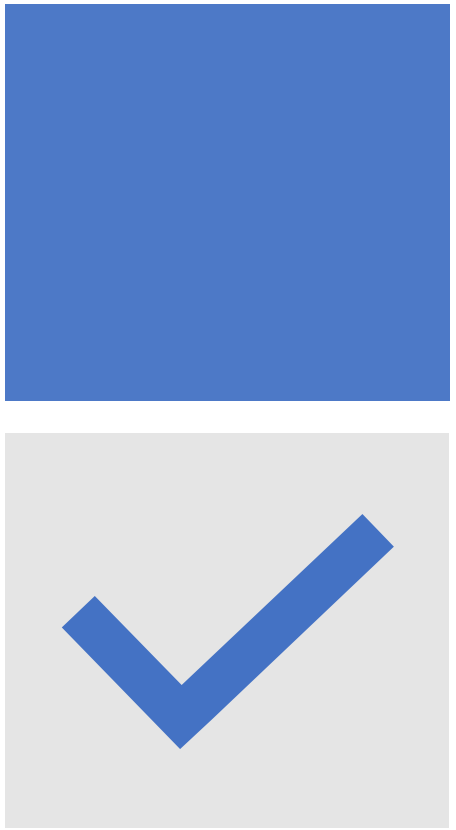
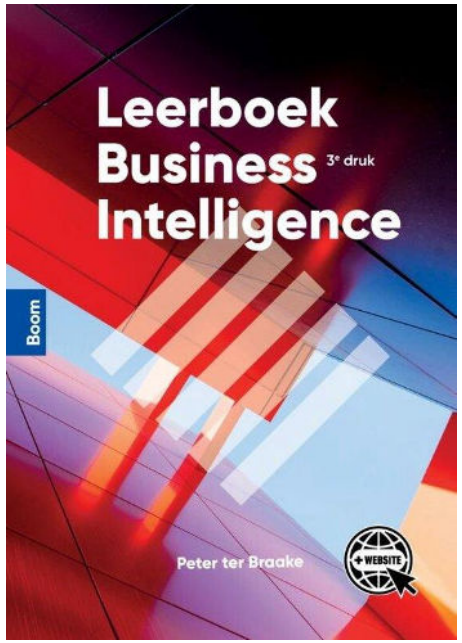
Fabric in the Age of AI

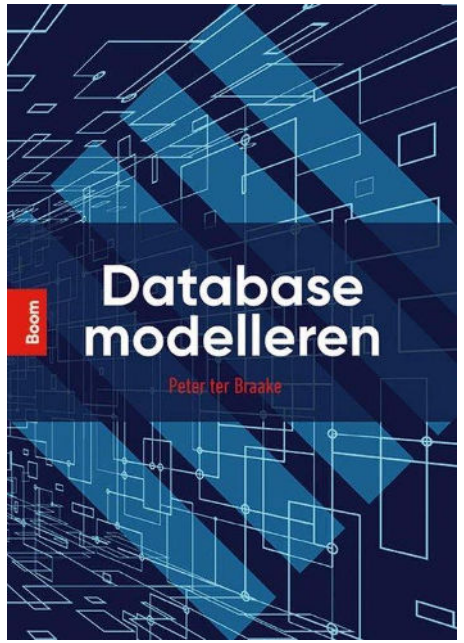


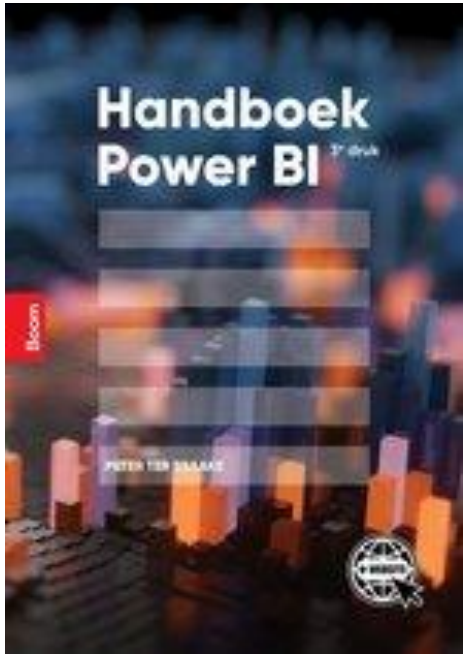
Fabric IQ & Agents

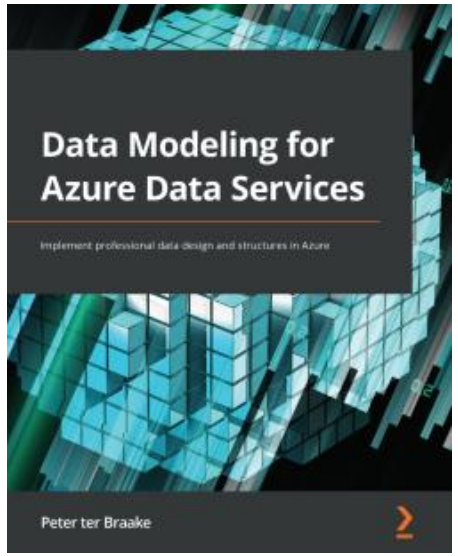
Introducing myself

- Peter ter Braake
  - trainsql@live.nl
  - Independent contractor
  - MCT since 2002
- Microsoft Data Platform
  - SQL Server
    - Incl. SSRS
  - Business Intelligence
  - Power BI
- Author









Agenda

- Problem: Why Fabric needs IQ
- Ontology
  - From Fabric Tables
  - From Semantic model
- Agents
- Vector search and embeddings

Problem: Why Fabric needs IQ

Artificial Intelligence?



Software that imitates human capabilities

- Predicting outcomes and recognizing **patterns**
  - Based on historic data
- Recognize abnormal events
- Make decisions
- Interpret visual input
- Understand language
  - Engage in conversations
- Extract information from sources to gain knowledge

Artificial Intelligence
VS
Human Intelligence

Artificial Intelligence	Human Intelligence
Created by human intelligence	Created by Divine intelligence
Process information faster	Process information slower
Highly objective	May be subjective
More accurate	May be less accurate
Uses 2 watts	Uses 25 watts
Cannot adapt to changes well	Can easily adapt to changes
Cannot multitask that well	Can easily multitask
Below average social skills	Excellent social skills
Still working towards self-awareness	Has self-awareness
Optimization	Innovation

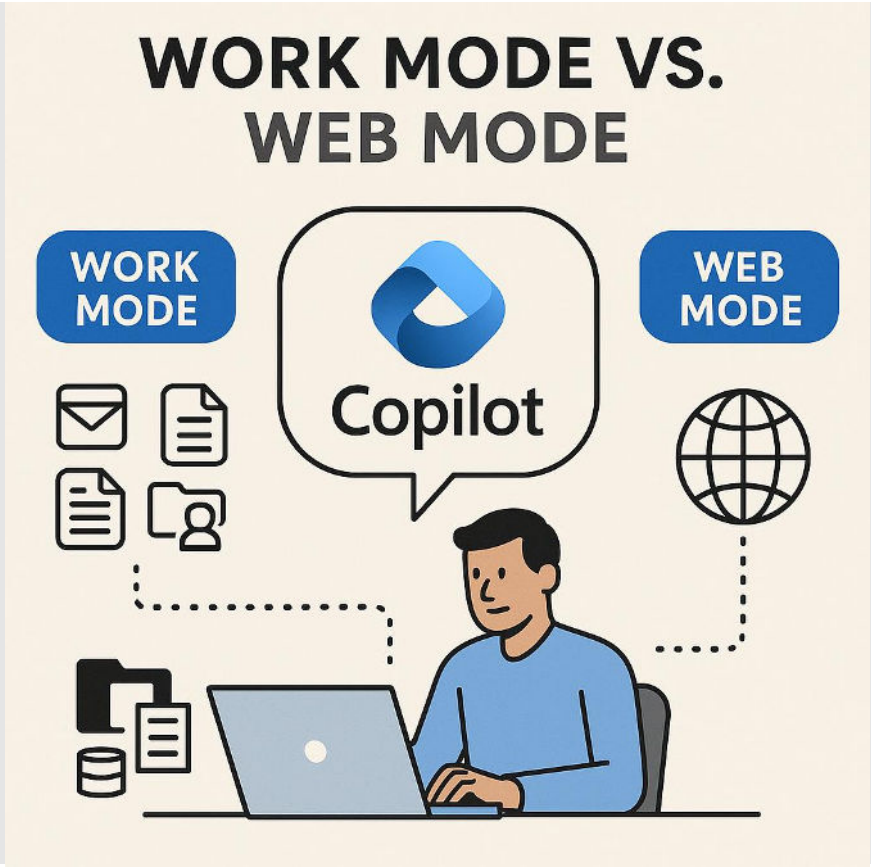
DB Difference Between.net

- Act as assistant
  - Boost productivity
- Automate repetitive tasks

Microsoft Principles of Responsible AI

		Challenge / Risk	Example
	Fairness	Bias can affect results	Discrimination due to bias in data
	Reliability & Safety	Errors may cause harm	Wrong medication due to wrong diagnosis
	Privacy & Security	Data could be exposed	Medical diagnostic bot is trained using sensitive patient data, which is stored insecurely
	Inclusiveness	Solutions may not work for everyone	Predictive app without audio output for visually impaired users
	Transparency	Users must trust complex systems	On what criteria decides a school to admit / reject new students
	Accountability	Who's liable for AI-driven decisions?	An innocent person is convicted of a crime based on evidence from facial recognition – who's responsible?

Generative AI



- Use human language to interact with computers
  - Based on LLM
- Web grounded
  - AI gets information from internet
- Work grounded
  - AI gets information from your data
- Question:
  - Does AI understand your data?
  - Does AI understand your business?

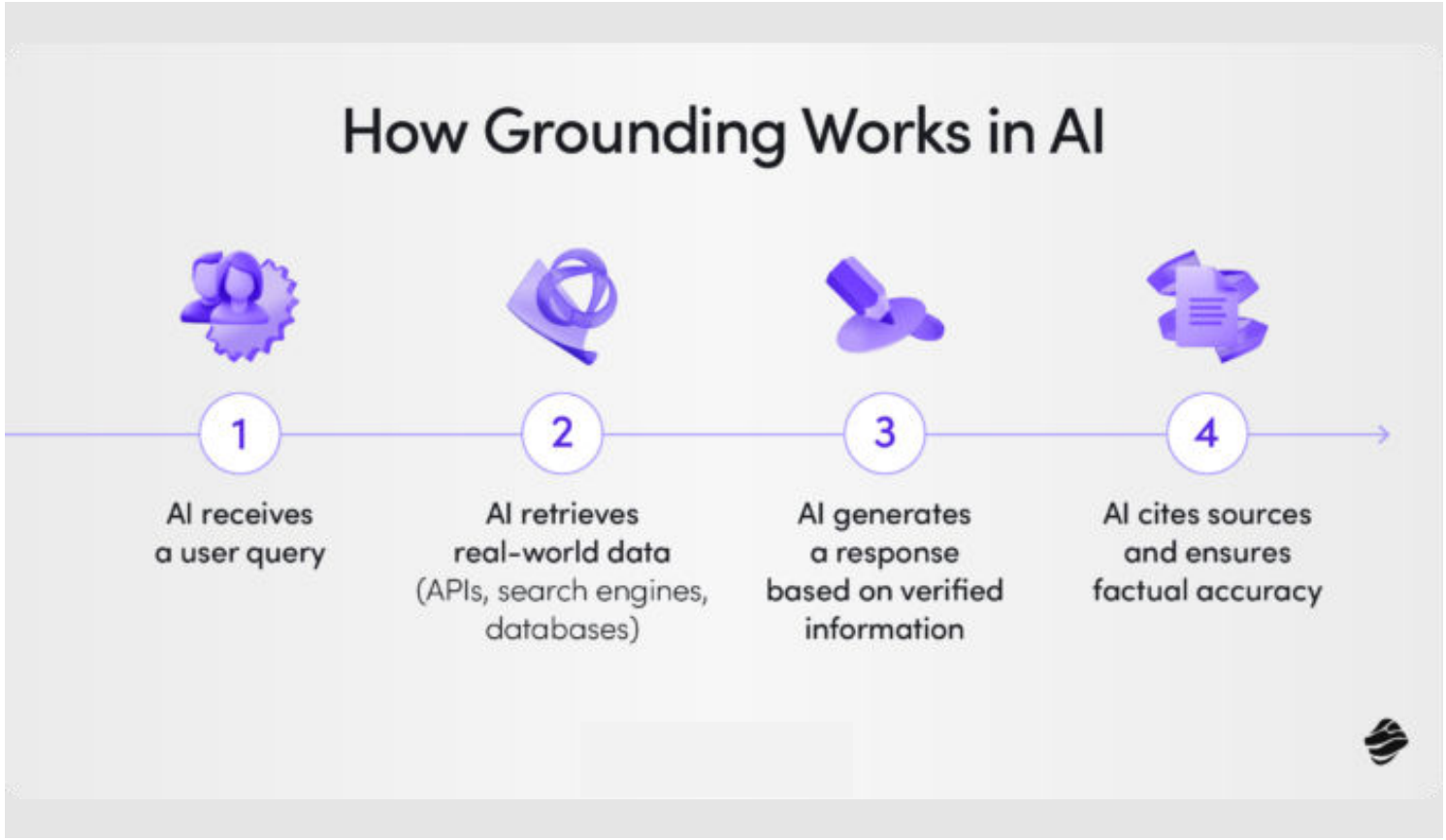
Data



- Operational relational databases
- Streaming data
- Exchange
- SharePoint
- SAAS – API’s

