

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



Marc Lelijveld

Technical Evangelist & Architect
At Macaw



@MarcLelijveld



linkedin.com/in/MarcLelijveld



Data-Marc.com



DutchFabricUsergroup.com



Erwin de Kreuk

Principal Consultant & Lead Data & AI
At InSpark



@ErwinDeKreuk



linkedin.com/in/erwindekreuk



Erwindekreuk.com



DutchFabricUsergroup.com



Thank you sponsors!



Rate Data Saturday Holland



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)

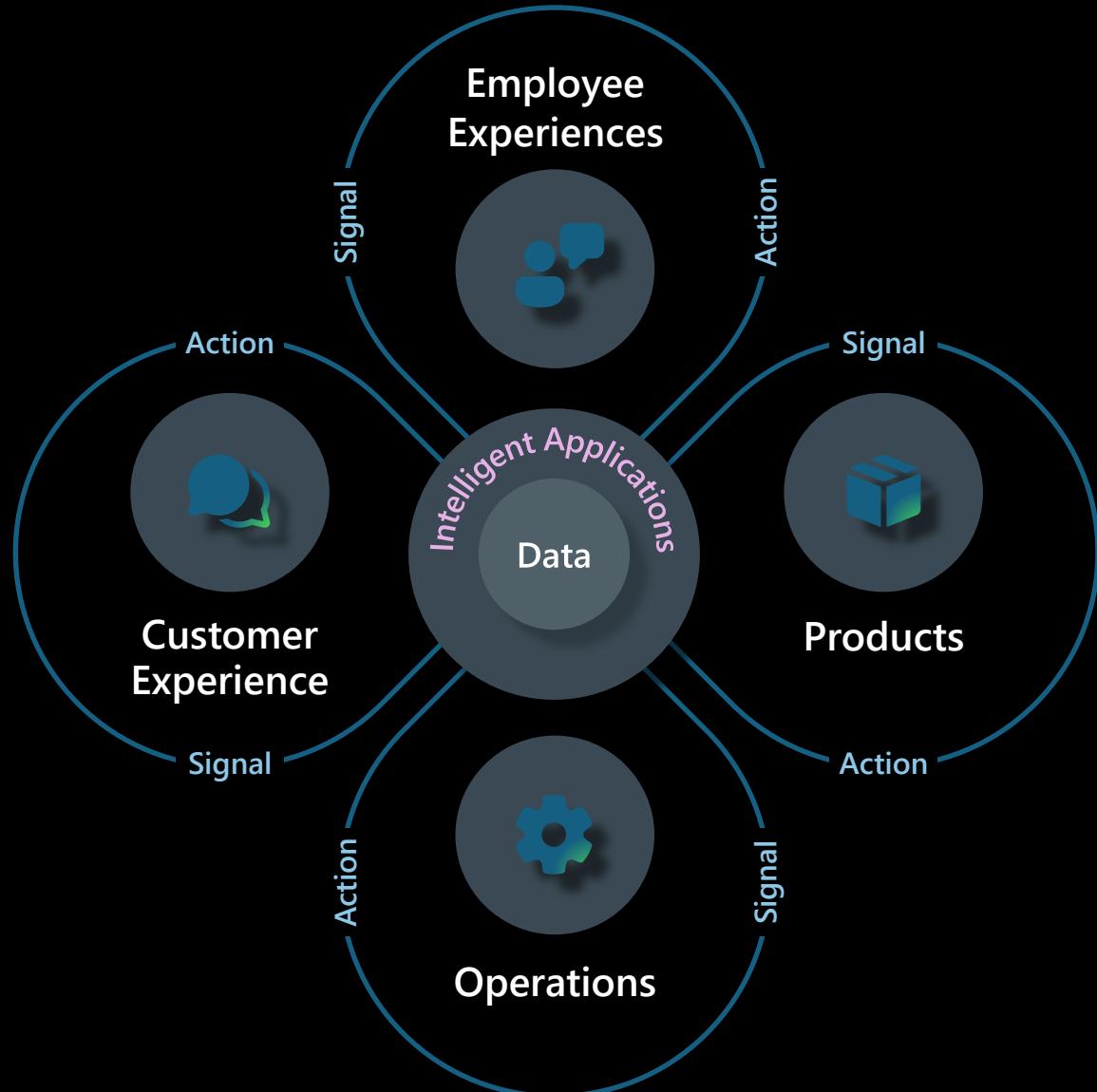
1

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

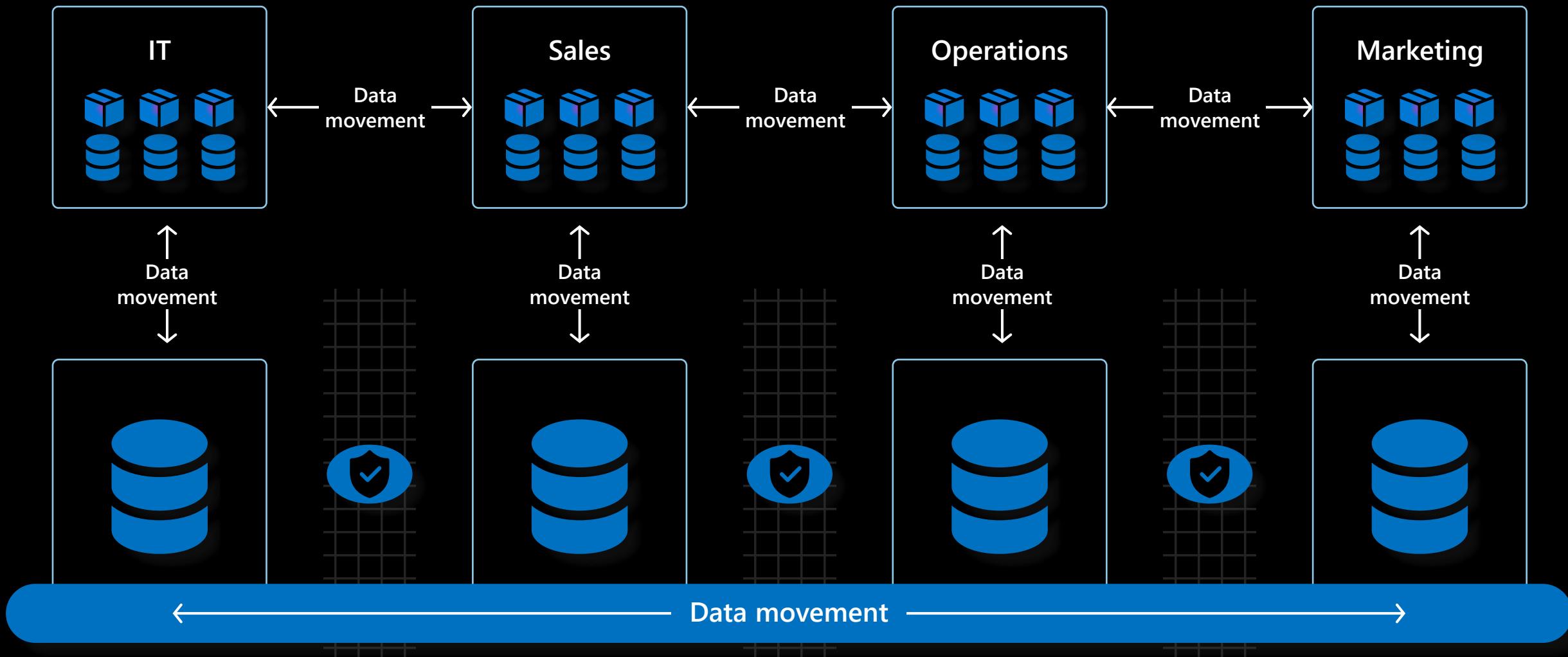


Gaining BI adoption to
streamline data sharing, collaboration



Data and security risks,
Governance & Compliance

Starting line: Multiple siloed lakes with lots of duplication



Today's data and analytics challenges

Limited scalability of legacy solutions as data demand rises exponentially



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



Delivering on the promise of analytics with limited resources



Microsoft Fabric

Data analytics for the era of AI

From

Isolated component

Single database

Gen AI Bolted-in

To

Unified stack

All the data

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

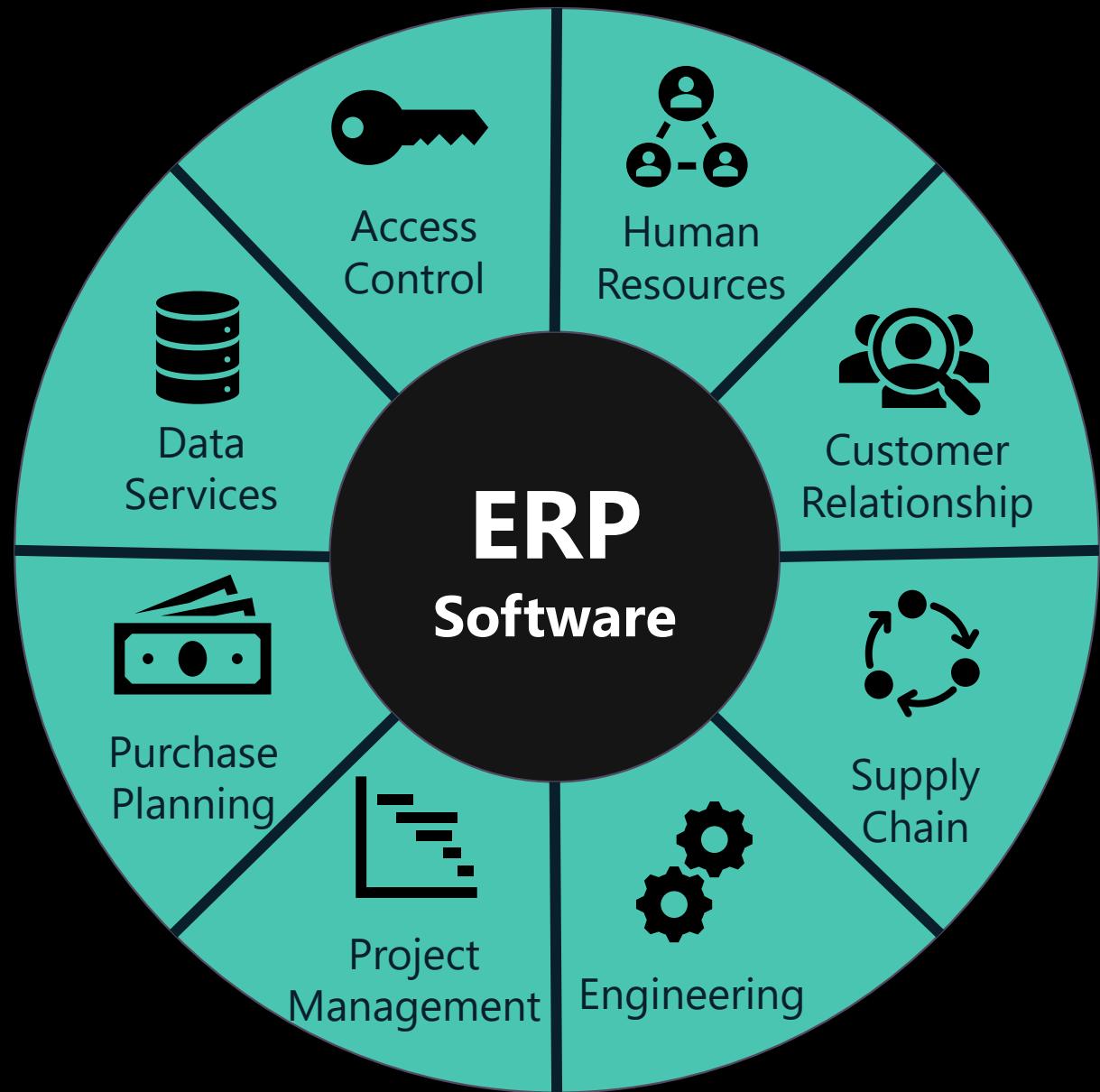
AI Powered

Copilot accelerated

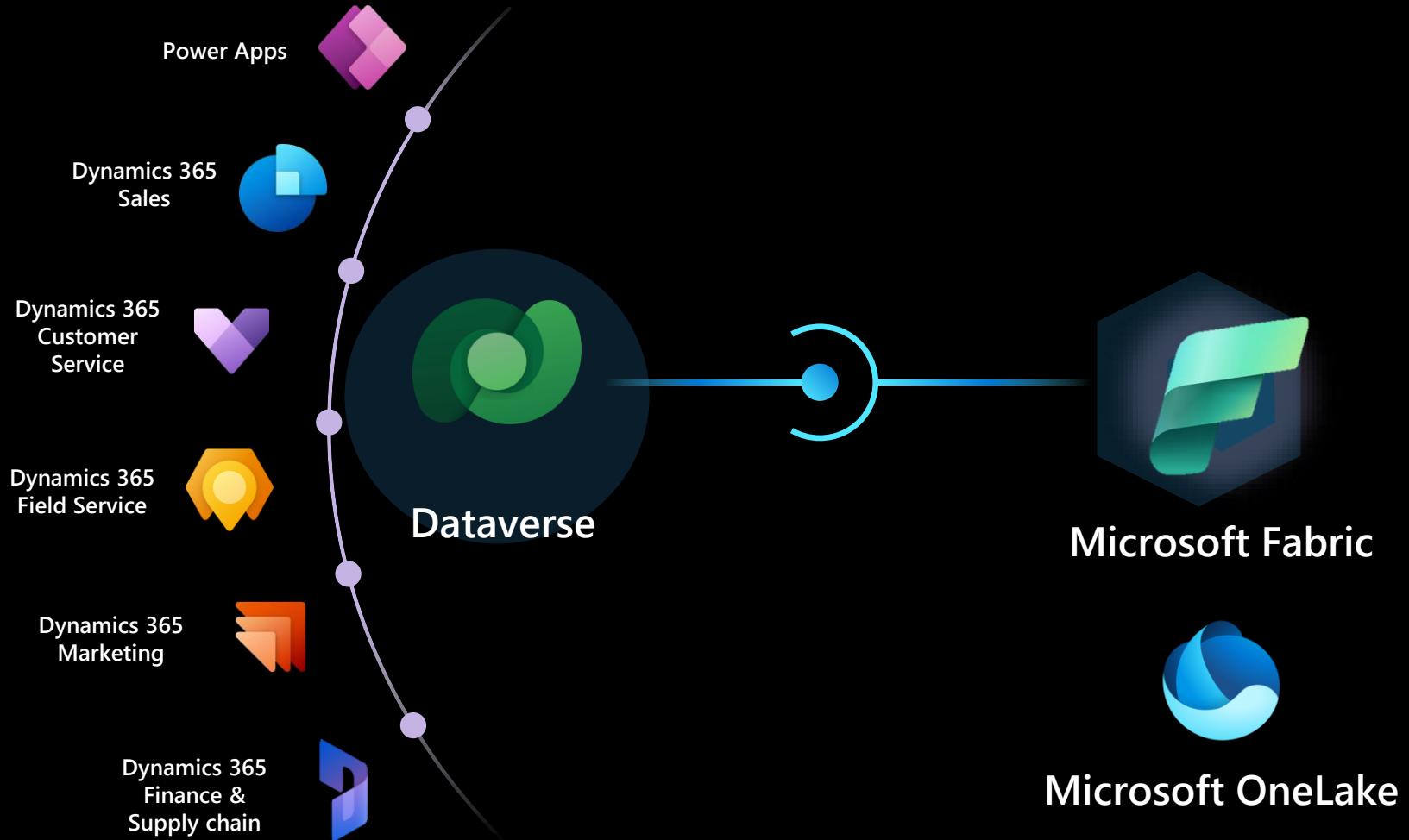
ChatGPT on your data

AI driven insights

The value in your ERP data



Bringing your data to Fabric



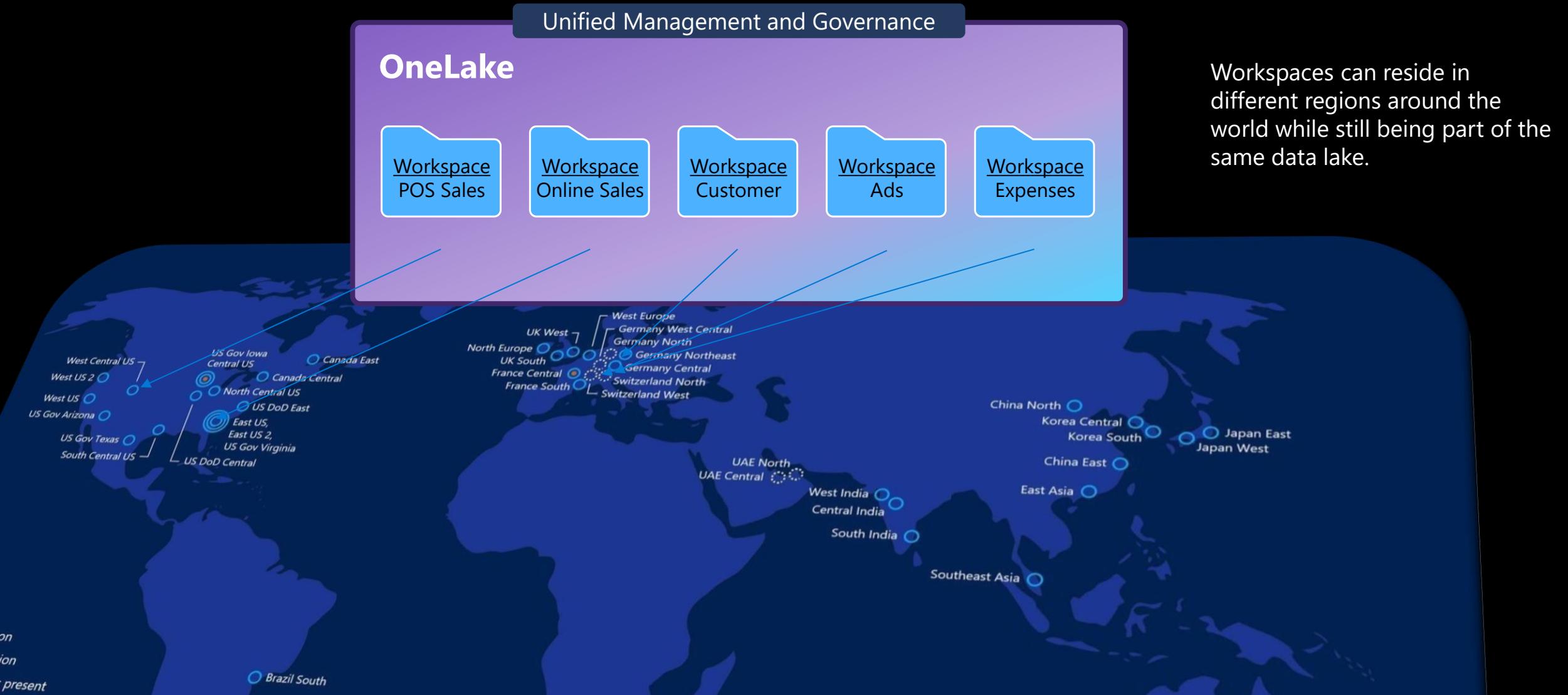


Any questions so far?

Onelake

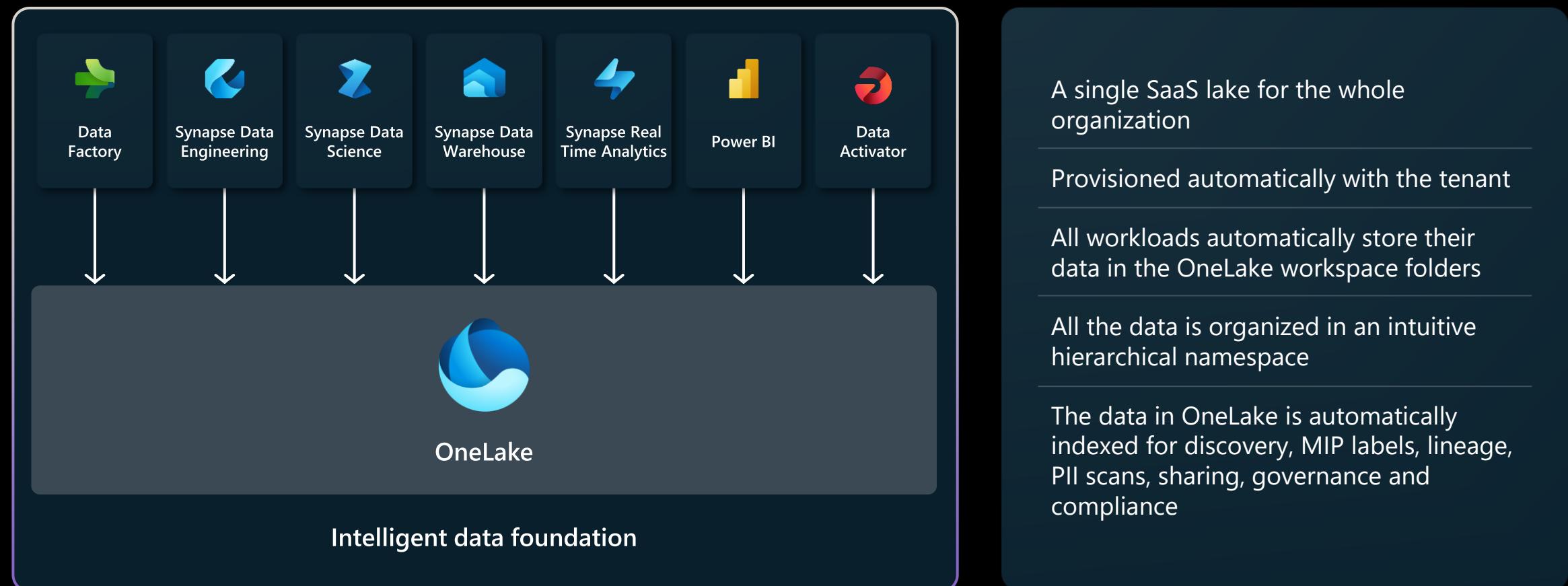
OneLake which logically spans the world

"The OneDrive for Data"



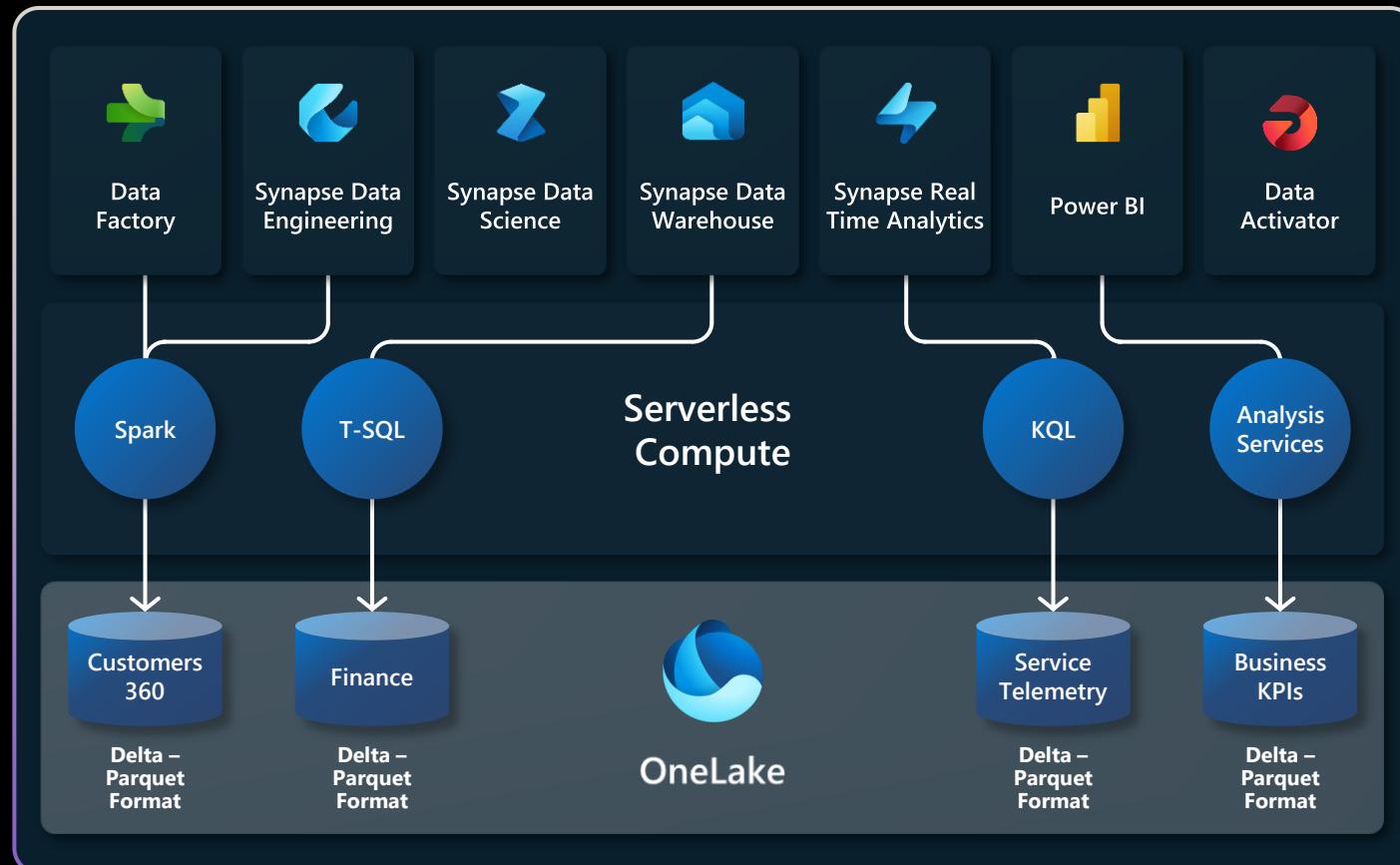
OneLake for all Data

“The OneDrive for Data”



