Embark on a transformative journey into building an end-to-end solution within the Microsoft Fabric.



Marc Lelijveld

Technical Evangelist & Architect
At Macaw

Microsoft® Most Valuable Professional

sessionize MOST ACTIVE SPEAKER 2022

sessionize MOST ACTIVE SPEAKER 2023

@MarcLelijveld

linkedin.com/in/MarcLelijveld

Data-Marc.com

DutchFabricUsergroup.com



Erwin de Kreuk

Principal Consultant & Lead Data & AI
At InSpark

Microsoft® Most Valuable Professional

sessionize MOST ACTIVE SPEAKER 2022

sessionize MOST ACTIVE SPEAKER 2023

@ErwinDeKreuk

linkedin.com/in/erwindekreuk

Erwindekreuk.com

DutchFabricUsergroup.com



Thank you sponsors!















Rate Data Saturday Holland

SUPPORT CASPER

1 review = 1 €

Towards beating pancreatic cancer

Objectives



- Data platform challenges
- Onelake
- Medallion Architecture
- Parameters
- Meta Data Framework
- Semantic Model & DirectLake
- CI / CD
- Copilot (all the things)

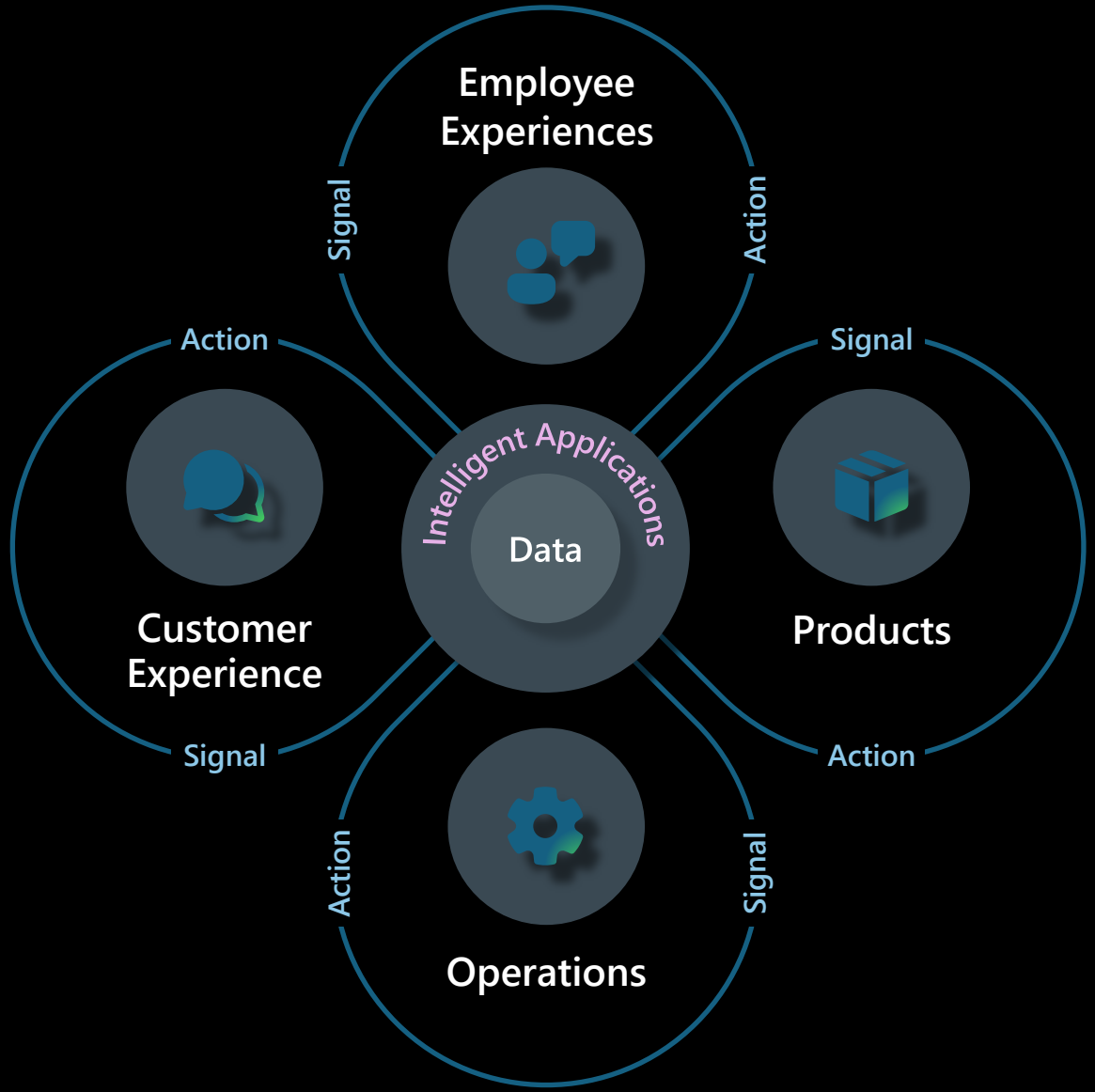
1

Growing importance of Data in the era of AI

Data is the oxygen of digital transformation

"A new kind of company — we call them insights driven businesses — has formed. They are growing at an average of more than 30% annually"

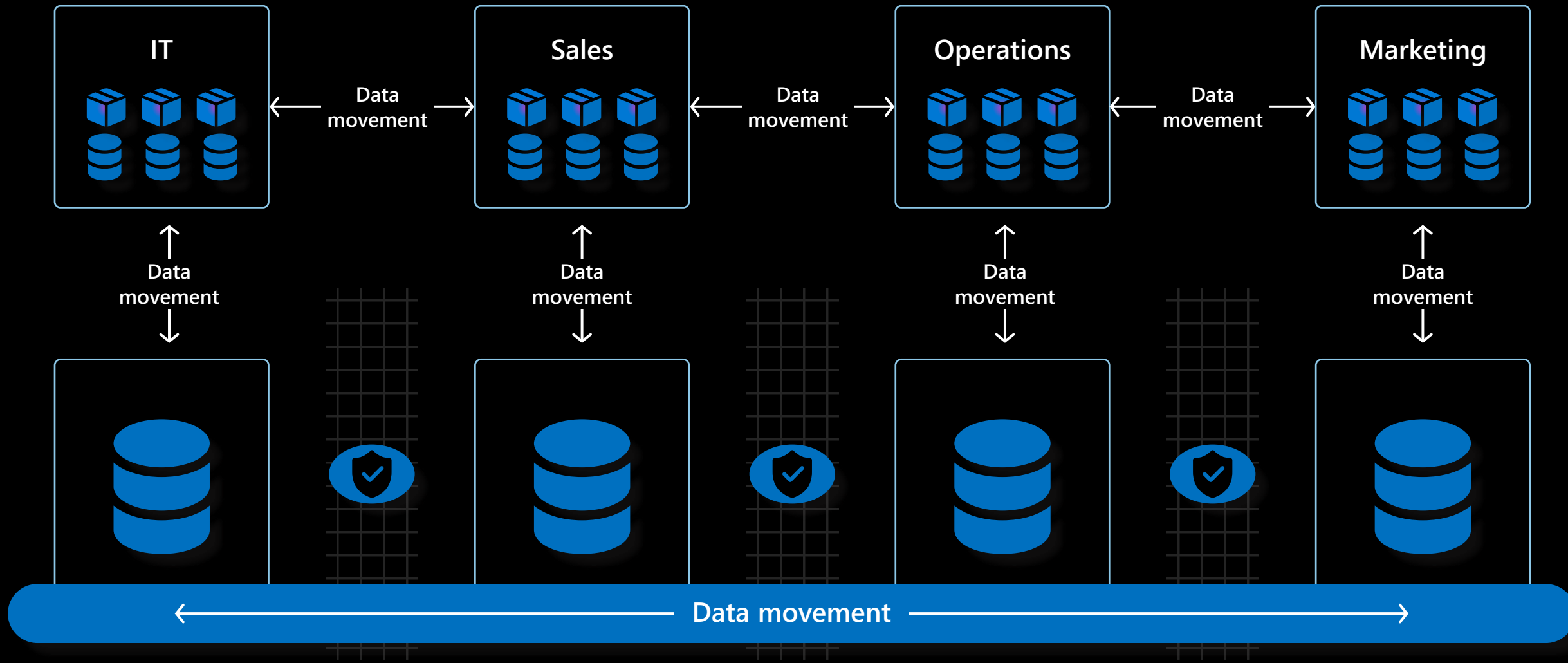
Forrester Analytics Business Technographics Global Data & Analytics Survey



Today’s data value creation challenges

- Siloed systems and data
- Fragmented toolset, Platforms requiring advanced skills
- Costly integration and ongoing maintenance
- Proprietary Platform, Vendor Lock in
- Infrastructure not scalable as business grows
- Gaining BI adoption to streamline data sharing, collaboration
- Data and security risks, Governance & Compliance

Starting line: Multiple siloed lakes with lots of duplication



Today's data and analytics challenges



Balancing the need for data access and self-service analytics while remaining governed

Breaking down data siloes across the organization into a unified source of truth

Limited scalability of legacy solutions as data demand rises exponentially

Delivering on the promise of analytics with limited resources



Microsoft Fabric

Data analytics for the era of AI

From		To
Isolated component	→	Unified stack
Single database	→	All the data
Gen AI Bolted-in	→	Gen AI built in

Microsoft Fabric

Data analytics for the era of AI

Complete Analytics Platform

Lake Centric and Open

Empower Every Business User

AI Powered



Microsoft Fabric

The data platform for the era of AI

Complete Analytics Platform

- Everything, unified
- SaaS-ified
- Secured and governed

Lake centric and open

- OneLake
- One Copy
- Open at every tier

Empower Every Business User

- Familiar and intuitive
- Built into Microsoft 365
- Insight to action

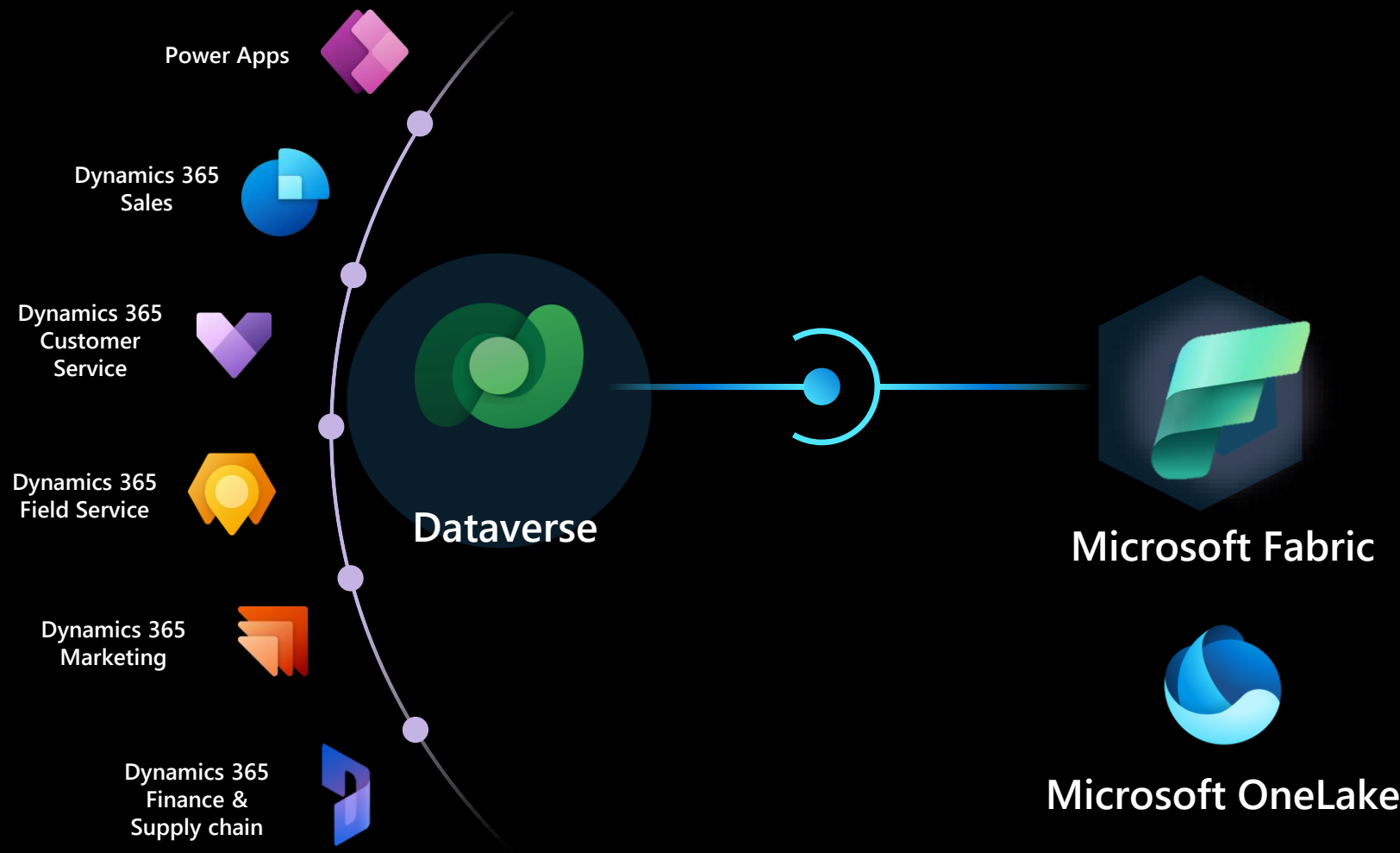
AI Powered

- Copilot accelerated
- ChatGPT on your data
- AI driven insights

The value in your ERP data



Bringing your data to Fabric





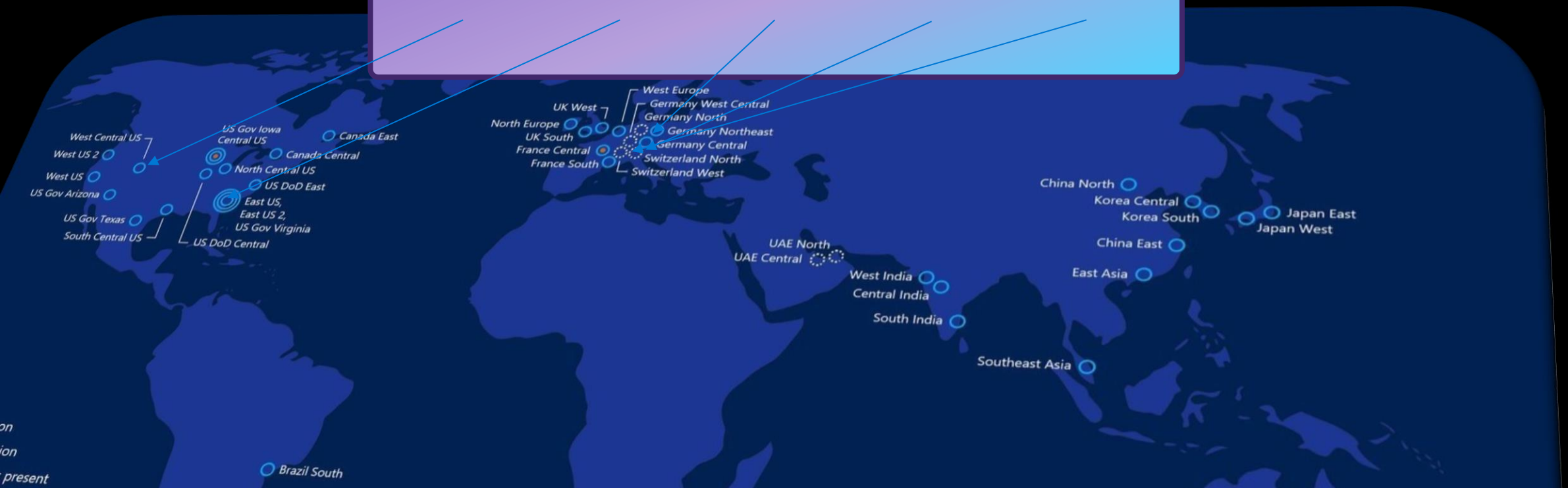
Any questions so far?

Onelake

OneLake which logically spans the world

"The OneDrive for Data"

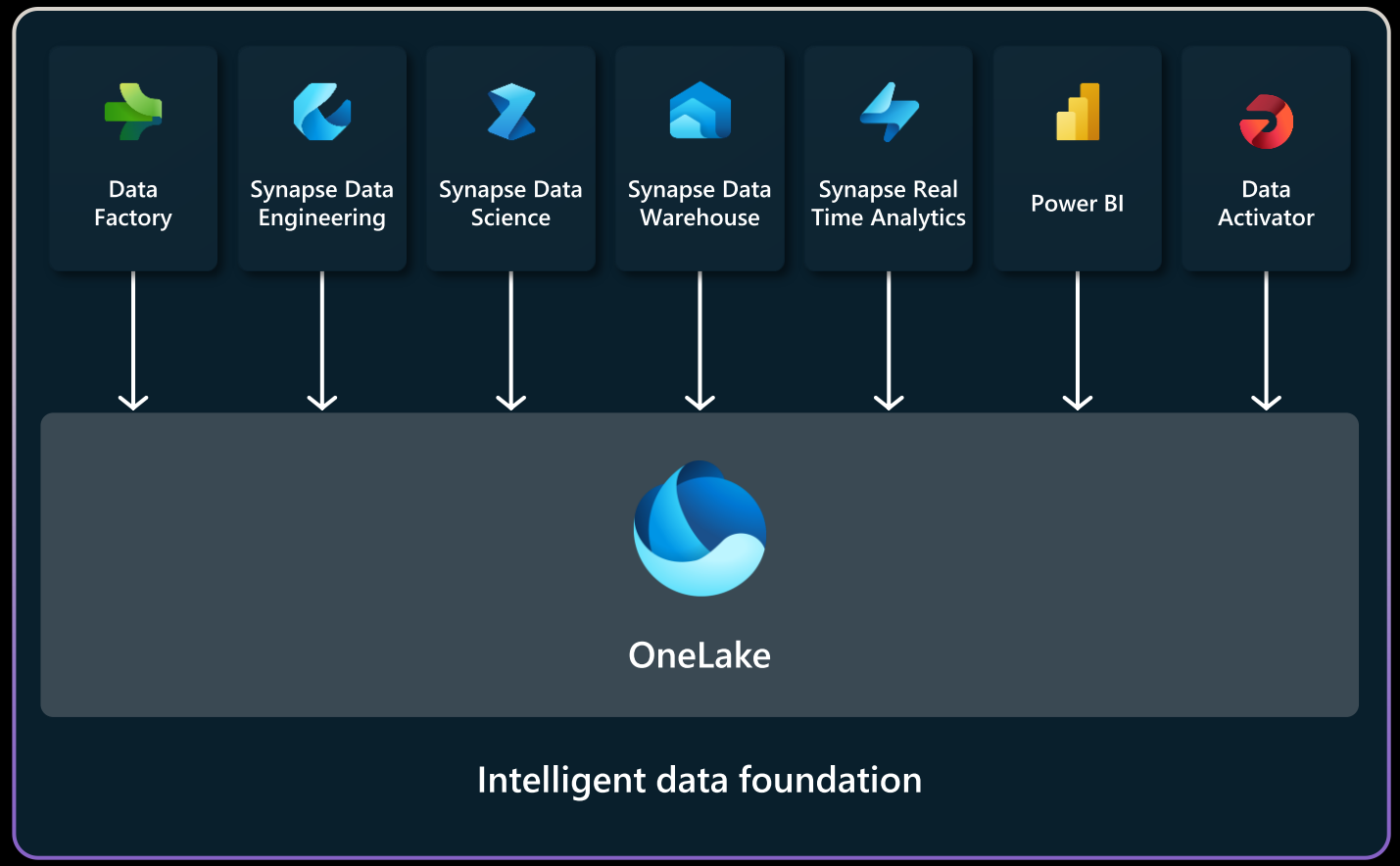




Workspaces can reside in different regions around the world while still being part of the same data lake.

OneLake for all Data

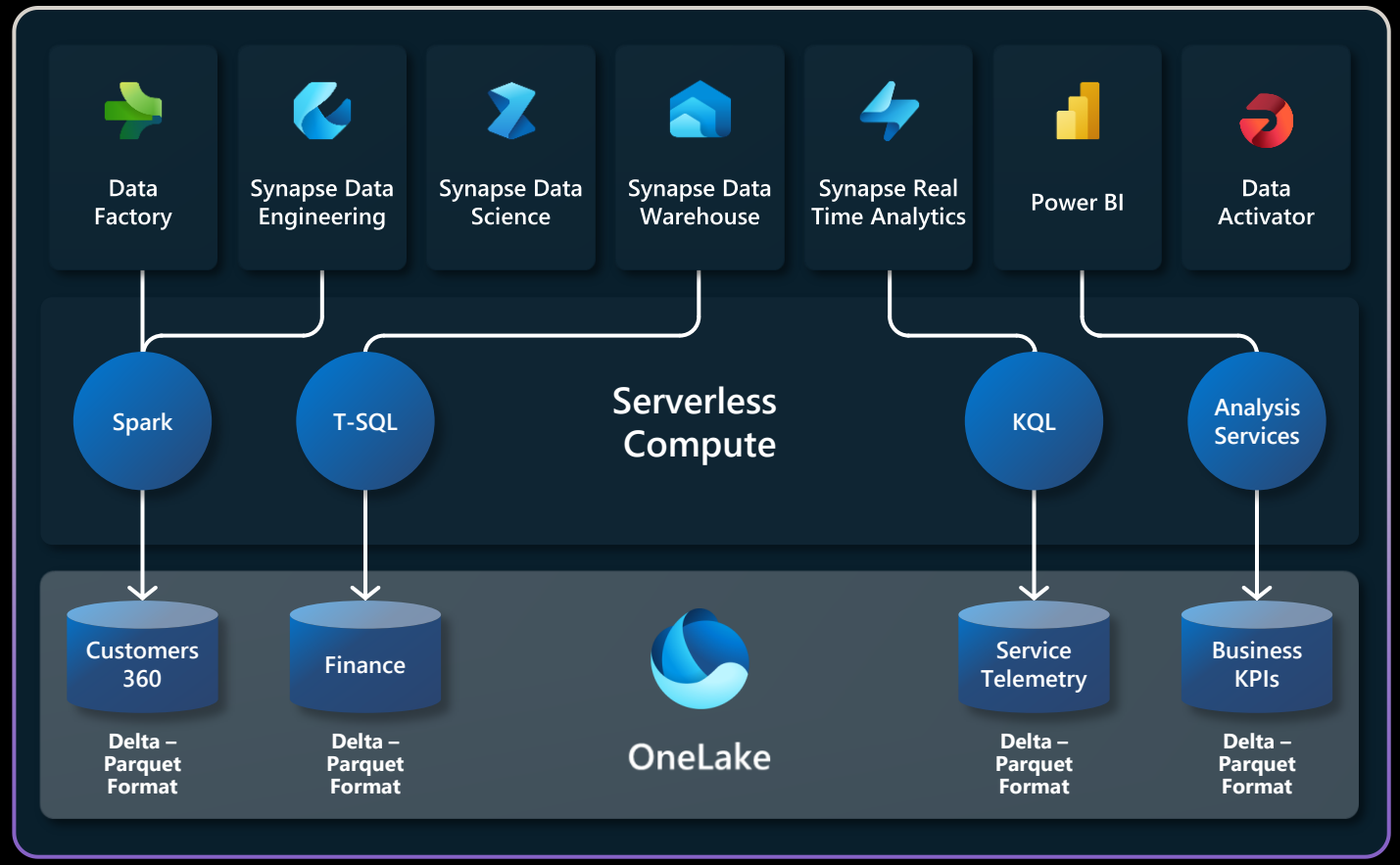
"The OneDrive for Data"



- A single SaaS lake for the whole organization
- Provisioned automatically with the tenant
- All workloads automatically store their data in the OneLake workspace folders
- All the data is organized in an intuitive hierarchical namespace
- The data in OneLake is automatically indexed for discovery, MIP labels, lineage, PII scans, sharing, governance and compliance

One Copy for all computes

Real separation of compute and storage



All the compute engines store their data automatically in OneLake

The data is stored in a single common format

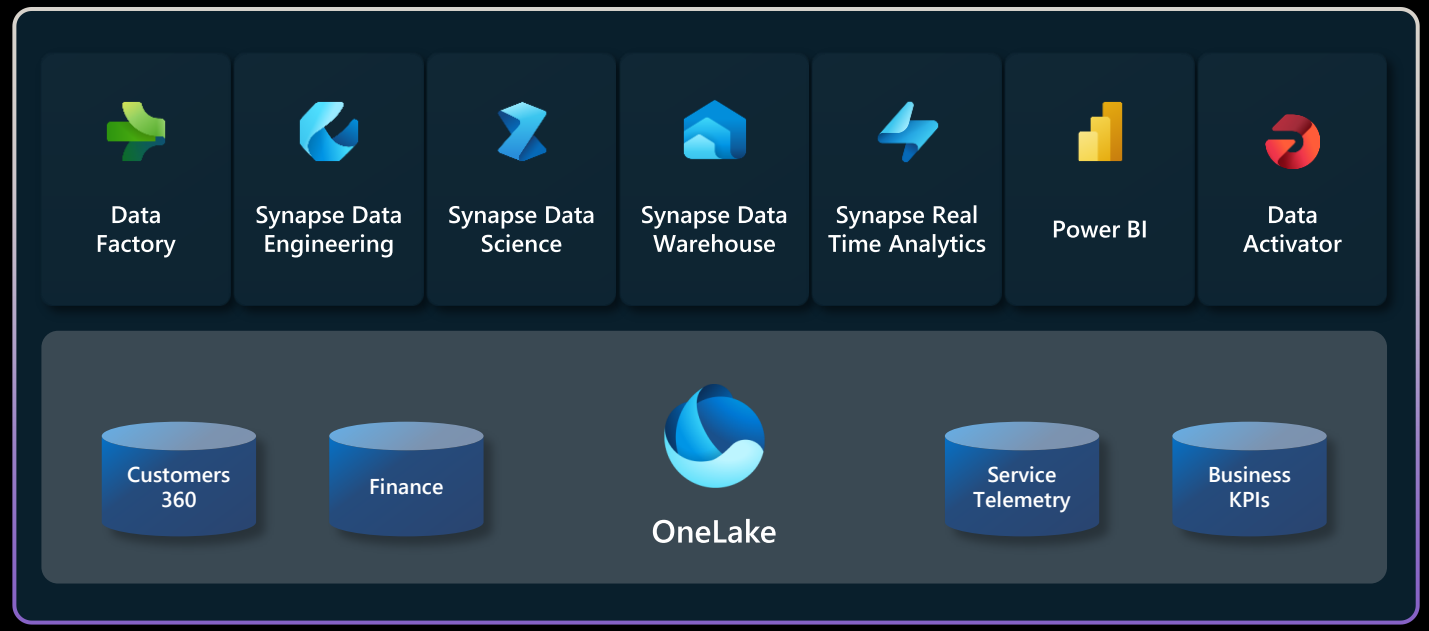
Delta – Parquet, an open standards format, is the storage format for all tabular data in Analytics vNext

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

Unifying data in OneLake

Shortcuts



Sharing data in OneLake is as easy as sharing files in OneDrive, removing the needs for data duplication

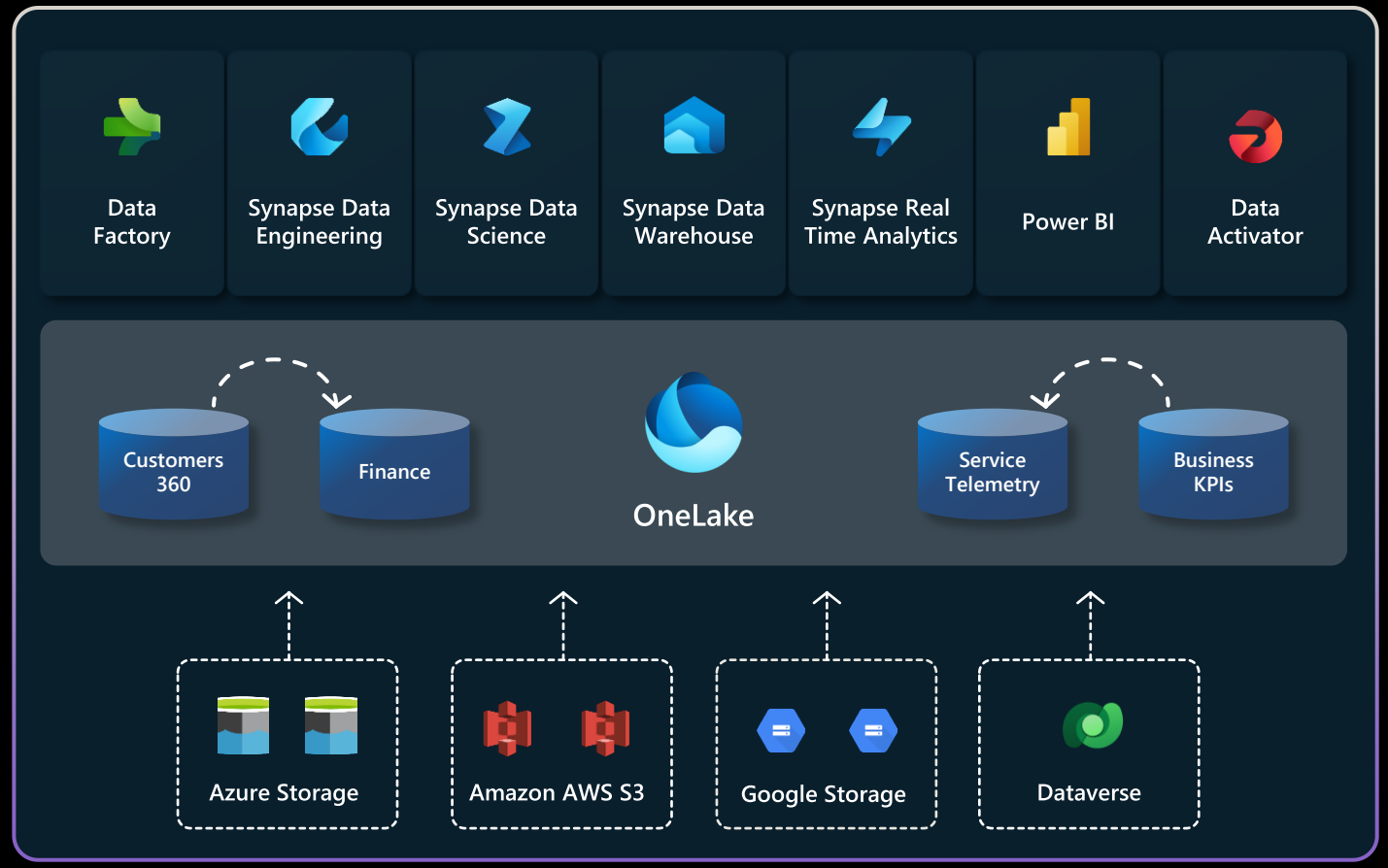
With shortcuts, data throughout OneLake can be composed together without any data movement

Shortcuts also allow instant linking of data already existing in Azure and in other clouds, without any data duplication and movement, making OneLake the first multi-cloud data lake

With support for industry standard APIs, OneLake data can be directly accessed by any application or service

