

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



Objectives



- Data platform challenges
- Medallion Architecture
- Parameters
- Framework
- Semantic Model & DirectLake
- Copilot (all the things)

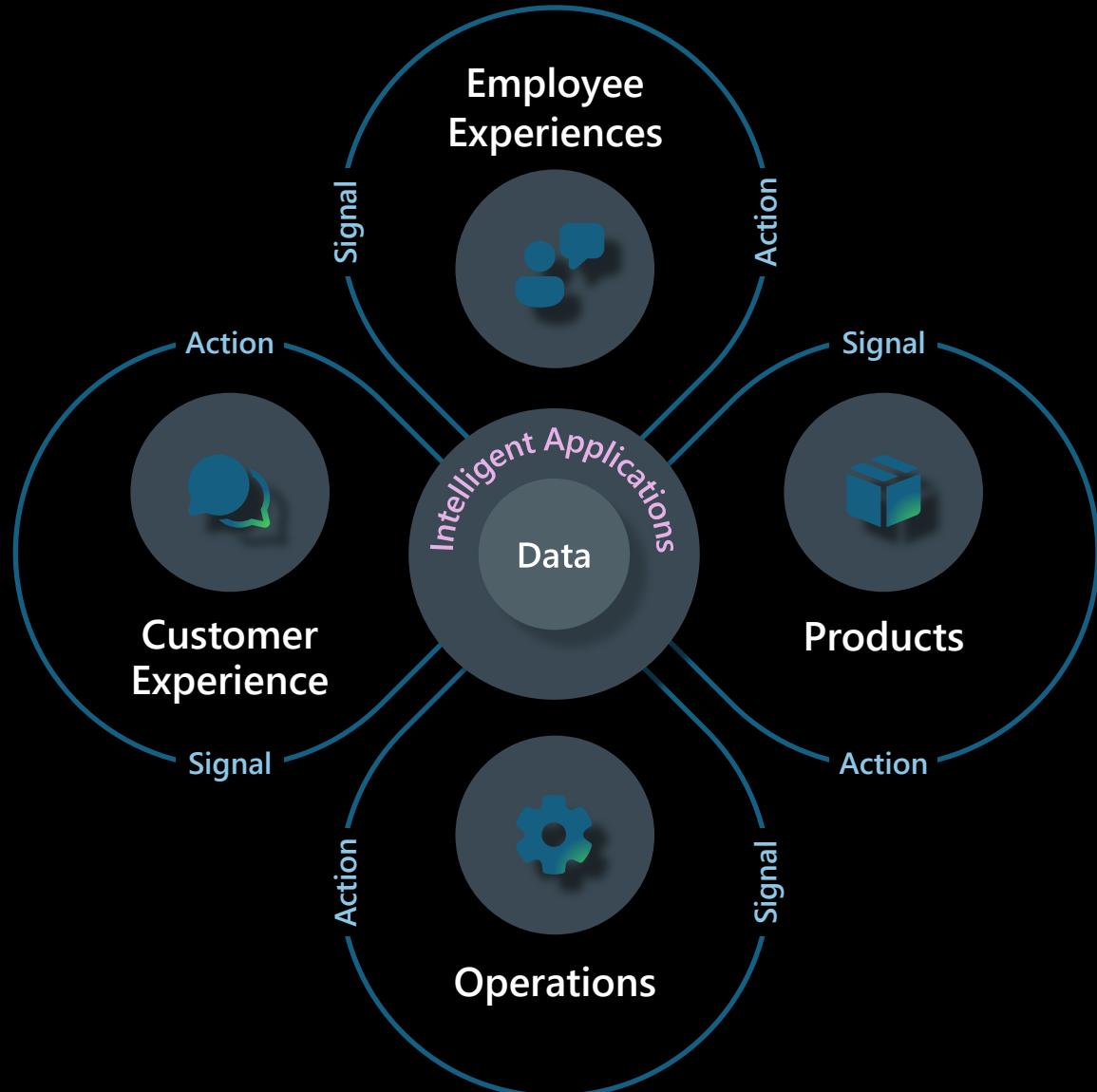
1

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

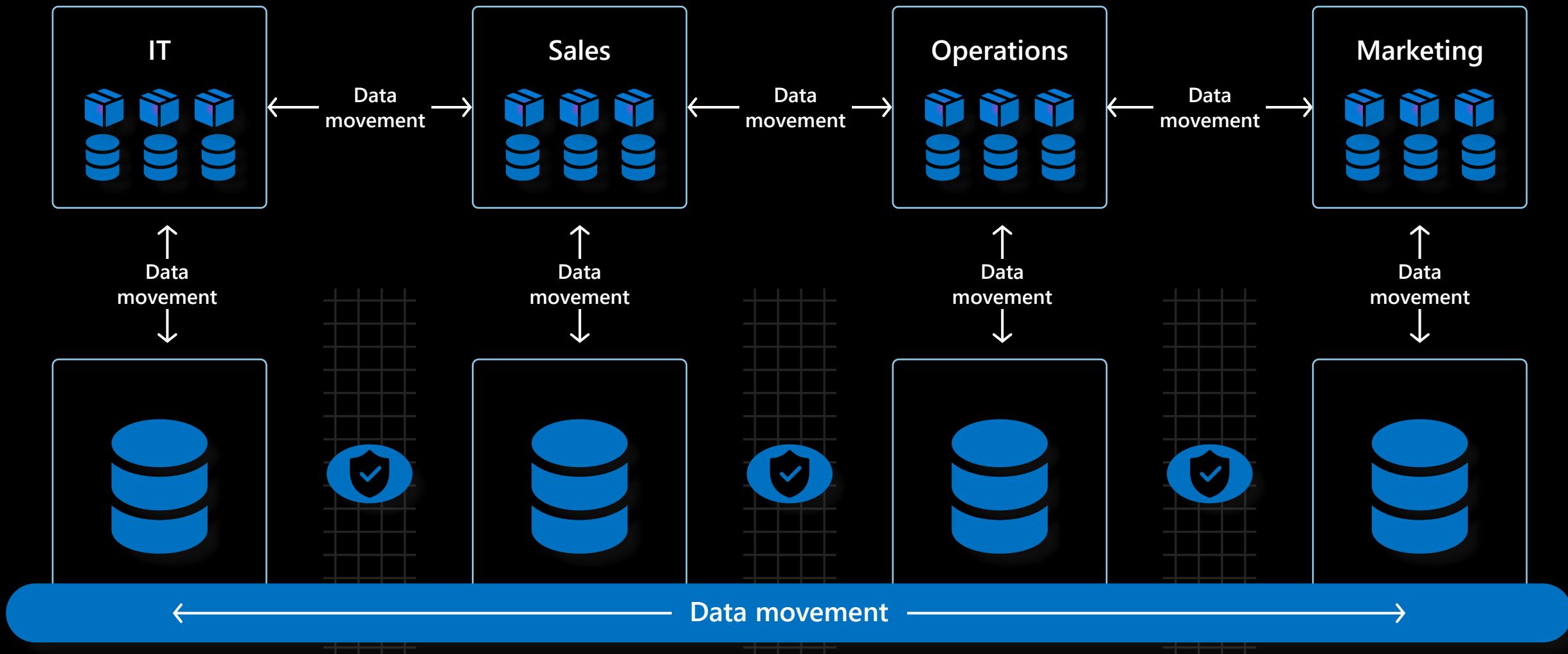


Gaining BI adoption to
streamline data sharing, collaboration



Data and security risks,
Governance & Compliance

