



Data & AI

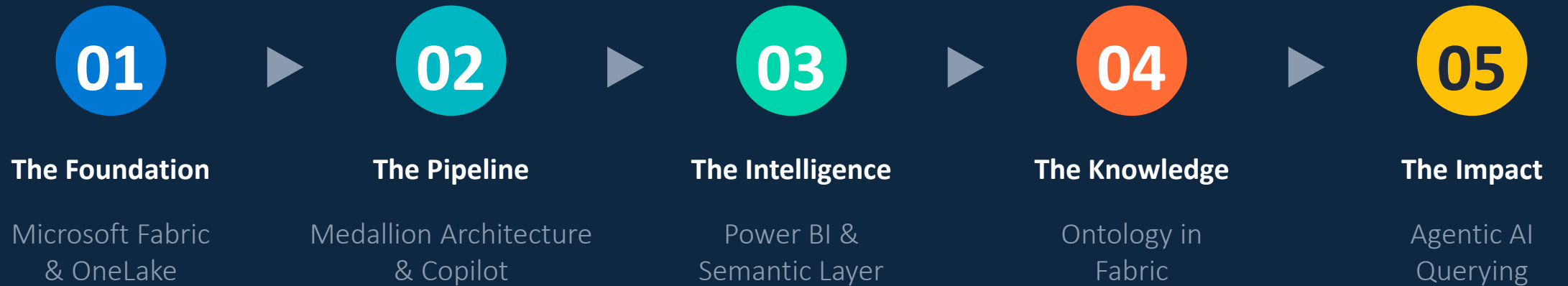
From Information to Impact

Great AI starts with trusted data—and great data delivers more value with AI.

Suneer Mehmood

Lead Data & AI Architect

Our Journey Today



Quick Overview

Foundation, Pipeline & BI – setting the stage

Deep Dive + Live Demos

Ontology & Agentic AI – where data gets democratized

The Data-AI Flywheel

Data and AI don't just coexist—they compound each other.

AI Enhances Data

Automates data preparation & cleaning
Discovers patterns humans miss
Scales quality across petabytes



Data Powers AI

Grounds AI responses in facts
Builds enterprise-specific context
Drives accurate, trusted decisions

Smarter Decisions

AI interprets data patterns
for strategic insights

Greater Efficiency

Automate repetitive tasks
across the data lifecycle

Measurable Results

Faster time-to-insight,
lower cost per analysis

Microsoft Fabric

One Platform. One Copy of Data. Every Workload.