Unifying data in OneLake

Shortcuts



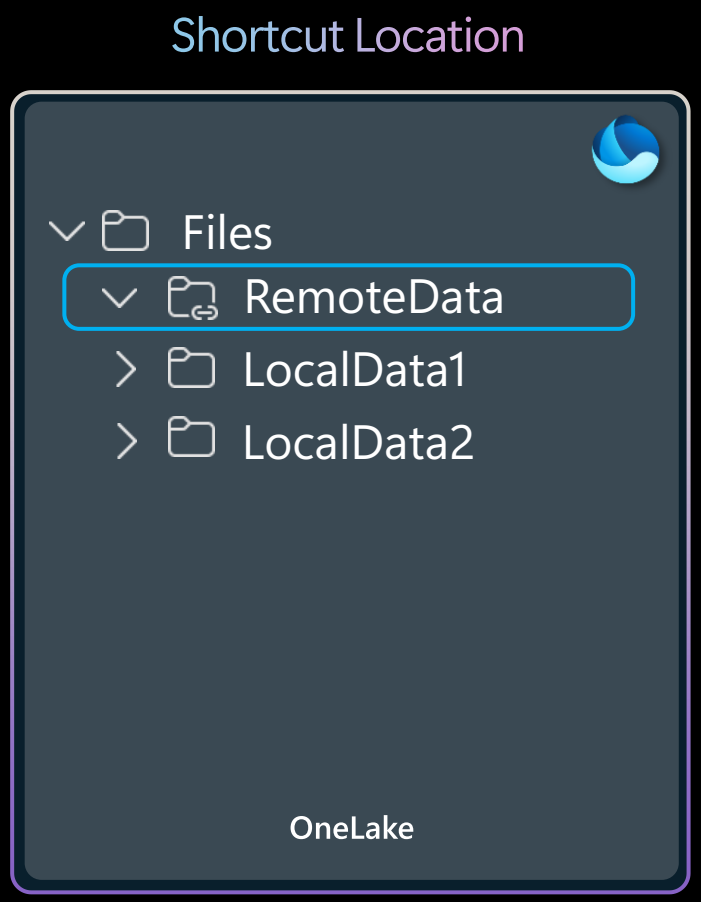
Sharing data in OneLake is as easy as sharing files in OneDrive, removing the needs for data duplication

With shortcuts, data throughout OneLake can be composed together without any data movement

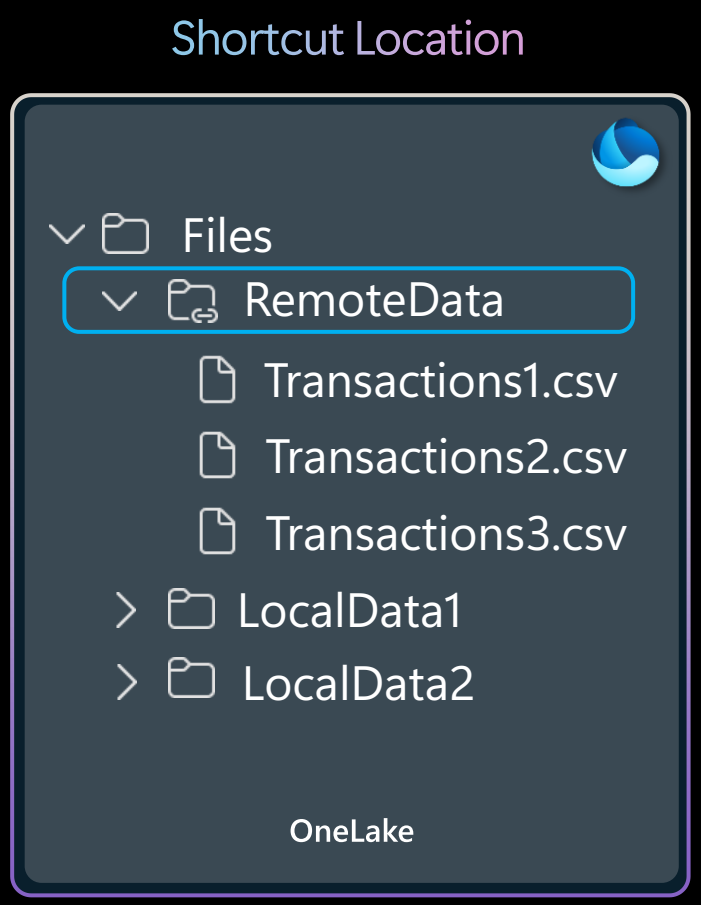
Shortcuts also allow instant linking of data already existing in Azure and in other clouds, without any data duplication and movement, making OneLake the first multi-cloud data lake

With support for industry standard APIs, OneLake data can be directly accessed by any application or service

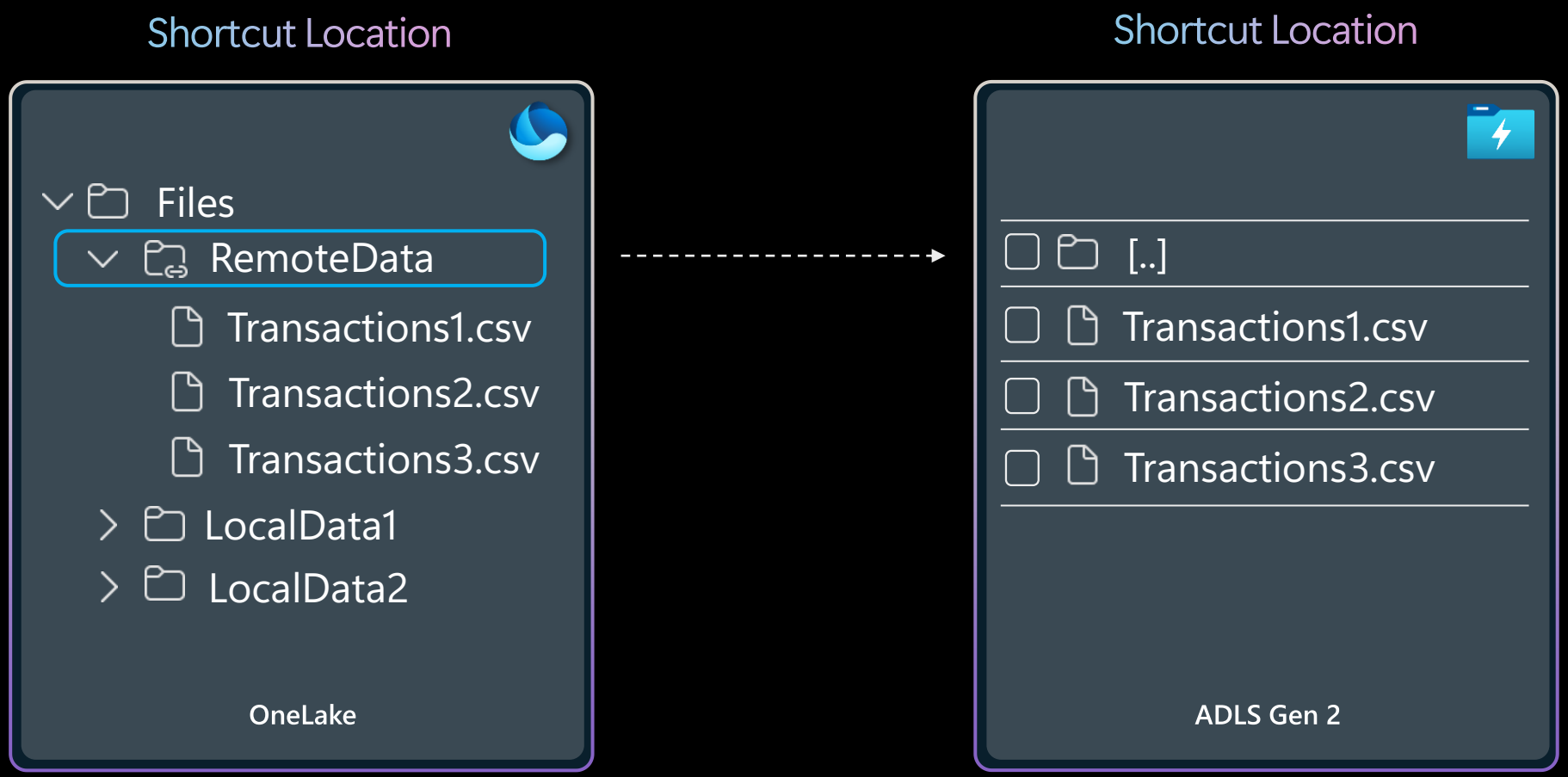
What is a shortcut?



What is a shortcut?



What is a shortcut?

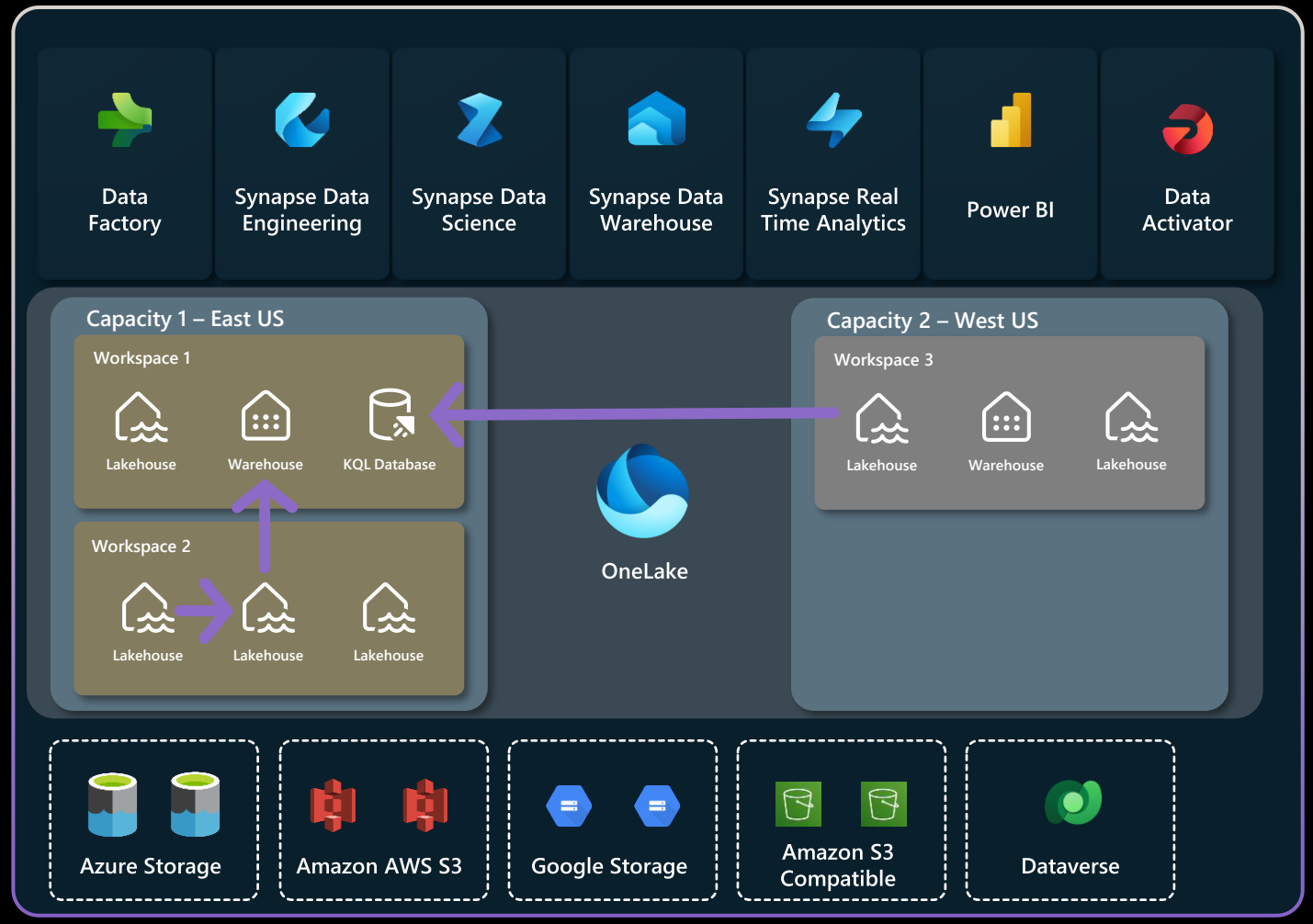




Any questions so far?

Unifying data in OneLake

Shortcuts



Sharing data in OneLake is as easy as sharing files in OneDrive, removing the needs for data duplication

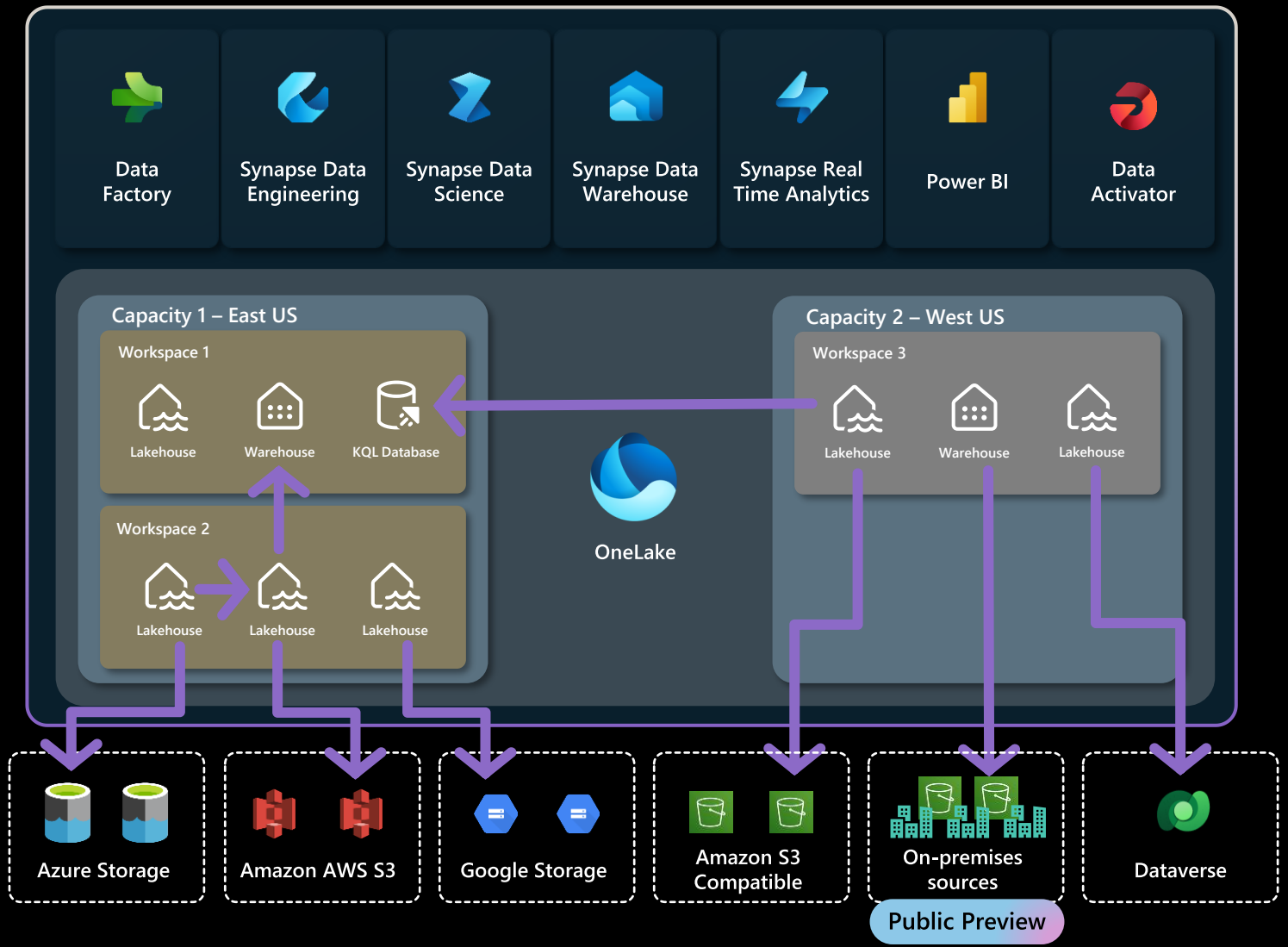
With shortcuts, data throughout OneLake can be composed together without any data movement

Shortcuts also allow instant linking of data already existing in Azure and in other clouds, without any data duplication and movement, making OneLake the first multi-cloud data lake

With support for industry standard APIs, OneLake data can be directly accessed by any application or service

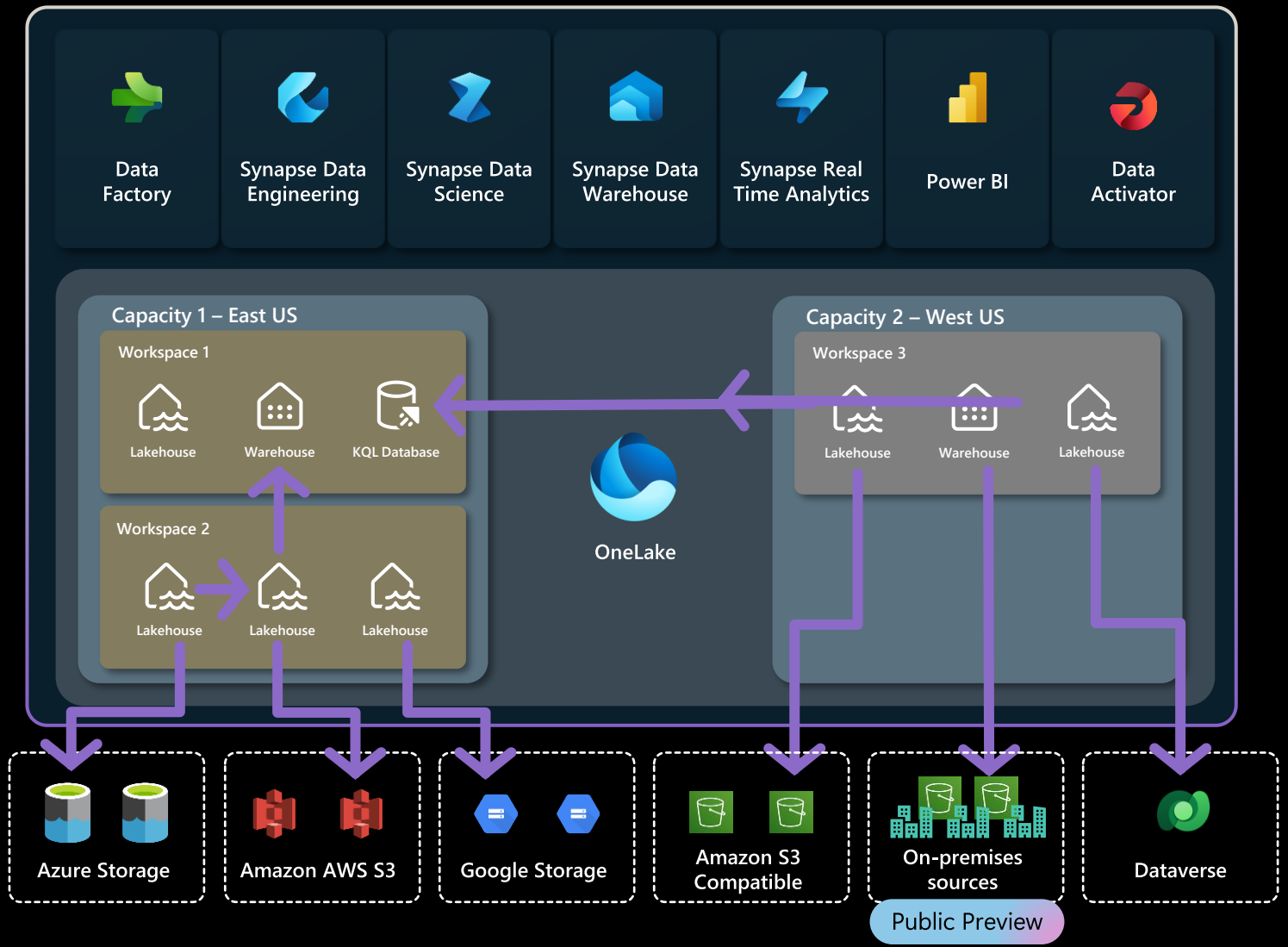
Unifying data in OneLake

Shortcuts



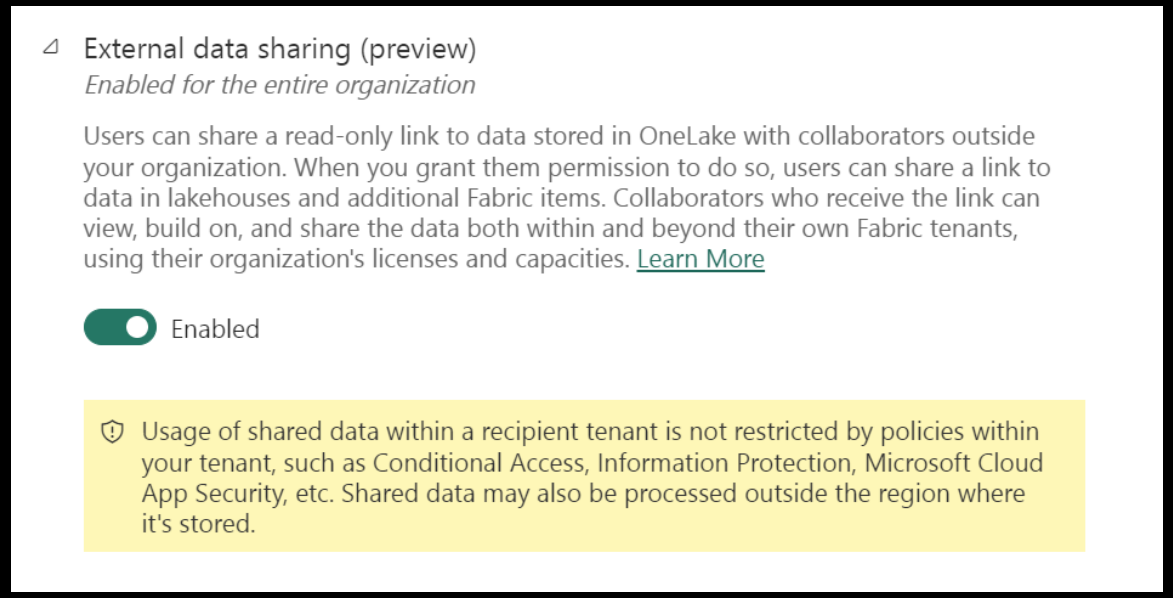
Unifying data in OneLake

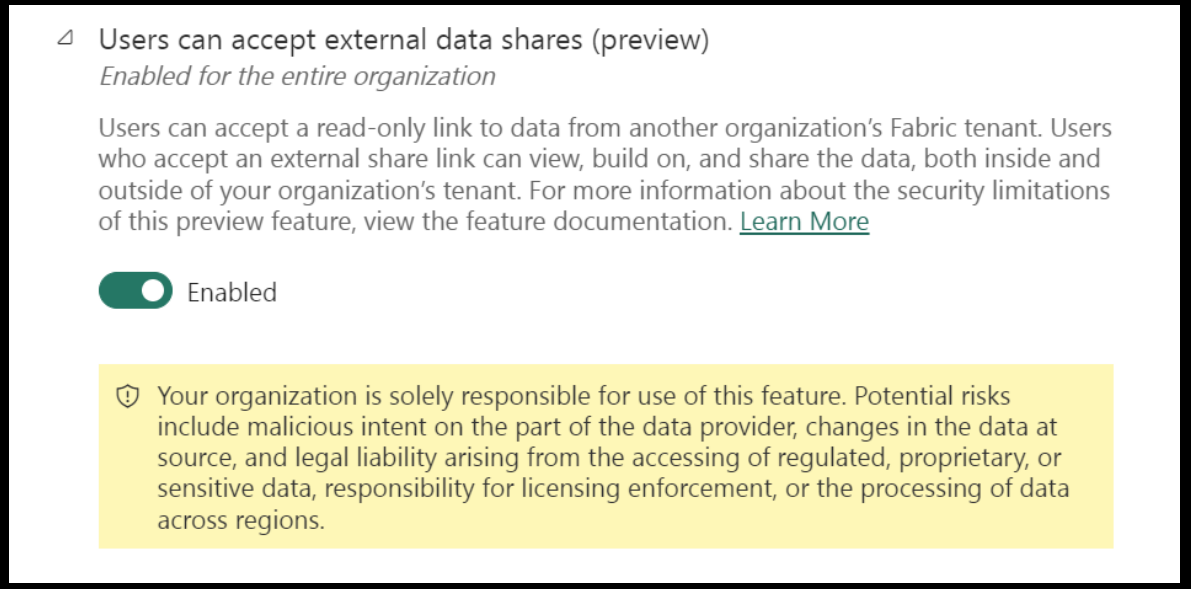
External Data Sharing (Preview)



Unifying data in OneLake

External Data Sharing (Preview)





Mirroring in Fabric

Connect your databases and data warehouses to OneLake seamlessly with Zero ETL (Preview)



Azure Cosmos DB



Mongo DB



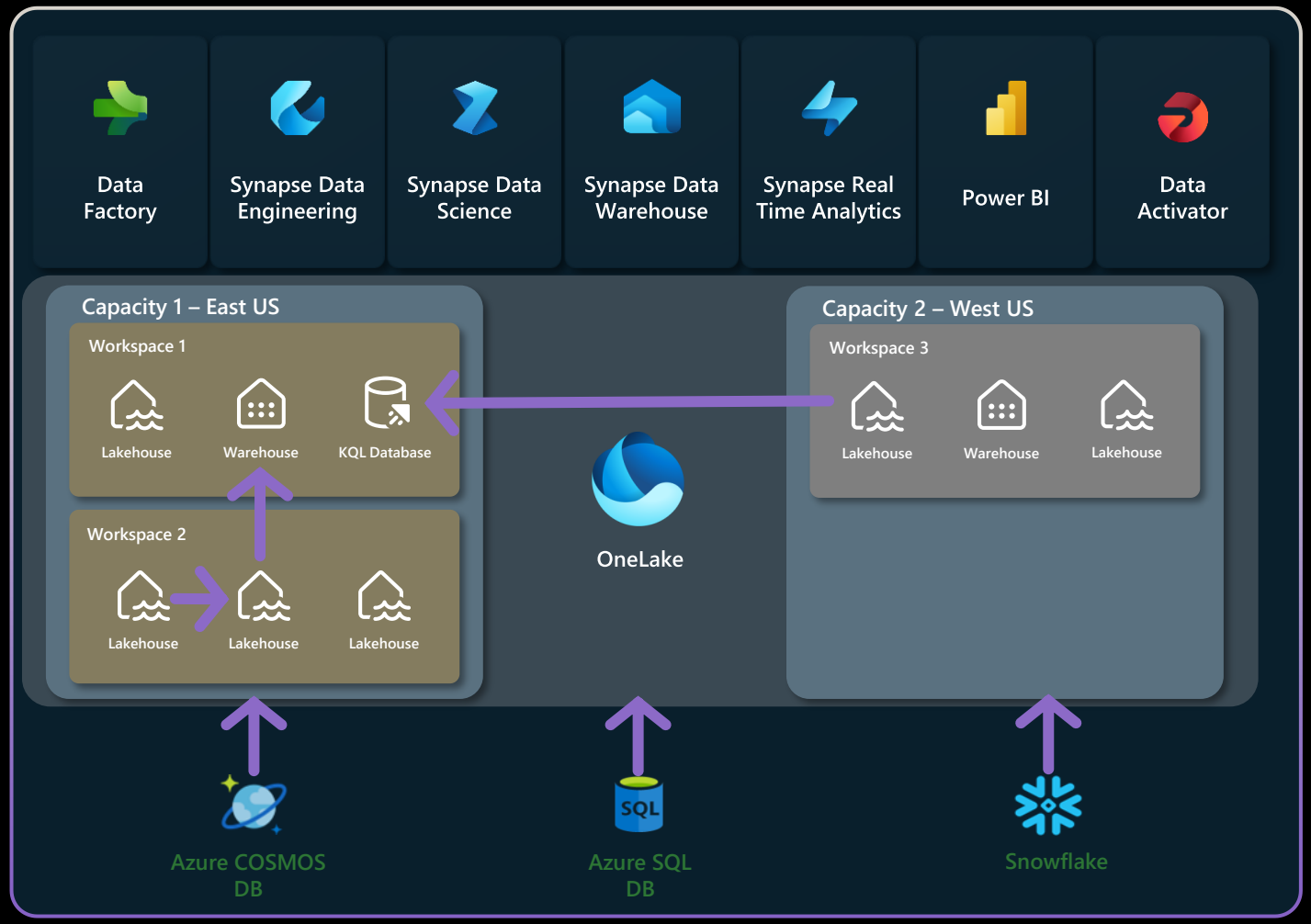
Azure SQL DB



Snowflake

Mirroring in Fabric

Connect your databases and data warehouses to OneLake seamlessly with Zero ETL (Preview)



Fabric Mirroring allows adding existing databases and data warehouses to Fabric without any ETL with a new item type Mirrored database

A full live querying experience of the source database is available for the Mirrored database

Data is replicated into OneLake in Delta format and kept up-to-date in near-real-time via change data feed technology

All the Fabric experiences instantly work with the OneLake replica, including data warehousing, Spark, BI experiences

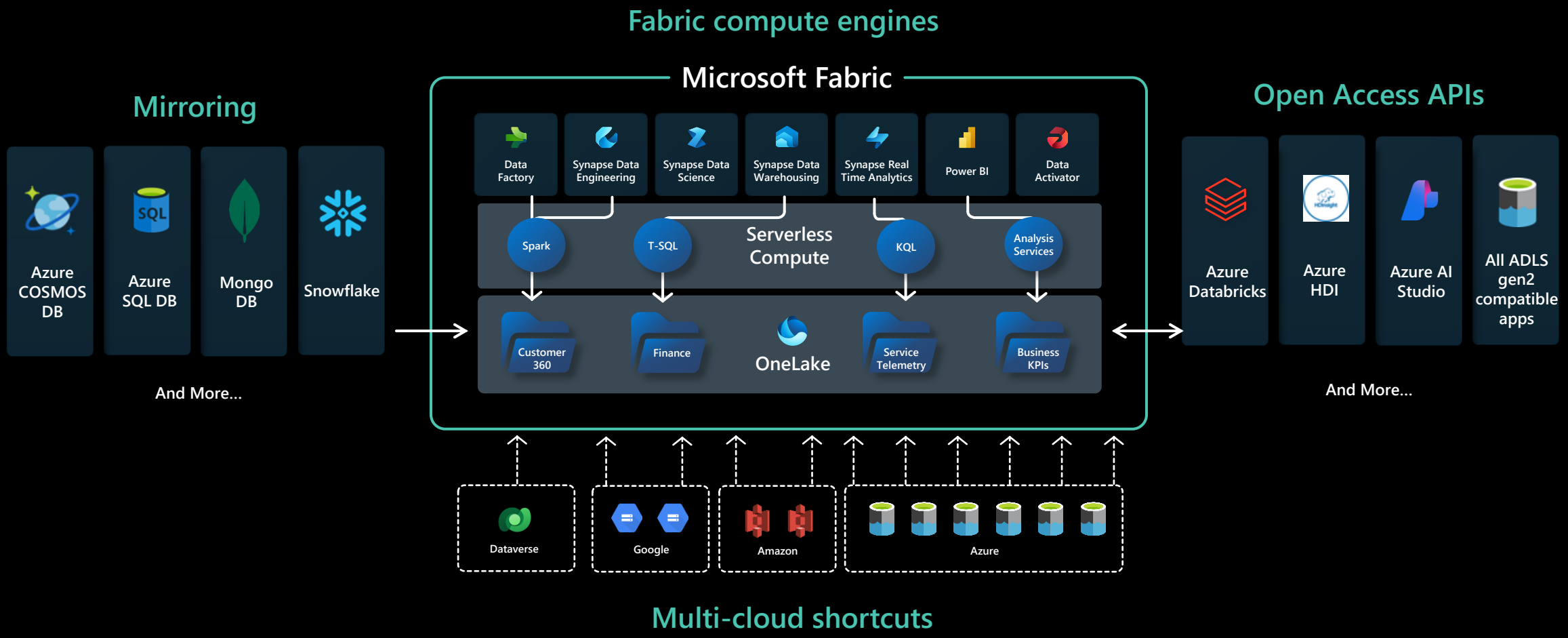
Analysts and Data Scientists can work with real-time data

The replica protects operational databases from analytical queries

But what if you’re not fully in the Microsoft ecosystem?

