



# Microsoft Ignite





# PREL14: Unify data, act on insights, and build AI solutions with Microsoft Fabric

Brad Schacht - Principal Program Manager, Fabric CAT  
John Hoang - Principal Program Manager, Fabric CAT



# John Hoang

Principal Program Manager, Fabric CAT  
Orange County, CA



John.Hoang@microsoft.com

Part of the DATAAllegro acquisition

Data warehousing specialist



@MSJohnPaulHoang

Plays tennis and pickleball

Have 4 kids 2 boys/2 girls



in/johnhoang869773299

Sings at local nursing home



<https://aka.ms/FabricIdeas>

# Brad Schacht

Principal Program Manager, Fabric CAT  
Saint Augustine, FL



Brad.Schacht@microsoft.com

Former consultant, trainer and cloud solution architect



<https://BradleySchacht.com>

Data warehousing and Microsoft BI



@BradleySchacht

I like pizza (Absolutely NO pineapple!!)



BradleySchacht.bsky.social

Member of The Church of Eleven22 in Jacksonville, FL



in/BradleySchacht

My wife is amazing. My two boys are ~~crazy~~ awesome.



Tales From the Field

Speak at SQL and BI events all over the place



<https://aka.ms/FabricIdeas>

Coauthor of 6 SQL Server and BI books

(Worth a read if you're having a hard time sleeping/adjusting to the time change)

Atlanta Braves baseball fan (It was a sad season...again)

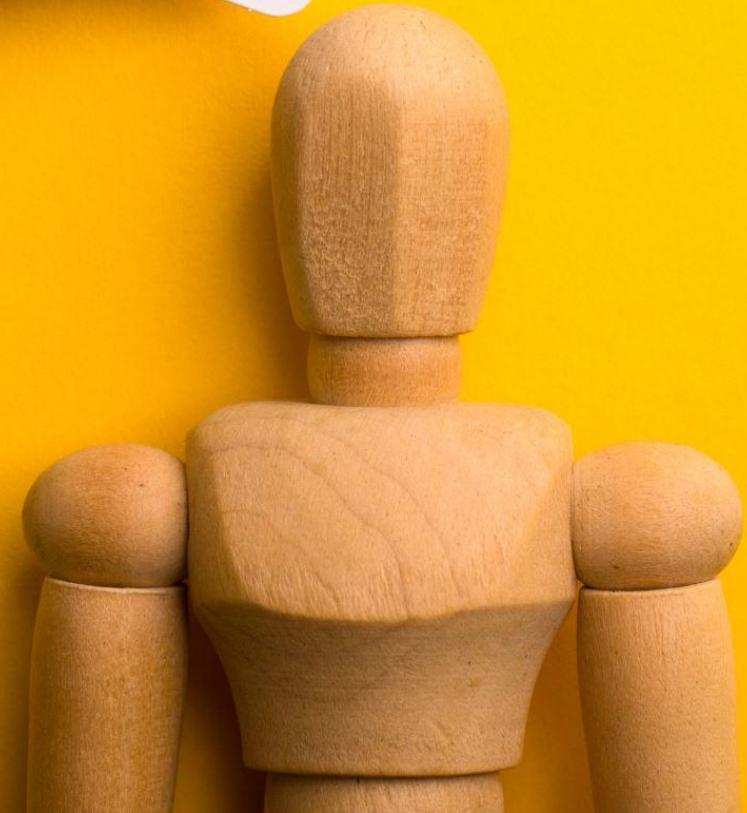
**Restrooms**

**Breaks**

**Questions**

**Pre-day content**

**Session feedback**



# How did we do?

Tell us your thoughts  
about our sessions and  
complete the event survey



# Resources

Computer

Fully equipped lab environment

- Hosted on Skillable

- Virtual machine with local user

- Entra user for Fabric

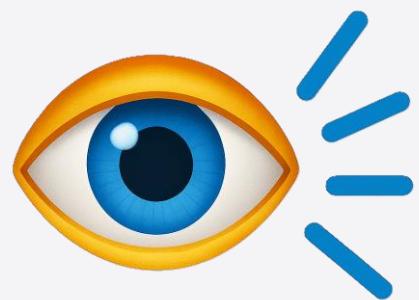
- Fabric capacity

Lab content hosted on GitHub



# What is your learning style?

Visual



Auditory



Tactile



# Agenda

01

Introduction to Fabric

02

Data Integration in Fabric

03

OneLake, Lakehouse, and Spark

04

Data warehouse

05

Data Agents

06

Power BI and Copilots

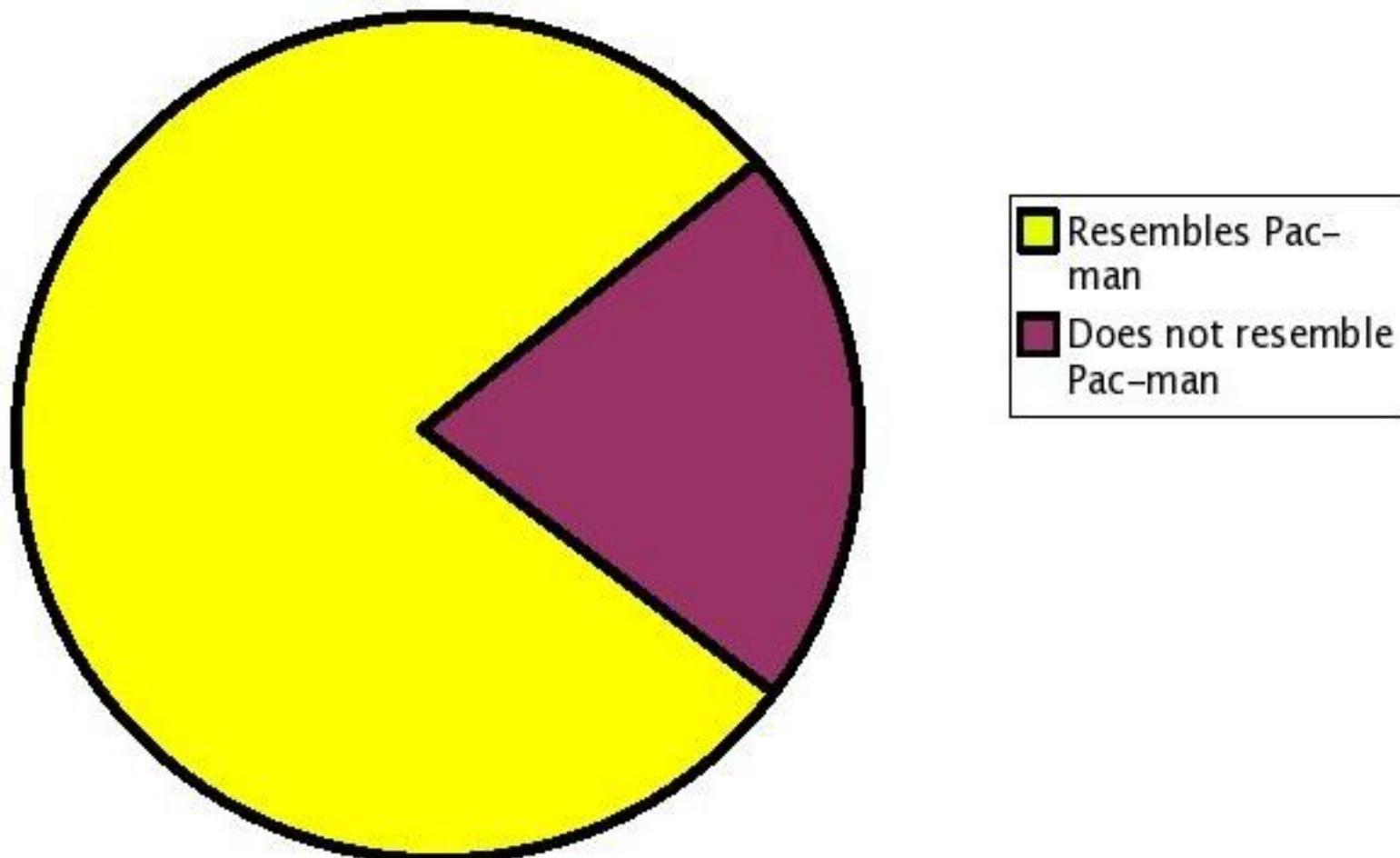
07

Architecture patterns

# Introduction to Fabric



## Percentage of Chart Which Resembles Pac-man



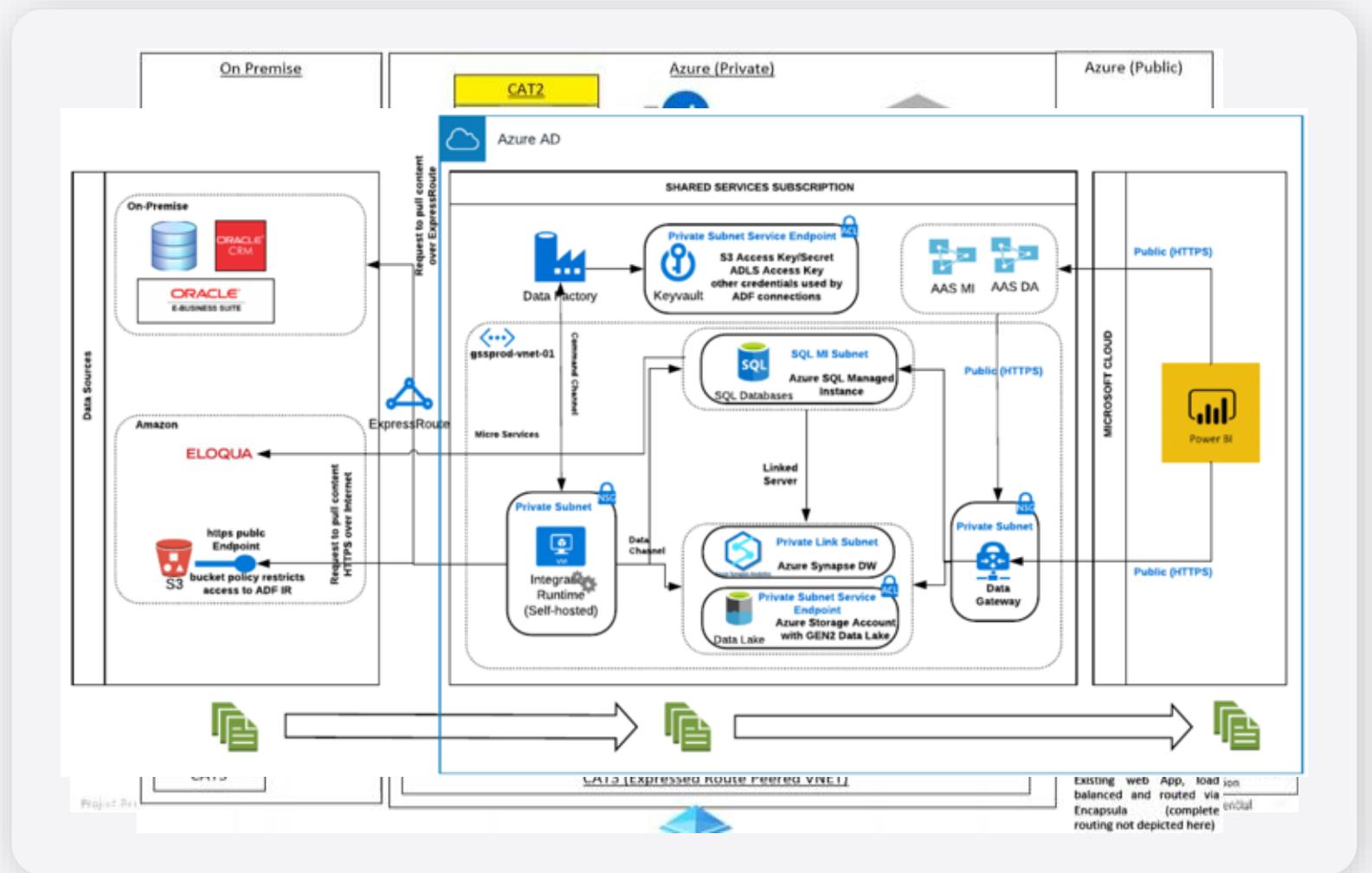
# Analytics is complex and fragmented

Every project has many subsystems

Every subsystem need a different class of product

Products often comes from multiple vendors

Integration is complex, fragile and expensive



# Microsoft Fabric

## The unified data platform for AI transformation



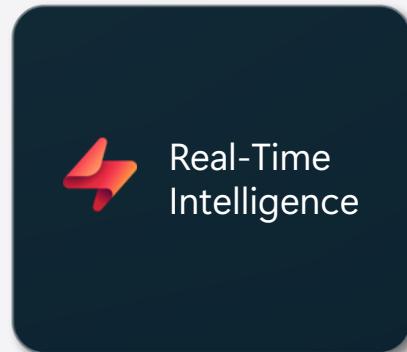
Data  
Factory



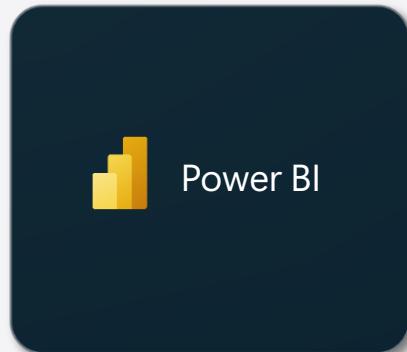
Analytics



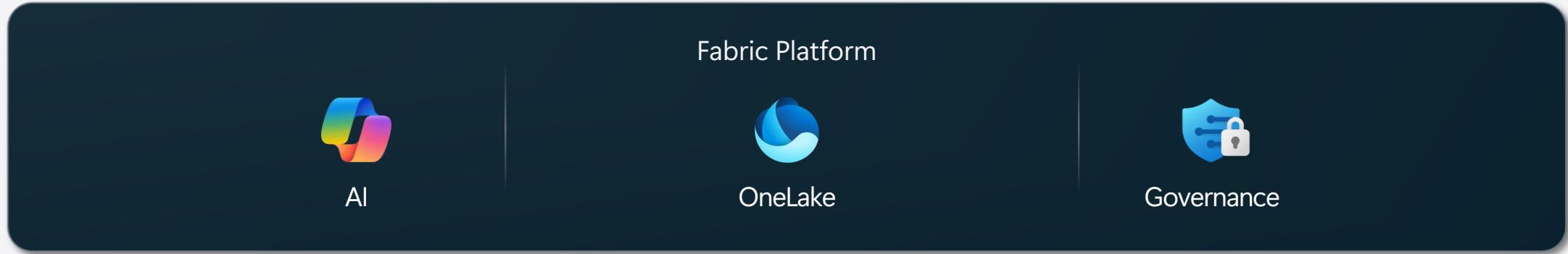
Databases



Real-Time  
Intelligence



Power BI



# The data platform for the era of AI

## Complete Analytics Platform

Unified product, experience, and architecture

Delivered as SaaS

## Lake Centric and Open

Common SaaS data lake shared by all compute engines

Deep commitment for open formats and APIs

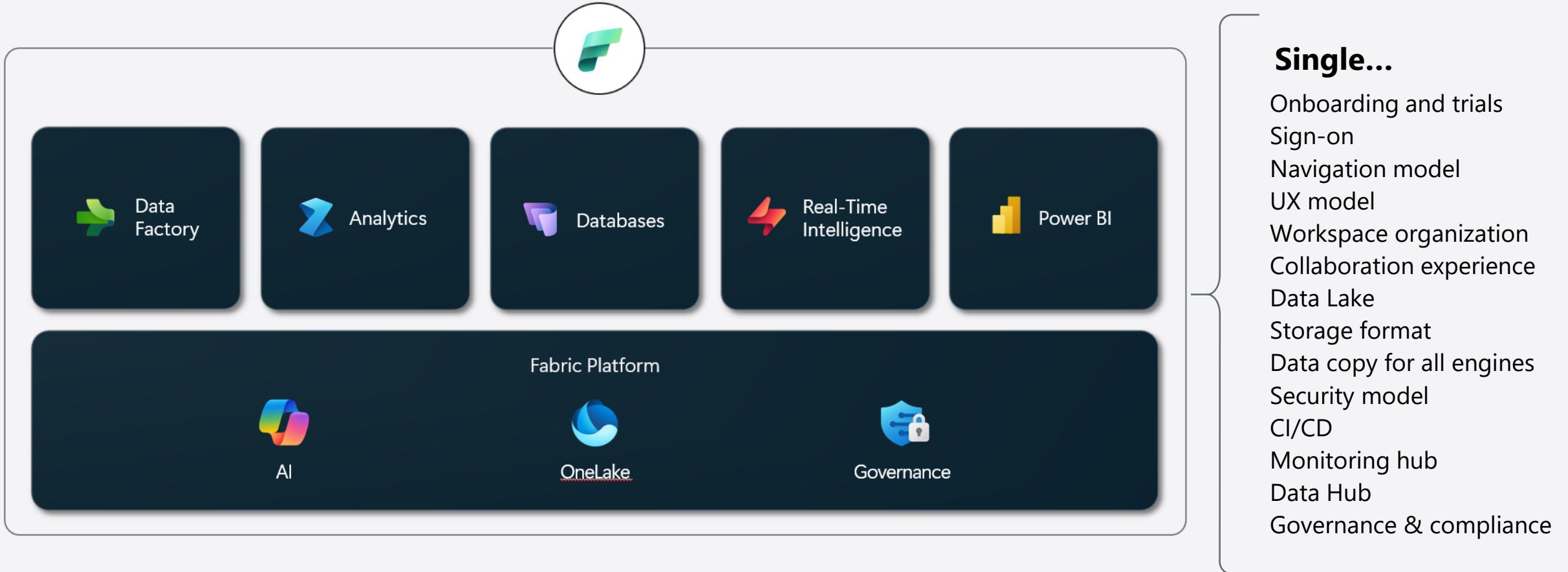
## Empower Every Business User

Deliver data directly to the users in their favorite Office applications

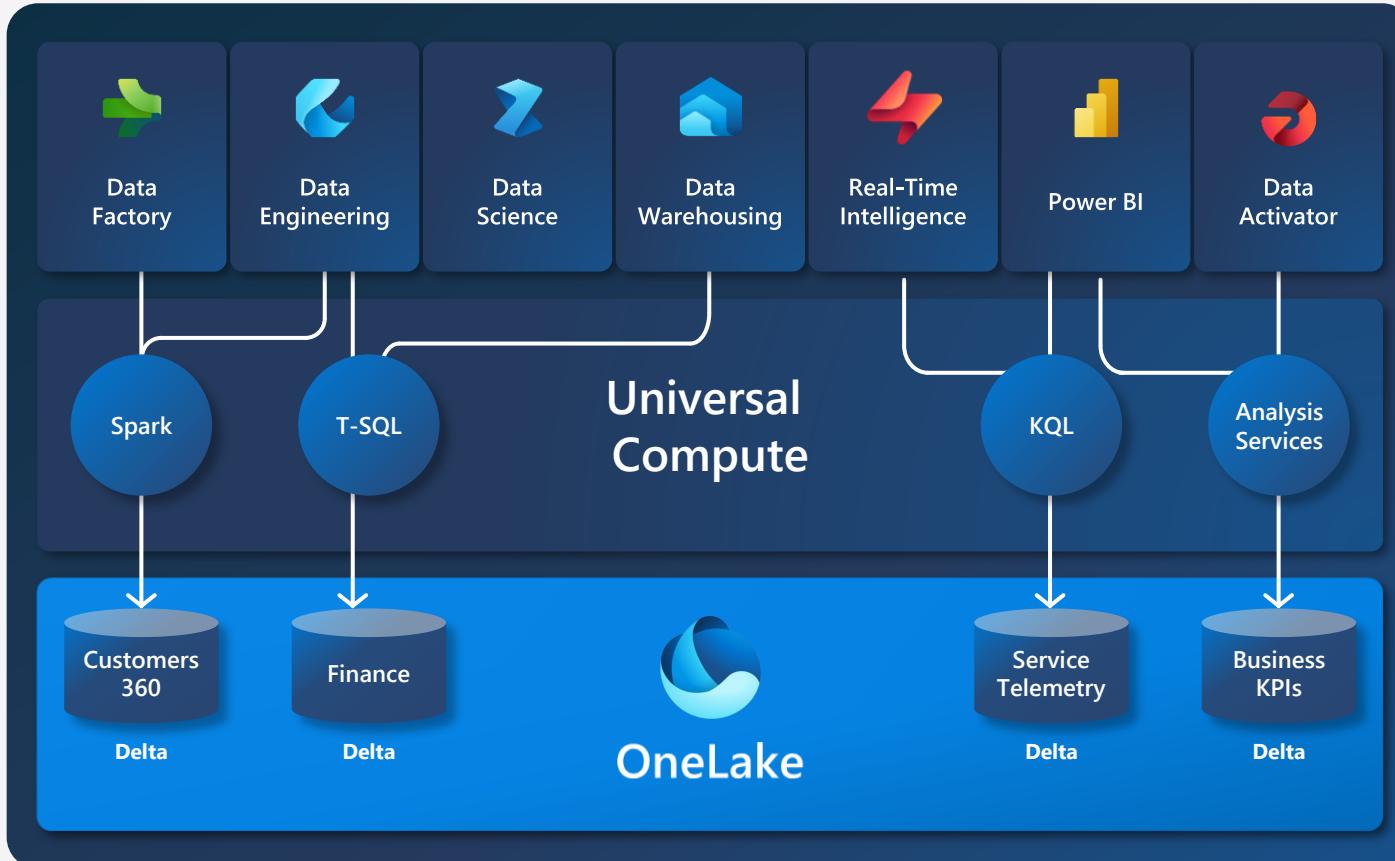
## AI Powered

Designed from the ground up for AI

# Microsoft Fabric



# Microsoft Fabric



All the compute engines store their data automatically in OneLake

The data is stored in a single common format

[Delta](#), an open standards format, is the storage format for all tabular data in Analytics vNext

Once data is stored in the lake, it is directly accessible by all the engines without needing any import/export

All the compute engines have been fully optimized to work with Delta Parquet as their native format

Shared universal security model is enforced across all the engines

# AI-powered experiences in Fabric

Gen AI accelerates your data journey



Copilot accelerated  
experiences



Chat with your data  
experiences



Custom generative  
AI for your data

# Fabric topology and organization

# Fabric Tenant

## Capacity 1 – North Europe

### Workspace 1



Data  
Factory



Data  
Engineering

## Capacity 2 – East US

### Workspace 3



Data  
Science

## Capacity 3 – East US

### Workspace 2



Data  
Warehousing



Power BI



Data  
Activator

### Workspace 4



Data  
Engineering



Data  
Warehousing



Power BI

## OneLake

# Comparing P and F SKUs

<b>SKU</b>	<b>Capacity Units (CU)</b>	<b>Power BI SKU</b>	<b>Power BI v-cores</b>
F2	2	-	0.25
F4	4	-	0.5
F8	8	EM/A1	1
F16	16	EM2/A2	2
F32	32	EM3/A3	4
<b>F64</b>	<b>64</b>	<b>P1/A4</b>	<b>8</b>
F128	128	P2/A5	16
F256	256	P3/A6	32
F512	512	P4/A7	64
F1024	1024	P5/A8	128
F2048	2048	-	256

# **Getting started with Fabric**

# Getting started with Fabric in your environment

## 1. Enable Fabric at the tenant level

- Override the tenant level setting at the capacity
- Enable for only specific security groups

Users can create Fabric items

*Enabled for the entire organization*

Users can use production-ready features to create Fabric items. Turning off this setting doesn't impact users' ability to create Power BI items. This setting can be managed at both the tenant and the capacity levels. [Learn More](#)



Enabled

## 2. Get capacity

- Start a 60-day free trial
- Use existing P SKU
- Provision an F SKU

Users can try Microsoft Fabric paid features

*Enabled for the entire organization*

When users sign up for a Microsoft Fabric **trial**, they can try Fabric paid features for free for 60 days from the day they signed up. [Learn More](#)



Enabled

## 3. Create a workspace and assign a capacity

## 4. Build something cool!

# Preparing to use data agents

1. Enable the data agent feature at the tenant level
2. Enable Copilot and Azure OpenAI features
3. If outside US or EU regions
  - Allow data to be processed outside capacity's geographic region
  - Allow data to be stored outside capacity's geographic region

The screenshot shows the Microsoft Admin portal interface. On the left, a sidebar lists various tenant settings like Usage metrics, Users, Premium Per User, Audit logs, Domains, Workloads, Tags, Capacity settings, Refresh summary, Embed Codes, Organizational visuals, Azure connections, Workspaces, Custom branding, Fabric identities, Featured content, Microsoft Purview setting, Help + support, and Data Policies. The 'Tenant settings' tab is selected.

In the main content area, under the 'Microsoft Fabric' section, there is a message: "Users can create and share **Data agent** item types (preview) Enabled for the entire organization". A red box highlights this message and the 'Enabled' toggle switch below it. The 'Enabled' toggle is turned on (green). Below the toggle, there are options to "Apply to:" "The entire organization" (radio button selected), "Specific security groups", and "Except specific security groups". There is also a checkbox for "Delegate setting to other admins" and another for "Capacity admins can enable/disable". At the bottom are "Apply" and "Cancel" buttons.