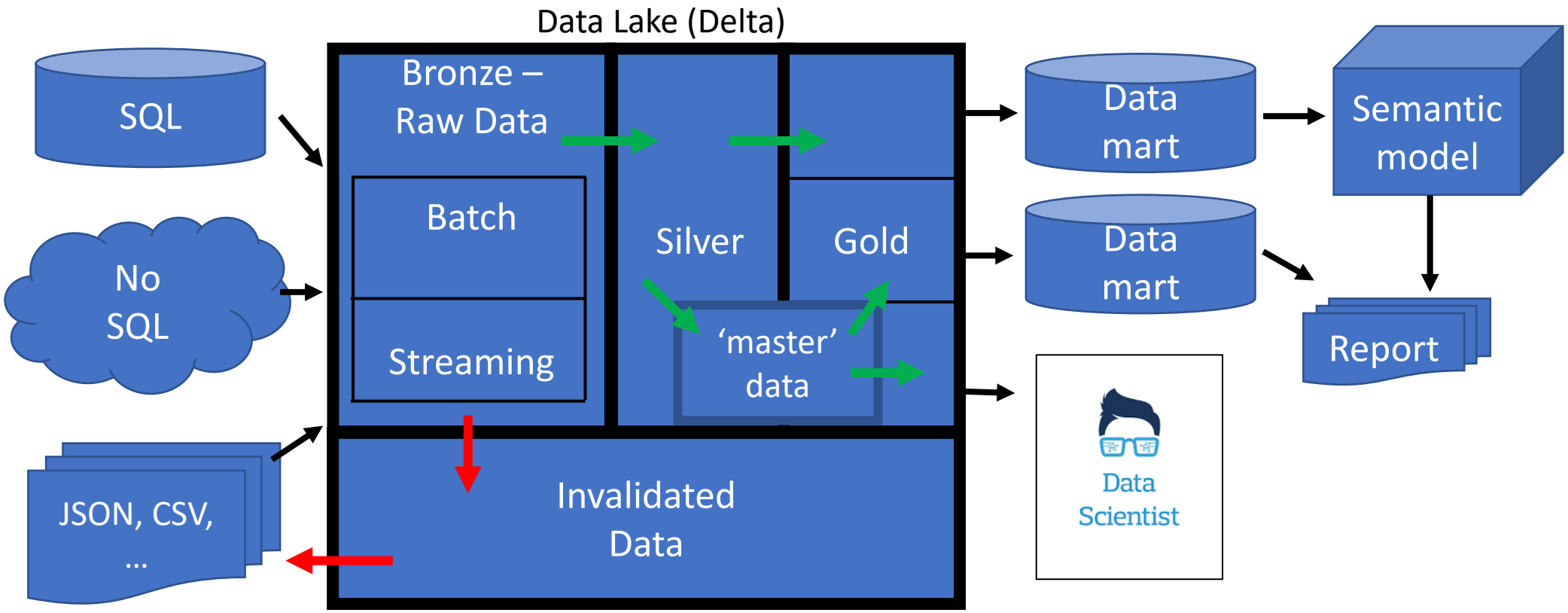
- Often:
  - Complex structure
  - Quality issues
  - Not well documented
  - Inconsistent naming

Data ≠ Meaning

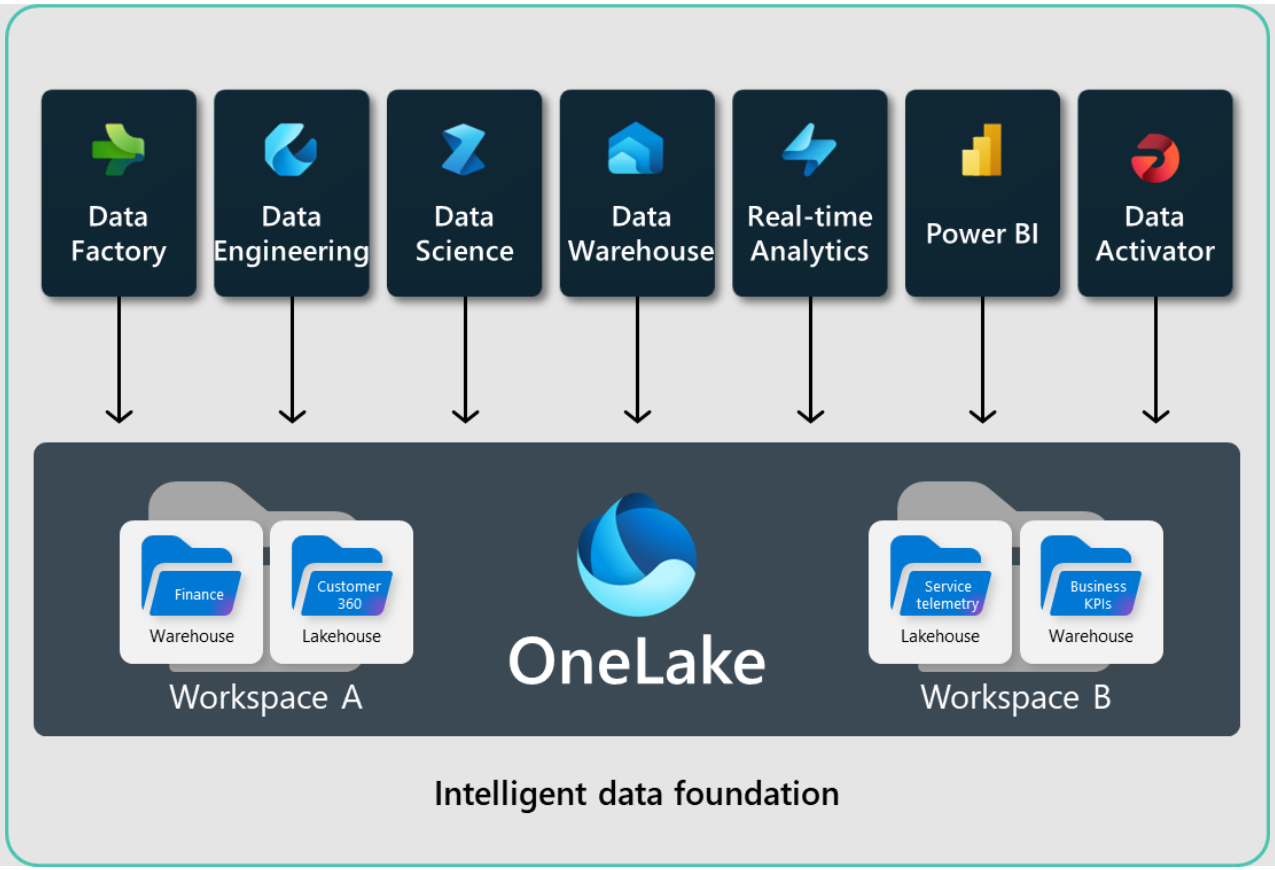


- Traditional BI
  - Data -> Reports -> Humans -> Decisions
- AI needs context
  - Cannot reason on tables
  - Needs ‘human readable’ column / table names
  - Needs ‘knowledge’ of entities and their roles
  - Needs structure
- AI needs semantic layer / business layer

Unified Data platform



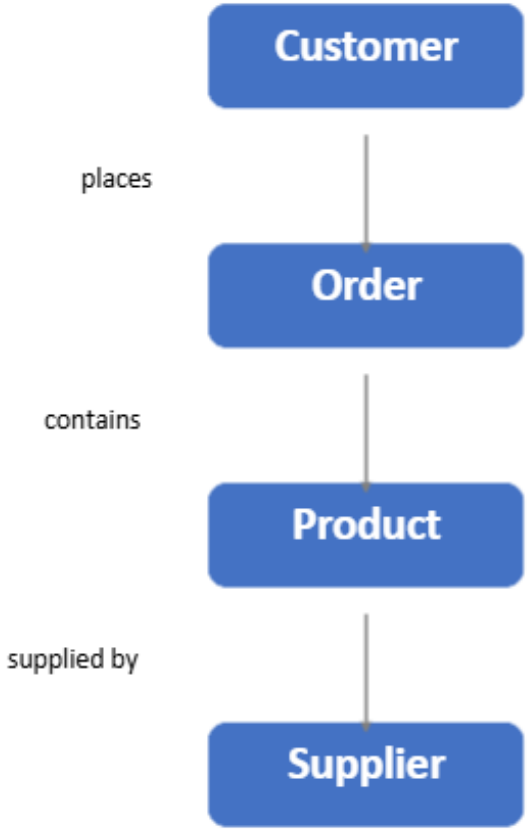
Fabric architecture



- Lakehouse
  - Semi structured data
  - Unstructured data
- Warehouse
  - Structured data
  - Data mart
  - No Ontology (Yet)
- Eventhouse
  - Streaming data
- Semantic model
  - Drag-and-drop reporting
- SQL Database

- OneLake
  - Single storage for all data

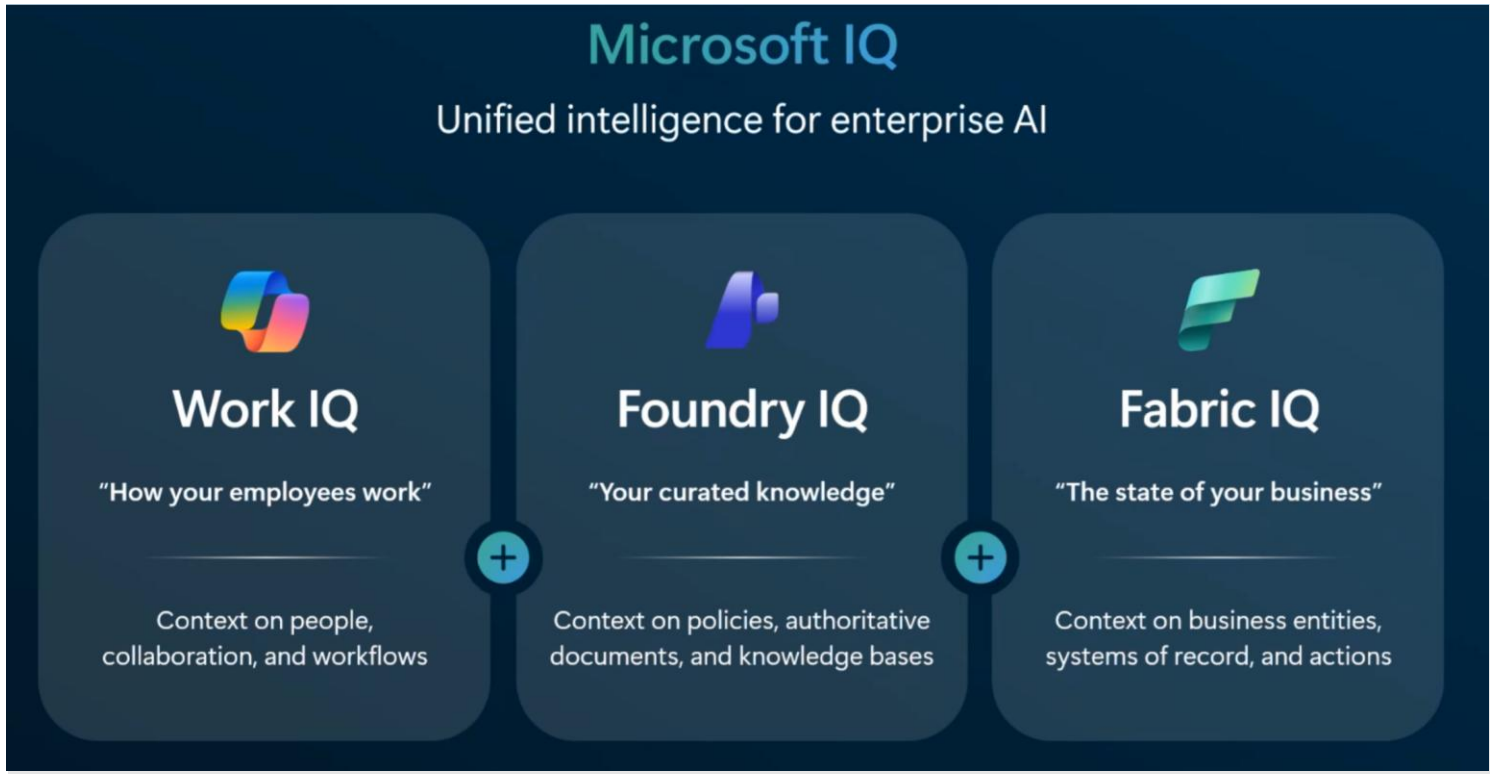
Ontology



Data
+
Meaning
+
Relationships
=
Business Understanding

- Traditional data platforms store
  - Tables
  - Columns
  - Keys
  - Relationships
- An Ontology models
  - Business entities
  - Relationships between entities
  - Business definitions
  - Rules and metrics

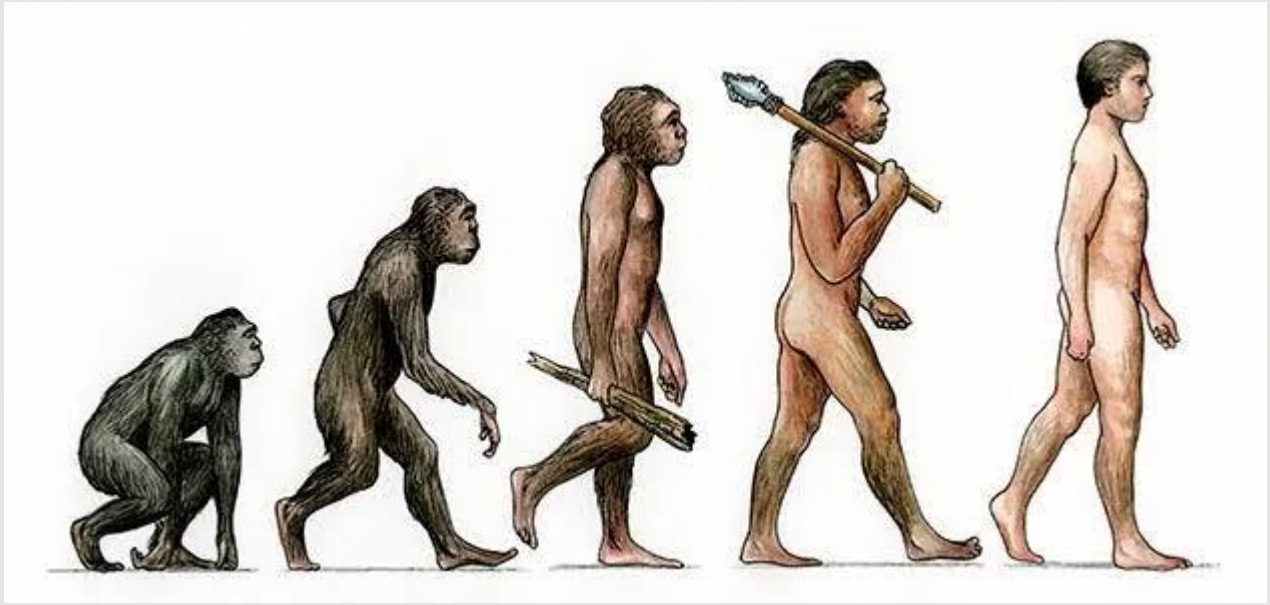
Microsoft IQ



Together they provide enterprise-wide reasoning context for AI

- Work IQ
  - Understands
    - Emails
    - Meetings
    - Files
    - Collaboration patterns
  - 'Knows' how work happens
  - Context-aware Copilot
  - People, meetings, documents
- Fabric IQ
  - Data context
    - Raw data -> business concepts
  - Semantic models
  - Ontology
  - Data-aware agents
    - Reason and act on trusted, governed enterprise data
  - Business data
- Foundry IQ
  - AI/runtime context
  - Manage models, prompts, tools, and agent execution in governed environment
  - Agents, models, tools

Evolution



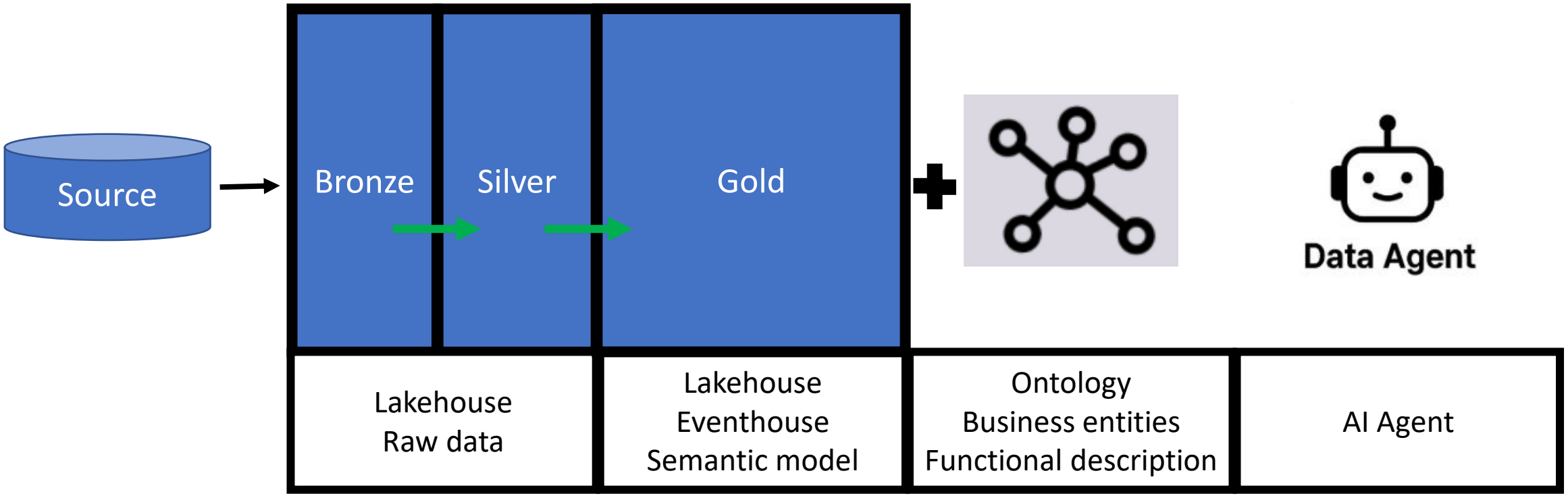
- Traditionally
  - Tables -> columns -> Queries
- Semantic model
  - Customer -> Sales -> Margin
- Fabric IQ
  - Customer buys Product
  - Customer segment drives Revenue
  - Revenue contributes to KPI
- Agent
  - Why is revenue dropping in Germany?