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

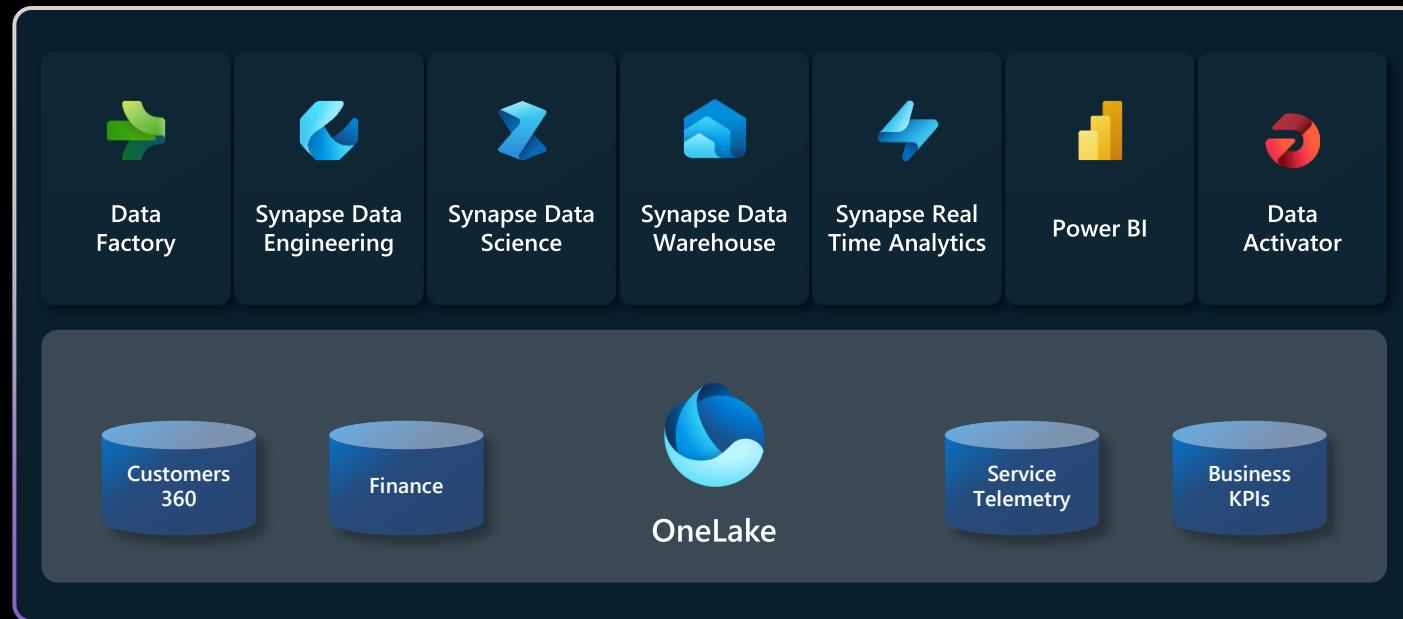
Delta – Parquet, an open standards format, is the storage format for all tabular data in Analytics vNext

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

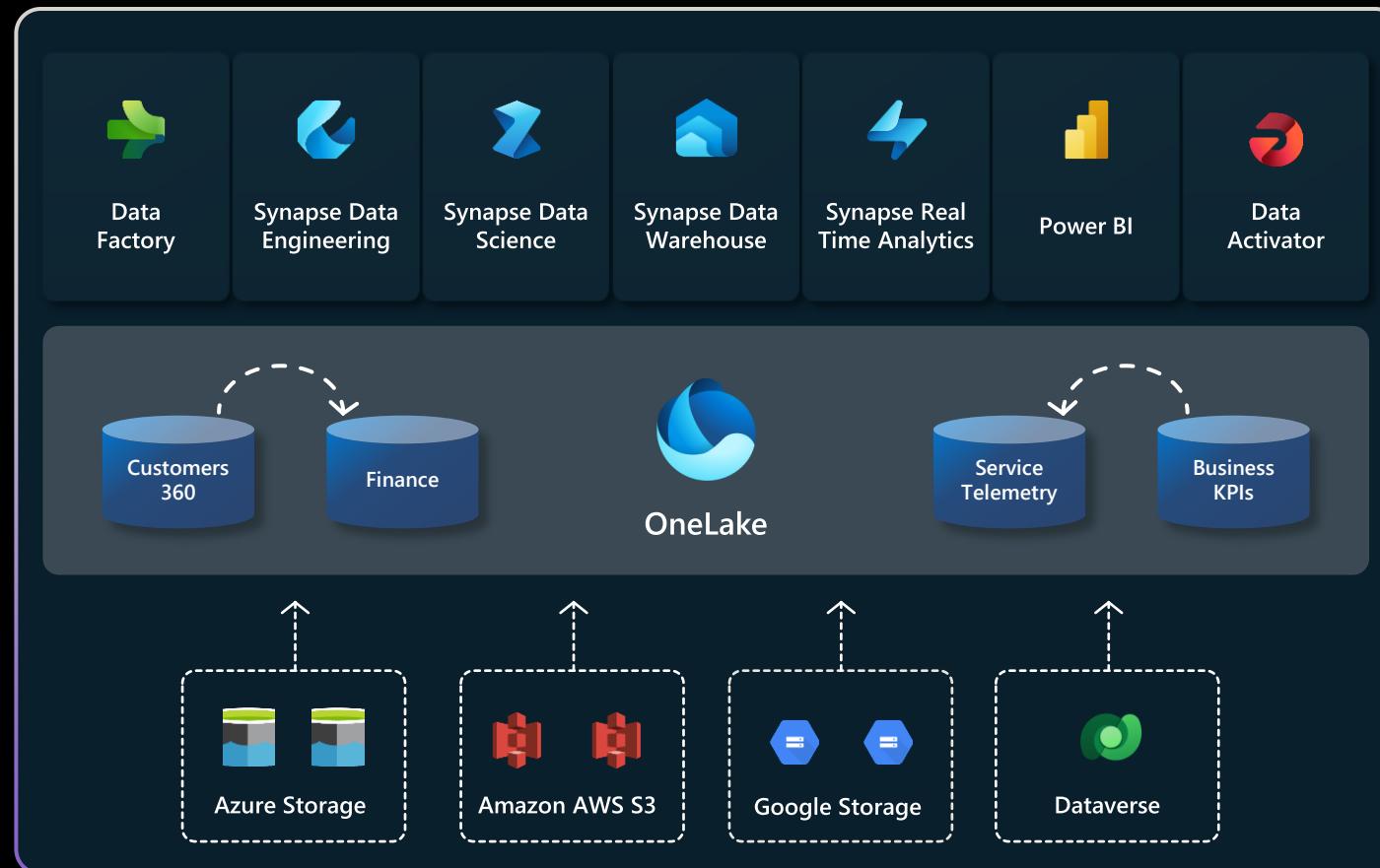
With shortcuts, data throughout OneLake can be composed together without any data movement

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

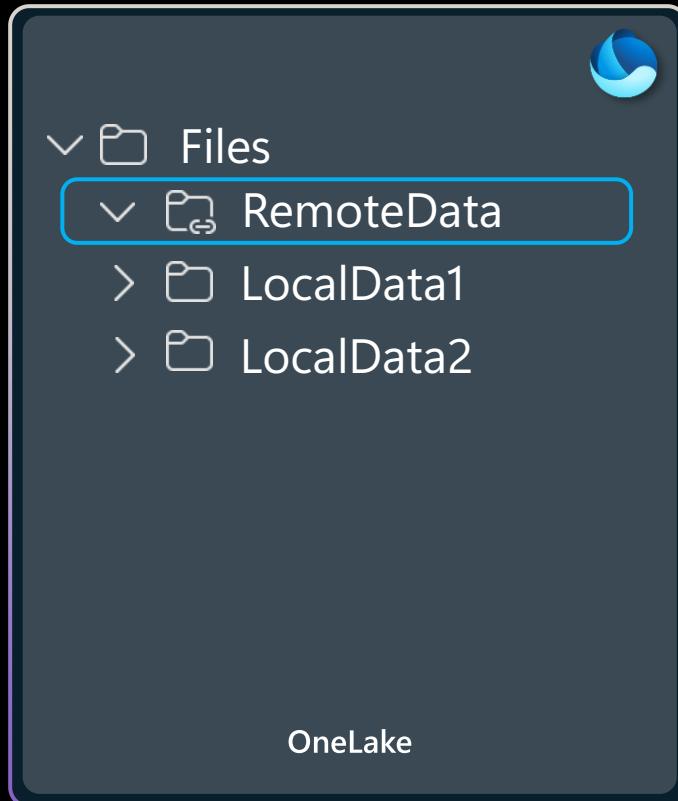
With shortcuts, data throughout OneLake can be composed together without any data movement

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

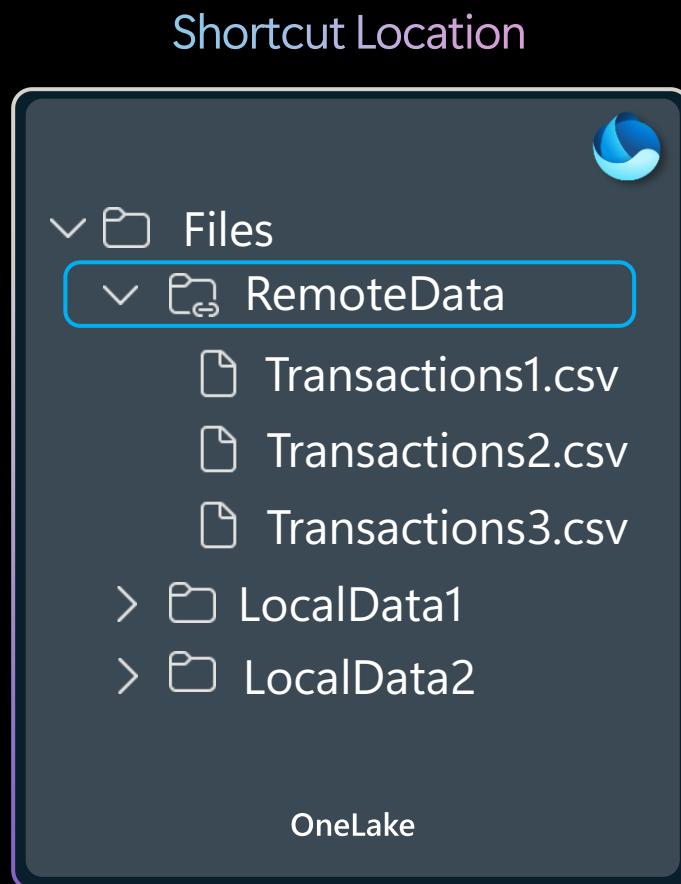
With support for industry standard APIs, OneLake data can be directly accessed by any application or service

What is a shortcut?

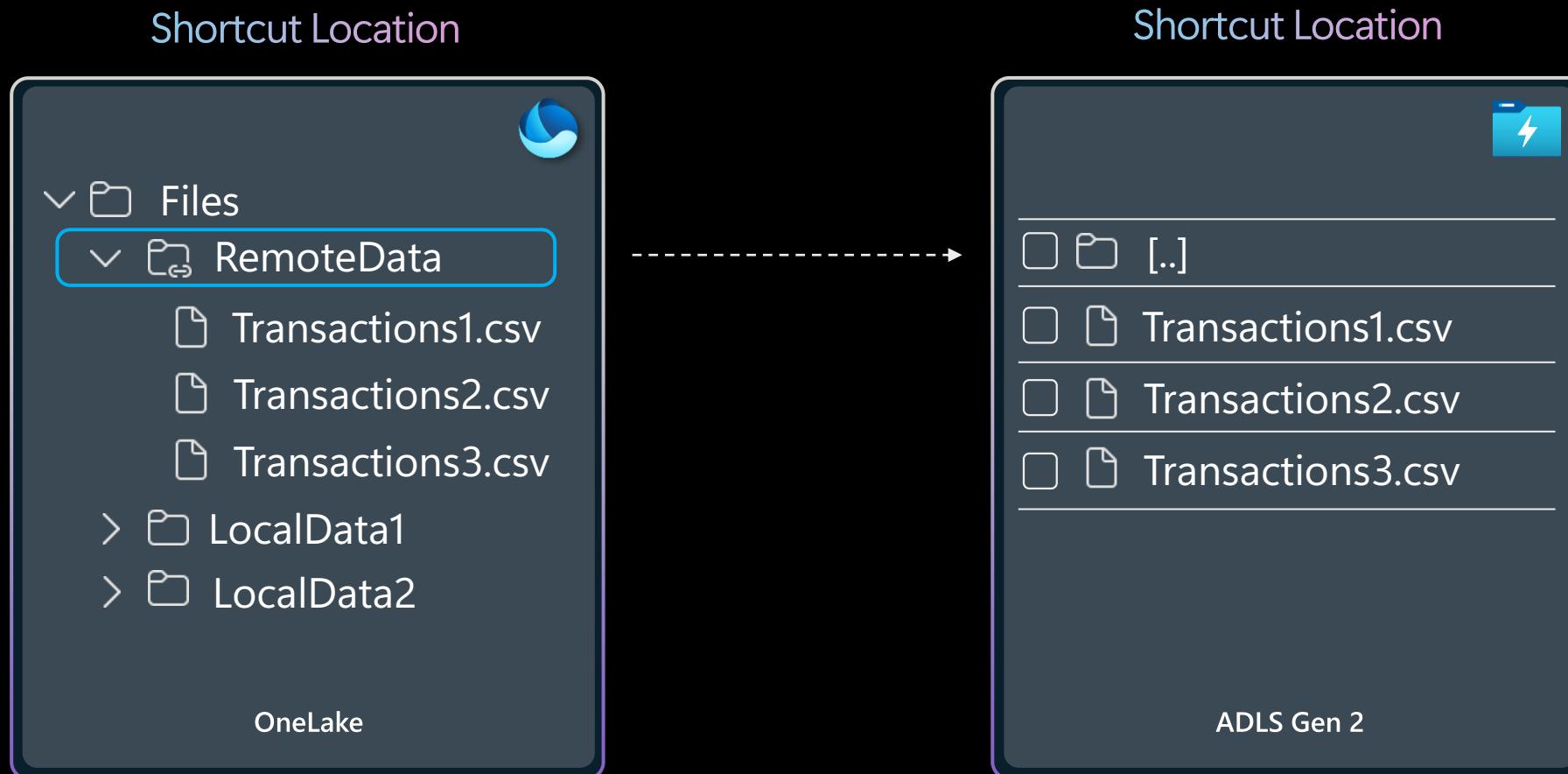
Shortcut Location



What is a shortcut?



What is a shortcut?

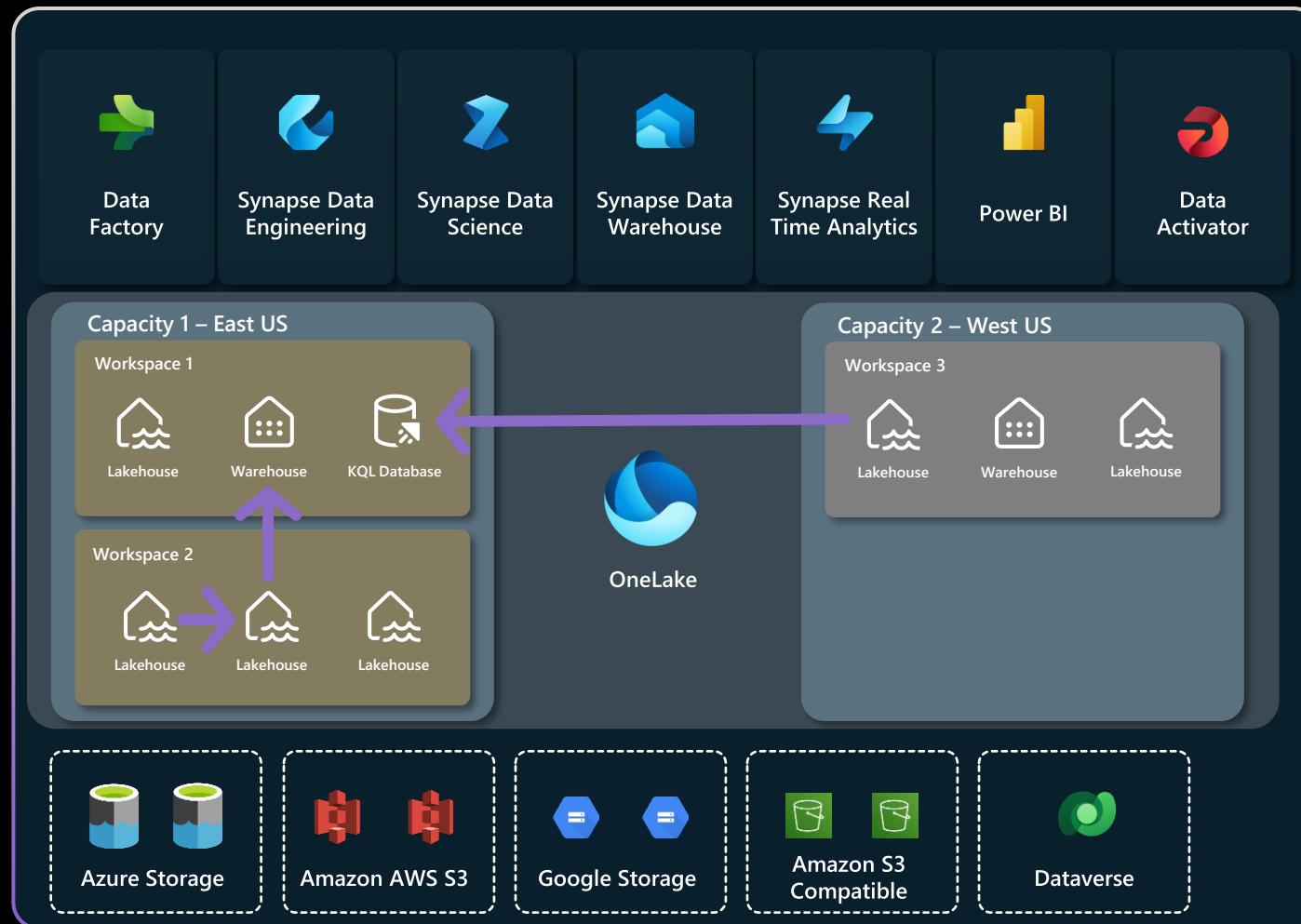




Any questions so far?

Unifying data in OneLake

Shortcuts



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

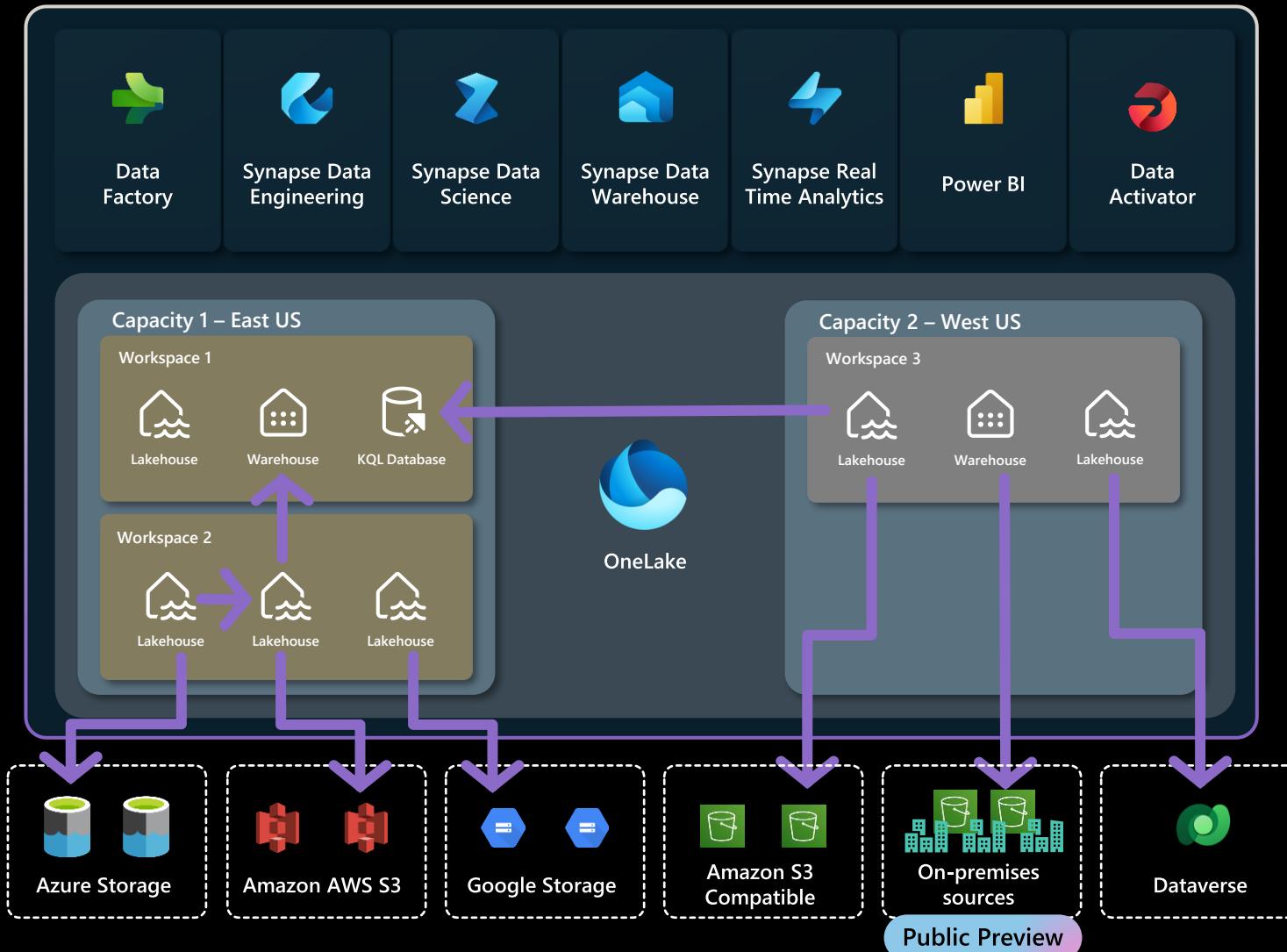
With shortcuts, data throughout OneLake can be composed together without any data movement

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

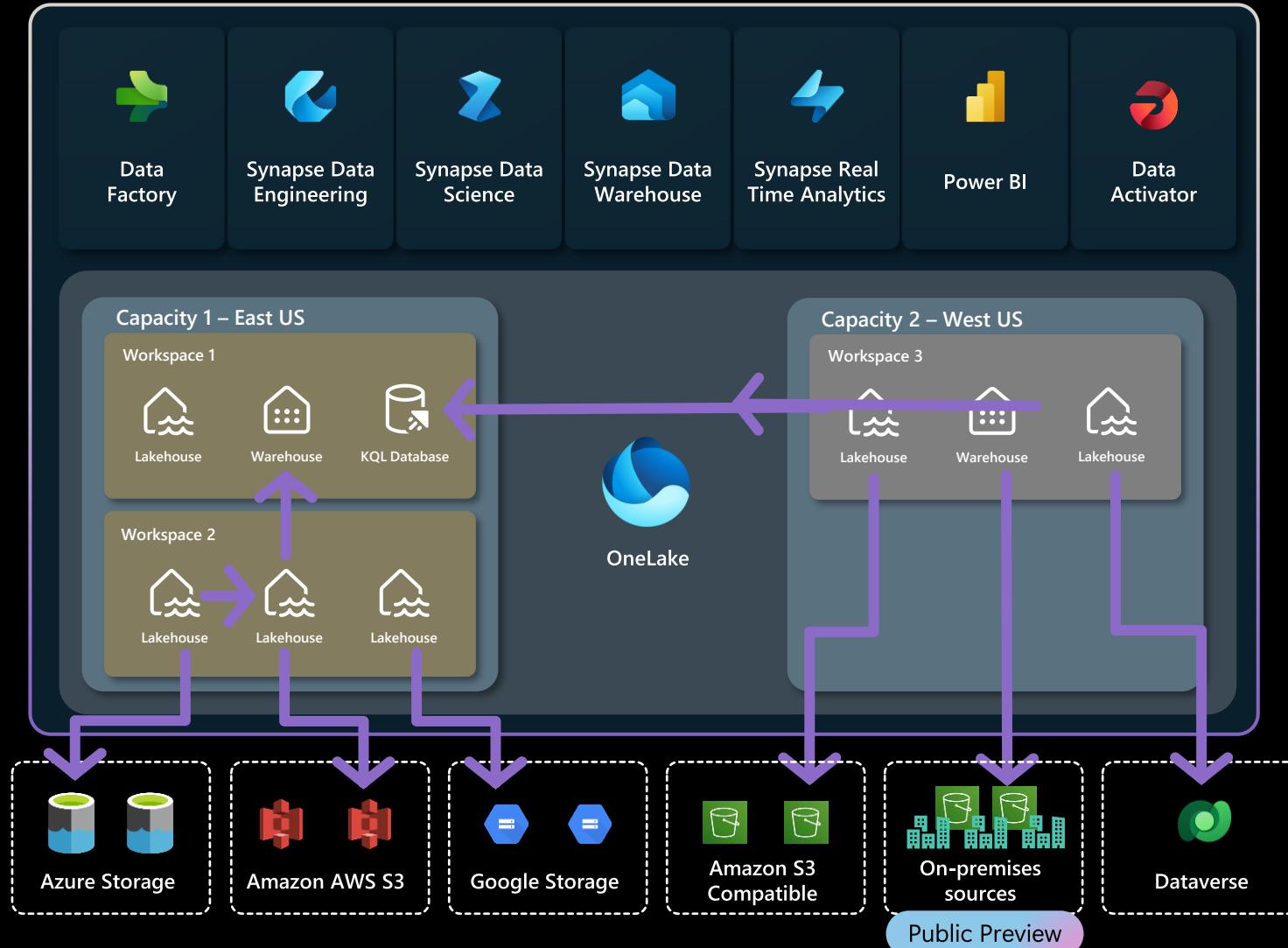
Unifying data in OneLake

Shortcuts



Unifying data in OneLake

External Data Sharing (Preview)



Unifying data in OneLake

External Data Sharing (Preview)

▫ External data sharing (preview)
Enabled for the entire organization

Users can share a read-only link to data stored in OneLake with collaborators outside your organization. When you grant them permission to do so, users can share a link to data in lakehouses and additional Fabric items. Collaborators who receive the link can view, build on, and share the data both within and beyond their own Fabric tenants, using their organization's licenses and capacities. [Learn More](#)



⚠️ Usage of shared data within a recipient tenant is not restricted by policies within your tenant, such as Conditional Access, Information Protection, Microsoft Cloud App Security, etc. Shared data may also be processed outside the region where it's stored.