Starting line: Multiple siloed lakes with lots of duplication



Typical sentiment about data-backed decision processes

*"Too difficult
to get to the
data"*

*"Data is
highly
fragmented"*

*"We cannot
re-use what
we have"*

*"We do not
have that
expertise"*

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

The unified data platform for the era of AI



Data
Factory



Synapse Data
Engineering



Synapse Data
Science



Synapse Data
Warehousing



Synapse Real
Time Analytics



Power BI



Data
Activator



AI



OneLake



Purview

Unified
architecture

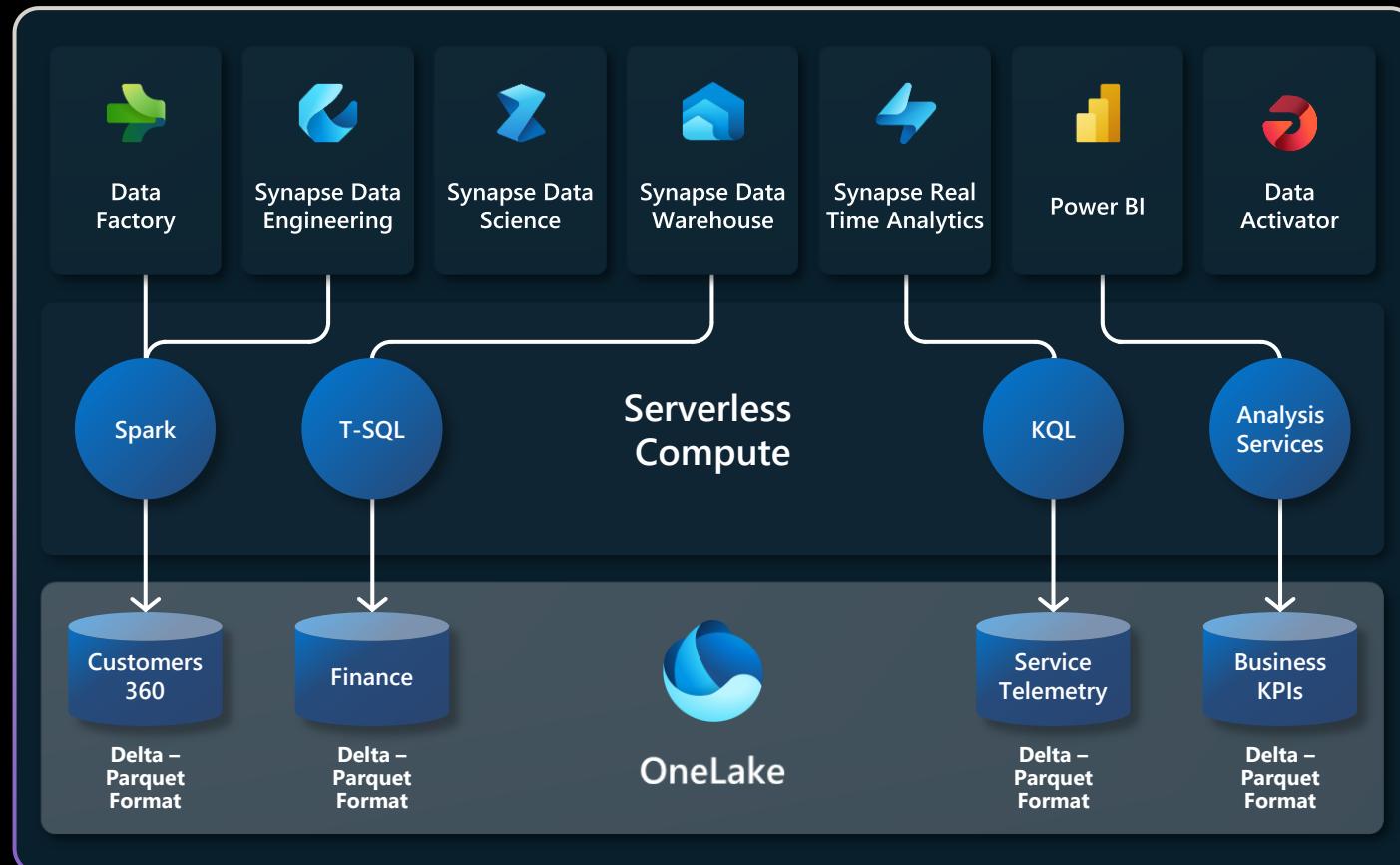
Unified
experience

Unified
governance

Unified
business model

One Copy for all computers

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

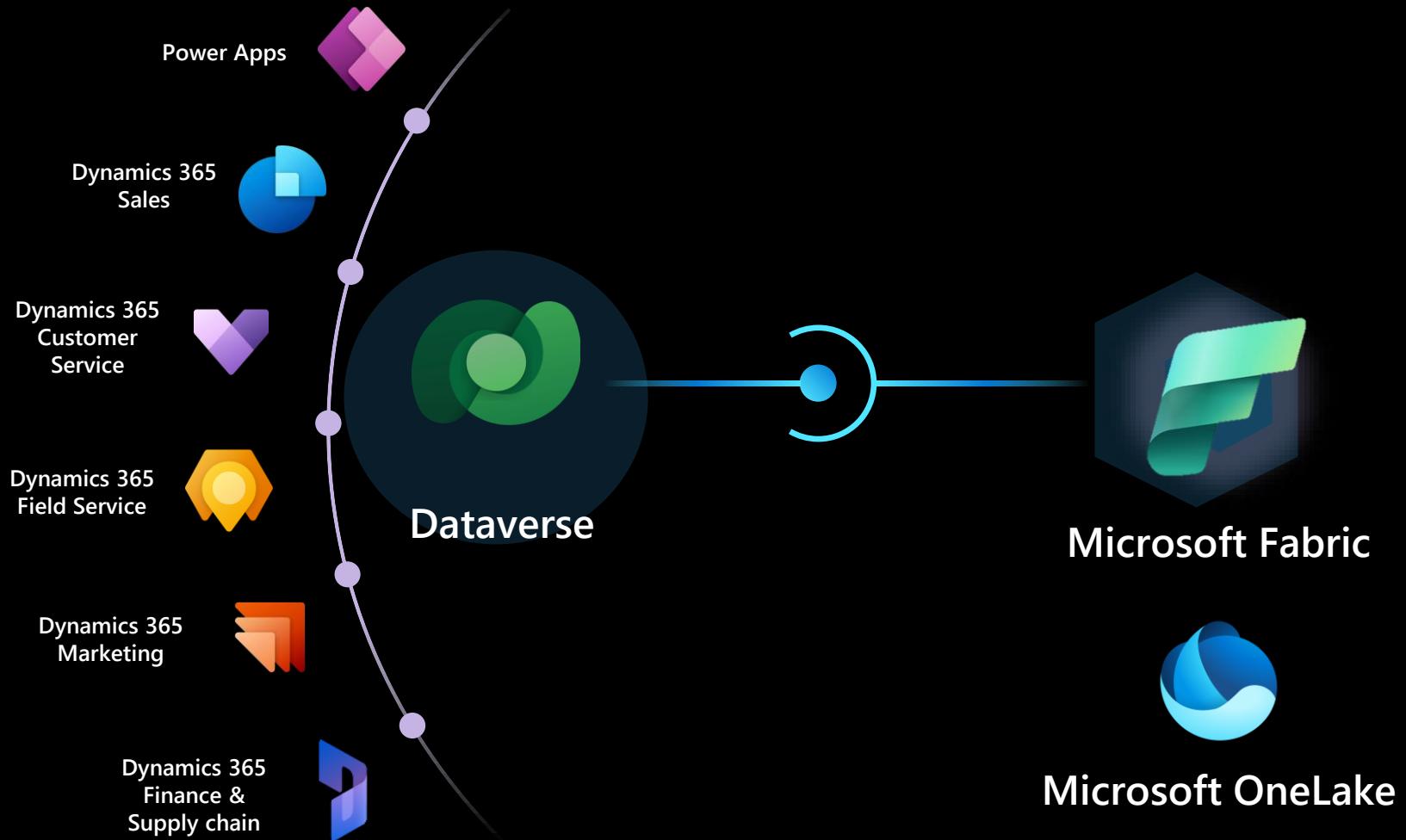
Once data is stored in the lake, it is directly accessible by all the engines without needing any import/export