Demo

Create Lakehouse

Ontology

Architecture



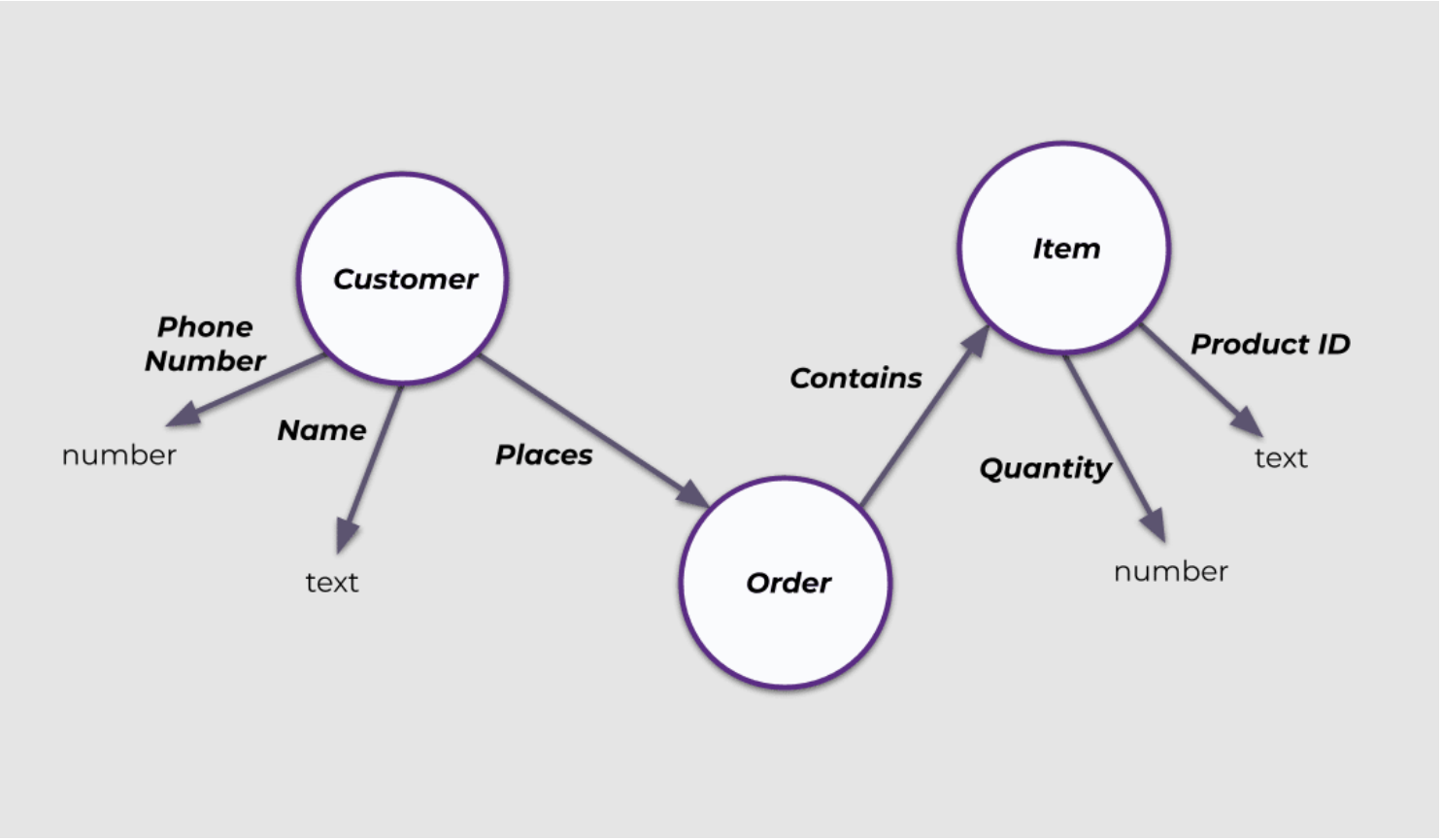
Semantic gap

```
1   ALTER TABLE Sales.factSales
2   ADD CONSTRAINT FK_Sales_Customer FOREIGN KEY (CustomerKey)
3   REFERENCES Sales.dimCustomer (CustomerKey) NOT ENFORCED;
4
5   ALTER TABLE Sales.factSales
6   ADD CONSTRAINT FK_Sales_Employee FOREIGN KEY (EmployeeKey)
7   REFERENCES Sales.dimEmployee (EmployeeKey) NOT ENFORCED;
```

Ontology:
Customer purchases Product
Employee manages Customer

- Ontology makes relationships
  - Explicit
  - Understandable
- AI agents reason over business concepts
  - Not table structures

Ontology components



- Entities describe what exists
  - Entities can be
    - People
    - Objects
    - Events
  - Entities have
    - Properties
    - Metadata
      - Natural language
- Relationships describe how things connect
  - Not technically (PK-FK)
  - Semantically
    - Usually verbs
- Rules describe expected behavior

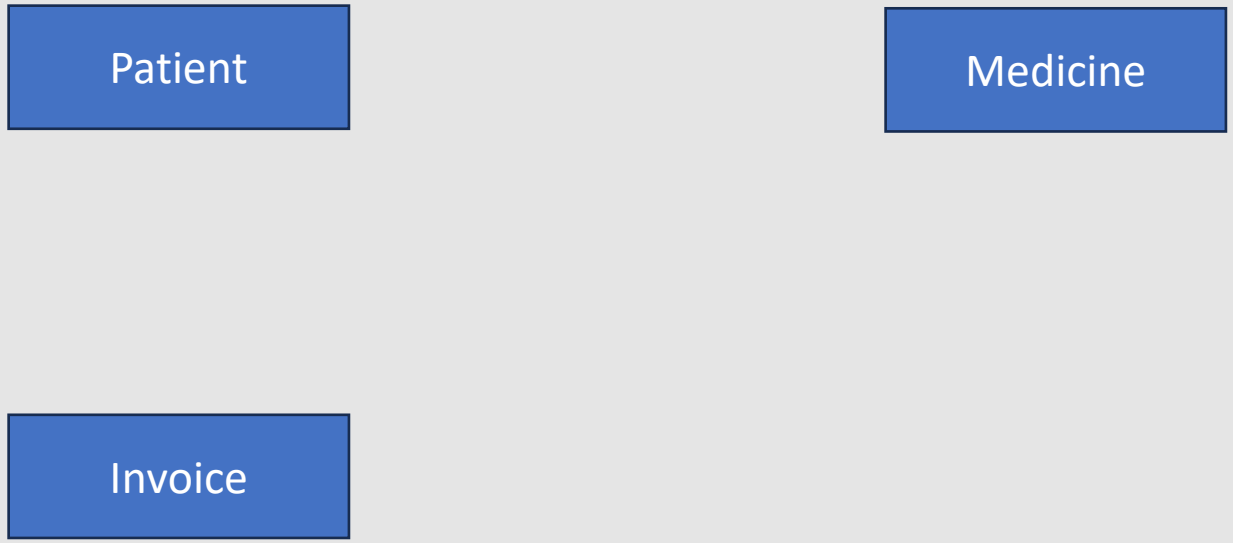
Case: Pharmacy

People go to a pharmacy to get medicines. To prevent wrongful usage of medicines the pharmacy keeps track of who uses which medicines. It also has data on how to produce it’s own medicines from other (base) medicines.

Customers get invoices or are allowed to pay cash at the register.

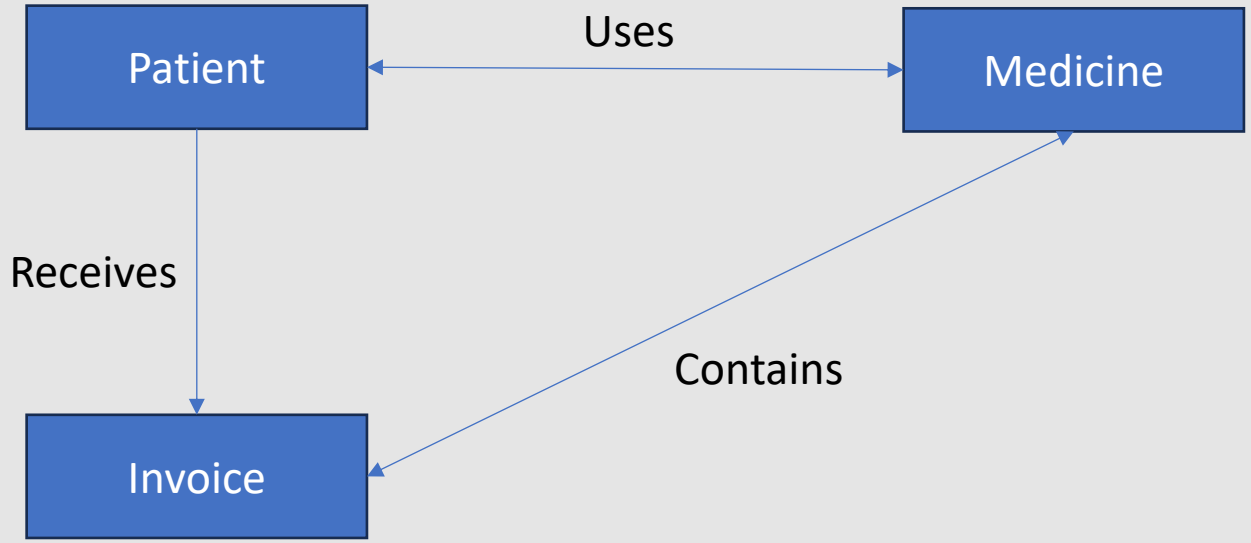
- Start with nouns
- Pay attention to verbs