▫ Users can accept external data shares (preview)
Enabled for the entire organization

Users can accept a read-only link to data from another organization's Fabric tenant. Users who accept an external share link can view, build on, and share the data, both inside and outside of your organization's tenant. For more information about the security limitations of this preview feature, view the feature documentation. [Learn More](#)



⚠️ Your organization is solely responsible for use of this feature. Potential risks include malicious intent on the part of the data provider, changes in the data at source, and legal liability arising from the accessing of regulated, proprietary, or sensitive data, responsibility for licensing enforcement, or the processing of data across regions.

Mirroring in Fabric

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



Azure Cosmos DB



Mongo DB



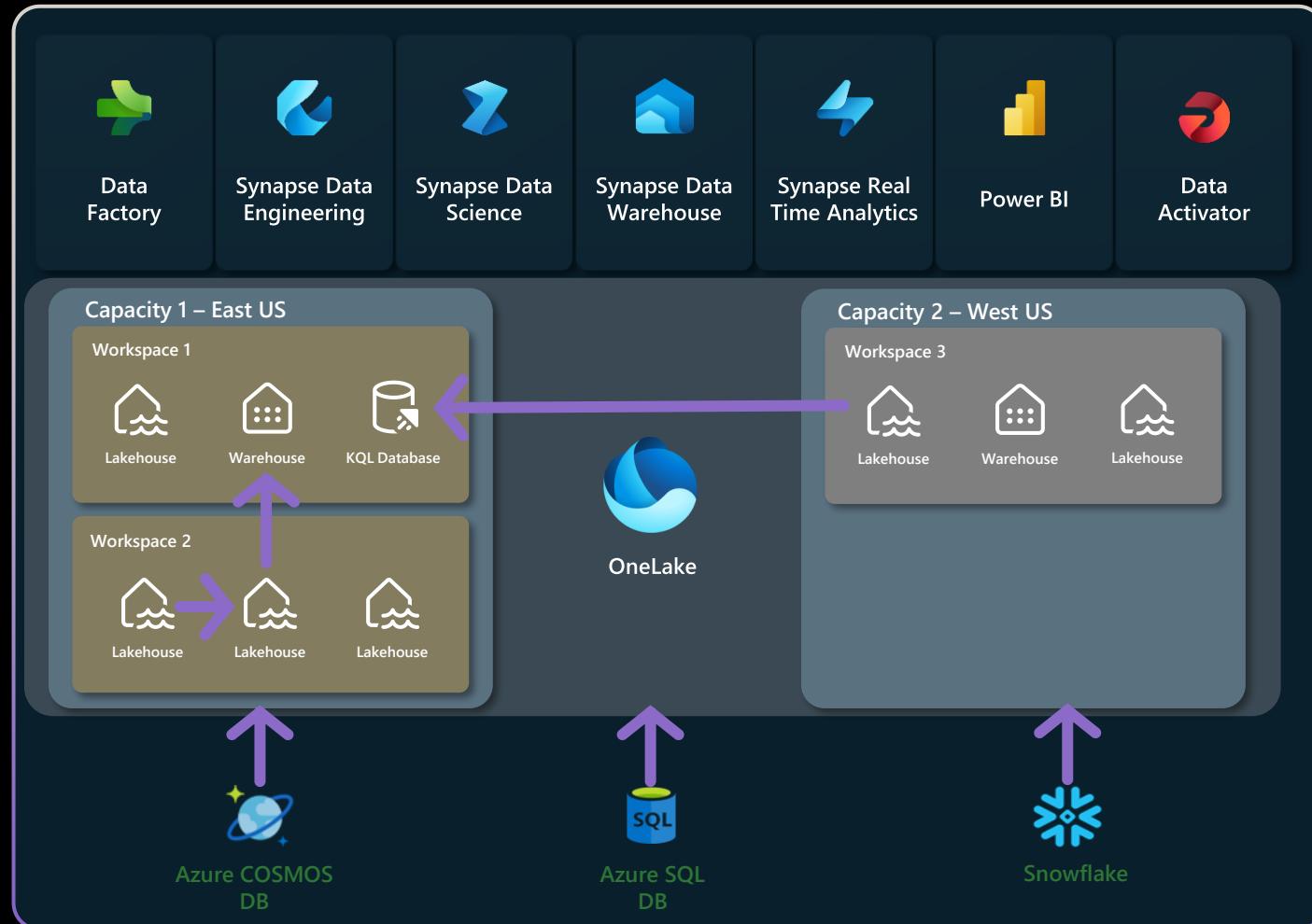
Azure SQL DB



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

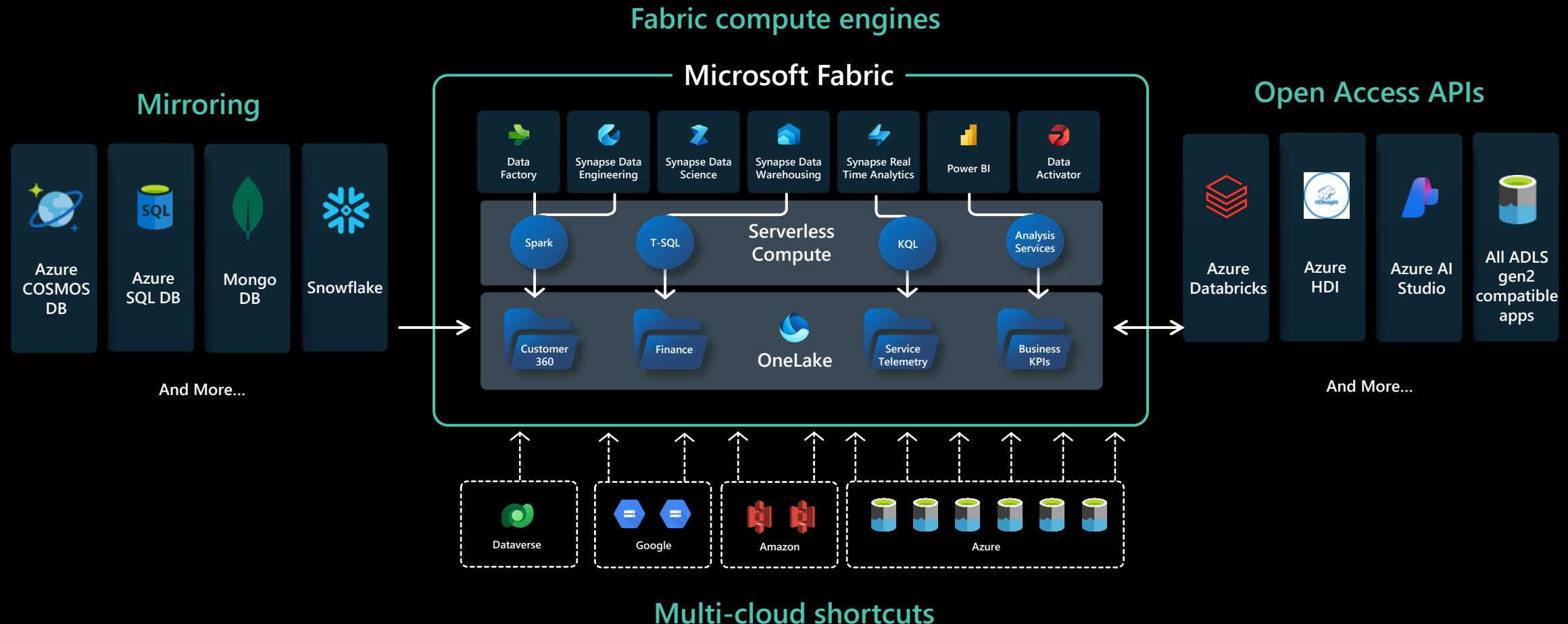
Analysts and Data Scientists can work with real-time data

The replica protects operational databases from analytical queries

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

All roads lead to OneLake

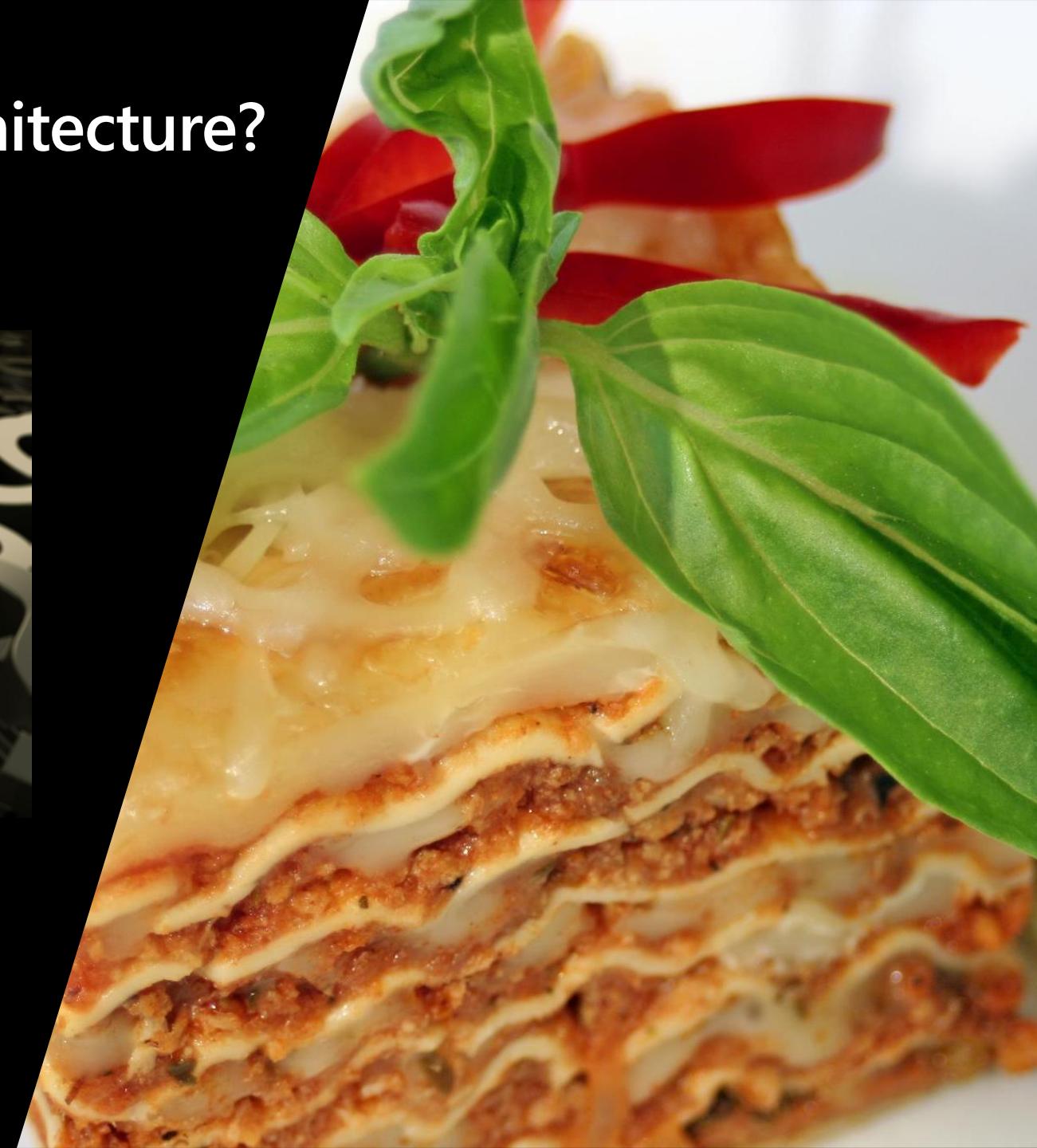
Creating Data Gravity in OneLake



Who is using a Medallion architecture?



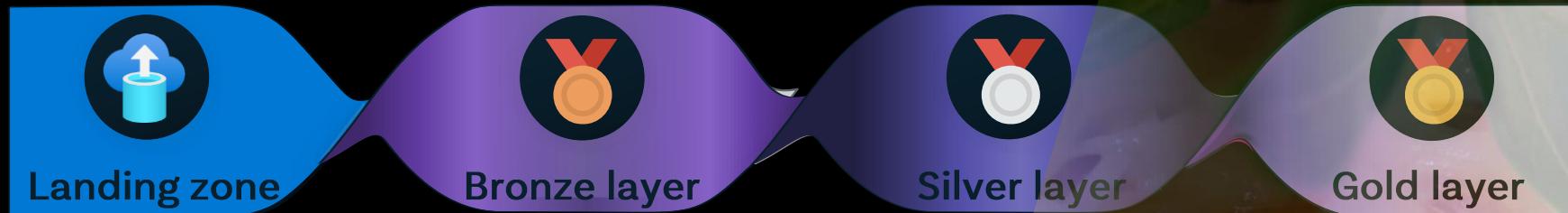
'Uniform data architecture'
From data "Spaghetti to Lasagna"



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



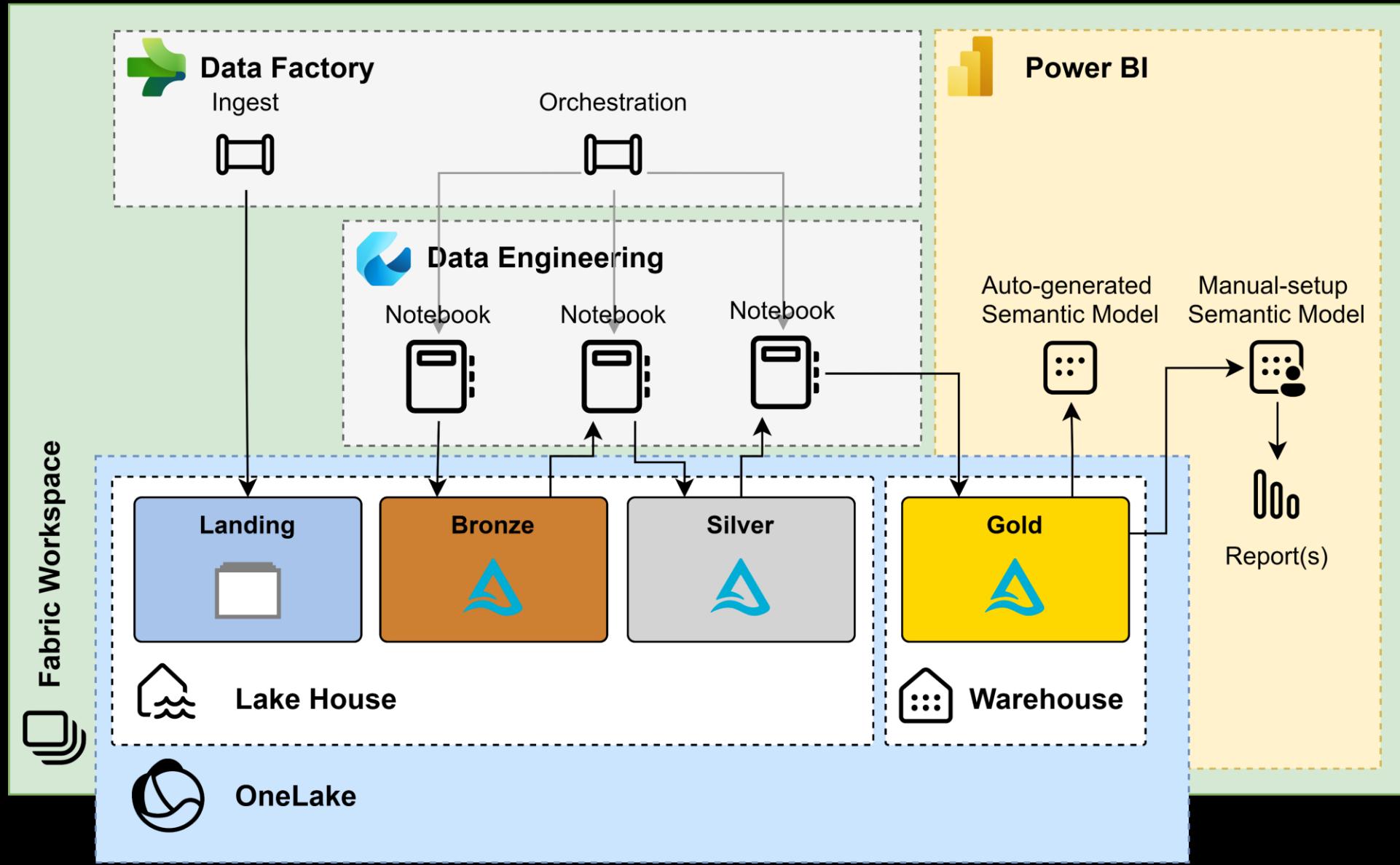
Landing zone

- Structured data
- Unstructured data
- Incremental loads
- Data as is
- Stored in Datetime folder structure
- No Schema





Microsoft Fabric



Demo





Who is already using Parameters?

Parameters

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

IMPLEMENTING
DEFAULT PARAMETERS
THAT DEPEND ON
OTHER PARAMETERS

Understand Parameters

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

The screenshot shows the 'Parameters' tab in the Azure Data Factory Copy Data Activity configuration pane. It displays a table with one row, where the 'Name' is 'cw_items_64o', the 'Type' is 'Array', and the 'Default value' is a JSON array containing four objects. To the right of the screenshot is the corresponding JSON configuration code.

Name	Type	Default value
cw_items_64o	Array	[{"source": {"table": "Department"}, "destination": {"fileName": "HumanResourcesDepartment.par"}, {"source": {"table": "Employee"}, "destination": {"fileName": "HumanResourcesEmployee.par"}, {"source": {"table": "EmployeeDepartmentHistory"}, "destination": {"fileName": "HumanResourcesEmployeeDep"}, {"source": {"table": "EmployeePayHistory"}, "destination": {"fileName": "HumanResourcesEmployeePay"}}]

```
{
  "source": {
    "table": "Department"
  },
  "destination": {
    "fileName": "HumanResourcesDepartment.par"
  }
}

{
  "source": {
    "table": "Employee"
  },
  "destination": {
    "fileName": "HumanResourcesEmployee.par"
  }
}

{
  "source": {
    "table": "EmployeeDepartmentHistory"
  },
  "destination": {
    "fileName": "HumanResourcesEmployeeDep"
  }
}

{
  "source": {
    "table": "EmployeePayHistory"
  },
  "destination": {
    "fileName": "HumanResourcesEmployeePay"
  }
}
```

Pipeline Parameters

- Pass through from
 - Pipeline to Activity

The screenshot shows the Azure Data Factory pipeline editor interface. At the top, there's a navigation bar with Home, Activities, Run, View, and other buttons like Validate, Run, Schedule, Trigger (preview), View run history, Copy data, Dataflow, Notebook, and a search icon.

The main area displays a pipeline component. A purple arrow points from the text "Items" in a white box at the bottom left to the "Items" field in the "Activities (1)" tab of the pipeline settings. The "Activities (1)" tab is currently selected, showing:

- General tab (disabled)
- Settings tab (selected)
- Activities (1) tab (disabled)

Under the Settings tab, there are two fields:

- Sequential checkbox (unchecked)
- Batch count input field (empty)

Under the Activities (1) tab, there is one activity named "ForEach_64o". It contains a single child activity named "Copy_64o".

A callout box at the bottom right highlights the expression `@pipeline().parameters.cw_items_64o`.