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

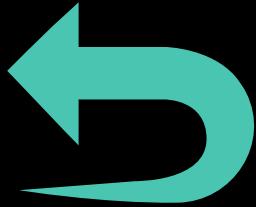
The value in your ERP data



Bringing your data to Fabric



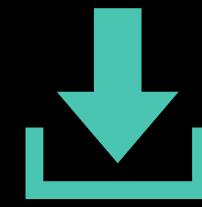
Concepts to bring data to Fabric.



Shortcuts



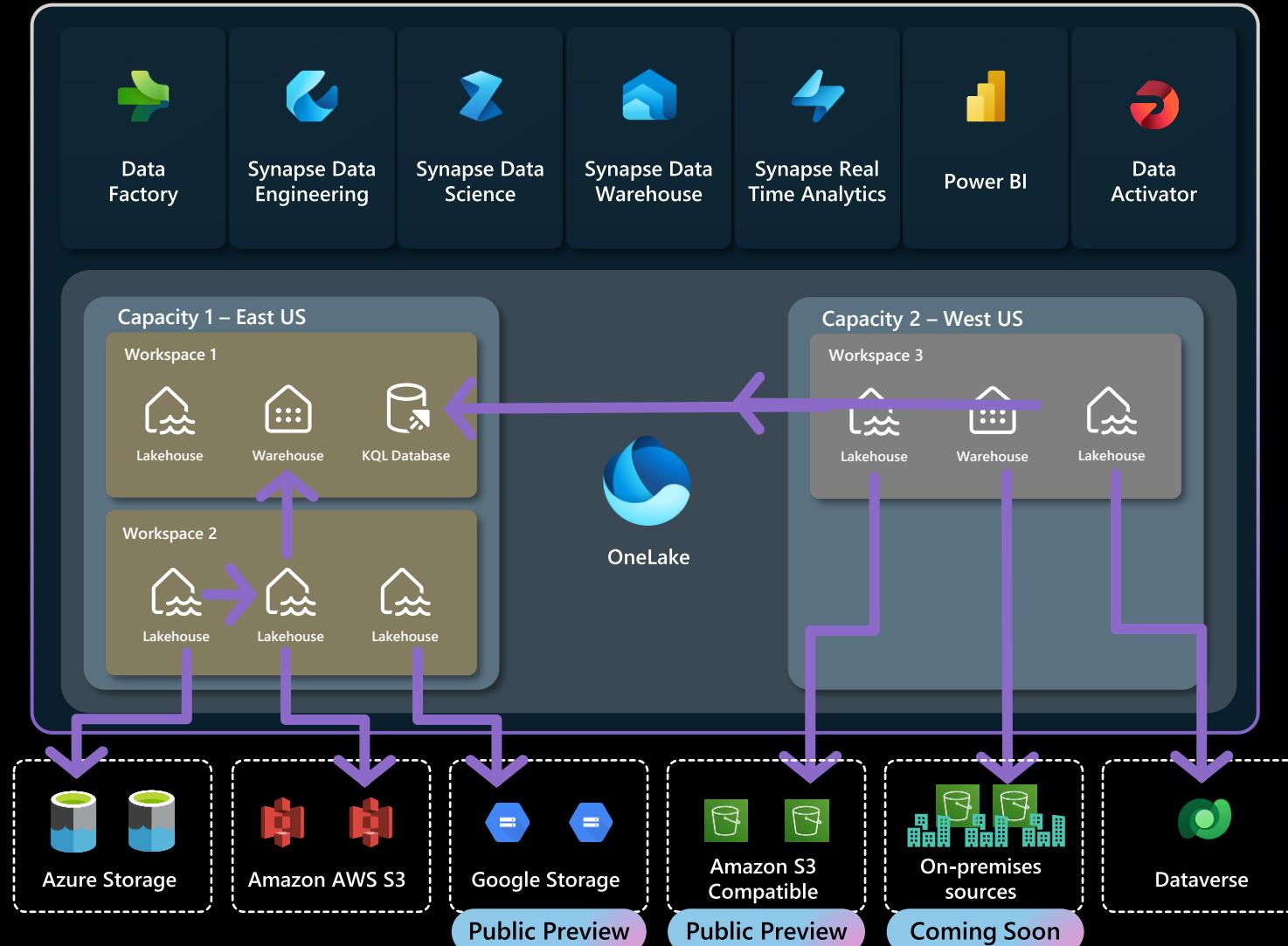
Mirror



Import

Unifying data in OneLake

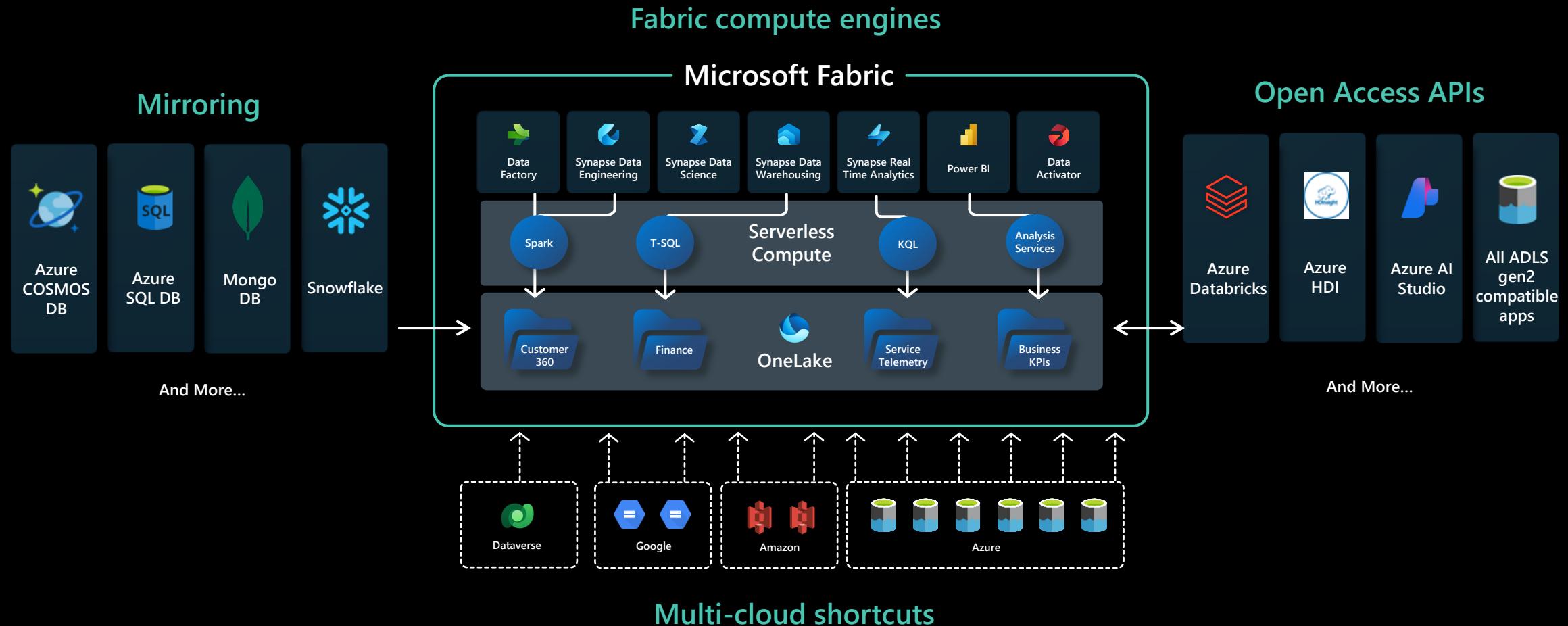
External Data Sharing (Preview)