Pharmacy: step 1



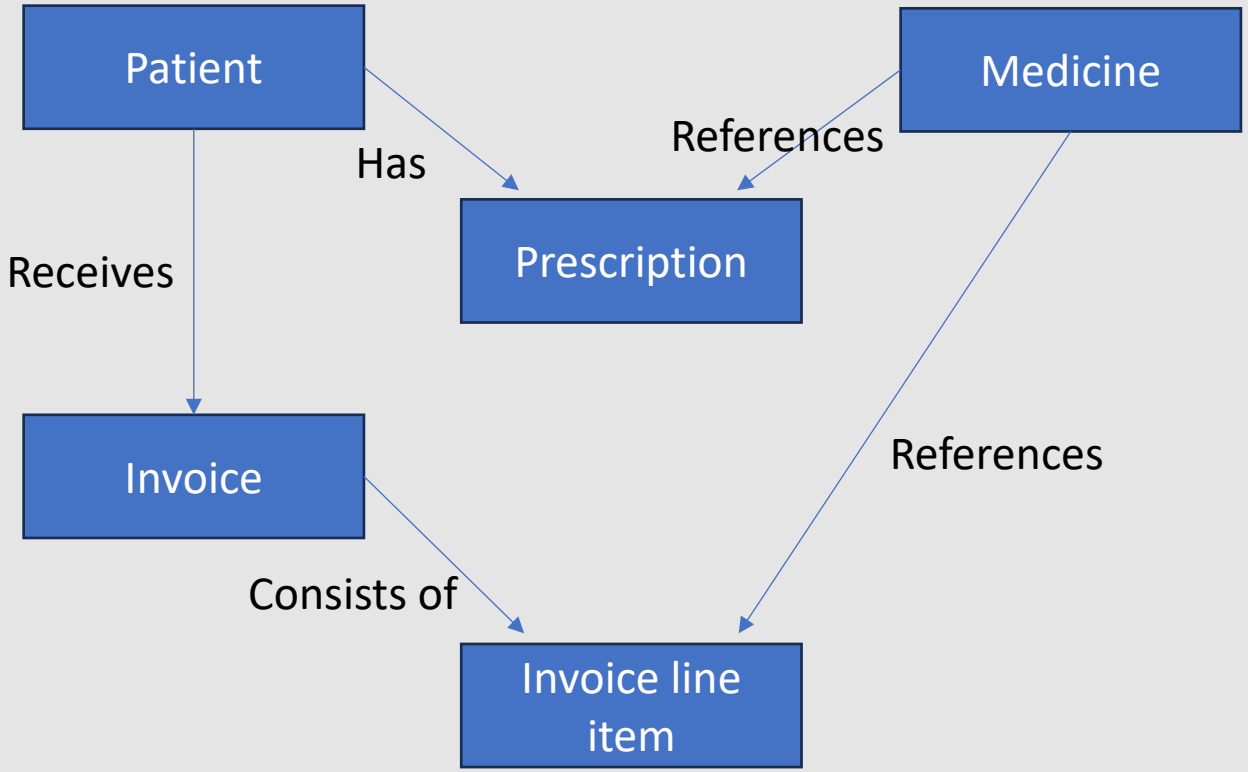
- Entities

Pharmacy: step 2



- Relationships

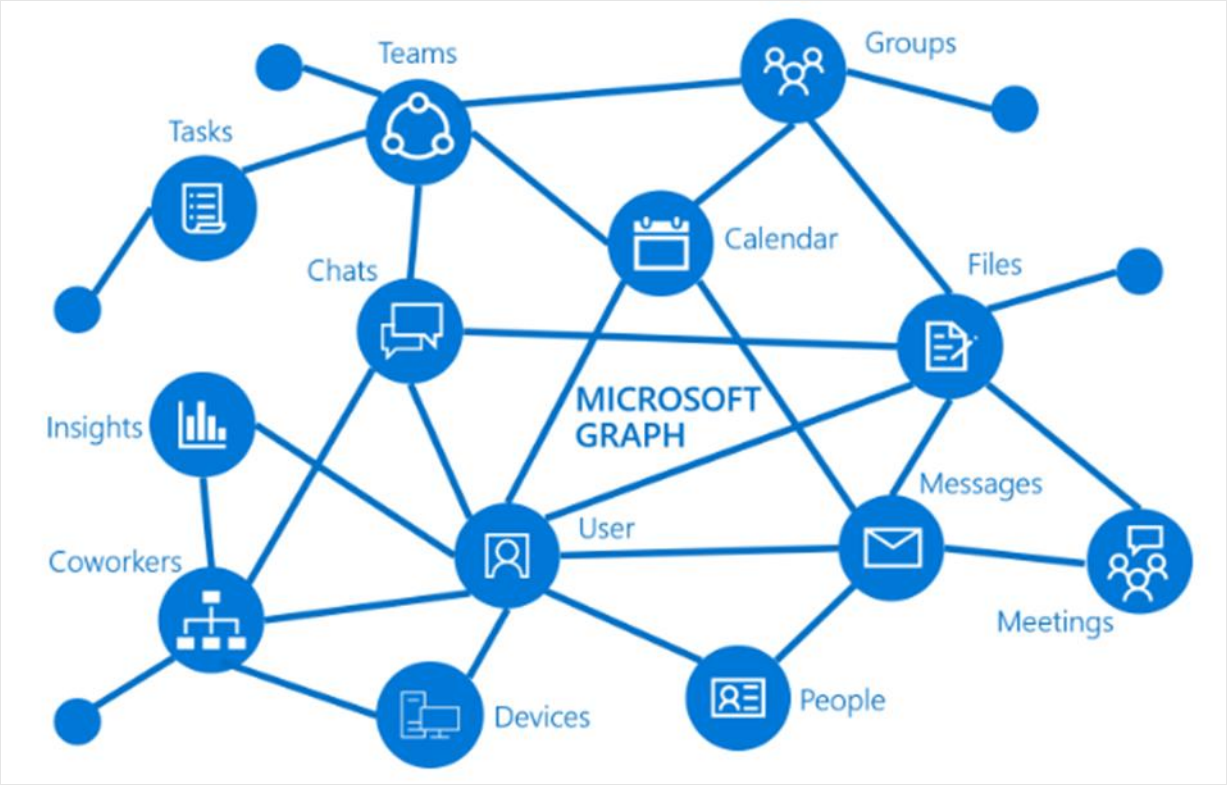
Pharmacy: step 3



- Normalized tables
- ERD = Entity (Type) Relationship Diagram

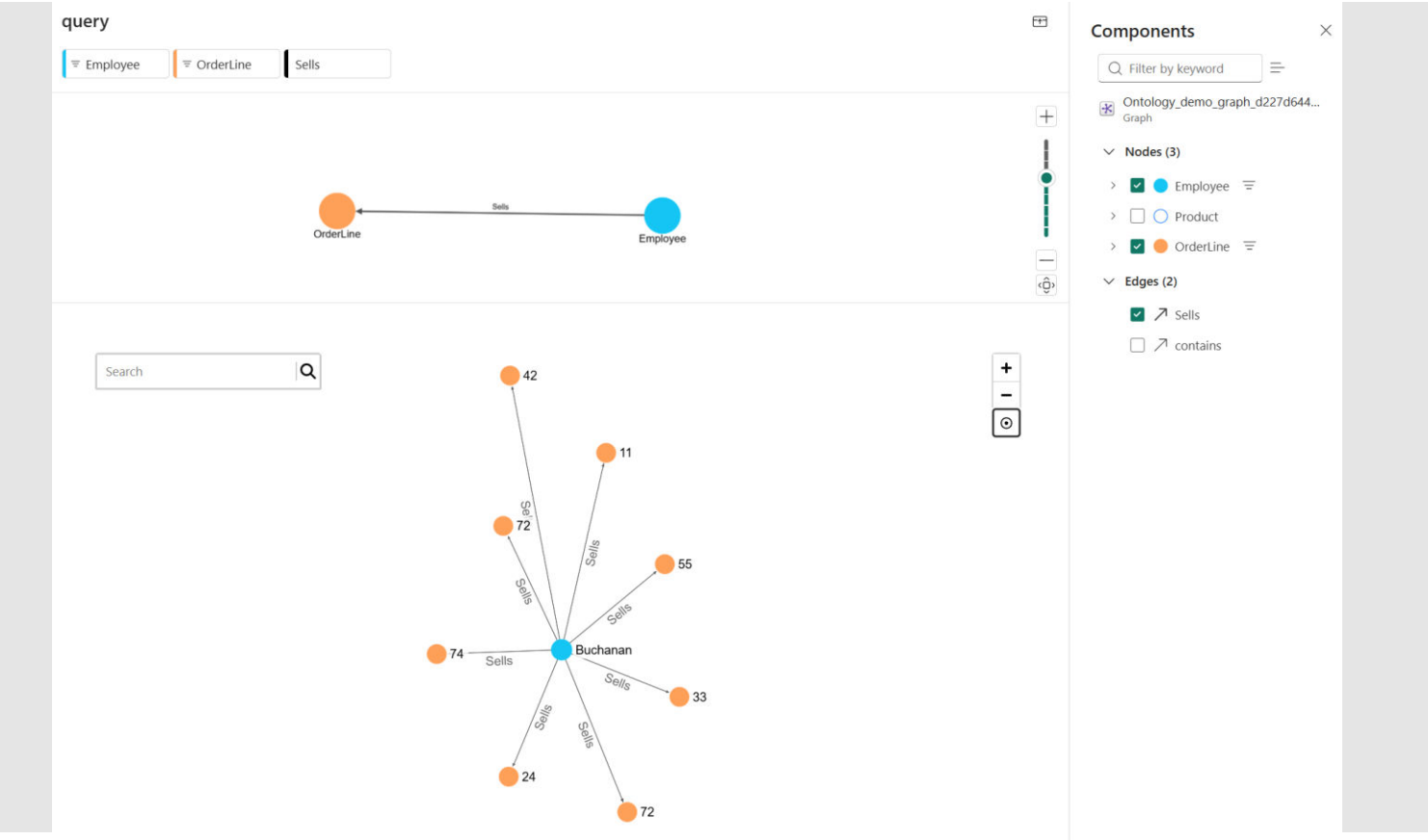
- Entity type
  - General ‘category’
  - Table
- Entity
  - Specific existing occurence
  - Row in table

Graph database



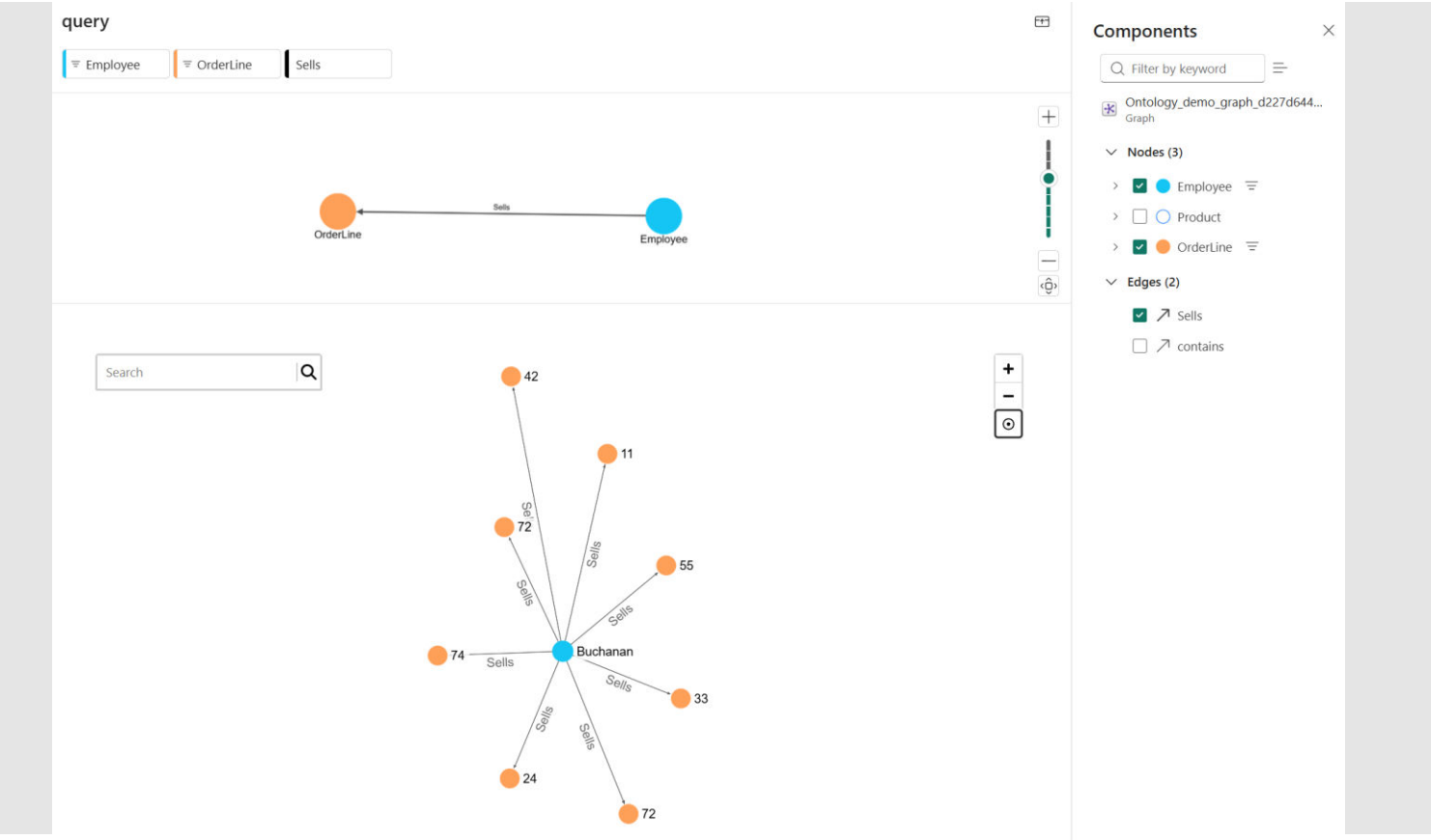
- Database based on Graph theory
  - Relational databases are based on Set theory
- Contains nodes and edges
- Is about relationships more than individual entities
- Examples:
  - LinkedIn
  - Navigation

Microsoft Graph



- Foundation of Microsoft Work IQ
  - Links
    - Mails to users
    - Users to groups to teams
    - Groups to SP libraries
    - …
- Fabric Ontology applies graph concepts to enterprise data understanding
  - Entity types
  - How and why (semantically) they are linked

Together



- Ontology
  - Describe relevant entity types
  - Describe attributes of entity types
  - Describe semantic relationship between entity types
- Well defined descriptions help LLM to understand the meaning of the data
  - and by that, the business
- Underlying technology
  - Graph database

Demo

Create + Test Ontology

Demo

Create + Test Ontology on Semantic model

Ontology scope

```
Entire Company
    └── One Ontology
```

OR

```
Enterprise Ontology
│
├── Sales Domain
├── Finance Domain
├── Supply Chain Domain
├── HR Domain
└── Manufacturing Domain
```

+

```
Sales Ontology
│
├── Customer
├── Product
├── Order
├── Employee
└── Shipper
```

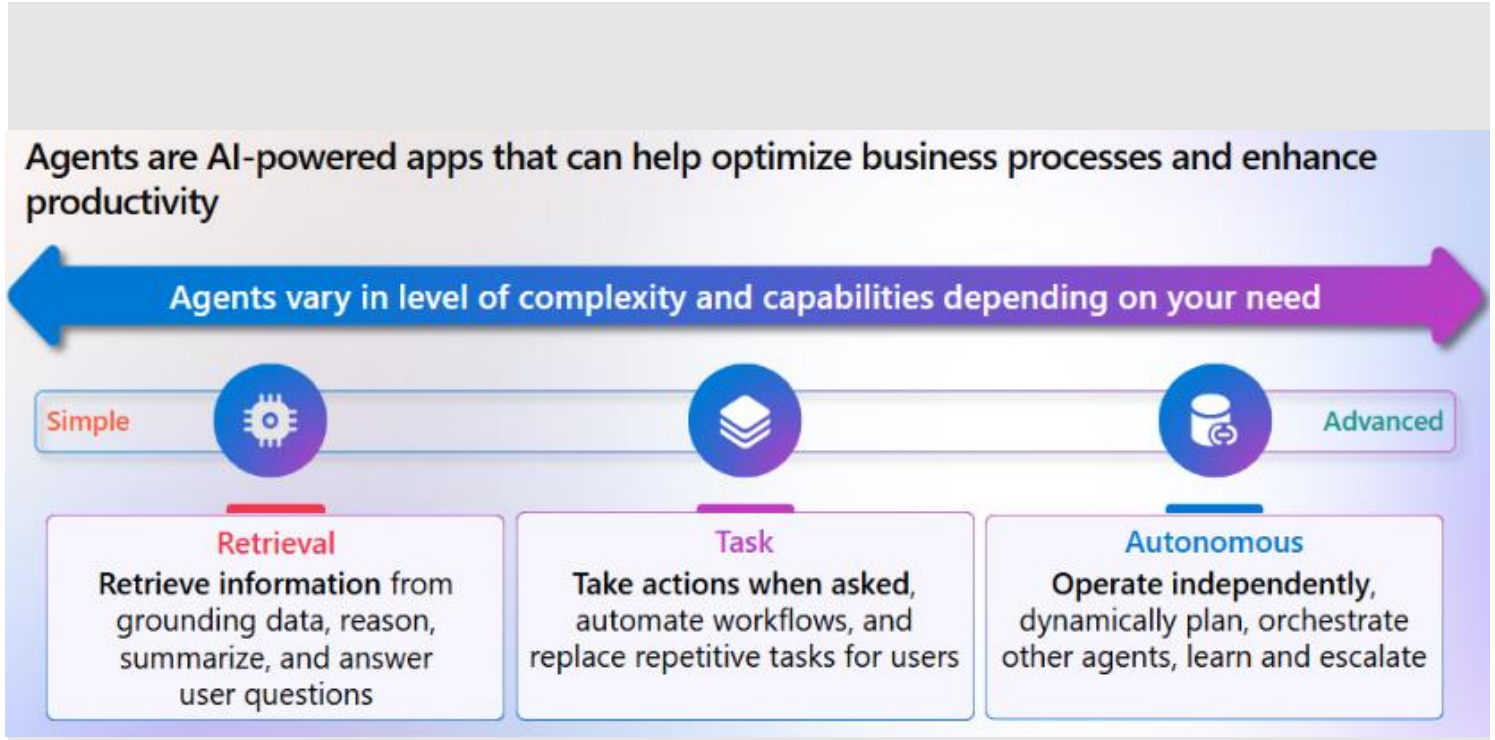
- One overall ontology is hard to create / maintain
- Gold layer
  - Domain oriented
- Per domain
  - Semantic model for reporting
  - Ontology for agents

Ontology versus Semantic model

Semantic Model	Ontology
Designed for reporting and analytics	Designed for business knowledge and relationships
Optimized for Power BI reports and dashboards	Optimized for AI reasoning and business understanding
Focuses on facts, dimensions, measures, and aggregations	Focuses on entities, concepts, and relationships
Answers "How much?" and "How many?"	Answers "What is related to what?"
Typically modeled as a star schema	Typically modeled as a business graph
Measures such as Revenue, Margin, Quantity Sold	Relationships such as Employee sells Product
Excellent for KPI reporting	Excellent for AI agents and knowledge discovery
Business users consume reports	Agents consume business meaning
Enables DAX calculations	Enables semantic reasoning across entities
Best for analytics	Best for understanding context and relationships

Agents

What are agents?