# Data Integration

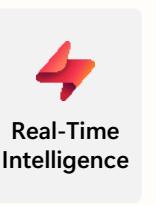


# Data Factory workload

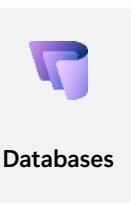
Dataflows and data pipelines bring together low-code, AI-based experiences, multi-cloud connectivity, and persistent data security and governance to help solve complex ETL scenarios for all developers



Data Factory



Real-Time  
Intelligence



Databases



Data  
Engineering



Data  
Warehouse



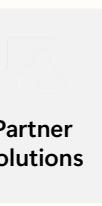
Data  
Science



Industry  
Solutions



Power  
BI



Partner  
Solutions



170+ native data source connectors



300+ data transformations in dataflows designer to transform data more easily



Cloud-scale data movement with Data Factory



Low-code interface for ingesting data from hundreds of data sources using Dataflows Gen2



Out-of-the-box rich data orchestration capabilities to compose flexible workflows



Powerful, enterprise-grade Data Factory workload with the best of ADF and Power Query together



Copilot in Microsoft Fabric



OneLake



Security, Governance and Administration with Purview





# Data Factory



Microsoft  
Azure

Enterprise data  
integration in Azure

- Azure Data Factory
- Synapse Link
- Purview Integrated



Self-service data prep  
with Power Query

- Excel
- Power BI
- Office 365
- Dynamics 365
- Power Platform



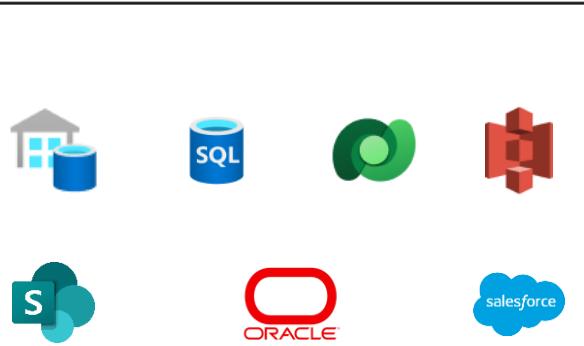
Microsoft  
Fabric

Fabric  
Data Factory

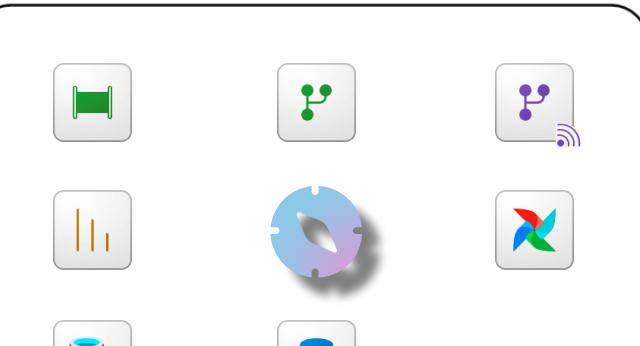
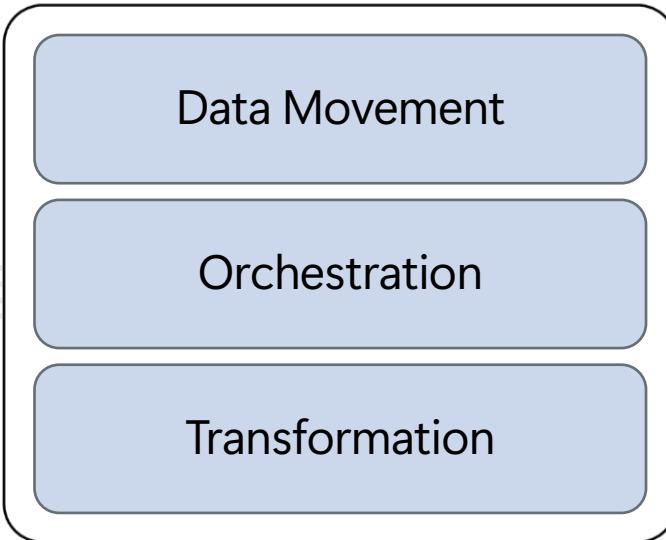
- Integrated SaaS platform
- No-code and code tools
- Copilot and AI-powered
- Open Source



# Data Factory



Best-in-class connectivity



Deployment & observability



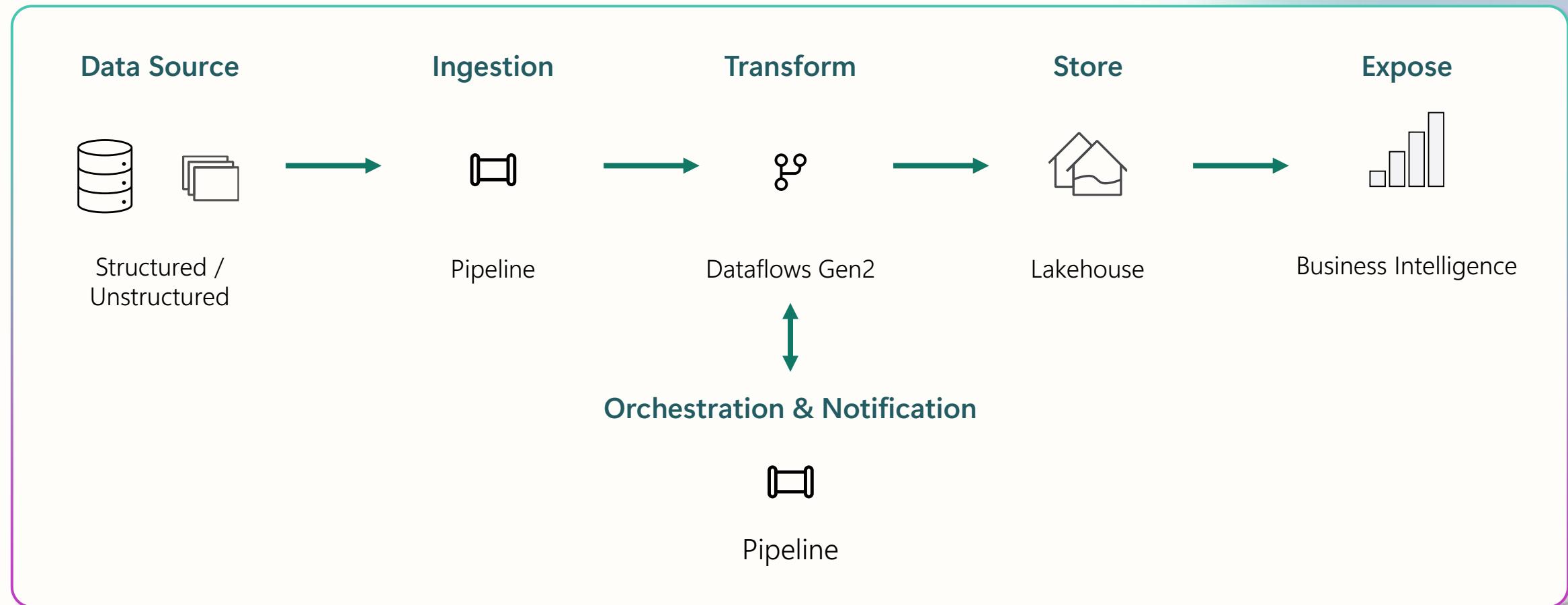
OneLake

AI-powered intelligence



# Data Factory scenario

End-to-end analytics scenario





# Data Factory

Provide cloud-scale data movement, workflow orchestration, and data transformation services

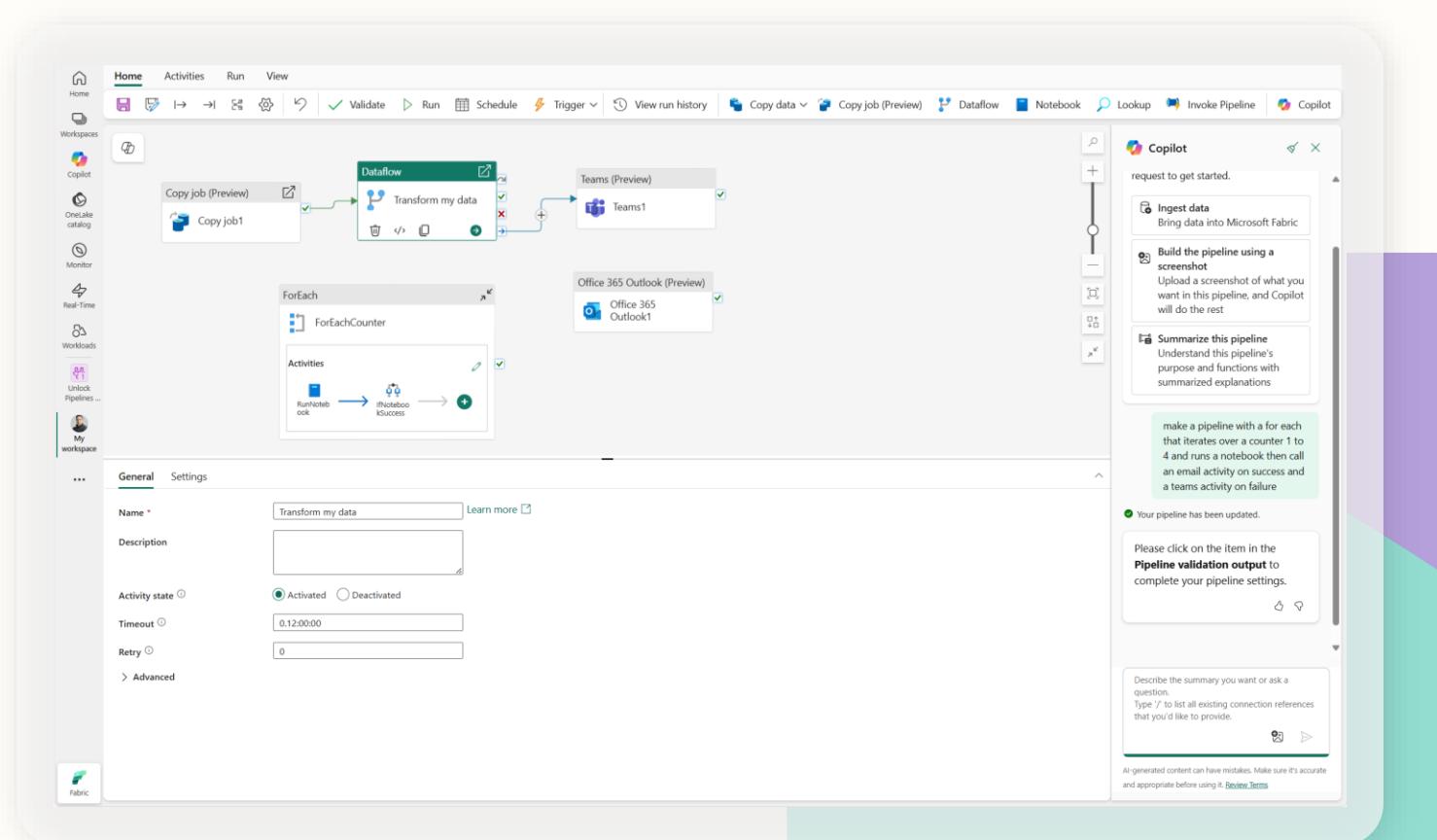
Consisted of Data Pipelines and Dataflows to give users the option to a low-code, collaborative and enterprise scale approach for their ETL process.

Support for both low-code and code-first data integrators

170+ connectors available in Data Factory

Built-in monitoring and lineage across all workspace items

Automated scheduling, event-based triggering, and real-time data movement



*AI-generated ETL using Data Factory co-pilot unlocks operational efficiencies to help orchestrate, monitor and manage pipeline performance.*



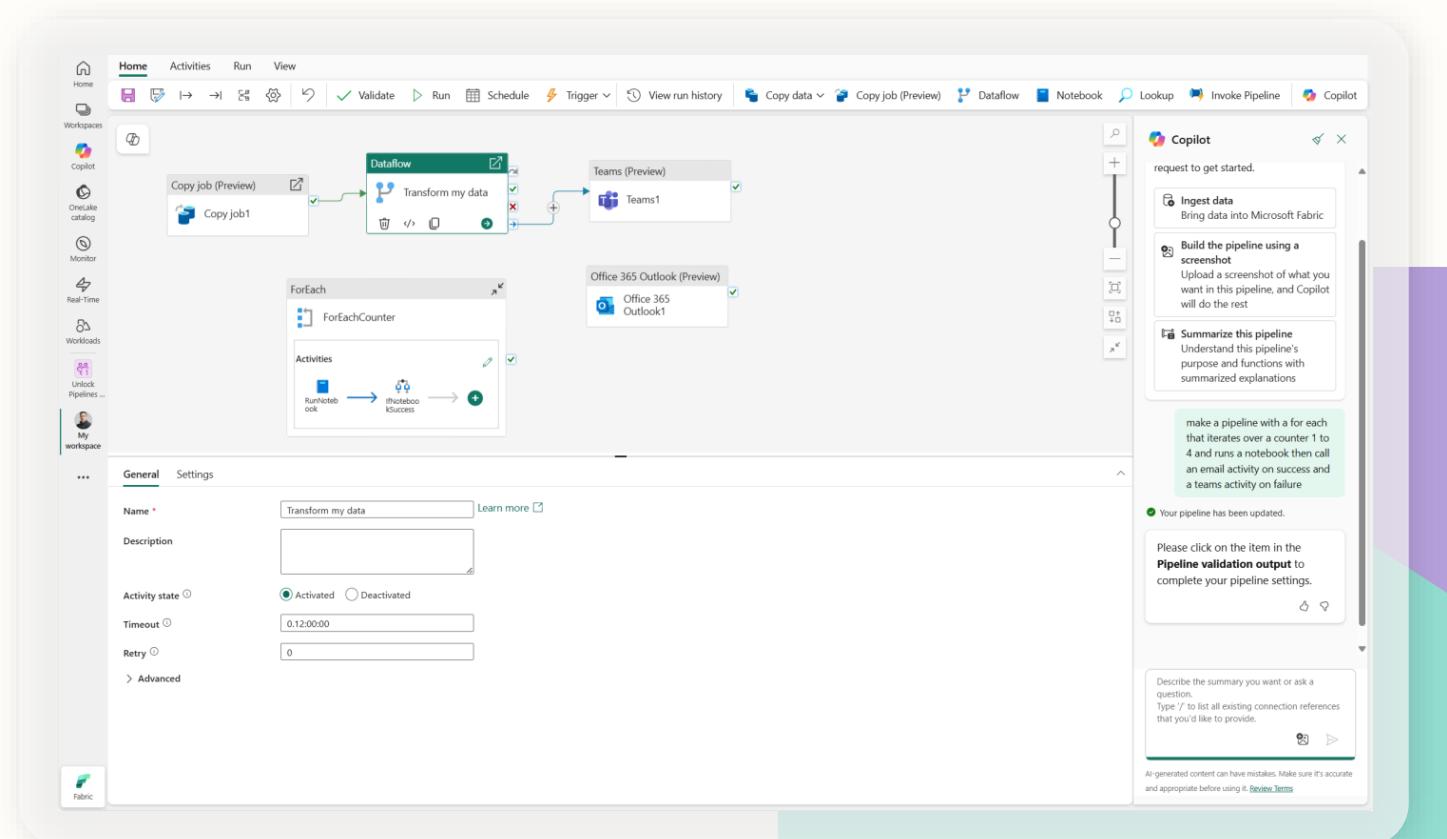
# Data Pipelines

**Data Pipelines** enable powerful workflow capabilities

Building complex workflows, moving PB-size data, and defining sophisticated control flow pipelines.

Build complex ETL and data factory workflows that can perform multiple different tasks at scale.

Control flow capabilities are built into pipelines so you can build workflow logic which provide loops and conditional execution.





# Connectors

**New Connectors** provide a low-code interface for ingesting data from a variety of data sources.

## Connectors:

- Fabric native Connectors
- Connect to Azure and non-Azure resources
- Integrated with modern get data experience
- 170+ connectors across Data Factory
- Access to on-premises data
- Access protected data inside of a VNET

The screenshot shows the Microsoft Fabric Data Factory interface with a 'Copy data' wizard open. The left sidebar includes icons for Home, Activities, Run, View, Create, Browse, Data Hub, Messaging hub, Workspaces, Visual Studio, Pipelines, and Dataflows. The main area has tabs for Home, Activities, Run, View, Validate, Run, Schedule, View run history, Copy data, Dataflow, Notebook, Hookup, and Invoke pipeline. The 'Copy data' dialog is centered, showing the first step: 'Choose data source'. It displays a message: 'Build your data ingestion task to move objects from a data source to a data destination. Learn more [?]. holidays package and Wikipedia, covering 18 countries or regions from 1970 to 2009.' Below this is a 'Data sources' grid with categories: All categories, Workspace, Azure, Database, File, Generic protocol, Services and apps. The grid lists various connectors: Amazon RDS for SQL Server Database, Amazon Redshift Database, Amazon S3 File, Amazon S3 Compatible File; Apache Impala Database, Azure Blob Storage, Azure Cosmos DB for NoSQL, Azure Data Explorer (Kusto); Azure Data Lake Storage Gen1, Azure Data Lake Storage Gen2, Azure Database for PostgreSQL, Azure SQL Database; Azure SQL Database Managed Instance, Azure Synapse Analytics, Azure Table Storage, Data Warehouse Workspace; Databricks Services and apps, Dynamics CRM Services and apps, Google Cloud Storage File, Hive Database; HTTP Generic protocol, KQL Database Workspace, LinkedIn Database, Microsoft 365 Services and apps; OData Generic protocol, PostgreSQL Database, REST Generic protocol, Services and apps, SharePoint Online List Services and apps; Snowflake Services and apps, Spark Database, SQL server Database. At the bottom of the dialog are 'Back', 'Next', and 'Cancel' buttons.



# Copy job

**Copy job** simplifies data movement with built-in batch and incremental copy

Including change data capture (CDC) replication patterns.

It delivers petabyte-scale performance and reliability, connecting on-premises and cloud sources through secure, compliant access.

With integrated scheduling, monitoring, and fine-grained control in a no-code experience, Copy Job accelerates building data integration solutions while ensuring security, governance, and operational efficiency.

The screenshot shows the Microsoft Fabric interface with the 'copyjob2' workspace selected. A modal window titled 'Copy job' is open, showing a step-by-step process: 'Choose data source' (selected), 'Choose data', 'Choose data destination', 'Map to destination', 'Settings', and 'Review + save'. In the background, the 'OneLake catalog' tab is active, displaying a grid of data connectors categorized by type (File, Database, Power Platform, Azure, Online services, Other). Some visible connectors include SQL Server database, MySQL database, PostgreSQL database, Oracle database, SAP HANA database, Snowflake, Vertica, Dataverse, Azure Synapse Analytics (SQL DW), Azure Blob storage, Azure Data Lake Storage Gen2, SharePoint Online list, ODBC, Oracle Cloud Storage, Amazon RDS for Oracle, and Azure Database for PostgreSQL.



# Apache Airflow

**Apache Airflow jobs** provide a managed orchestration environment within Data Factory, enabling Python DAG workflows at scale without infrastructure management.

Ideal for teams that prefer code-first orchestration with flexible scheduling and logic.

Orchestrates Python-based DAG with full Apache Airflow compatibility (operators, plugins, etc.)

Provides near-instant runtime startup to begin running workflows in seconds

Supports Git sync, Azure Key Vault, and custom package installs for secure, flexible development

Runs as a fully managed service in Fabric – no infrastructure to manage

Start building your Apache Airflow jobs by creating DAGs and plugins



New DAG file

Define and schedule Apache Airflow jobs or pipelines



New plugin file

Define additional operators, hooks, and/or executors



Run Fabric Artifact

Create a DAG to run Fabric pipelines, notebooks, etc



# Dataflow

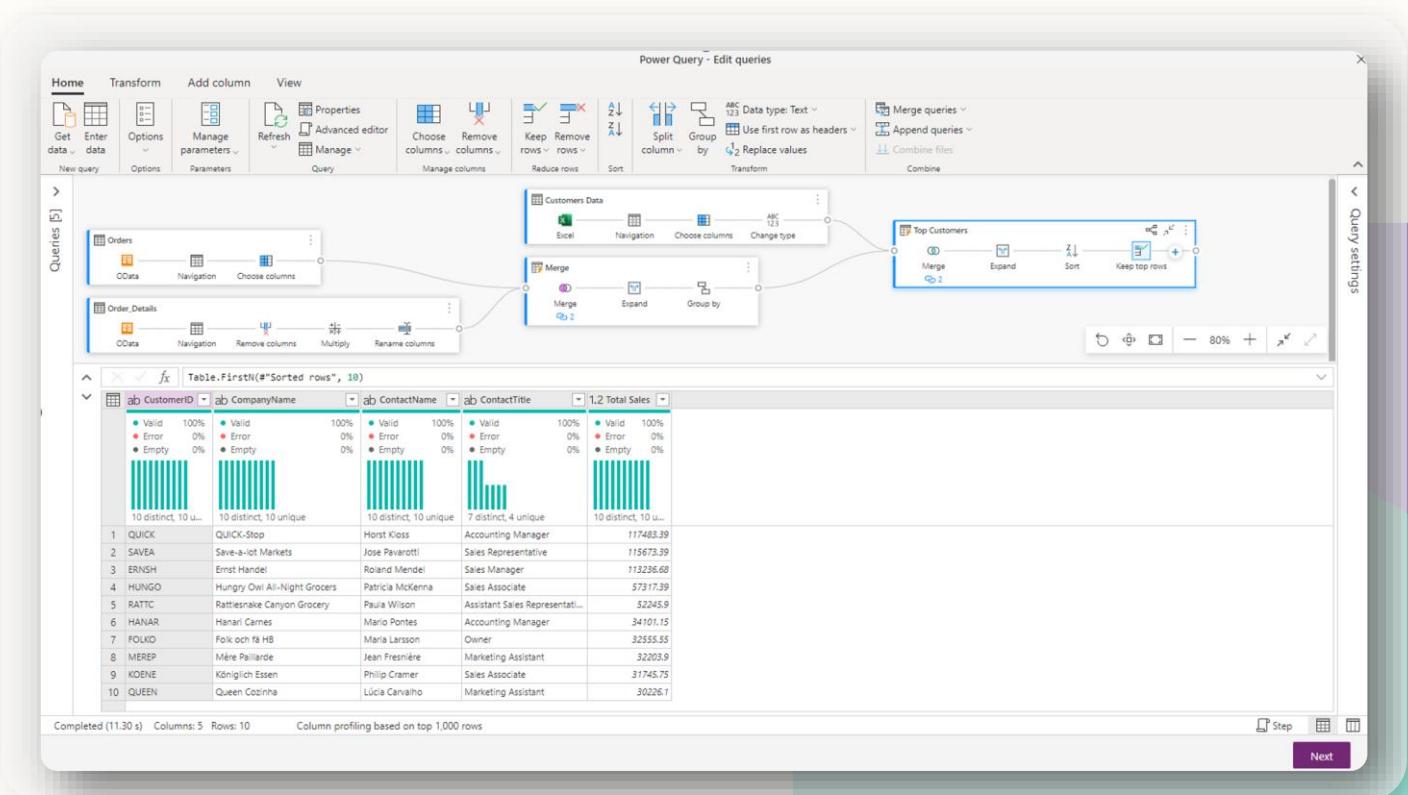
**Dataflow** provides a low-code interface for ingesting & transforming data from hundreds of data sources.

Dataflows quickly and easily unify disparate data sources, establish a more collaborative analytics approach, and promote more informed, agile decision making.

Accelerate data transformation with code-free dataflows

Achieve high-scale ingestion & transformation using Fabric compute and Data Factory fast copy

Load results of data transformations into multiple destinations (Azure SQL Databases, Lakehouse, SharePoint, etc.)





# Dataflow | Output to Lakehouse

Simply write into a Lakehouse from a Dataflow.

Users select the Lakehouse output destination from the list and configure the connection.

Dataflow output supports both fixed and dynamic schema, giving flexibility for structured or evolving datasets.

When writing to a Lakehouse, you can choose to append new rows or replace existing ones, ensuring the right balance between historical preservation and data refresh.

These options make Lakehouse a versatile target for BI, AI, and Data Science workloads across Fabric.

The screenshot shows the Microsoft Power Query Dataflow interface. In the center, there's a table view of data with columns like 'CustomerKey', 'CustomerID', 'CustomerName', etc. At the top, the 'Add data destination' dropdown is open, and 'Lakehouse' is highlighted with a red box. To the right of the table, there are various transformation tools and a 'Query settings' pane. The 'Entity type' is set to 'Customer' and 'Applied steps' include 'Promoted columns' and 'Changed values'. The bottom right corner has a 'Publish' button.



# Fast copy

**Fast copy** in Dataflow Gen2 is a high-performance ingestion mode that optimizes large-scale data loading by switching to a powerful petabyte-scale backend engine automatically when dataset size surpasses a threshold

Automatically switches to a faster engine for high-volume data

Preserves your existing dataflow, no manual refactoring needed

Supports sources like ADLS Gen2, Blob Storage, Azure SQL DB, Lakehouse, Warehouse, Oracle, Snowflake, PostgreSQL, Fabric SQL DB and more

Optimized for loading directly to Lakehouse

The screenshot shows the Microsoft Power BI Dataflow interface. At the top, there's a navigation bar with tabs for Home, Transform, Add column, View, and Help. Below the navigation bar is a toolbar with various icons for data operations like Get data, Enter data, Manage connections, Options, Manage parameters, Refresh, Advanced editor, Properties, Manage data destination, Choose columns, Remove rows, Keep rows, Filter rows, Reduce rows, Suggested transforms column, Split by, Group by, Use first row as headers, Replace values, Sort, Merge queries, Append queries, Combine files, Map to entity, CDM, Insights, Copilot, and Export template.

The main area is titled "Queries [5]" and shows a flow of data steps. It starts with a "File(helper query...)" step, followed by three "Transform file from Sample file" steps, each with two steps. These are connected to a final "Transform file (2)" step, which has one step. This final step connects to a "Sample file" step, which is defined as the "Source". The "Query settings" pane on the right shows properties for the "Sample file" source, including "Name: Sample file", "Entity type: Custom", and a list of applied steps: Source, Navigation, Filtered hidden..., Invoke custom..., Renamed..., Removed..., Expanded..., and Changed... (with 123 instances).

At the bottom, a preview pane displays a table titled "Table.TransformColumnTypes#\"Expanded table column\"". The table has 10 columns and 11 rows of data. The columns are: Source.Name, event\_time, event\_type, product\_id, category\_id, category\_code, brand, price, user\_id, and user\_session. The data includes entries for brands like apple, force, bosch, nika, and ikea, along with their respective product details and prices.



# Copilot Data Factory

**Copilot in Data Factory** uses natural language to build, transform, and troubleshoot pipelines with ease. It accelerates integration while reducing the need for complex coding, making data pipelines faster to create and easier to manage.

Use natural language to generate and edit powerful data pipelines

Apply data transformations without writing complex code

Get instant explanations and summaries of pipeline logic

Surface and resolve error messages with guided fixes

Accelerate pipeline development with an interactive, conversational interface

The screenshot displays the Microsoft Copilot Data Factory interface, which integrates AI-powered natural language processing into the Azure Data Factory pipeline creation process.

**Top Left:** A "Queries [5]" panel shows a hierarchical tree of five queries. The first query, "Customers", has three steps: "Employees", "Order\_Details", and "Products". The "Products" step is expanded, showing its OData source URL and a complex filter expression using LINQ-like syntax to select rows where the country is Germany, France, Spain, or United Kingdom.

**Top Right:** A "Copilot Preview" window shows a conversational interface. It starts with a message "What would you like to do? I can transform your data or explain how it's being transformed. Please try starting with this option to connect to your data: Get data". Subsequent messages show the user asking to "Only keep European customers.", and the AI responding with a suggested step: "Filter rows" with the query "Customers".

**Middle:** A "Copy data" dialog box is open, showing a green checkmark icon and a "copy0" connection name. Below it is a "Copy" pipeline configuration card with tabs for General, Source, Destination, Mapping, and Settings. The "Source" tab is selected, showing a "Connection" dropdown set to "demosql-blogdemo", "Connection type" as "Azure SQL Database", "Database" as "demo", and "Use query" options for "Table" (selected), "Query", and "Stored procedure".

**Bottom Right:** A "Copilot PREVIEW" window provides detailed explanations for the pipeline components. It states: "Source connection of 'copy0' (Copy) is 'demosql-blogdemo'" (AzureSqlDatabase); "Destination connection of 'copy0' (Copy) is 'LakehouseDemo'" (Lakehouse). It also lists the tables involved: "table of Source connection 'demosql-blogdemo' (AzureSqlDatabase) in 'copy0' (Copy) is SalesLT.Customer; table of Destination connection 'LakehouseDemo' (Lakehouse) in 'copy0' (Copy) is CustomerNew".



# Mirroring

**Mirroring** is a zero-ETL, no-cost, low-latency way to synchronize operational databases and unify them in OneLake.

Data instantly available as analytic-ready Delta tables for BI and AI.

Supports major databases such as Azure SQL, Cosmos DB, Oracle, Google Big Query, and Snowflake

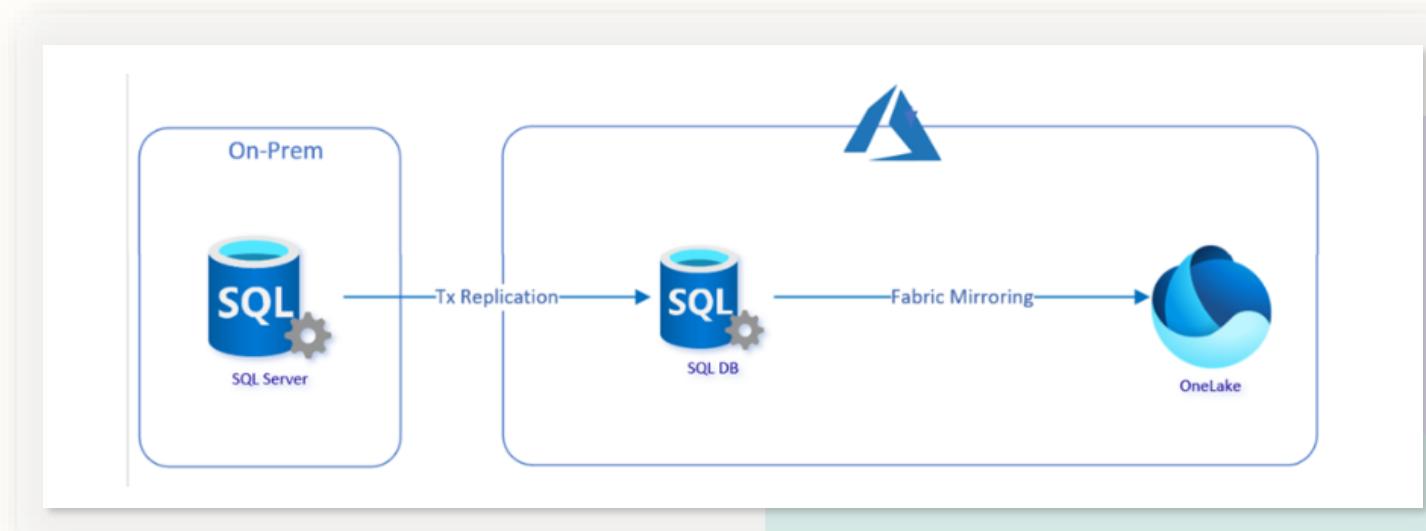
Uses Change Data Capture (CDC) to keep data in sync with the source system

Supports both full database mirroring and metadata mirroring (virtual access via shortcuts)

Eliminates ETL, extra compute, and pipeline maintenance

Delivers near real-time, analytics-ready data to Fabric services like Power BI, Spark, and notebooks

Free offering covering both compute and storage up to a certain limit (1TB/ CU)





# Open mirroring

**Open mirroring** is an extensibility platform that empowers developers and ISVs to build their own custom Mirroring source.