All roads lead to OneLake

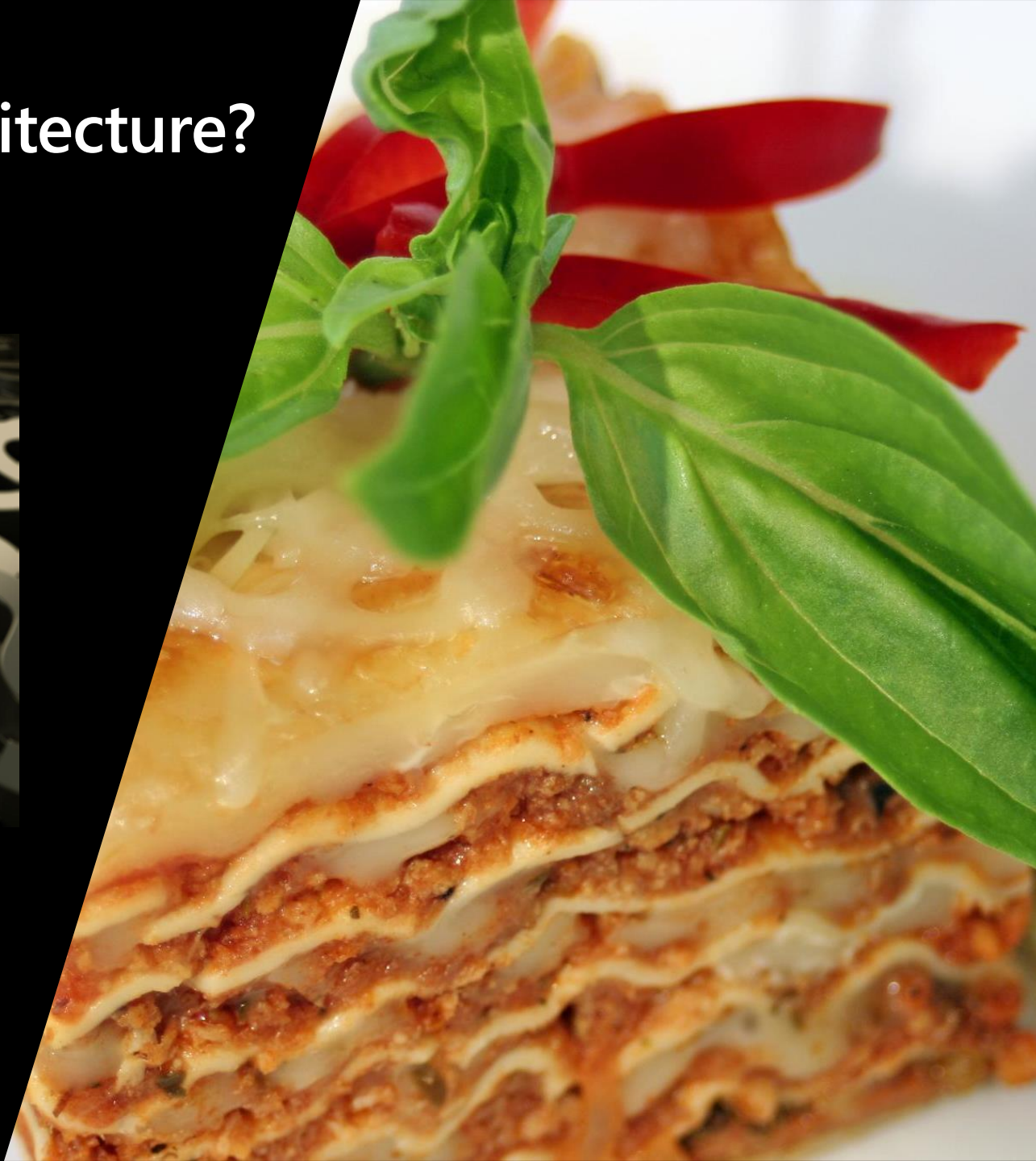
Creating Data Gravity in OneLake



Who is using a Medallion architecture?



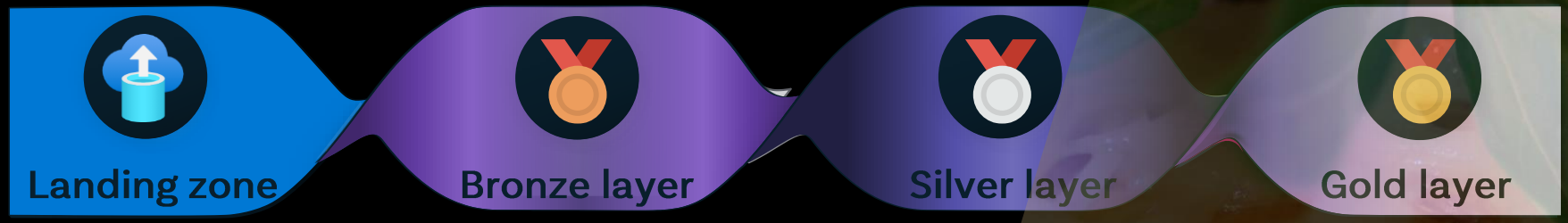
'Uniform data architecture'
From data "Spaghetti to Lasagna"



Medallion Architecture

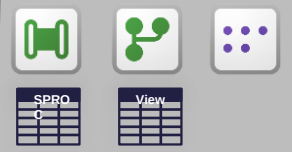
'Data processing in different stages'

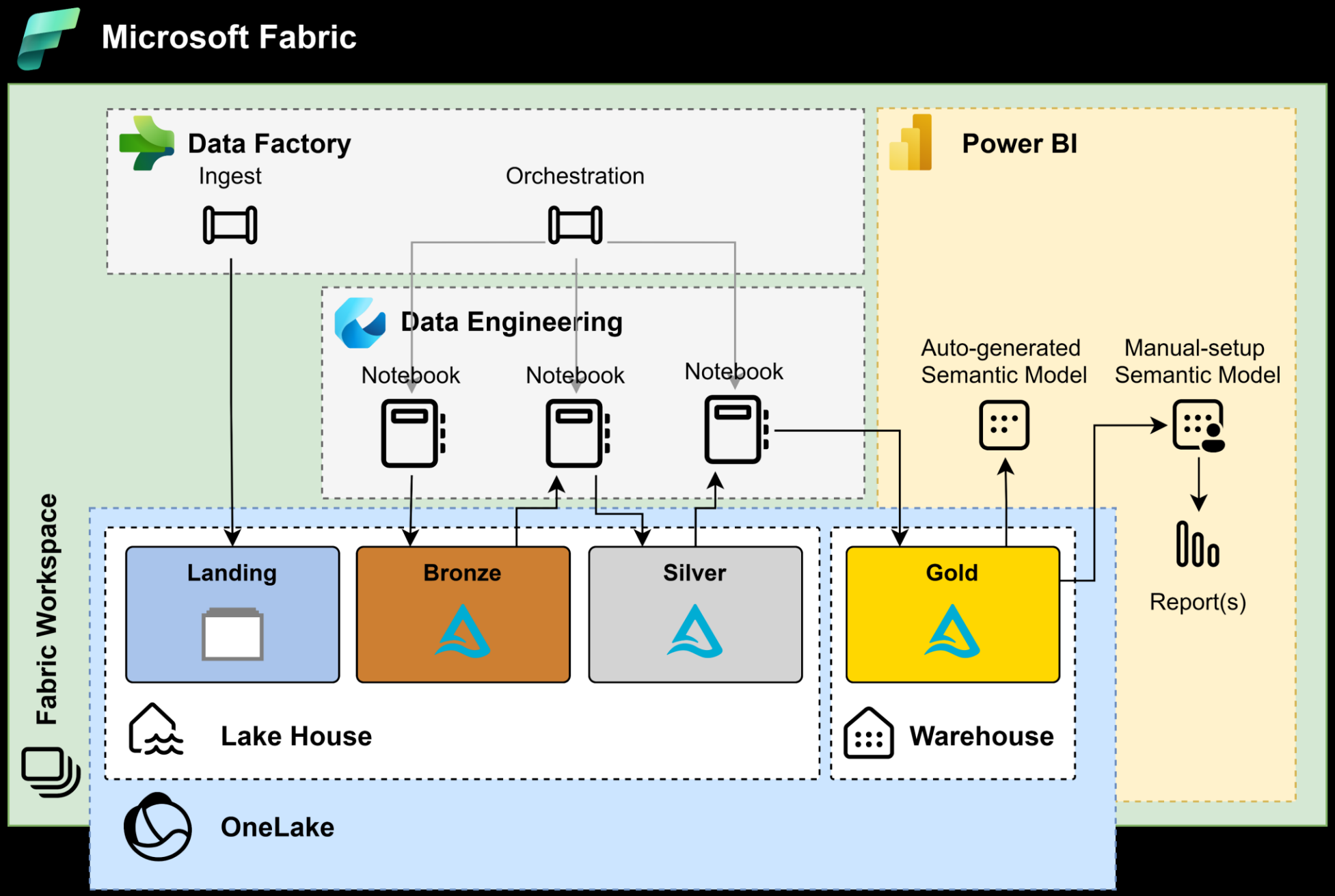
Stages



Medallion Architecture

'Data processing in different stages'

Stage:	Definition:	Filetype:	Files/Tables:	Fabric:
Gold layer	• Dimensions & Facts (Star Schema) • Historical Analysis • Business rules • Documentation • Aggregated data • Logical table names			
Silver layer	• Historical Data (Type 1 or 2) • Data quality rules • Data Cleansing • Validated data • No business model/data			
Bronze layer	• Deduplicate data • Add datatypes • Data can be inconsistent • Mostly a copy of the source • Schema			
Landing zone	• Structured data • Unstructured data • Incremental loads • Data as is • Stored in Datetime folder structure • No Schema			


Microsoft Fabric
Data Factory
Ingest
Orchestration
Power BI
Data Engineering
Notebook
Notebook
Notebook
Auto-generated
Semantic Model
Manual-setup
Semantic Model
Fabric Workspace
Landing
Bronze
Silver
Gold
Report(s)
Lake House
Warehouse
OneLake

Demo



Who is already using Parameters?



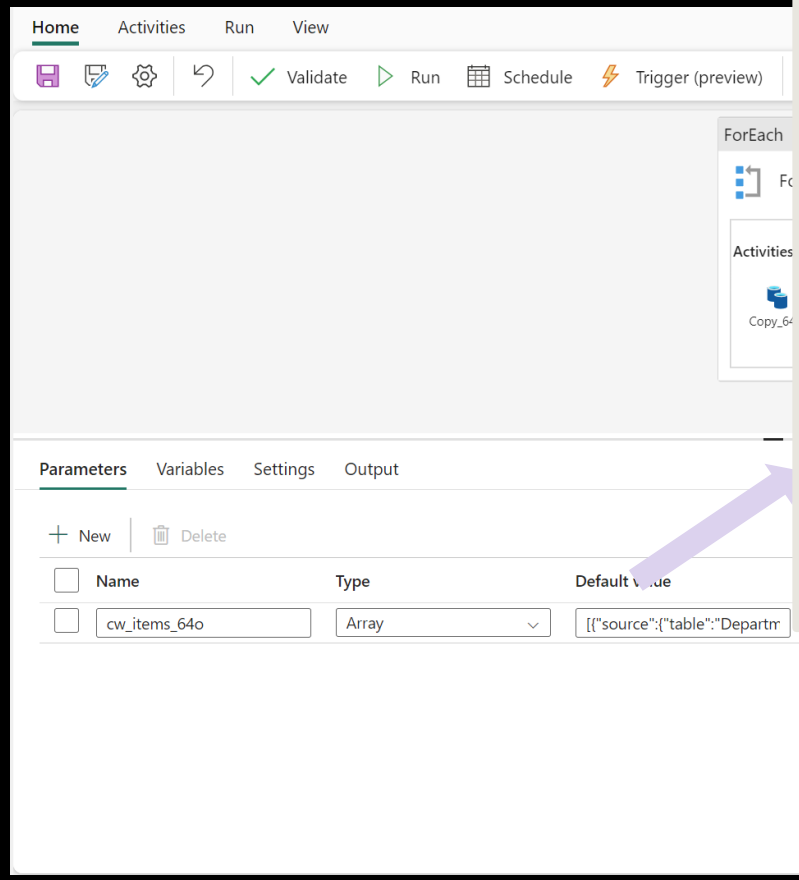
Parameters

- Pipeline Parameters
  - Copy Activity Parameters
  - Foreach Parameters
- Notebook Parameters



Understand Parameters

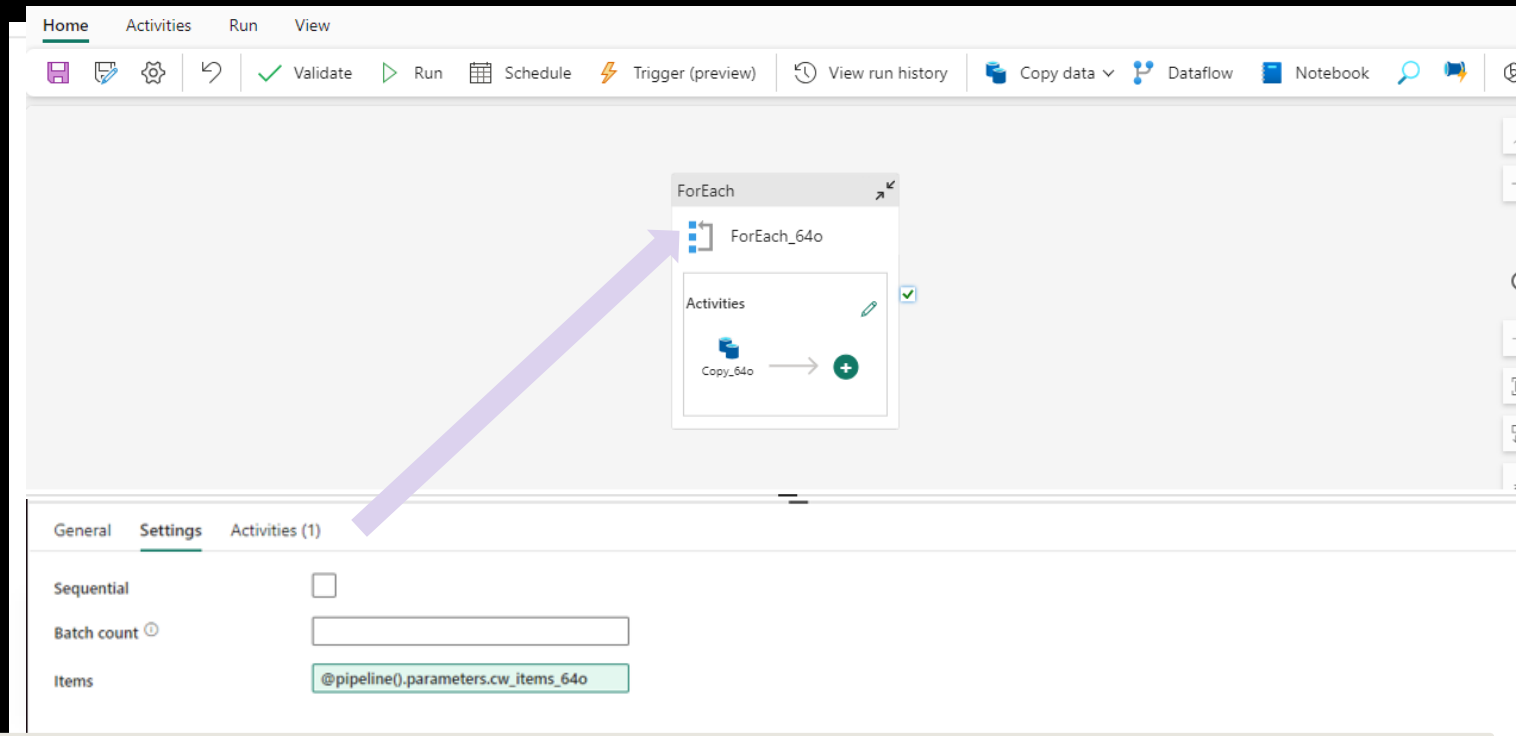
- Copy data assistant
  - Select Data Source
  - Select Tables
  - Select Destination
  - Select Filetypes
  - Review actions




```
{
    "source": {
        "table": "Department"
    },
    "destination": {
        "fileName": "HumanResourcesDepartment.
    }
},
{
    "source": {
        "table": "Employee"
    },
    "destination": {
        "fileName": "HumanResourcesEmployee.pa
    }
},
{
    "source": {
        "table": "EmployeeDepartmentHistory"
    },
    "destination": {
        "fileName": "HumanResourcesEmployeeDep
    }
},
{
    "source": {
        "table": "EmployeePayHistory"
    },
    "destination": {
        "fileName": "HumanResourcesEmployeePay
    }
```

Pipeline Parameters

- Pass through from
  - Pipeline to Activity

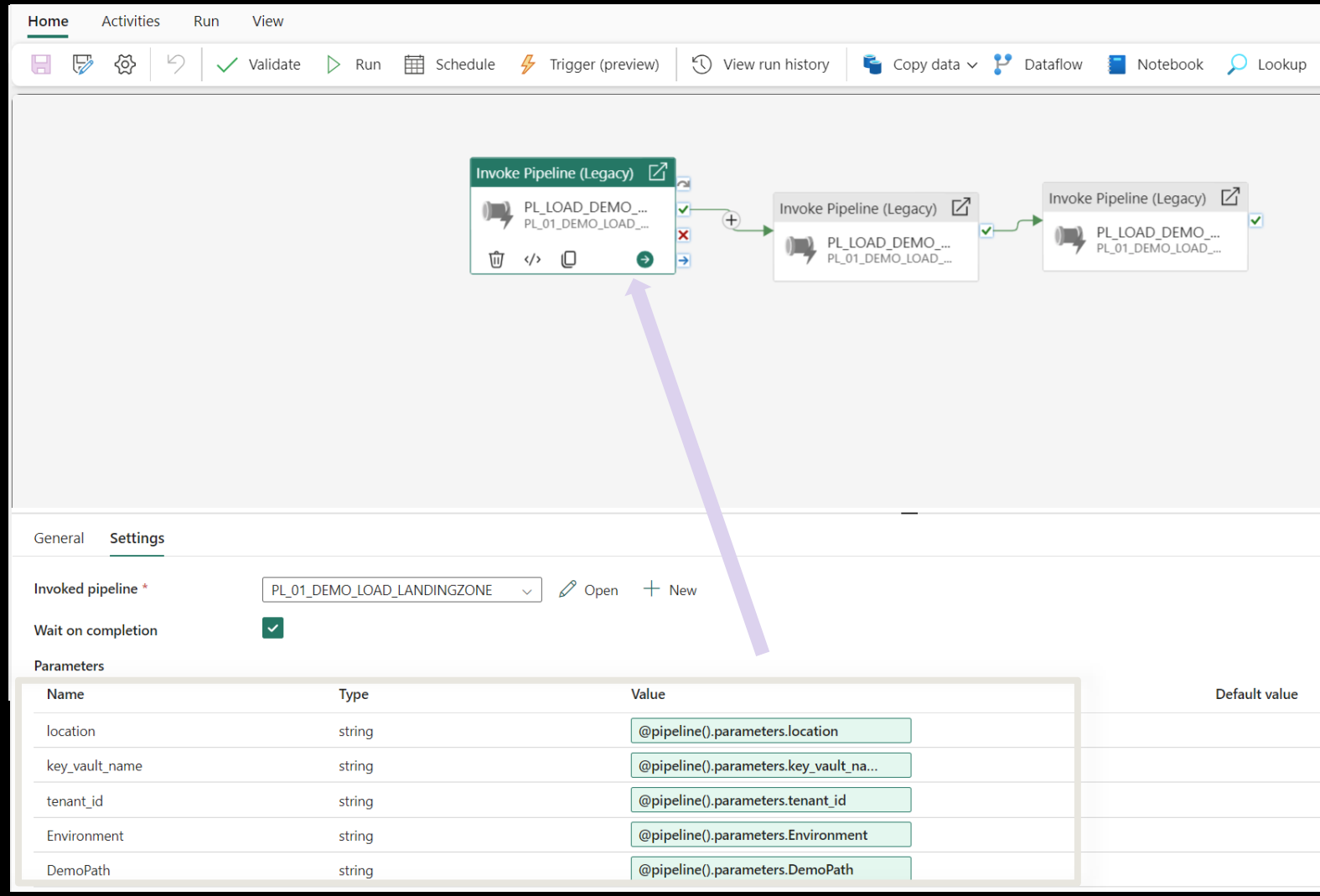


Items

@pipeline().parameters.cw_items_64o

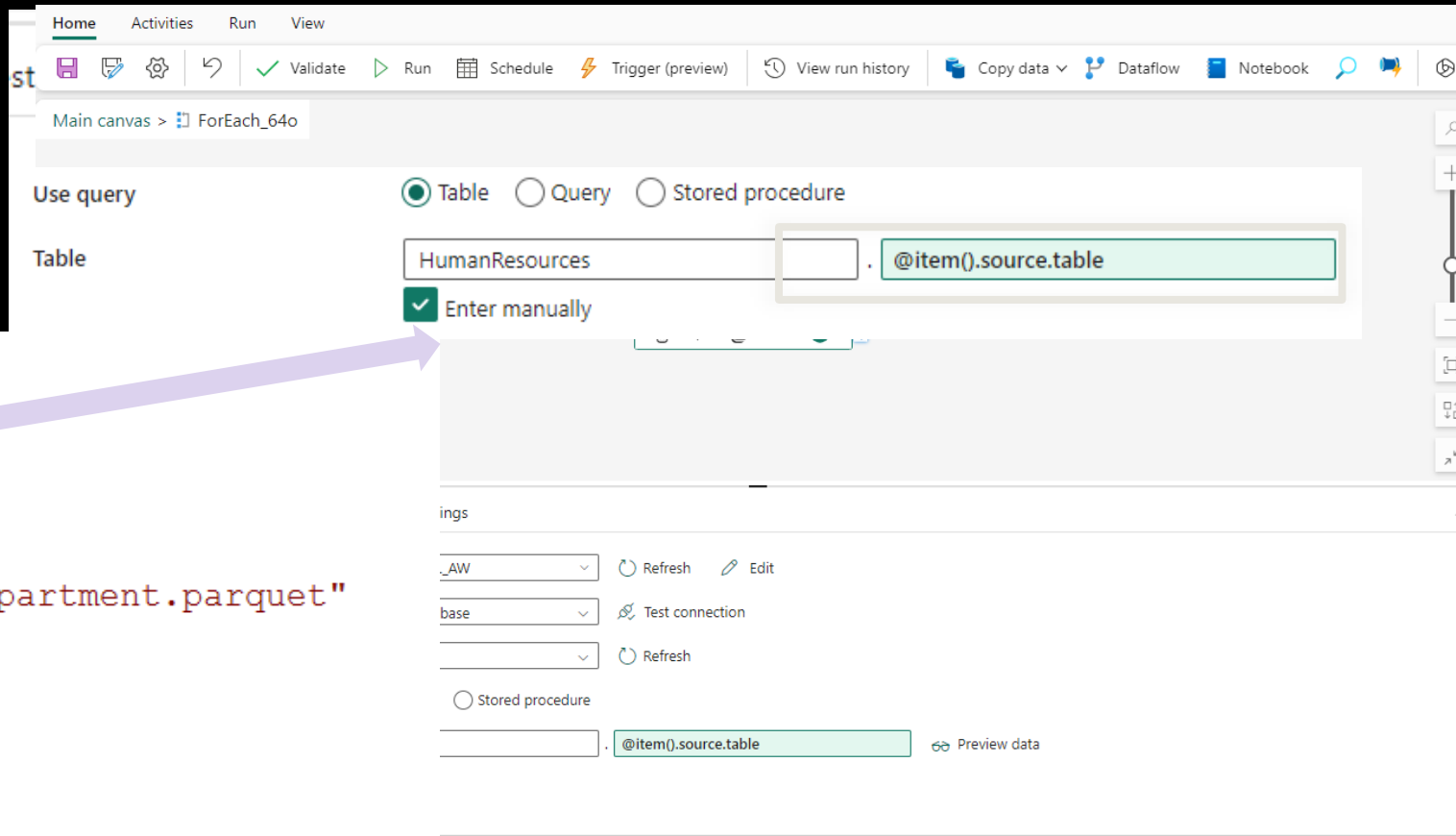
Pipeline Parameters

- Pass through from
  - Pipeline to Pipeline



Copy Activity Parameters

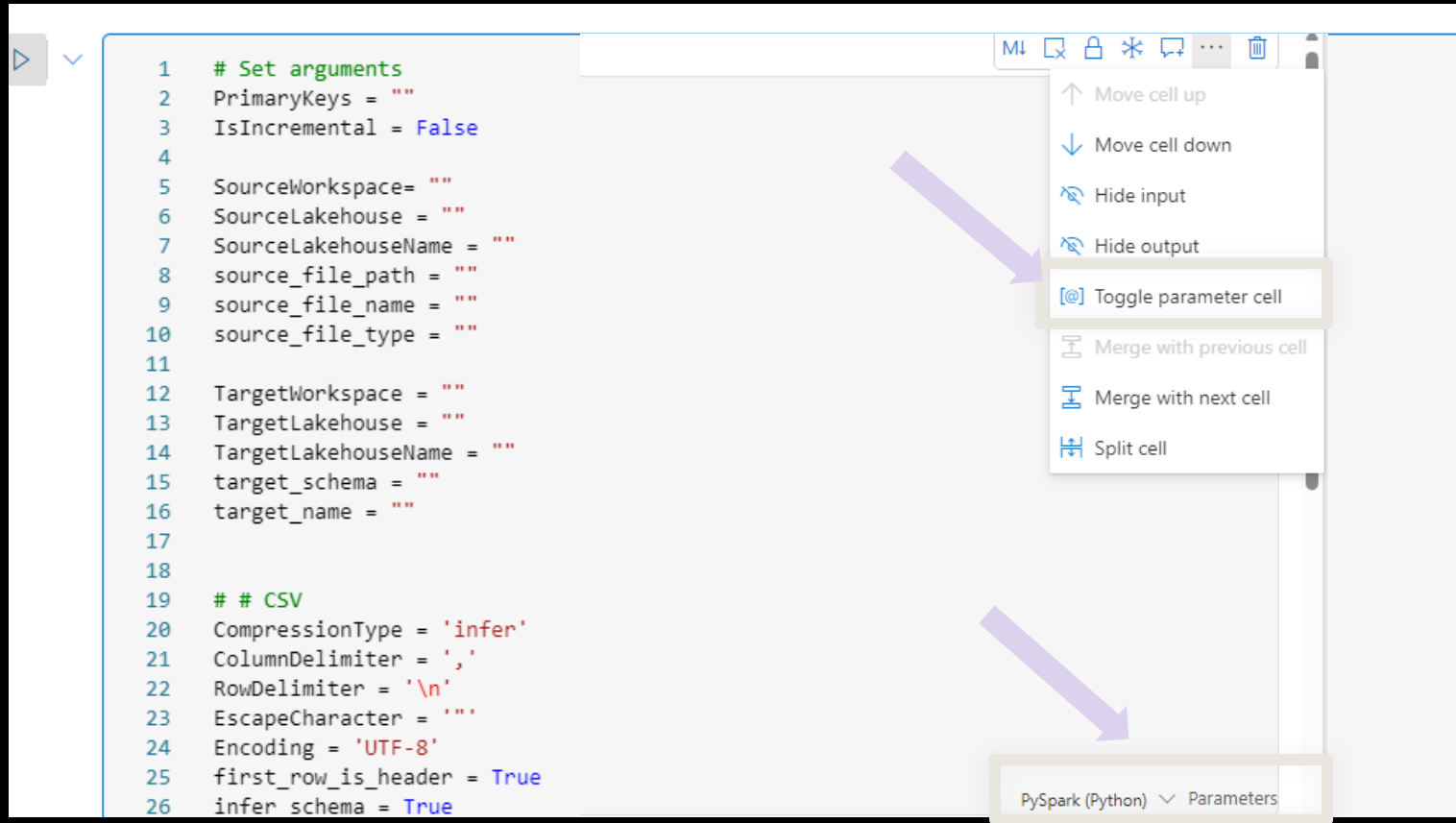
- Pass Parameters from Pipeline to Copy activity
- Use Parameters from For Each Activity




```
[
    {
        "source": {
            "table": "Department"
        },
        "destination": {
            "fileName": "HumanResourcesDepartment.parquet"
        }
    },
    {
        "source": {
            "table": "Employee"
        },
        "destination": {
            "fileName": "HumanResourcesEmployee.parquet"
        }
    },
```

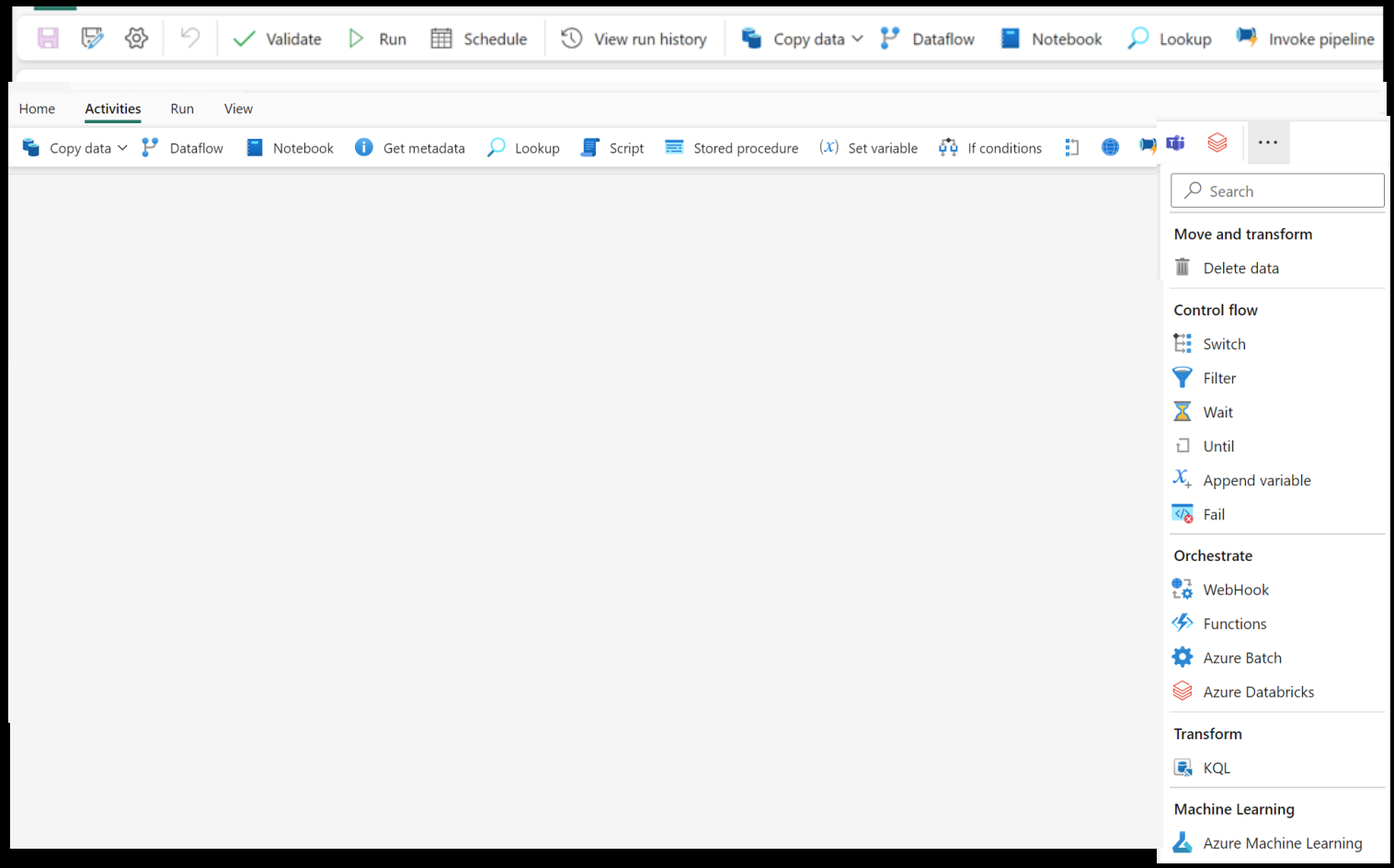
Notebook Parameters

- Pass Parameters from Data Pipeline to Notebook Activity
  - Toggle parameter cell