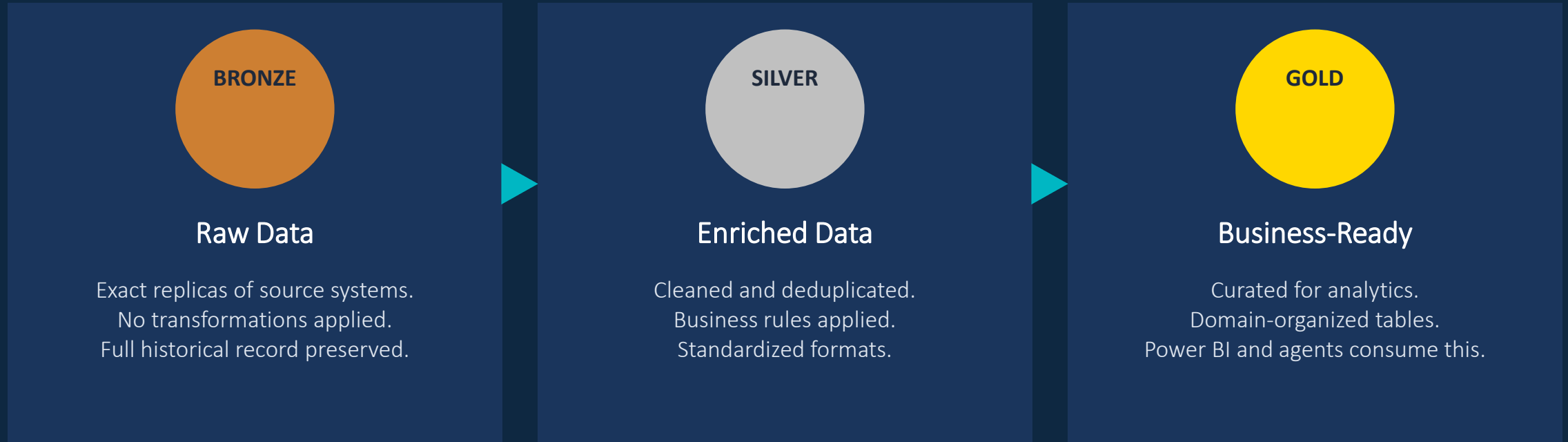
OneLake

A single, unified data lake for your entire organization. All services read from and write to the same data—no copies, no movement, no silos.

Built on open formats (Delta Lake, Parquet) • ACID compliant • Enterprise governance built-in

The Medallion Architecture

Incrementally refining data from raw to business-ready.



Each layer is a separate Lakehouse in OneLake • Incremental processing • Full audit trail from source to consumption

Copilot in Action: Data Engineering

AI-powered coding and troubleshooting in Fabric Notebooks

LIVE DEMO

Using Copilot in a Fabric Notebook to generate PySpark code, transform Bronze-to-Silver data, and troubleshoot errors in real-time.

Generate Code

Describe your transformation in plain English. Copilot writes the PySpark.

Troubleshoot

Paste an error message. Copilot explains the root cause and suggests fixes.

Optimize

Copilot reviews your code and suggests performance improvements.

“What used to take hours of debugging now takes seconds of conversation.”

Copilot in Action: Data Engineering

AI-powered coding and troubleshooting in Fabric Notebooks

The screenshot displays a Fabric Notebook interface. At the top, a search bar contains the text: "Load data from DimProducts.csv, clean up any nulls and write to a new Table Dim_Products in Silver LakeHouse". Below the search bar are buttons for "Accept", "Discard", and a refresh icon. A message states: "Changed 20 lines of code — AI can make mistakes. Check generated code before running it." The notebook editor shows two code cells. The left cell contains a red-shaded header with the text: "# Welcome to your new notebook" and "# Type here in the cell editor to add code!". The right cell contains a green-shaded code block with the following Python code:

```
1+import pyspark.sql.functions as F
2+
3+# Define file path for DimProducts.csv
4+csv_path = "/lakehouse/OntologyDataLH/Files/DimProducts.
5+
6+# Load data from CSV
7+df = spark.read.option("header", True).option("inferSche
8+
9+# Save as Delta Table to Silver layer
10+# Create Silver table path
11+silver_table_path = "/lakehouse/OntologyDataLH/Tables/Di
12+
13+# Write as Delta Table (Silver layer best-practice: over
14+df.write.format("delta").mode("overwrite").option("merge
15+
16+# Register as Delta Table
17+spark.sql(f"CREATE TABLE IF NOT EXISTS Dim_Products USIN
18+
19+# Display confirmation
20+print("DimProducts.csv loaded successfully into Silver t
```

At the bottom of the notebook, a status bar shows: "[1] ✓ 12 sec - Session ready in 11 sec 838 ms. Command executed in 406 ms by Suneer Mehmood on 2/26/2026, 7:18:11 AM". The language is set to "PySpark (Python)".

Copilot for SQL Analytics

Natural language to SQL — here's what you could ask your data.

HOW IT WORKS

Type a question in plain English → Copilot generates T-SQL → Results from your Silver & Gold Lakehouse tables. Read-only, schema-aware.

Sample Queries You Could Ask

Revenue Analysis

"Show me total revenue by product category for the last quarter"

Customer Insights

"Which customers in the Midwest have placed more than 5 orders?"

Trend Detection

"Compare month-over-month sales trends for our top 3 products"

Inventory & Performance

"What products have the highest return rate by region?"

Power BI & The Semantic Layer

The bridge between curated data and business decisions.