- Specialized AI
  - Grounded chat
  - Special instructions
- Automate process
  - Can be autonomous

Agent types

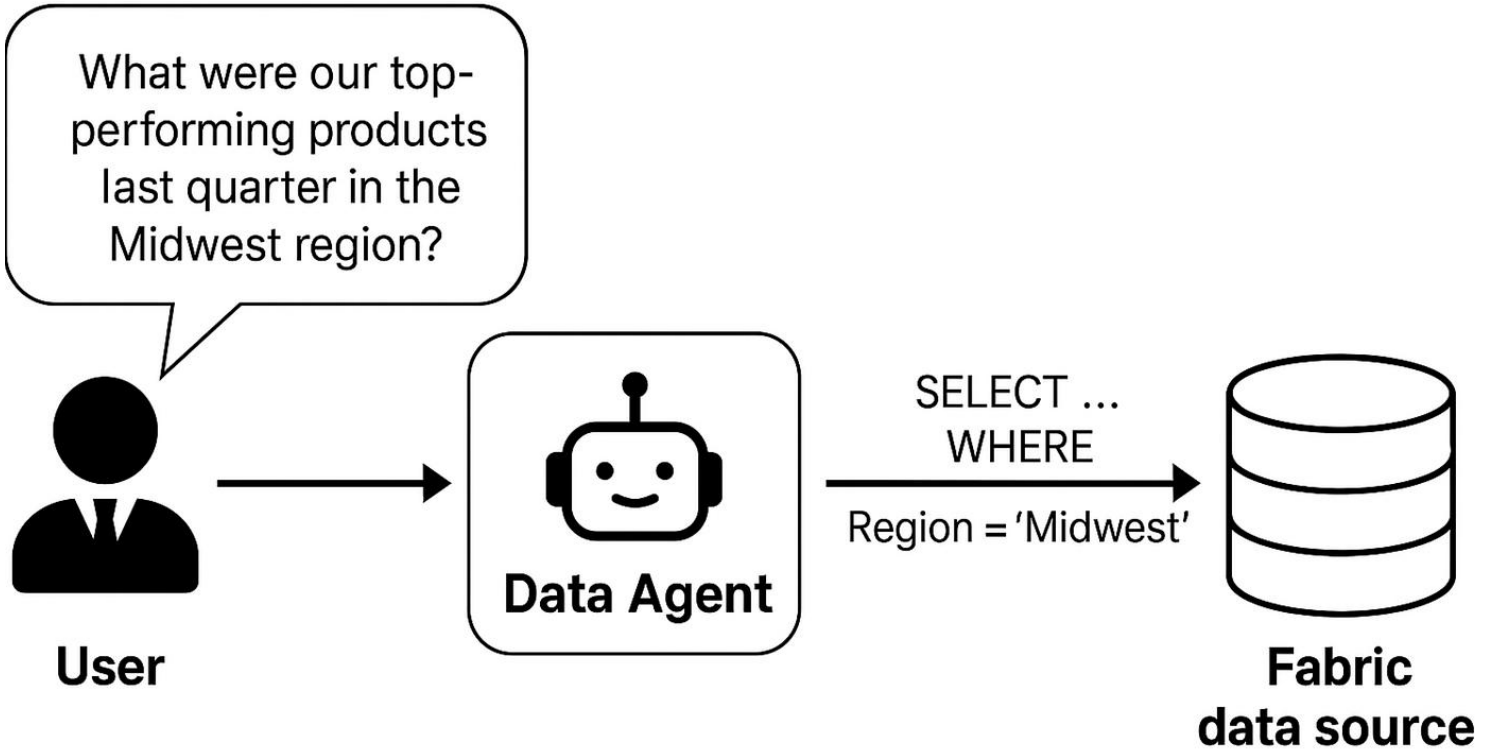
Agent	Primary goal	Usage	Data Source	Audience	Action	Customization
Data Agent	Q&A on data	Analysis	OneLake assets	Data Analysts Data Engineers	Read-only analytics	Medium
Operations Agent	Monitor + Act	Monitoring + action	Eventhouse Ontology	Ops teams	Notifications Flows Notebooks	Medium
Foundry	Custom AI apps	Reasoning + orchestration	Knowledge base Ontology	Developers	Tool calling	High
Copilot Studio	Business Agents	Business process automation	Ontology MCP Data Agent	Citizen developers	Worklows	High

Agents use RAG

RAG	Agent
Retrieves information	Can act
Single prompt	Multi-step reasoning
Read-only	Processes
Knowledge	Knowledge + Actions

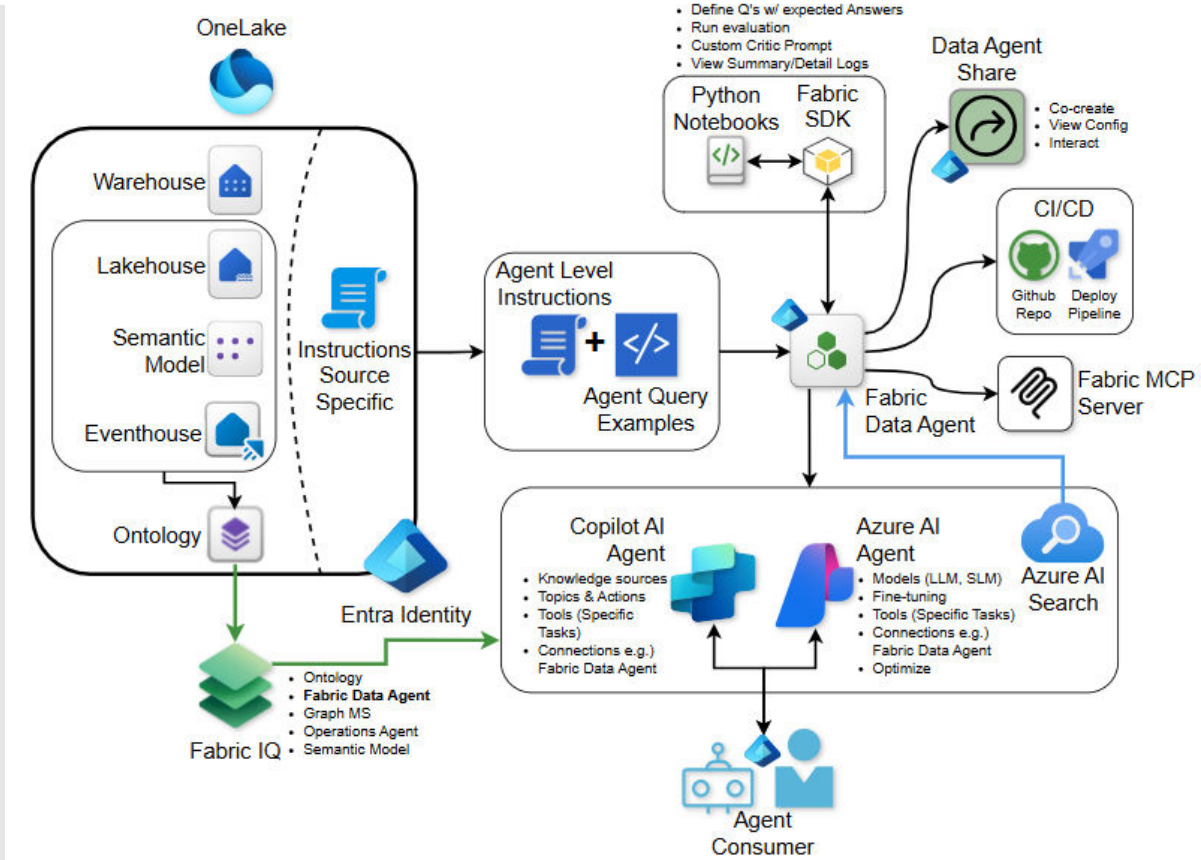
- Fabric agents retrieve your data
- Prompts get enriched with results of queries
- Answers are based on query results

Fabric Data Agent



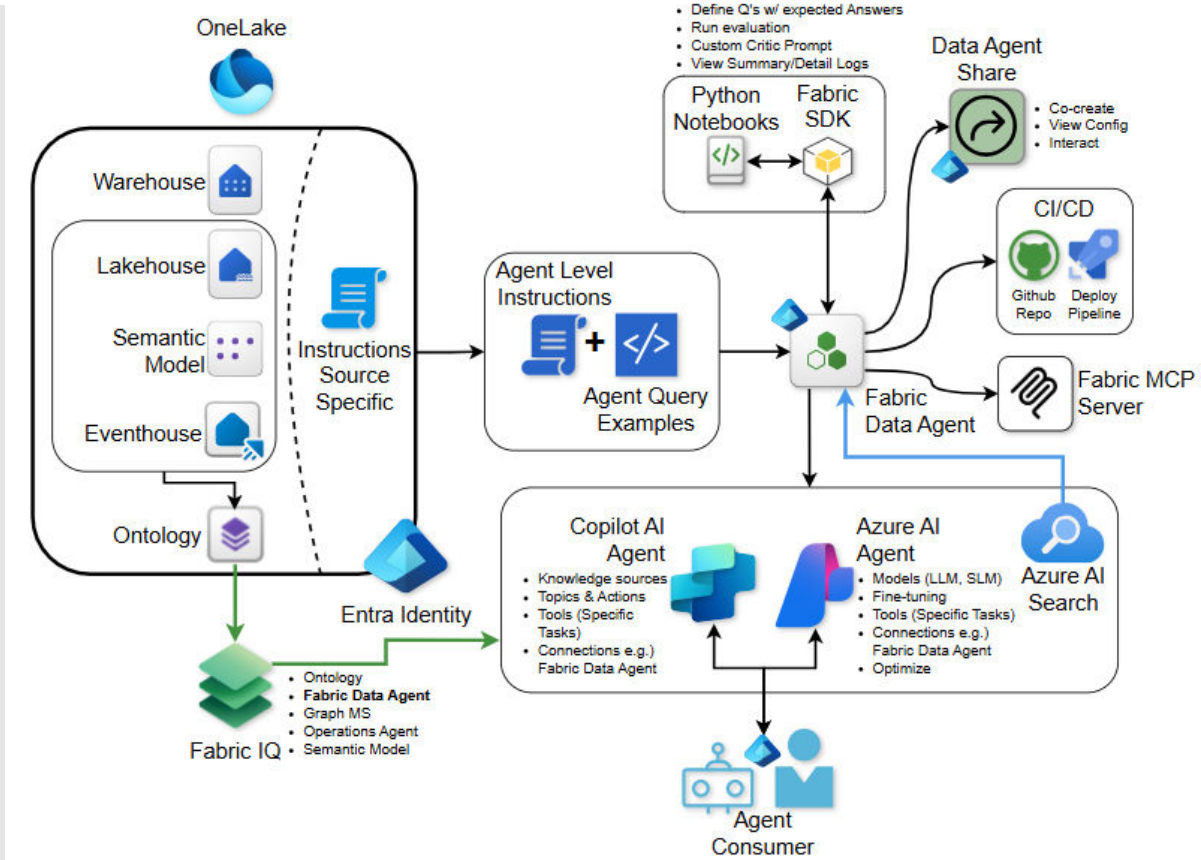
- Make data insights more accessible and actionable for everyone
  - Human language conversations about data stored in OneLake
  - Conversational analytics component
    - Connect to governed data in OneLake
      - Lakehouses
      - Warehouses
      - Semantic models
      - KQL databases
      - Ontology
- Add organization-specific
  - Instructions
  - Examples
  - Guidance
- Respects Purview governance controls

How it works - 1



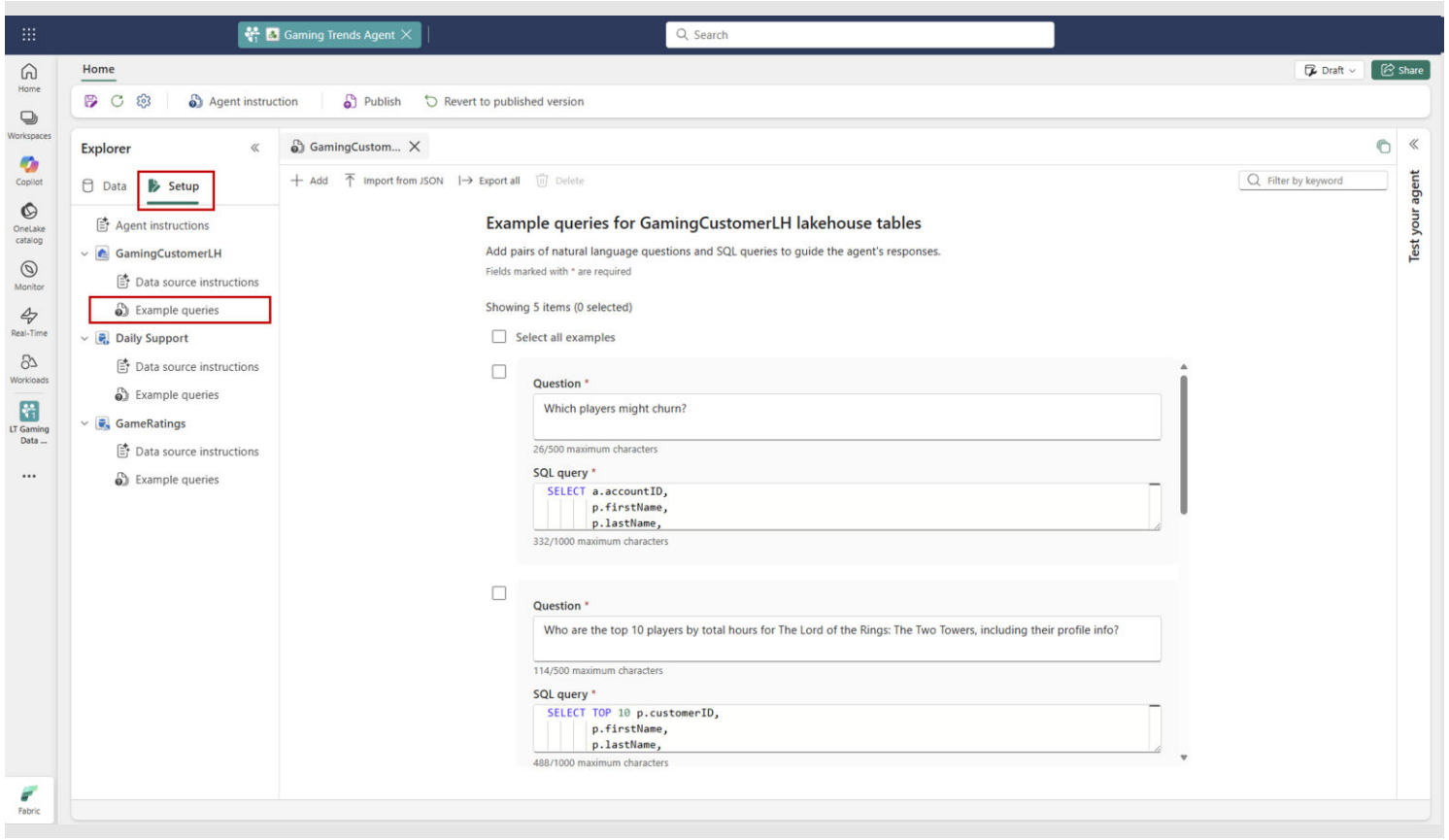
- Question parsing and validation
  - Uses Azure OpenAI Assistant APIs to process user questions
  - Ensures question complies with
    - Security protocols
    - Responsible AI policies
    - User permissions
- Enforcement mechanisms
  - Uses requesting user's credentials
  - Evaluates requests against tenant and workspace policy settings
  - Guardrails apply
- Data source identification
  - Question evaluated against all available data sources
  - Uses user-provided data agent instructions to determine most relevant source