Supports files converted from proprietary formats into a parquet or CSV format via API and through a drag-and-drop UI experience.

Build your own mirrored database or use one built by an ISV partner (Striim, CData, MongoDB, etc).

Acts as an API surface for building custom mirroring sources and integrations

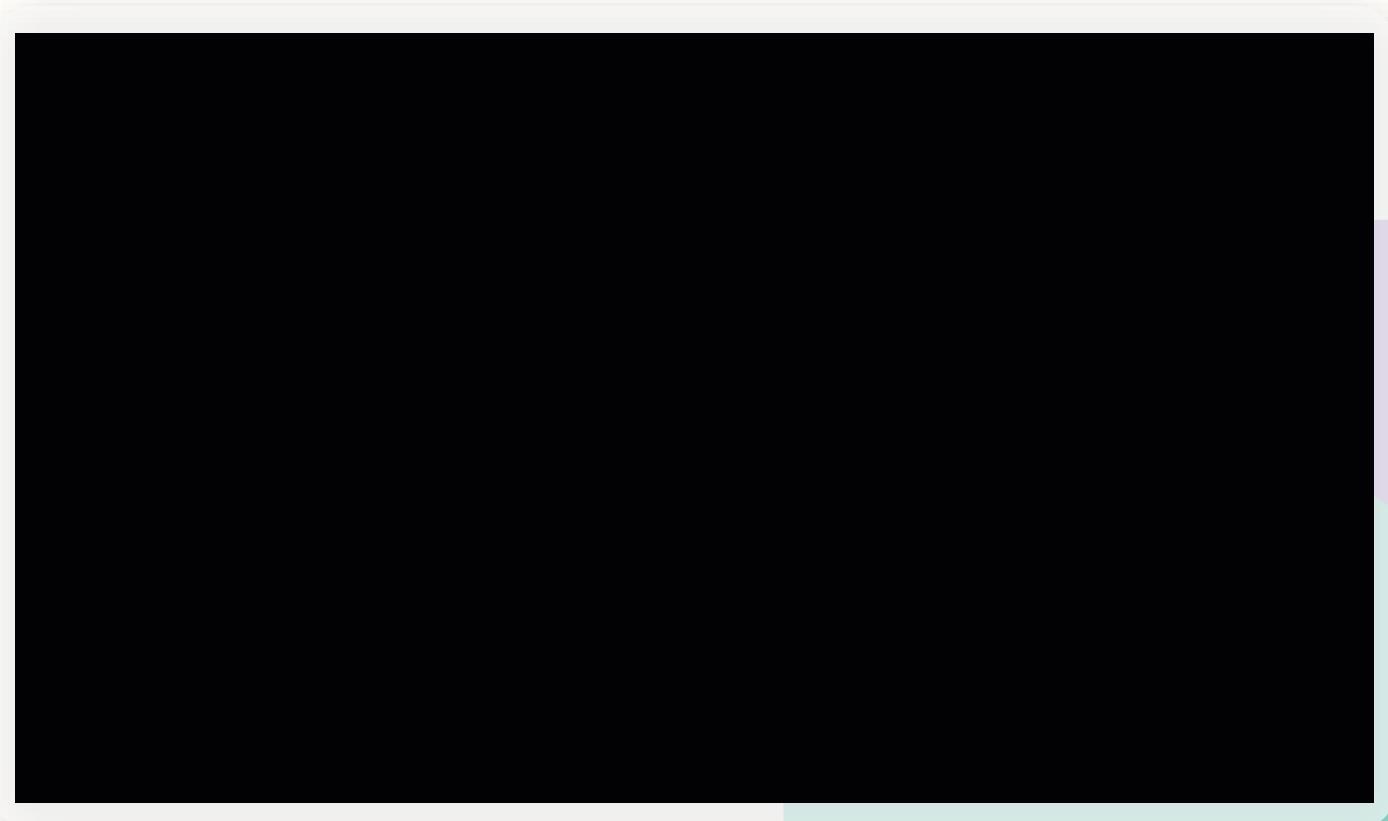
Handles complex data changes (updates, deletes, inserts, etc.)

Writes in open Delta Lake format for analytics and AI-ready data in OneLake

Generates a built-in SQL analytics endpoint for querying and governance

Enables downstream scenarios across BI, AI, Data Science, and Engineering

Provides partner and custom options for ingestion through Mirror Fabric alongside standard mirroring





# Shortcuts

**Shortcuts** in OneLake provide a way to unify data across domains, accounts, and clouds by creating symbolic links to external storage locations instead of copying data.

This makes data instantly available in OneLake's unified namespace without duplication or movement.

Reference existing data without duplication or movement

Consolidate across items or workspaces without changing ownership of the data

Support reuse of shortcuts in Lakehouses and KQL databases

Reduce latency and storage costs by avoiding physical ingestion

Manage ADLS Gen2 accounts and Amazon S3 externally while still accessing them in OneLake

Makes shortcut data fully usable across BI, AI, Data Engineering, and Data Science

The screenshot illustrates the Microsoft Fabric interface, specifically the 'Fabric Lakehouse' section. On the left, the 'Explorer' pane shows a local dataset named 'SampleLakehouse' containing 'Tables' and 'Files'. Within the 'Files' folder, there is a 'Campaign' folder and a 'Transaction' folder. The 'Transaction' folder is highlighted with a blue arrow pointing to it from the text 'Shortcut to ADLS'. To the right of the Explorer, a detailed view of the 'Transaction' folder shows three CSV files: 'Transaction\_1.csv', 'Transaction\_2.csv', and 'Transaction\_3.csv'. Above this, the 'Fabric Lakehouse' title is displayed. On the far right, a separate window titled 'ADLS Gen2 Account' shows the 'sampledata' container. It lists 'Transaction\_1.csv', 'Transaction\_2.csv', and 'Transaction\_3.csv' under the 'Name' column. Arrows from the text 'Local Data' point to the 'SampleLakehouse' node in the Explorer, and from 'Shortcut to ADLS' point to the 'Transaction' folder in the Explorer and the corresponding files in the ADLS account.

# Simplify complex operations with a modern data movement strategy

## Today's data movement

### Replication-heavy

Data copied multiple times before use

### Delayed processing

Insights slowed by nightly or scheduled batch loads

### Fragmented governance

Policies and controls inconsistent across sources

### Complex operations

High overhead moving data with slow time-to-value

## Tomorrow's data movement

### Direct access

Eliminate redundant copies with data-on-demand

### Continuous pipelines

Stream data in real-time for faster decision making

### Unified governance

Consistent policies and security across all data sources

### Automated orchestration

Streamlined management with faster time-to-insight

# Modern data movement starts with zero ETL and zero copy

Access data where it lives – securely, directly, in real-time



## Secure Access

Ensure every connection and pipeline is continuously validated



## Query in place

Enable direct access to data at its source without movement



## Minimize copies

Reduce redundancy to limit sprawl and strengthen compliance



## Agility at scale

Deliver instant connectivity for analytics, AI, and operations

# A zero copy, zero ETL approach hinges on flexible data ingestion

## Mirroring

**Gain near real-time data freshness** by continuously syncing operational databases

**Make data instantly analytics-ready** in OneLake and eliminate complex ETL

## Open mirroring

**Extend flexibility** with APIs and partner integrations to avoid custom pipelines

**Simplify movement and ensure consistency** by standardizing on open Delta format

## Shortcuts

**Access data across clouds** and domains without duplication and instantly make it useable in OneLake

**Reduce storage needs and latency** by eliminating physical copies

# Unifying data in OneLake

## Data Factory



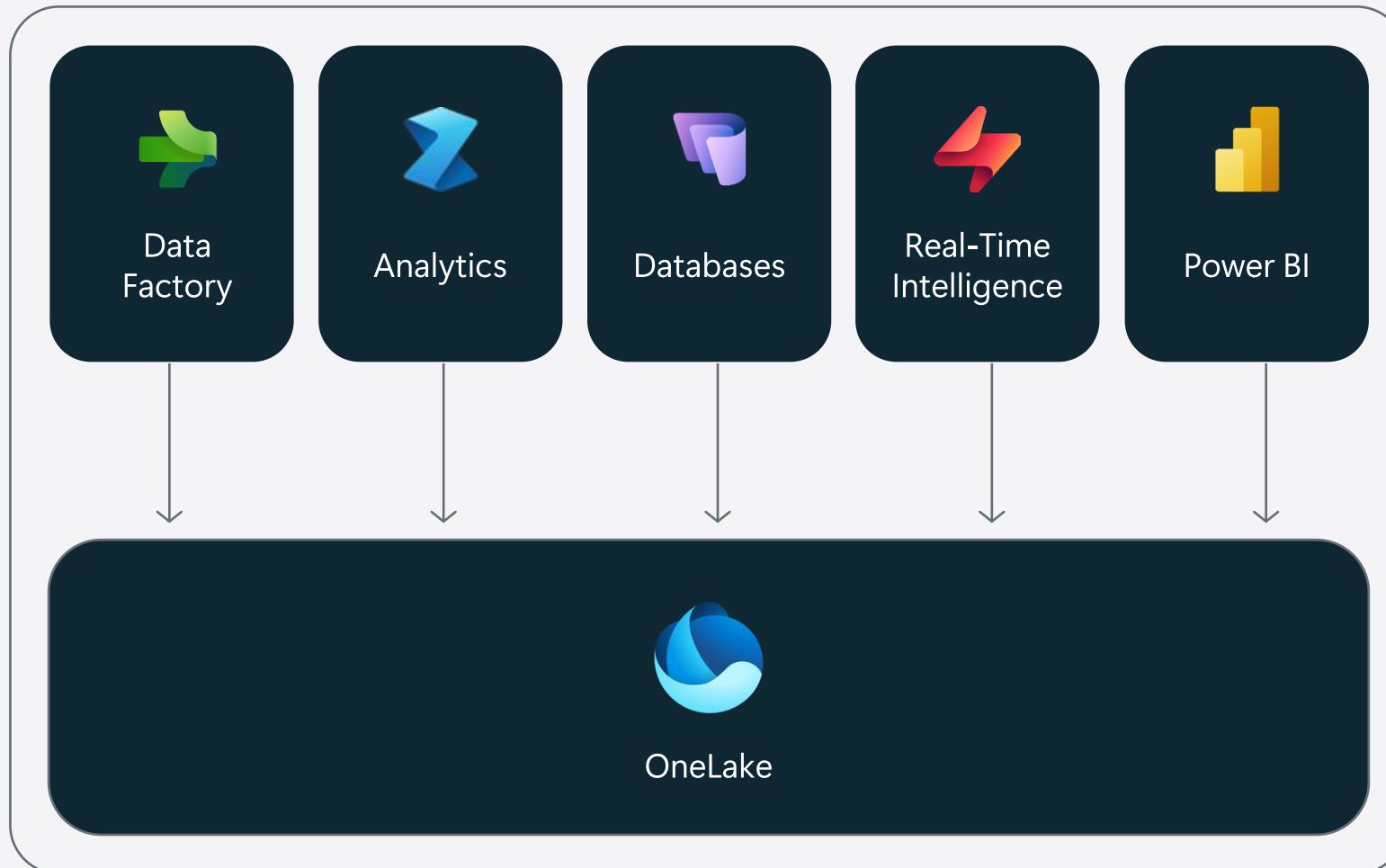
Azure Database for PostgreSQL	Azure Databricks Delta Lake	Amazon RDS for Oracle	Amazon RDS for SQL Server	Amazon Redshift	Phoenix	PostgreSQL	Presto	Magento (Preview)
Azure SQL Database	Azure SQL Database Managed Instance	Apache Impala	Azure SQL Database Managed Instance	DB2	SAP BW	SAP BW via MDX	SAP HANA	Oracle Eloqua (Preview)
Azure Table Storage	MongoDB Atlas	Drill	Google AdWords	Google BigQuery	SAP TABLE	SQL server	Spark	PayPal (Preview)
Azure Cosmos DB (MongoDB API)	Azure Cosmos DB (SQLAPI)	Greenplum	HBase	Hive	Amazon S3	Amazon S3 Compatible	FTP	SAP Cloud For Customer
Azure Data Lake Storage Gen1	Azure Data Lake Storage Gen1 for Cosmos Structured Stream	Informix	MariaDB	Microsoft Access	File system	Google Cloud Storage (S3APD)	HDFS	Salesforce Marketing Cloud
Azure Data Lake Storage Gen1 for Cosmos Structured Stream	Azure Database for MariaDB	MySQL	Netezza	Oracle	HTTP	Oracle Cloud Storage (S3AP)	SFTP	Shopify (Preview)
teradata	VERTICA	ODBC	OData	REST	Amazon Marketplace Web Service	Concur (Preview)	Dataverse (Common Data Service for App)	Web Table
Jira	Kusto	SharePoint Online List	Dynamics 365	Dynamics AX	Dynamics CRM	Cassandra	Couchbase (Preview)	MongoDB

# OneLake, Lakehouse, and Spark



# OneLake is the OneDrive for data

Your **single, secure foundation** for every workload



**One open format** for storing and querying data org-wide

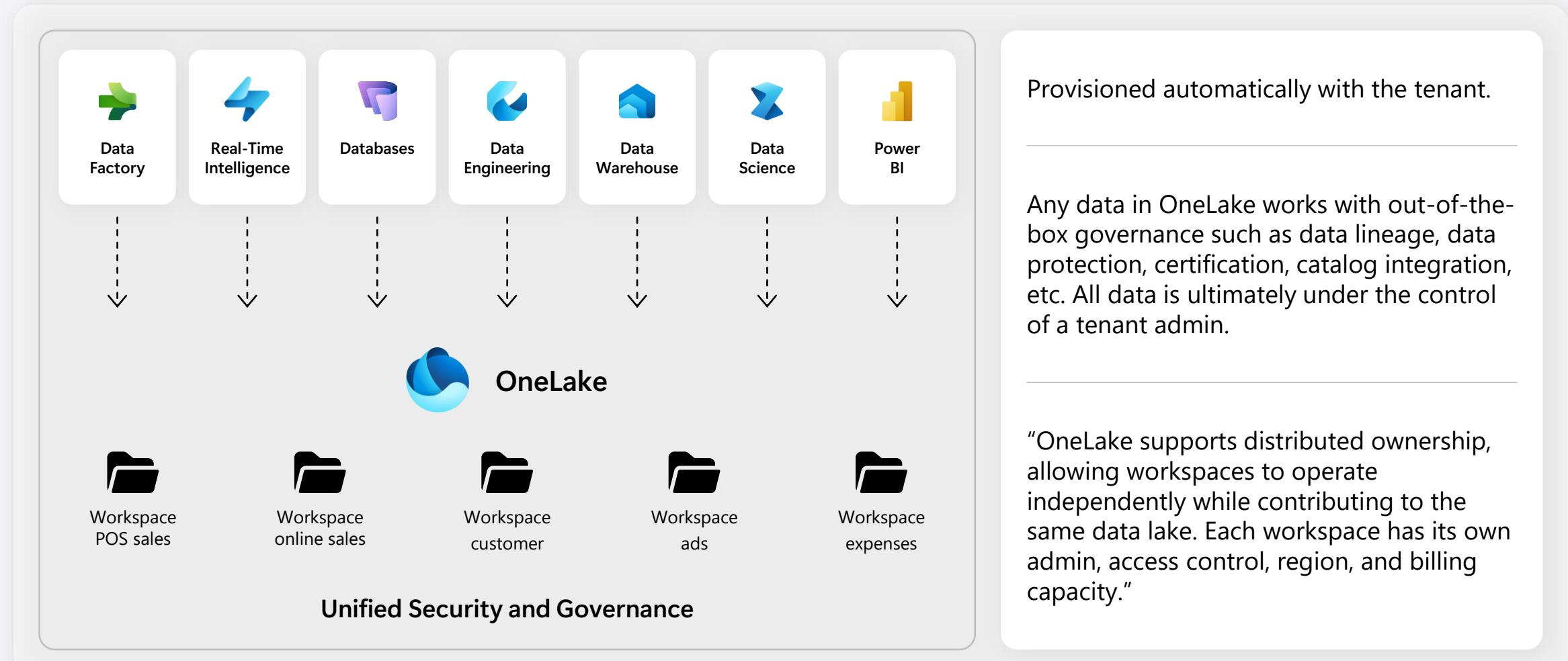
**Automatically indexed** to simplify discovery and sharing

Built-in lineage for **end-to-end traceability** across workloads

Protected at every layer with OneLake security

# A single unified SaaS data lake

The OneDrive for data with no silos



# OneLake which logically spans the world

Reside workspaces in different regions around the world while still in the same data lake.

Reside data in different regions without managing different storage resources.

Meet demand for any scale with dedicated storage resources for each workspace (capacity, throughput, and IOPS).

Underlying physical storage is virtualized away.

All storage is zone redundant by default with an option for Geo redundancy.

## Unified Security and Governance



OneLake



Workspace  
POS sales



Workspace  
online sales



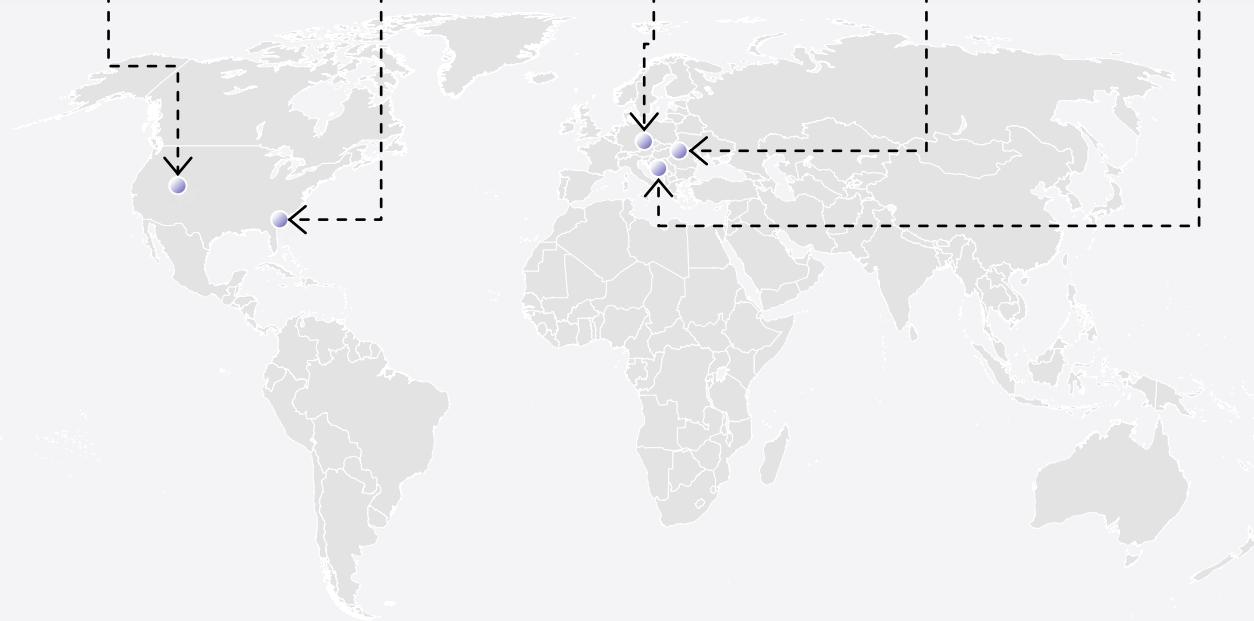
Workspace  
customer



Workspace  
ads

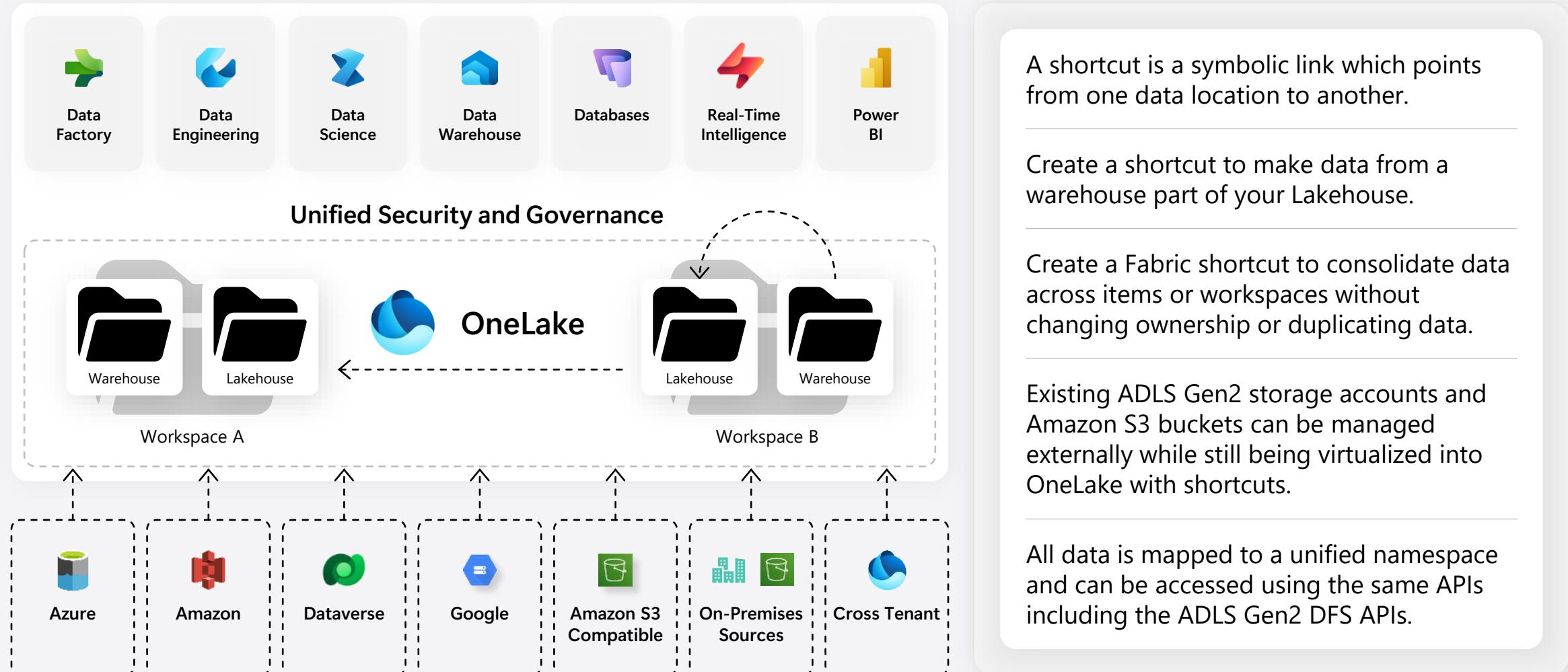


Workspace  
expenses



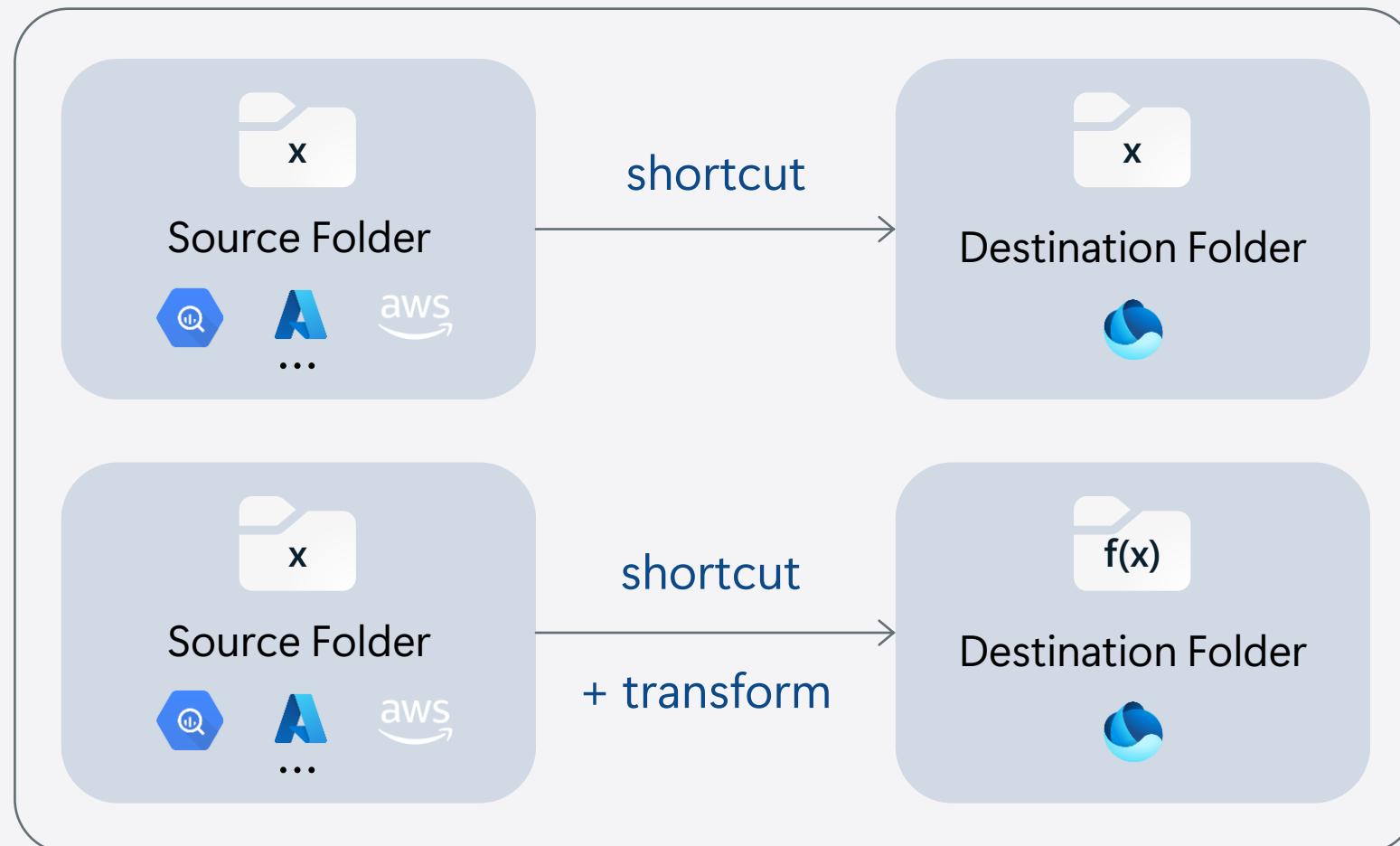
# Shortcuts virtualize data across domains and clouds

No data movements or duplication



# Shortcut transformations

Seamless enrichment with AI in OneLake



AI transformations built directly into OneLake

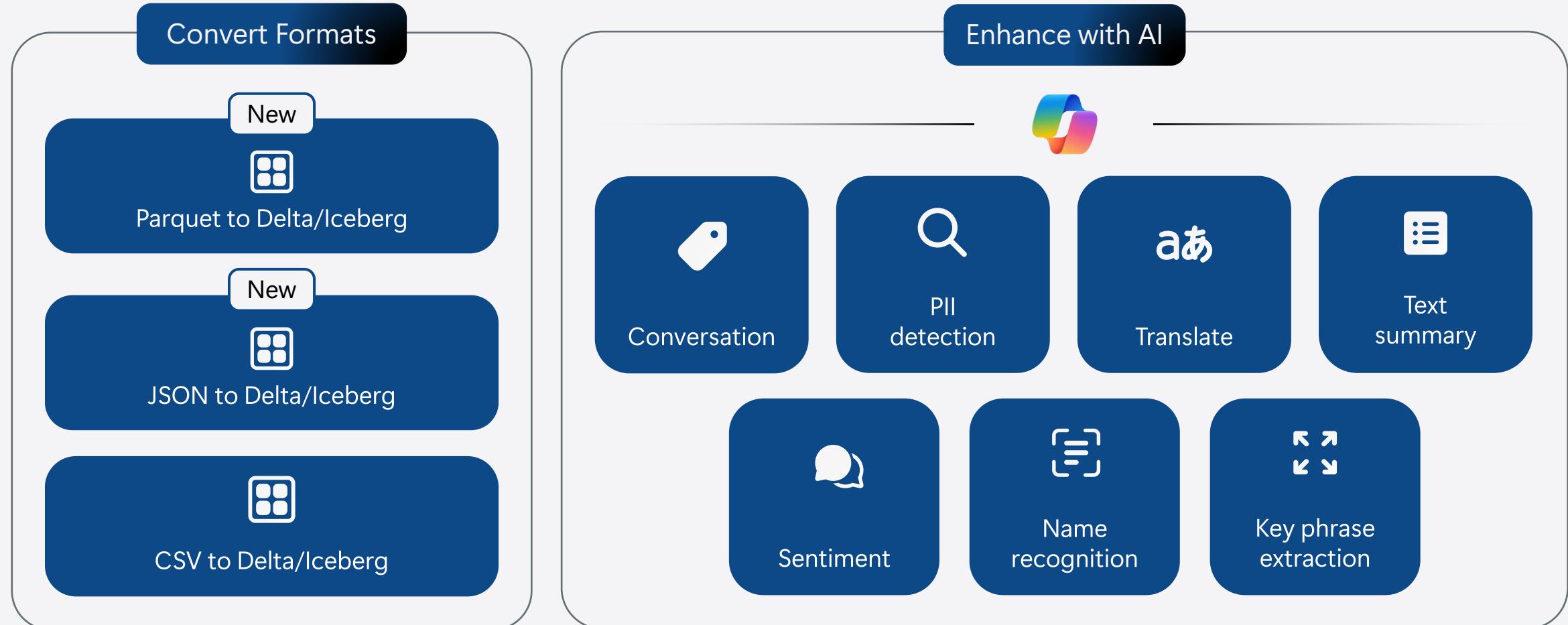
Extend easily by pointing to source folders and adding defined transformations