What the Semantic Model Does

- Defines business measures (Revenue, Profit, Margin)
- Creates relationships between tables (Star Schema)
- Applies row-level security per user/role
- Enables Direct Lake: query OneLake without import
- Single source of truth for all reports

Gold Layer (OneLake)



Semantic Model



Power BI Reports



Executive Dashboards

Most of you are familiar with Power BI. The key insight: with Fabric, the semantic model now sits directly on OneLake—no data movement needed.

What if every person in your organization could query data?

No SQL. No dashboards. Just a conversation.

FROM

Pre-built reports that answer yesterday's questions



TO

Conversational AI that answers tomorrow's questions

This requires two things: an Ontology to teach AI your business language, and Agents to have the conversation.

What Is an Ontology?

Teaching AI to speak your business language.

Think of It This Way

A database stores numbers in tables.

A semantic model defines calculations.

An ontology defines meaning.

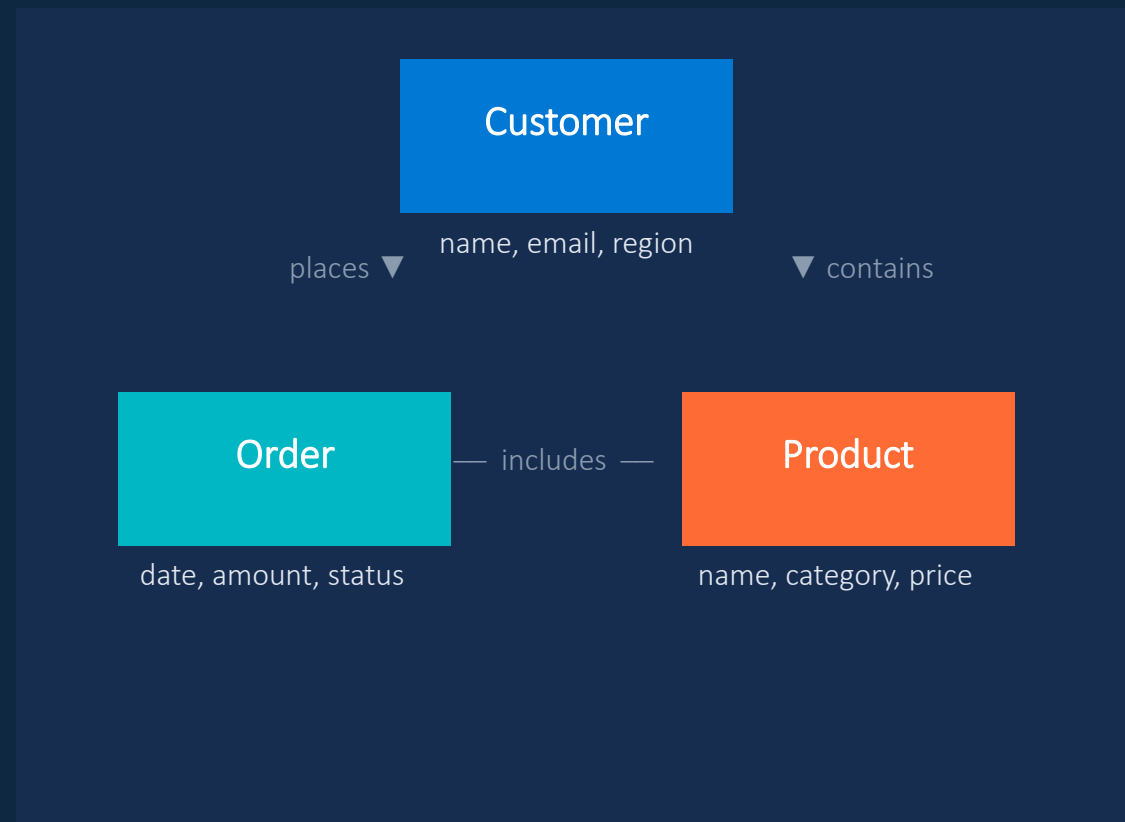
It captures: What is a 'Customer'?

What is a 'Product'? An 'Order'?

How do they connect to each other?

