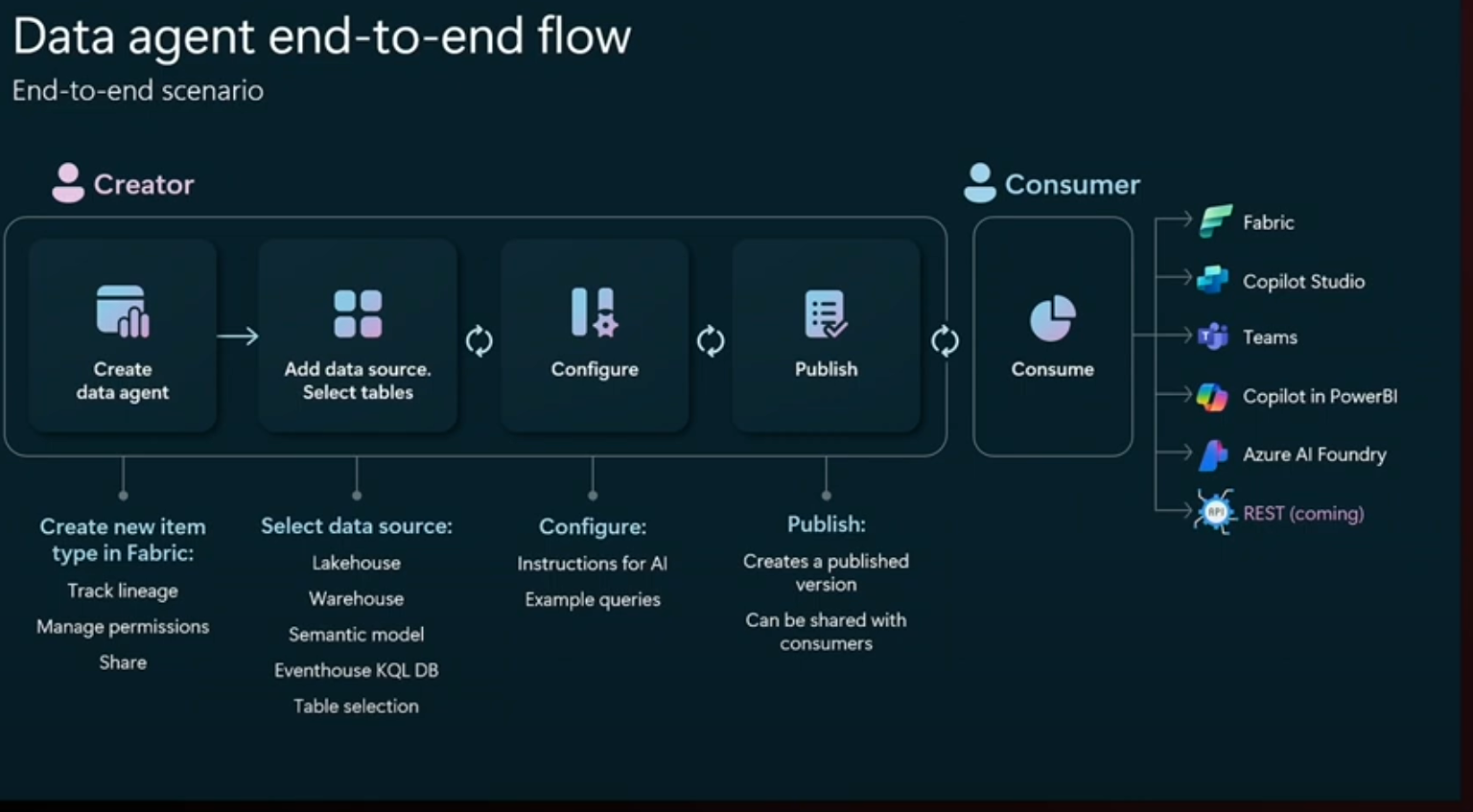


1. How to Create Your Data Agent (step-by-step)
   1. Open your Fabric workspace and click **+ New Item**.
   2. In the **All items** tab, search for and select **Fabric data agent**.
   3. When prompted, enter a name for your agent.
   4. After you name it, the OneLake catalog appears—this is where you add data sources.
   5. Select up to five sources (any mix of lakehouses, warehouses, semantic models, or KQL databases). Click **Add** for each.
   6. Use the filter icon to narrow down by type if needed.
   7. Once sources are added, the left-hand Explorer pane shows all tables. Tick the checkboxes next to tables you want the AI to access.