Right-click operations enabled on all folders across all clouds

Auto track changes in source and sync the destination

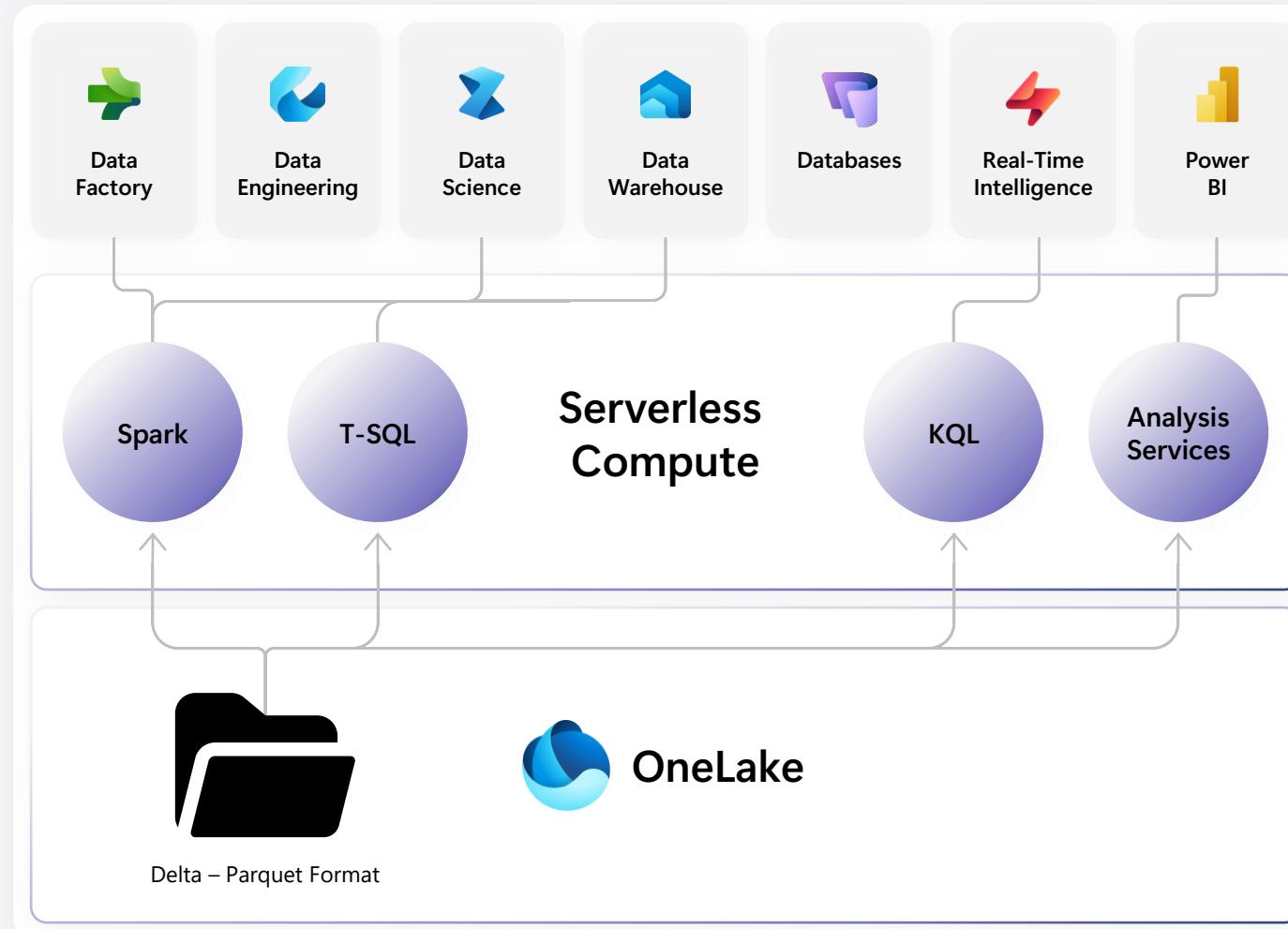
# Shortcut transformations

Seamless enrichment with AI in OneLake



# One Copy for all computers

One copy of data can be read by all engines



Once data is stored in the lake, it is directly accessible by all the engines without needing any import/export.

You are able to choose the right engine for the right job.

All the compute engines have been fully optimized to work with Delta Parquet as their native format.

Shared universal security model is enforced across all the engines (coming soon).



OneLake



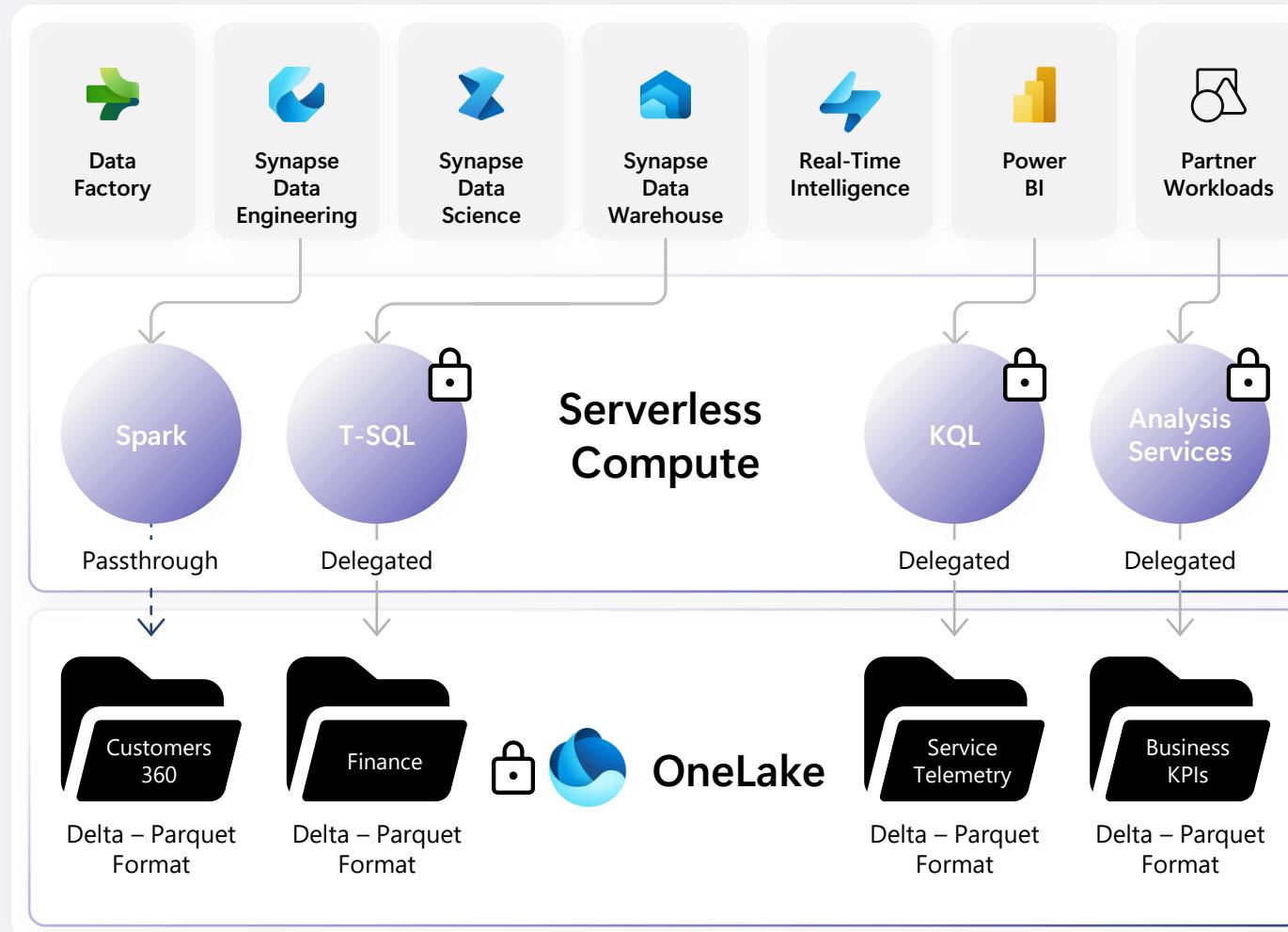
Delta Lake  
Format



Iceberg  
Format

Transparent simultaneous support for Delta Lake and Iceberg formats

# One Copy, many security definitions

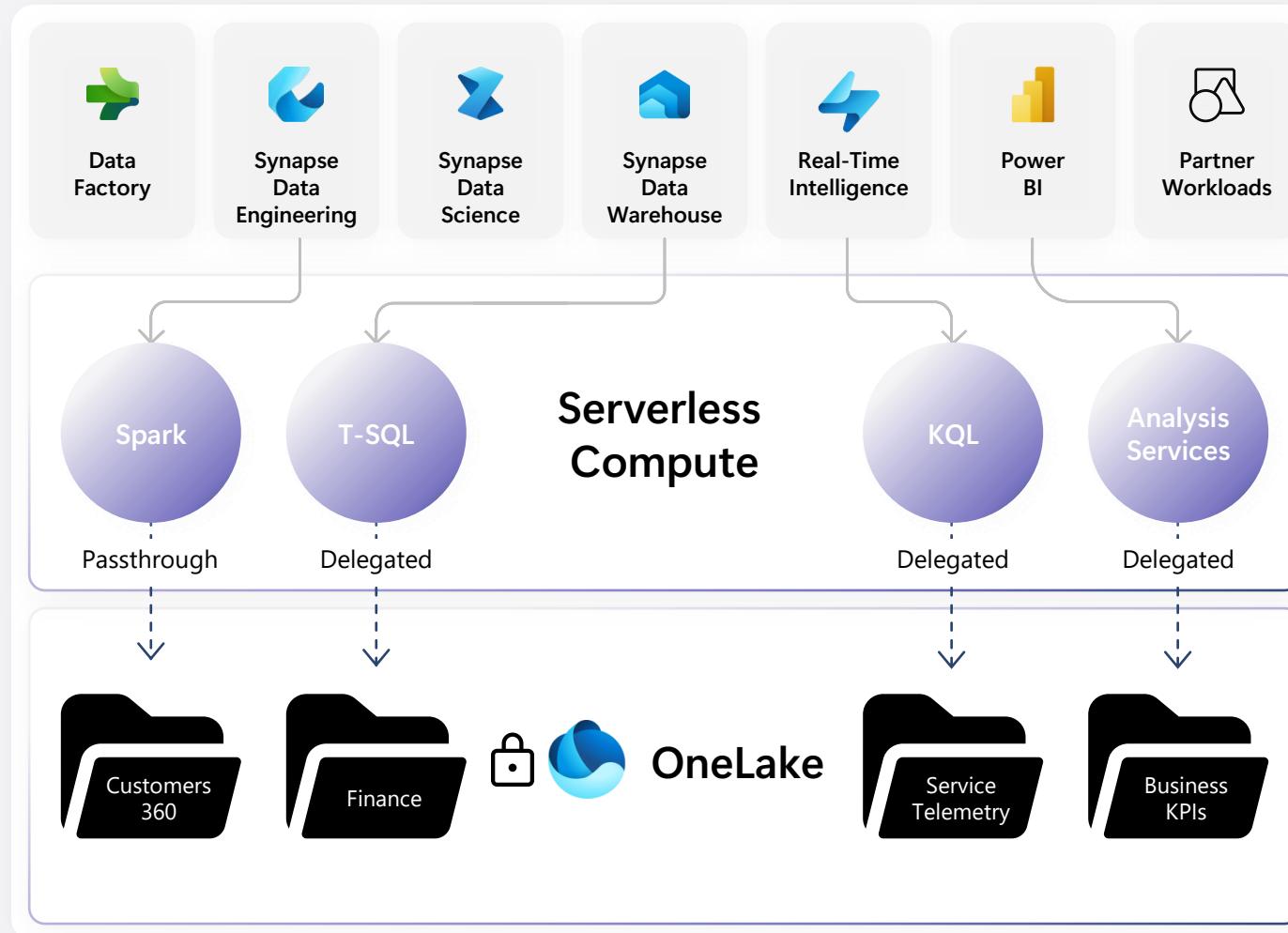


Most engines in Fabric use delegated mode to access the data.

Spark does not have native security, so it relies on passthrough mode to secure data.

In this state, customers have maximum flexibility of security. However, synchronizing security requires manual effort.

# One Copy, One Security Definition



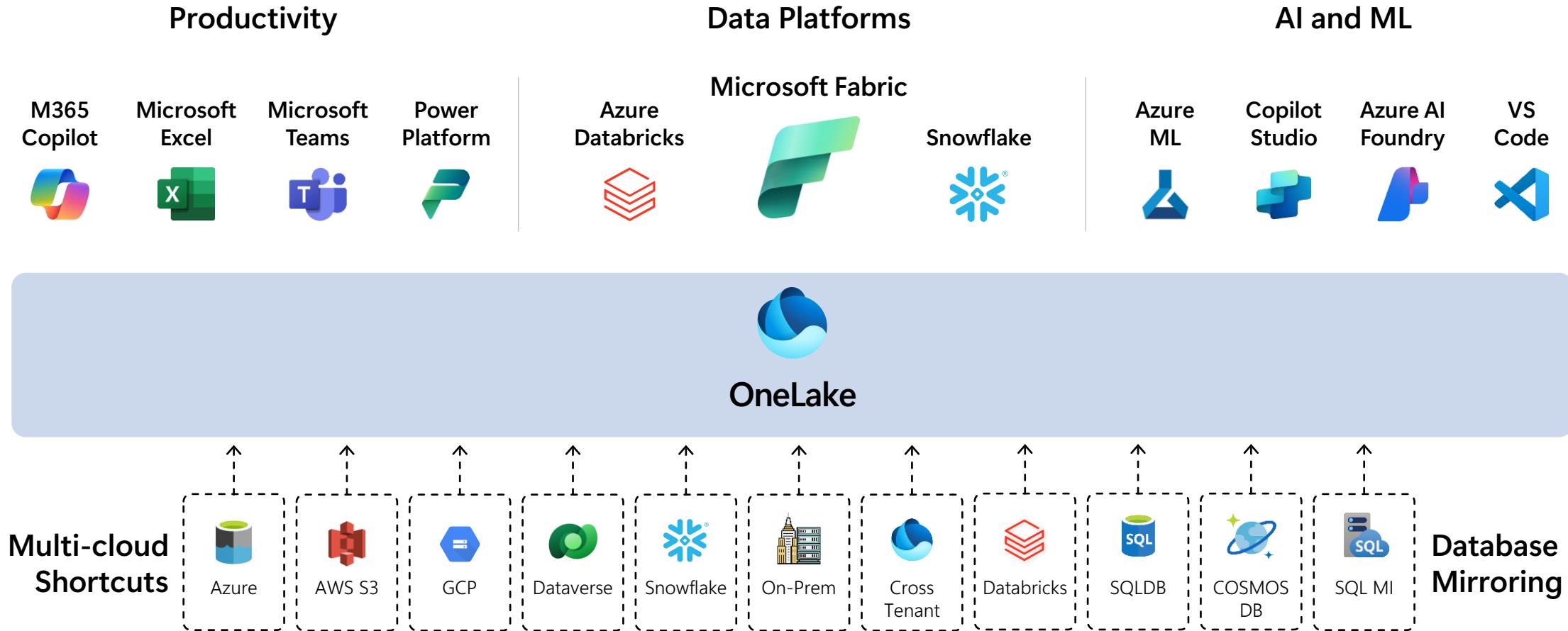
OneLake now has a complete set of fine-grained security features:

- Table/folder security
- Column level security
- Row level security
- + more coming soon...

Engines can then switch to using passthrough mode as the primary security model.

Security is now centrally defined and enforced consistently across all engines.

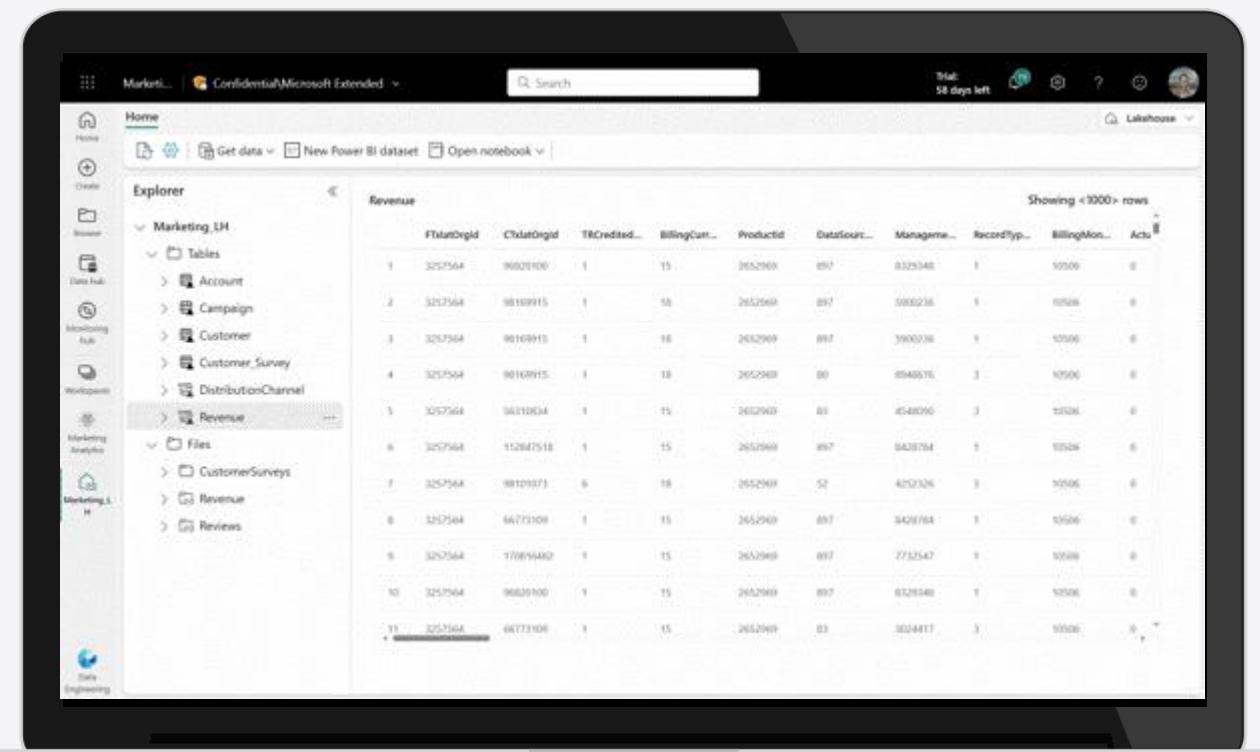
# OneLake data is available everywhere



# Lakehouse overview

Store, manage and analyze all your data in a single location and easily share across the entire enterprise

- Flexible and scalable solution that enables organizations to handle large data volumes of all types and sizes
- Built-in SQL endpoint unlocks data warehouse capabilities on top of your lakehouse with no data movement
- Use 'direct lake' mode to build reports in seconds directly on top of the data lake with blazing fast performance



The screenshot shows a Microsoft Power BI interface. On the left, there's a navigation pane with icons for Home, Create, Data Hub, Monitoring hub, Workbooks, Marketing Analytics, and Marketing Lakehouse. The main area has a title bar "Marketing\_LH | Confidential/Microsoft Extended" and a search bar "Search". Below the title bar, there are buttons for "Get data", "New Power BI dataset", and "Open notebook". The central part of the screen displays a table titled "Revenue" with 11 rows of data. The columns are: FTRanOrgId, ChDataOrgId, TICreated..., BillingCatt..., ProductId, DataSource..., Manage..., RecordTyp..., BillingMon..., and Actu... . The data includes various numerical values and some text entries like "Revenue" and "Reviews". The table has a header row and 10 data rows.

	FTRanOrgId	ChDataOrgId	TICreated...	BillingCatt...	ProductId	DataSource...	Manage...	RecordTyp...	BillingMon...	Actu...
1	3257564	9602100	1	15	2652969	897	8325346	1	10500	0
2	3257564	98169915	1	18	2652969	897	5880238	1	10500	0
3	3257564	98169915	1	18	2652969	897	3900736	1	10500	0
4	3257564	98169915	1	18	2652969	897	8548876	3	10500	0
5	3257564	98310034	1	15	2652969	897	4548056	3	10500	0
6	3257564	112847518	1	15	2652969	897	9428784	1	10500	0
7	3257564	98101977	6	18	2652969	52	4252326	3	10500	0
8	3257564	66773108	1	15	2652969	897	8428784	1	10500	0
9	3257564	17889482	1	15	2652969	897	2732547	1	10500	0
10	3257564	9602100	1	15	2652969	897	8325346	1	10500	0
11	3257564	68773108	1	15	2652969	897	3024417	3	10500	0

# Lakehouse overview

## Organize data, structured and unstructured

- The **Files** section is an unmanaged area containing any file type and organized by folders like the Windows file system.
- The **Tables** section is a managed area containing structured data in CSV, Parquet, or Delta format generally organized by schema.
- Only Delta tables are discovered automatically and made available in the **SQL analytics endpoint**.

The screenshot shows the Microsoft Data Lakehouse interface. On the left, the Explorer sidebar displays the Marketing\_LH schema, which includes Tables (Account, Campaign, Customer, Customer\_Survey, DistributionChannel, Revenue) and Files (CustomerSurveys, Revenue, Reviews). The Revenue table is selected and previewed on the right. The preview table has 11 rows and the following columns:

	FTdataOrgId	CTdataOrgId	TRCredited...	BillingCatt...	Productid	DataSource...	Manageme...	RecordTyp...	BillingMon...	Actu...
1	3257564	96020100	1	15	2652969	897	8325340	1	10500	0
2	3257564	98169915	1	18	2652969	897	5886236	1	10500	0
3	3257564	98169915	1	18	2652969	897	3906936	1	10500	0
4	3257564	98169915	1	18	2652969	897	8548676	3	10500	0
5	3257564	98310034	1	15	2652969	897	4548050	3	10500	0
6	3257564	112847518	1	15	2652969	897	9428784	1	10500	0
7	3257564	98169977	6	18	2652969	52	4292326	3	10500	0
8	3257564	66773108	1	15	2652969	897	8428784	1	10500	0
9	3257564	17889482	1	15	2652969	897	2732547	1	10500	0
10	3257564	96020100	1	15	2652969	897	8325340	1	10500	0
11	3257564	68773108	1	15	2652969	897	3024417	3	10500	0

# Fabric runtime

## Leverage Runtime 1.3 in Fabric for running all your Spark jobs

- › Runtime 1.3 brings major updates such as upgrading Spark to 3.5.0, Delta to 3.2 and Python to 3.11
- › Delta Lake enables 'OneCopy' allowing all Fabric compute engines to seamlessly share data
- › All Fabric artifacts automatically write Delta with V-Order optimization providing lightning-fast analytics on Vertiscan for reporting with Power BI
- › 'Load to Delta' in the lakehouse unlocks a visual experience to load common file formats and folders to Delta

## Spark compute

Configure and manage settings for Spark workloads in the capacity.

[Learn more about Spark compute](#) 

### Pool

#### Default pool for workspace

Select default or create custom pools which become the default pool option for workspaces and artifacts in the capacity.

 A starter pool with 10 nodes is provided for evaluation purposes. In the coming months, the starter pool will automatically be resized based on your purchased capacity size. [Learn more](#)

Starter pool 

#### Pool details

Node family  
Memory optimized

Node size  
Medium

Number of nodes  
1 - 10

### Runtime

#### Runtime Version

Runtime family defines which version of Spark your Spark pool will use.

[Learn more about Runtime Version](#) 

1.1 (Spark 3.3, Delta 2.2) 

### High concurrency

#### High concurrency

High Concurrency mode enables multiple notebooks to use the same Spark application to save session start time.

[Learn more about High concurrency](#) 

# Starter pools to get started quicker

## Get started with running Spark in a matter of seconds

- › Starter pools come pre-wired to Fabric workspaces meaning users can get started running Spark with no cluster setup necessary
- › Starter pools are kept 'live', meaning they provide Spark sessions within 10-15 seconds
- › High concurrency mode enables users to share Spark sessions across multiple notebooks
- › Attaching a notebook to an existing session results in lightning-fast session start up speed <5 seconds
- › High concurrency mode support will be available for notebooks in pipelines

Contoso's Data Ingestion Notebook Activity

```
1 pip install ydata_profiling
[1] ① - Apache Spark session started in 4 min 23 sec 281 ms. Running
2 Log
3 Output is hidden
4
5 Import numpy as np
6 Import pandas as pd
7 From ydata_profiling import ProfileReport
8
9 Import numpy as np
```

Session stopped High concurrency session

# Custom pools for more control

## Create custom Spark pools based on workload requirements

- › Workspace admins can create a new default pool for the workspace (if granted permissions by the capacity admin)
- › Option to choose node size, the number of nodes & executors as well as auto-scale and dynamic allocation of executors
- › Single node pools can be set up to run small jobs or test workloads while optimizing compute spend

**Create pool**

**Spark pool name \***  
 ! Name should not be empty.

**Node family**

**Node size**

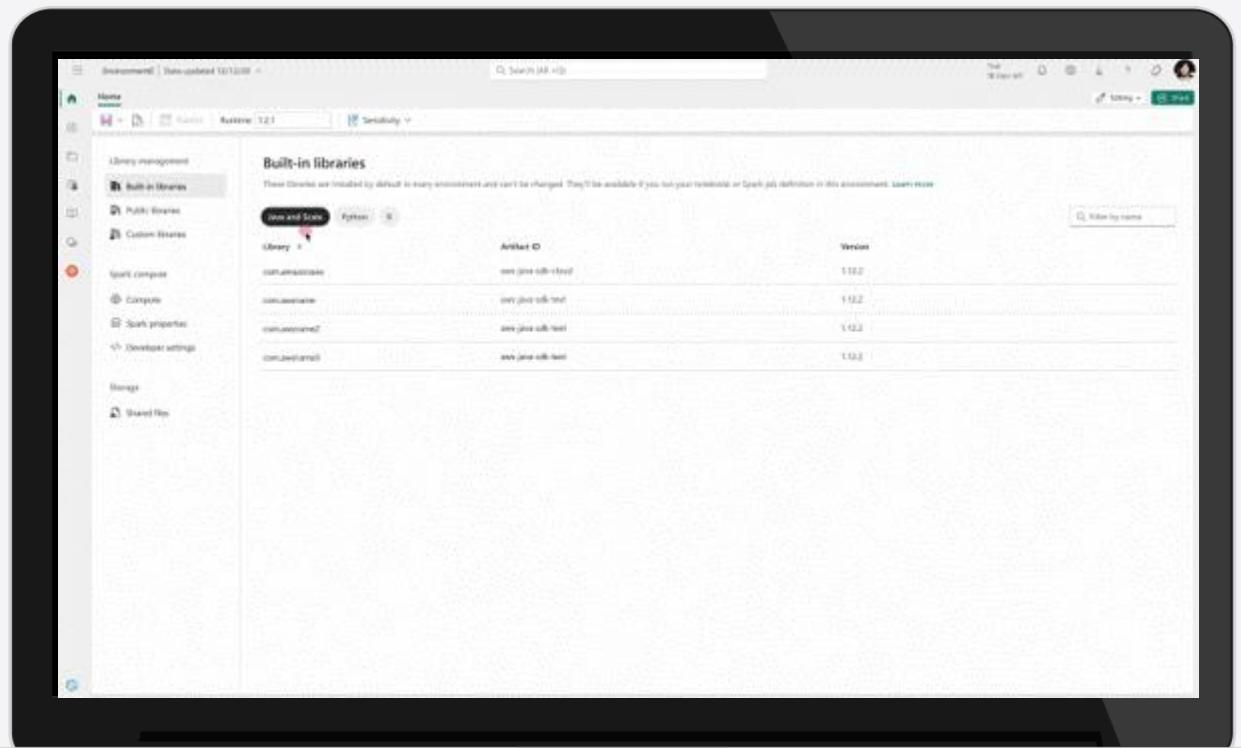
**Autoscale**  
If enabled, your Apache Spark pool will automatically scale up and down based on the amount of activity.  
 Enable autoscale  
A horizontal slider with two input fields at the ends labeled '1' and '10'. There are two green circular markers on a black track, positioned between the first and second tick marks, indicating a value of approximately 2.