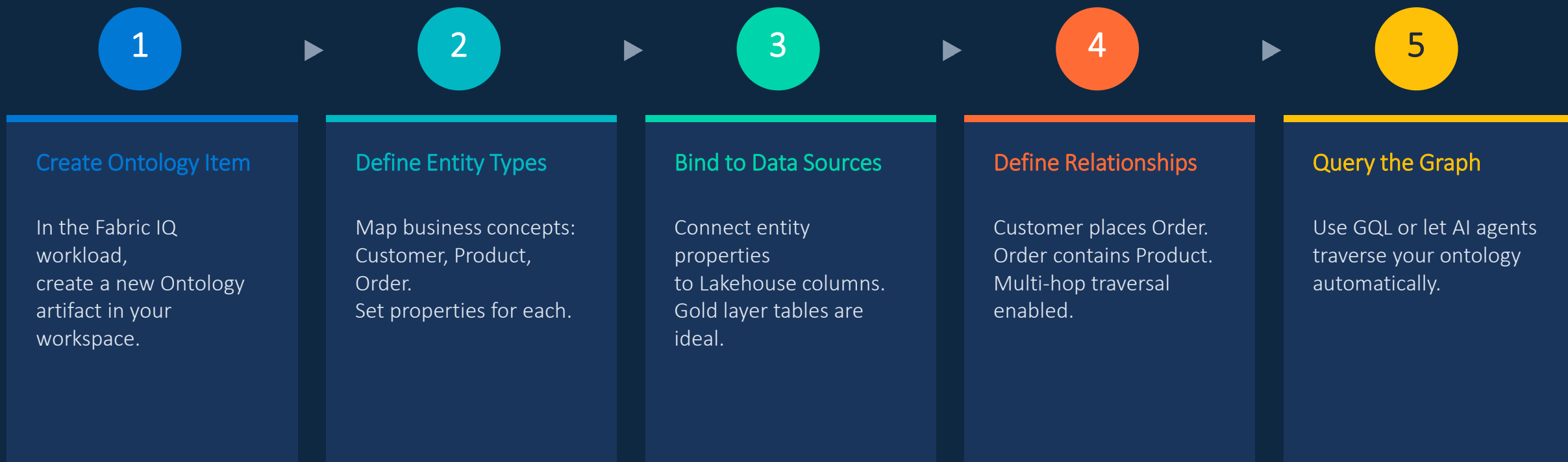
It's your enterprise vocabulary,

formalized for AI to understand.



Building an Ontology in Fabric

From Lakehouse tables to enterprise knowledge graph in minutes.



KEY INSIGHT:

The ontology doesn't duplicate your data. It adds a semantic layer of meaning on top of your existing Lakehouse tables.

LIVE DEMONSTRATION

Creating an Ontology in Microsoft Fabric

- Create ontology artifact in workspace
- Define entity types: Customer, Order, Product
- Bind properties to Gold Lakehouse columns
- Configure relationships between entities
- Preview the generated knowledge graph

LIVE DEMONSTRATION

Define entity types and bind properties to data

Add Entity Type ✕

Create an entity type that describes a concept in your ontology.

Entity type name *

Add Entity Type
Cancel

● Choose data source

○ Configure data binding

Choose data source

OntologyDataLH
CNCY Demo [Dev]

Select your table in the source above.

✔ Choose data source

● Configure data binding

Configure data binding

OntologyDataLH
CNCY Demo [Dev]

Binding type

Static Timeseries

Bind your properties

Static * ⓘ

Source column	Property name
<input type="text" value="ProductId"/>	<input type="text" value="ProductId"/>
<input type="text" value="ProductName"/>	<input type="text" value="ProductName"/>
<input type="text" value="Category"/>	<input type="text" value="Category"/>
<input type="text" value="Subcategory"/>	<input type="text" value="Subcategory"/>
<input type="text" value="Brand"/>	<input type="text" value="Brand"/>

+ Add static property

Entity Type Keys

A key is a unique identification property or sequence of properties that we use to reliably and securely connect your downstream data to this entity type. Note: the sequence you create the properties will be how it is read. You can edit the key at any time.