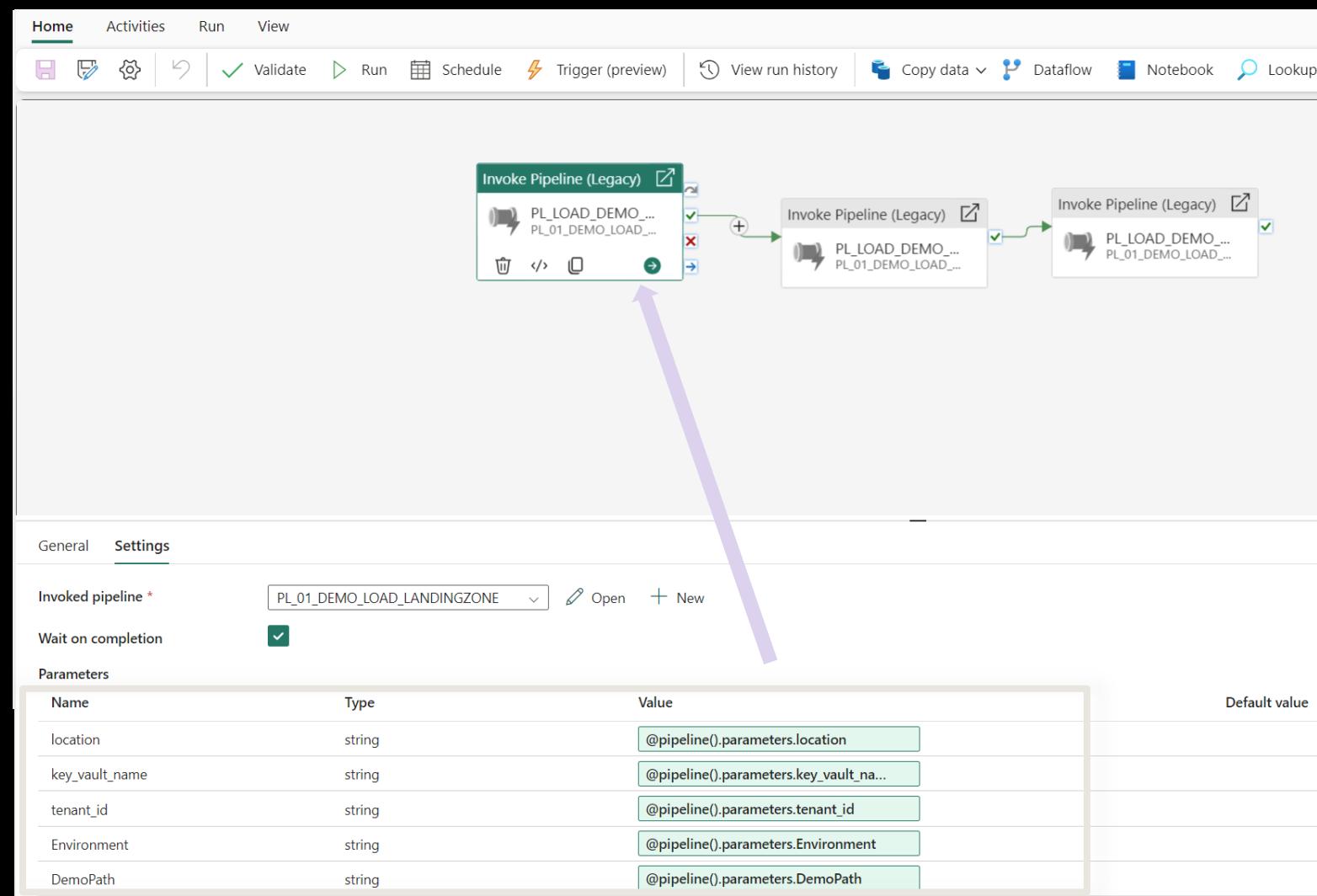
```
graph LR; Items[Items] --> Pipeline[Pipeline]; Pipeline --> Settings[Settings]; Settings --> Activities[Activities]; Activities --> ForEach[ForEach]; ForEach --> Copy[Copy]; Copy --> Callout[Callout]
```

Items

@pipeline().parameters.cw_items_64o

Pipeline Parameters

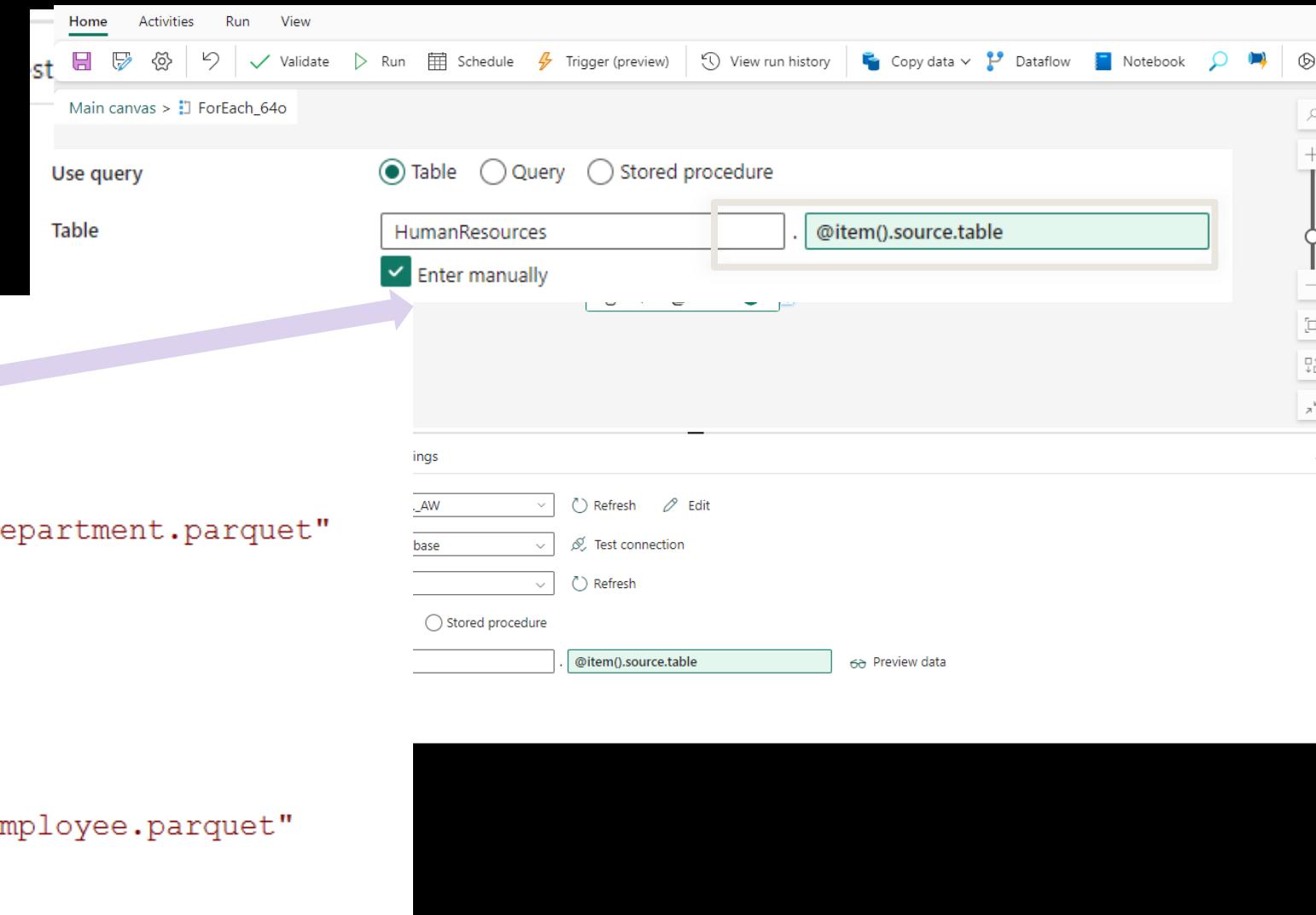
- Pass through from
 - Pipeline to Pipeline



Copy Activity Parameters

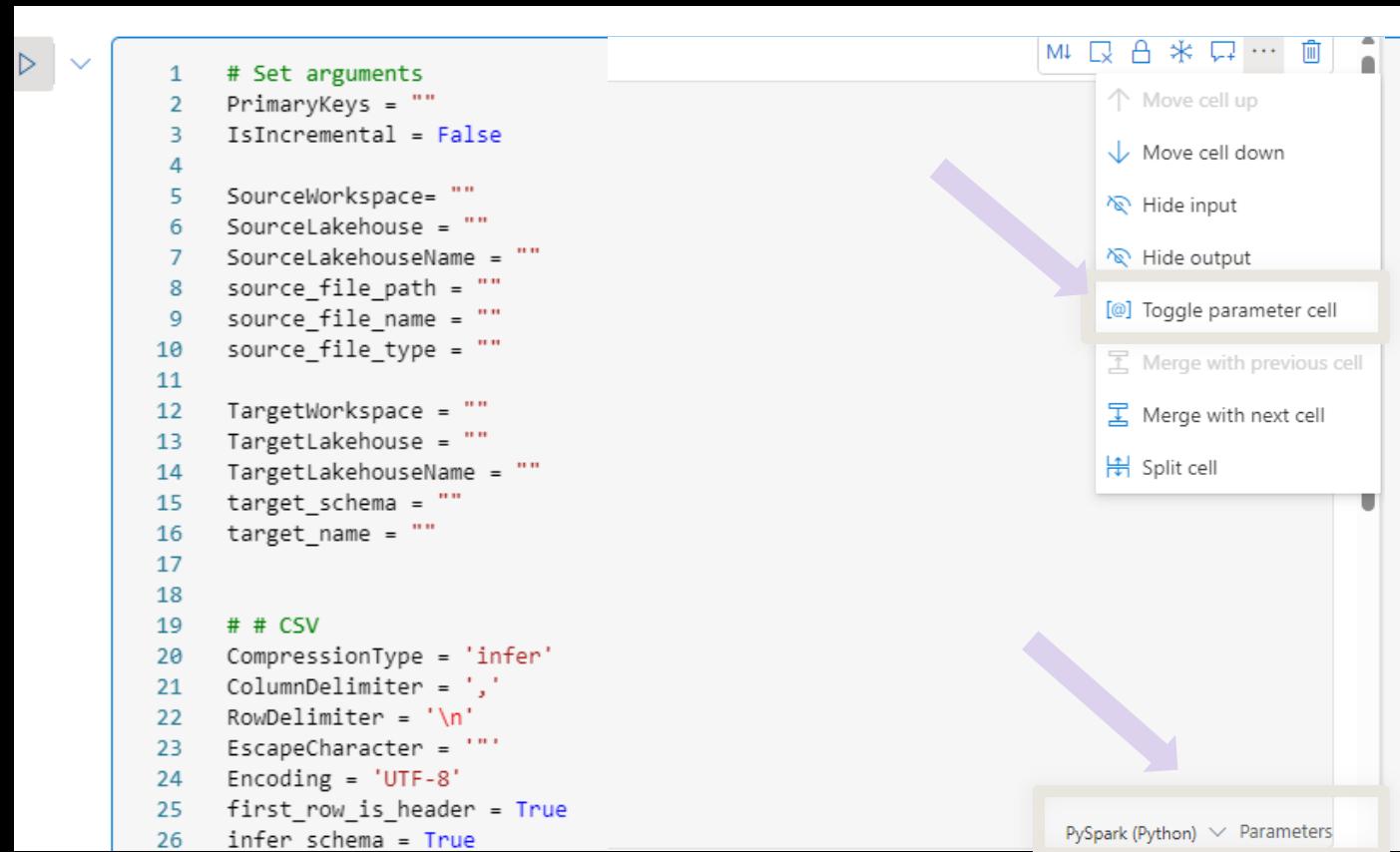
- Pass Parameters from Pipeline to Copy activity
- Use Parameters from For Each Activity

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



Notebook Parameters

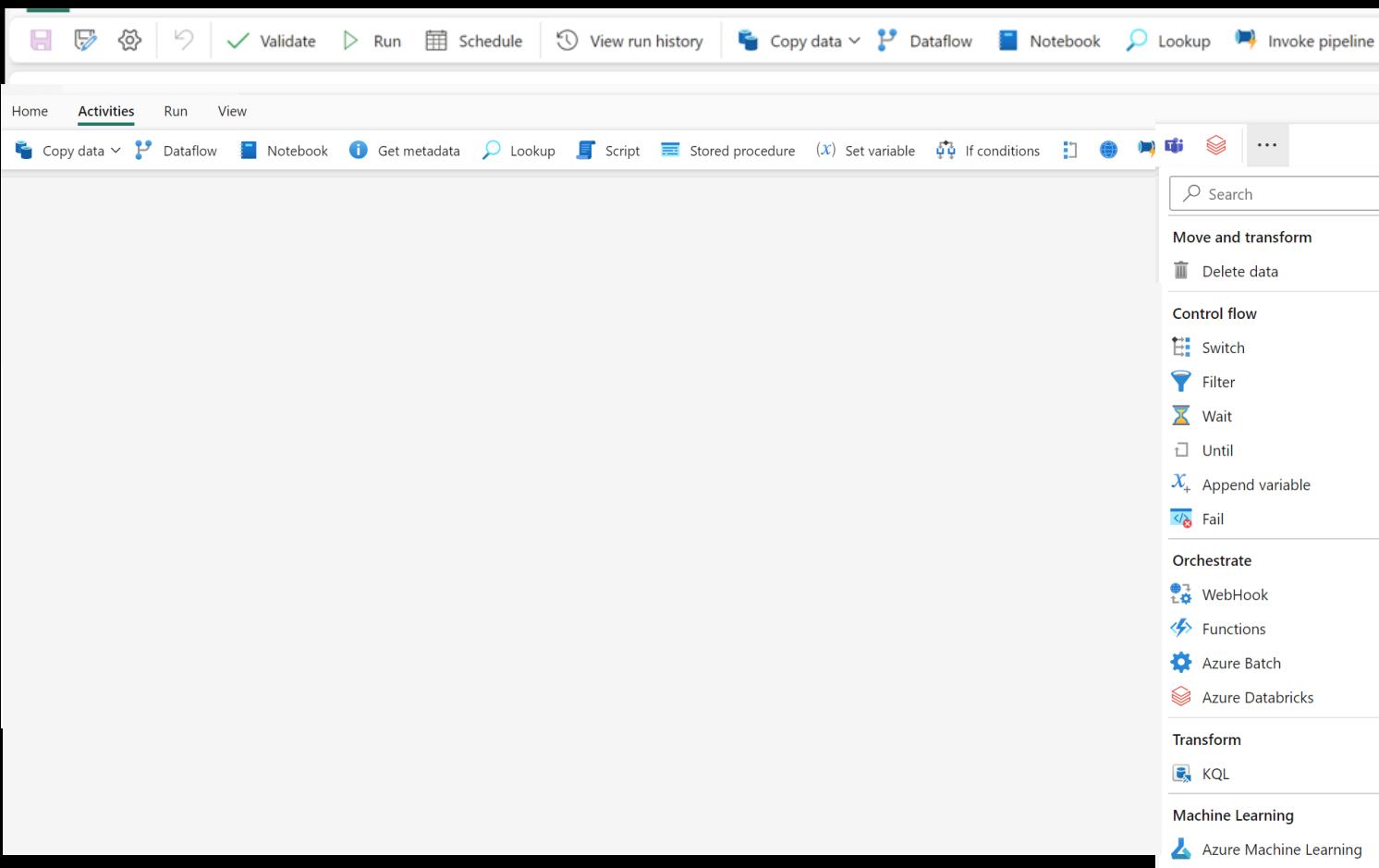
- Pass Parameters from Data Pipeline to Notebook Activity
 - Toggle parameter cell



```
1 # Set arguments
2 PrimaryKeys = ""
3 IsIncremental = False
4
5 SourceWorkspace= ""
6 SourceLakehouse = ""
7 SourceLakehouseName = ""
8 source_file_path = ""
9 source_file_name = ""
10 source_file_type = ""
11
12 TargetWorkspace = ""
13 TargetLakehouse = ""
14 TargetLakehouseName = ""
15 target_schema = ""
16 target_name = ""
17
18
19 # # CSV
20 CompressionType = 'infer'
21 ColumnDelimiter = ','
22 RowDelimiter = '\n'
23 EscapeCharacter = ''
24 Encoding = 'UTF-8'
25 first_row_is_header = True
26 infer schema = True
```

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 data from Azure SQL DB to Lakehouse Table

About this template

Use this template to copy data from your Azure SQL database to a specified table in your Lakehouse.

If you want to copy data from a number of tables, please use the "Copy assistant" to create your pipeline.

Inputs

Connection *
For Copy from SQL to LH (Copy source)

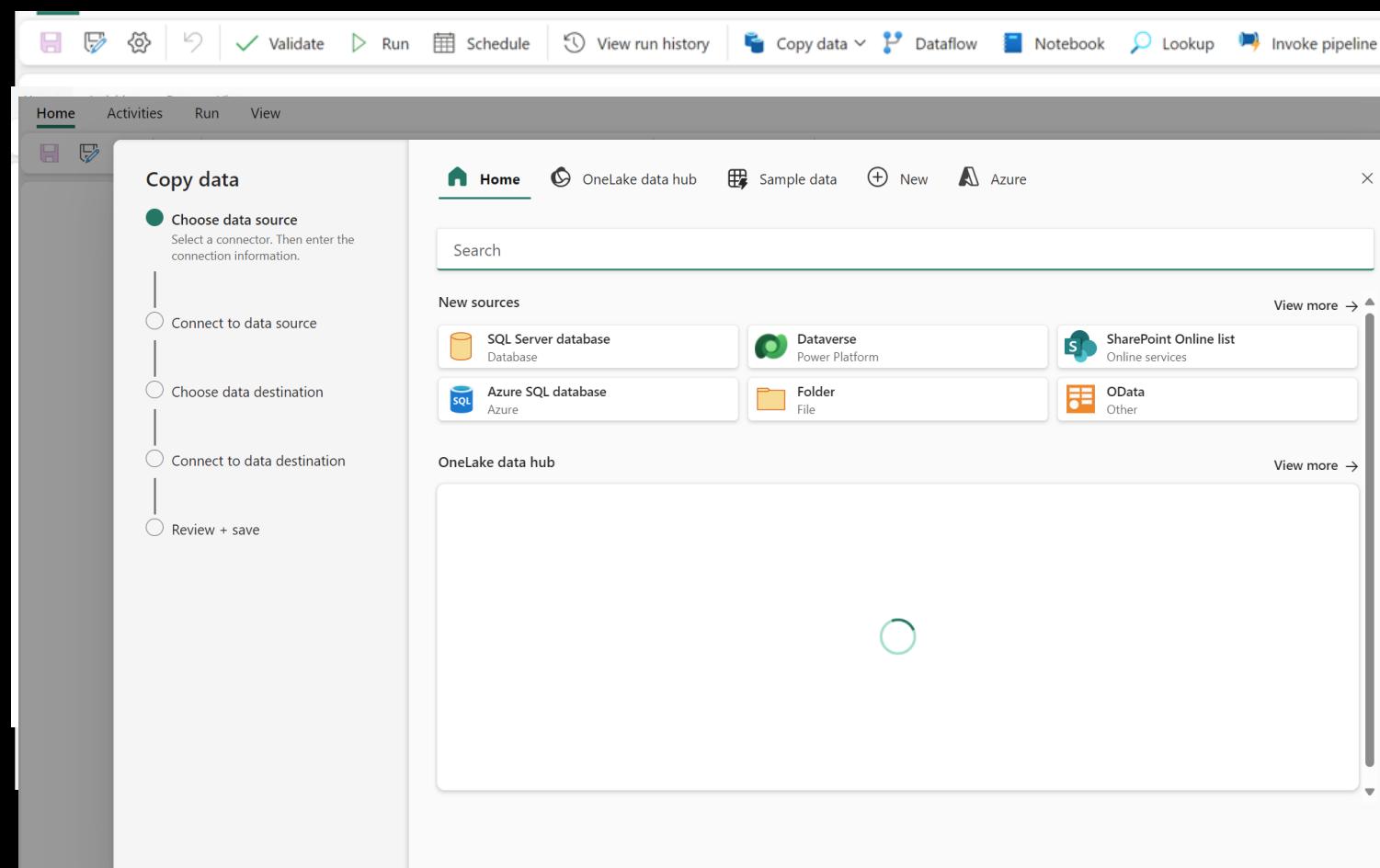
Select... ↻ + New

Lakehouse *
For Copy from SQL to LH (Copy destination)