**Dynamically allocate executors**  
 Enable allocate  
A horizontal slider with two input fields at the ends labeled '1' and '9'. There is one green circular marker on a black track, positioned between the first and second tick marks, indicating a value of approximately 2.

# Spark Environments

**Use environments for more flexibility  
in developing your project**

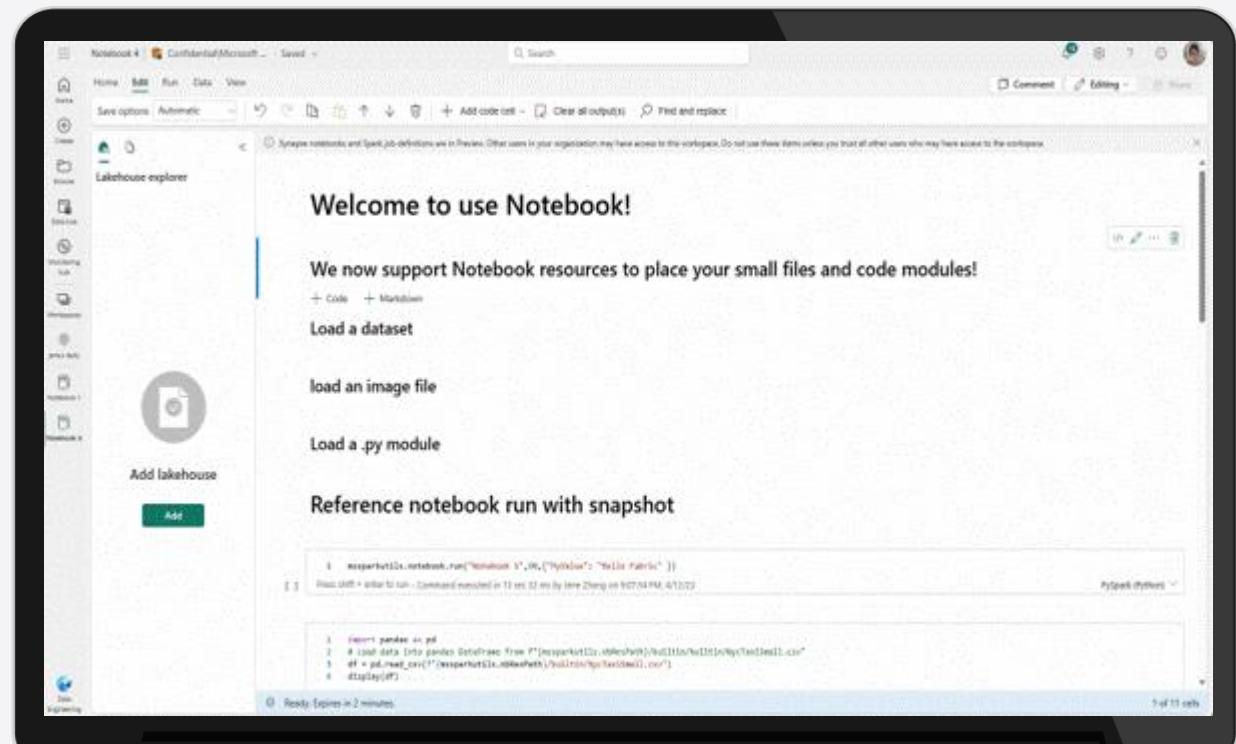
- The environment item allows users to configure all their Spark and library related settings in a single place
- Users can do the following:
  - Install libraries
  - Choose from a pre-defined user pool
  - Configure a new user defined pool
  - Set Spark properties
  - Leverage the environment file system
- Environments can be attached to notebooks and Spark jobs, overriding the default workspace pool



# Notebook overview

# Immersive authoring experience for data developers

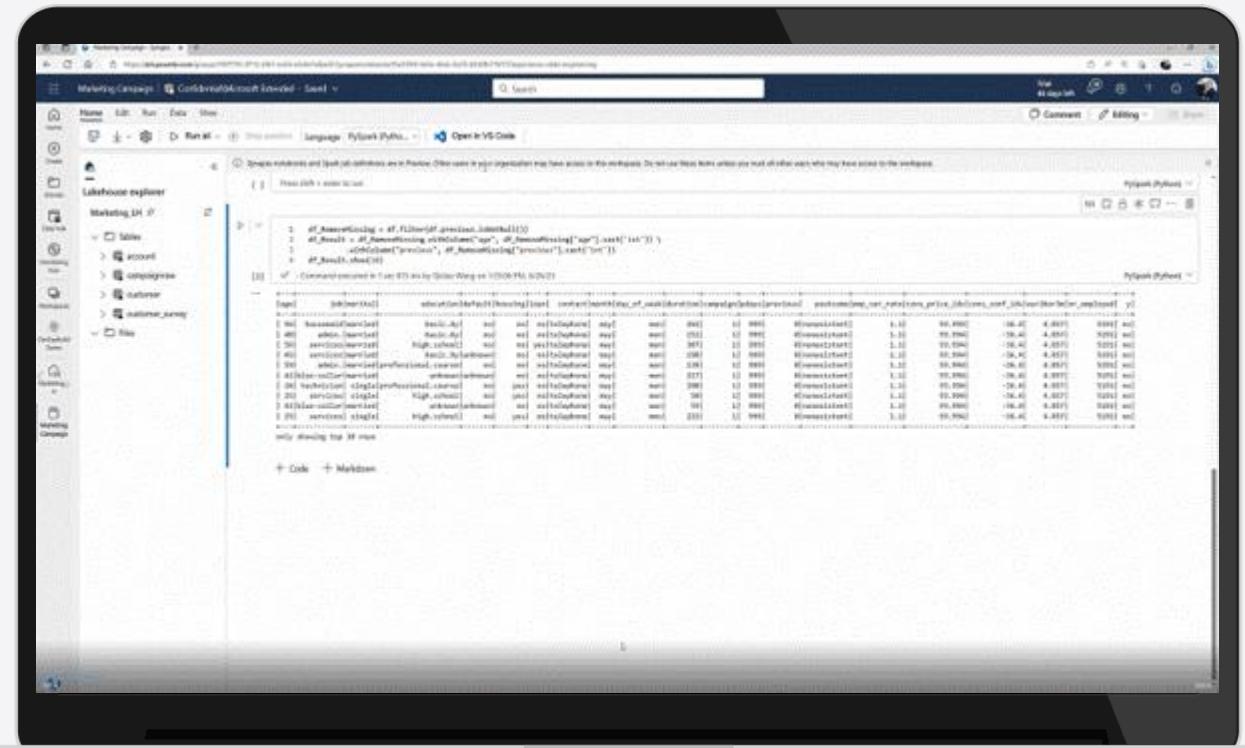
- Rich notebook capabilities including native lakehouse integration, real-time collaboration with commenting support, auto-save support, lightweight scheduling and pipeline integration
  - Manage your Python and R libraries through in-line installs using commands like %pip install
  - Advanced notebook development support with ability to reference notebooks in notebooks, and snapshots for easy troubleshooting
  - Switch between PySpark, Scala, R, and SQL within the same notebook.



# VS Code integration

Empower pro developers to use their tools of choice for their development

- Launch the Fabric VS Code extension, navigate through your workspace and work with Notebooks, Spark Jobs and Lakehouses directly in the IDE
- Run and debug notebook cells or standard batch jobs on the remote Spark cluster
- Have the flexibility to work offline with a local environment and push your changes online when needed
- Fully remote way of working with vscode.dev
- Better dependency management across local and remote environments



# Transform and enrich data with low-code AI functions

## Accessible to every role

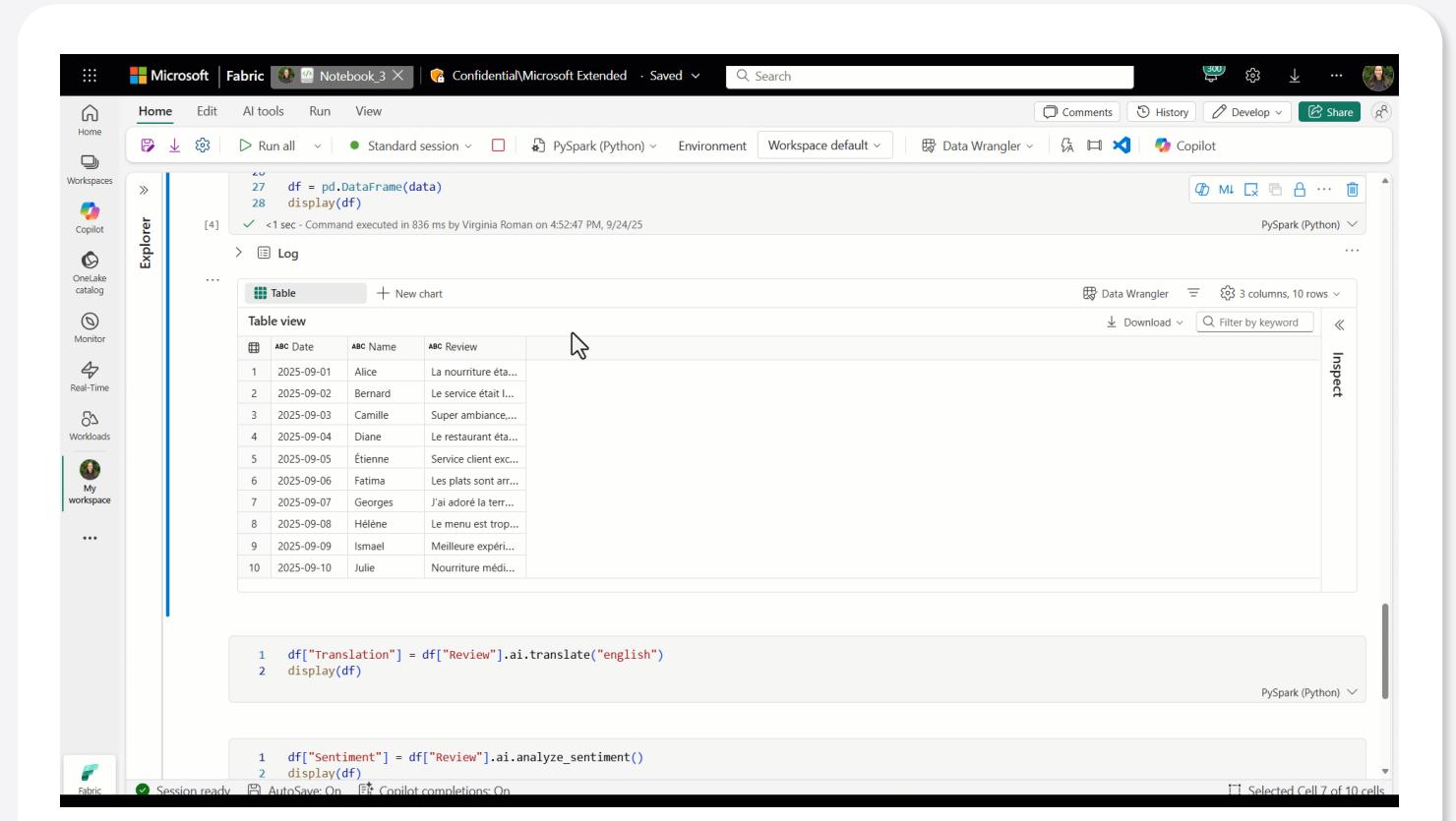
From developers to analysts, all data professionals can use AI functions to enrich enterprise data with generative AI

## Harness industry-leading LLMS

Use built-in AI functions for summarization, sentiment, translation, classification, grammar correction, and more to derive instant value from data

## Seamless in workflows

AI functions integrate directly into data science and data engineering workflows in Fabric — no infrastructure setup or expertise required



The screenshot shows a Microsoft Fabric Notebook interface. On the left, there's a sidebar with icons for Home, Workspaces, Copilot, OneLake catalog, Monitor, Real-Time, Workloads, and My workspace. The main area has tabs for Home, Edit, AI tools, Run, and View. The AI tools tab is active, showing a code editor with PySpark (Python) code:

```
1 df["Translation"] = df["Review"].ai.translate("english")
2 display(df)
```

Below the code editor is a Data Wrangler table view showing 10 rows of data:

	ABC Date	ABC Name	ABC Review
1	2025-09-01	Alice	La nourriture éta...
2	2025-09-02	Bernard	Le service était i...
3	2025-09-03	Camille	Super ambiance...
4	2025-09-04	Diane	Le restaurant éta...
5	2025-09-05	Étienne	Service client exc...
6	2025-09-06	Fatima	Les plats sont arr...
7	2025-09-07	Georges	J'ai adoré la terr...
8	2025-09-08	Helène	Le menu est trop...
9	2025-09-09	Ismael	Meilleure expéri...
10	2025-09-10	Julie	Nourriture médi...

At the bottom of the notebook, there are status indicators: Session ready, AutoSave: On, Copilot completions: On, and Selected Cell 7 of 10 cells.

# Out-of-the-box AI functions in Fabric

Transform and enrich data with generative AI for summarization, classification, text generation and more – no setup or coding required

Use AI functions directly in Fabric Data Science and Data Engineering workflows with pandas or Spark

Run tasks with a single line of code, without infrastructure or advanced expertise

Apply out-of-the-box functions for:

- Categorizing text
- Detecting sentiment
- Answering a custom prompt
- Extracting entities
- Summarizing text
- Translating text
- Calculating similarity
- Fixing grammar

The screenshot shows two examples of AI functions being used in a Jupyter Notebook environment.

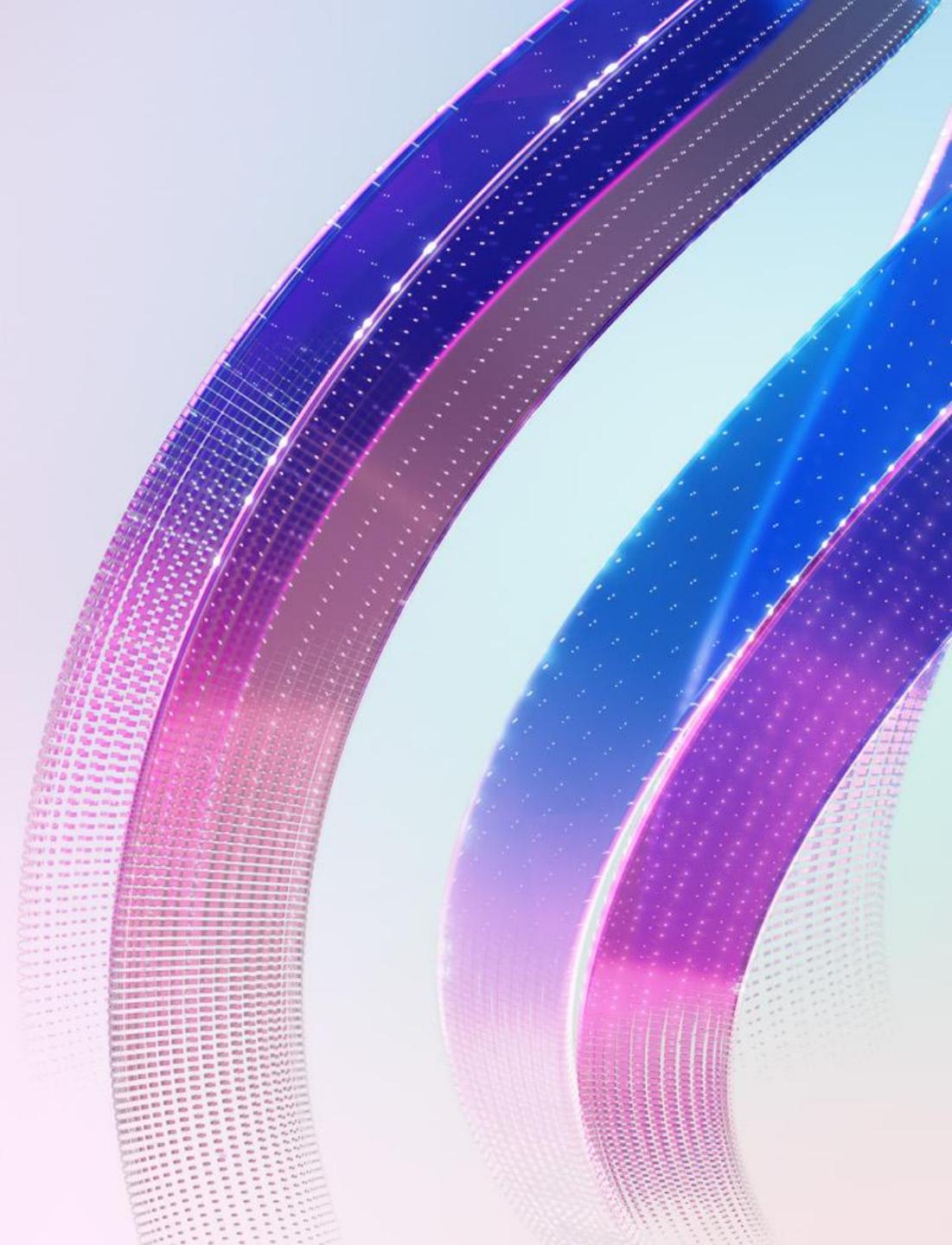
**Example 1:** The code `df["Category"] = df["Description"].ai.classify("Luxury", "Necessity", "Discretionary")` uses the `ai.classify` function to categorize transaction descriptions into Luxury, Necessity, or Discretionary. The resulting DataFrame is displayed as a table.

ABC DateTime	ABC Description	ABC Amount	ABC Category
1 8/1/2025 22:03	Safeway #1032 SEATTLE	266.47	Necessity
2 8/2/2025 10:38	Swedish Medical Center	265.78	Necessity
3 8/5/2025 15:14	Serenity Spa	300.98	Luxury
4 8/6/2025 10:10	Netflix.com	20.79	Discretionary

**Example 2:** The code `df["Alert"] = df.ai.generate\_response(prompt="")` uses the `ai.generate\_response` function to detect unusual transactions. The resulting DataFrame includes an 'Alert' column where transactions like Shell Gas #39485 are flagged as suspicious.

ABC DateTime	ABC Description	12 Amount	ABC Alert
1 8/6/2025 19:08	Alaska Airlines	901.38	N/A
2 8/10/2025 3:53	Allstate Home I...	83.24	N/A
3 8/13/2025 9:39	Spotify	15.26	N/A
4 8/14/2025 3:06	Shell Gas #39485	2050.75	Suspicious activity: A gas station transaction of \$2050.75 is unusually high ..
5 8/15/2025 18:26	AMC Pacific Place	30.45	N/A
6 8/15/2025 22:35	Ballard Apartme...	2100.0	N/A

# Lab 1



**10-minute break**

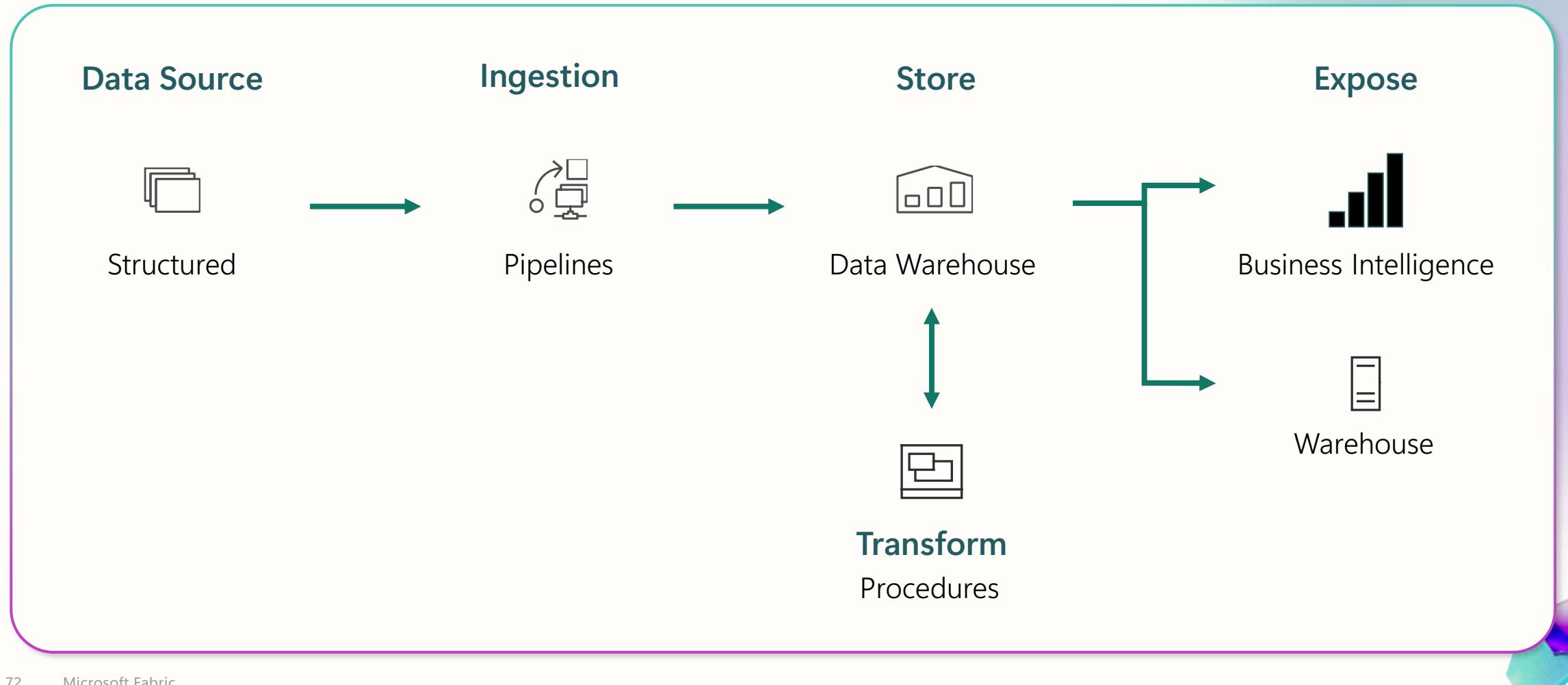
# Data Warehouse





# Data warehouse scenario

End-to-end analytics scenario





# Data Warehouse | Overview

Enterprise scale data warehouse with open standard format

No knobs performance with minimal set-up and deployment, no configuration of compute or storage needed

## **Key Capabilities:**

- Lake-centric warehouse stores data in OneLake in open Delta format with easy data recovery and management
- Use Fabric Mirroring for Zero-ETL integration of data from Azure SQL, Snowflake, or Azure Cosmos DB
- Data loading and transforms at scale, with full multi-table transactional guarantees provided by the SQL engine
- Virtual warehouses with cross-database querying and a fully integrated semantic layer
- Flexibility to build data warehouse or data mesh based on organizational needs and choice of no-code, low-code, or T-SQL for transformations

The screenshot shows the Microsoft Fabric Data Explorer interface. The left sidebar displays a tree view of databases, tables, and schemas. The main area shows a T-SQL query being run:

```
CREATE TABLE [TravelWarehouse].[dbo].[InFlightMeals]
(
    MealId int NOT NULL,
    MealCategory varchar(50) NOT NULL,
    MealName varchar(100) NOT NULL,
    HostName varchar(75) NOT NULL,
    IsVegan varchar(5) NULL,
    IsVegetarian varchar(5) NULL,
    IsDietaryFree varchar(5) NULL
)
```

The query was successful, taking 743 ms. The interface also includes tabs for Data, Query, and Model, and a status bar indicating 0 columns and 0 rows.

# Table Design

- No need to define distribution key
- Define datatype precision accurately and avoid “lazy” datatype e.g. varchar(8000)
- Specify NOT NULL constraint where applicable
- Default collation is CS (Latin1\_General\_100\_BIN2\_UTF8)
- Use [REST API](#) to create CI collation  
**(Latin1\_General\_100\_CI\_AS\_KS\_WS\_SC\_U TF8)**

```
CREATE TABLE orders
(
    o_orderkey          bigint NOT NULL,
    o_custkey            bigint NOT NULL,
    o_orderstatus        char(1) NOT NULL,
    o_totalprice         decimal(15,2) NOT NULL,
    o_orderdate          date NOT NULL,
    o_orderpriority      char(15) NOT NULL,
    o_clerk              char(15) NOT NULL,
    o_shippriority       integer NOT NULL,
    o_comment             varchar(79) NOT NULL
);
```

# Table Design

- Define constraints
  - Only needed for PBI semantic modeling
  - Constraints are NOT ENFORCED
  - Primary key – Supports NONCLUSTERED
  - FOREIGN KEY
  - UNIQUE - Supports NONCLUSTERED
- Not supported
  - Partitioning
  - CCI
  - Clustered index, non-clustered index
  - Default constraints

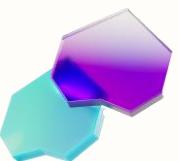
```
CREATE TABLE orders
(
    o_orderkey      bigint NOT NULL,
    o_custkey        bigint NOT NULL,
    o_orderstatus    char(1) NOT NULL,
    o_totalprice     decimal(15,2) NOT NULL,
    o_orderdate      date NOT NULL,
    o_orderpriority  char(15) NOT NULL,
    o_clerk          char(15) NOT NULL,
    o_shippriority   integer NOT NULL,
    o_comment         varchar(79) NOT NULL
);
```

# Microsoft Fabric

## File Size

### File Size Impact on Load Performance

- Minimum 4 MB, avoid thousands of KB files
- Avoid single large file (i.e. 1 TB)



# Microsoft Fabric

## Number of Files factor

### Number of files impact on load performance

- Keep the number of files to be at least 1,000
- Best throughput using 1,000 files is 7x faster than single large file
- Total file size 194GB, each file is 194MB, 1,008 uncompressed CSV files
- Using SKU F64



# Files	Duration	MB/sec	GB/hr
1	0:12:00	277	975
24	0:03:04	946	3,324
100	0:05:27	710	2,495
500	0:04:59	776	2,729
1,008	0:01:42	1,944	6,835
10,008	0:01:44	1,906	6,703



# Microsoft Fabric

## Parallel Loads Factor

### Number of parallel loads (threads) impact on load performance

- Avoid using a single thread
- For F64, 3 parallel loads into the same table or different tables achieve the max throughput
- With parallel loads greater than 3, other loads have to wait for the resource available
- If more parallel loads are needed to achieve higher throughput, SCALE to higher capacity
- Loading into the same table or different table achieved the same throughput.



# Threads	Duration	MB/sec	GB/hr
1	0:01:42	1,944	6,835
2	0:01:39	3,993	14,036
3	0:01:40	5,929	20,844
4	0:03:09	3,682	12,945
5	0:03:04	4,728	16,621
10	0:03:43	5,072	17,832



# Microsoft Fabric

## Scaling Factor

### Scaling to F128 to get more resources for higher concurrency

- Avoid using a single thread
- Load throughput stays the same for <=3 threads, but linearly scaled 2x for 4, 5, 6 threads
- For F128, 6 parallel loads into the same table or different tables achieve the max throughput
- If more parallel loads are needed to achieve higher throughput, SCALE to higher capacity



# Threads	Duration	MB/sec	GB/hr
1	0:01:43	1,925	6,768
2	0:01:40	3,966	13,943
3	0:01:43	5,776	20,305
4	0:01:45	7,554	26,558
5	0:01:43	9,626	33,842
6	0:01:48	11,019	38,730





# Microsoft Fabric

## Create table as select (CTAS)

### CTAS

- CTAS creates a new table based on the output of a SELECT statement.
- Cross-workspace, cross database, DW/LH/SC queries are supported.
- The simplest and fastest way to create table from existing data with a single command.