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

All roads lead to OneLake

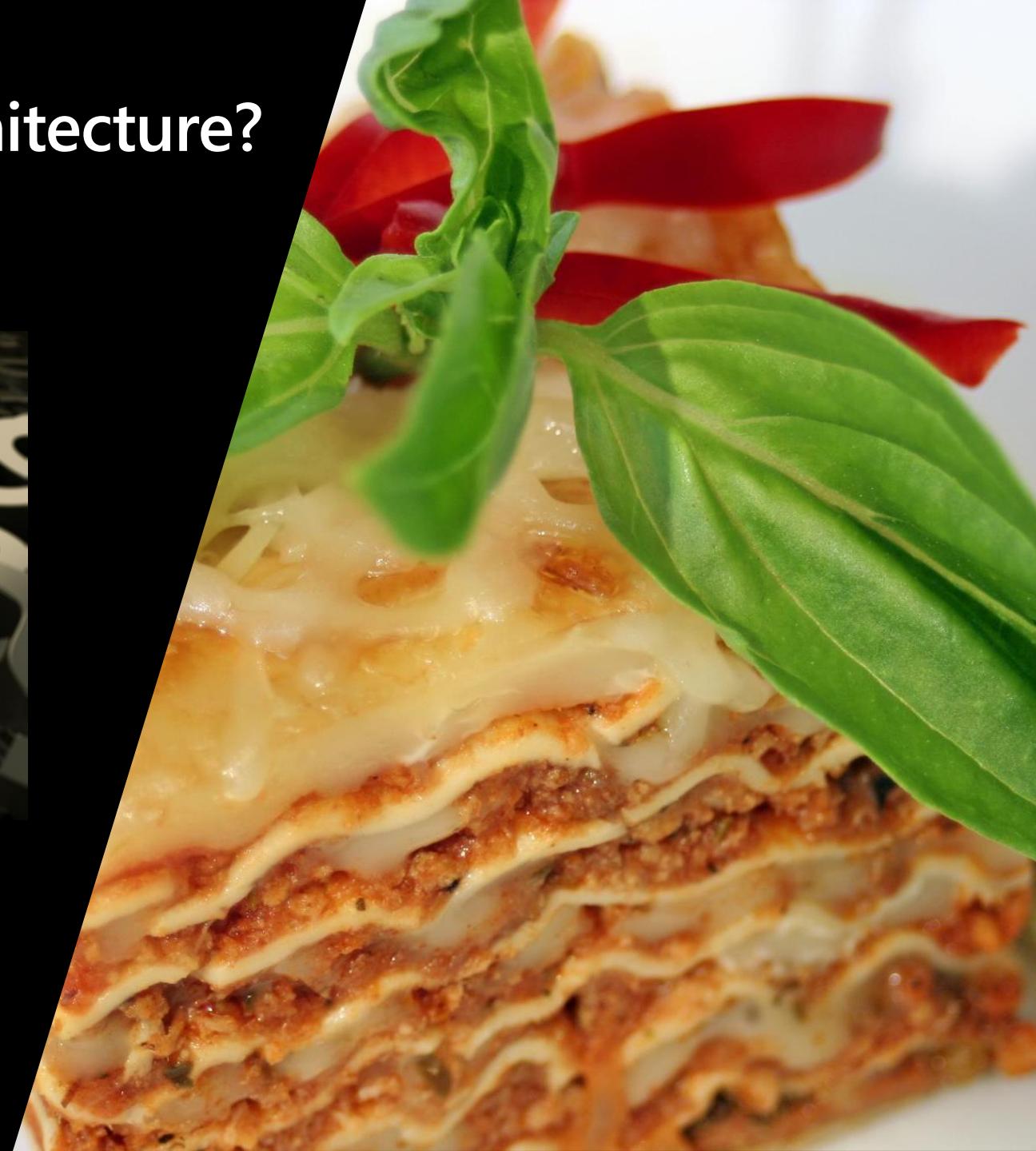
Creating Data Gravity in OneLake



Who is using a Medallion architecture?



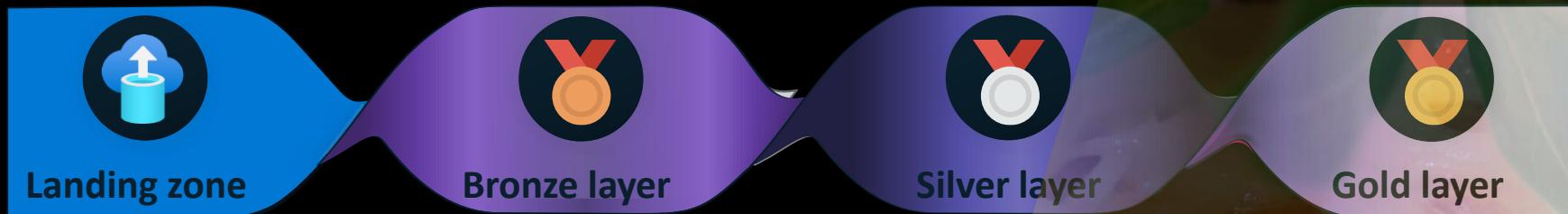
'Uniform data architecture'
From data "Spaghetti to Lasagna"