- Brand
- Category
- ProductId
- ProductName
- Subcategory

Save
Close

LIVE DEMONSTRATION

Configure relationships between entities and preview your Graph

Relationship configuration

Relationship type name * ⓘ

Source data ⓘ
Workspace *

Lakehouse *

Schema *

Table *

Define the relationship between entity types by binding columns from your selected source to the key defined on each entity type.

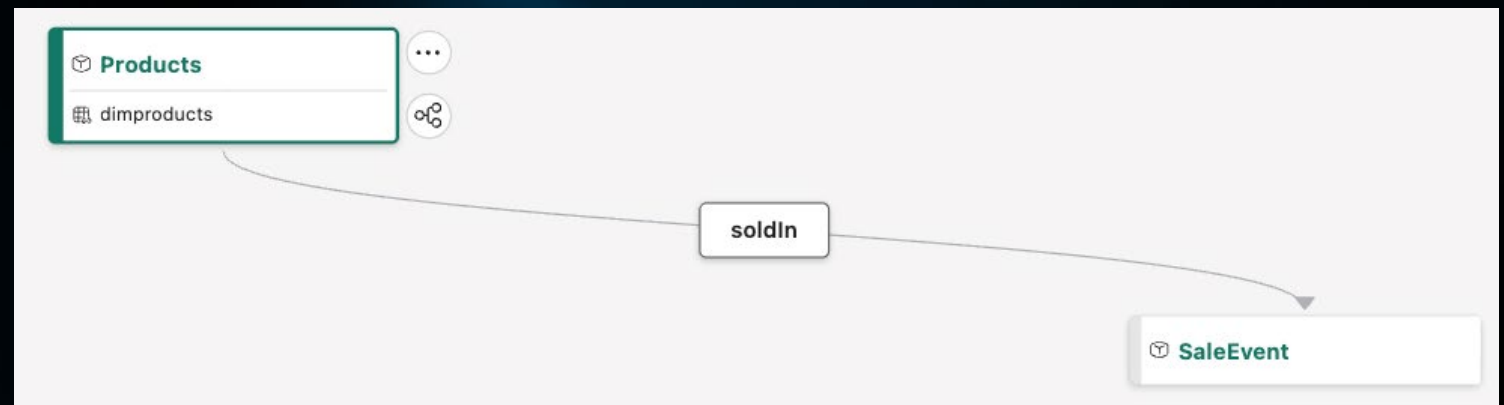
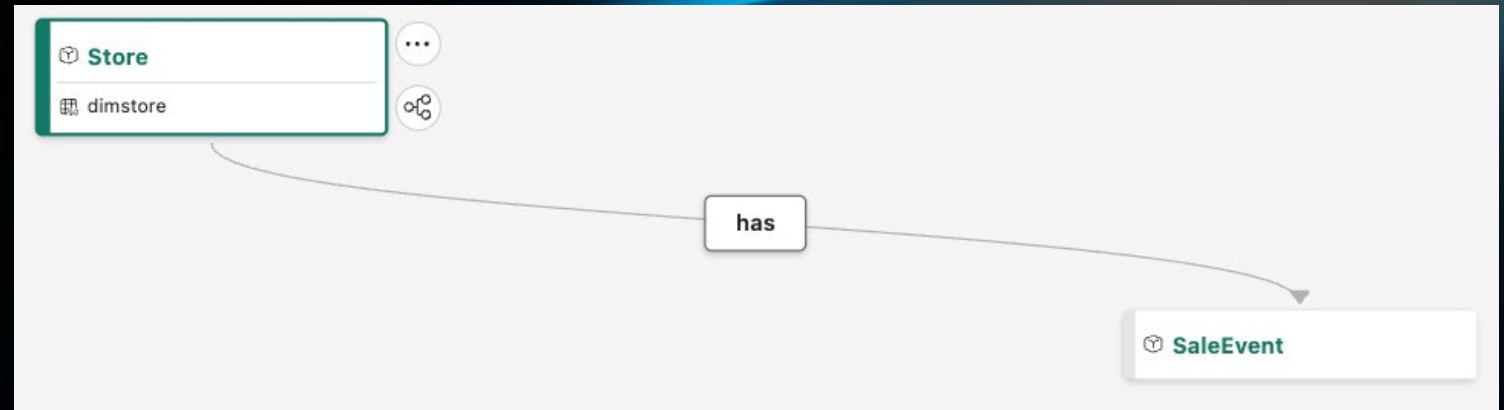
1. Source entity type *