### Syntax:

```
CREATE TABLE  
[dbo].[FactInternetSales_new]  
AS  
SELECT * FROM  
[dbo].[FactInternetSales];
```



# Threads	Duration	MB/sec	GB/hr
1	0:01:31	1,912	6,721
3	0:01:57	4,461	15,683
5	0:03:53	2,240	7,875
10	0:06:11	4,689	16,486

Create  
table as  
select →

### Fabric Workspace

Query joining two or  
more tables from  
different warehouses



Warehouse1.  
TableA



Warehouse2.  
TableB

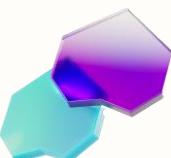


New Table

Warehouse3.  
TableC



Warehouse4.  
TableC



# Data Agents



# AI-powered experiences in Fabric

Gen AI accelerates your data journey



Copilot accelerated  
experiences



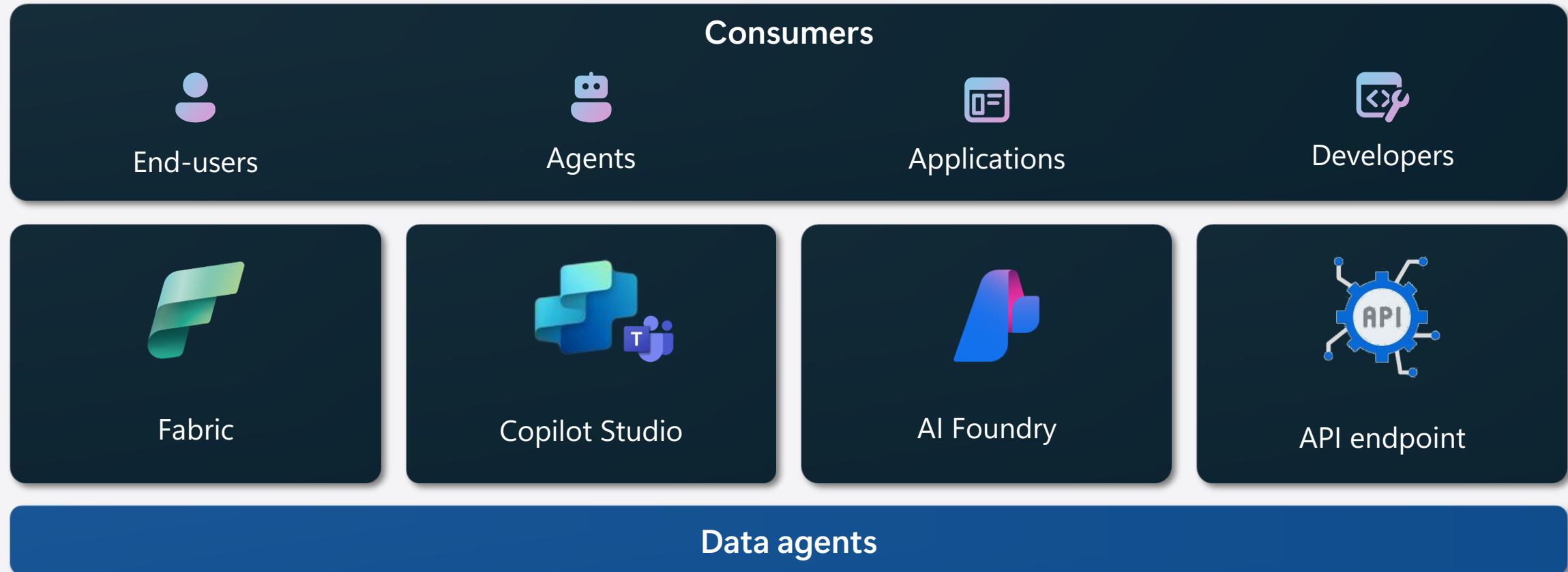
Chat with your data  
experiences



Custom generative  
AI for your data

# Why build Fabric Data Agents?

To ground your AI solutions in Enterprise Knowledge



# Integration with Copilot Studio and Azure AI Foundry

The image displays two side-by-side screenshots of cloud-based AI development platforms.

**Copilot Studio Screenshot:** This screenshot shows the interface for creating and managing agents. It features a sidebar with navigation links like Home, Create, Agents, Flows, and Tools. The main area is titled "Master-Sales-Agent" and includes sections for "Your agent is ready! Here's what's next:", "Details" (Name: Master-Sales-Agent), "Description" (Master Agent that answers queries about customer sales and their behavioral habits), "Orchestration" (Enabled), "Response model" (GPT-4o (default)), "Analytics" (Total sessions: 9, Engagement: 0%, Satisfaction score: --), and "Instructions" (Type your instructions here...). A central panel shows a preview of the agent's responses, and a bottom panel provides steps for defining and exploring the agent.

**Azure AI Foundry Screenshot:** This screenshot shows the "Overview" page for a project named "zhuoqunli-5026". It includes a sidebar with links for Overview, Model catalog, Playgrounds, AI Services, Agents, Templates, Fine-tuning, Prompt flow, Assess and improve, Tracing, Evaluation, Safety + security, and My assets. The main content area shows the AI Services resource (ai-agentswestus2093072344981), API Key (redacted), Included capabilities (Azure OpenAI Service), and Project details (Subscription: Azure OpenAI - Agents - Development (Mumford), Subscription ID: 47fc1c14-e299-4953-a99d-3e34644cfec1, Location: westus2). It also features a "Nail the basics with these steps" section for "Define and explore", "Build and customize", and "Assess and improve".

Fabric data agents can reason over and synthesize data in OneLake for insights

Build accurate, relevant, and context-aware multi-agent solutions

Security permissions, including RLS and CLS, are respected

# What do I need to build a Data Agent?

## Data domain knowledge

Entity relationships & business logic

Organizational terminology & semantics

Decision-making context

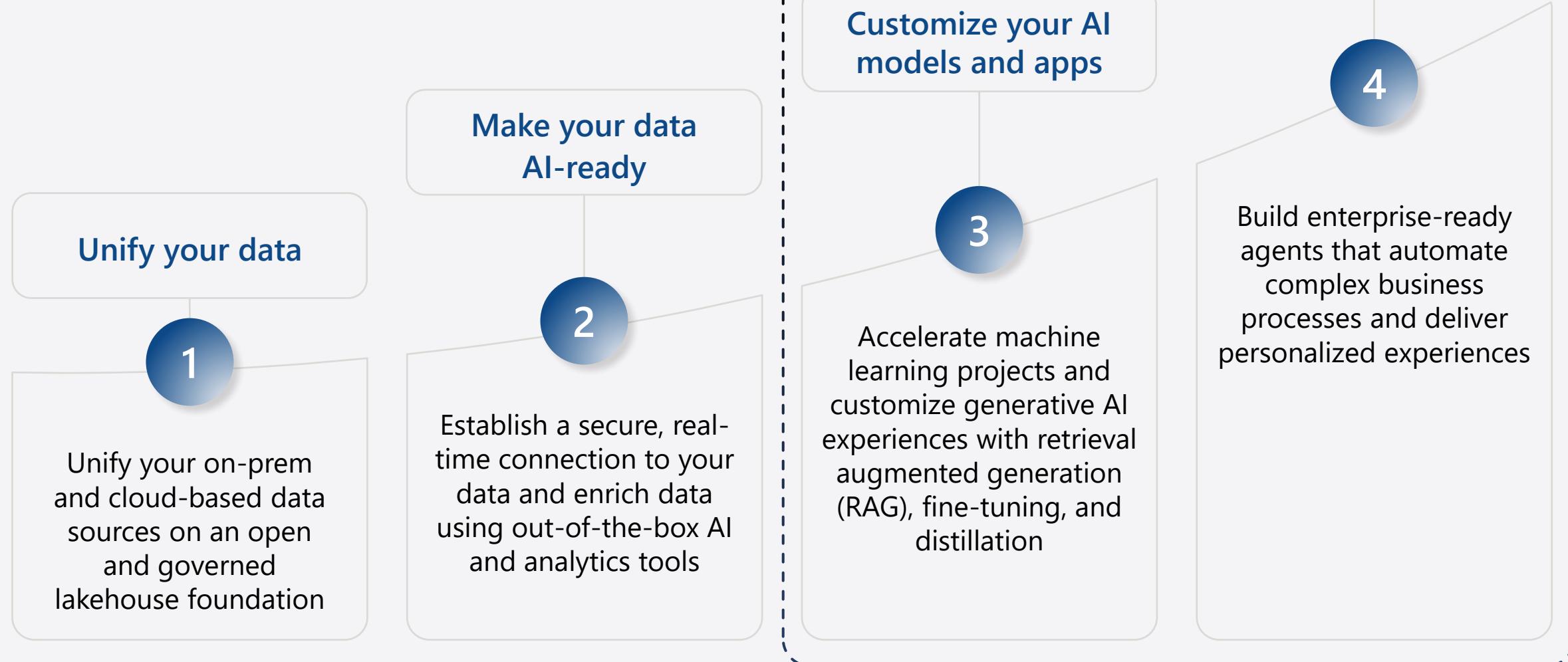
## Data source knowledge

Location & access

Structure & schema

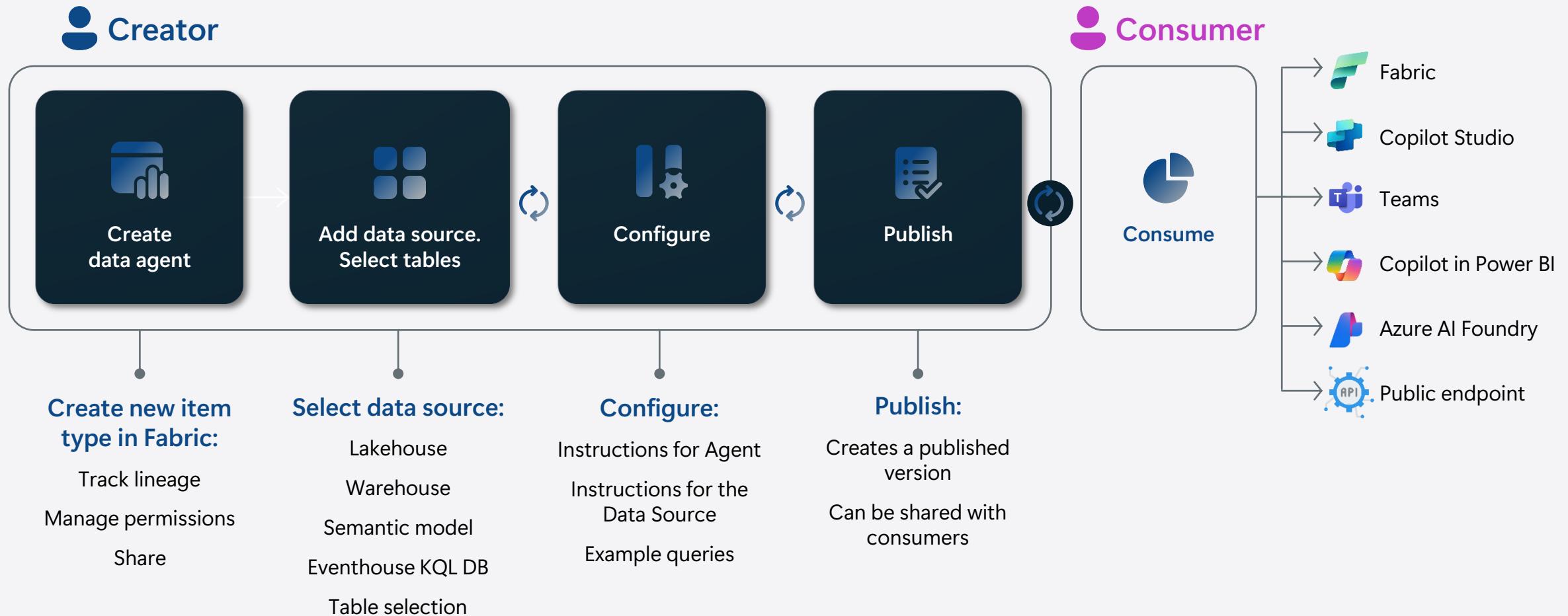
Quality & lineage

# AI is only as good as the data it's based on

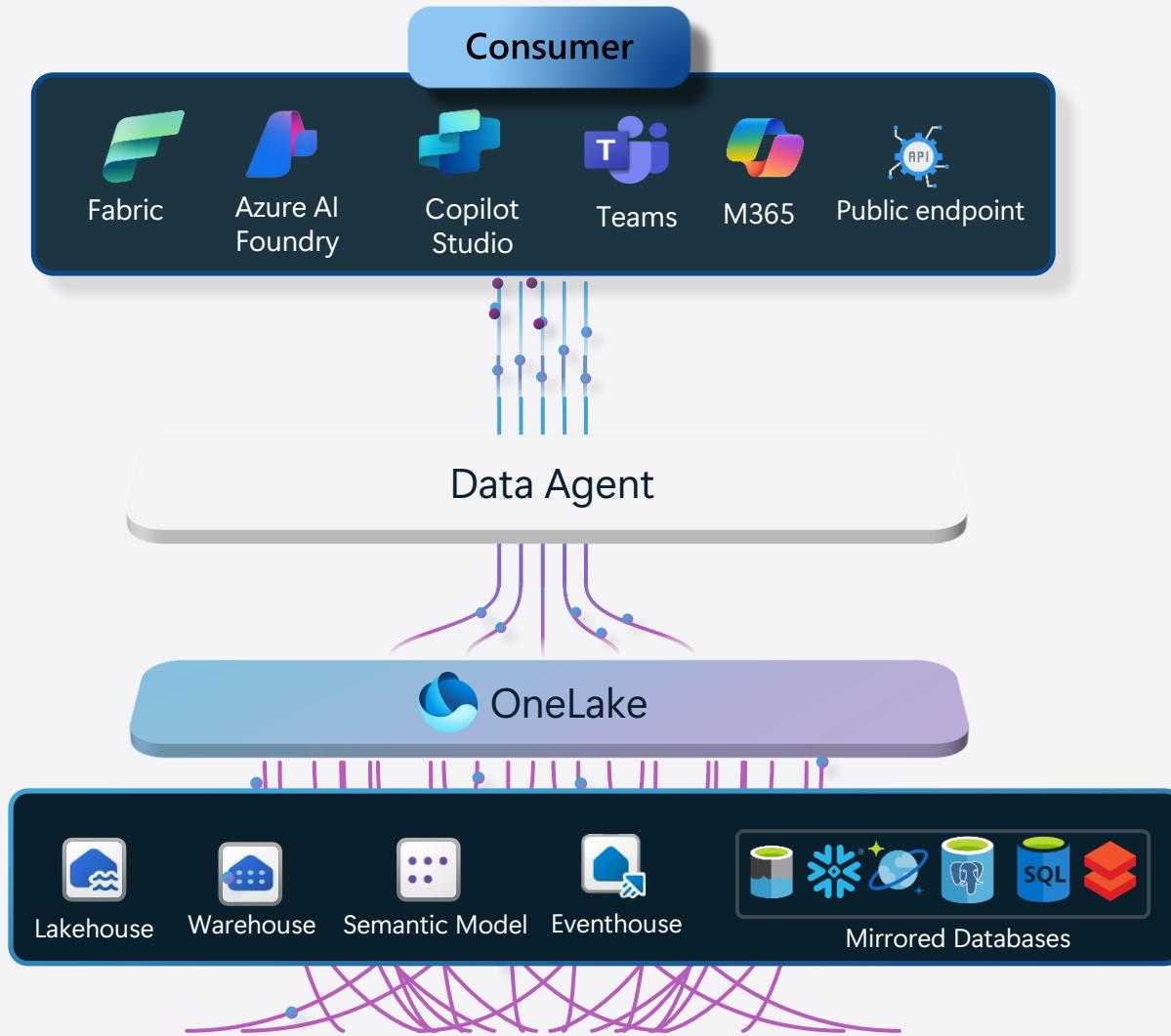


# Data Agent end-to-end flow

## End-to-end scenario



# Fabric Data Agents



This [conversational data agent](#) allows users to interact with a virtual analyst



Tooling for creators from UI and SDK, with [new debugging capabilities](#), making it easier to understand and refine responses

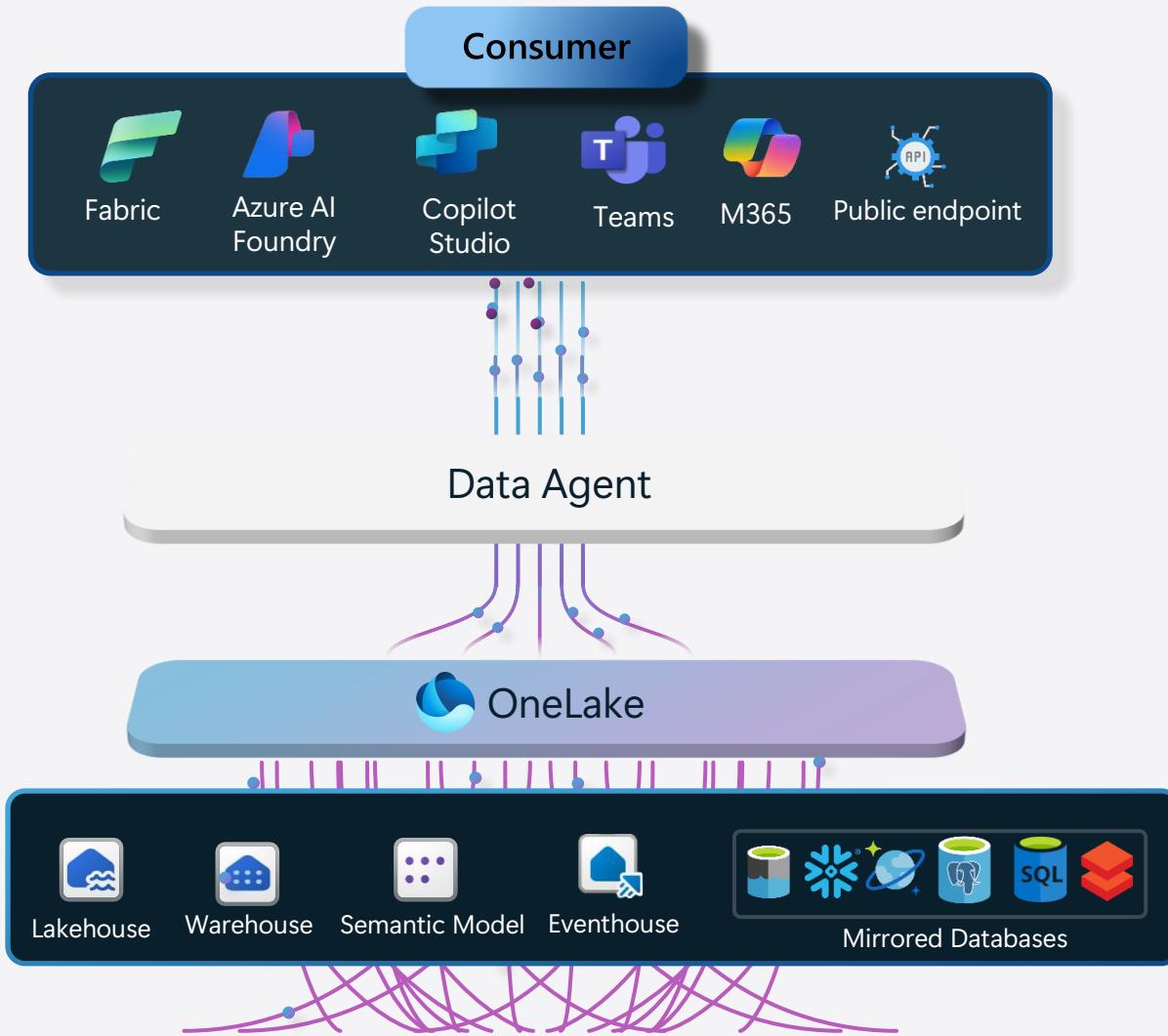


Seamlessly reason over data in OneLake, including [shortcuts](#) and [mirrored databases](#) to reason over [tables](#), [files](#) and [real-time events](#), to create a powerful data expert tailored to your data domain



Your Data Agents can be consumed inside and outside of Fabric. Integrate with [AI Foundry](#), [Copilot Studio](#), [Teams](#) and with your own [custom applications](#)

# Fabric Data Agents



A virtual analyst that allows users to interact with enterprise data in OneLake

New

Seamlessly reason over data in OneLake, including **mirrored databases** to create a powerful data expert tailored to your data domain

New

Support for data agents in **Git** and **deployment pipelines**, enabling agent ops for more **streamlined AI development**

New

Invoke data agents externally with **detailed responses** including **SQL query execution steps**

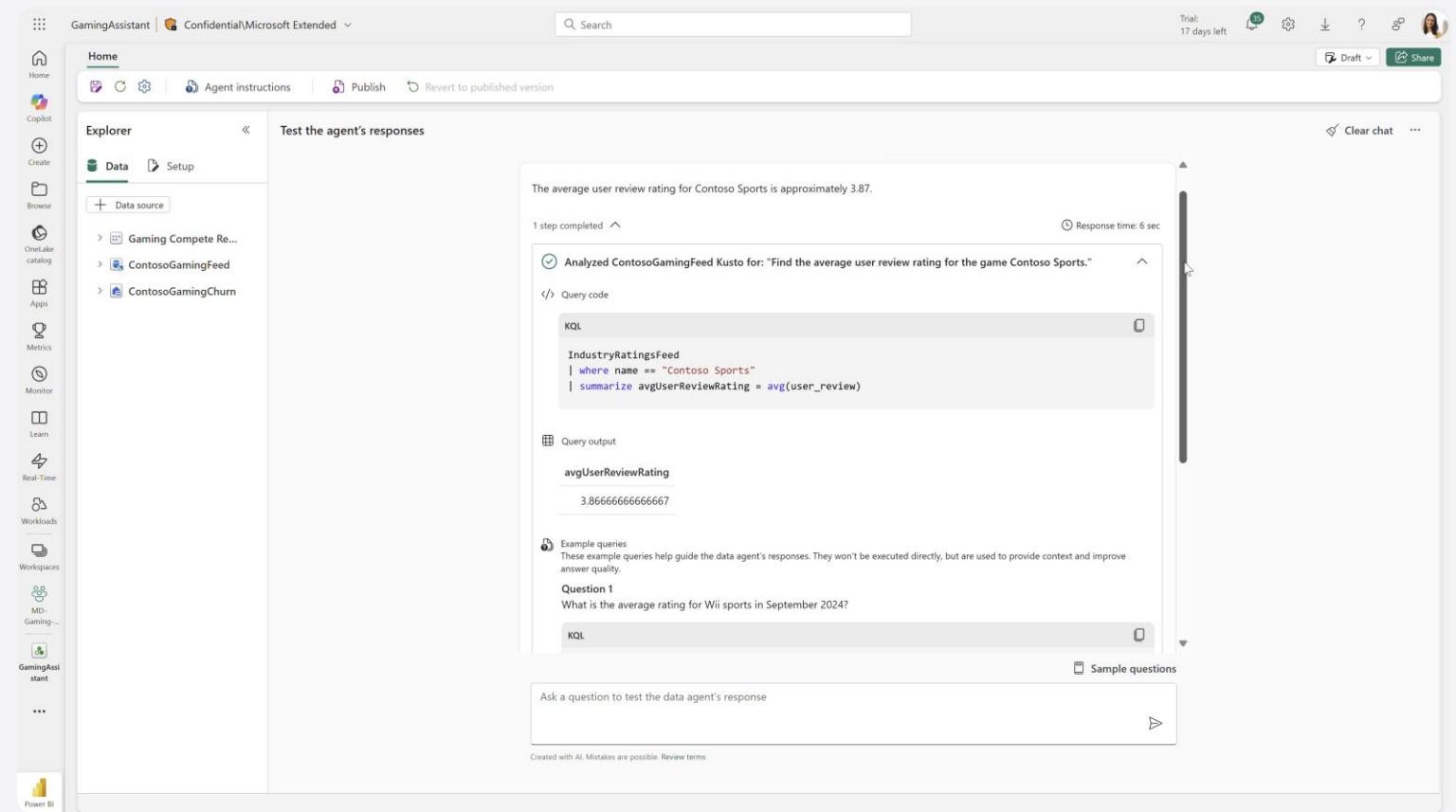
# Creator Tooling Improvements

Add [datasource instructions](#) to communicate table usage and join logic

Use the new [multi-tasking experience](#) for side-by-side authoring and testing

Use [markdown editor](#) for clear, organized instructions

Leverage [diagnostics tools](#) to trace referenced example queries and underlying steps

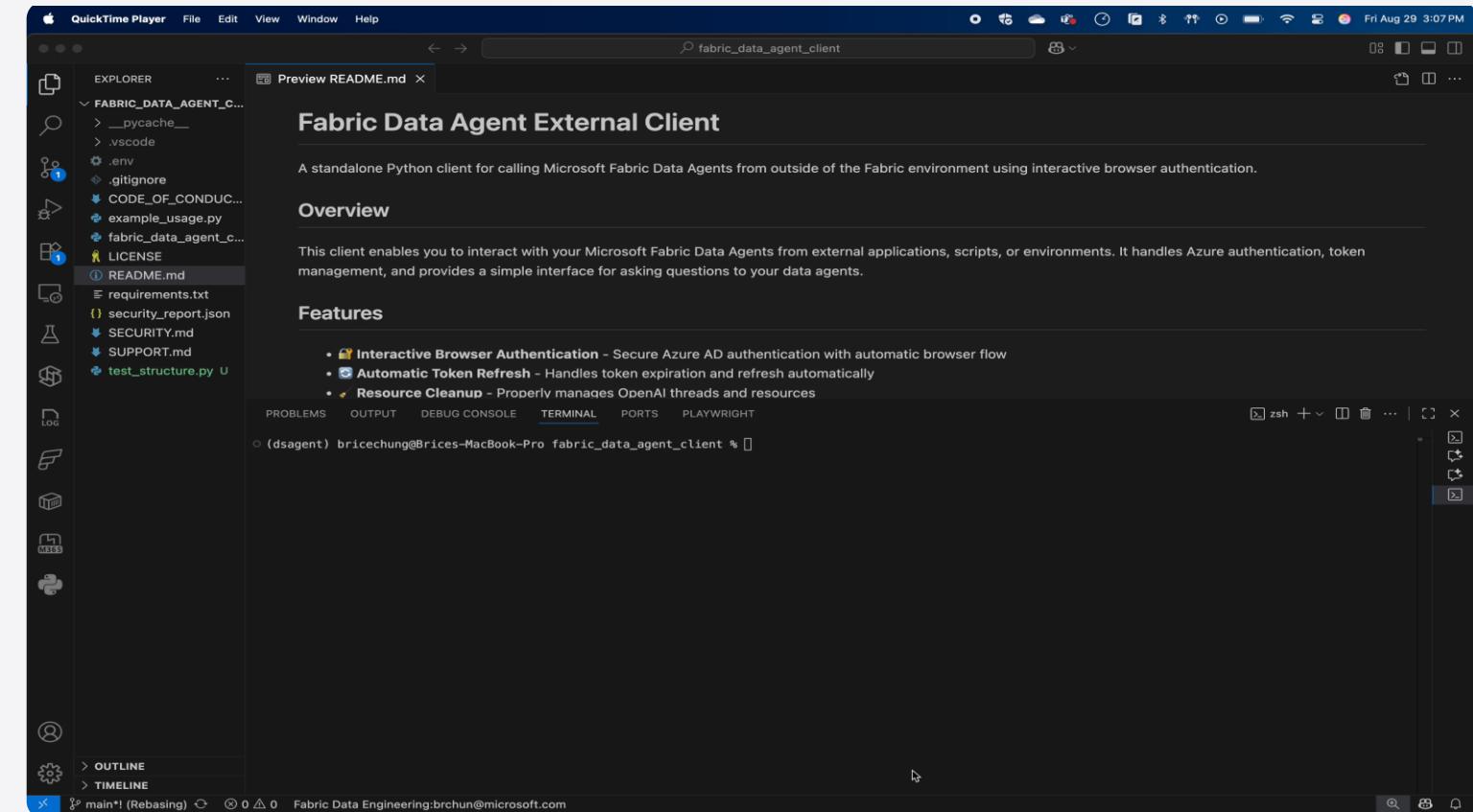


# Fabric Data Agents public endpoint

Connect from VSCode or external applications with secure authentication

Use familiar OpenAI client patterns

Query your data agents externally with detailed responses including SQL query execution steps