How it works - 2



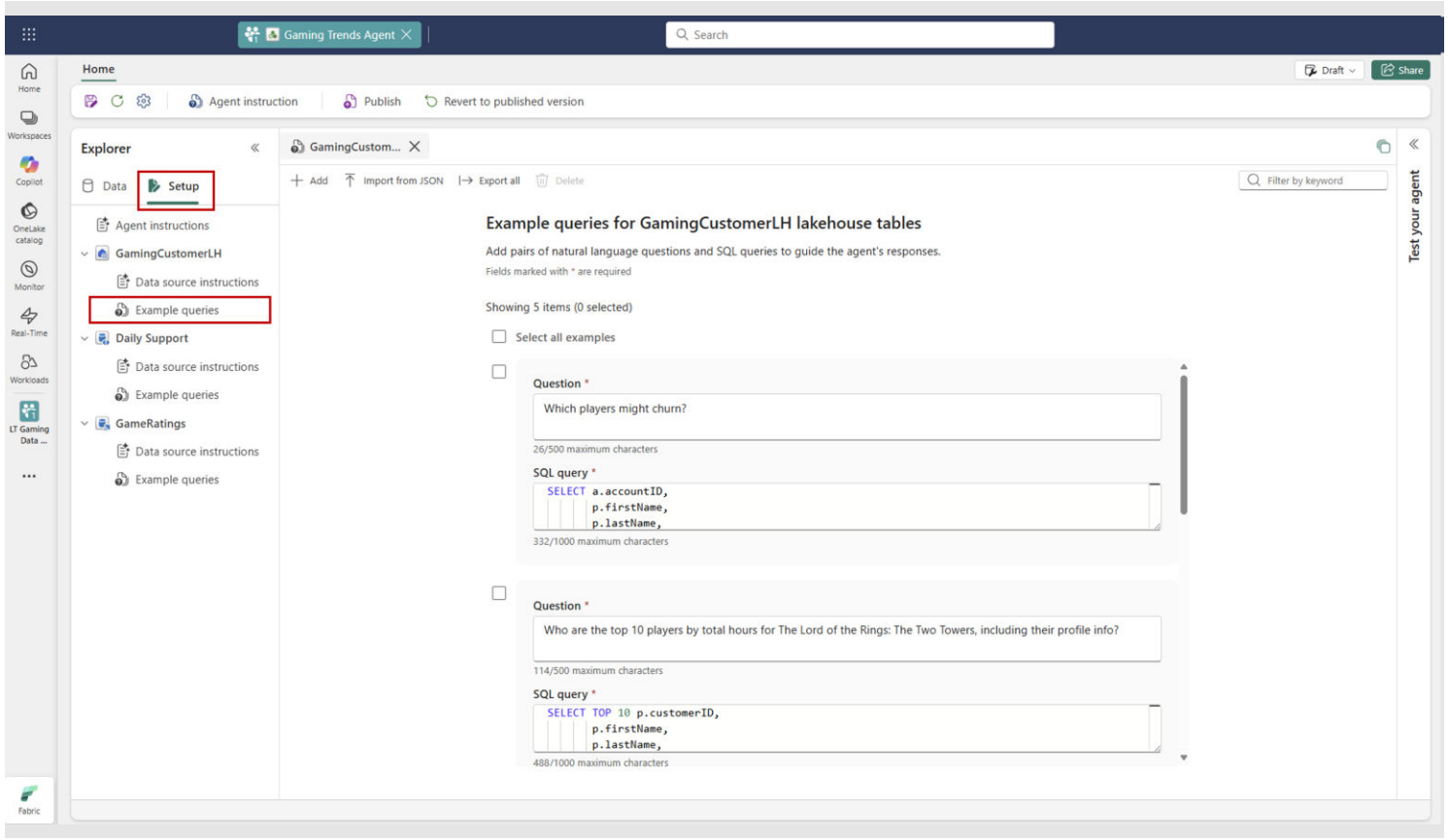
- Tool invocation and query generation
  - Rephrases question for clarity and structure
  - Invokes corresponding tool to generate structured query
    - Natural language to SQL
    - Natural language to DAX
    - Natural language to KQL
    - Graph queries
- Query validation
- Query execution and response
  - Executes query
  - Result formatted into human-readable response

Fabric data agent configuration



- Select data sources
  - Up to five data sources
- Choose relevant tables
  - Retrieve accurate results by focusing on relevant data
- Add context
  - Instructions
    - Guide agent in determining best data source
    - Provide custom rules / definitions to clarify organizational terminology or specific requirements
  - Example queries
    - Sample question-query pairs

Fabric data agent usage

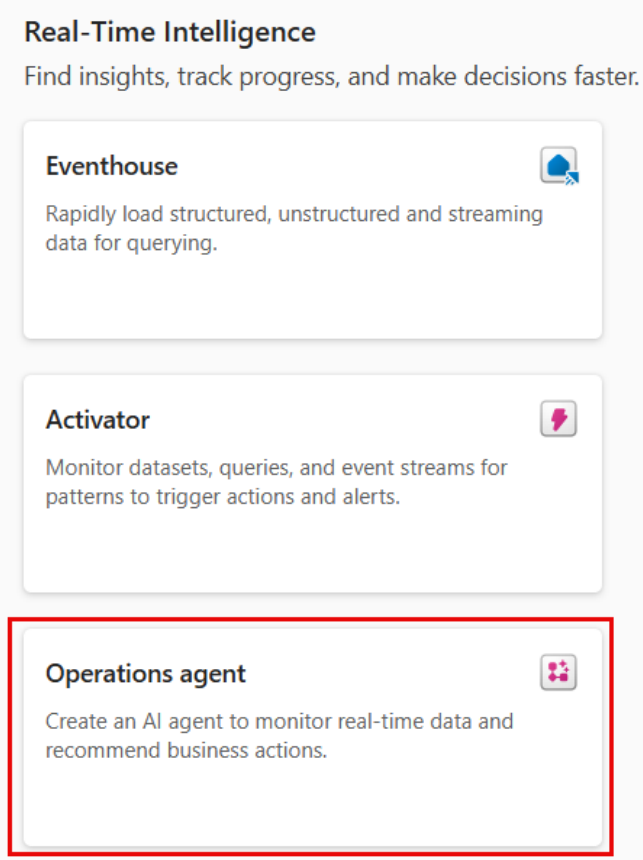


- Consume from
  - Foundry
  - Copilot in Power BI
  - Copilot Studio
  - Python SDK
  - M365 Copilot
  - AS MCP server

Demo

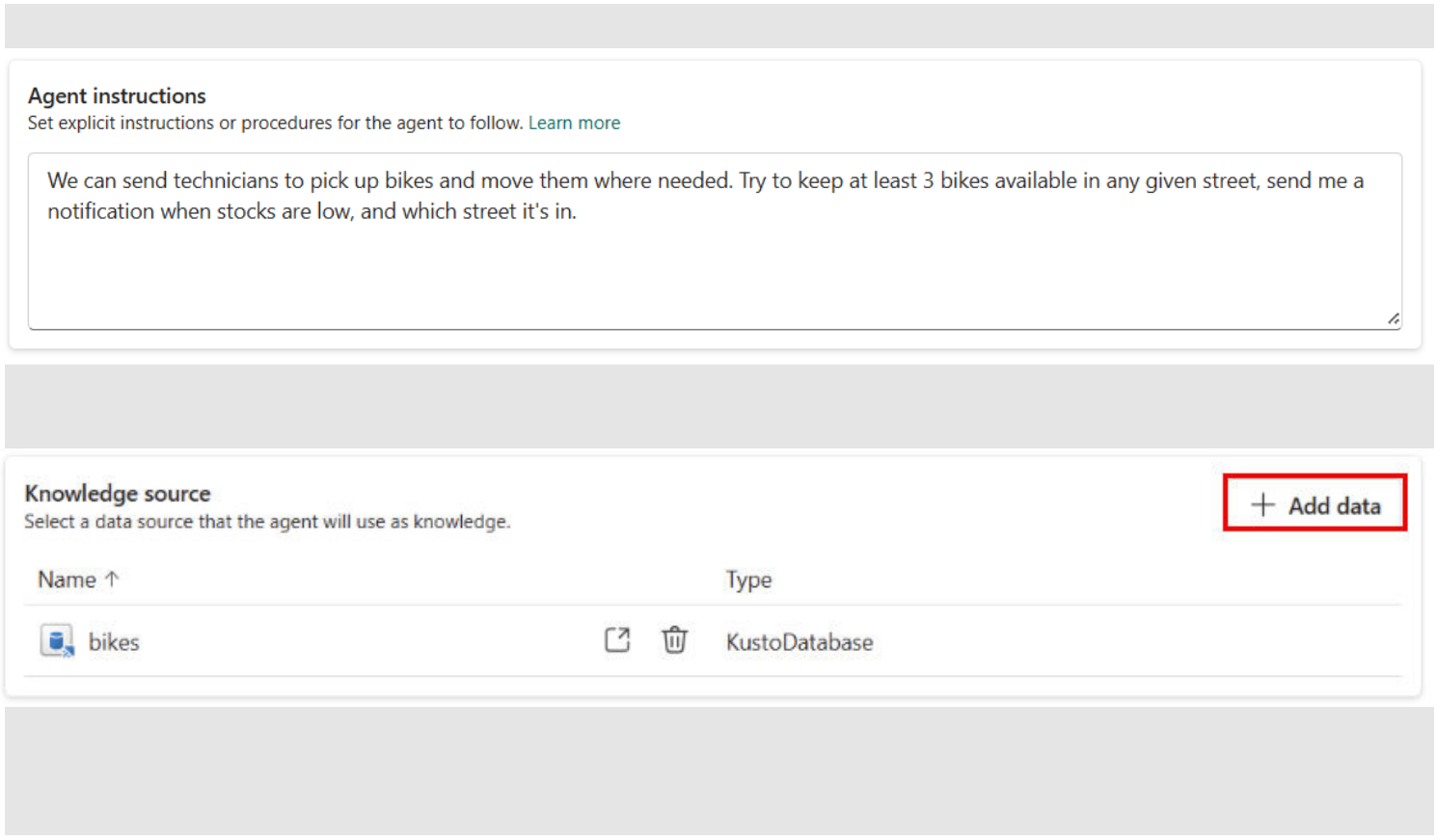
Create + Test Data Agent

Fabric Operations Agent



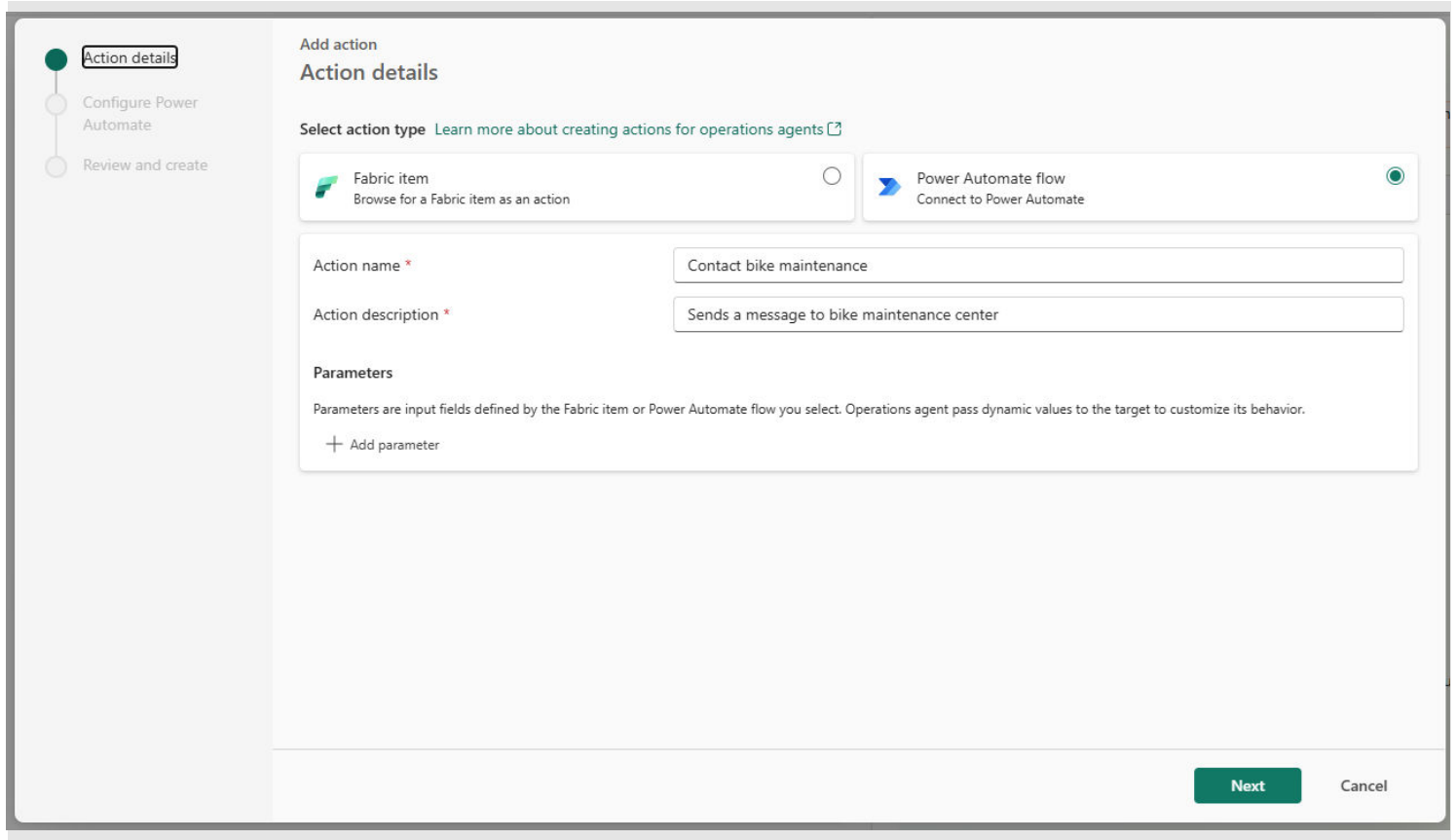
- Turn real-time data into immediate, actionable decisions
- Track key metrics continuously
  - Surface insights
  - Recommend actions
- Each operations agent is dedicated Fabric item, designed for specific business process