Select... ↻ + New

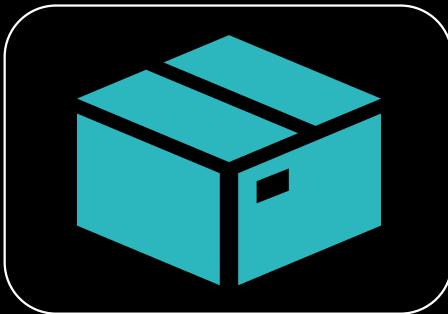
Copy Assistant

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

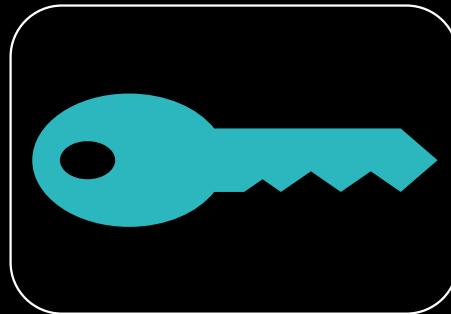


Frameworks

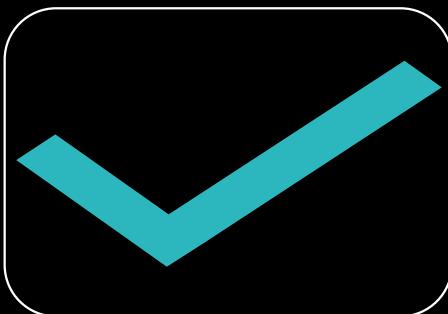
Why



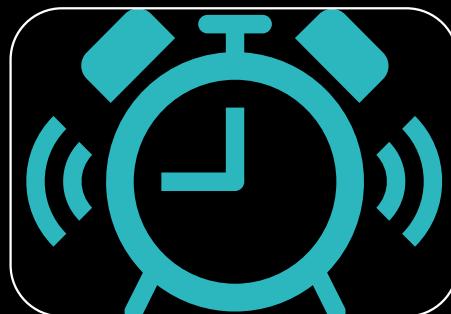
Scalability



Automation



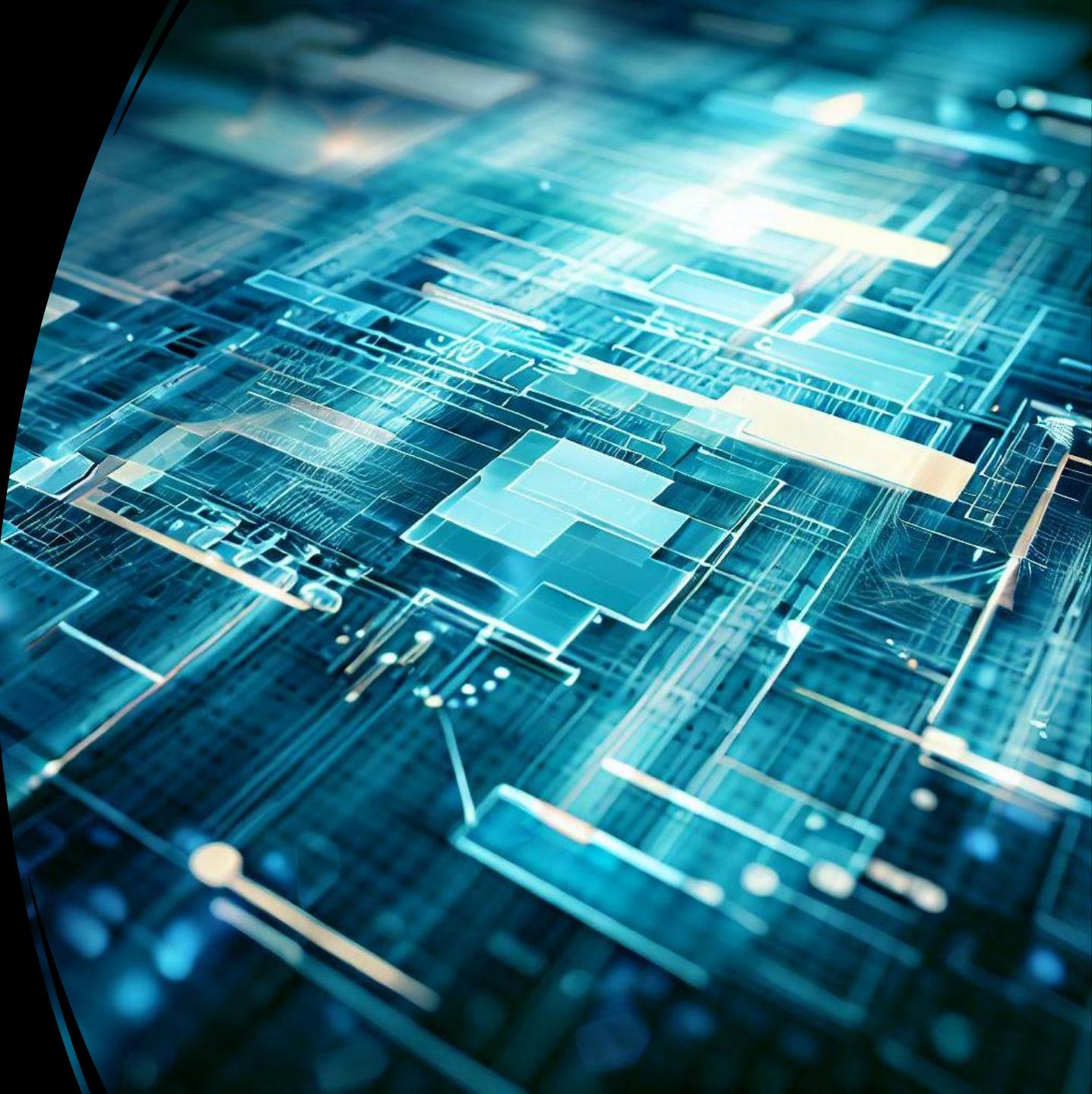
Traceability



Flexibility

Out-of-the-Box Framework

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



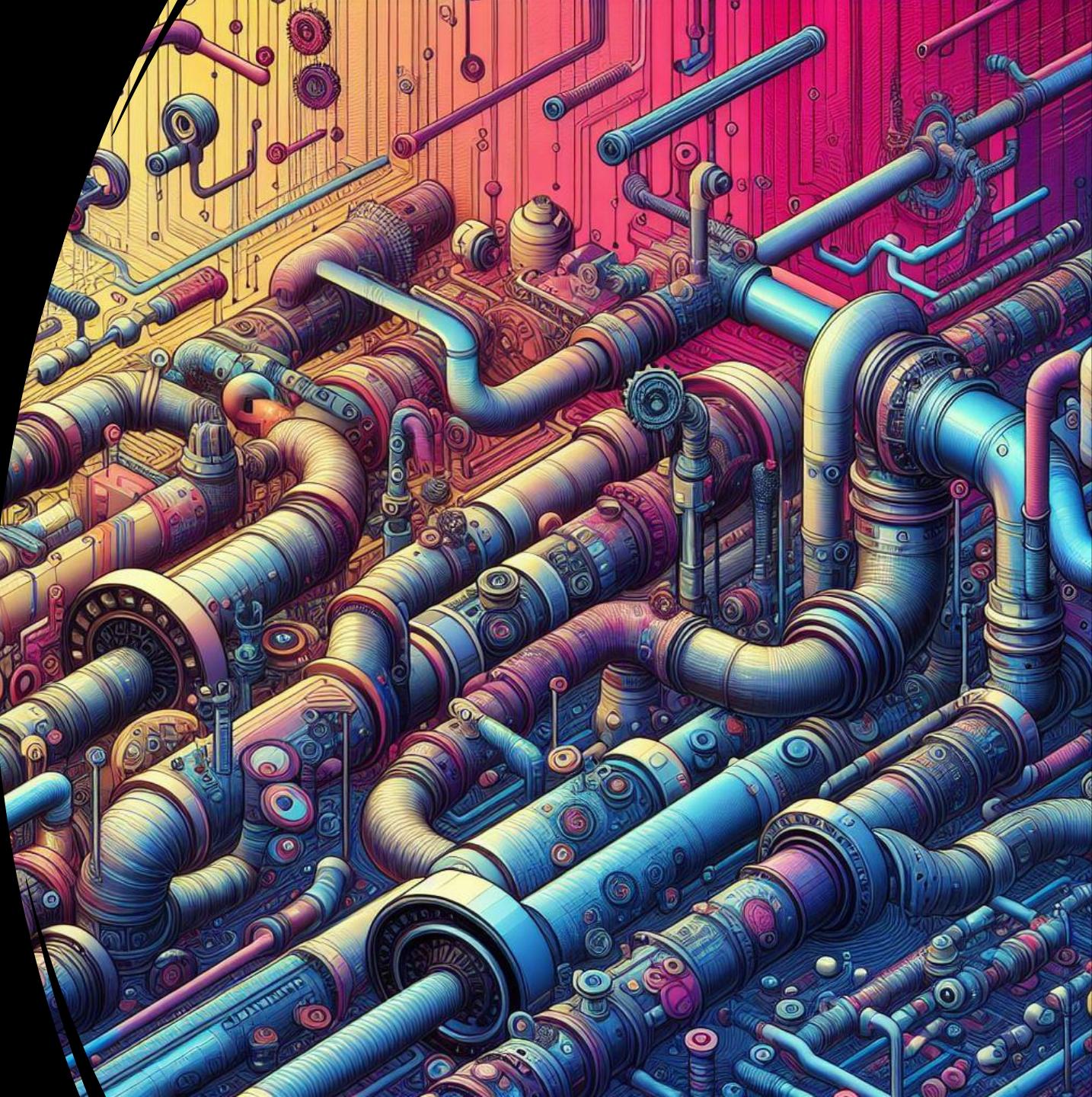
Custom-Made Framework

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

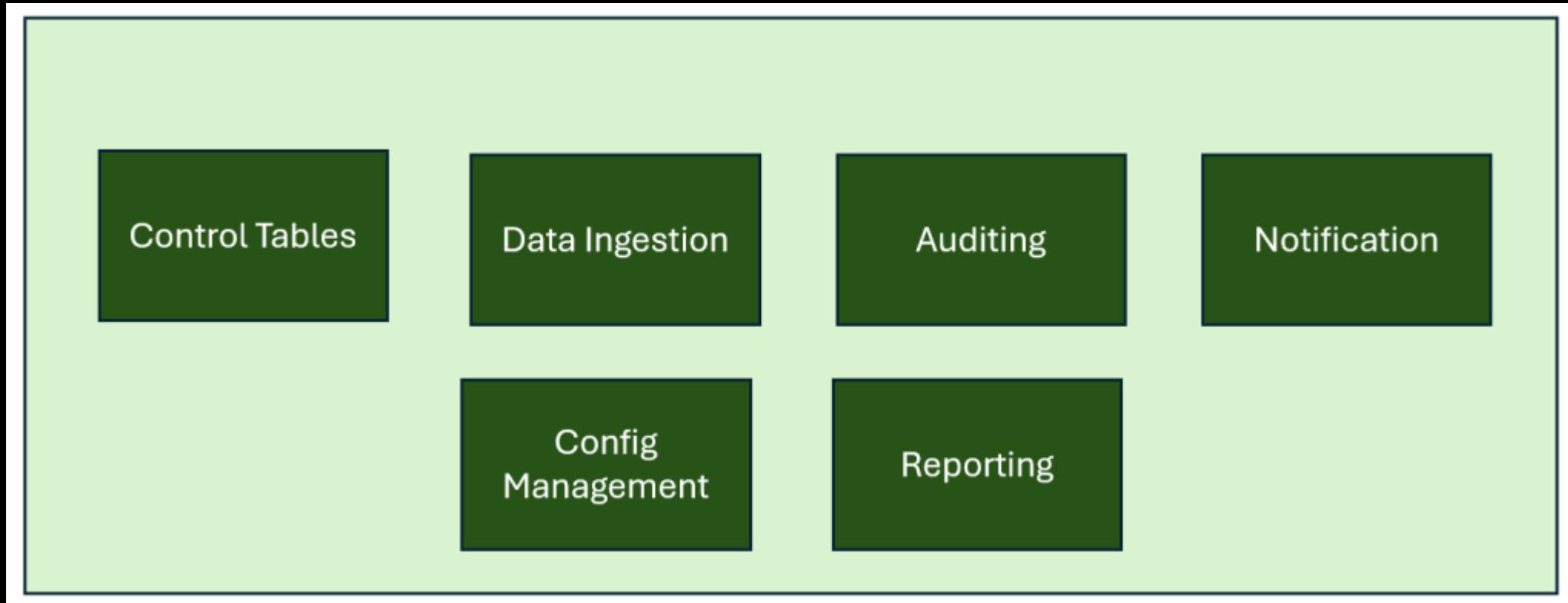


Custom-Made Framework

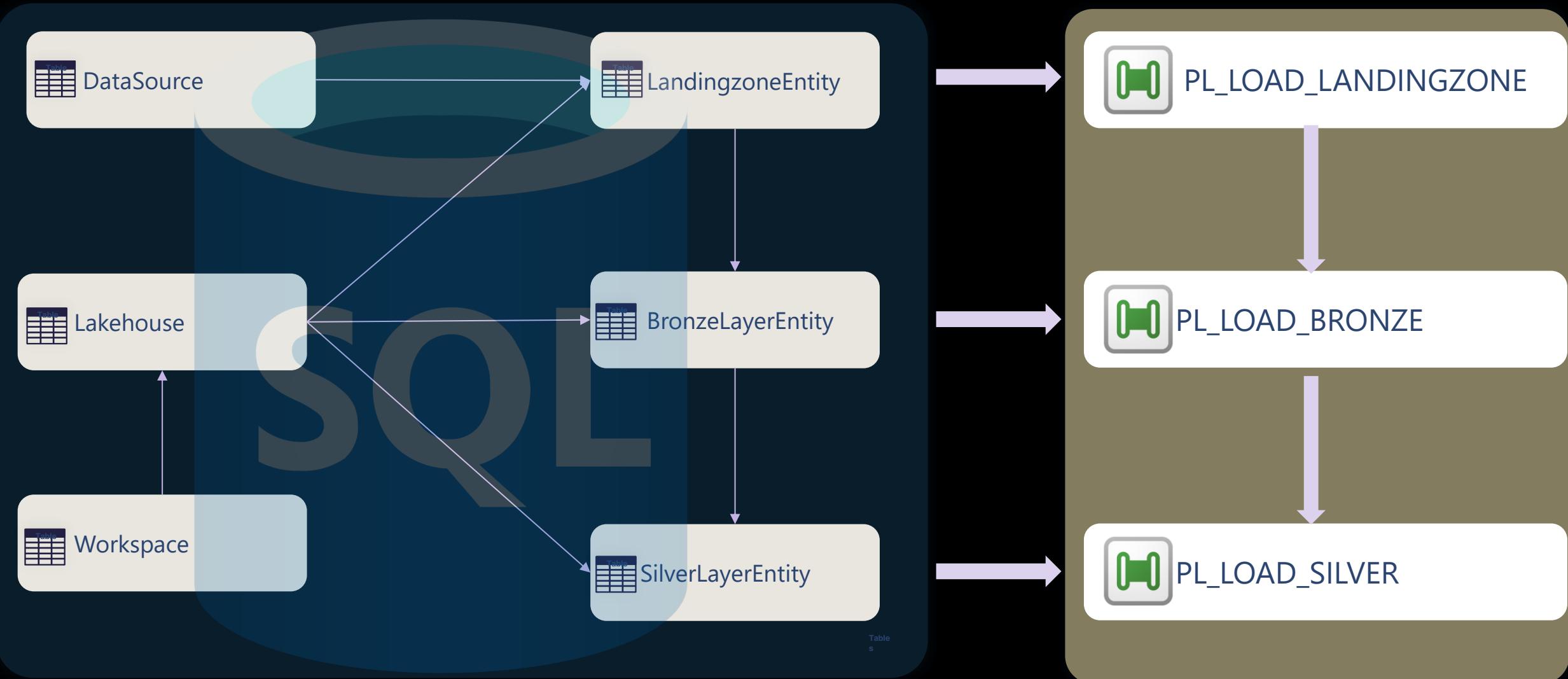
- Based on parameters
- Meta data => Azure SQL Database / Json /
- Microsoft Fabric but also on Azure Synapse Analytics and Azure Data Factory
- Based on the Medallion Architecture



Custom-Made Framework

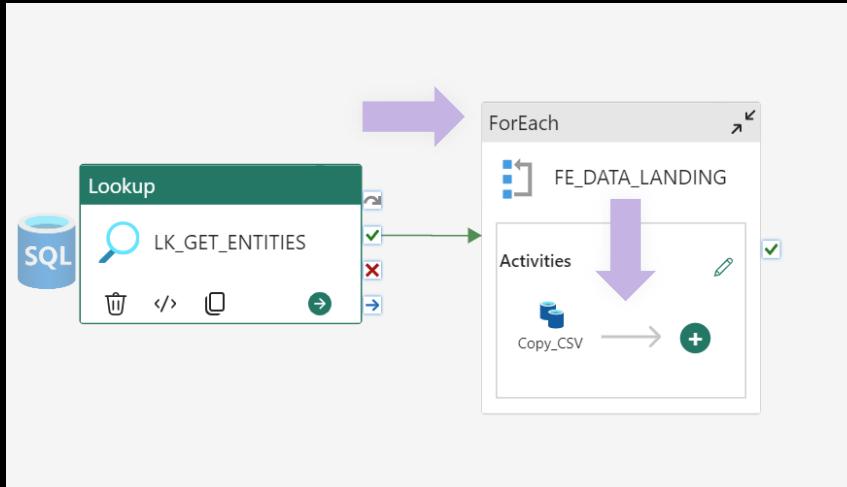


Framework



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	T
Jwdvlmdl01	ADLS	ADLS	true		customers.csv	demo/2024/09/10	customers_2024091015.csv	csv	O
Jwdvlmdl01	ADLS	ADLS	true		organizations.csv	demo/2024/09/10	organizations_2024091015.csv	csv	O
Jwdvlmdl01	ADLS	ADLS	true		people.csv	demo/2024/09/10	people_2024091015.csv	csv	O

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`

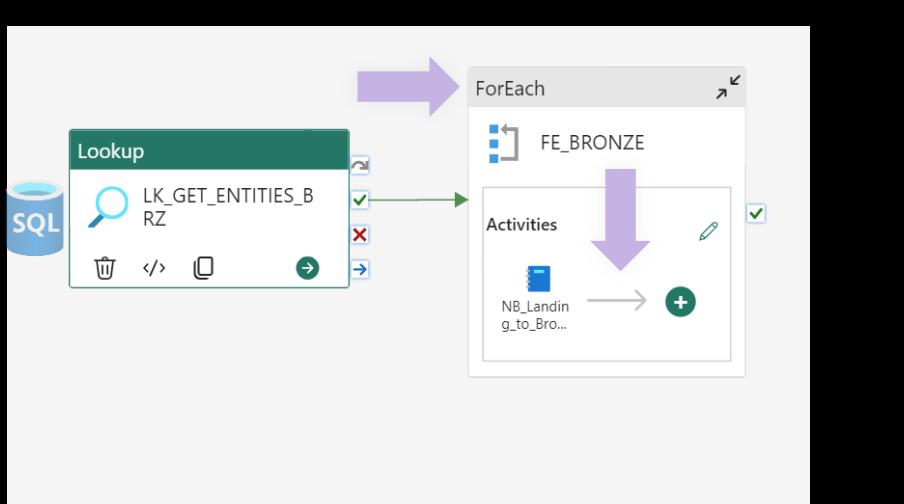
File format * `Binary`

> Advanced

Two purple arrows point upwards from the `TargetFilePath` and `TargetFileName` fields in the Destination settings to the corresponding parameters in the table above, indicating they are being passed as pipeline variables.

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	SourceLakehouseName
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

Base parameters

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

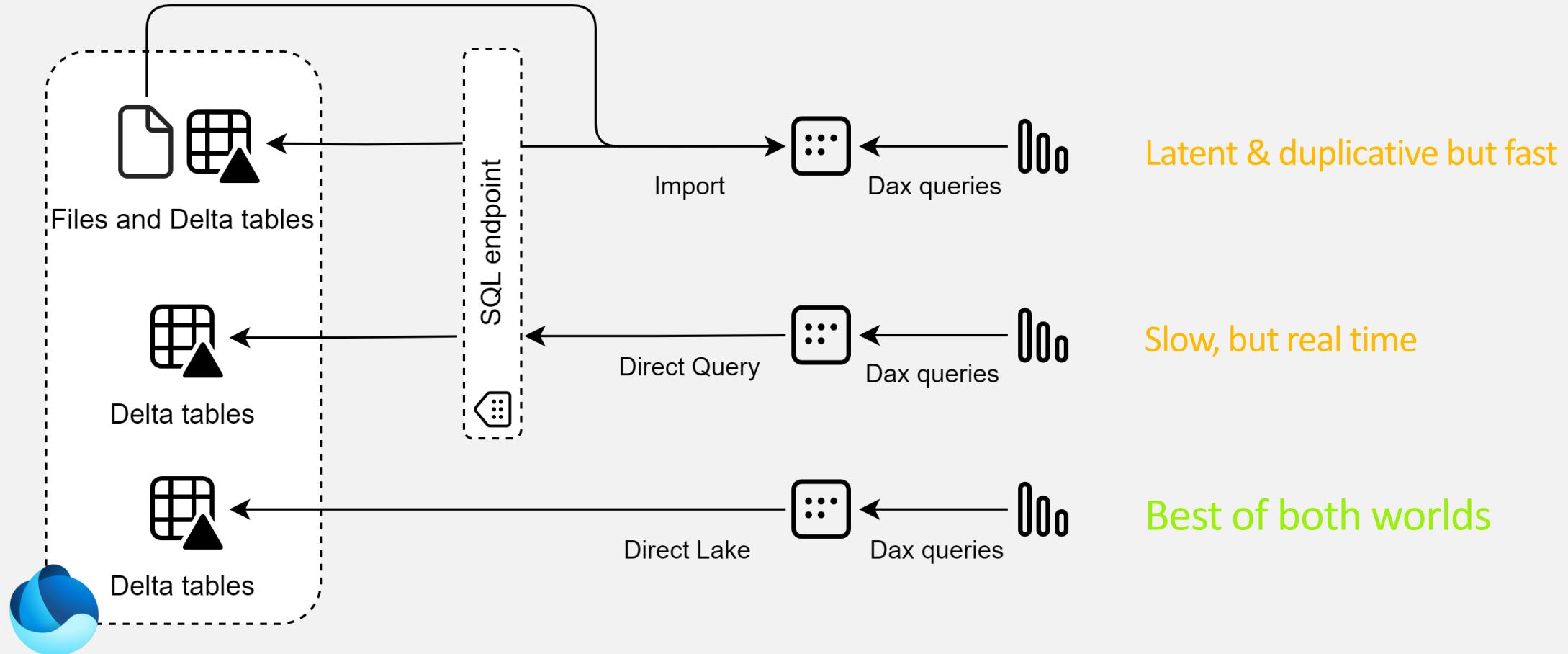
Demo





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*

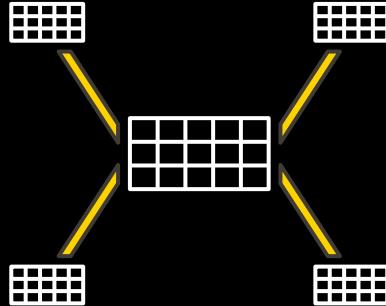


Data transformations should be done as far upstream as possible

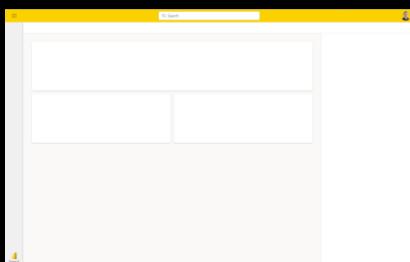


Data transformation directly in the Lakehouse unlocks “new” possibilities

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 ambiguous data models
- ... etcetera

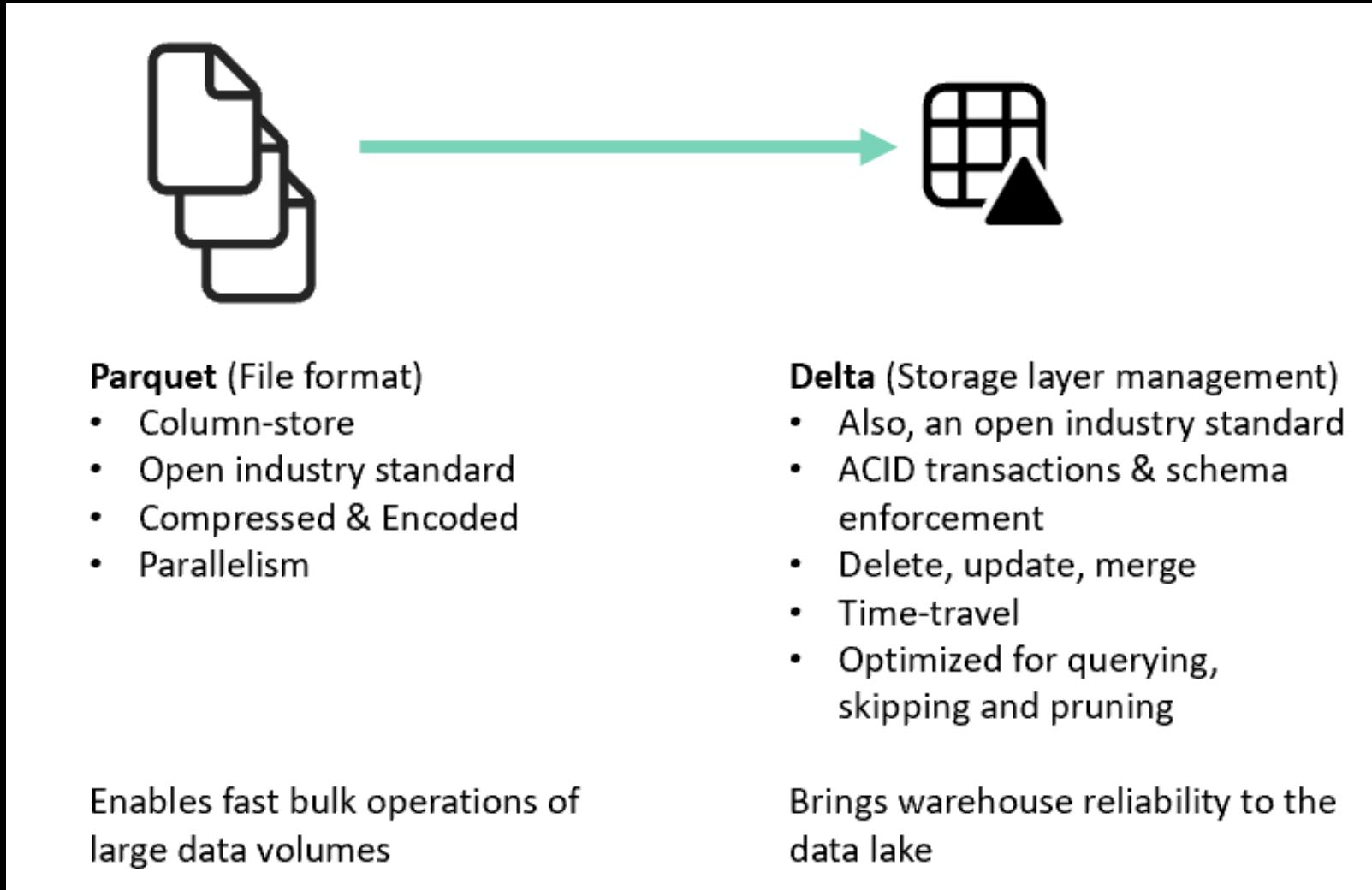
Demo





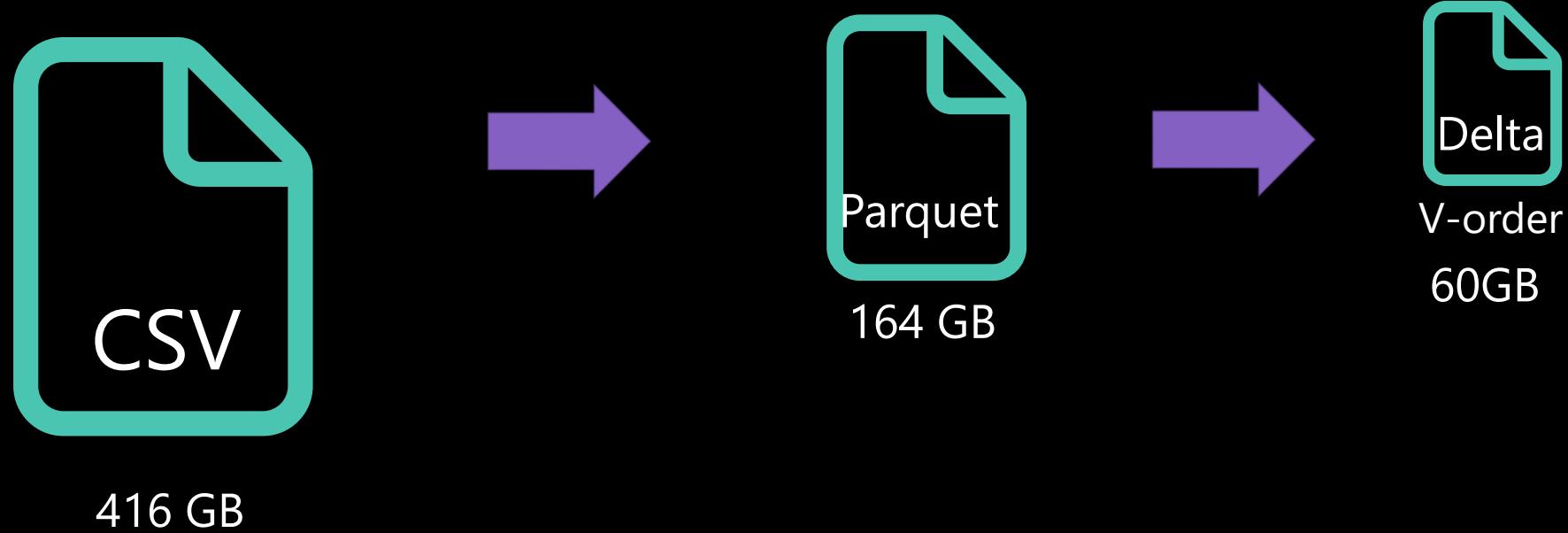
Direct Lake Internals & performance

Delta (Parquet)



From Z-order to V-order

Yellow taxi (3 Billion rows)