# Unstructured Data in Data Agents

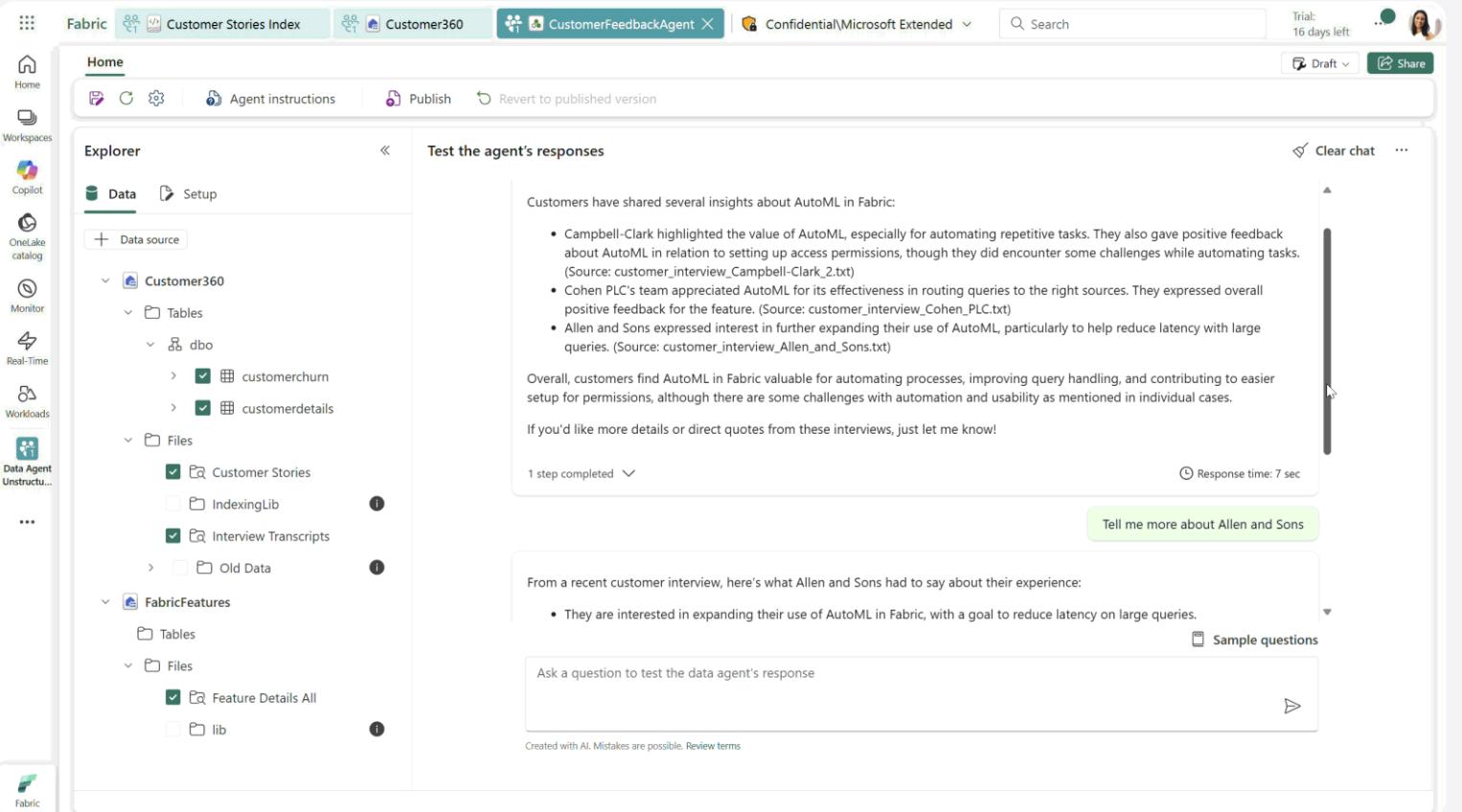
Coming Soon

Leverage built-in notebook utilities to index PDFs and TXT files

Eliminate the need to provision or manage search resources

Select Lakehouse folders and ask natural language questions over unstructured data

Sign up for the Private Preview at  
[aka.ms/unstructured-data-prpr](http://aka.ms/unstructured-data-prpr)



The screenshot shows the Microsoft Fabric Data Agent interface. On the left, the Explorer sidebar displays a tree view of data sources, including Customer360 (Tables: dbo, Files: Customer Stories, IndexingLib, Interview Transcripts, Old Data), and FabricFeatures (Tables, Files: Feature Details All, lib). The main pane is titled "Test the agent's responses". It contains a section titled "Customers have shared several insights about AutoML in Fabric:" with a bulleted list of three items. Below this, a message from "Allen and Sons" discusses their interest in expanding AutoML use. A "Tell me more about Allen and Sons" button is present. At the bottom, there is a text input field "Ask a question to test the data agent's response" and a "Sample questions" link.

# Lab 2

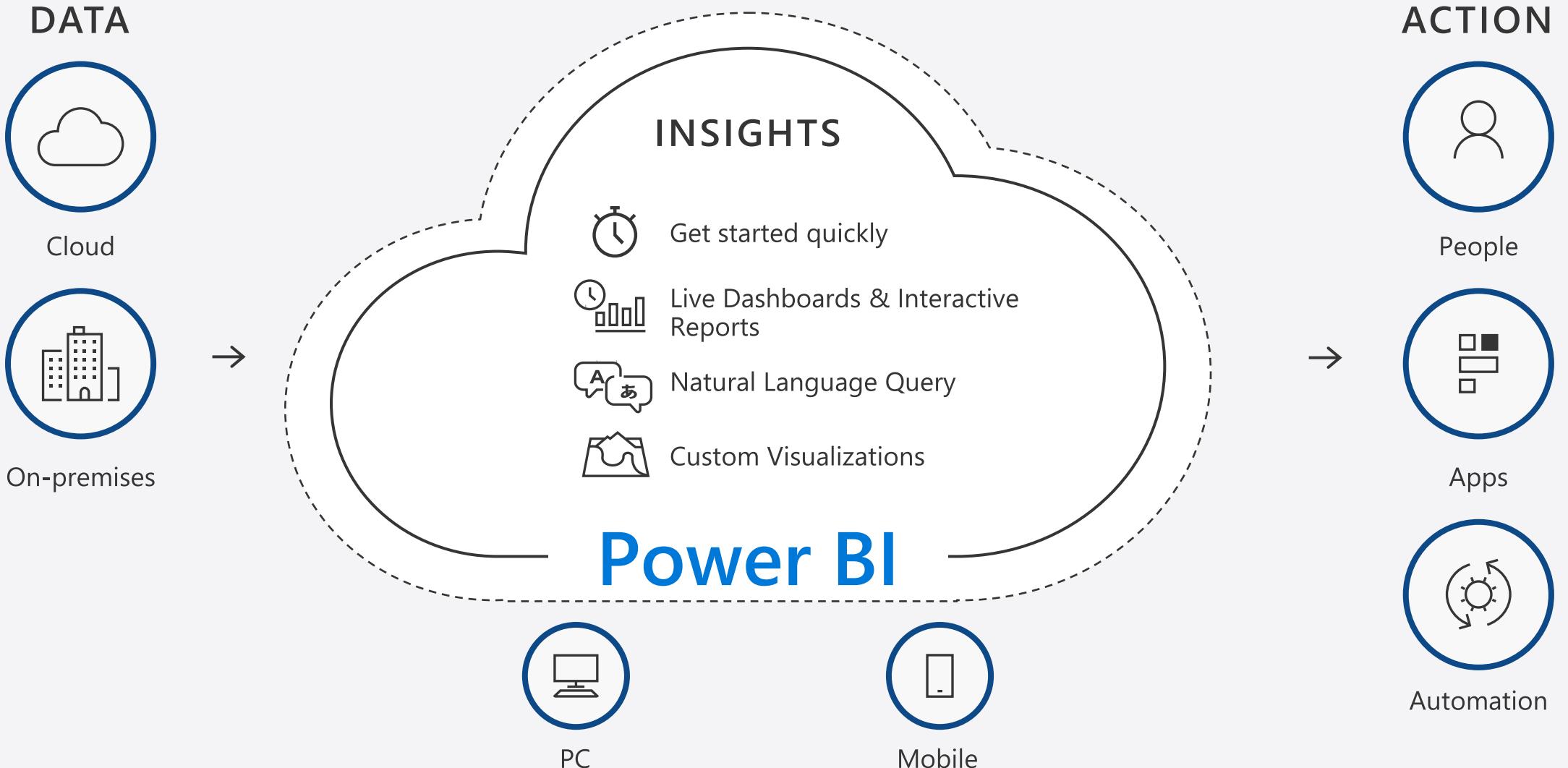


# Power BI



# Power BI: experience your data

Any data, any way, anywhere



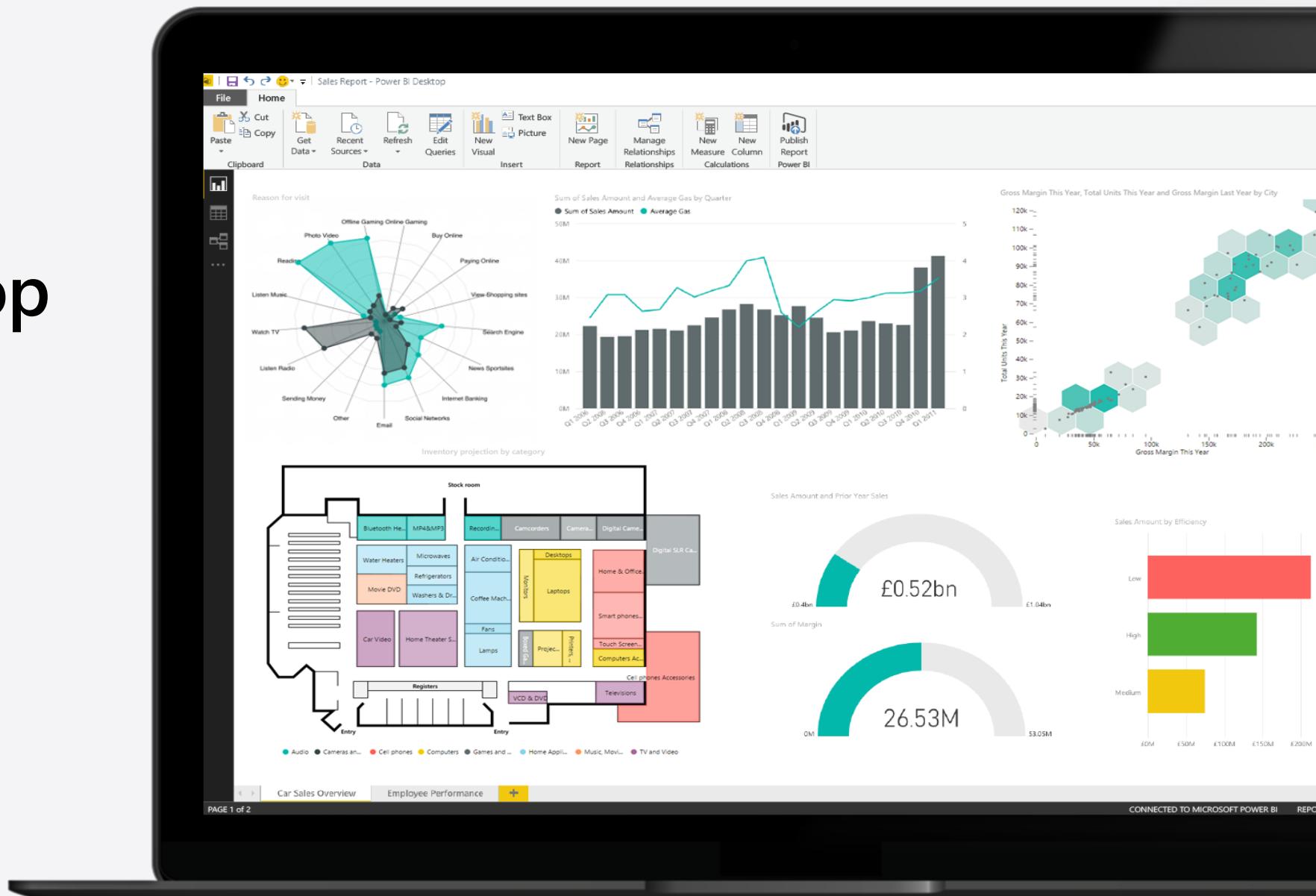
# Create powerful reports with Power BI Desktop

Discovery & exploration

Easy report authoring

Custom visualizations

R integration



# Connect to all the data around you

Connect to data in the cloud and on-premises

Shape, transform, and clean data for analysis

Join and model data from multiple sources/types

Extend with advanced analytics technologies like R



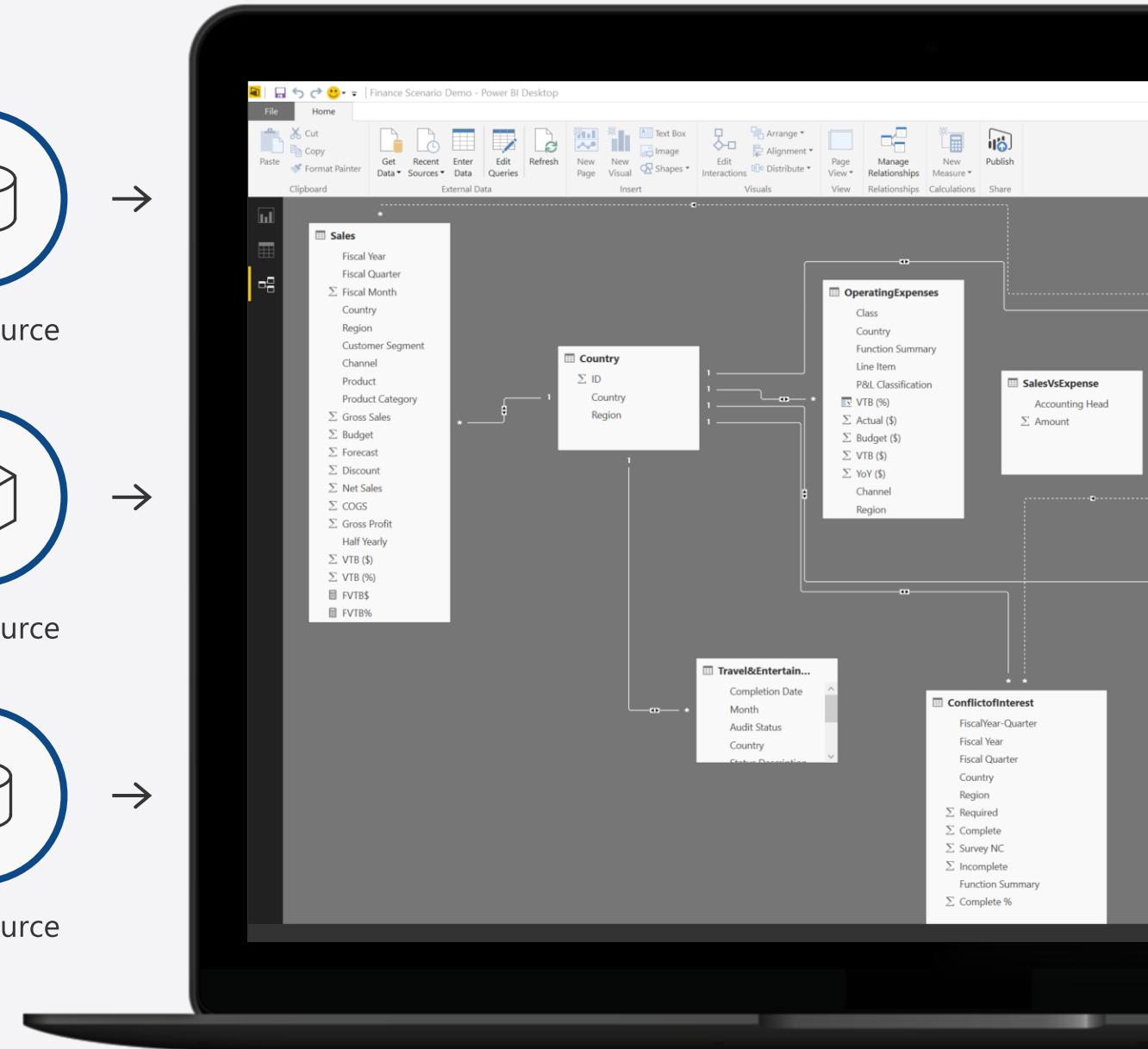
Data source



Data source



Data source

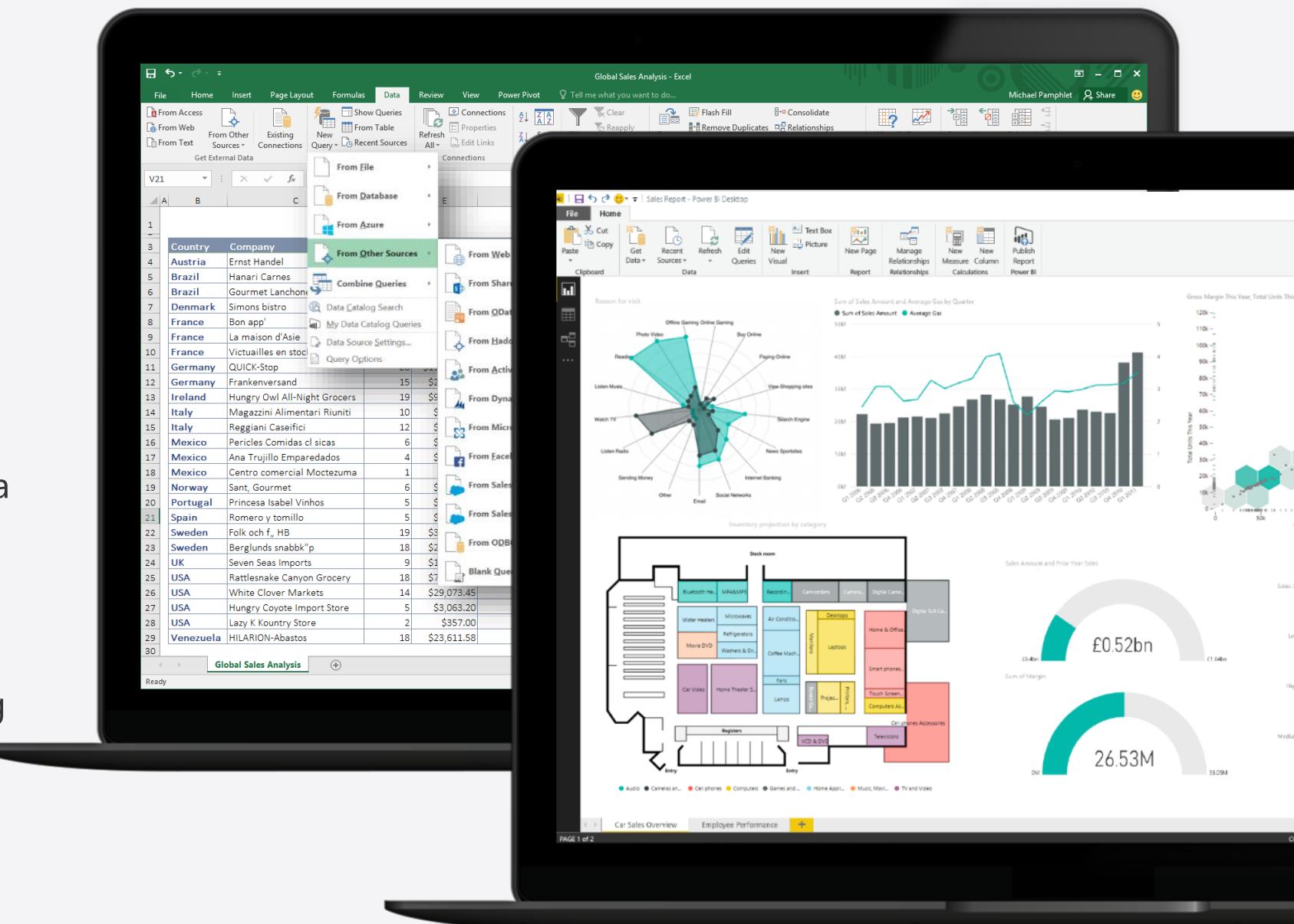


# Powerful self-service analysis

**Web**  
A full analytical experience from data modeling to visualization

**Power BI Desktop**  
Visual drag-and-drop data exploration and interactive reporting

**Excel**  
Ad-hoc analysis for blending structured and unstructured data



# Components of Power BI

## Semantic model

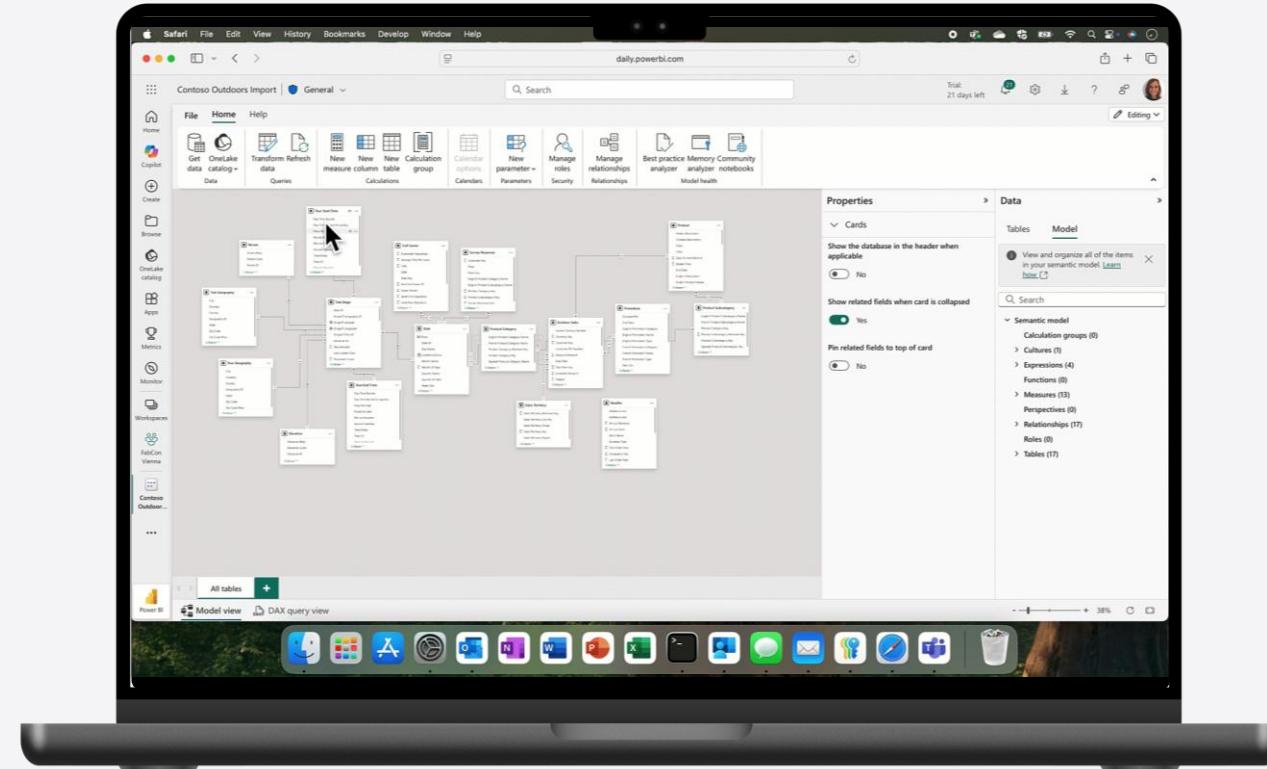
The governed, reusable **business layer**—tables, relationships, and DAX measures—that powers reports, dashboards, Excel, and Copilot across the org.

## Report

An **interactive visual experience** built on a semantic model to communicate insights for a specific audience or task.

# Web Modelling in Power BI

All the capabilities you know and love, now in your browser



Create models and  
reports from your browser,  
Mac users included

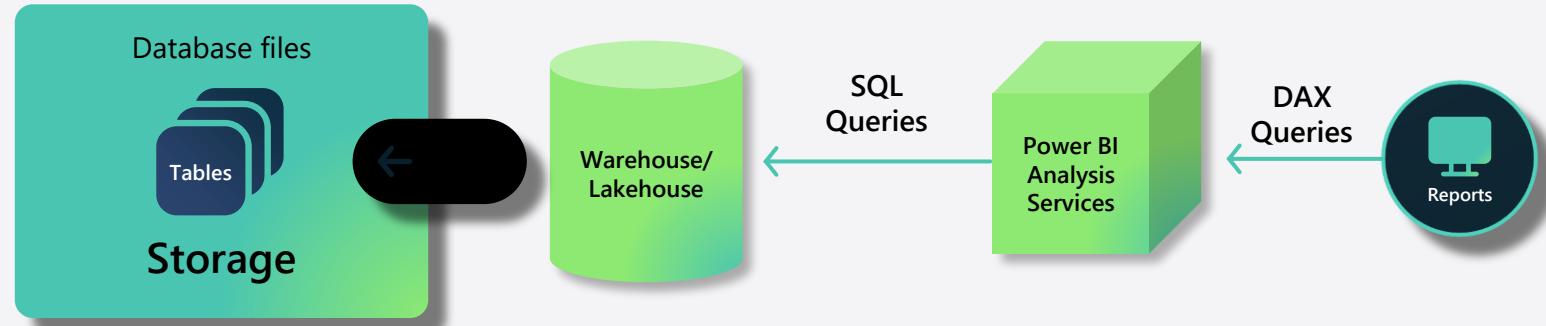
Combine Direct Lake  
and Import modes for  
composite models

Power Query support  
for Fabric—plus 100s  
of other sources

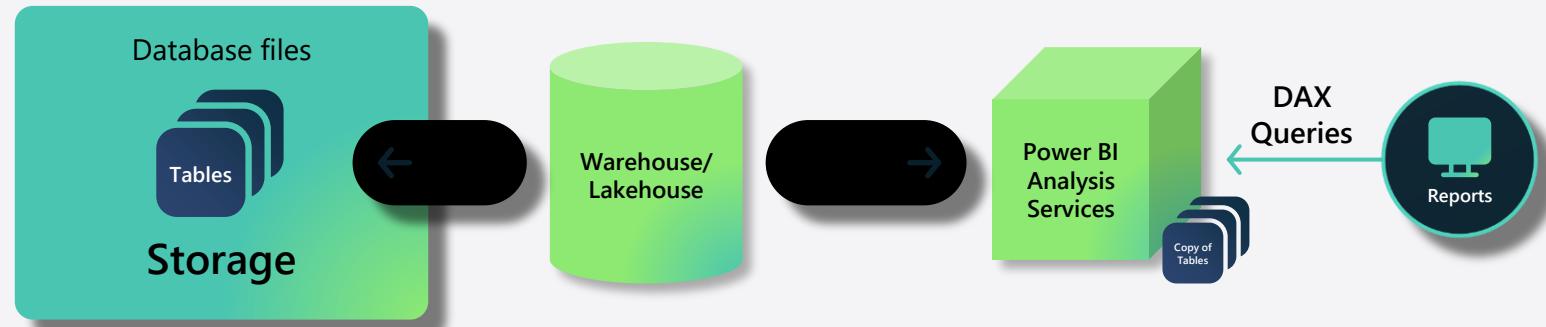
Seamless integration  
with OneLake and  
all of Fabric

# Direct Lake mode for blazing performance

Direct Query Mode  
Slow, but real time

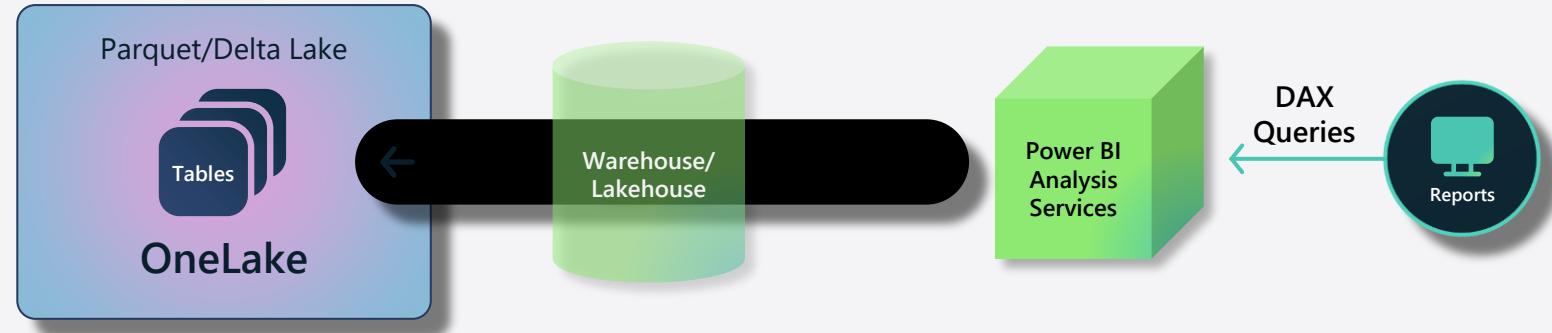


Import Mode  
Fast, but duplicative



# Direct Lake mode for blazing performance

Direct Lake Mode  
Fast, real time, no duplication



Optimized data layout

Direct Lake mode

Seamless fallback

# Query performance results



Power BI DirectQuery  
(Snowflake)