<https://learn.microsoft.com/en-us/fabric/data-science/how-to-create-data-agent>



https://www.youtube.com/watch?v=\_yraA8DUIxE



https://www.youtube.com/watch?v=\_yraA8DUIxE

1. How to Consume and Share Your Data Agent

<https://learn.microsoft.com/en-us/fabric/data-agents/consume-data-agent>

1. Integration Steps: Fabric Data Agent → Azure AI Foundry
2. Publish Your Fabric Data Agent

* In your Fabric workspace, build and configure your Data Agent.
* Click **Publish**.
* Copy the REST endpoint URL shown in the dialog; it follows the pattern:

<https://api.fabric.microsoft.com/v1/workspaces/{workspaceId}/agents/{artifactId}/operations/query?api-version=2024-03-31>

1. Extract Workspace ID & Artifact ID

From the published endpoint URL, note the two GUID segments:

**workspaceId** (your Fabric workspace’s unique ID)

**artifactId** (your Data Agent’s artifact ID)

You’ll supply these as custom keys when creating the connection.

1. Create a Microsoft Fabric Connection in Azure AI Foundry

In the Azure AI Foundry portal, navigate to **Management center** → **Connected resources**.

Click **+ New** and choose **Microsoft Fabric**.

Fill in:

**Workspace ID**: <Your **Workspace ID>**

**Artifact ID**: <Your **Artifact ID>**

**Optional** Base URL: use the same host as your endpoint (e.g., https://api.fabric.microsoft.com)

Click **Save** to register the connection.

1. Wire Up the Data Agent in Your Chat Agent

Open your Chat Agent project in Azure AI Foundry.

Under **Tools**, click **+ Add** and select **Microsoft Fabric Data Agent**.

Choose the Fabric connection you just created.

In your agent’s system instructions, reference the tool’s purpose (for example:

“For invoice-related questions, invoke the Fabric Data Agent.”)

1. Test and Publish

In the **Test** pane, run queries like “Show me invoice 12345.”

Verify that results come from your Data Agent via the workspace and agent IDs.

Once validated, click **Publish** on your Chat Agent to make it available.

1. Real-World Use Cases
   1. 1. Data Analysts – Quickly explore datasets and validate hypotheses without manually crafting queries. For example, an analyst could ask, “What were the top five product categories by revenue last quarter?” and instantly receive results ready for visualization.
   2. 2. BI Developers – Rapidly prototype and test metrics in semantic models by querying them directly in natural language before embedding results into dashboards or reports.
   3. 3. Operations Teams – Monitor performance metrics in near real time by connecting to operational data sources. A support manager might ask, “How many tickets were resolved in under 24 hours this week?” without opening a report builder.
   4. 4. Executives & Decision-Makers – Empower leadership to get quick answers on KPIs without waiting for report refresh cycles or ad-hoc analysis requests.
2. Tips & Best Practices (your list, formatted as callouts)

* Start Small – When working with large datasets, run smaller, targeted queries first to confirm accuracy before scaling to full datasets.
* Use Clear, Specific Language – Phrasing your question precisely (e.g., “monthly sales revenue in USD by region for 2024”) helps the Data Agent generate more accurate queries.
* Validate Results – Always cross-check a sample of answers against known reports or calculations to ensure accuracy.
* Leverage Context Injection – Add definitions, rules, or example questions to improve how the Data Agent interprets business terms.
* Optimize Data Sources – Ensure tables have clean, well-labeled columns and appropriate indexing to speed up query generation and execution.
* Stay Updated – Republish your agent when using the new Assistants API UI to avoid compatibility issues.

1. Next Steps & Resources (links to docs, videos, SDKs)
2. H**ow to Share?**

• Review your configuration to make sure the right tables and sources are included.

• Validate the agent, then share or publish it so colleagues can ask questions in natural language and get data-driven answers.

<https://learn.microsoft.com/en-us/fabric/data-science/data-agent-sharing>