Configure Operations Agent



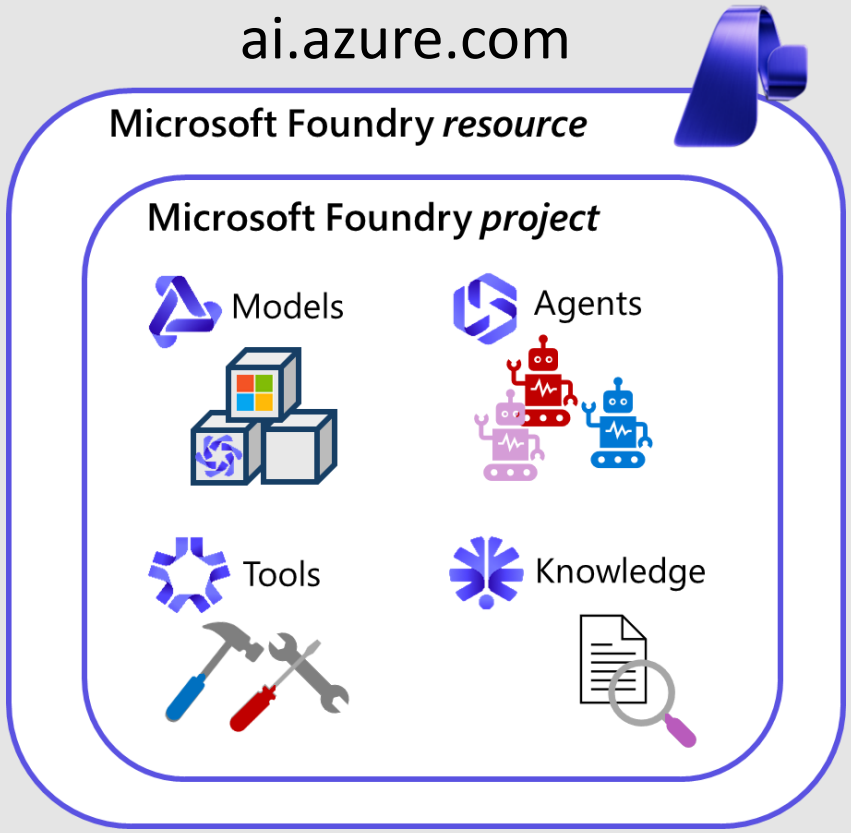
- Provide instructions
  - What metric to monitor
  - Threshold
  - Optionally use Copilot chat to create instructions
- Choose relevant data source
- Send message in Teams
  - Optionally configure additional actions
- Runs query every 5 minutes
- 2 rules
  - State conditions
    - E.g. Temp > 80
    - Can trigger repeatedly as long as condition remains true
  - Transition conditions
    - E.g. Temp crosses above 80
    - Trigger only when value changes from not meeting condition to meeting it

Operations Agent actions



- Message in teams
- Fabric item
  - Notebooks
  - Pipelines
  - User-defined functions
- Power Automate flow

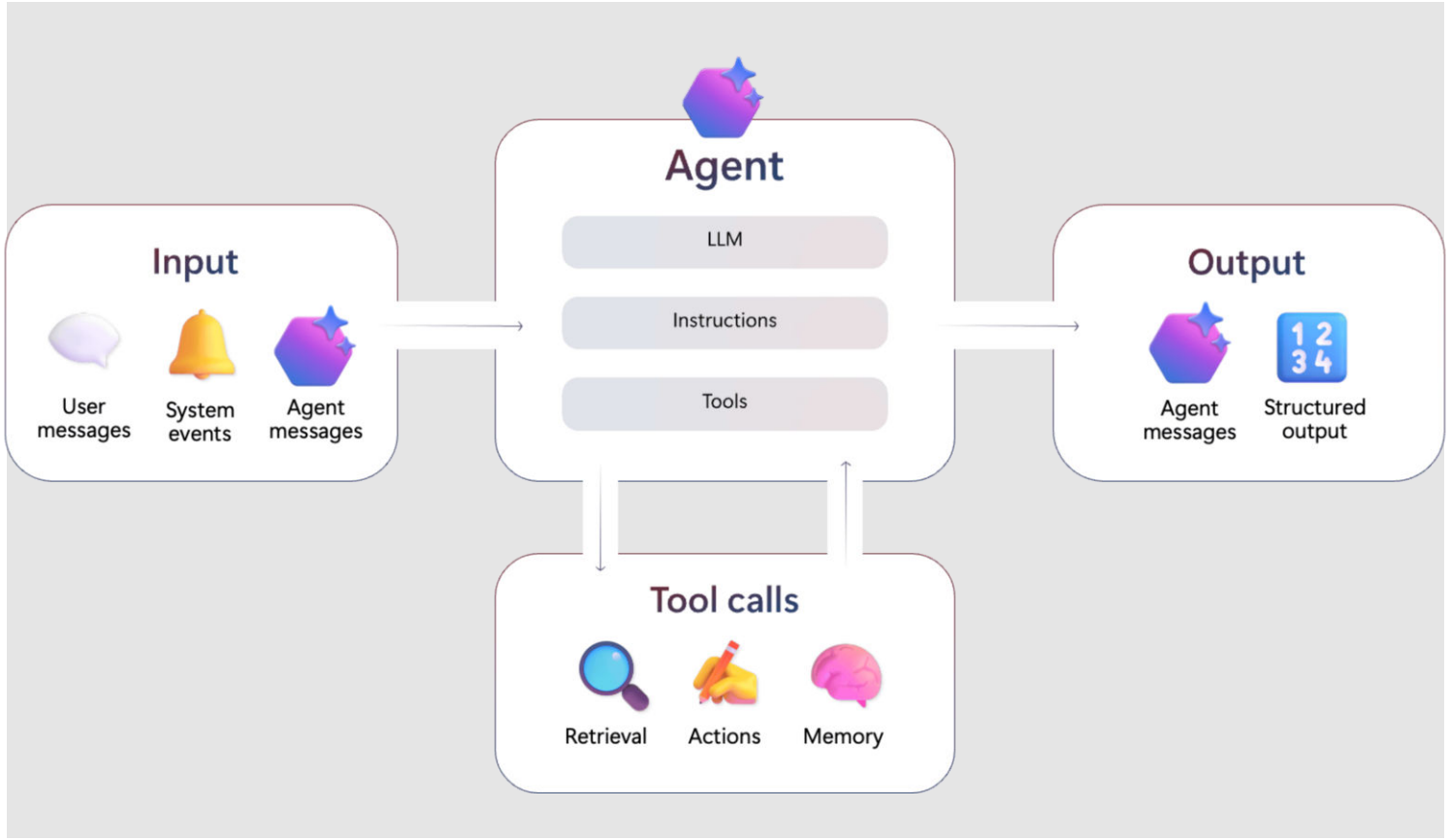
Microsoft Foundry



Unified platform for AI apps and agents

- Foundry resources provide
  - Compute
  - Storage
  - Model delivery
  - Other services
- Foundry project
  - Isolated work environment for app and agent development
- In a project, you work with
  - Models
    - Choose / deploy models from model catalog
  - Agents
    - Create / manage agents
    - Use models for language and reasoning
  - Tools
    - Connect agents to tools so they can perform actions and access information
  - Knowledge
    - Manage data source access
    - Ground AI -> ONTOLOGY

Foundry Agent



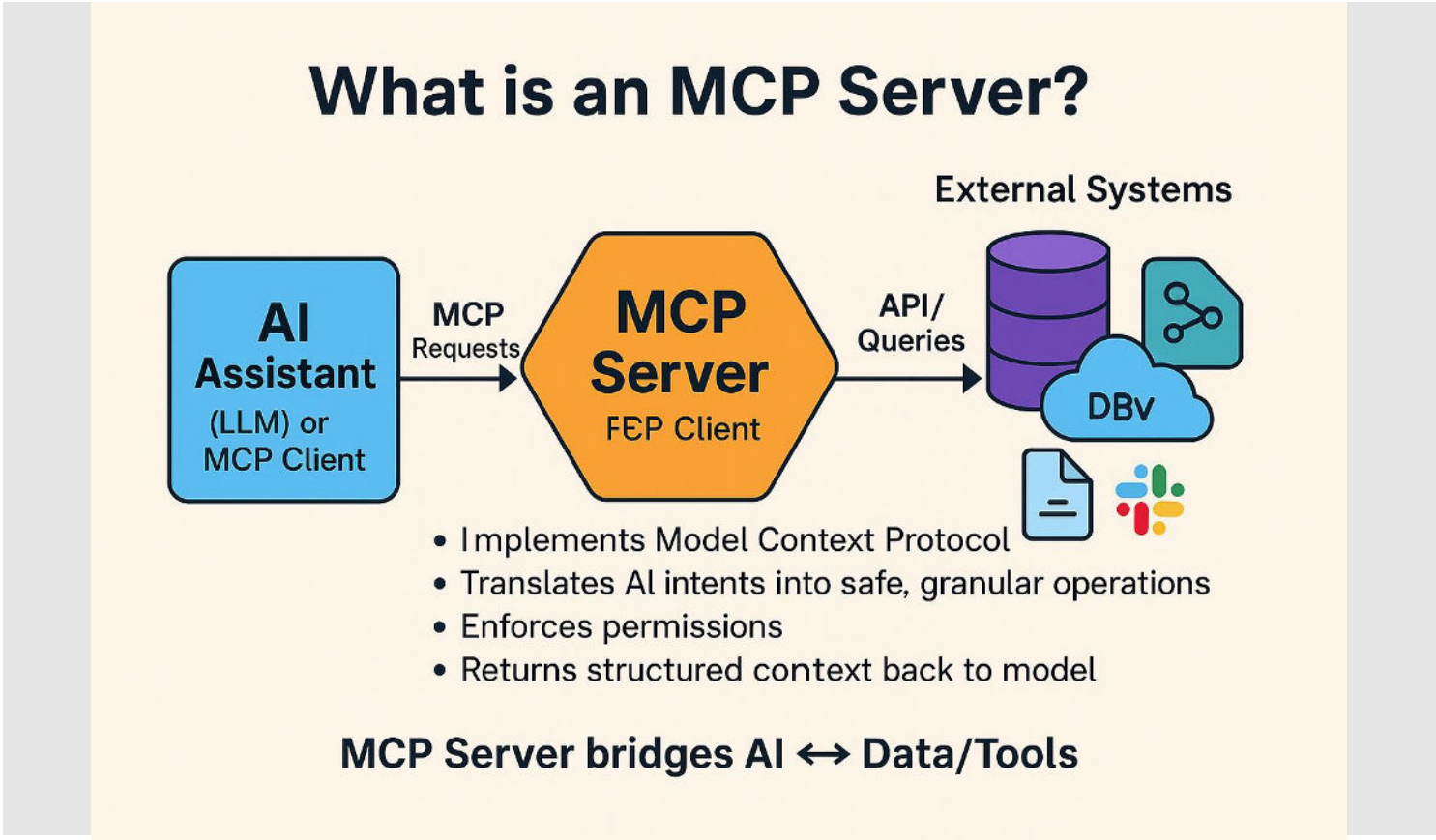
- Create project
  - Create agents
    - Use AI model
    - Set instructions
    - Add tools
    - Add grounding source
      - Ontology
  - Create workflows
    - Combine multiple agents -> orchestrated process
  - Browse models

Foundry versus Copilot studio

Scenario	How Fabric is connected	MCP needed?
Azure AI Foundry Agent + Fabric IQ Ontology	Ontology becomes a **Knowledge Source** in Foundry IQ / Knowledge Base	❌ No
Copilot Studio Agent + Fabric IQ Ontology	Ontology is exposed through the **Fabric IQ MCP tool**	✅ Yes
Copilot Studio Agent + Fabric Data Agent	Agent-to-agent integration via MCP-based connection	✅ Yes
Copilot Studio + SharePoint/Websites/Documents	Native knowledge sources	❌ No

- Copilot studio
  - Low code / no code

MCP Server



- Model Context Protocol
- Bridge program
  - let's AI models connect safely to outside tools, files, and databases
- Microsoft Fabric
  - Fabric IQ Ontology MCP
    - Hosted in Fabric
  - Power BI MCP Server (Remote)
    - Microsoft-hosted endpoint
  - Power BI MCP Server (Local)
    - Install/run locally
  - Fabric MCP (Preview)
    - Fabric API's
    - Install/run locally

Vector search and embeddings

Traditional search

```
1  USE [AdventureWorks];
2  GO
3  SELECT name AS [Name],
4         SCHEMA_NAME(schema_id) AS schema_name,
5         type_desc,
6         create_date,
7         modify_date
8  FROM sys.objects
9  WHERE modify_date > GETDATE() - 60
10 ORDER BY modify_date;
11 GO
```