Performance analyzer

Start recording Refresh visuals Stop

Clear Export

Name	Duration (ms)
Sum of ADJUSTEDREVENUEAMT	12372
Changed a filter	-
Card	72
Edited a visual	-
Card	446
Sum of ACTUALREVENUEAMT	4275
Changed a filter	-
Sum of ACTUALREVENUEAMT by FISCAL_MO...	8382
Changed a filter	-
Sum of ACTUALREVENUEAMT by FISCAL_MO...	6163
Changed a filter	-
Sum of ACTUALREVENUEAMT by FISCAL_MO...	13901
Cross-highlighted	-
Card	5574

Total query time: 51,185 ms



Power BI Direct Lake  
(Fabric DW)

Performance analyzer

Start recording Refresh visuals Stop

Clear Export

Name	Duration (ms)
Sum of AdjustedRevenueAmt	2034
Changed a filter	-
Card	70
Edited a visual	-
Card	232
Sum of ActualRevenueAmt	424
Changed a filter	-
Sum of ActualRevenueAmt by Fiscal Month	1049
Changed a filter	-
Sum of ActualRevenueAmt by Fiscal Month	138
Changed a filter	-
Sum of ActualRevenueAmt by Fiscal Month an...	1213
Cross-highlighted	-
Card	626

Total query time: 5,786 ms

# Copilot





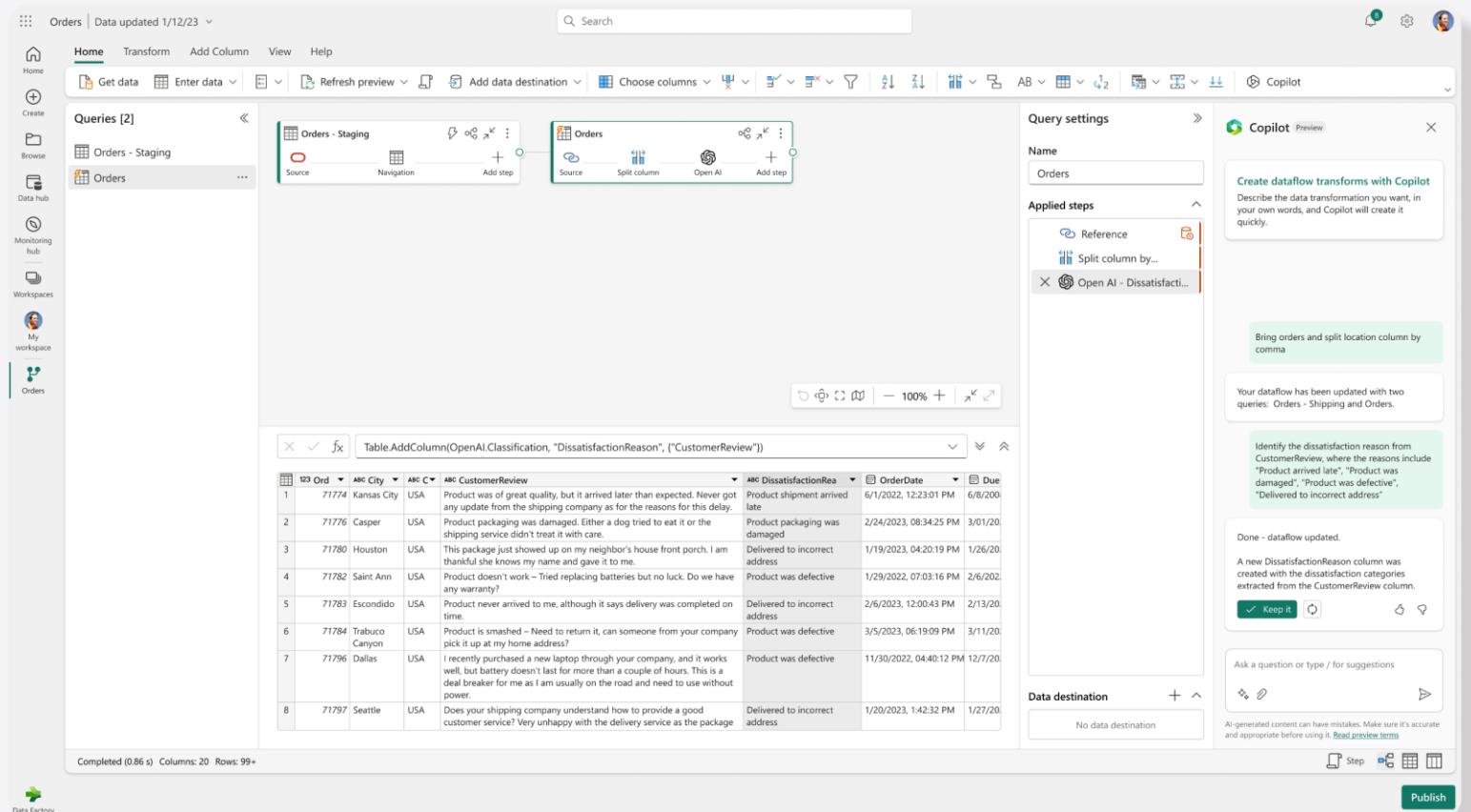
# Copilot in Data Factory

Easily integrate generative AI into your dataflows and pipelines using Copilot

 Chat with **Copilot** to describe data transformations in natural language

 Tap into generative AI capabilities from **Azure Open AI** as data transformation steps

 Use **Copilot** to schedule and run and manage dataflows



The screenshot shows the Microsoft Data Factory interface. On the left, there's a sidebar with options like Home, Create, Browse, Data hub, Monitoring hub, Workspaces, and My workspace. The main area shows a dataflow named 'Orders' with two stages: 'Orders - Staging' and 'Orders'. The 'Orders' stage includes a 'Source' step, a 'Navigation' step, and an 'Add step' button. To the right of the stages is a 'Query settings' panel with a 'Copilot' tab. The 'Copilot' tab has a 'Name' field set to 'Orders' and a 'Create dataflow transforms with Copilot' section. Below this, the 'Applied steps' section lists 'Reference' and 'Split column by...'. A preview window shows a table with columns: Ord, ABC City, ABC C, ABC CustomerReview, ABC DissatisfactionRea, OrderDate, and Due. The 'CustomerReview' column contains customer feedback, and the 'DissatisfactionRea' column contains generated dissatisfaction reasons. The 'Data destination' section at the bottom indicates 'No data destination'. At the bottom right, there are 'Step', 'Publish', and other UI elements.



# Copilot in Notebooks

**Use Copilot to enrich, model, analyze and explore your data in notebooks**



Work with **Copilot** to understand how best to analyze your data



Chat with **Copilot** to create and configure ML models



Write code faster with inline code suggestions from **Copilot**



Use **Copilot** to summarize and explain code to understand how it works

Sales analysis | Data updated 1/12/23

Home Edit Run Data science View

Comments Editing Share

Lakehouse explorer

- Customer360
- Tables
  - Customer
  - CustomerFeedback...
  - Inventory
  - Product
  - Sales
  - Transaction
  - Unidentified
- Files
- Sales
- Excel-data

# Welcome to your new notebook  
# Type here in the cell editor to add code!  
\* Press shift + enter to execute cells

```
1 # Welcome to your new notebook
2 # Type here in the cell editor to add code!
3 * Press shift + enter to execute cells
4
5
6
7
8
```

# AI-generated code

```
1 # AI-generated code
2
3 import pandas as pd
4 customer_data = pbi.read_table('CustomerProfitabilitySamplePBIX', 'Customer')
5 sales_data = pbi.read_measure('CustomerProfitabilitySamplePBIX', 'Total Revenue', [('Customer', 'Name')])
6 customer_sales = customer_data.merge(sales_data, on='Name')
7 print(customer_sales.head())
8
```

(2) ✓ - Command executed in 7 sec 427 ms by Sonia

Python

Table Chart Data profile Export result Showing rows 1 - 1000 Statistics Search

Country/Region Ab	Name Ab	City Ab	Industry ID ##
1 US	Energy	Irving	31.0
2 US	Materials	Chicago	30.0
3 US	Capital Goods	Westchester	30.0
4 US	Transportation	Plano	13.0
5 US	Consumer Services	Fort Worth	34.0
6 US	Food, Beverage & Tobacco	Irving	31.0
7 US	Pharmaceuticals	Chicago	30.0
8 US	Energy	Westchester	30.0
9 US	Materials	Plano	13.0
10 US	Capital Goods	Fort Worth	34.0
11 US	Transportation	Irving	31.0
12 US	Consumer Services	Chicago	30.0
13 US	Food, Beverage & Tobacco	Westchester	30.0
14 US	Pharmaceuticals	Plano	13.0
15 US	Financial Services	Fort Worth	34.0
16 US	Insurance	Irving	31.0
17 US	Software & Services	Chicago	30.0

Code Markdown

Session ready dev / my\_feature\_branch Save option: Automatic

Copilot Preview

Transform your data with Copilot

Describe what you want, in your own words, and Copilot will suggest code you can use.

Can you create a DataFrame for me which segments my data by industry?

Here's a pandas DataFrame query that pulls in relevant data about the company's sales history and profitability for 2022.

```
1 import pandas as pd
2 customer_data =
3 pbi.read_table('CustomerProfitabilitySamplePBIX', 'Customer')
4 tySamplePBIX', 'Customer')
5 sales_data =
6 pbi.read_measure('CustomerProfitabilitySamplePBIX', 'Total Revenue', [
7 ('Customer', 'Name')])
8 customer_sales =
9 customer_data.merge(sales_data,
10 on='Name')
11 print(customer_sales.head())
12
```

Ask a question or request, or type / for suggestions

All generated content can have mistakes. Make sure it's accurate and appropriate before using it. Read [license terms](#).

Selected Cell 1 in 1 cells



# Copilot in Data Warehousing

**Use Copilot to work with your data and generate SQL queries**



Write code faster with inline code suggestions from **Copilot**



Let **Copilot** clean-up and document your code



Use **Copilot** to accelerate data warehouses migrations by automatically updating queries and stored procedures

The screenshot shows the Microsoft Power BI Data Warehouse interface. On the left, there's a sidebar with navigation links like Home, Create, Browse, Data hub, Monitoring hub, Workspaces, and Sales analysis. The main area has tabs for Home, Explorer, and Copilot. The Explorer tab is active, showing a tree view of a Sales Analytics warehouse with Schemas (dbo), Tables (Customer, Date, Geography, Products, Sales, Views, Stored procedures, Functions, Information\_schema), and Queries (My queries). A query named "SQL query 1" is selected. The Copilot tab is also visible. The central part of the screen displays an SQL query:

```
1 SELECT top 10
2     c.Customer as Customer,
3     SUM(S.Sales) Sales
4 FROM Sales s
5     inner join Date d on d.date = s.invoice_date_key
6     inner join Customer c on c.customer_key = s.customer_key
7 WHERE d.Calendar_Year <= YEAR(DATEADD(MONTH, -24, GETDATE()))
8 group by C.Customer
9 order by SUM(s.sales)
10
11 CREATE OR ALTER FUNCTION dbo.TopCustomersBySales (@months int)
12 RETURNS TABLE
13 AS
14 RETURN (
15     SELECT top 10
16         c.Customer as Customer,
17         SUM(S.Sales) Sales
18     FROM Sales s
19     inner join Date d on d.date = s.invoice_date_key
```

Below the query, there's a results table titled "Results 1" showing 10 rows of data:

Customer	Sales
Contoso	640487122.59
Adventure Works Cycles	655245582.38
Relecloud...	512378605.96
Contoso Pharmaceuticals	500879293.45
Ned Publishers...	492876225.56
Consolidated Messenger	38915282.96
Contoso Pharmaceuticals...	95242585.12
Relecloud	84232922.89
Consolidated Messenger	79205631.12
Adventure Works Cycles	52592128.91

The status bar at the bottom says "Succeeded (3.66 s)". On the right side, there's a "Copilot Preview" pane with some AI-generated SQL code and a message about creating a table-valued function.



# Copilot in Power BI

**Stay focused on your business outcomes and unlock insights in your data with Copilot**

Create beautiful and insightful reports just by chatting with **Copilot**

Define metrics and calculations for your data model just by describing them in **natural language**

Use **Copilot** to find and summarize insights in your data

The screenshot shows a Power BI report titled "Contoso Daily Sales" with a data update of 1/12/23. The main area displays a "Sales Overview" card with metrics: Revenue Won (\$7,811,851), Close % (37.7%), AVG Days to Close (121), and Opportunities Won (526). Below this are three charts: "Revenue Won by Month" (line chart showing growth from June 2022 to January 2023), "Close % by Month" (bar chart showing monthly close percentages from May to January), and "Close % by Region" (map of the United States color-coded by region). To the right, a sidebar titled "Copilot Preview" shows AI-generated insights and suggestions:

- Create a report to identify trends in sales and promotions to inform marketing strategies.
- Sales overview page added
- What are the biggest drivers for close %?
- Close % drivers page added
- Callback within 3 hours (11.75%) has the biggest influence on close %.
- Add a narrative summary describing insights in the data.
- Summary created
- Summary in bullet points

At the bottom, there is a note: "Ai-generated content can have mistakes. Make sure it's accurate and appropriate before using it. [Read preview terms](#)".



# Chat with your data

## Something here



Ask Copilot to find reports, semantic models, apps, and data agents



Copilot's summary allows you to quickly identify the most interesting data within your report



Ask questions of Fabric's AI powered data assistants, data agents

The screenshot shows the Microsoft Power BI Copilot interface. The top navigation bar includes Microsoft DXT, Power BI, and Copilot. A search bar is at the top right, along with various icons for notifications, settings, and help. On the left is a sidebar with icons for Home, Copilot (selected), Create, Browse, OneLake catalog, Apps, Metrics, Monitor, Learn, Real-Time, Workspaces, pbifabricdotcomnotdelete, and Power BI. The main area features a "Copilot Preview" section with a "Hi Tori" logo and the tagline "Uncover insights in your data with the help of AI". Below this are three cards: "Find an item", "Get a summary", and "Ask a question", each with a list of AI prompts. At the bottom is a large input field with a placeholder "+ Add items for better results" and a "View prompts" button.

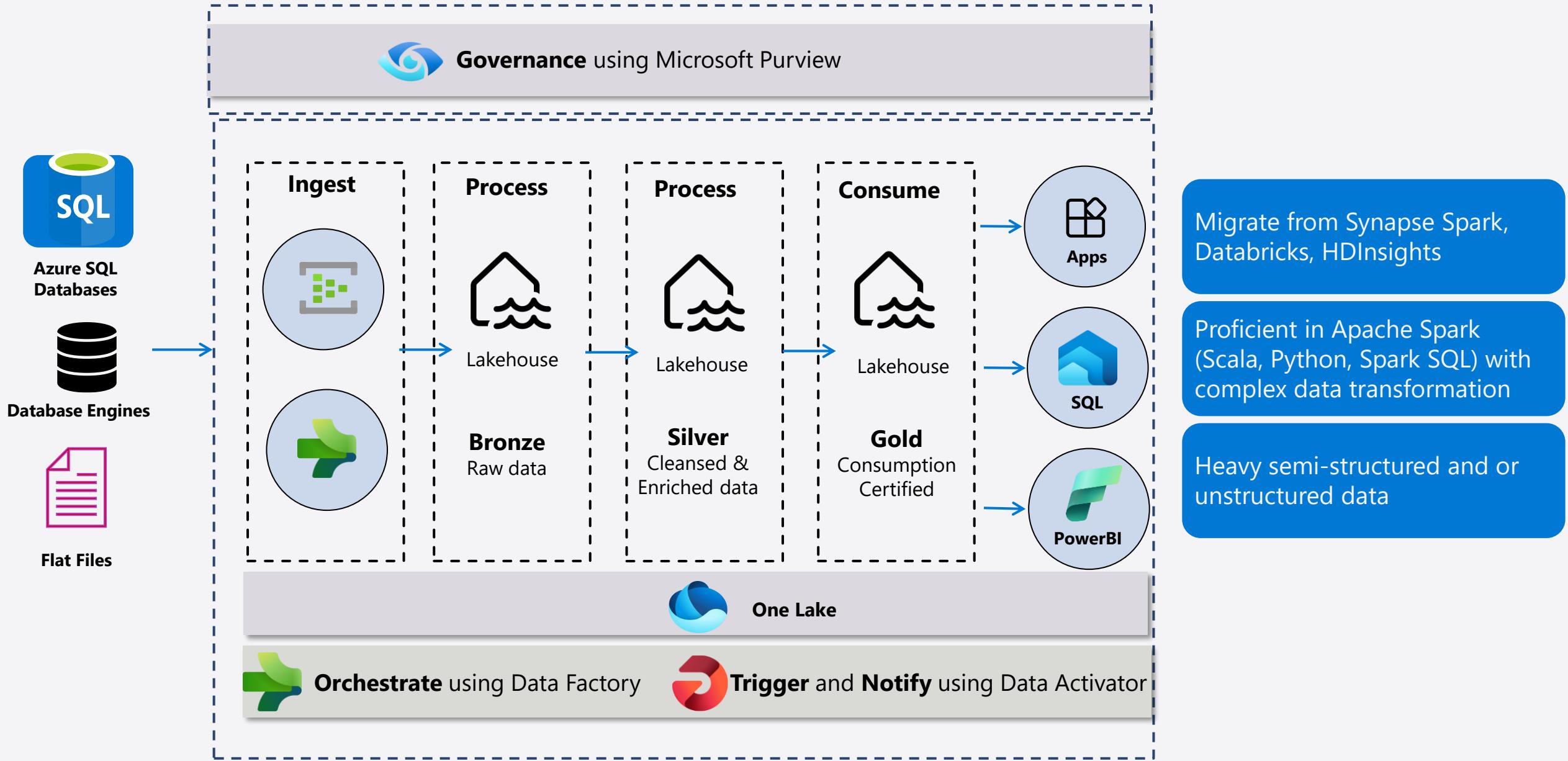
# Lab 3



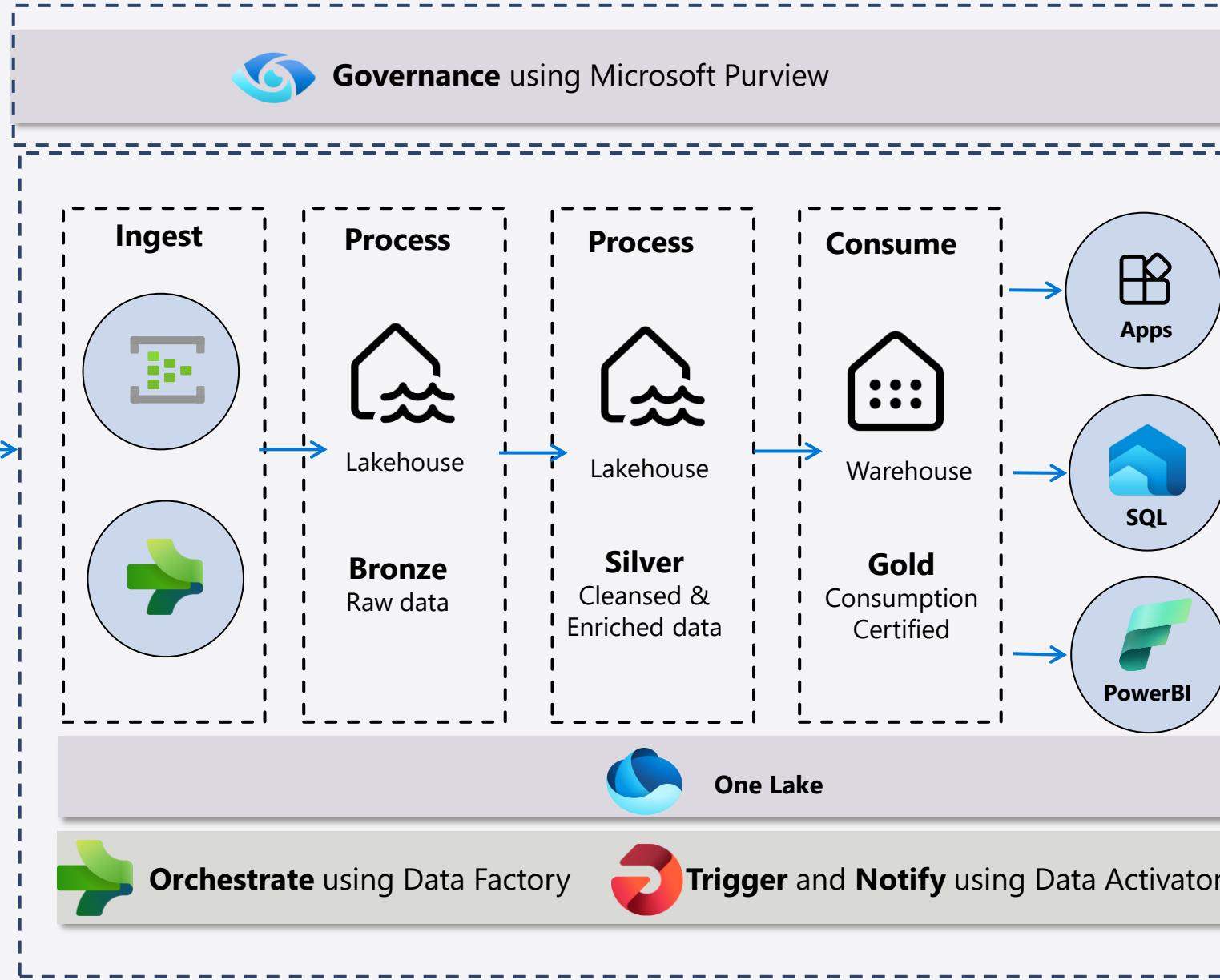
# Architecture Patterns



# The Lakehouse Pattern



# The Hybrid Pattern



Migration can be Greenfield, or traditional Warehouse or Lakehouse architecture

Mixed skillets or multiple teams collaboration

Full benefits of Fabric Analytic for a OneLake architecture

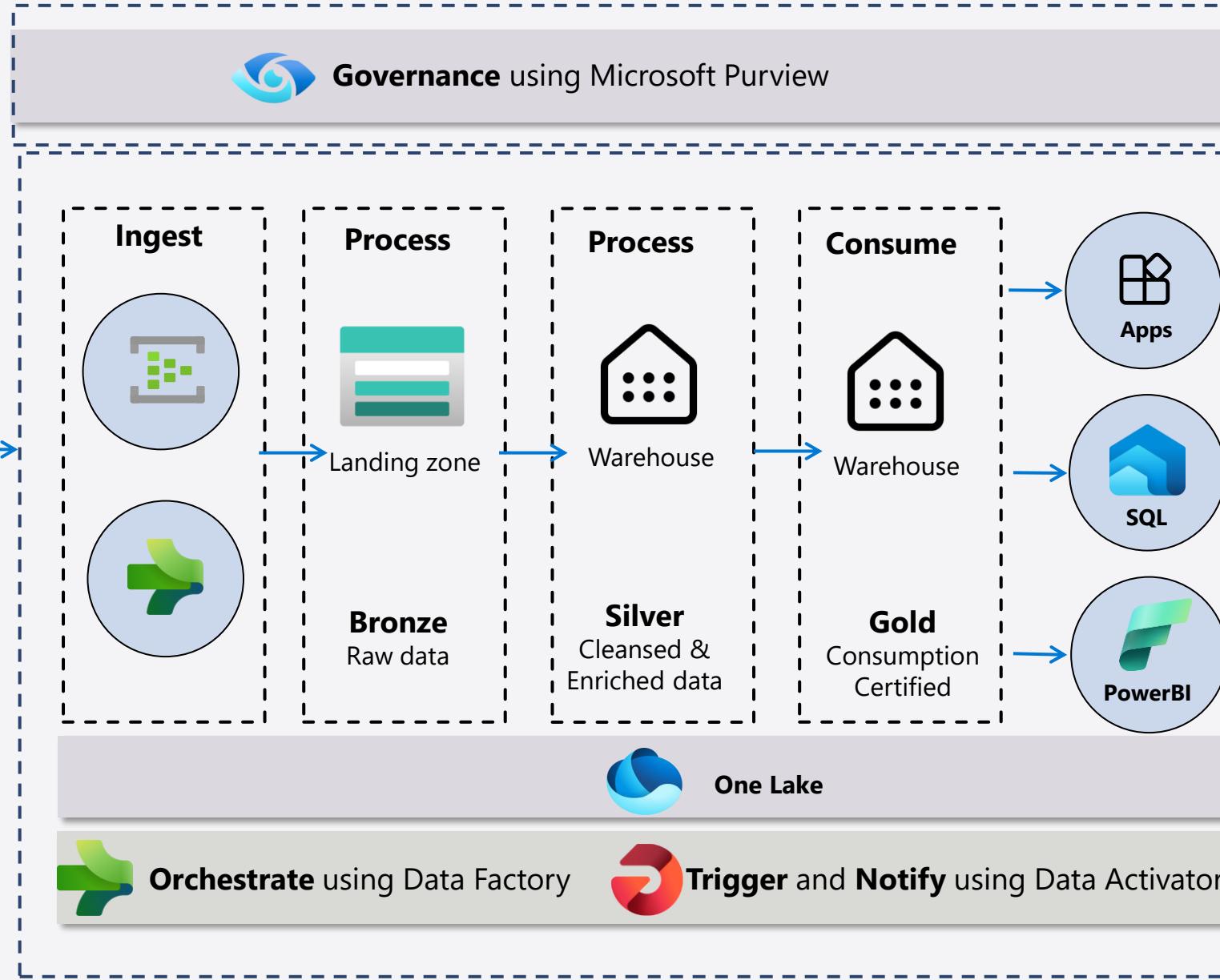


Orchestrate using Data Factory



Trigger and Notify using Data Activator

# The Warehouse Pattern



Migrate from Synapse Dedicated Pool or other RDBMS databases

Microsoft heavy investments on migration programs

SaaS benefits: Auto maint, auto tuned, SIMPLIFIED



Home



Workspaces



Copilot



OneLake catalog



Monitor



Real-Time



Workloads



fabricperuser12345678

...

fabricperuser12345678



Create deployment pipeline



Create app



Manage access



Workspace settings

+ New item

New folder

Import

Migrate

Filter by keyword

Filter

Filter

Add

### Load + Prepare

Prepare data

1 item

+ New item



### Lakehouse

Store data

2 items

+ New item



### Conversational AI

Analyze and train data

1 item

+ New item



### Load + Model

Prepare data

1 item

+ New item



### Data Warehouse

Store data

1 item

+ New item



### Semantic Model

Store data

1 item

+ New item



### Visualize

Visualize data

1 item

+ New item



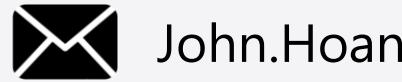
Name	Status	Type	Task	Owner	Refreshed	Nex
Build Dimensional Model		Notebook	Load + Model	Brad Schacht	—	—
Customer Demographics and Country Region Analysis		Report	Visualize	fabricperuser1...	10/23/2025, 12:...	—
MarketResearch		Lakehouse	Lakehouse	Brad Schacht	—	—
MarketResearch		SQL analytics end...	Lakehouse	Brad Schacht	—	—
Prepare Market Research		Notebook	Load + Prepare	Brad Schacht	—	—
Sales		Warehouse	Data Warehouse	Brad Schacht	—	—
Sales Analysis Agent		Data agent	Conversational...	Brad Schacht	—	—
Sales and Marketing		Semantic model	Semantic Model	fabricperuser1...	10/23/2025, 1...	N/A



Fabric

# John Hoang

Principal Program Manager, Fabric CAT  
Orange County, CA



John.Hoang@microsoft.com



@MSJohnPaulHoang



in/johnhoang869773299

<https://aka.ms/AnalyticsPreDayIgnite2025>

<https://aka.ms/FabricIdeas>

# Brad Schacht

Principal Program Manager, Fabric CAT  
Saint Augustine, FL



Brad.Schacht@microsoft.com



<https://BradleySchacht.com>



@BradleySchacht



BradleySchacht.bsky.social



in/BradleySchacht



Tales From the Field

# How did we do?

Tell us your thoughts  
about our sessions and  
complete the event survey