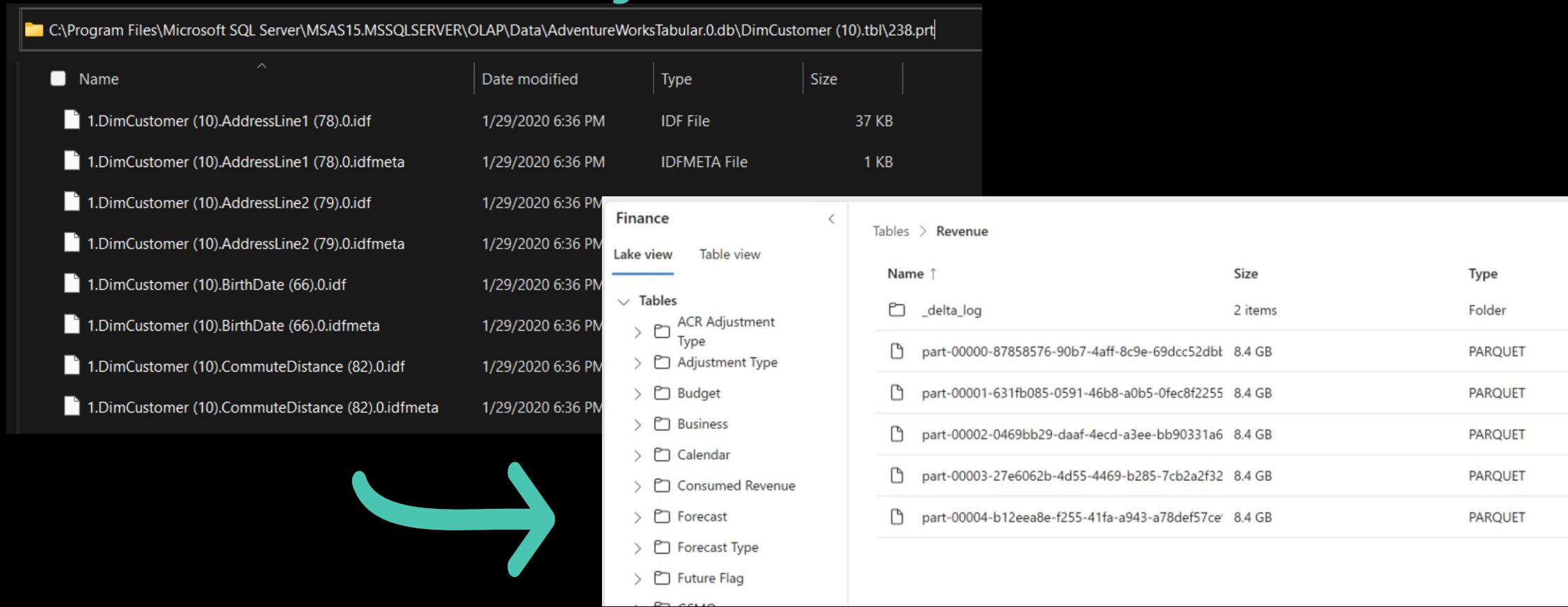


x3.2
Less I/O for all*
workloads

Data saving

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

SSAS, AAS, Power BI large models



C:\Program Files\Microsoft SQL Server\MSAS15.MSSQLSERVER\OLAP\Data\AdventureWorksTabular.0.db\DimCustomer (10).tbl\238.prt

Name	Date modified	Type	Size
1.DimCustomer (10).AddressLine1 (78).0.idf	1/29/2020 6:36 PM	IDF File	37 KB
1.DimCustomer (10).AddressLine1 (78).0.idfmeta	1/29/2020 6:36 PM	IDFMETA File	1 KB
1.DimCustomer (10).AddressLine2 (79).0.idf	1/29/2020 6:36 PM		
1.DimCustomer (10).AddressLine2 (79).0.idfmeta	1/29/2020 6:36 PM		
1.DimCustomer (10).BirthDate (66).0.idf	1/29/2020 6:36 PM		
1.DimCustomer (10).BirthDate (66).0.idfmeta	1/29/2020 6:36 PM		
1.DimCustomer (10).CommuteDistance (82).0.idf	1/29/2020 6:36 PM		
1.DimCustomer (10).CommuteDistance (82).0.idfmeta	1/29/2020 6:36 PM		

Finance

Lake view Table view

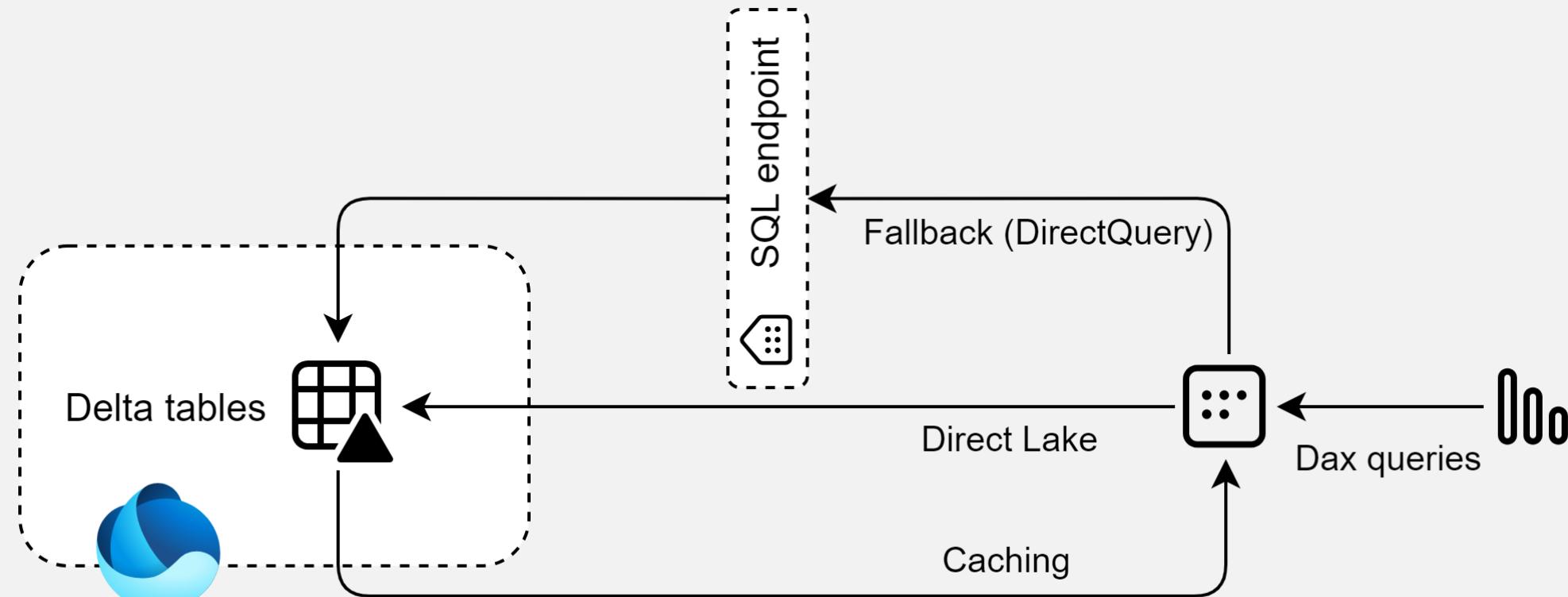
Tables

- ACR Adjustment
- Adjustment Type
- Budget
- Business
- Calendar
- Consumed Revenue
- Forecast
- Forecast Type
- Future Flag
- SSMO

Tables > Revenue

Name ↑	Size	Type
_delta_log	2 items	Folder
part-00000-87858576-90b7-4aff-8c9e-69dcc52db1	8.4 GB	PARQUET
part-00001-631fb085-0591-46b8-a0b5-0fec8f2255	8.4 GB	PARQUET
part-00002-0469bb29-daa5-4ecd-a3ee-bb90331a6	8.4 GB	PARQUET
part-00003-27e6062b-4d55-4469-b285-7cb2a2f32	8.4 GB	PARQUET
part-00004-b12eea8e-f255-41fa-a943-a78def57ce'	8.4 GB	PARQUET

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

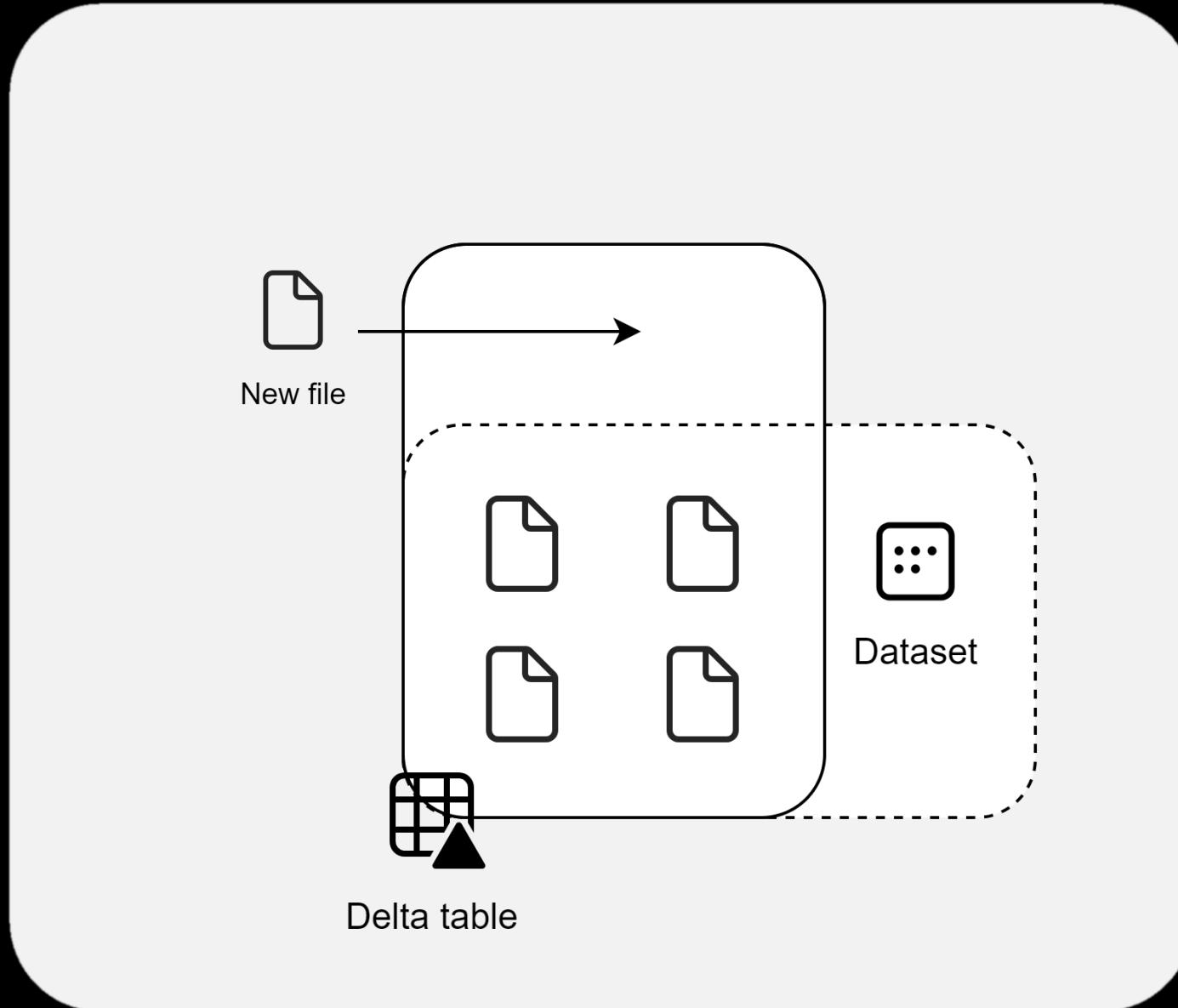
Item level on lakehouse

Introducing Framing

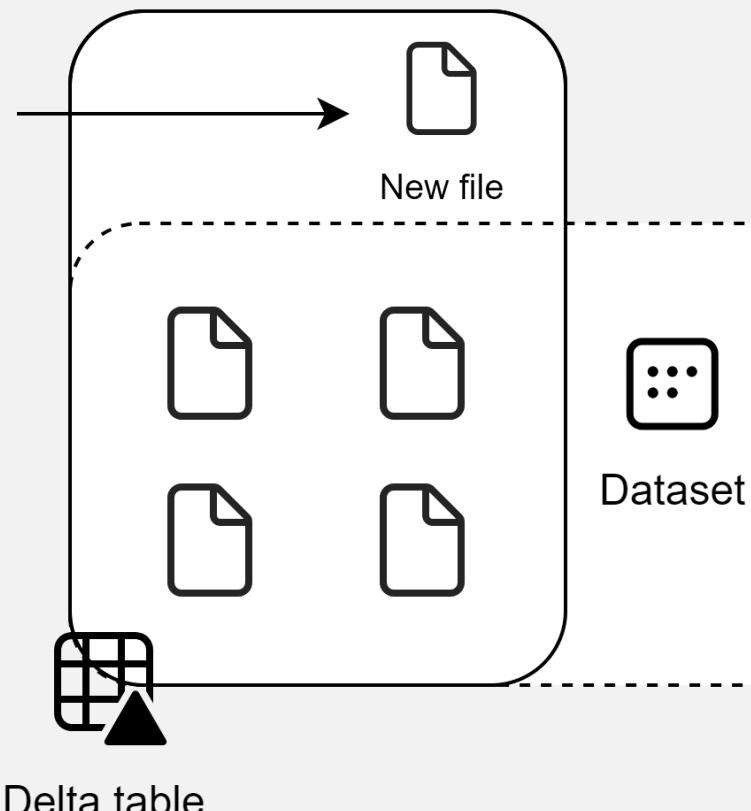
- Metadata **refresh** which does not actually load the data, but only the delta table definitions.



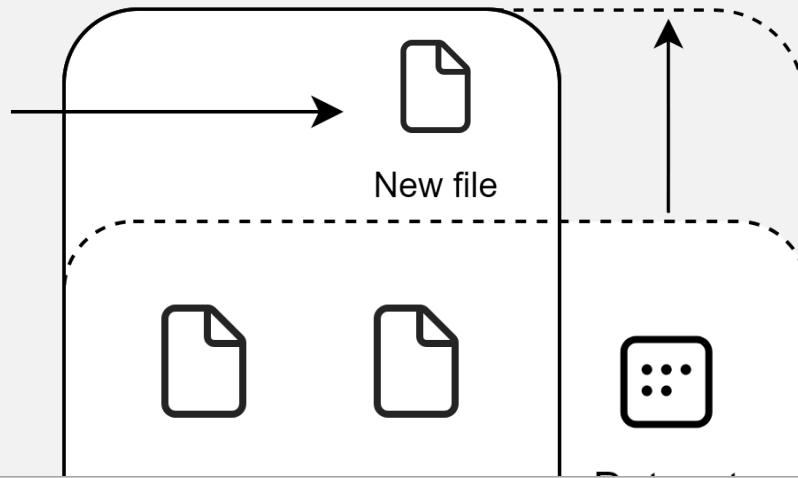
Framing



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

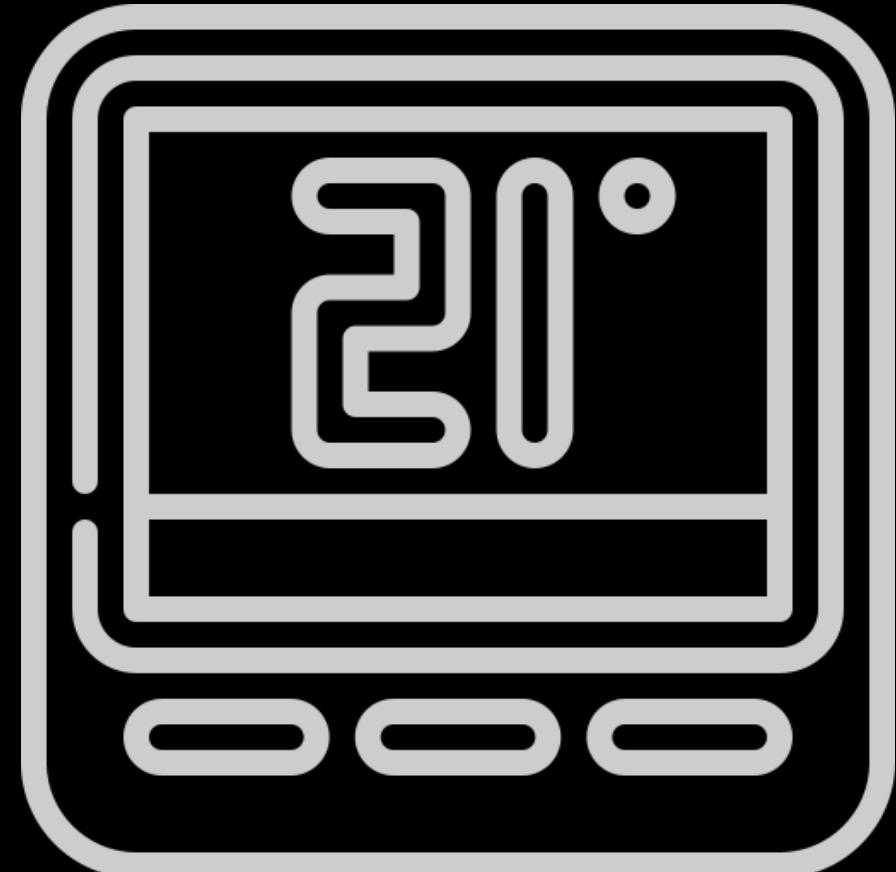
Demo



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

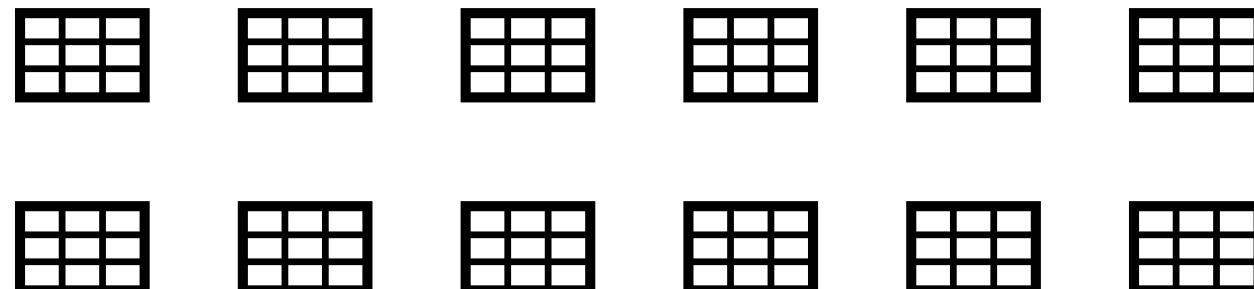
Capacity utilization:

Cool

Active memory



Storage



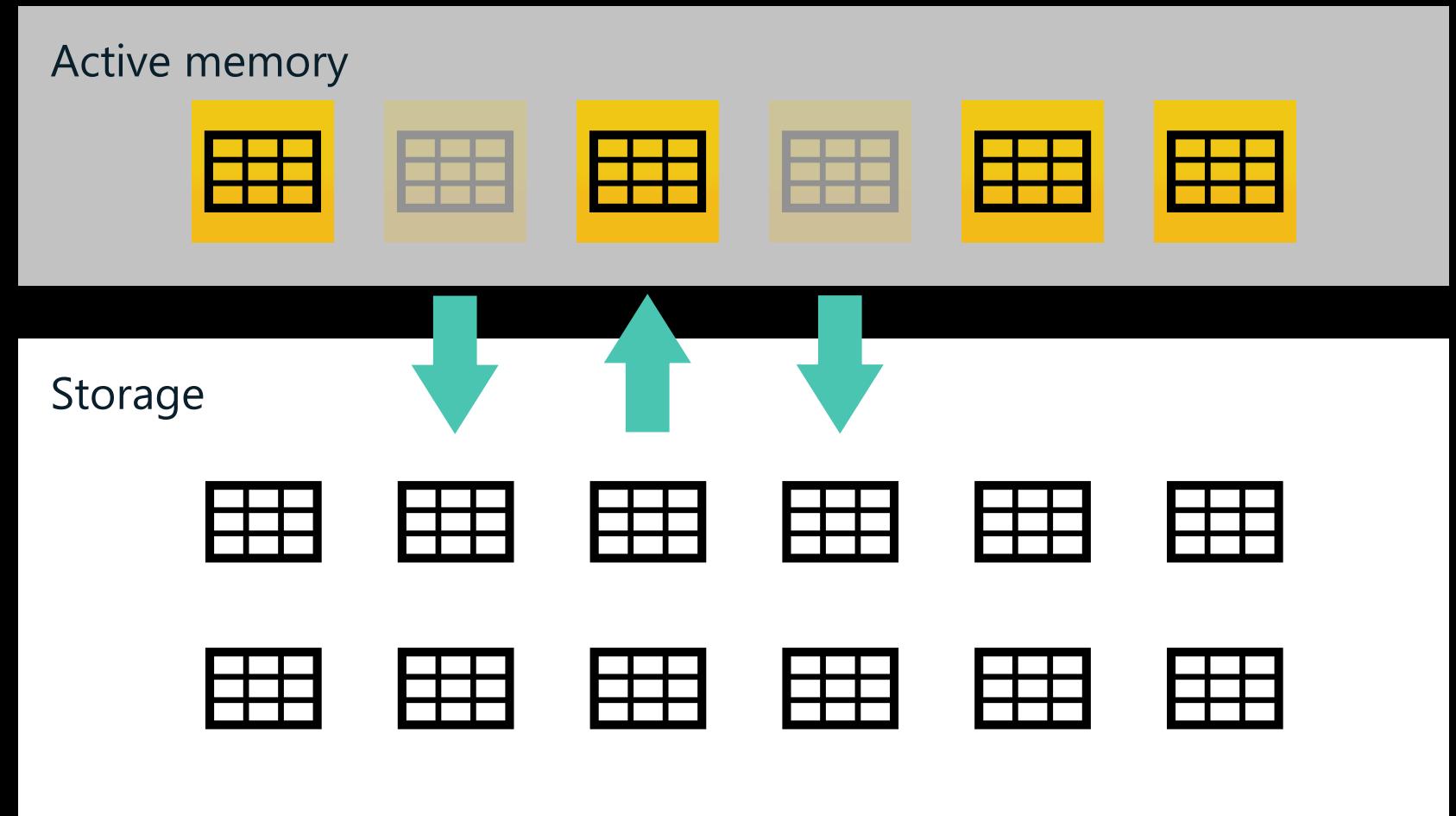
Eviction

Capacity: F64 / P1

Memory: 25 GB

Capacity utilization:

Warm



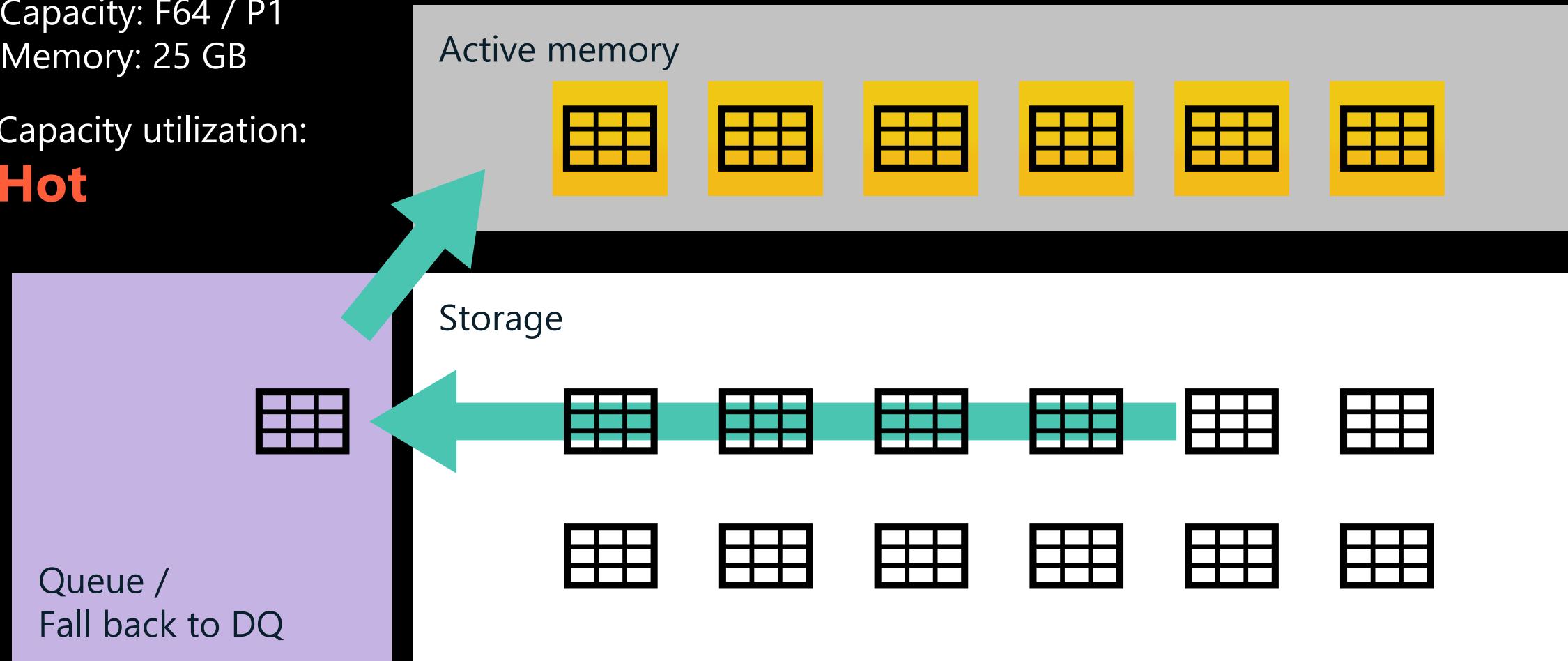
Eviction – queue / fall-back

Capacity: F64 / P1

Memory: 25 GB

Capacity utilization:

Hot



Temperature management

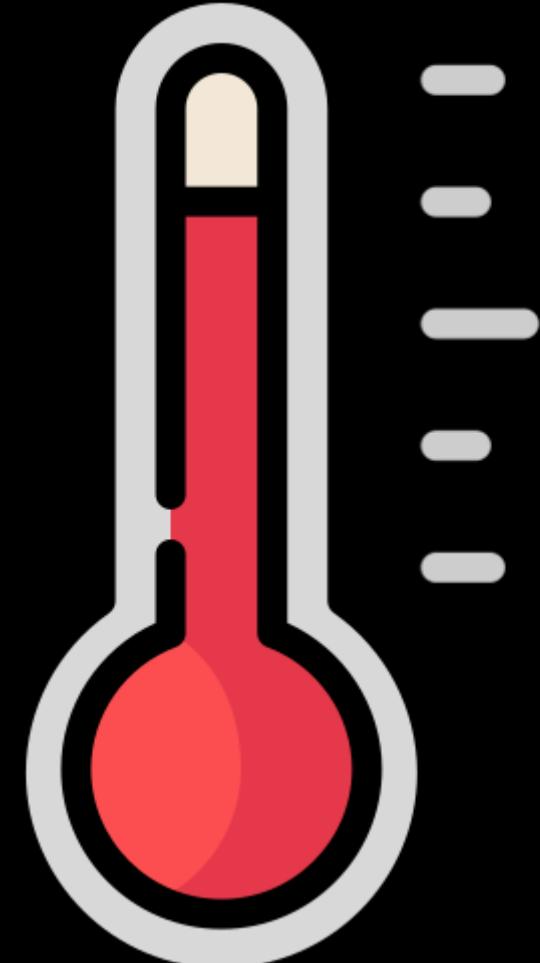
What will be evicted?