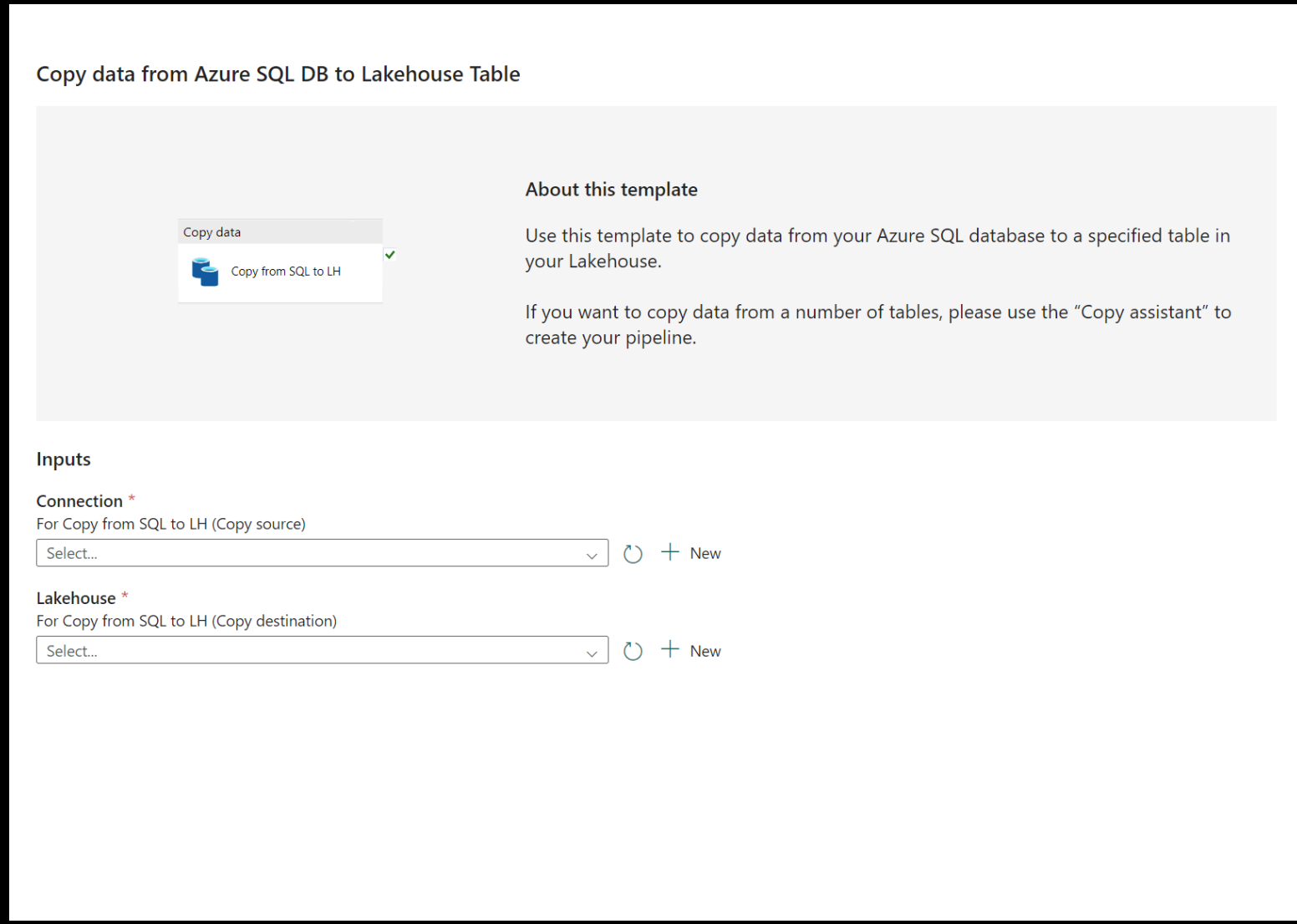
Blank canvas

- Everything build from scratch with no predefined options



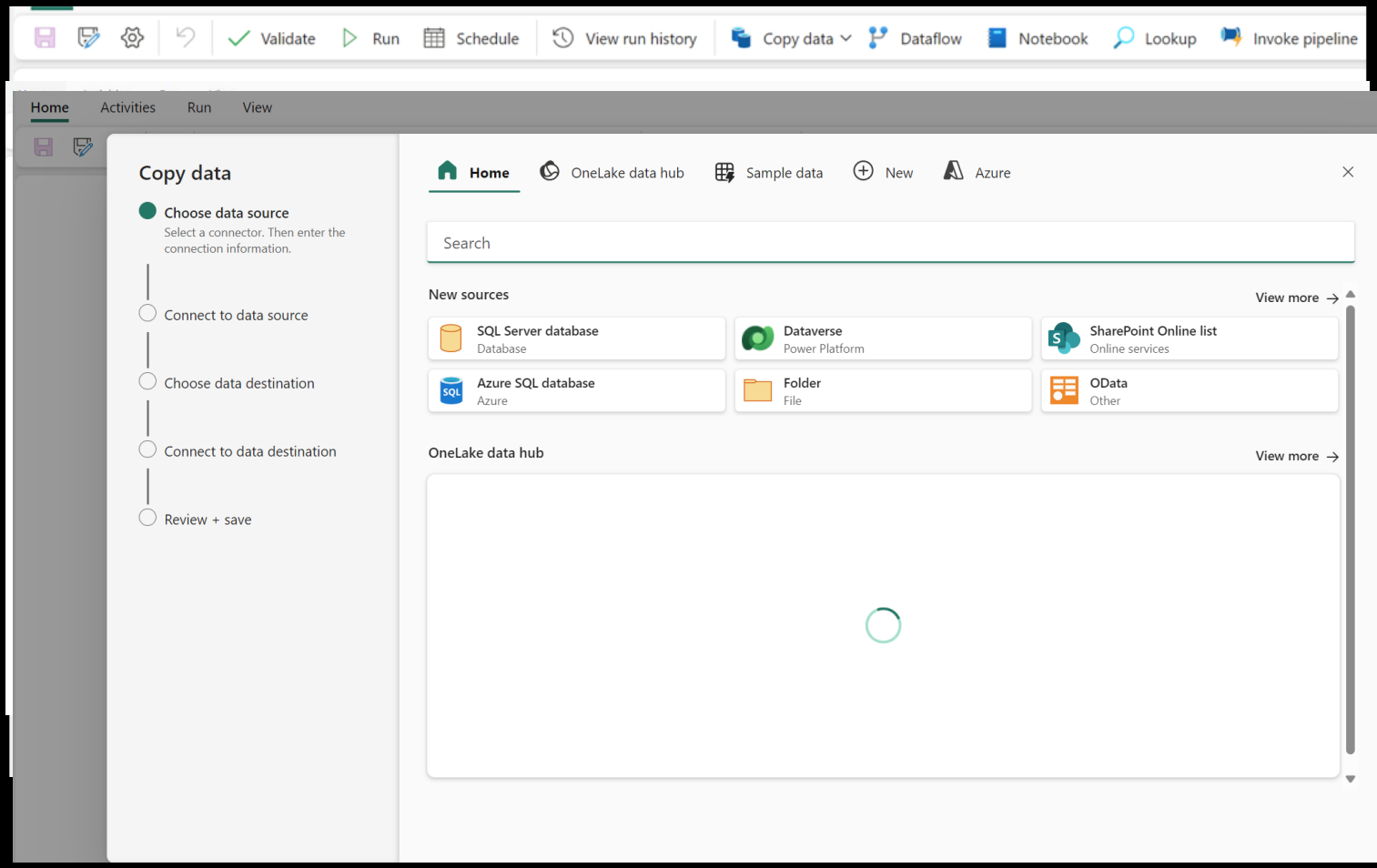
Templates

- Templates are pre-defined pipelines that allow you to get started quickly with Data Factory.
- These templates help to reduce development time by providing an easy way to create pipelines.
- Templates are available for common data integration scenarios.
- Templates can be customized to meet specific requirements



Copy Assistant

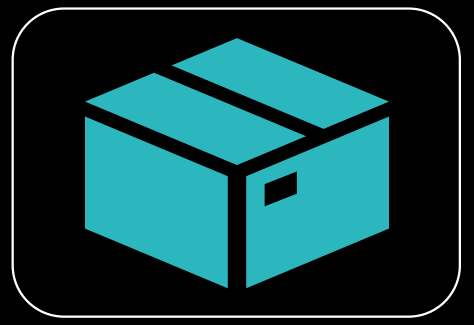
- Reads data from a source data store.
- Performs serialization/deserialization, compression/decompression, column mapping, and so on.
- Writes data to the destination data store.



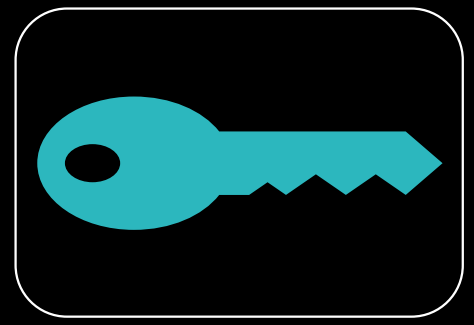
Frameworks

Why

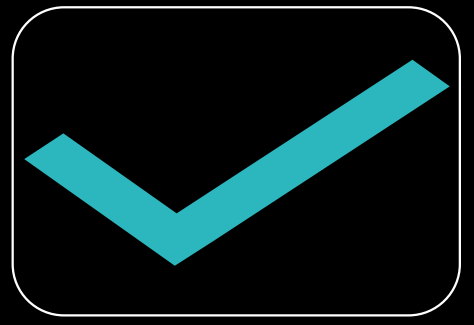




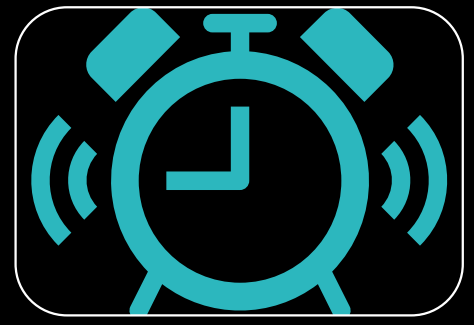
Scalability



Automation



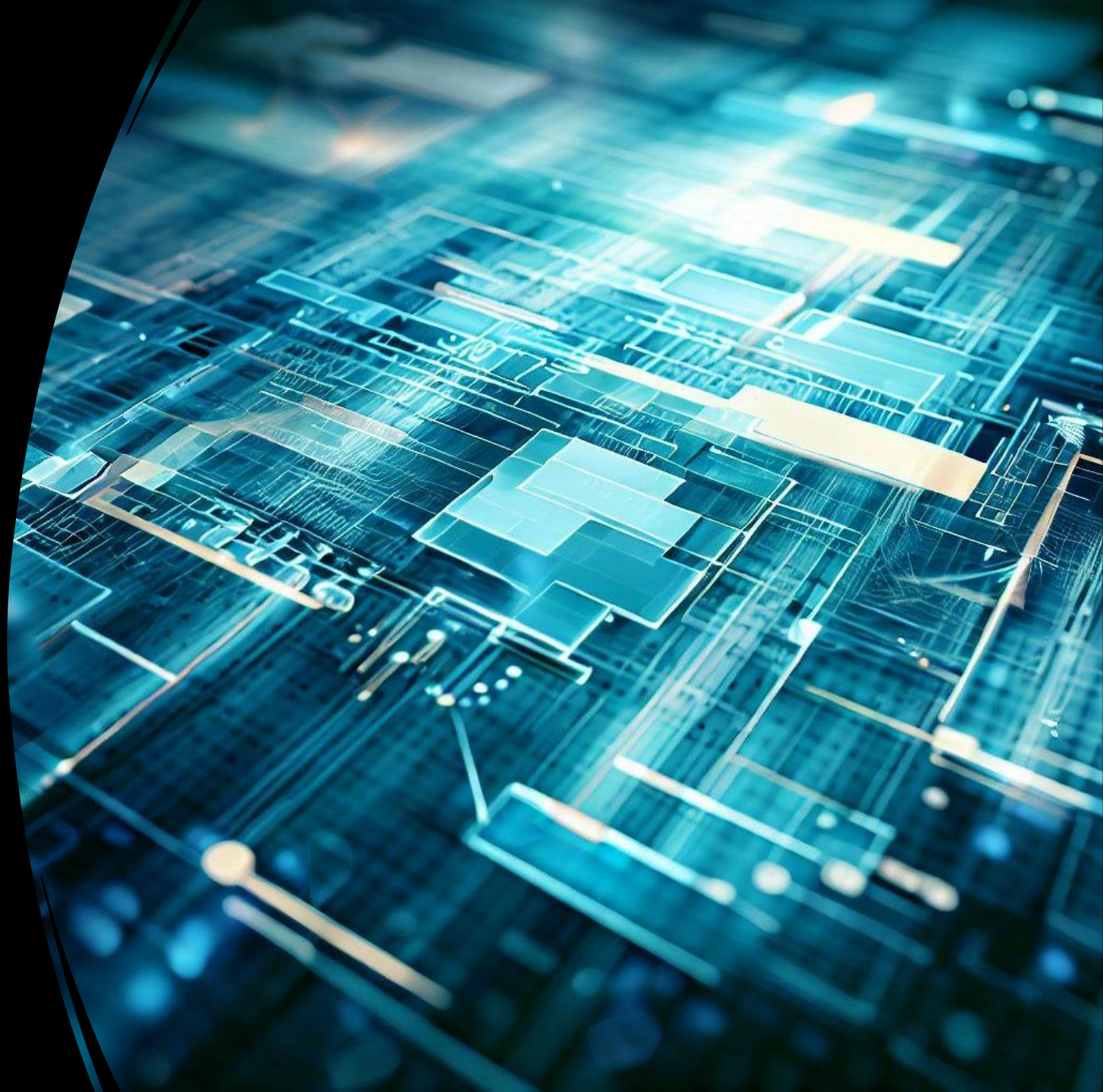
Traceability



Flexibility

Out-of-the-Box Framework

- Ready-to-use.
- Rapid implementation.
- Limited customization.
- Lower development effort.
- Lower upfront costs.
- Ongoing support and updates.



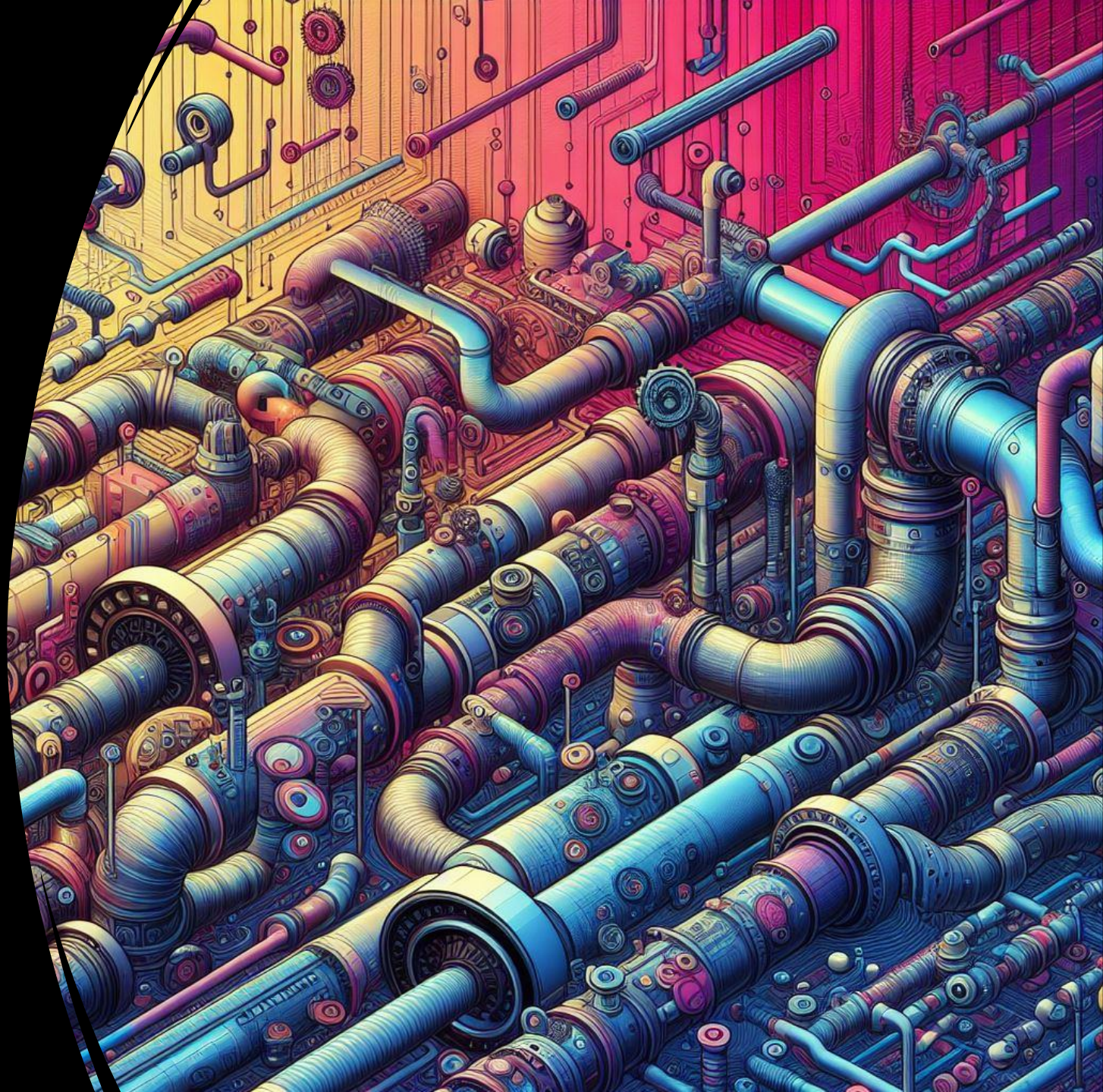
Custom-Made Framework

- Tailored to specific needs.
- Full control over design and features.
- Higher development effort.
- Flexibility and extensibility.
- Higher upfront costs.

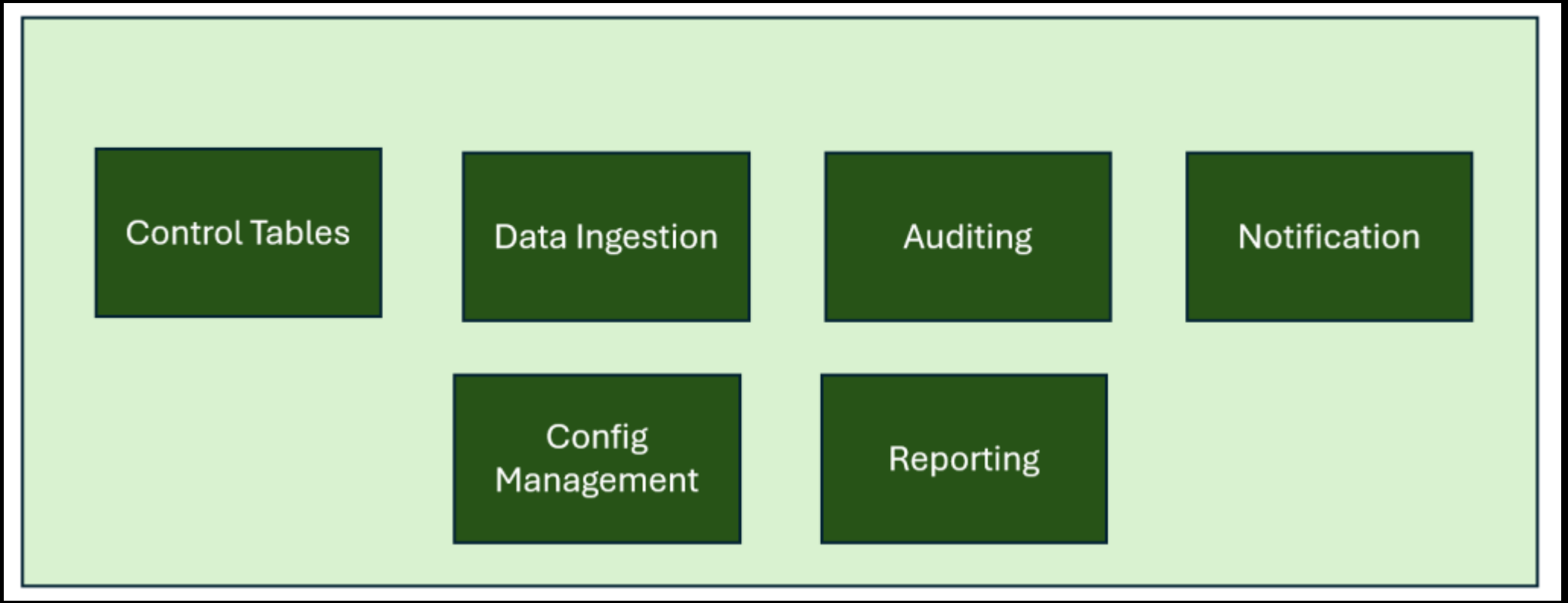


Custom-Made Framework

- Based on parameters
- Meta data => Azure SQL Database / Json / ……
- Microsoft Fabric but also on Azure Synapse Analytics and Azure Data Factory
- Based on the Medallion Architecture

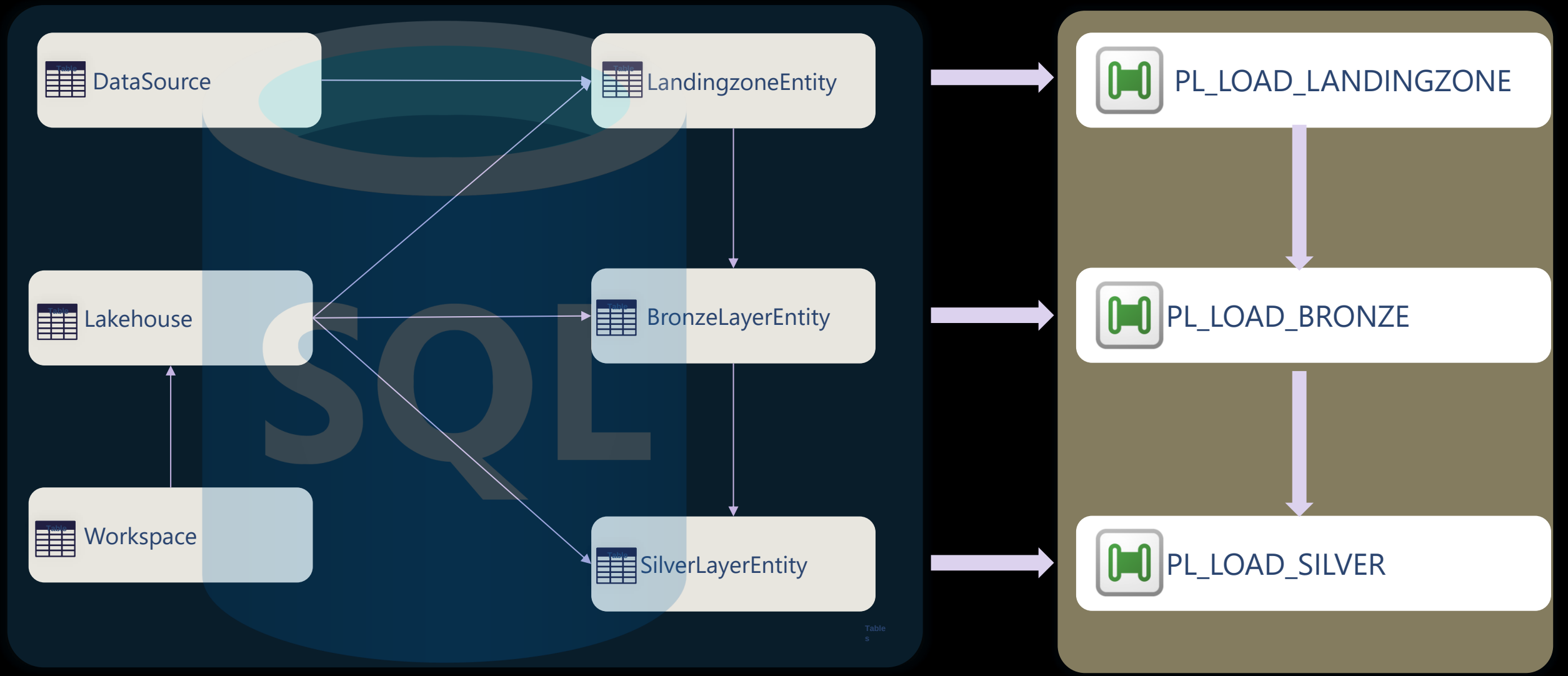


Custom-Made Framework



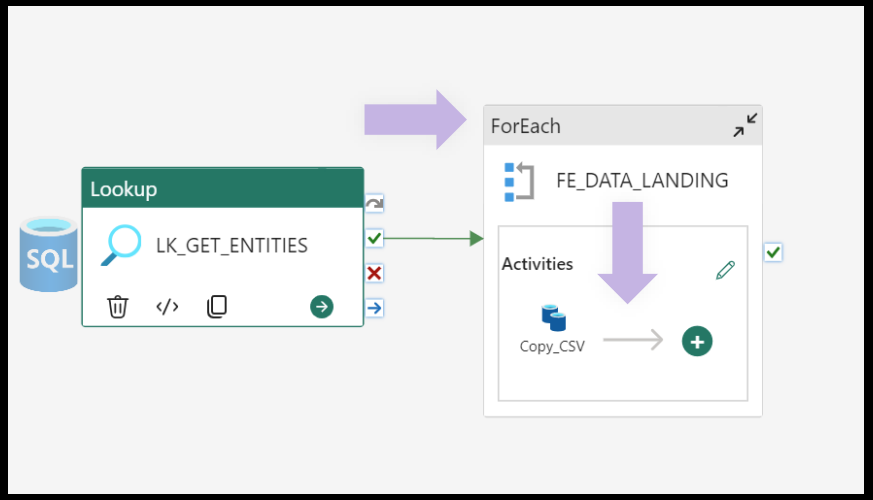
Demystifying Data Ingestion in Fabric: Fundamental Components for Ingesting Data into a Fabric Lakehouse using Fabric Data Pipelines

Framework


DataSource
LandingzoneEntity
Lakehouse
BronzeLayerEntity
Workspace
SilverLayerEntity
PL_LOAD_LANDINGZONE
PL_LOAD_BRONZE
PL_LOAD_SILVER

Copy Activity Parameters

- Pass Parameters from Pipeline to Copy activity
- Use Parameters from For Each Activity



Preview data

ceName	DataSourceAbbreviation	DataSourceType	IsActive	SourceSchema	SourceName	TargetFilePath	TargetFileName	TargetFileType
uwdvlmdl01	ADLS	ADLS	true		customers.csv	demo/2024/09/10	customers_2024091015.csv	csv
uwdvlmdl01	ADLS	ADLS	true		organizations.csv	demo/2024/09/10	organizations_2024091015.csv	csv
uwdvlmdl01	ADLS	ADLS	true		people.csv	demo/2024/09/10	people_2024091015.csv	csv

General | Source | **Destination** | Mapping | Settings

Connection * : @item().TargetLakehouseGuid

Connection type * : Lakehouse

Workspace ID : @item().WorkspaceGuid

Root folder : Tables / Files

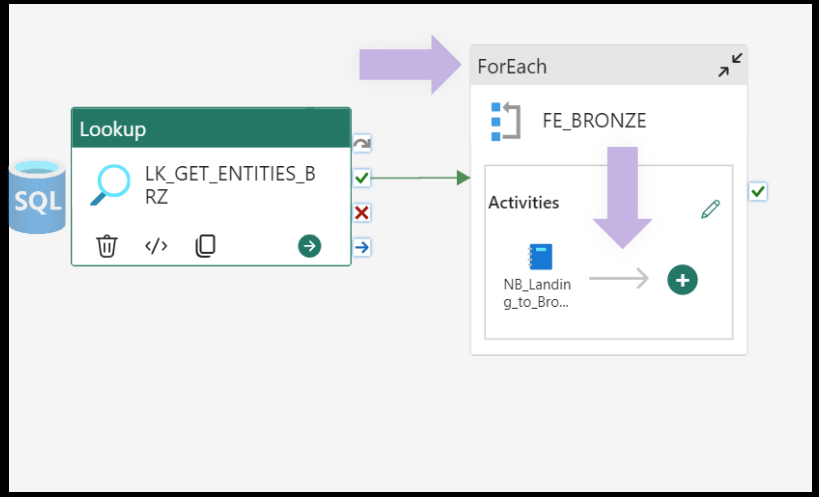
File path : @item().TargetFilePath / @item().TargetFileName Browse

File format * : Binary Settings

Advanced

Notebook Parameters

- Pass Parameters from Data Pipeline to Notebook
  - Toggle parameter cell



Preview data

	EntityId	SourceFilePath	SourceFileName	SourceFileType	TargetSchema	TargetName	TargetWorkspaceId	SourceWorkspaceId	TargetLakehouseId	SourceLakehouseId	TargetLakehouseName	SourceLakehouseNa
1	1	demo/2024/09/10	customers_2024091017.csv	csv	demo	Customers	586fc19d-fa6a-4cb1-9ca3-e518a524f5da	586fc19d-fa6a-4cb1-9ca3-e518a524f5da	e130dba7-c8c3-438a-85ad-2cd4c9d59a09	009058ac-b71c-4774-a8ee-7c7d945c3972	LH_Bronze_Layer	LH_Data_Landingzor
2	3	demo/2024/09/10	organizations_2024091017.csv	csv	demo	Organizations	586fc19d-fa6a-4cb1-9ca3-e518a524f5da	586fc19d-fa6a-4cb1-9ca3-e518a524f5da	e130dba7-c8c3-438a-85ad-2cd4c9d59a09	009058ac-b71c-4774-a8ee-7c7d945c3972	LH_Bronze_Layer	LH_Data_Landingzor
3	4	demo/2024/09/10	people_2024091017.csv	csv	demo	People	586fc19d-fa6a-4cb1-9ca3-e518a524f5da	586fc19d-fa6a-4cb1-9ca3-e518a524f5da	e130dba7-c8c3-438a-85ad-2cd4c9d59a09	009058ac-b71c-4774-a8ee-7c7d945c3972	LH_Bronze_Layer	LH_Data_Landingzor