Source column Entity type key property
 ⇌ StoreId

Links to

2. Target entity type *

Source column Entity type key property
 ⇌ SaleId



LIVE DEMONSTRATION

Configure relationships between entities and preview your Graph

The screenshot shows the Neo4j Query Builder interface. At the top, there are tabs for 'Query builder', 'Run query', 'Clear query', 'Add filter', 'Remove filter', 'Reset filters', and 'Diagram view'. The 'Query' section shows a graph with two nodes: 'Store' (orange circle) and 'SaleEvent' (purple circle), connected by an edge labeled 'has'. A filter is applied to the 'SaleEvent' node: 'RevenueUSD > 150'. The 'Nodes (4)' list includes 'Store', 'Products', 'SaleEvent', and 'Freezer'. The 'Edges (3)' list includes 'has', 'soldin', and 'operates'. Below the graph, there is a search bar and a zoom control. The bottom part of the interface shows a search result for 'SaleEvent: P-ICE-001' with its relationships to 'Paris' and 'Amsterdam'.

This close-up shows the filter configuration for the 'SaleEvent' node. It includes a 'Remove filter' button, a 'Reset filters' button, and a text input field containing 'RevenueUSD > 150' with a trash icon to its right.

```

MATCH (node_Store:`Store`)-[edge1_has:`has`]->(node_SaleEvent:`SaleEvent`)
WHERE node_SaleEvent.`RevenueUSD` > 150
RETURN TO_JSON_STRING(node_Store) AS `Store`,
       TO_JSON_STRING(edge1_has) AS `edge_1_has`,
       TO_JSON_STRING(node_SaleEvent) AS `SaleEvent`
LIMIT 1000
    
```



The Agentic AI Revolution

From dashboards you read to agents that answer.

What Changes With Agents?

Before

Users navigate to dashboards, apply filters, interpret charts, and draw their own conclusions.

After

Users ask questions in plain English. The agent queries data, reasons over results, and delivers insights.