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

How can you influence that?

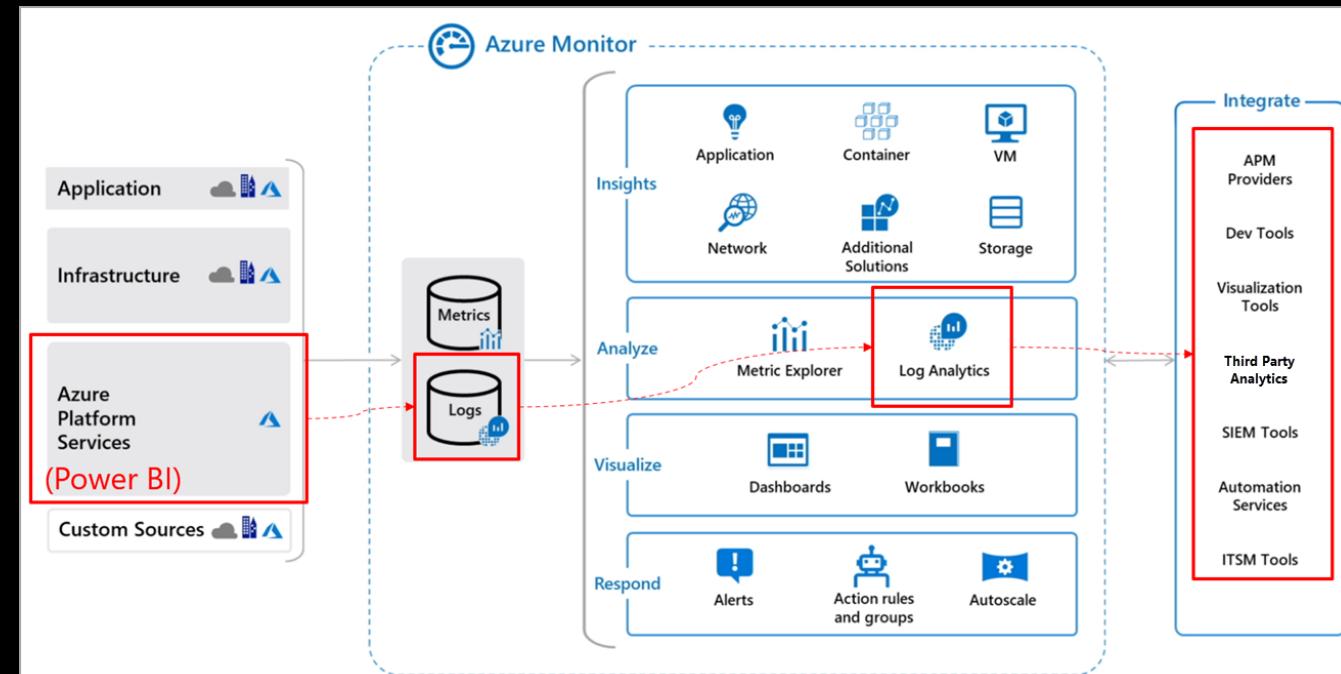
Consider setting up a process (notebook, other automated setup) to pro-actively execute queries to keep certain data **WARM**!

.



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

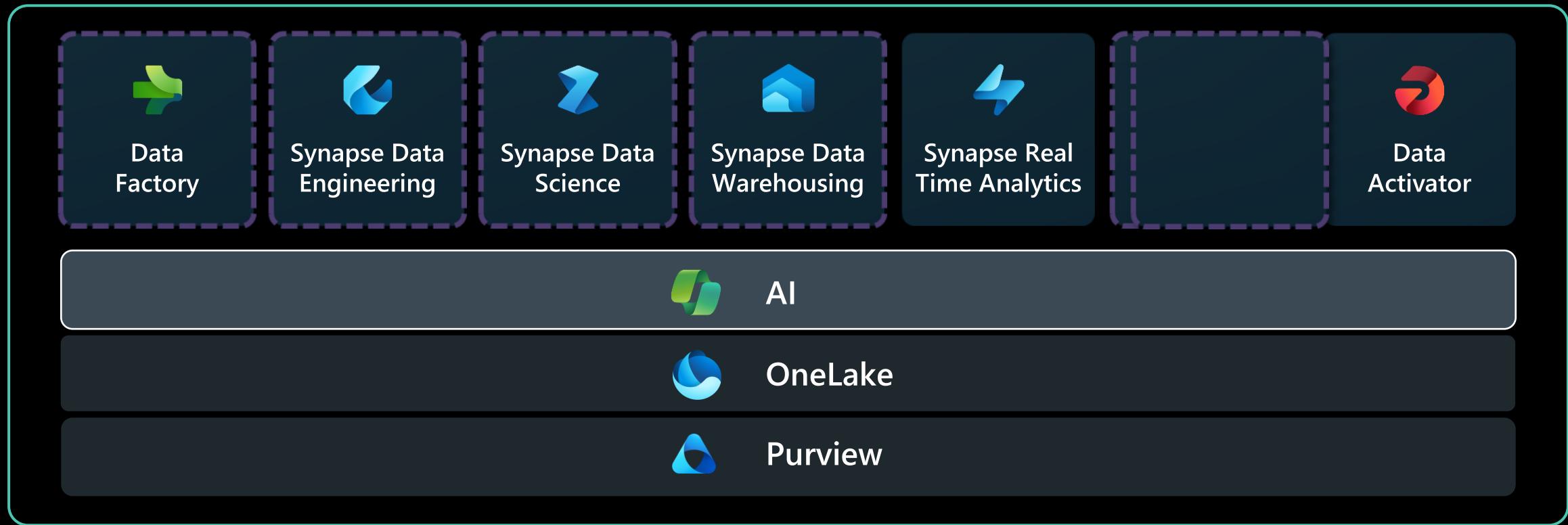
Autopilot refers to allowing AI to make all the decisions, while copilot involves AI assisting humans in making informed





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)

↳ Users can use a preview of Copilot and other features powered by Azure OpenAI New
Enabled for a subset of the organization

When this setting is enabled, users can access a preview and use preview features powered by Azure OpenAI, including Copilot. This setting can be managed at both the tenant and the capacity levels. [Learn More](#)

For customers in the EU Data Boundary, this setting adheres to Microsoft Fabric's EU Data Boundary commitments. [Learn More](#)

By enabling this setting, you agree to the [Preview Terms](#).

Enabled

ⓘ Note: If Azure OpenAI is not available in your geographic region, your data may need to be processed outside your capacity's geographic region, compliance boundary, or national cloud instance. To allow data to be processed outside your capacity's geographic region, turn on the related setting, "Data sent to Azure OpenAI can be processed outside your capacity's geographic region, compliance boundary, or national cloud instance".

Data sent to Azure OpenAI can be processed outside your capacity's geographic region, compliance boundary, or national cloud instance New
Enabled for the entire organization

This setting is only applicable for customers who want to use Copilot and AI features in Fabric powered by Azure OpenAI, and whose capacity's geographic region is outside of EU Data Boundary and US. [Learn More](#)

When this setting is enabled, data sent to Azure OpenAI can be processed outside your capacity's geographic boundary or national cloud boundary. This setting can be managed at both the tenant and the capacity levels. [Learn More](#)

By enabling this setting, you agree to the [Preview Terms](#).

Enabled

ⓘ Note: Even if this setting is on, you will also need to turn on the related setting "Users can use a preview of Copilot and other features powered by Azure OpenAI" for these features to work.

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.

The screenshot shows a Data Science workspace interface. On the left, there's a sidebar with options like Home, Create, Browse, Data hub, Monitoring hub, Workspaces, My workspace, and Sales analysis. The main area is a notebook titled "Sales analysis | Data updated 1/12/23". The notebook contains two code cells:

```
1 # Welcome to your new notebook
2 # Type here in the cell editor to add 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())
```

The second cell has a green checkmark and the text "- Command executed in 7 sec 427 ms by Sonia". Below the code, there's a table titled "Country/Region Ab" with 17 rows of data:

	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	Piano	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	Piano	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	Piano	13.0
15	US	Financial Services	Fort Worth	34.0
16	US	Insurance	Irving	31.0
17	US	Software & Services	Chicago	30.0

On the right side of the notebook, there's a "Copilot" panel with the heading "Transform your data with Copilot" and the sub-instruction "Describe what you want, in your own words, and Copilot will suggest code you can use." Below this, there's a snippet of Python code:

```
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 litySamplePBIX', 'Total Revenue',
8 [('Customer', 'Name')])
9 customer_sales =
10 customer_data.merge(sales_data,
11 on='Name')
12 print(customer_sales.head())
```

At the bottom of the Copilot panel, there are buttons for "Ask a question or request, or type / for suggestions" and "Selected Cell 1". A note at the bottom of the Copilot panel says: "AI-generated content can have mistakes. Make sure it's appropriate before using it. Read preview terms".

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

The screenshot shows a Jupyter Notebook interface titled "Sales analysis | Data updated 1/12/23". The left sidebar displays a "Lakehouse explorer" with sections for Customer360, Tables (Customer, CustomerFeedback..., Inventory, Product, Sales, Transaction), and Files (Sales, Excel-data). The main area has two code cells:

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

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

The second cell has a green checkmark indicating it was executed successfully. Below the code is a table titled "Country/Region Ab" with 17 rows of data:

	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	Piano	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	Piano	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	Piano	13.0
15	US	Financial Services	Fort Worth	34.0
16	US	Insurance	Irving	31.0
17	US	Software & Services	Chicago	30.0

On the right side, there's a "Copilot Preview" panel with the text "Transform your data with Copilot" and "Describe what you want, in your own words, and Copilot will suggest code you can use." Below it is another code cell:

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

At the bottom, there are buttons for "Code" and "Markdown", and a status bar at the bottom of the notebook.



(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

The screenshot shows a Microsoft OneLake data hub interface. At the top, there's a navigation bar with icons for Home, Create, Browse, OneLake data hub, Monitoring hub, Workspaces, My workspace, and MyDataExpert. On the far right of the header, there are trial status indicators, a user profile icon, and a help (?) icon.

The main content area is titled "OneLake data hub" and displays a list of lakehouse SQL endpoints. A sub-header says: "Select a lakehouse SQL endpoint for the model to reference. You'll select specific tables in the next step." Below this, there are four filter buttons: "All" (selected), "My data", "Endorsed in your org", and "Favorites". A search bar with a magnifying glass icon and a "Filter by keyword" button are also present.

The table lists the following endpoints:

Name	Owner	Refreshed	Endorsement	Sensitivity
aiassistanticm	Markus Cozowicz	-	-	Confidential\Mi...
HolidaysLH	Nellie Gustafsson	-	-	Confidential\Mi...
test	Nellie Gustafsson	-	-	Confidential\Mi...
superstorelh	Markus Cozowicz	-	-	Confidential\Mi...
Sales_Lakehouse	Nellie Gustafsson	-	-	Confidential\Mi...
TestLHNellie	Nellie Gustafsson	-	-	Confidential\Mi...
CustomerChurnLakehouse	Misha Desai	-	-	Confidential\Mi...
Banana	Nellie Gustafsson	-	-	Confidential\Mi...
Customer_Sales_Lakehouse	Nellie Gustafsson	-	-	Confidential\Mi...
Mercury	Dhwani Sudanagunta	-	-	Confidential\Mi...
AdventureWorksLH	Richard Wydrowski	-	-	Confidential\Mi...
AdventureWorksNeustl H	Nellie Gustafsson	-	-	Confidential\Mi...

At the bottom right of the dialog are "Connect" and "Cancel" buttons.



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

The screenshot shows the Power BI desktop application interface. On the left, the ribbon includes Home, Create, Browse, Data hub, Metrics, Apps, Deployment pipelines, Workspaces, My workspace, and Regional Sales Rep... A central dashboard 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 two charts: "Revenue Won by Month" (line chart) and "Close % by Month" (bar chart). To the right is a "Close % by Region" map of the United States. A sidebar titled "Copilot Preview" contains a list of AI-generated insights and actions:

- 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 text input field for asking questions and a note about AI-generated content.

Contoso Daily Sales | Data updated 1/12/23

File View Reading view Mobile layout

Sales Overview

Revenue Won: \$7,811,851 Close %: 37.7% AVG Days to Close: 121 Opportunities Won: 526

Revenue Won by Month

Close % by Month

Close % by Region

Summary of insights with Copilot

- Revenue trended up, resulting in a 490.36% increase in the 8 months ending in December, 2022 [1].
- Top 3 states by close % are Washington, Texas, and California [2].
- Revenue start trending sharply upward in November, 2022, rising by 71.57% (\$645,634) in 22 days [3].

Ask a question or request, or type '/' for suggestions

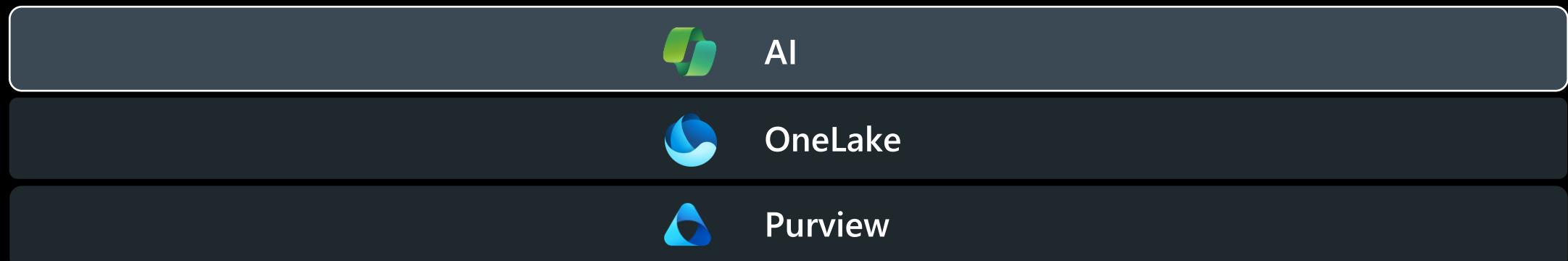
Power BI

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



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

Coordinator
(manager)

How many exams are planned for class X in the upcoming semester?

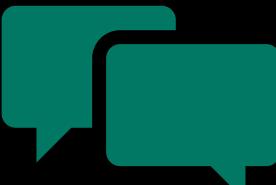


Teacher

Which students in class X can use some additional help with Mathematics?

Student

Which homework do I need to finish this week?



Mentor

Which of my students have been late in class this week?

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.

The screenshot shows the macaw AI Skills in Fabric interface. On the left, the Explorer sidebar lists various tables: Date, Enrollment, ExamParticipation, Group, GroupEnrollment, GroupMentor, LearningUnit, Location, OrganizationUnit, Program, ResultStructure, Results, Student, StudentObservations, and Test. The 'Results' table is selected. The main area contains a text input field with placeholder text 'Enter a question or request to test the model's response.' Below it, a note states 'Conversation history is off. The model does not retain context from previous queries. Note that AI-generated content can have mistakes. [Review terms](#)'. To the right, the 'Model behavior' section includes a 'SQL query variations' dropdown set to '3', a 'Show executed SQL query' toggle switch (on), and a 'Notes for model' section with a note about the 'student' table. The 'Example SQL queries' section shows '0 examples added'. At the bottom, a 'Preview data' section features a large circular icon with a clipboard symbol and the text 'Select a table' and 'Select a table from the explorer to preview it here.'

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.

The screenshot shows a dark-themed chat interface. At the top right, there are three icons: a triangle pointing right, a circle with a dot, and a cross. To the left of these is the text "Test your copilot". Below this is a message from the AI: "Hello, I'm EduChat, a virtual assistant. Just so you are aware, I sometimes use AI to answer your questions. If you provided a website during creation, try asking me about it! Next try giving me some more knowledge by setting up generative AI." This message is timestamped "Just now". Below the message is a text input field with a placeholder "Ask a question or describe what you need" and a character icon. The input field also shows "0/2000" and a send button with a right-pointing arrow. At the bottom, a note reads: "Make sure AI-generated content is accurate and appropriate before using. See terms".

▶ Test your copilot

... C X

Hello, I'm EduChat, a virtual assistant. Just so you are aware, I sometimes use AI to answer your questions. If you provided a website during creation, try asking me about it! Next try giving me some more knowledge by setting up generative AI.

Just now

Ask a question or describe what you need

0/2000

Make sure AI-generated content is accurate and appropriate before using. [See terms](#)

CI/CD & DTAP for Fabric



CI / CD

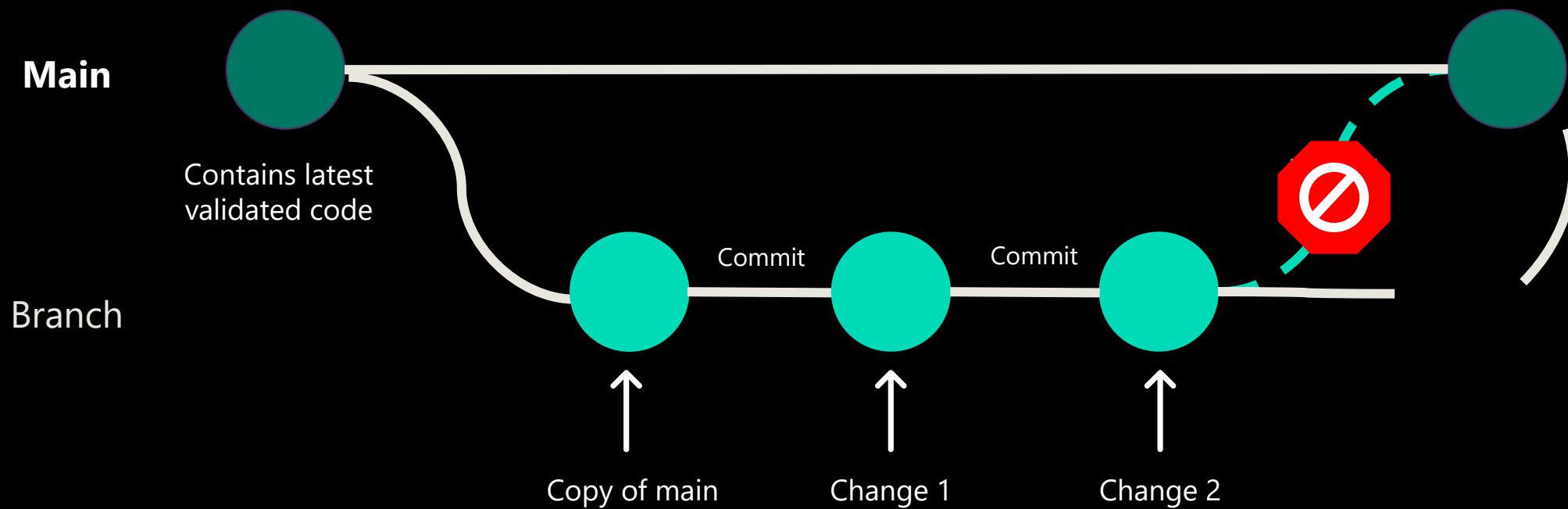
Continuous Integration

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

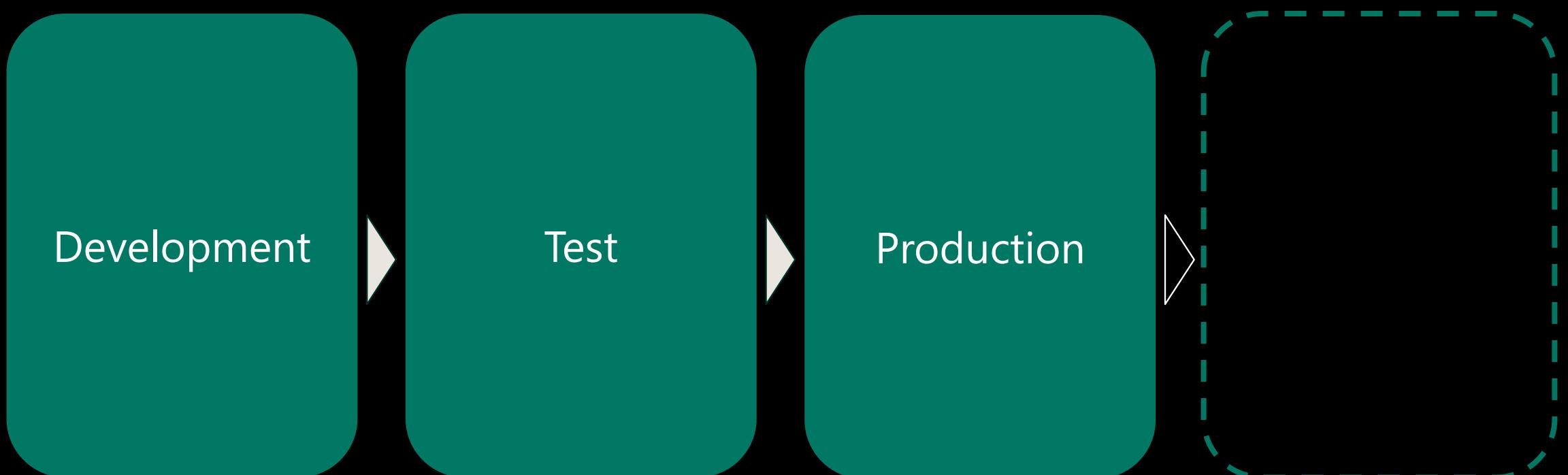
Continuous Deployment

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

Git concept



Staged approach



Fabric CI/CD platform

What do we have today?