Notebook * NB_LANDING_BRONZE Refresh Open New

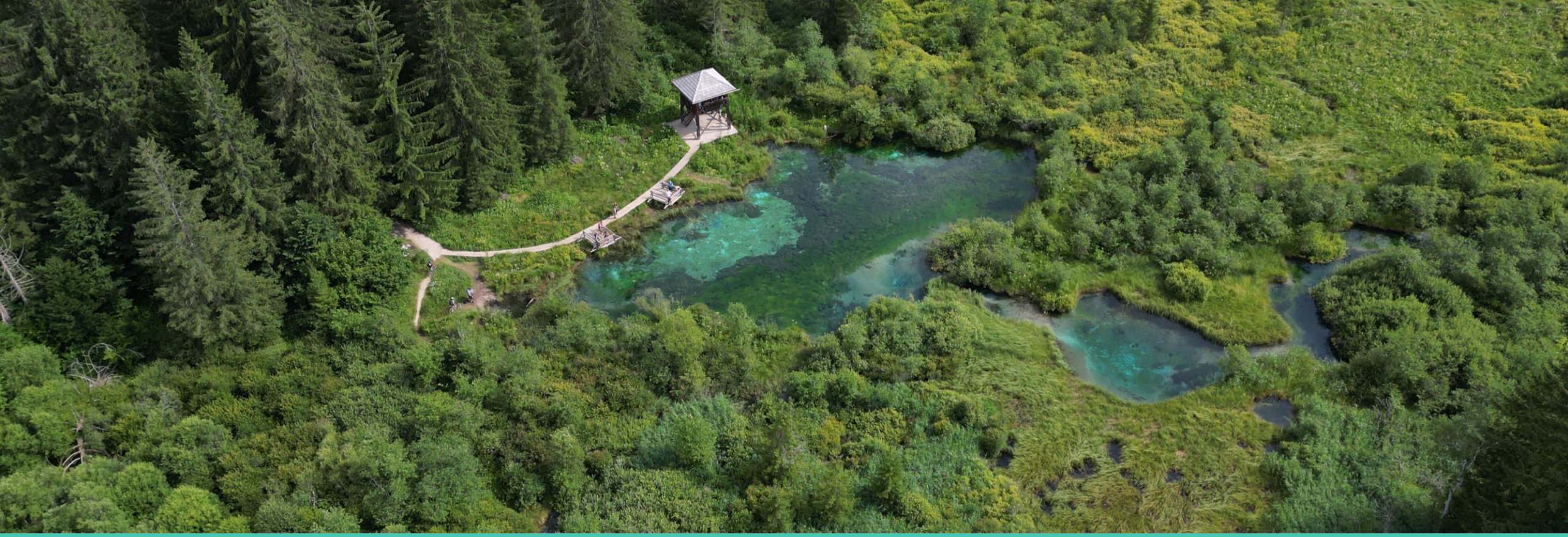
Base parameters

New Delete

Name	Type	Value
SourceLakehouse	String	@item().SourceLakehouseId
source_file_path	String	@item().SourceFilePath
source_file_name	String	@item().SourceFileName
PrimaryKeys	String	@item().PrimaryKeys
TargetLakehouse	String	@item().TargetLakehouseId
target_schema	String	@item().TargetSchema
target_name	String	@item().TargetName
SourceWorkspace	String	@item().SourceWorkspaceId
TargetWorkspace	String	@item().TargetWorkspaceId
source_file_type	String	@item().SourceFileType

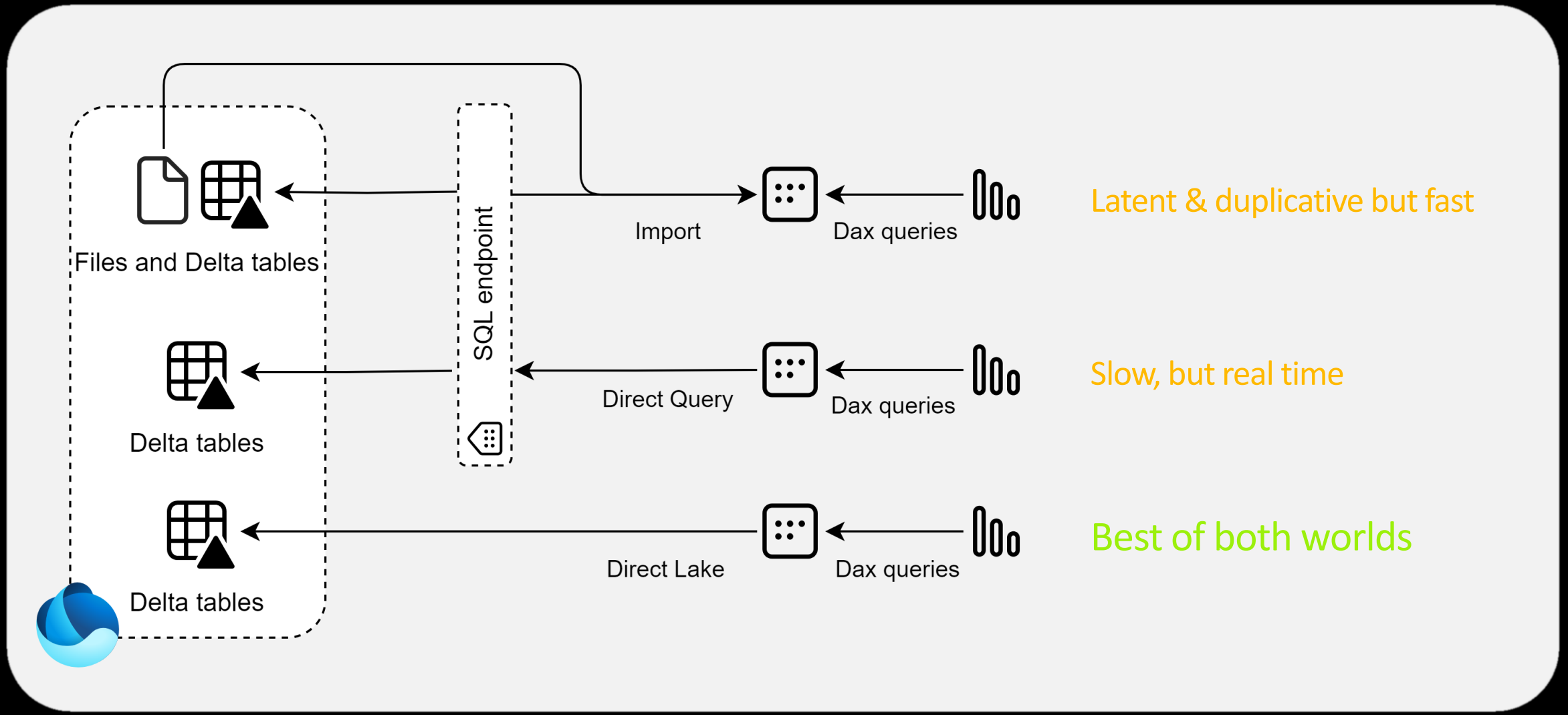
Demo





Build your semantic model benefitting Direct Lake

Understanding Direct Lake



Direct Lake is only applicable to Fabric

Data transformations



No Power Query or other data transformation capabilities*

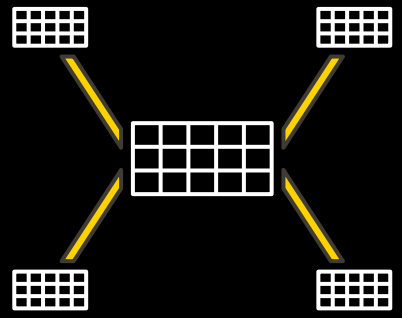


Data transformations should be done as far upstream as possible

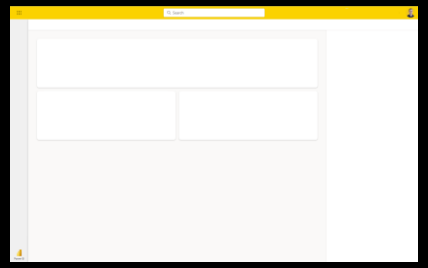


Data transformation directly in the Lakehouse unlocks “new” possibilities

Data modeling



General best practice to have a star schema still applies



Web (browser) experience only to develop data models (for now)

Data modeling best practices unchanged

- Starschema all the things!
- Avoid bi-directional or many-to-many relationships
- Avoid limited relationships
- Implement role-playing dimensions rather than duplication
- Minimize redundant measure using calculation groups
- Avoid ambigiuous data models
- … etcetera

Demo





Direct Lake Internals & performance

Delta (Parquet)



Parquet (File format)

- Column-store
- Open industry standard
- Compressed & Encoded
- Parallelism

Enables fast bulk operations of large data volumes

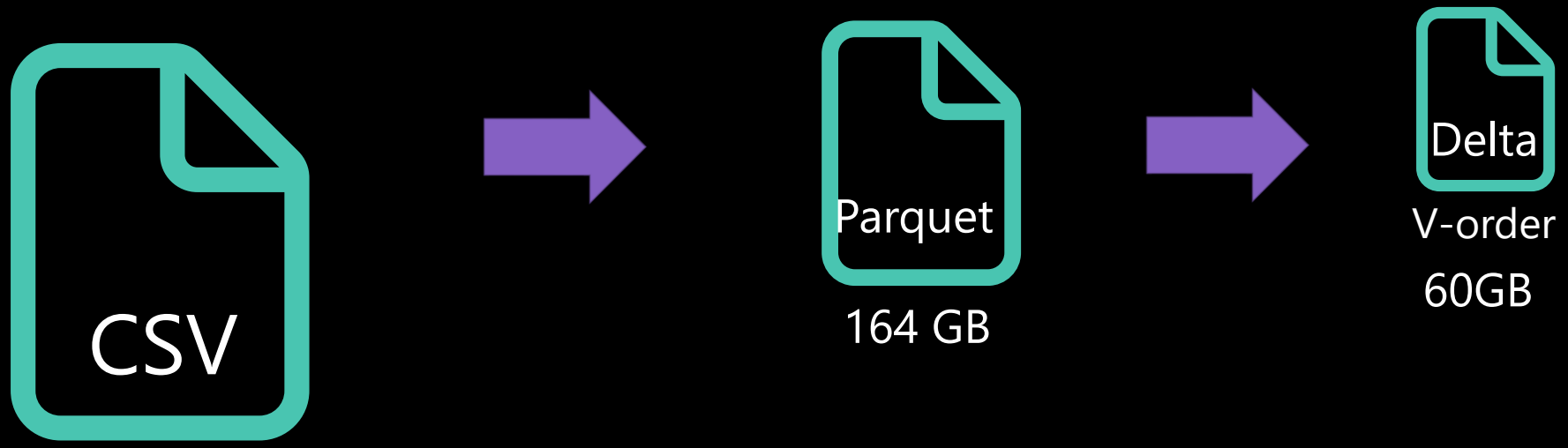
Delta (Storage layer management)

- Also, an open industry standard
- ACID transactions & schema enforcement
- Delete, update, merge
- Time-travel
- Optimized for querying, skipping and pruning

Brings warehouse reliability to the data lake

From Z-order to V-order

Yellow taxi (3 Billion rows)



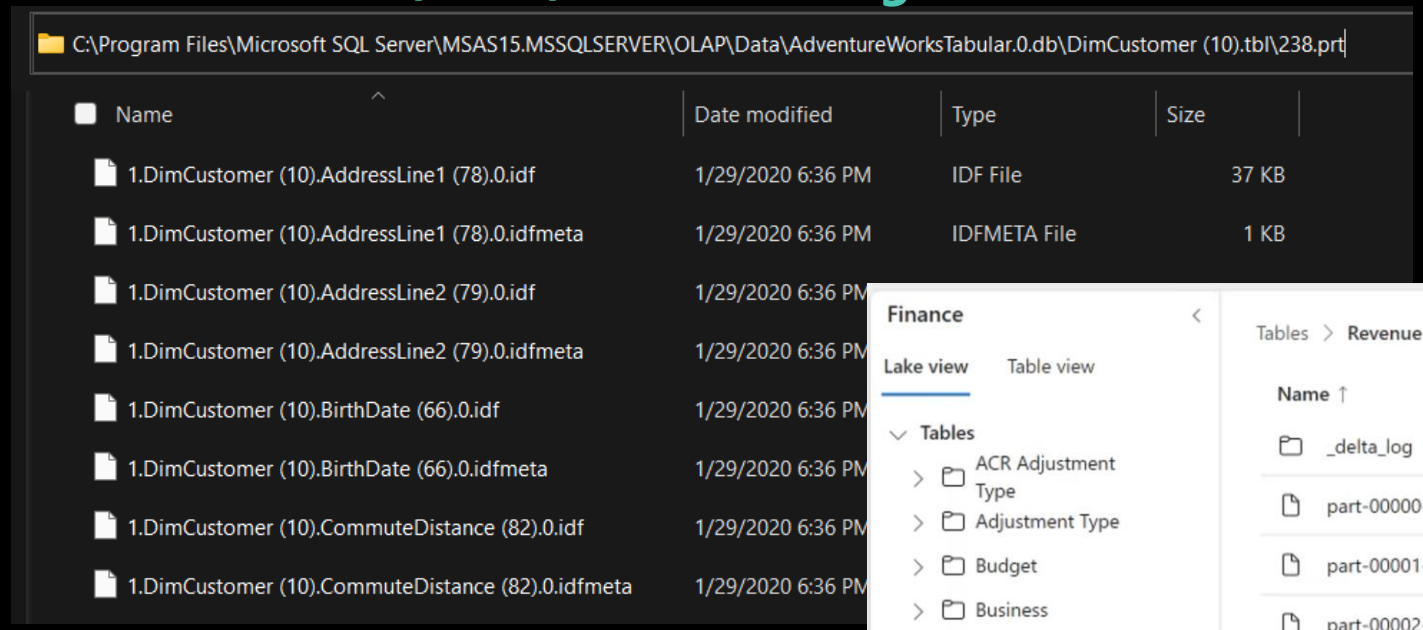
x3.2
Less I/O for all* workloads

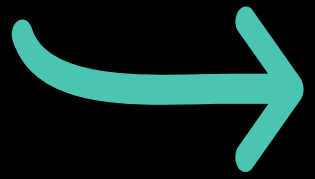
* Microsoft benchmark

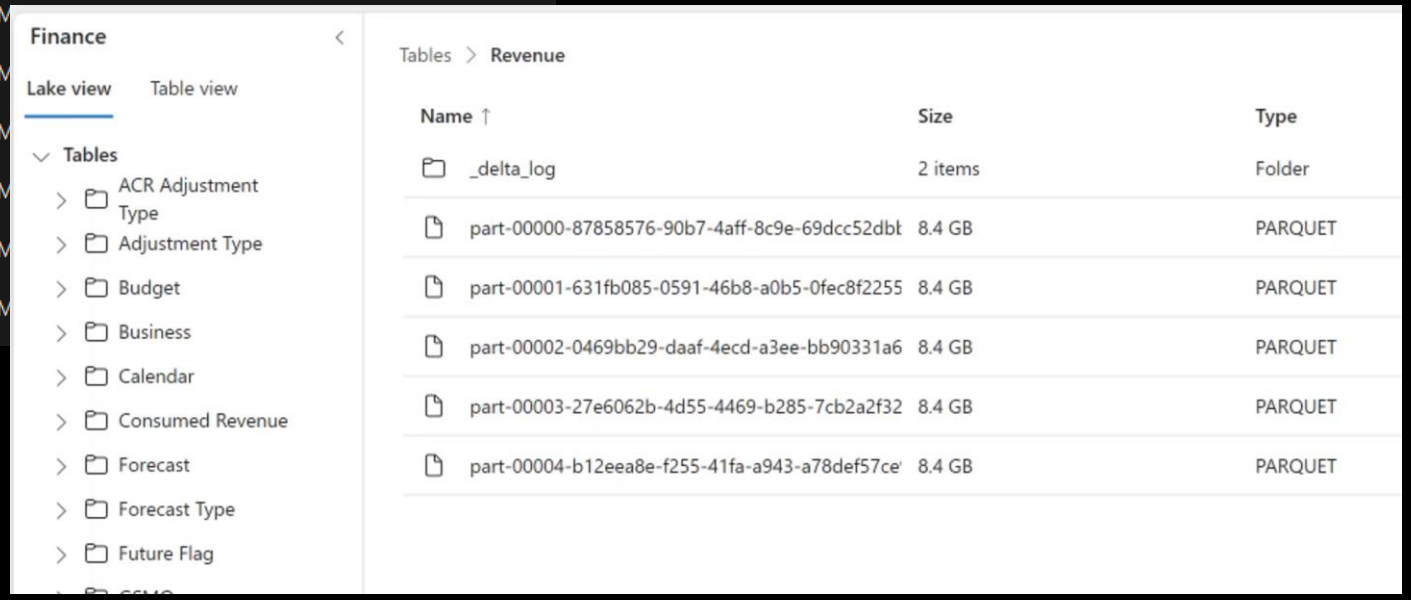
Data saving

The Analysis Services column-oriented storage using Delta Lake/Parquet open standard for Direct Lake

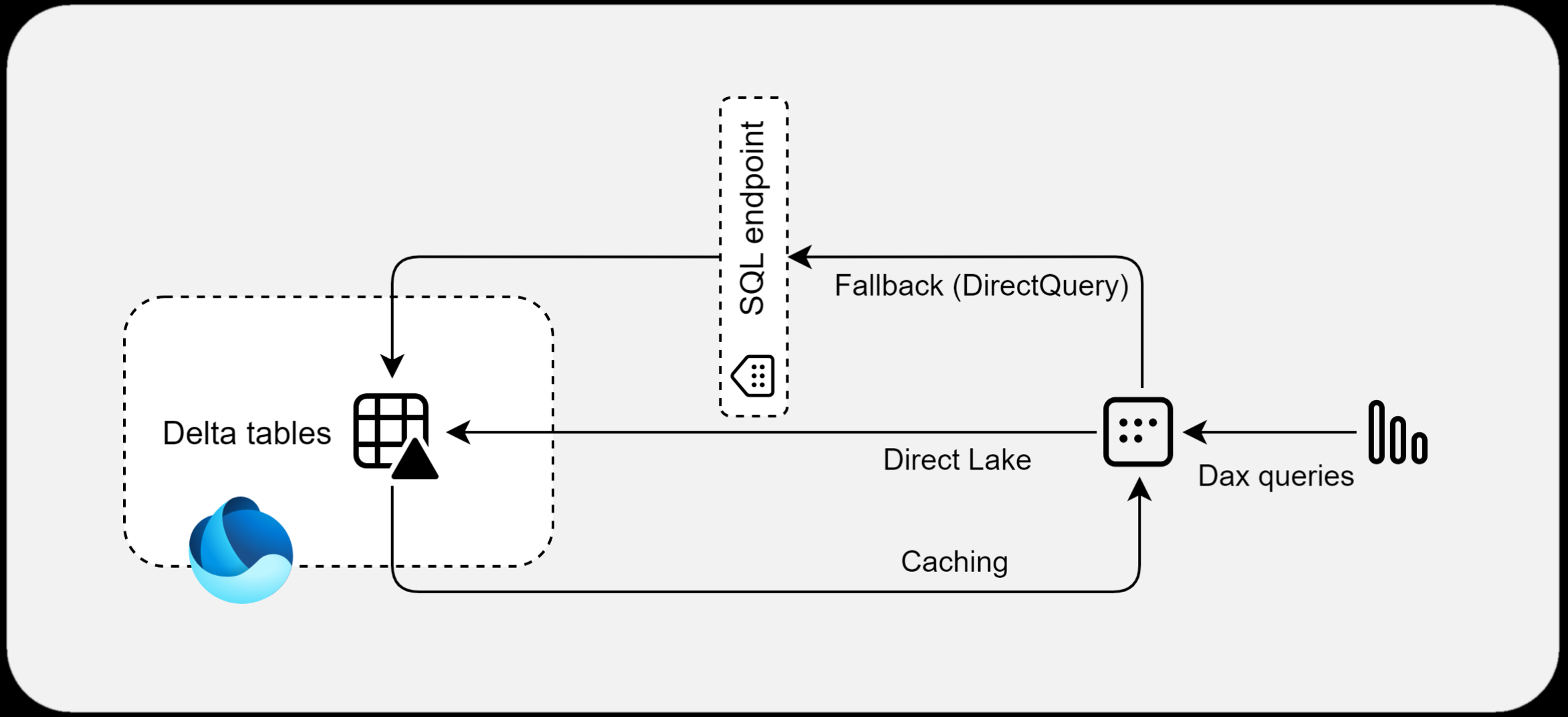
SSAS, AAS, Power BI large models







Fallback & Caching



Fallback

When could fallback to DirectQuery happen?

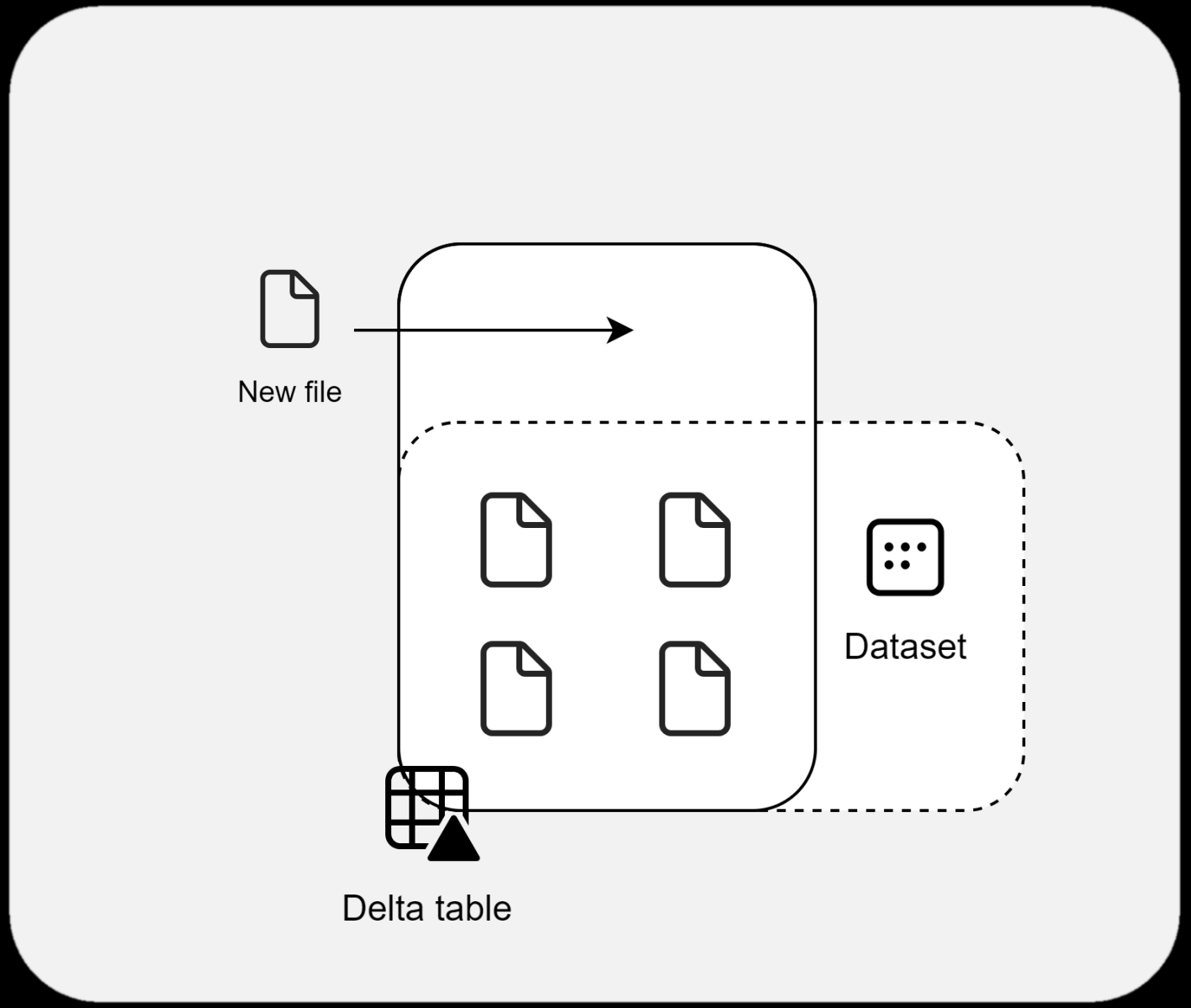
- Special data types
- Large data volumes that does not fit the capacity size
- Composite models
- When you manually configure security
  Item level on lakehouse

Introducing Framing

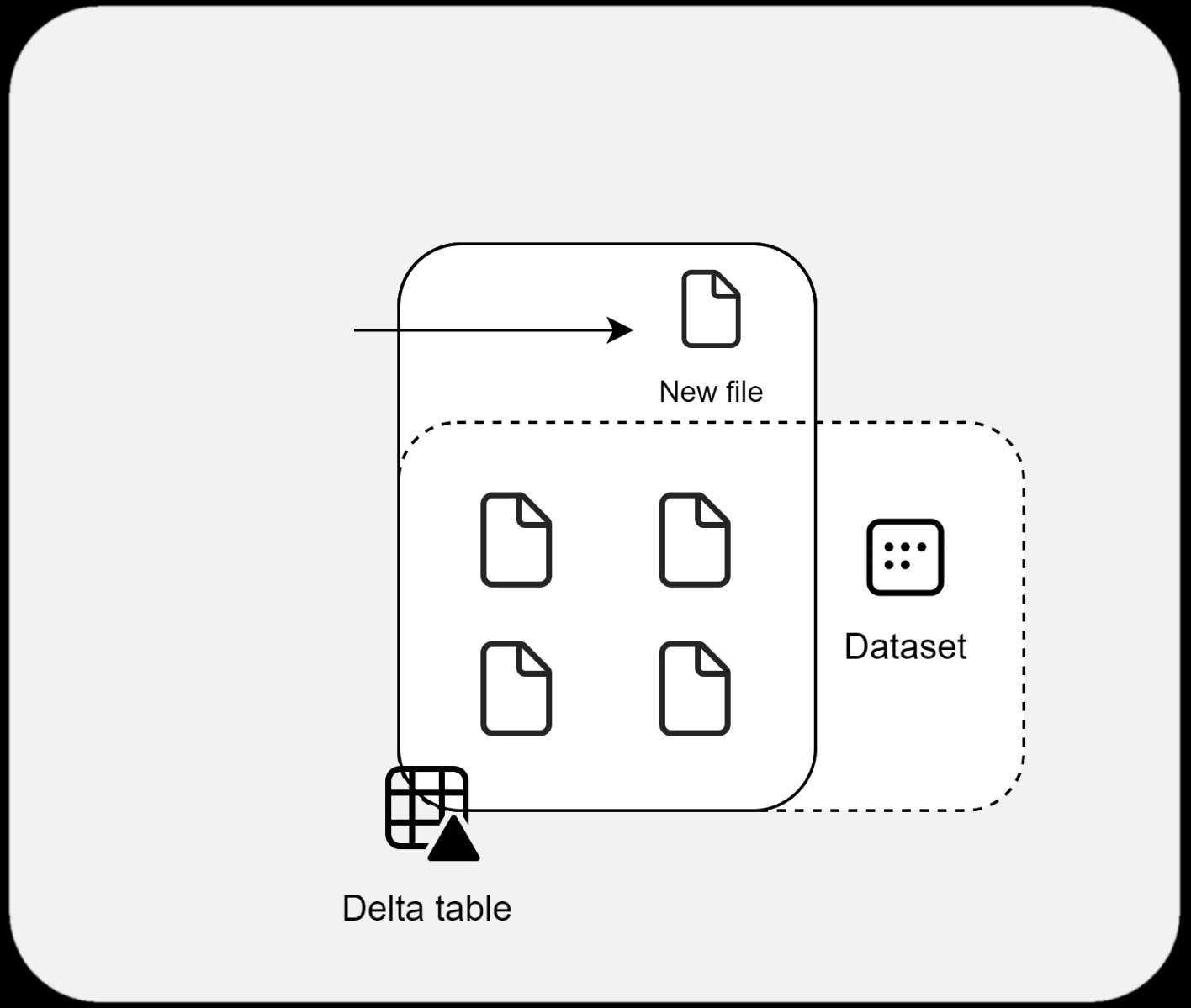
- Metadata **refresh** which does not actually load the data, but only the delta table definitions.



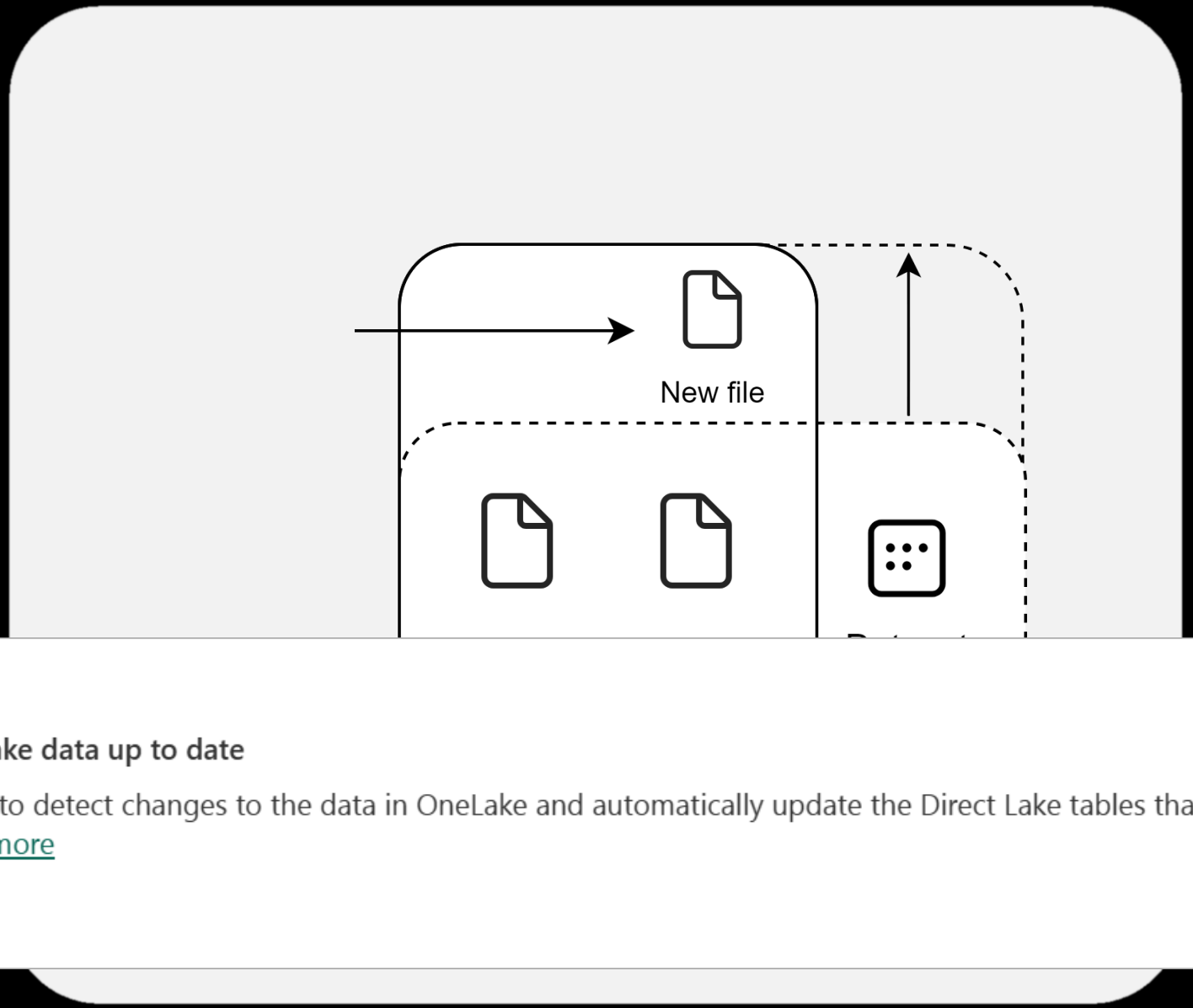
Framing



Framing



Framing



Refresh

Keep your Direct Lake data up to date

Configure Power BI to detect changes to the data in OneLake and automatically update the Direct Lake tables that are included in this dataset. Learn more

Off

Demo



Temperature management

Keep it WARM!