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



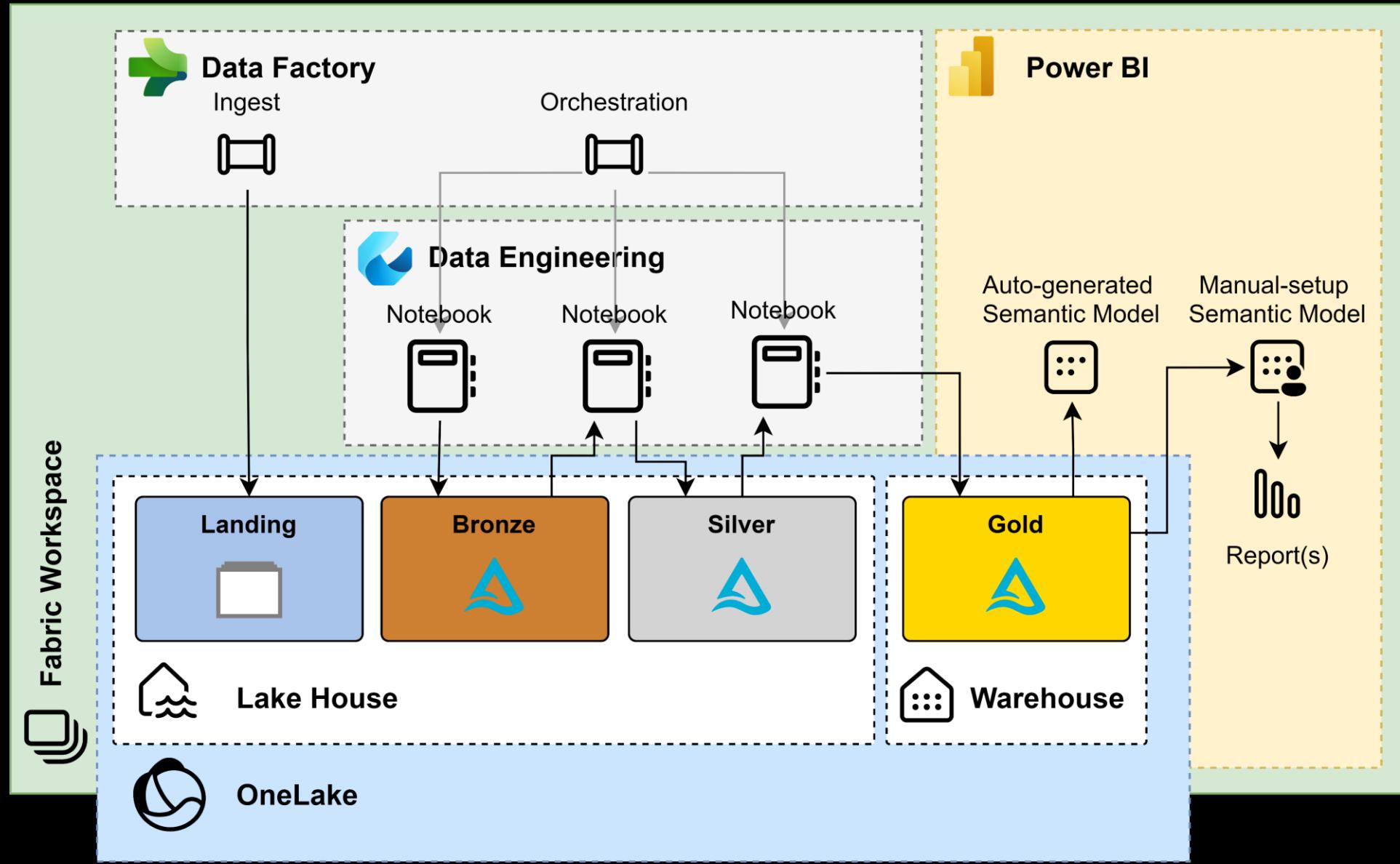
Landing zone

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





Microsoft Fabric



Demo



Understand Parameters

- 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

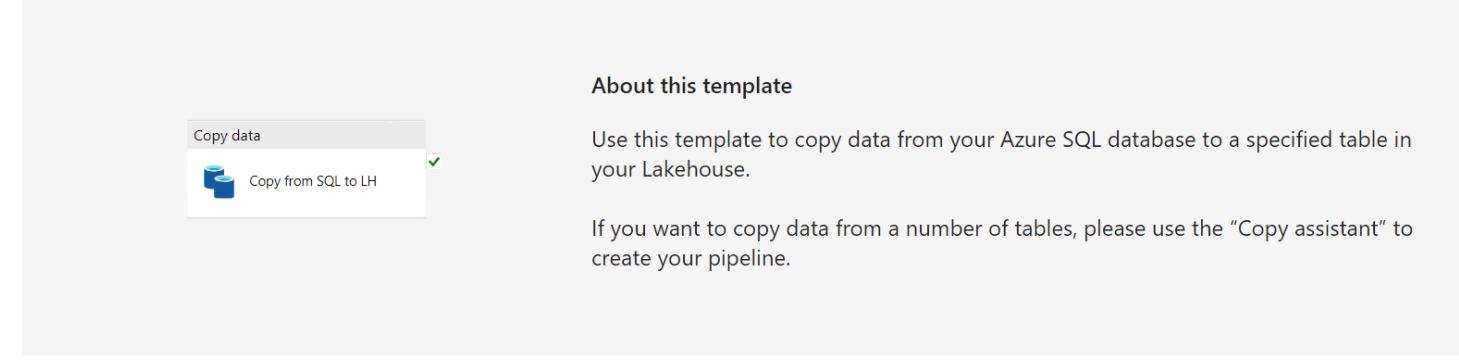
Use this template to copy data from your Azure SQL database to a specified table in your Lakehouse.