Built-in git integration

NB_LH_demo

Name	Git status	Type	Owner	Refreshed	Next ref
Orders_LH	Synced	Lakehouse	Nimrod Shalt	—	—
Orders_LH	Unsupported	Semantic model (...)	NB_LH_demo	1/22/24, 11:18:46 AM	N/A
Orders_LH	Unsupported	SQL analytics end... NB_LH_demo	—	—	N/A
Sales orders - read & transform NB	Uncommitted	Notebook	Nimrod Shalt	—	—
Sales orders - write to target NB	Synced	Notebook	Nimrod Shalt	—	—

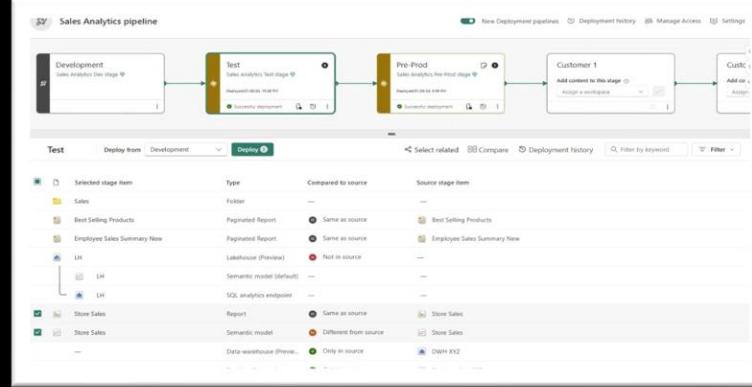
Source control

Commit message

Commit Undo

Public preview

Deployment pipelines



GA feature

Fabric REST APIs

CORE APIs – Fabric generic APIs

WL specific APIs

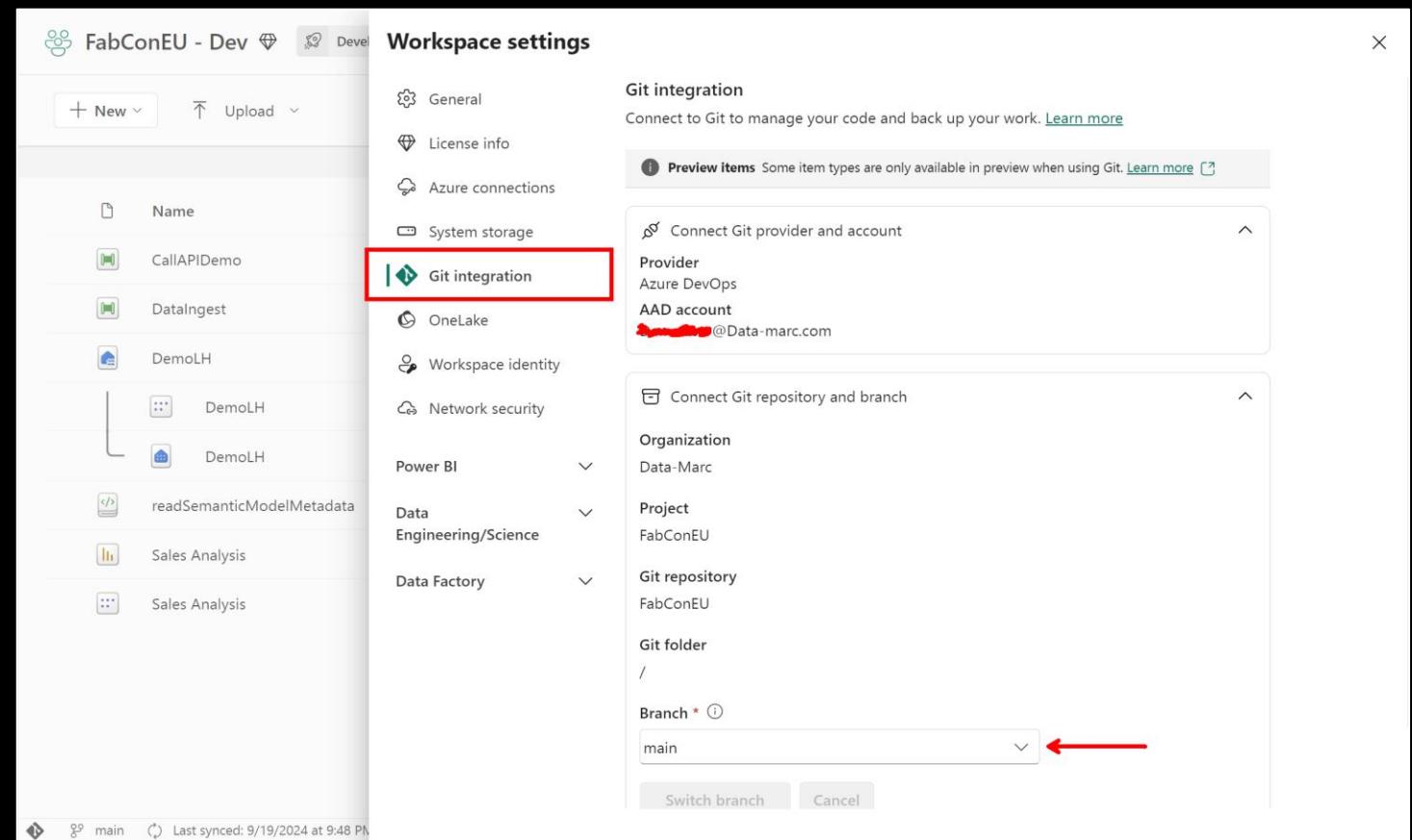
Industry integrations XMLA, TDS, Apache Livy, MLFlow ...

Fabric Public API surface

Public preview

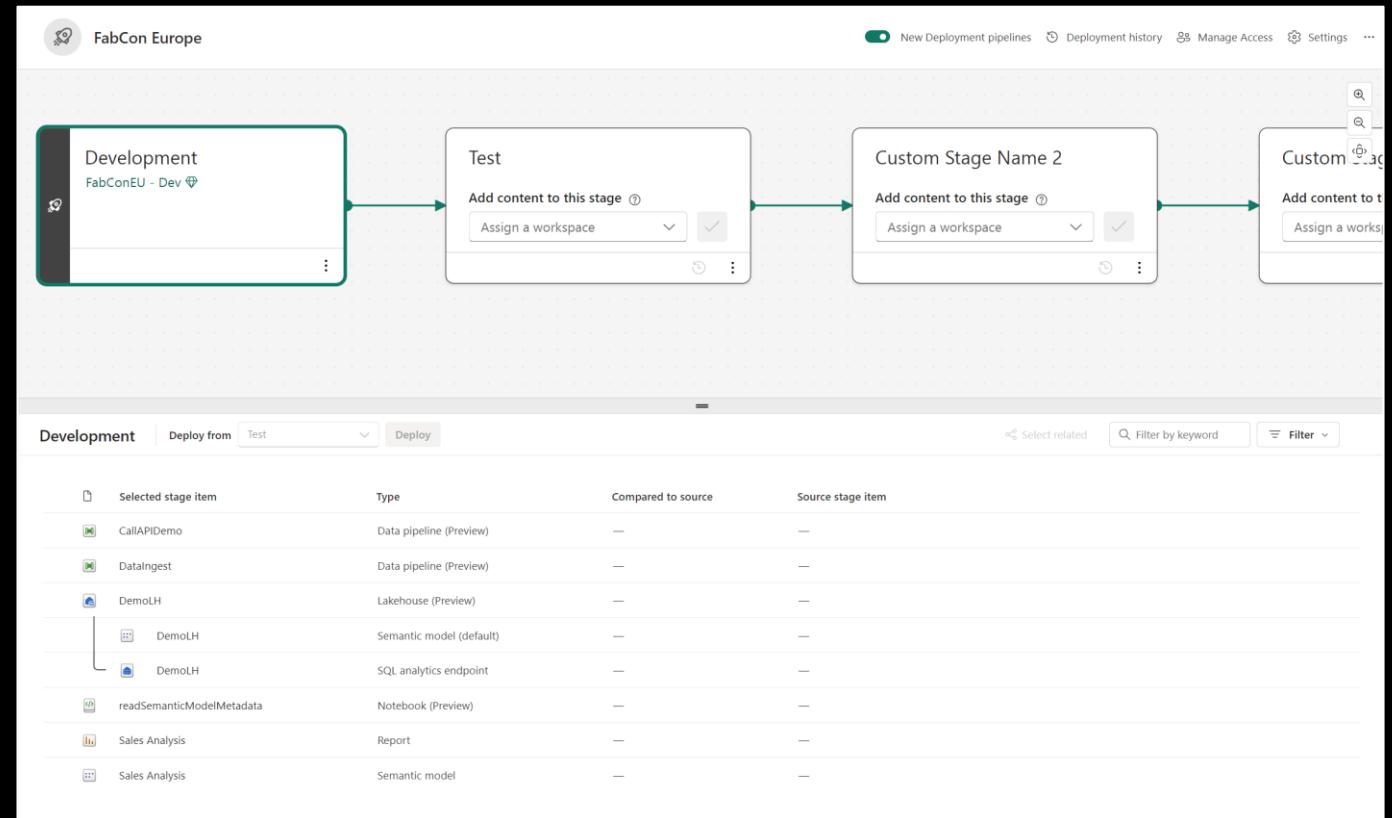
Git integration

- 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



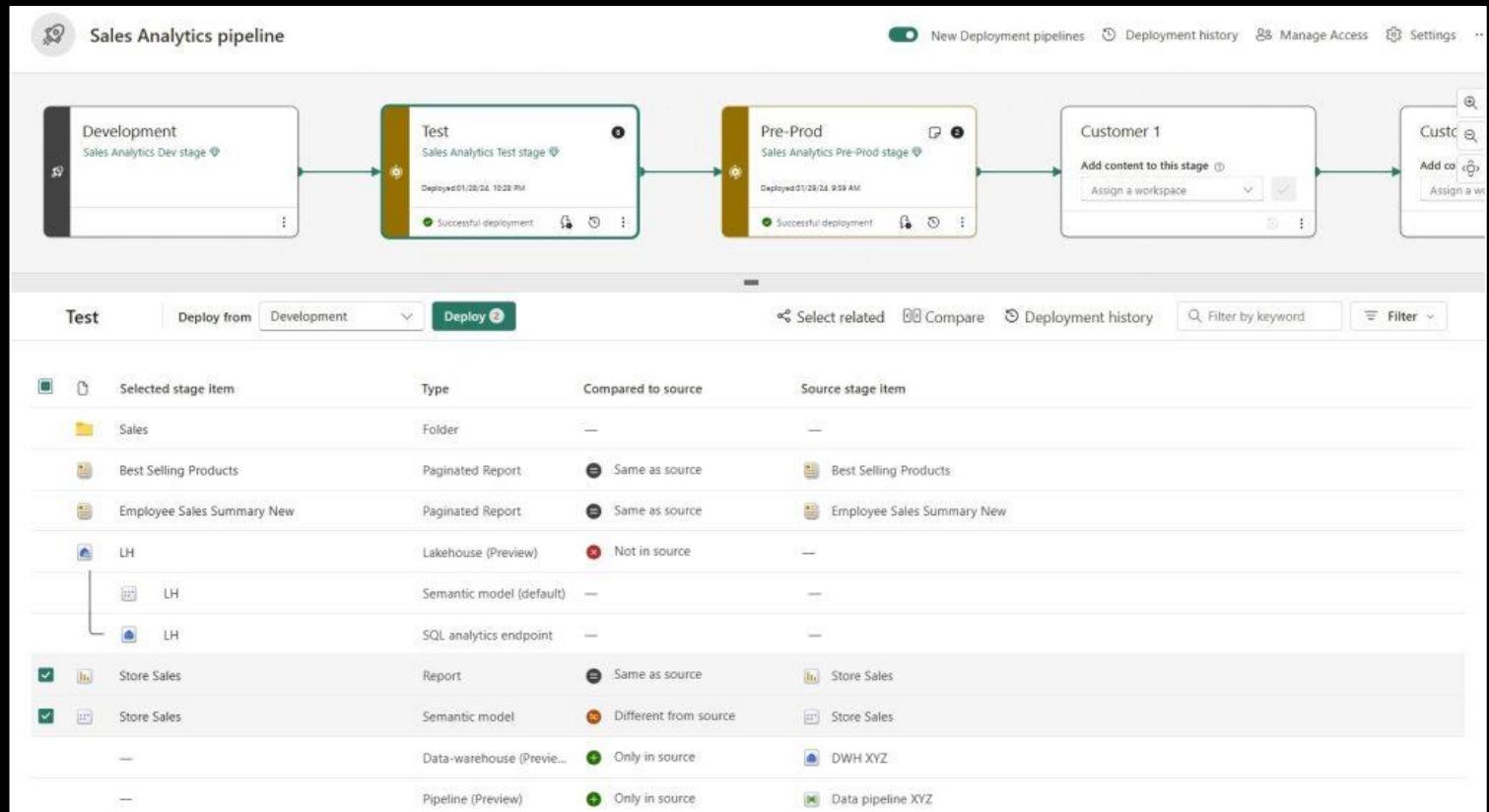
Fabric User APIs

- 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

The screenshot shows the 'Workspace settings' page in the Azure DevOps portal. On the left, there's a sidebar with a search bar and links for General, License info, Azure connections, System storage, Git integration (which is selected and highlighted in green), and OneLake. The main content area is titled 'Git integration' and includes a sub-section '(Preview)'. It explains that users can sync workspace items with GitHub repositories and select GitHub as their provider. A toggle switch is shown as 'Enabled'. Below this, there's a 'Connect Git provider and account' section with two options: 'Azure DevOps' and 'GitHub', where 'GitHub' is also highlighted in green.

Git integration

Users can sync workspace items with **GitHub** repositories
Enabled for the entire organization

Users can select **GitHub** as their Git provider and sync items in their workspaces with **GitHub** repositories.

Enabled

Git integration (Preview)

Connect to Git to manage your code and back up your work. [Learn more](#)

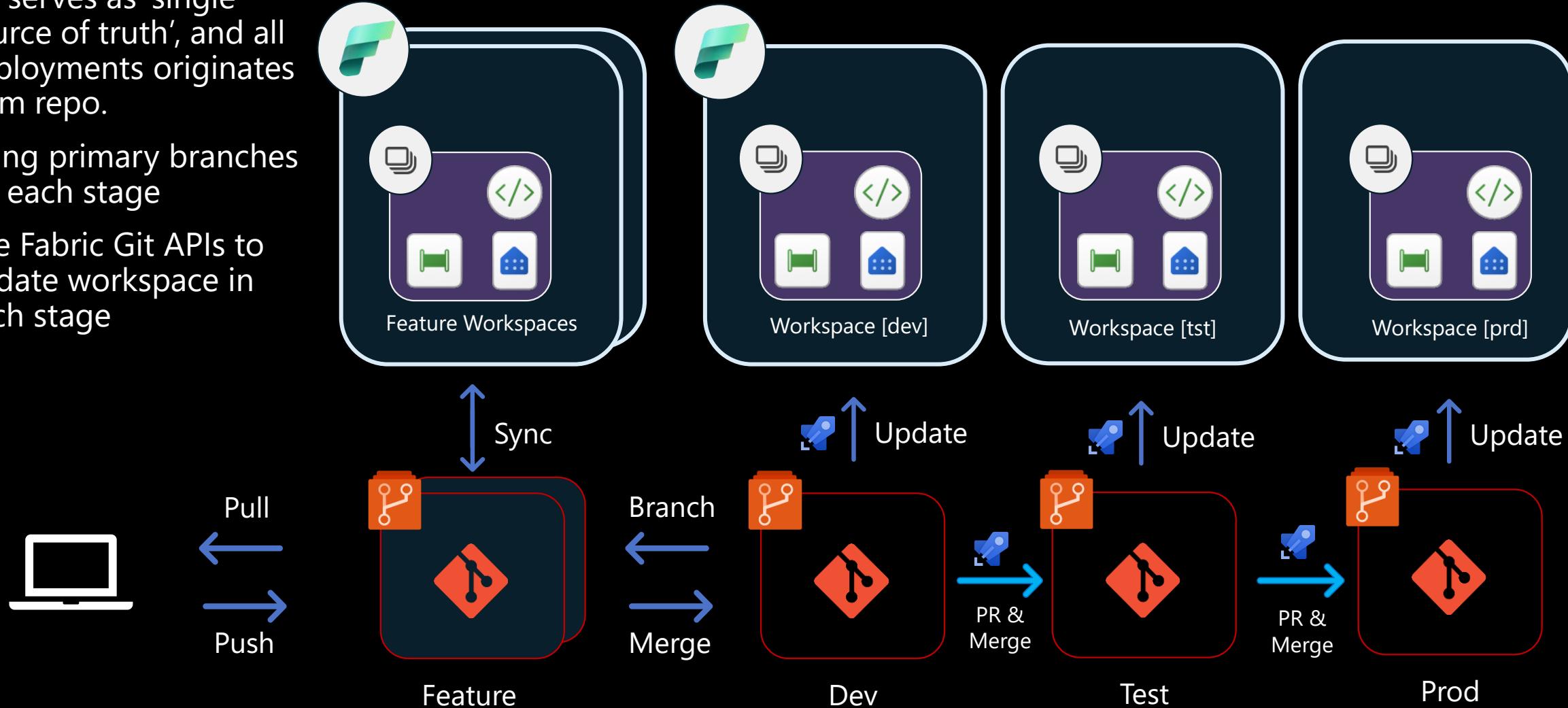
Connect Git provider and account

Git provider

Azure DevOps GitHub

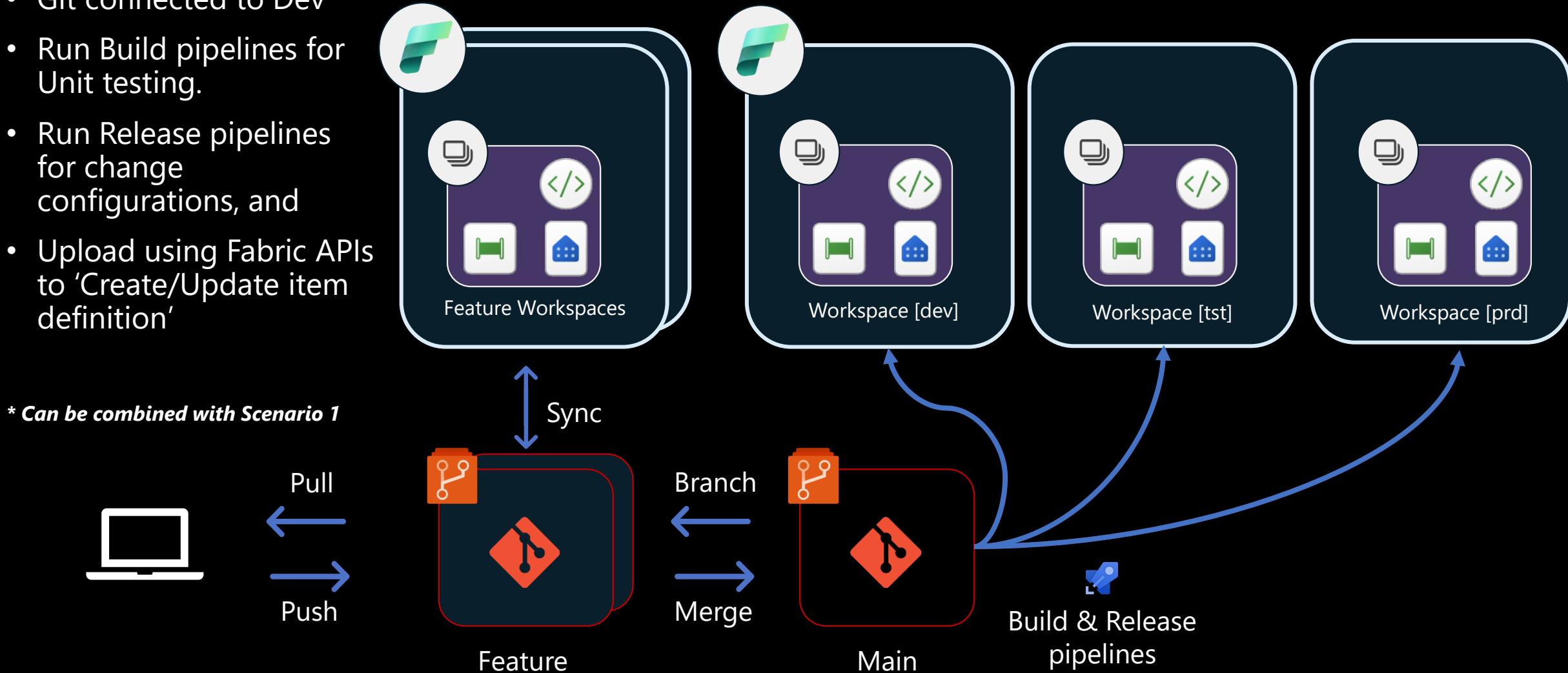
Scenario 1 – Git based deployments

- 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



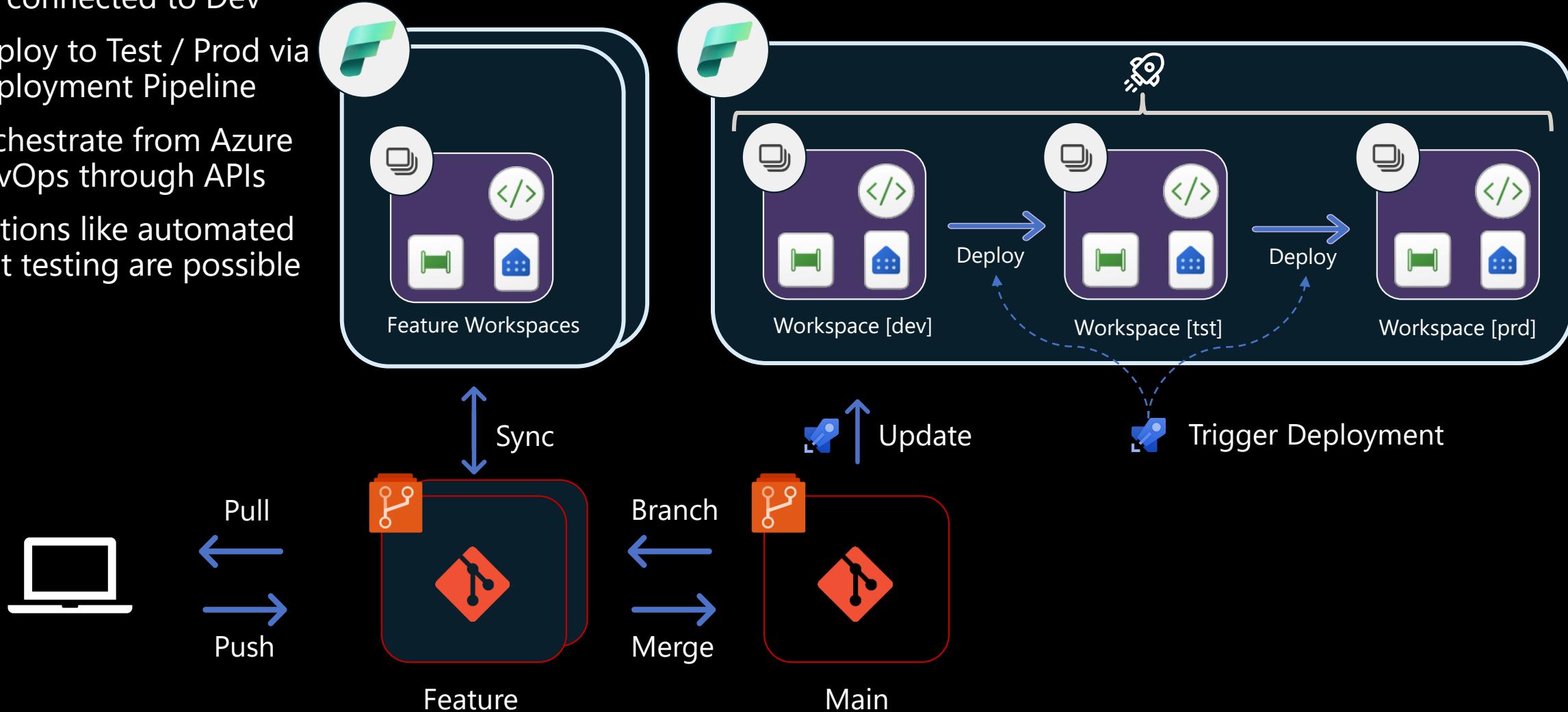
Scenario 2* – Git & Build environments

- 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'



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

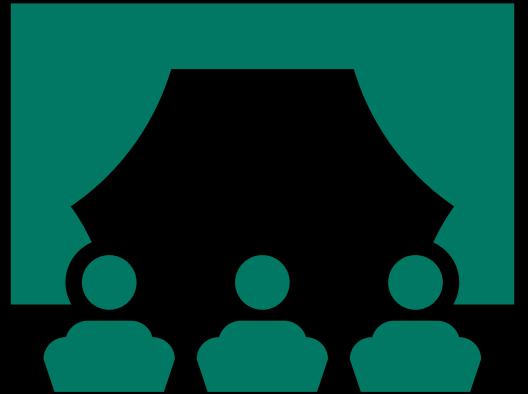


Opinionated view!

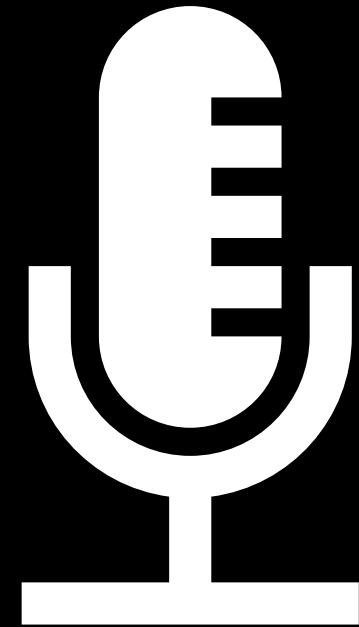
Entry point	Scenario	Code heaviness
Prefer an user interface?	Start with deployment pipelines interface, grow into scenario 3 over time	Low
Comfortable working with git/code?	Your go-to scenario will be scenario 1 .	Medium
Require a lot of customization?	Using the APIs in scenario 2 allows you to customize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