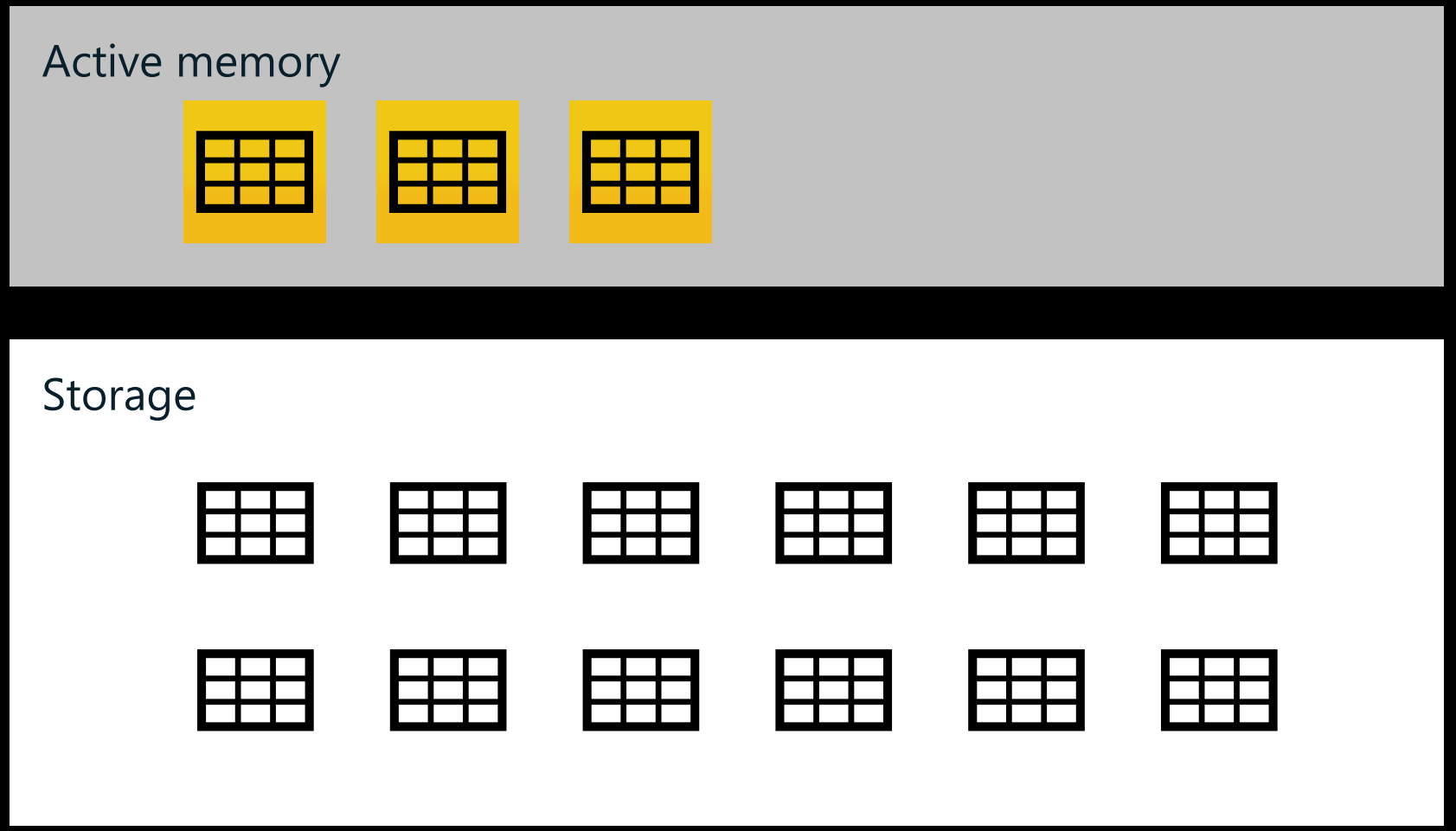
Make sure your users are served optimally and avoid the capacity memory to be flushed.



Eviction

Capacity: F64 / P1
Memory: 25 GB

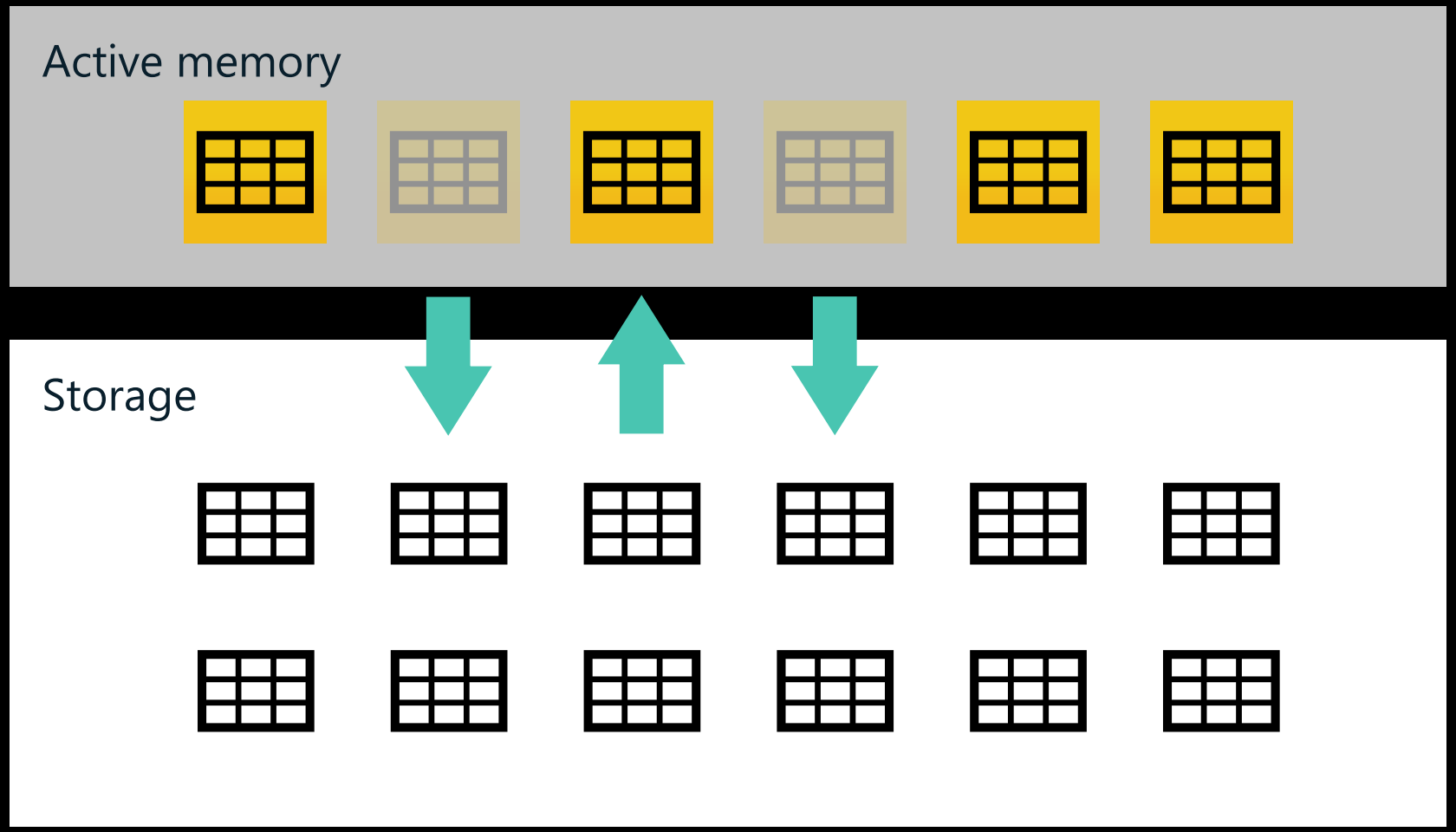
Capacity utilization:
Cool



Eviction

Capacity: F64 / P1
Memory: 25 GB

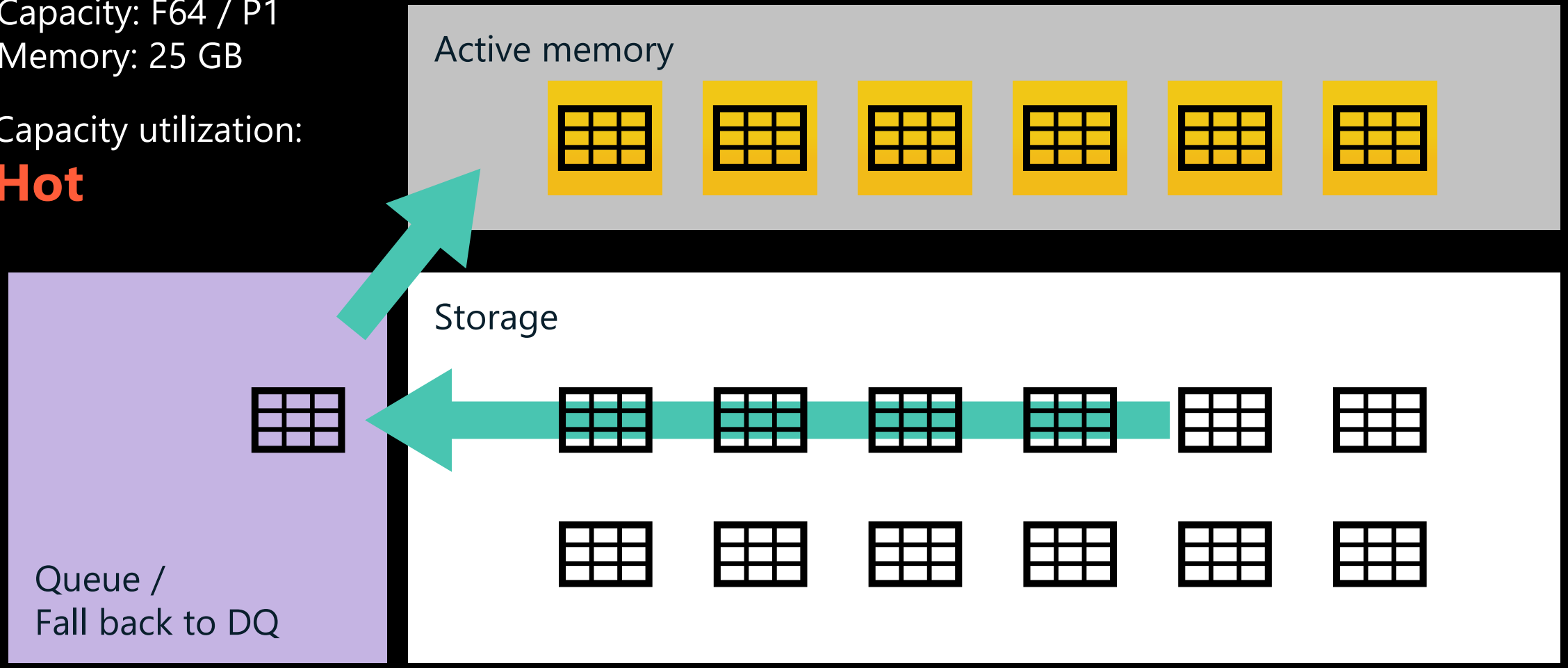
Capacity utilization:
Warm



Eviction – queue / fall-back

Capacity: F64 / P1
Memory: 25 GB

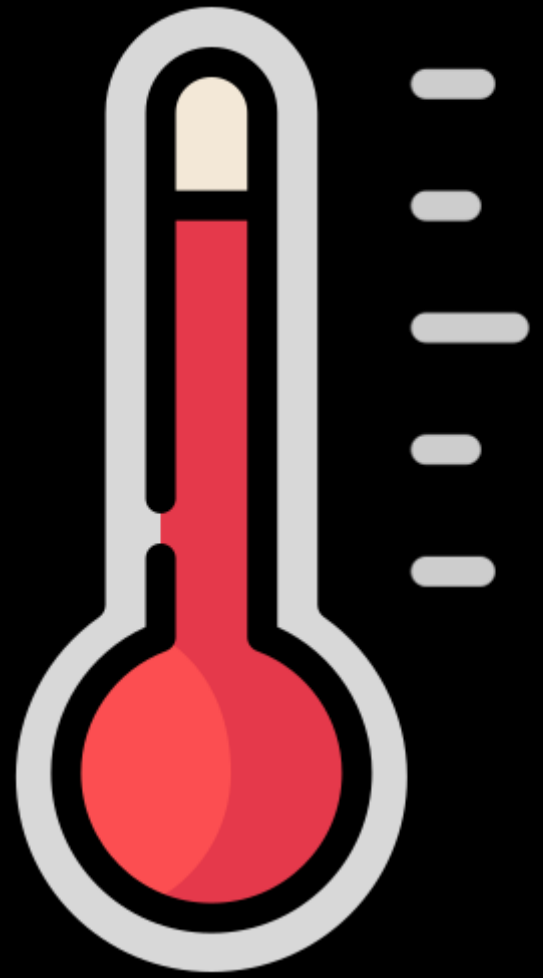
Capacity utilization:
Hot



Temperature management

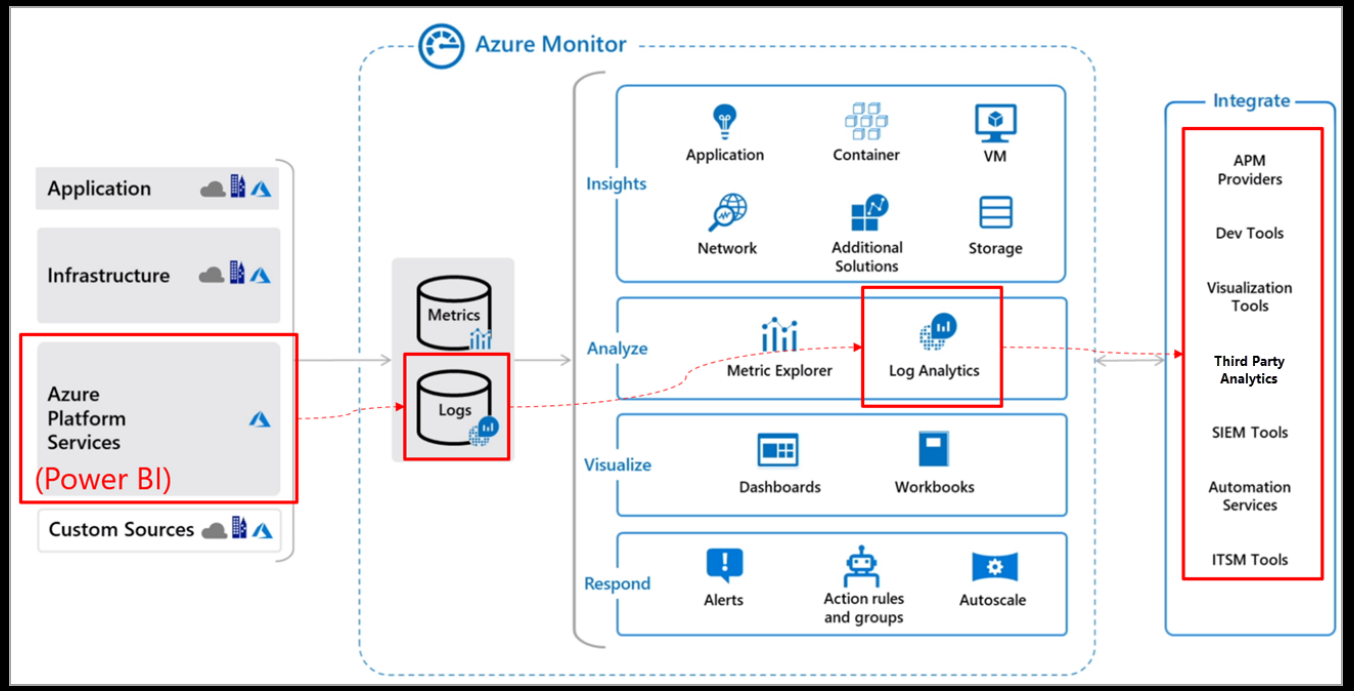
What will be evicted?
Basically, your data will be evicted from active memory, that you want to always have available!

How can you influence that?
Consider setting up a process (notebook, other automated setup) to pro-actively execute queries to keep certain data **WARM**!
.



What should stay in memory?

Azure Monitor delivers a comprehensive solution for collecting, analyzing, and acting on telemetry from your cloud and on-premises environments. It helps you understand how your applications are performing and proactively identifies issues affecting them and the resources they depend on.



Demo



Copilot in Microsoft Fabric



Microsoft Fabric Copilot

Unlock the full potential of your data



Supercharge Productivity

Let AI summarize actions for you, and optimize your time



Uplevel Skills

Be better at what you're good at and master what you've yet to learn



Discover Insights

Harness the power of AI so you never have to start from scratch

Microsoft Cloud Runs on trust...

Your data is your data

Your data from any fine-tuning is not used to train the foundation AI models

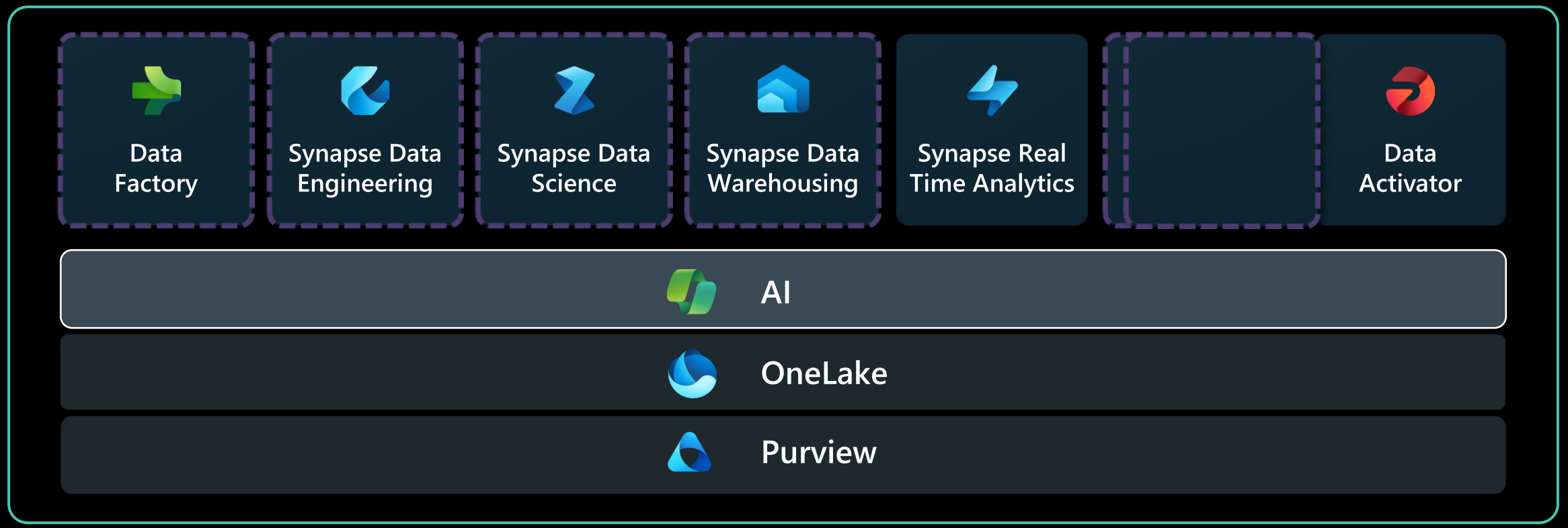
Your data is protected by the most comprehensive enterprise compliance and security controls

Autopilot refers to allowing AI to make all the decisions, while copilot involves AI assisting humans in making informed



Copilot experiences

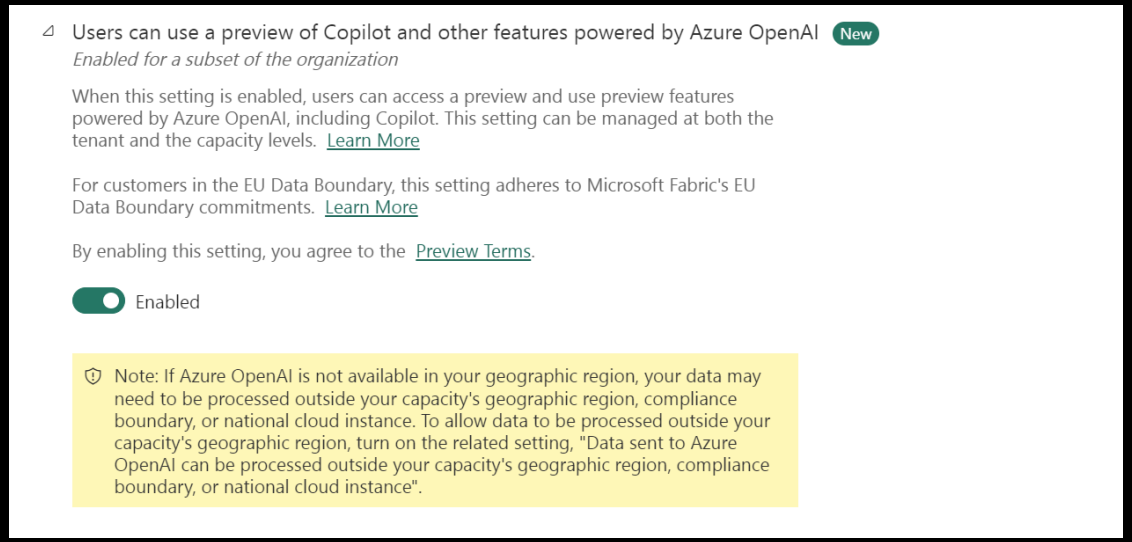
The unified data platform for the era of AI

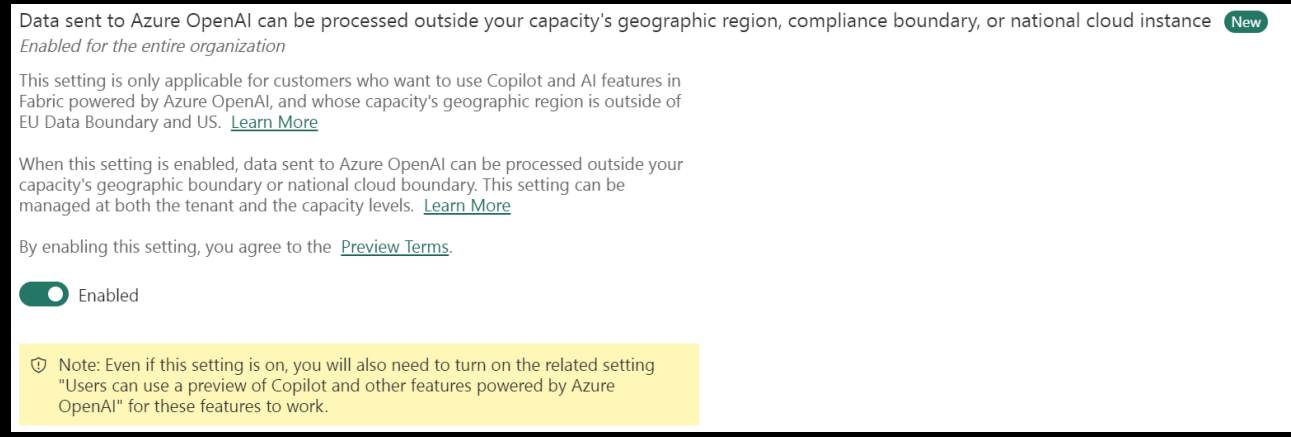


Unified architecture | Unified experience | Unified governance | Unified business model

Copilot requirements

- Paid SKU (F64 or higher, or P1 or higher) No Trial
- Fabric Region WestEurope / NorthEurope (and much more)
- Copilot enabled in admin portal (as of May 20th by default)





Privacy, security, and responsible use for Copilot in Microsoft Fabric

Your business data is secure

•Copilot features use Azure OpenAI Service, which is fully controlled by Microsoft. Your data isn't used to train models and isn't available to other customers.

•You retain control over where your data is processed. Data processed by Copilot in Fabric stays within your tenant's geographic region, unless you explicitly allow data to be processed outside your region—for example, to let your users use Copilot when Azure OpenAI isn't available in your region or availability is limited due to high demand. Learn more about admin settings for Copilot.

Check Copilot outputs before you use them

•Copilot responses can include inaccurate or low-quality content, so make sure to review outputs before you use them in your work.

•Reviews of outputs should be done by people who can meaningfully evaluate the content's accuracy and appropriateness.

•Today, Copilot features work best in the English language. Other languages may not perform as well.

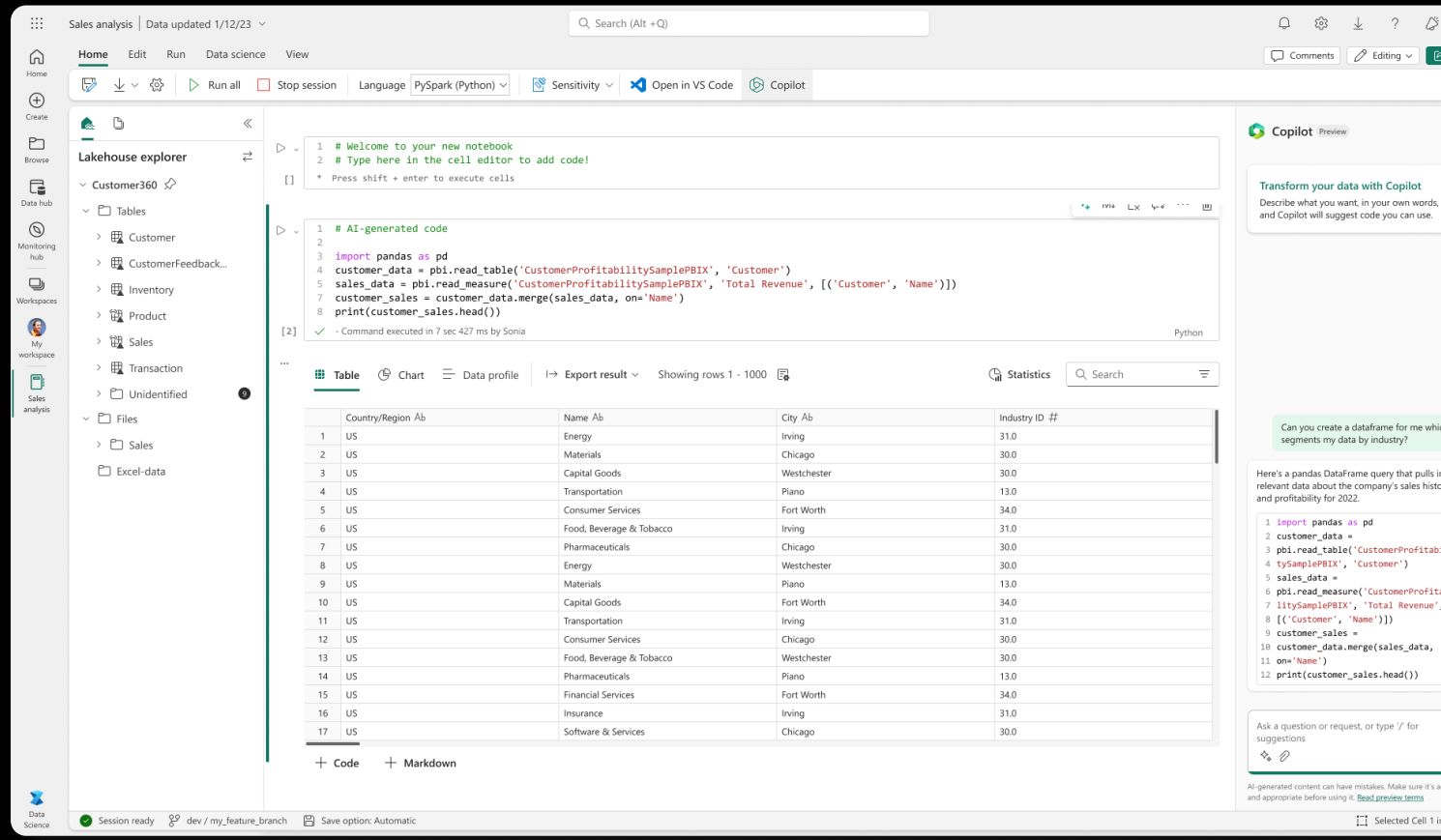
Copilot for Data Factory

- Use Copilot to get data, transform , and enriched data in Dataflows

Use Copilot to provide a summary of the query and the applied steps.

Use Copilot to generate new transformation steps for an existing query.

Use Copilot to generate a new query that may include sample data or a connection to a data source that requires configuring authentication.