Fabric Data Agent Capabilities

Natural Language

Ask in English, get answers from your enterprise data

Multi-Source

Queries across Lakehouses, Warehouses, and Ontology

Read-Only Safe

Enforces read-only access. No accidental data changes

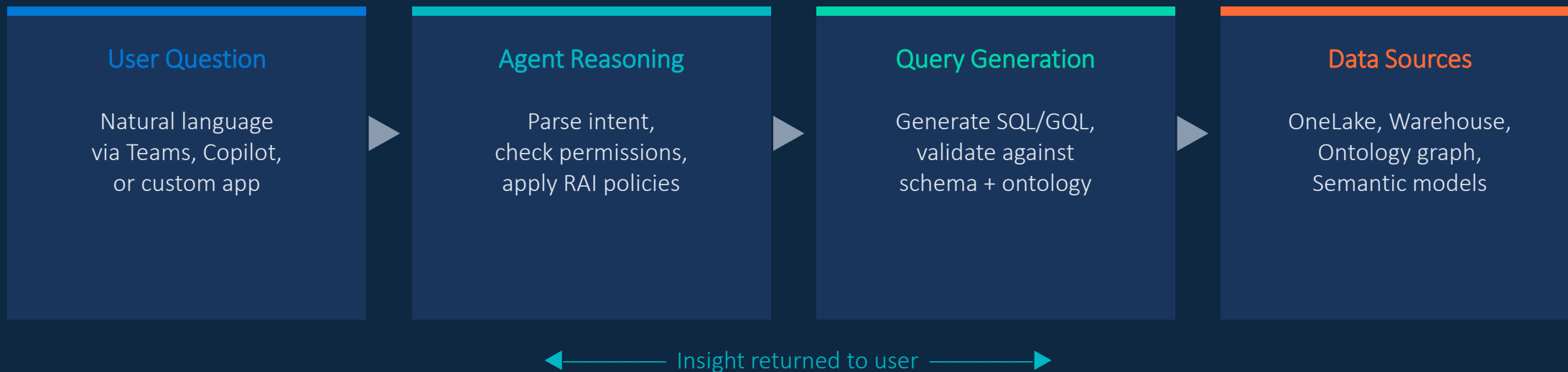
Multi-Channel

Deploy to Teams, Copilot Studio, or custom apps

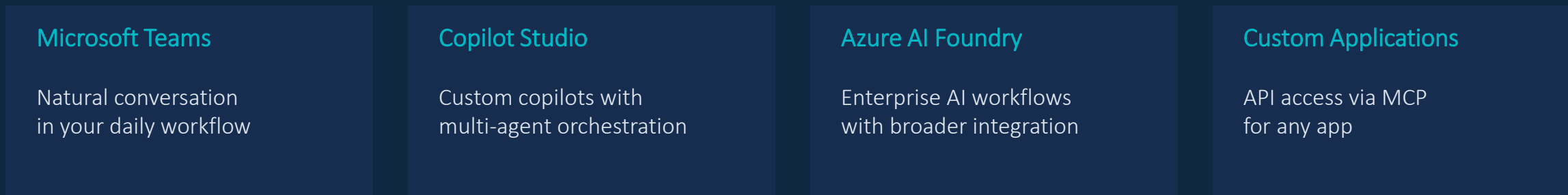
2026 is the Year of the Agent

MCP interoperability • Ontology-aware reasoning • Deeper data analysis • Announced at Microsoft Ignite 2025

How Fabric Data Agents Work



Deployment Channels



LIVE DEMONSTRATION

Agentic Revolution – Talk to your data

What is the top product by revenue across all stores?

The top product by revenue across all stores is Classic Vanilla Pint, with a total revenue of \$465 USD.



🕒 Response time: 3 sec

Break that down by region

Here is the revenue for Classic Vanilla Pint broken down by region:

- France: \$170 USD
- Germany: \$140 USD
- Netherlands: \$155 USD

LIVE DEMONSTRATION

Agentic Revolution – Talk to your data

Which product is our MVP — most valuable product — and by how much does it beat the runner-up?

The MVP (most valuable product) is Classic Vanilla Pint, with a total revenue of \$465 USD. It beats the runner-up, Dark Chocolate Pint (with \$282 USD), by \$183 USD.

👍 👎

1 step completed ⌵ 🕒 Response time: 15 sec

"If each region were a team competing for the highest sales trophy, who wins?"

If each region were a team competing for the highest sales trophy, France wins with the highest total revenue of \$302 USD.

👍 👎

1 step completed ⌵ 🕒 Response time: 12 sec

Which store is our hidden gem — the one with the highest revenue per product?

The hidden gem store, with the highest revenue per product, is Lakeshore Paris 1. It has a total revenue of \$302 USD from 2 unique products, resulting in \$151 revenue per product.

LIVE DEMONSTRATION

Agentic Revolution – Talk to your data

Which store is our hidden gem — the one with the highest revenue per product?

The hidden gem store, with the highest revenue per product, is Lakeshore Paris 1. It has a total revenue of \$302 USD from 2 unique products, resulting in \$151 revenue per product.

👍 🗨️

1 step completed ^ ⌚ Response time: 16 sec

✔️ Analyzed RetailSalesOntology Ontology for: "For each store, calculate the total revenue divided by the number of unique products sold to find the revenue per product. Identify the store with the highest revenue per product, and provide the store name, total revenue, number of products, and revenue per product."