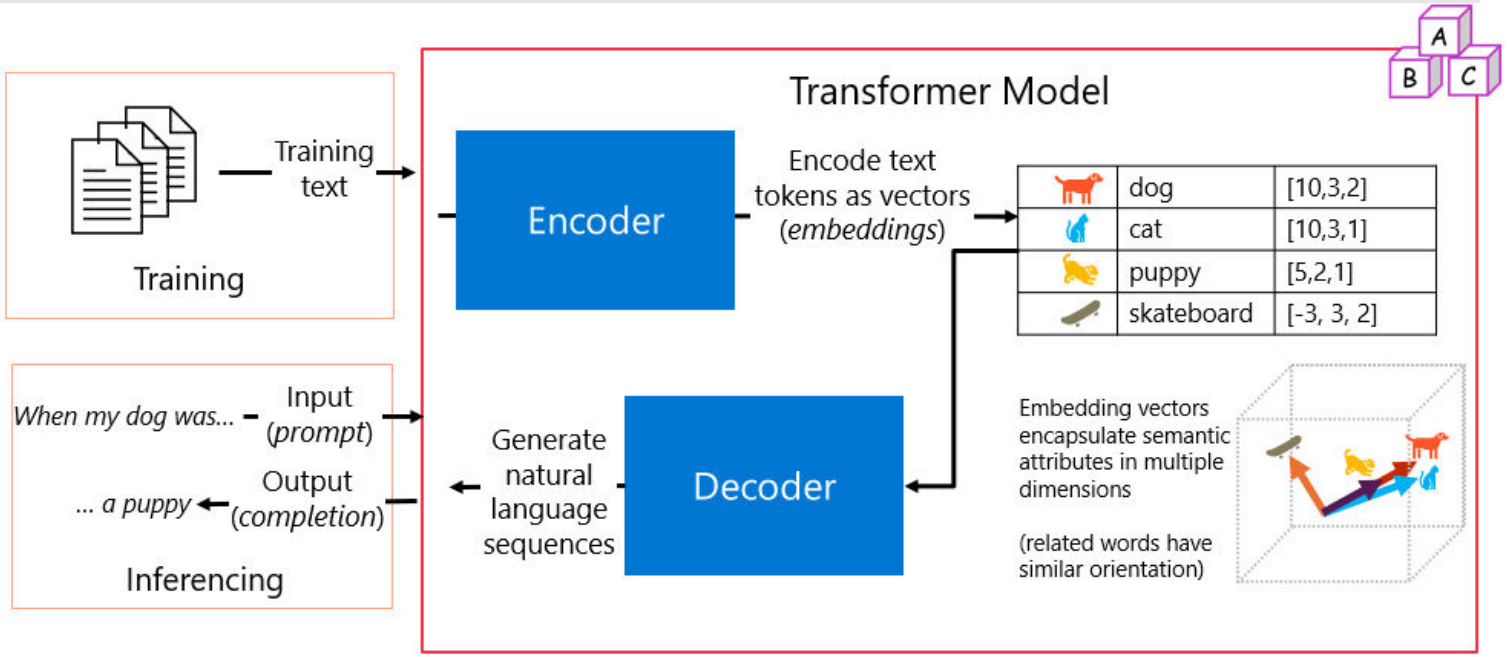
150 %

Results | Messages

	Name	schema_name	type_desc	create_date	modify_date
1	demotable	dbo	USER_TABLE	2020-05-02 12:59:16.657	2020-05-02 12:59:16.657
2	ufnToRawJsonArray	dbo	SQL_SCALAR_FUNCTION	2020-05-05 18:03:47.060	2020-05-05 18:03:47.060

- WHERE clause
- LIKE
  - Wildcard patterns
- Full-text search
  - CONTAINS
- Search text, NOT meaning

High level overview Language models



- Find semantic relationships between words
  - Use to predict probable sequences of text that make sense
- Different encoders exist
  - Split text into tokens
  - Transform tokens into vectors
    - multi-valued numeric representations

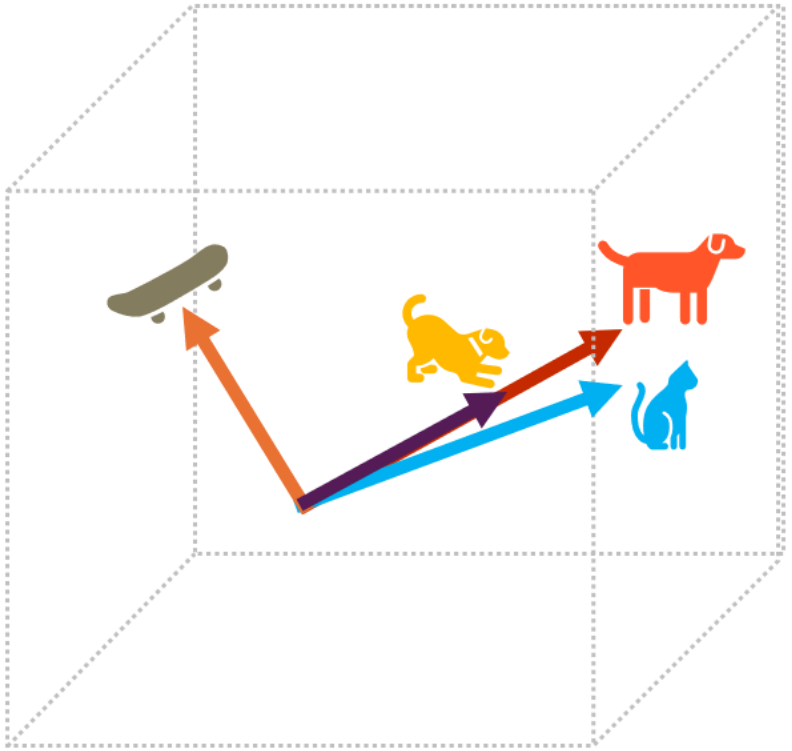
LM step 1: Step one: tokenization

Example sentence: *I heard a dog bark loudly at a cat.*

"I"=1	"heard"=2	"a"=3	"dog"=4	"bark"=5	"loudly"=6	"at"=7	"cat"=8

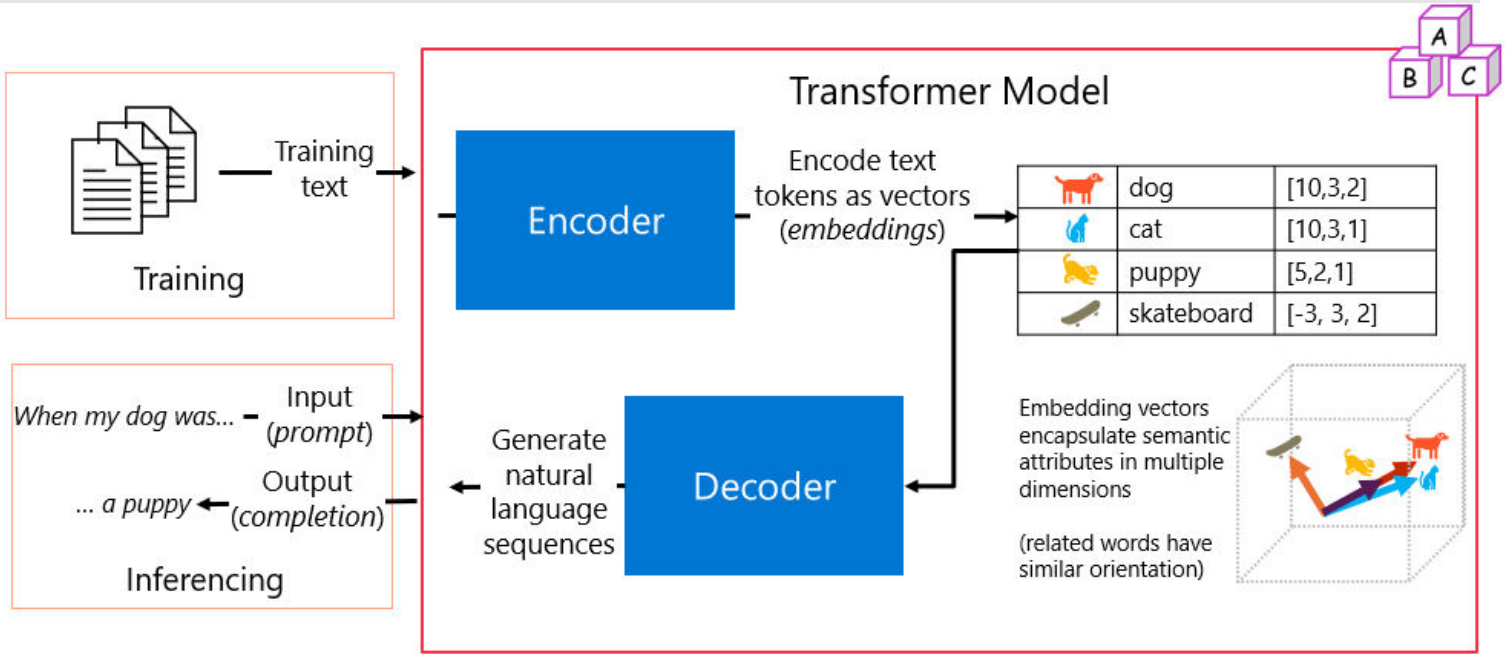
- The sentence is now represented with the tokens: *[1 2 3 4 5 6 7 3 8]*.
- Note "a" is tokenized as 3 only once.
- Similarly, the sentence "I heard a cat" could be represented with the tokens *[1 2 3 8]*.

LM step 2: Embeddings



	Token	Word	Embedding
	4	dog	[10,3,2]
	8	cat	[10,3,1]
	9	puppy	[5,2,1]
	10	skateboard	[-3, 3, 2]

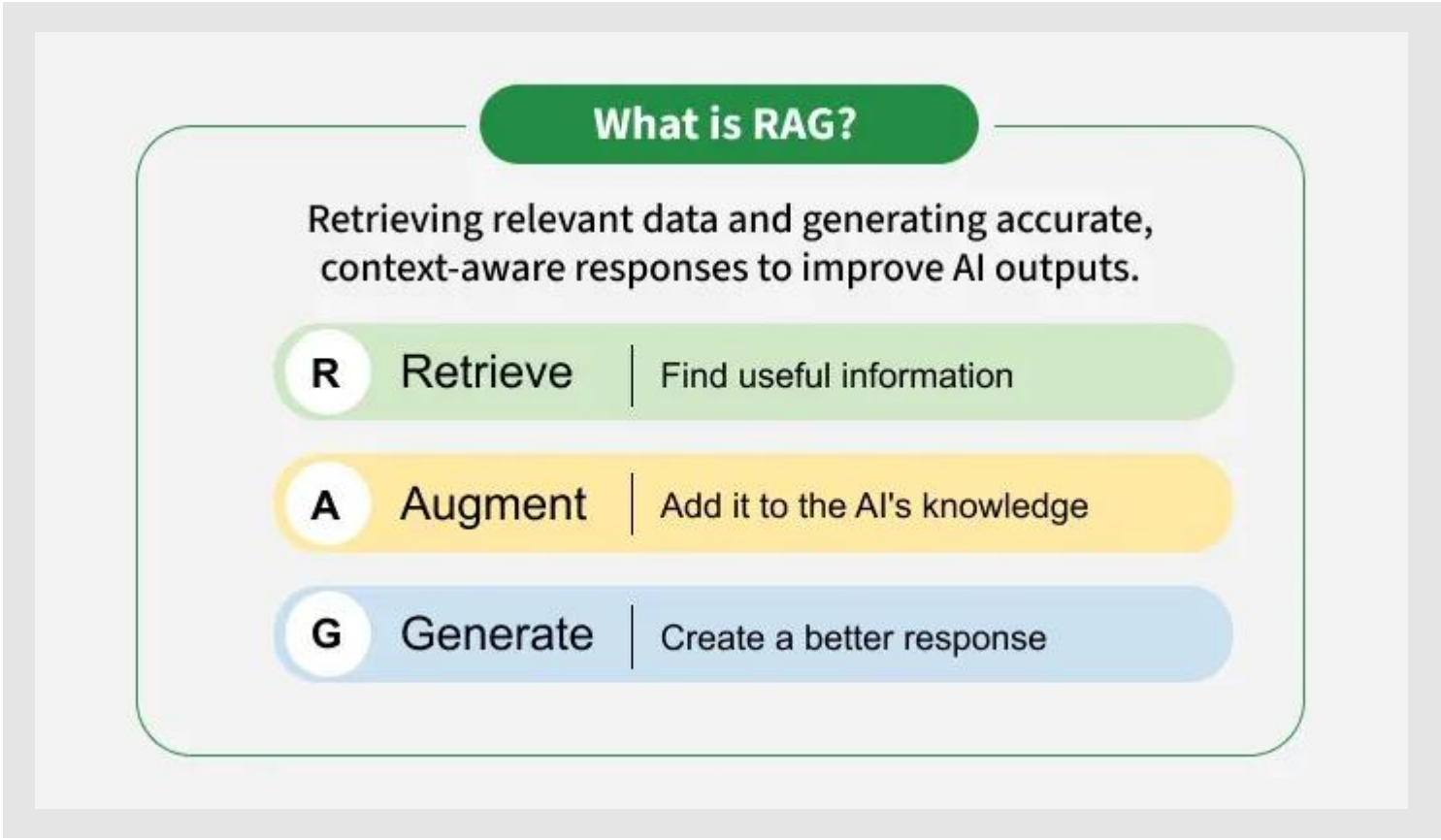
SQL Server / Fabric SQL database



- Create table storing embeddings
- Generate embeddings

- Usage
  - Create embedding for user query
  - Execute similarity search
    - VECTOR_DISTANCE

RAG



- Retrieval Augmented Generation

- Create embedding from prompt
- Use vector search in SQL to retrieve relevant data
- Add result of vector search to prompt
- Generate response

Demo embeddings

Create + Test Fabric SQL Database with embeddings

Demo

DAG from Python notebook

T-SQL

```
DECLARE @SearchVector VECTOR(1536);

SET @SearchVector = AI_GENERATE_EMBEDDINGS(
            'I am looking for a hot spicy tea beverage'
            USE MODEL EmbeddingsDemoTextEmbedding3Small);

-- Step 2: Run similarity search:
SELECT TOP 5
    ProductName
    , ProductDescription
    , VECTOR_DISTANCE('cosine', ProductVector, @SearchVector) AS Score
FROM dbo.ProductEmbeddings
ORDER BY Score ASC;
```

Steps

```
-------------------------------------------------------------------------------
-- 2. Create credential for Azure OpenAI
-------------------------------------------------------------------------------

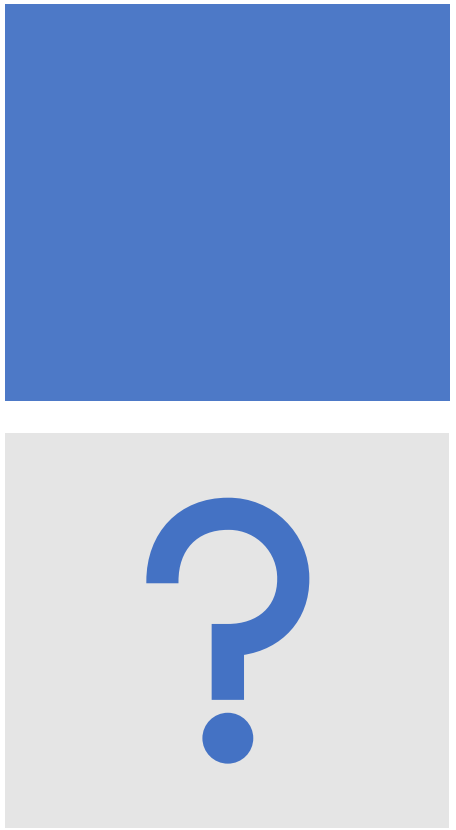
CREATE DATABASE SCOPED CREDENTIAL
    [https://embeddingsdemo-resource.openai.azure.com/]
WITH
    IDENTITY = 'HTTPEndpointHeaders',
    SECRET = '{"api-key":"<paste your key here>"}';
GO
GO


-------------------------------------------------------------------------------
-- 3. Create external model
-------------------------------------------------------------------------------

CREATE EXTERNAL MODEL [EmbeddingsDemoTextEmbedding3Small]
WITH
(
    LOCATION = '<paste URL to Foundry deployment here>'
    , API_FORMAT = 'Azure OpenAI'
    , MODEL_TYPE = EMBEDDINGS
    , MODEL = 'text-embedding-3-small'
    , CREDENTIAL = [https://embeddingsdemo-resource.openai.azure.com/]
);
GO
```

- Create master key
- Create database scoped credential
  - Name is url
- Create external model
  - AI is not running inside SQL engine
  - Model can be
    - Web hosted
    - Locally installed

Questions ?



Thank you

Peter ter Braake

trainsql@live.nl