Copilot for Data Science and Engineering

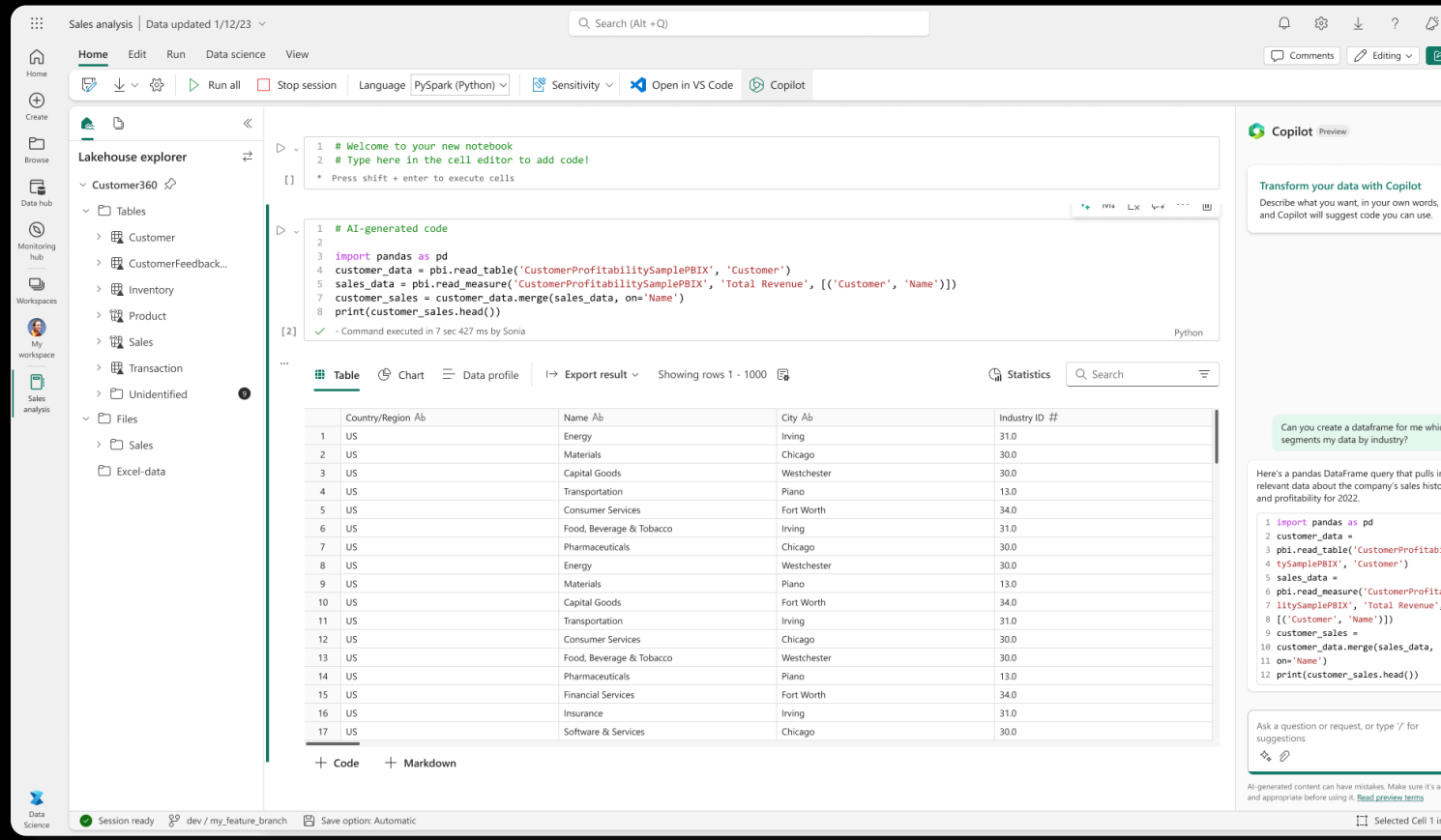
- Use Copilot to **enrich**, **model**, **analyze**, and **explore** data in notebooks

Work with **Copilot** to understand how to explore/transform 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



Generative AI on your data

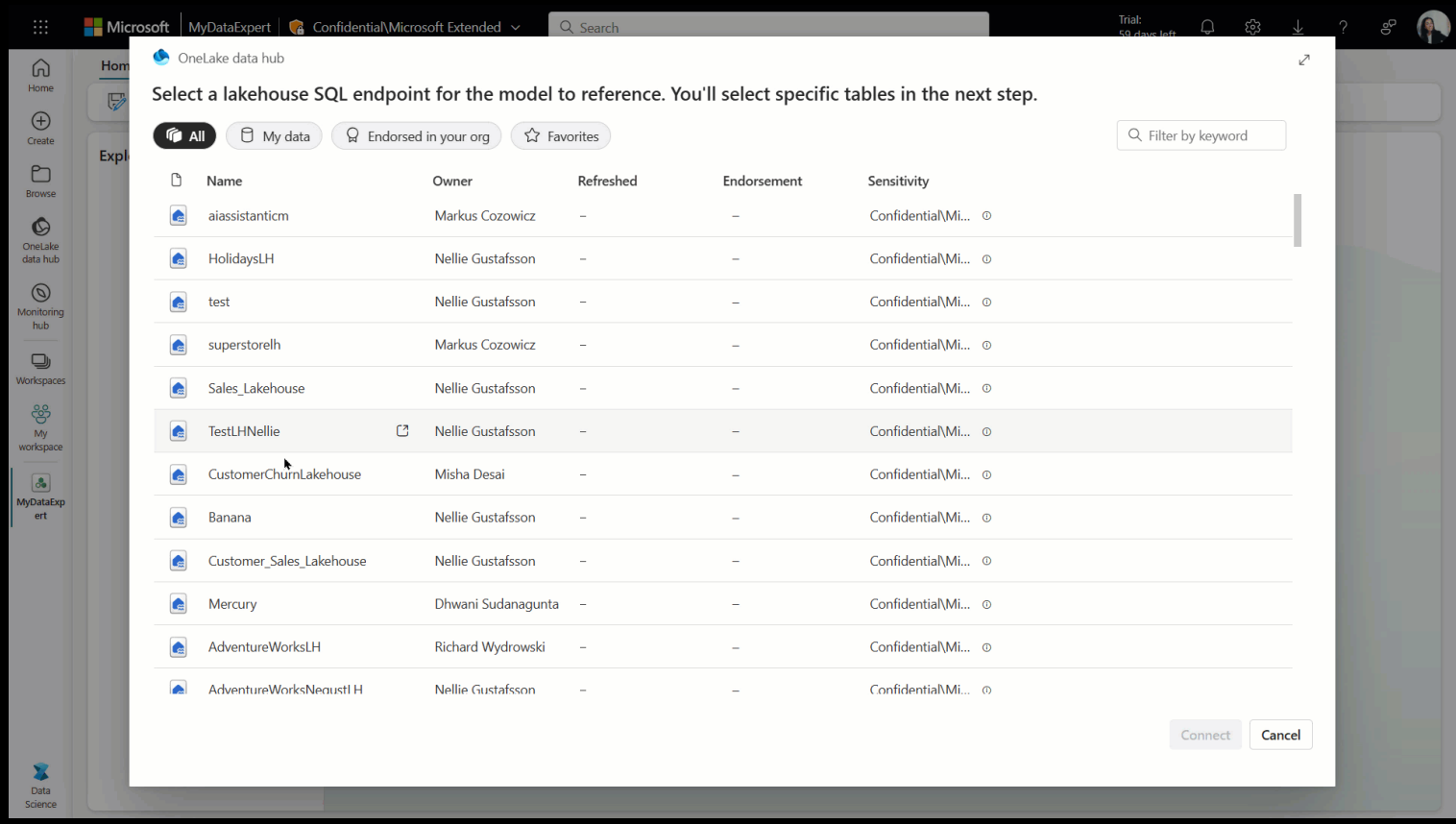
(Public Preview)

Deliver custom generative AI experiences for your data

Enable **custom Q&A** on your data in Fabric

Define custom business semantics and grounding **unique to your org**

Scale the custom experiences to **M365 Chat**, **Copilot Studio**, and **Azure AI Studio**



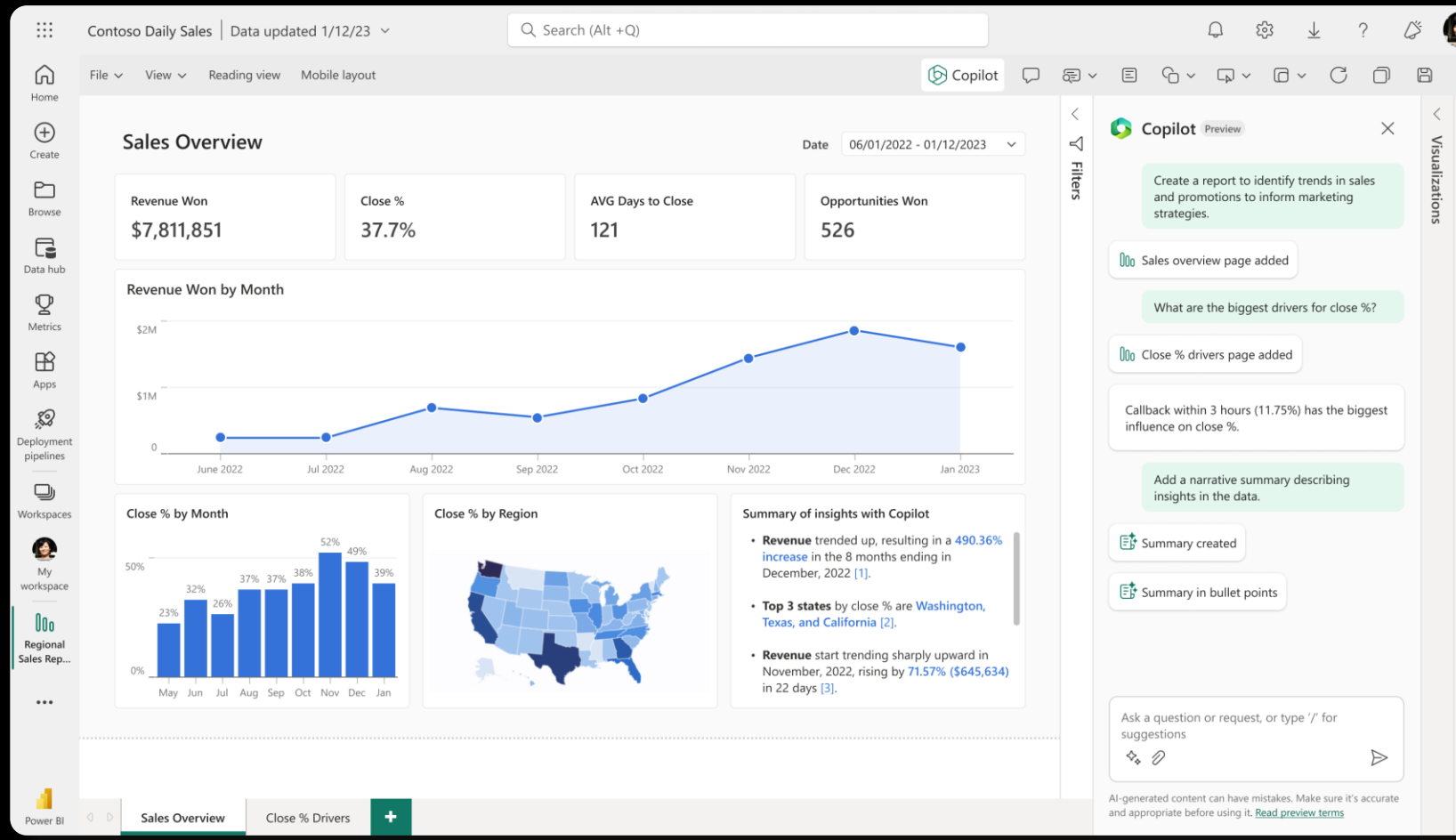
Copilot for Power BI

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

Ask **Copilot** to create beautiful and insightful reports based on your needs

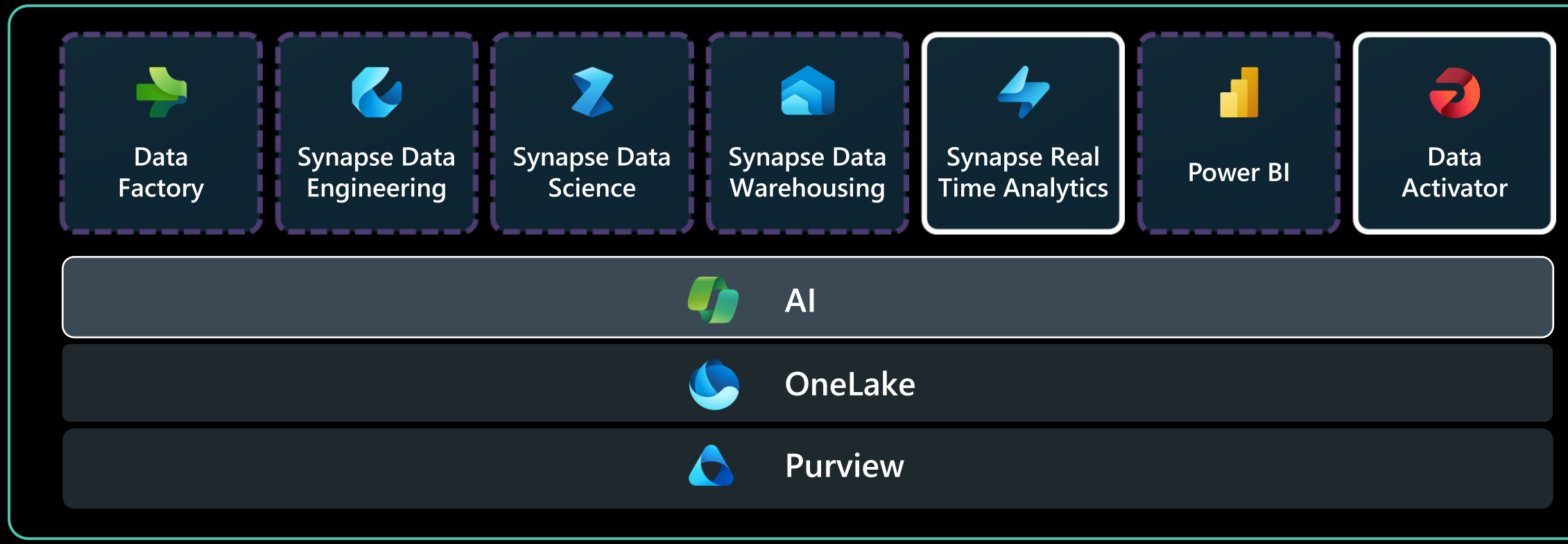
Define metrics and calculations using **Copilot** for your semantic model just by describing them in natural language

Use **Copilot** to tap into Power BI's advanced AI features for finding and visualizing insights



Copilot experiences

The unified data platform for the era of AI



Unified architecture | Unified experience | Unified governance | Unified business model

Demo



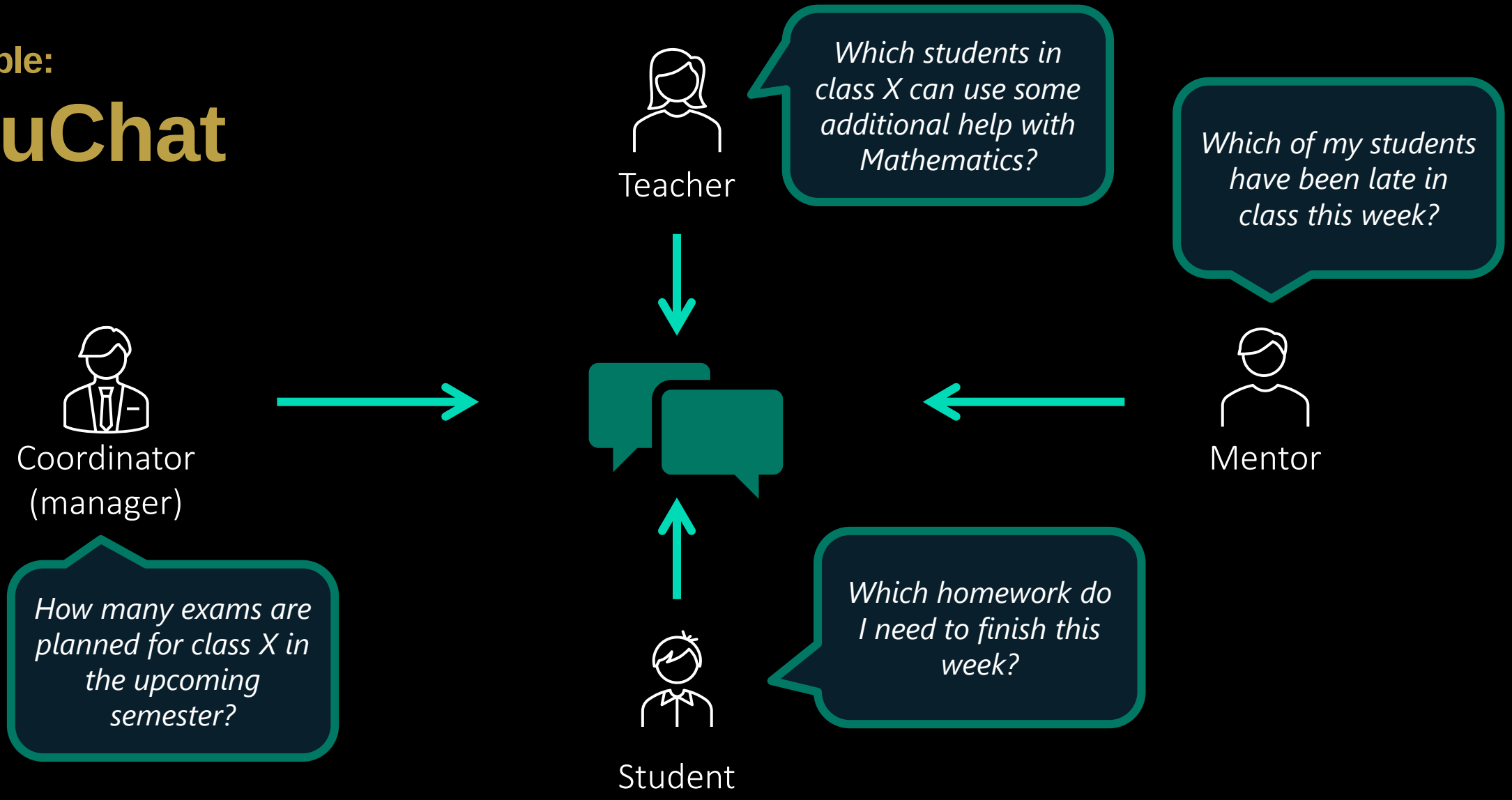
AI Skills

AI Skills conceptual

- Ask questions directly to your data in natural language
- All data fetched from lakehouses in Fabric
- Natural Language to SQL capabilities (NL2SQL)
- Enrich model with:
  - Example queries
  - Business context

Example:

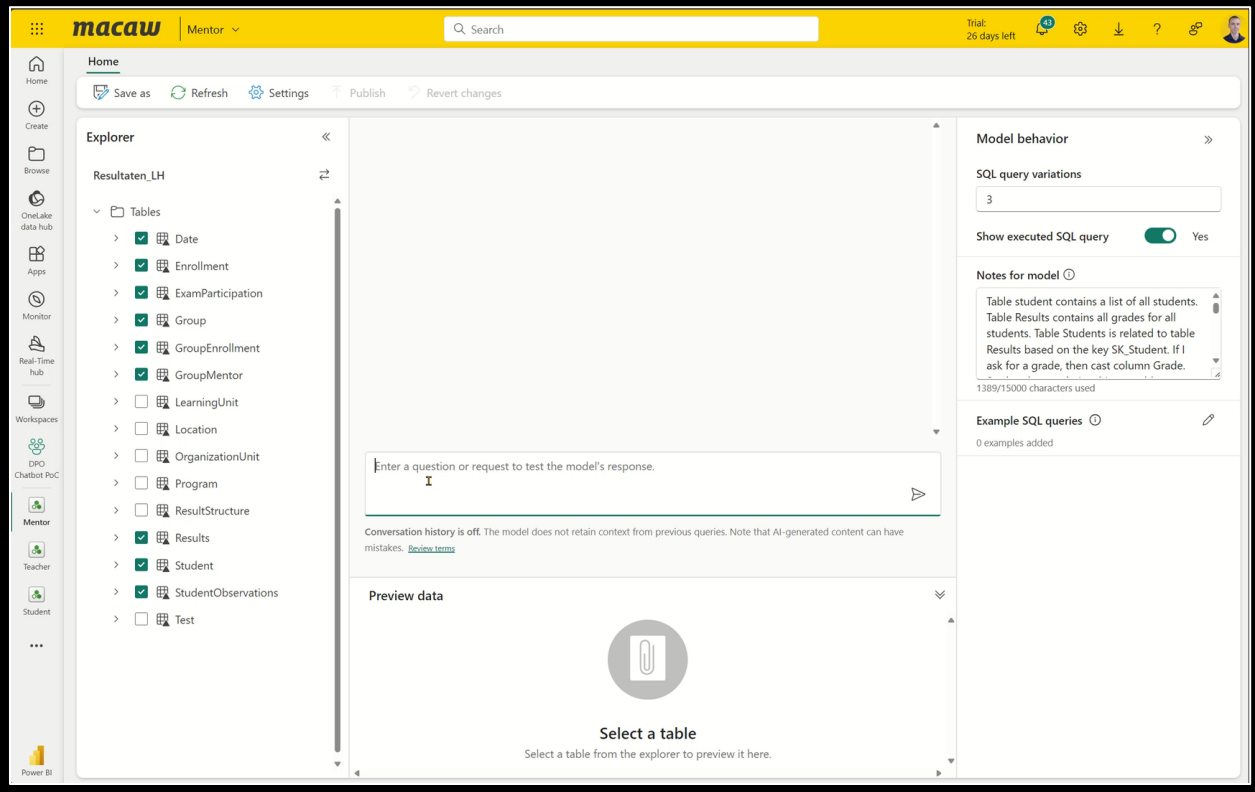
EduChat



EduChat in action

AI Skills in Fabric

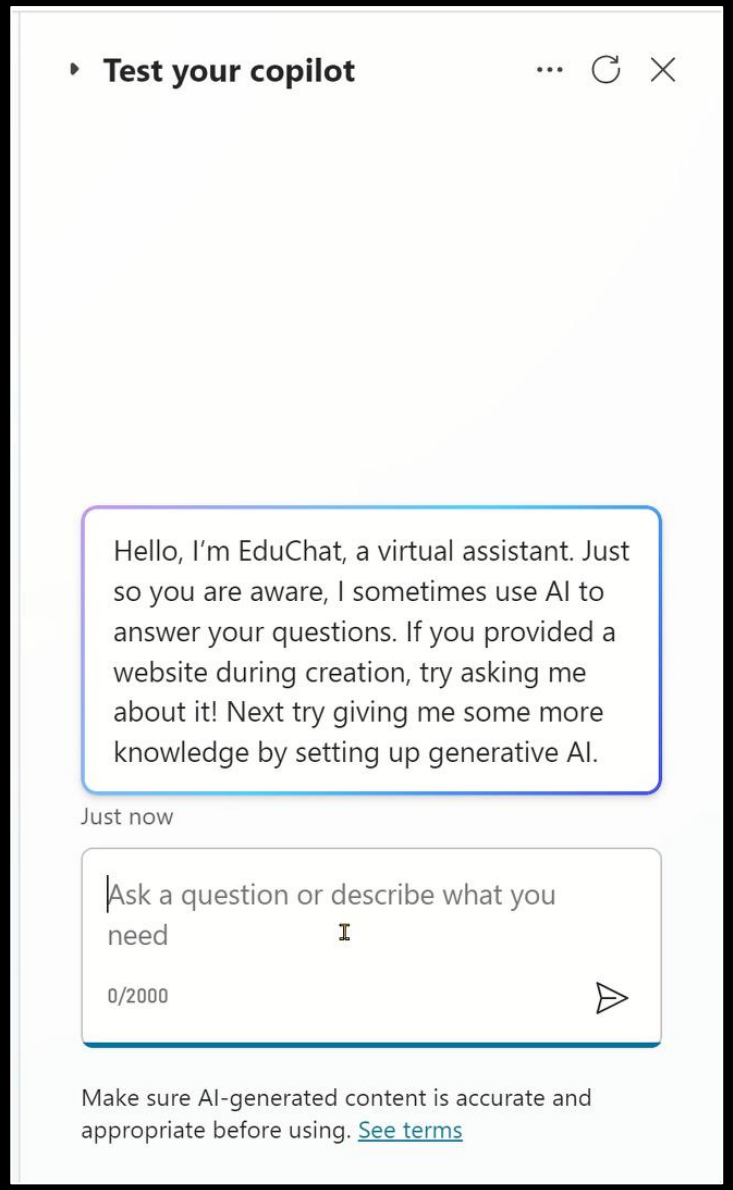
As a teacher, I want to find out what the average grade was, of the test students made last week.



EduChat in action

CoPilot Studio integration

As a teacher, I want to find out what the average grade was, of the test students made last week.



CI/CD & DTAP for Fabric



CI / CD

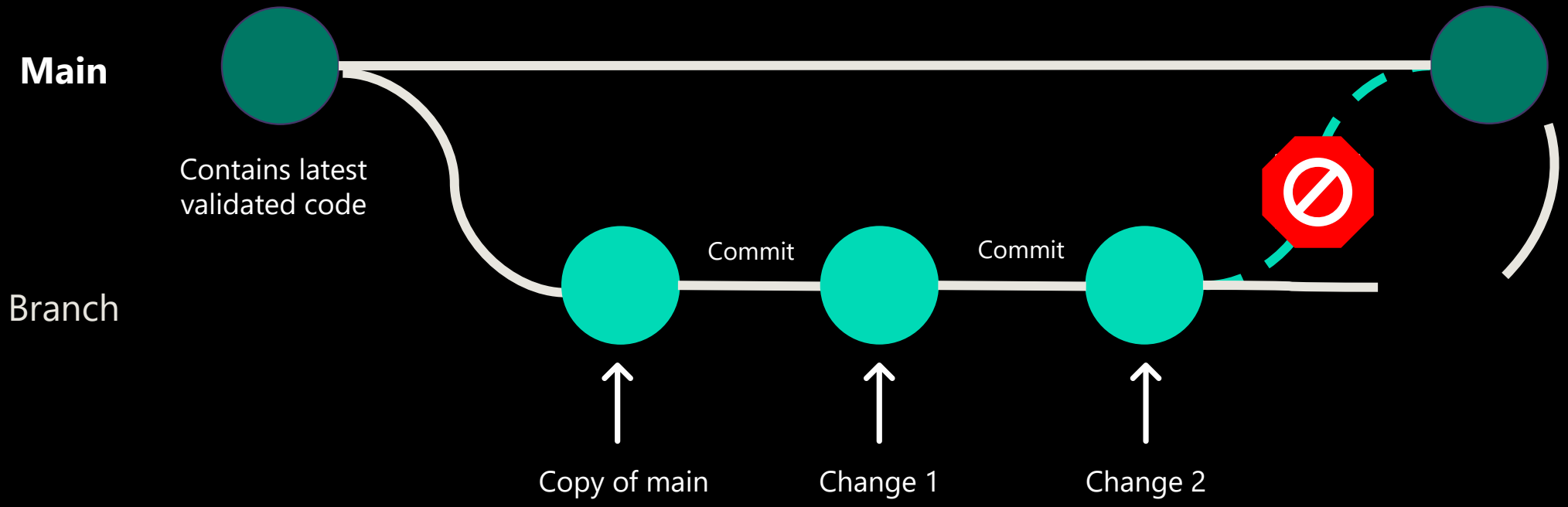
Continuous Integration

- Small but frequent changes
- Working with multiple people on various aspects of the solution
- Merging back into main branch

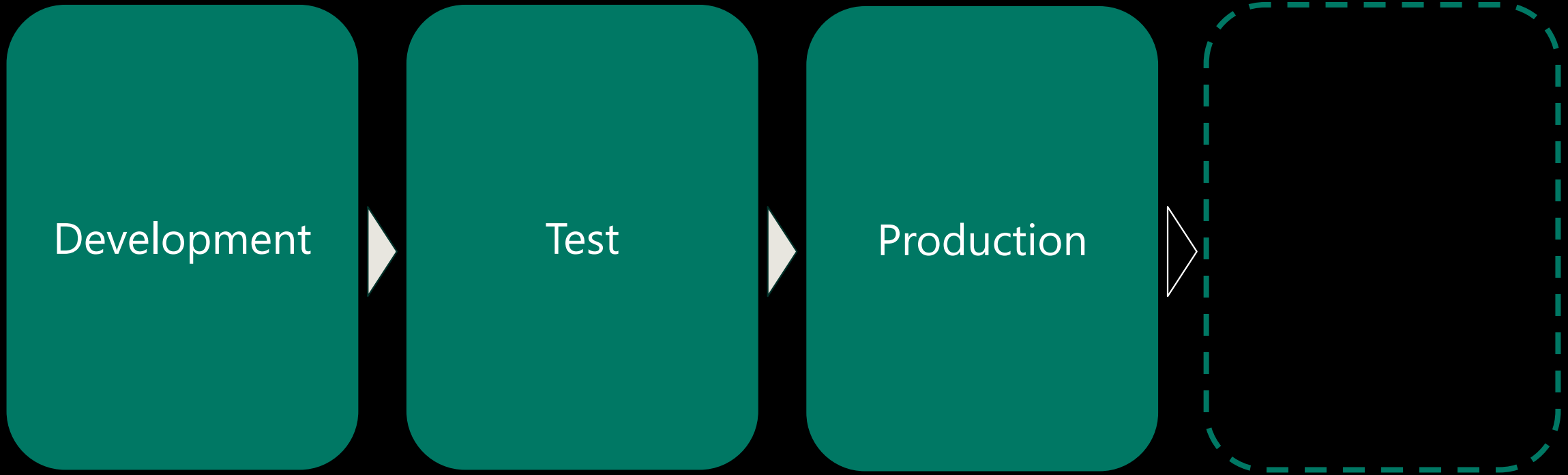
Continuous Deployment

- Working in short cycles (often Agile)
- High frequency updates and releases
- Repeatable deployment process

Git concept


Main
Contains latest validated code
Branch
Commit
Commit
Copy of main
Change 1
Change 2

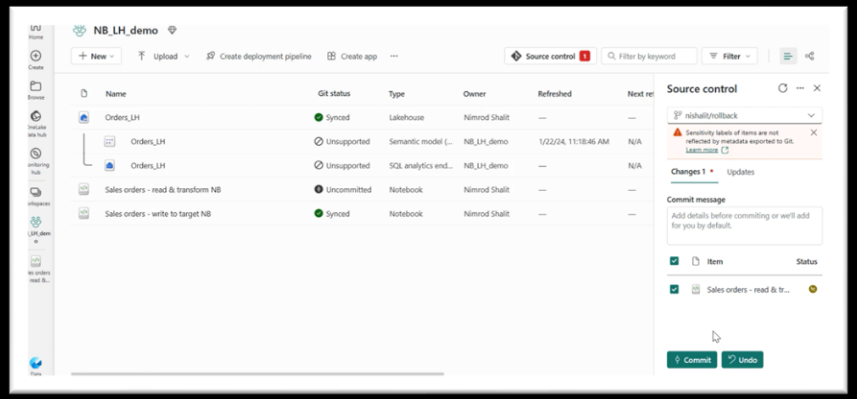
Staged approach



Fabric CI/CD platform

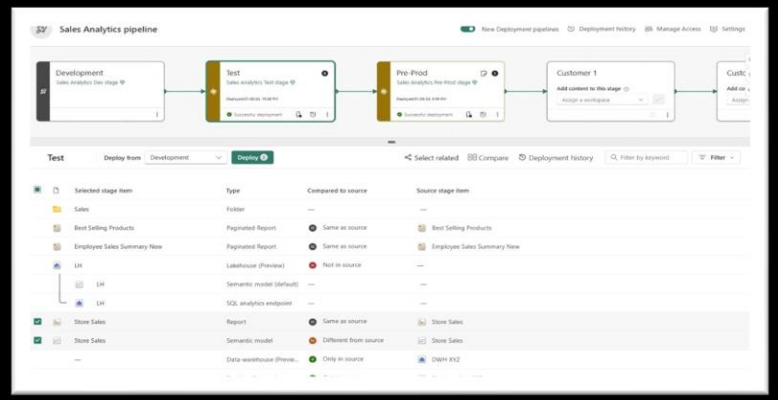
What do we have today?