</> Query code

```
Ontology
{"entitySelector":{"query":"MATCH (node_Store:`Store`)-[:`has`]->
(node_SaleEvent:`SaleEvent`)\nRETURN node_Store.`StoreId` AS `store_id`, node_Store.`StoreName` AS `store_name`, SUM(node_SaleEvent.`RevenueUSD`) AS `total_revenue`, COUNT(DISTINCT node_SaleEvent.`ProductId`) AS `num_products`, SUM(node_SaleEvent.`RevenueUSD`) / COUNT(DISTINCT node_SaleEvent.`ProductId`) AS `revenue_per_product`\nGROUP BY store_id, store_name\nORDER BY revenue_per_product DESC\nLIMIT 1","queryType":"GQL"}}
```

🗃️ Query output

store_id	store_name	total_revenue	num_products	revenue_per_product
S-PAR-01	Lakeshore Paris 1	302	2	151

LIVE DEMONSTRATION

Querying Data in Agentic Mode

Ask your data anything. In plain English.

Sample Questions We'll Ask:

“Which product categories drove the most revenue last quarter?”

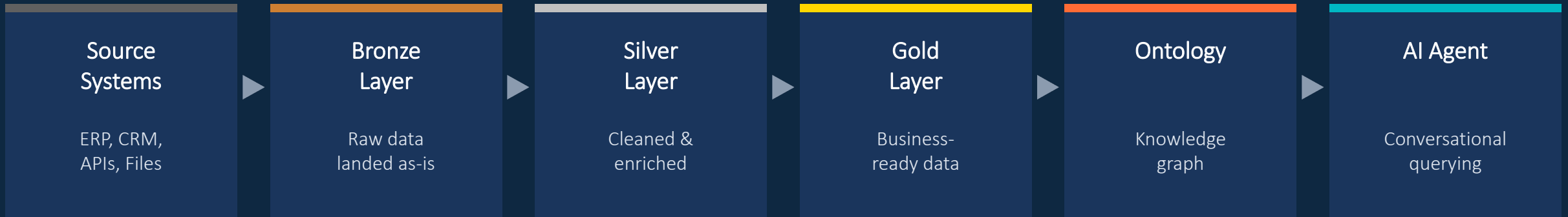
“Show me customers with declining order frequency in the Midwest.”

“What's the relationship between our premium products and customer retention?”

“Compare Q3 vs Q4 performance across all regions.”

From Information to Impact

The end-to-end journey we've just walked through.



COPILLOT AI — Assisting at Every Stage: Code Generation • SQL Writing • Troubleshooting • Query Optimization

ONELAKE — Single Data Lake: One Copy of Data • Open Formats • Enterprise Governance • ACID Compliance

The Impact

Democratized Access

Every employee can query data

Faster Decisions

Seconds, not days to insight

Trusted Results

Grounded in governed data

Lower Cost

One platform, zero duplication

Real Business Impact

40-60%

Reduction in
Time-to-Insight

Unified platform eliminates
tool-switching delays

80%

Fewer Ad-Hoc
Data Requests

Self-service querying reduces
data team ticket burden

1

Single Source
of Truth

End the “my numbers
don't match” debates

24 / 26

Always-On
Data Access

Agents answer questions
whenever they arise

The Question Isn't Whether to Start—It's How Fast You Can Move

Organizations that build this data-AI flywheel today will compound their advantage every quarter.

Key Takeaways

- 1** | **The Data-AI Flywheel Is Real**
AI and data reinforce each other. Invest in both simultaneously, not sequentially.
- 2** | **Unified Platforms Win**
Microsoft Fabric eliminates silos with OneLake. One copy of data, every workload, one governance model.
- 3** | **Ontology Is the Game-Changer**
Teaching AI your business language through an ontology transforms what's possible with data querying.
- 4** | **Agents Democratize Data**
When anyone can ask a question and get a trusted answer, data stops being a bottleneck and becomes an accelerator.
- 5** | **Start Now, Compound Later**
The sooner you build trusted data foundations, the faster AI delivers compounding returns.



Thank You

Questions & Discussion

Suneer Mehmood

Lead Data & AI Architect

smehmood@concurrency.com

Milwaukee AI Summit 2026