Built-in git integration



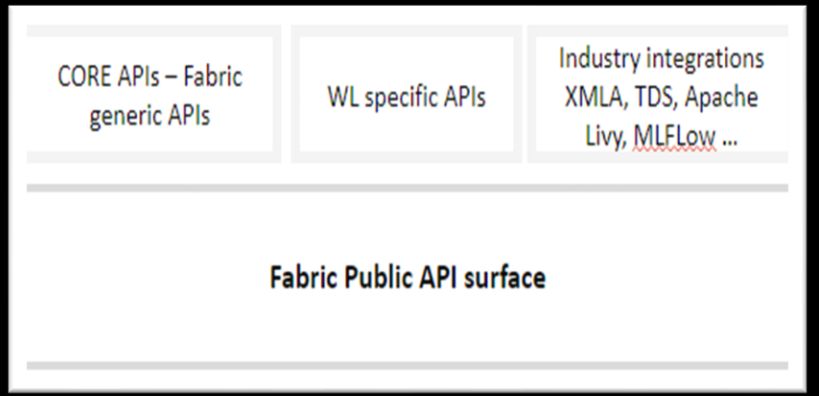
Public preview

Deployment pipelines



GA feature

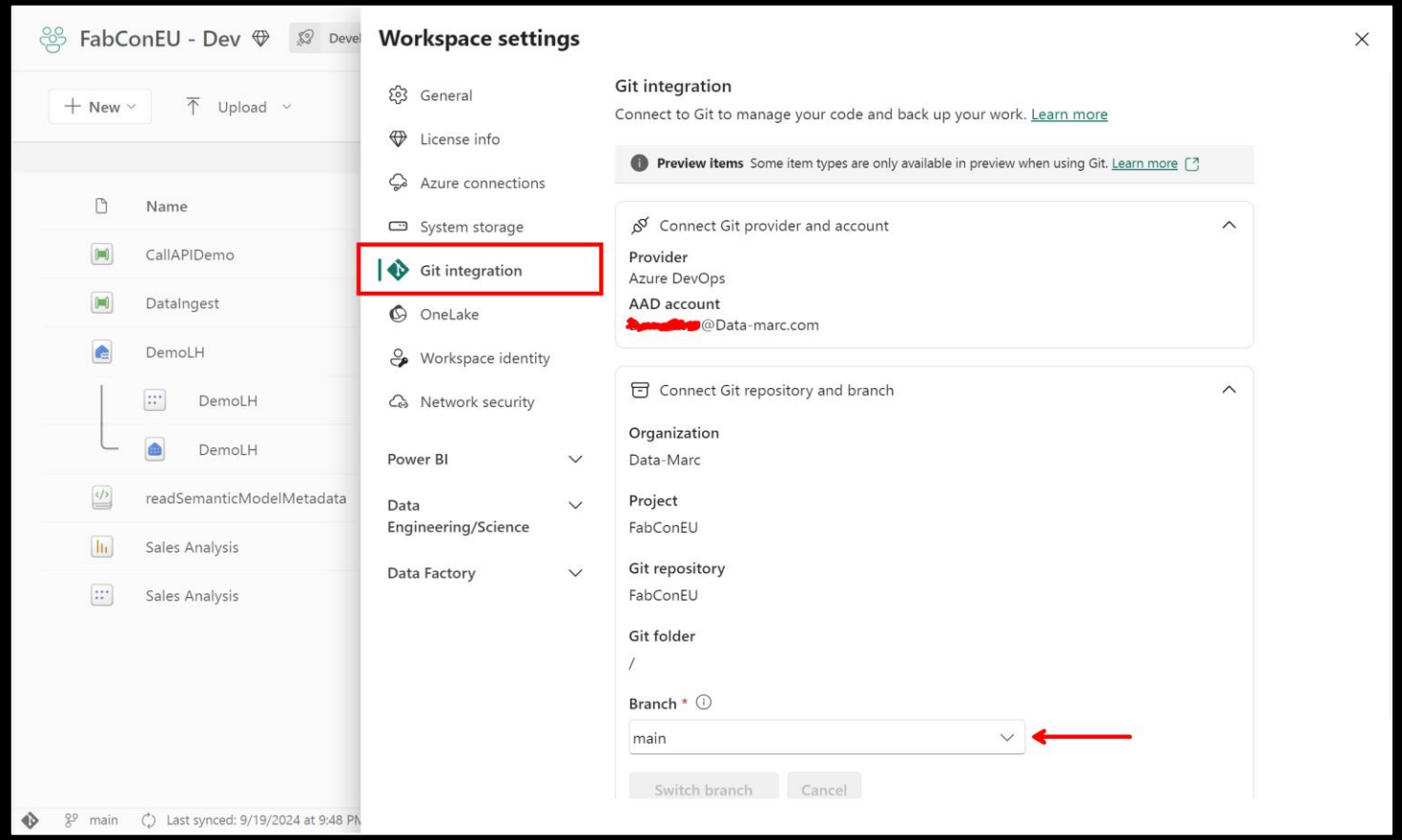
Fabric REST APIs



Public preview

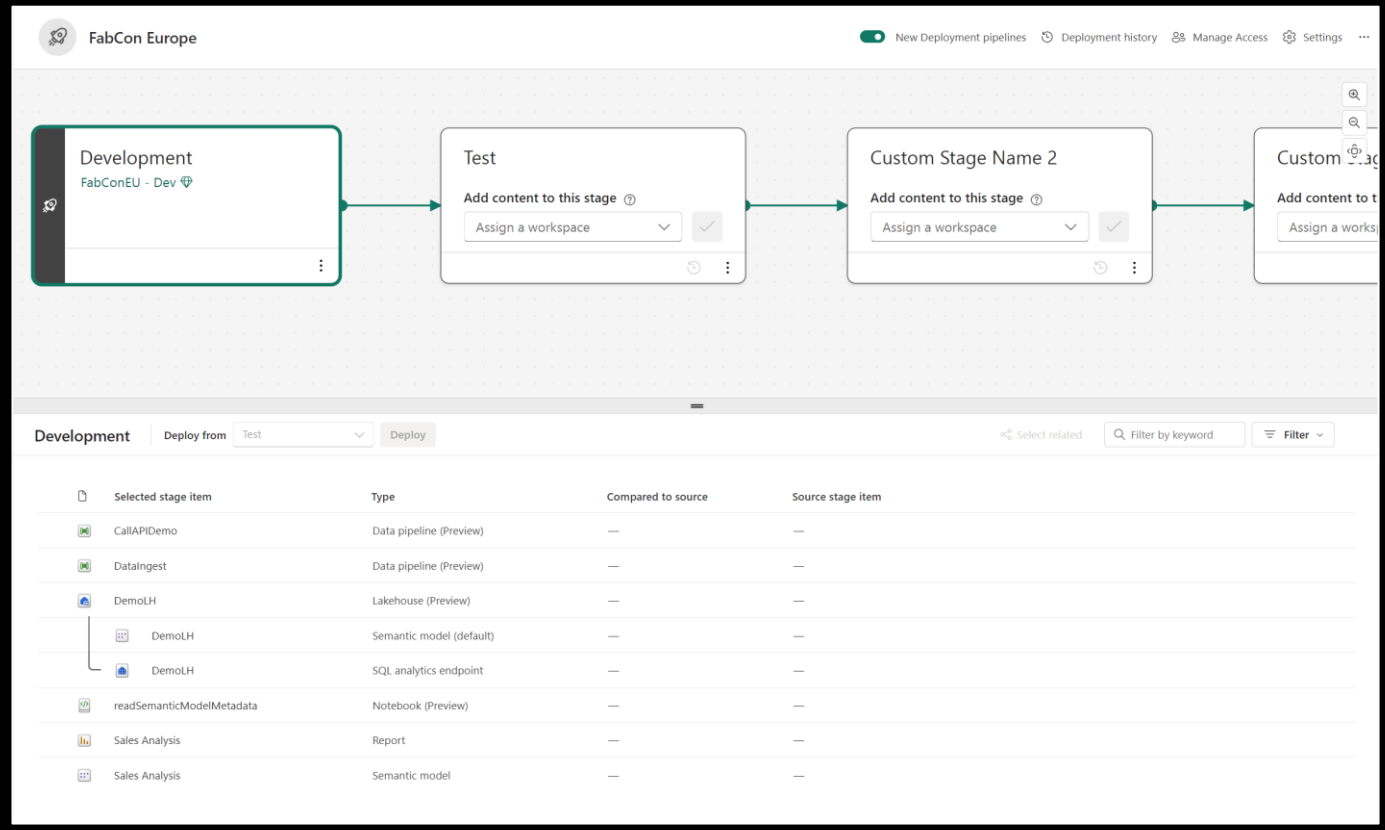
Git integration

- Sync a Workspace to a Git branch
- Git providers
  - Azure DevOps
  - GitHub New
  - GitHub Enterprise New
- Fabric git APIs – REST APIs & PowerShell samples.
- Manage branches
  - Switch branch
  - Checkout new branch
  - Branch out to new workspace



Deployment pipelines

- Deploy items across Workspaces
- Apply rules on configuration
- Majority of Fabric items supported *(and more to come)*
- Compare changes on code-level *(only for semantic models)*
- Create a pipeline of 2-10 stages
  - Pipeline designer at creation
  - Ability to add custom stage names



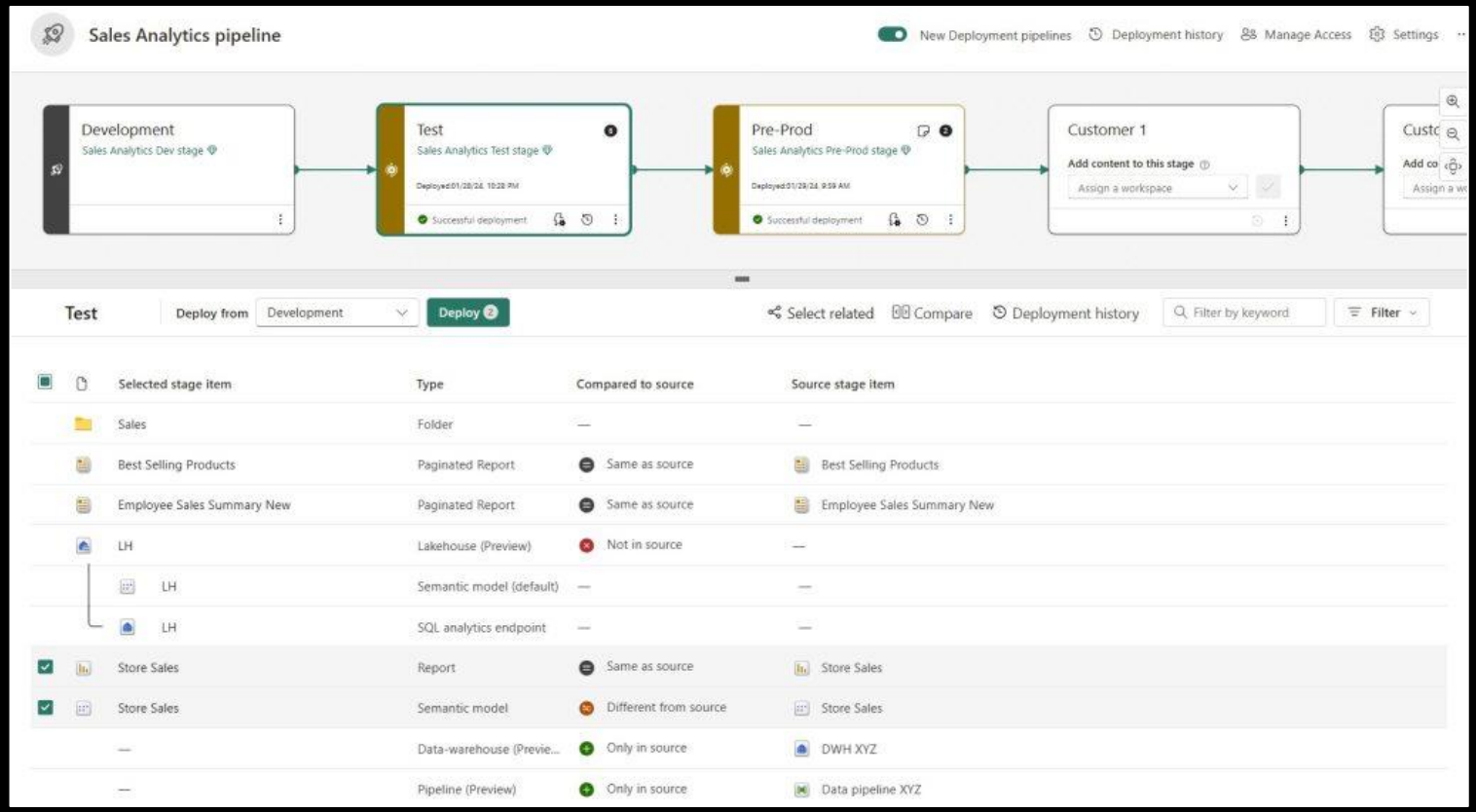
Fabric User APIs

- Automated operations on behalf of Fabric users.
- Supporting CRUD operations
- Example usage scenarios:
  - Item management *(see table)*
  - Item definition
  - Workspace management
  - Workspace access management
  - Execute item jobs

Item type	Create (without definition)	Get	Update	Delete	List
Dashboard	✗	✗	✗	✗	✓
DataPipeline	✗	✓	✓	✓	✓
Datamart	✗	✗	✗	✗	✓
Eventhouse	✓	✓	✓	✓	✓
Eventstream	✓	✓	✓	✓	✓
KQLDatabase	✗	✓	✓	✓	✓
KQLQueryset	✓	✓	✓	✓	✓
Lakehouse	✓	✓	✓	✓	✓
MLExperiment	✓	✓	✓	✓	✓
MLModel	✓	✓	✓	✓	✓
MirroredWarehouse	✗	✗	✗	✗	✓
Notebook	✓	✓	✓	✓	✓
PaginatedReport	✗	✗	✗	✗	✓
Report	✗	✓	✗	✓	✓
SemanticModel	✗	✓	✗	✓	✓
SparkJobDefinition	✓	✓	✓	✓	✓
SQLEndpoint	✗	✗	✗	✗	✓
Warehouse	✓	✓	✓	✓	✓

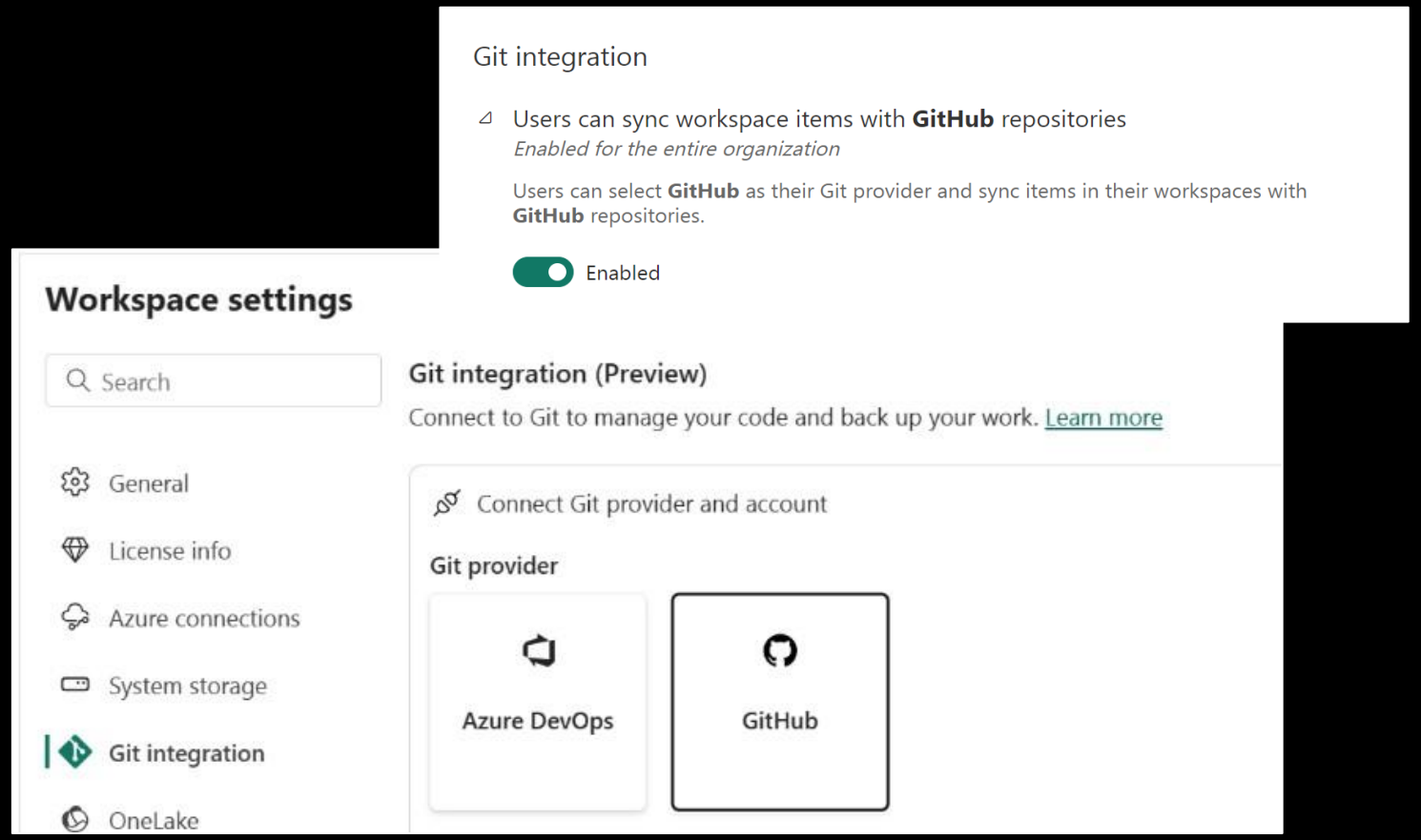
New UI for Deployment Pipelines

- Switch to enable/disable UI
- Easier navigate
- More focused (per stage)
- Smoother flow
- Folder structure
- Identify unsupported items



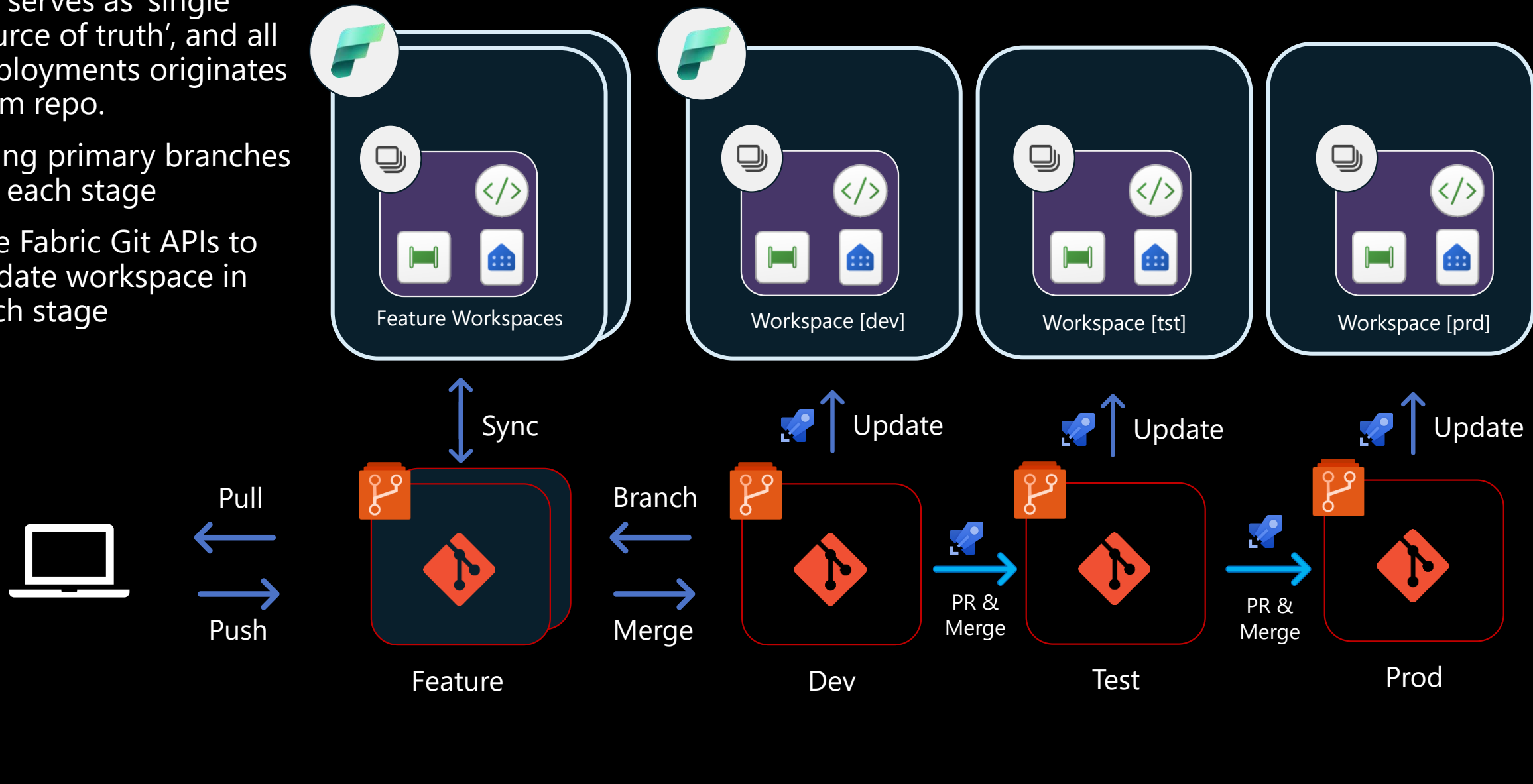
GitHub integration

- Second git provider next to Azure DevOps
  - GitHub + GitHub Enterprise
- Tenant admin explicitly must activate the feature
- Potential multi-geo restrictions not enforced
- API support coming soon



Scenario 1 – Git based deployments

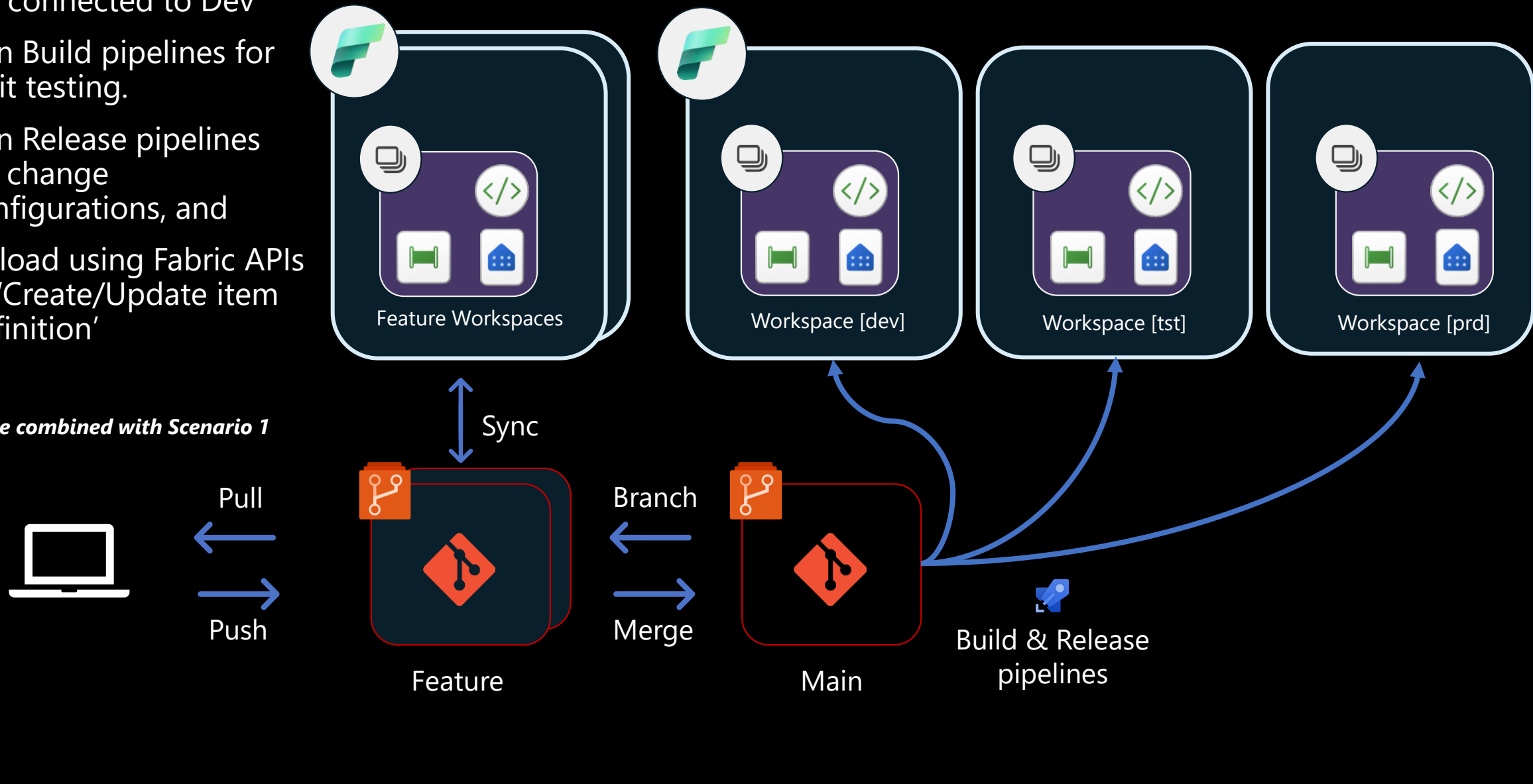
- Git serves as ‘single source of truth’, and all deployments originates from repo.
- Using primary branches for each stage
- Use Fabric Git APIs to update workspace in each stage



Scenario 2* – Git & Build environments

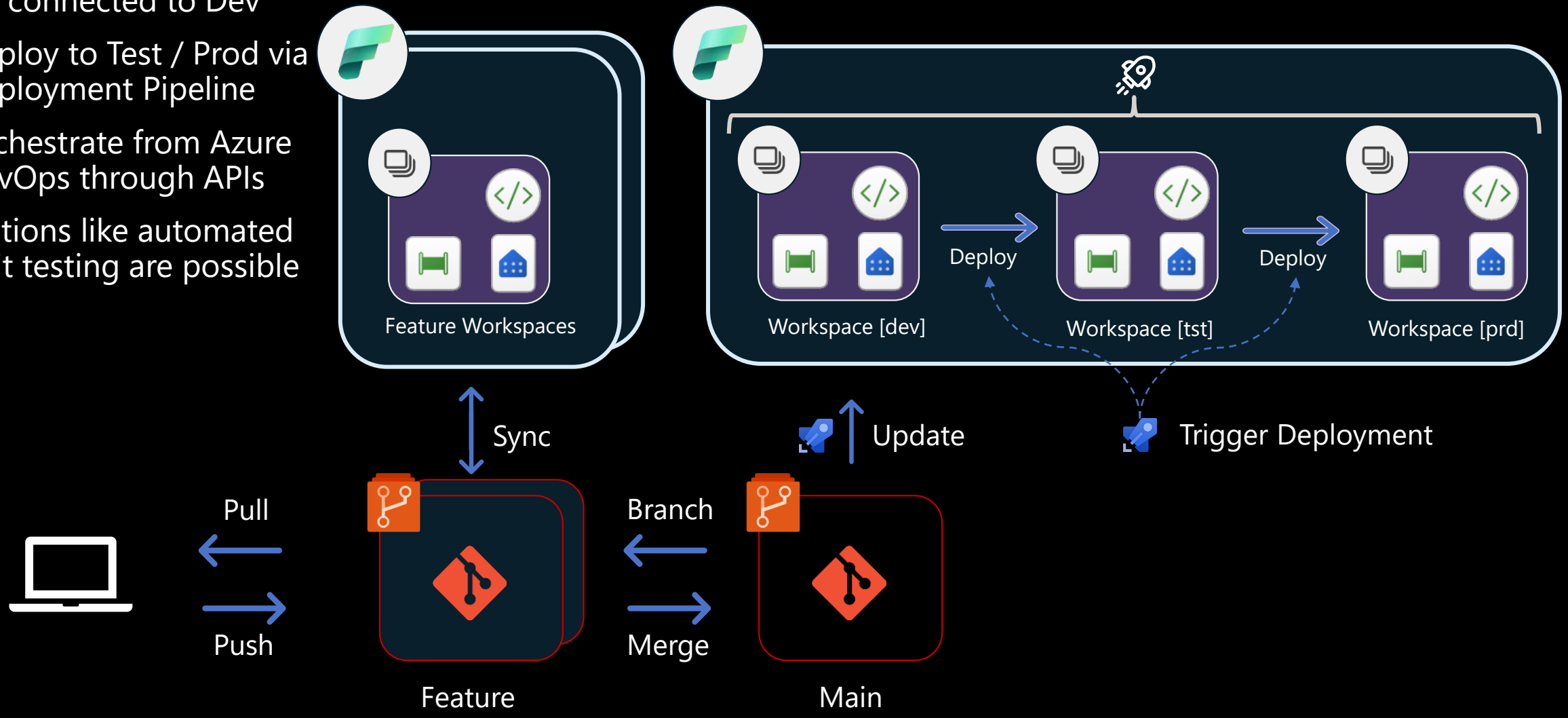
- Git connected to Dev
- Run Build pipelines for Unit testing.
- Run Release pipelines for change configurations, and
- Upload using Fabric APIs to ‘Create/Update item definition’

**** Can be combined with Scenario 1***



Scenario 3 – Git & Deployment pipelines

- Git connected to Dev
- Deploy to Test / Prod via Deployment Pipeline
- Orchestrate from Azure DevOps through APIs
- Options like automated unit testing are possible

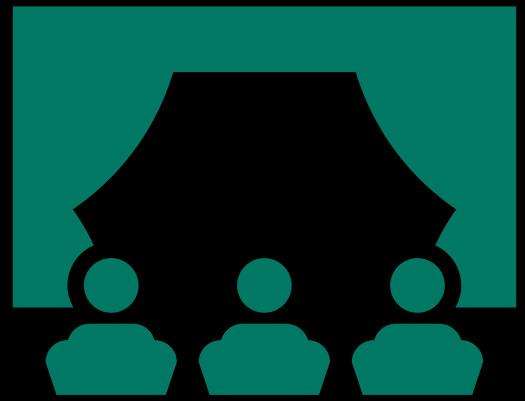


Opinionated view!

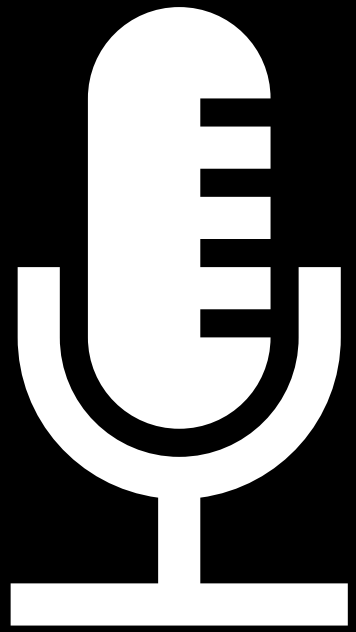
Entry point	Scenario	Code heaviness
Prefer an user interface?	Start with deployment pipelines interface, grow into **scenario 3** over time	Low
Comfortable working with git/code?	Your go-to scenario will be **scenario 1**.	Medium
Require a lot of customization?	Using the APIs in **scenario 2** allows you to customize everything to your wish	High

In greenfield scenario – opt for scenario 1!

DEMO TIME!



What else do you want to see/discuss?



Embark on a transformative journey into building an end-to-end solution within the **Microsoft Fabric**.



Erwin de Kreuk



Marc Lelijveld

Rate this session