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

Inputs

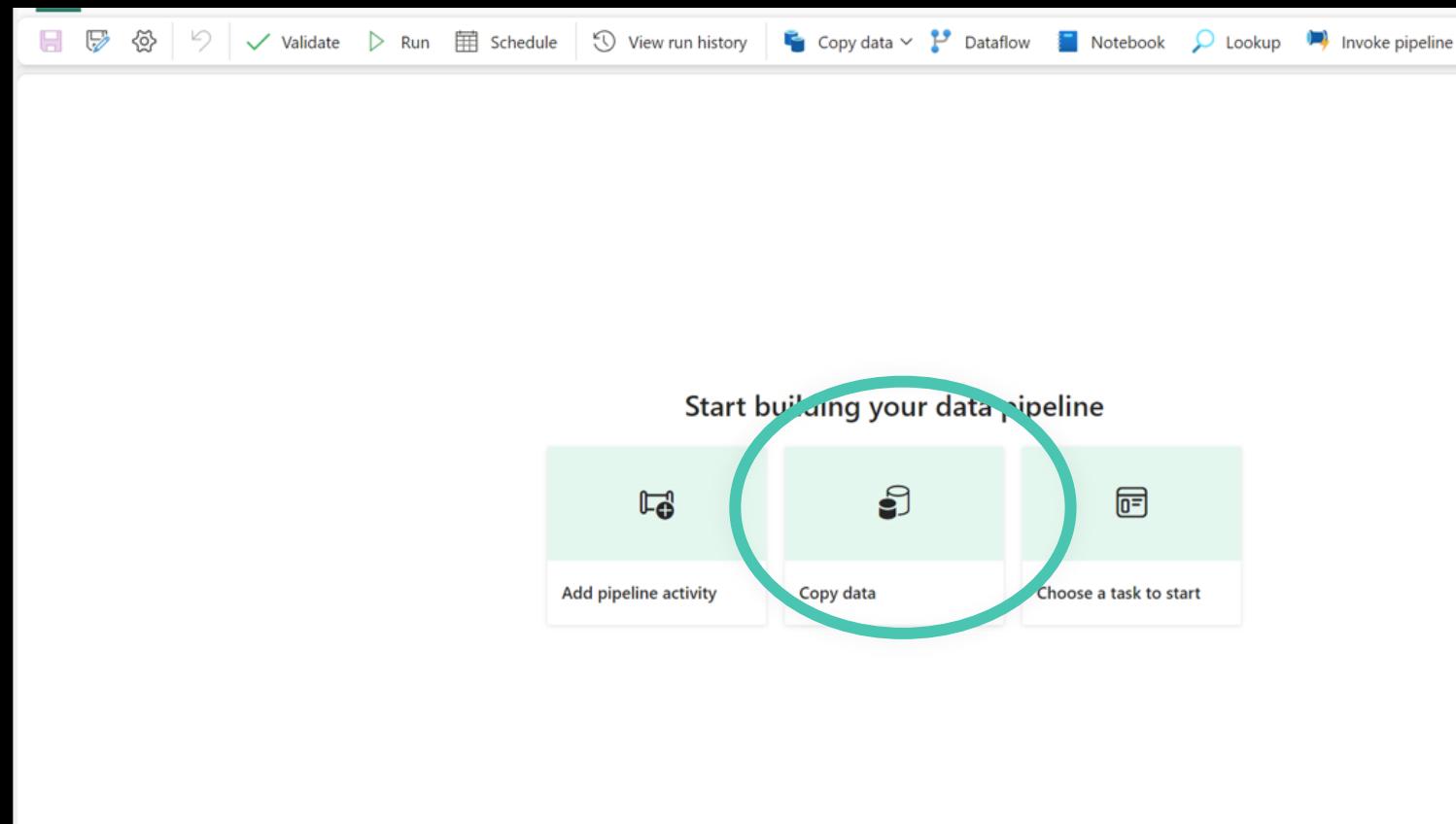
Connection *
For Copy from SQL to LH (Copy source)
Select... (refresh) (new)

Lakehouse *
For Copy from SQL to LH (Copy destination)
Select... (refresh) (new)



Understand Parameters

- Copy Assistant



Understand Parameters

- Copy Assistant
- Select Data Source
- Select Tables
- Select Destination
- Select Lakehouse
- Select Filetypes
- Data is stored as Delta Parquet

The screenshot shows the Azure Data Factory interface for configuring a Copy Activity. On the right, a large JSON configuration block is displayed, detailing multiple copy operations:

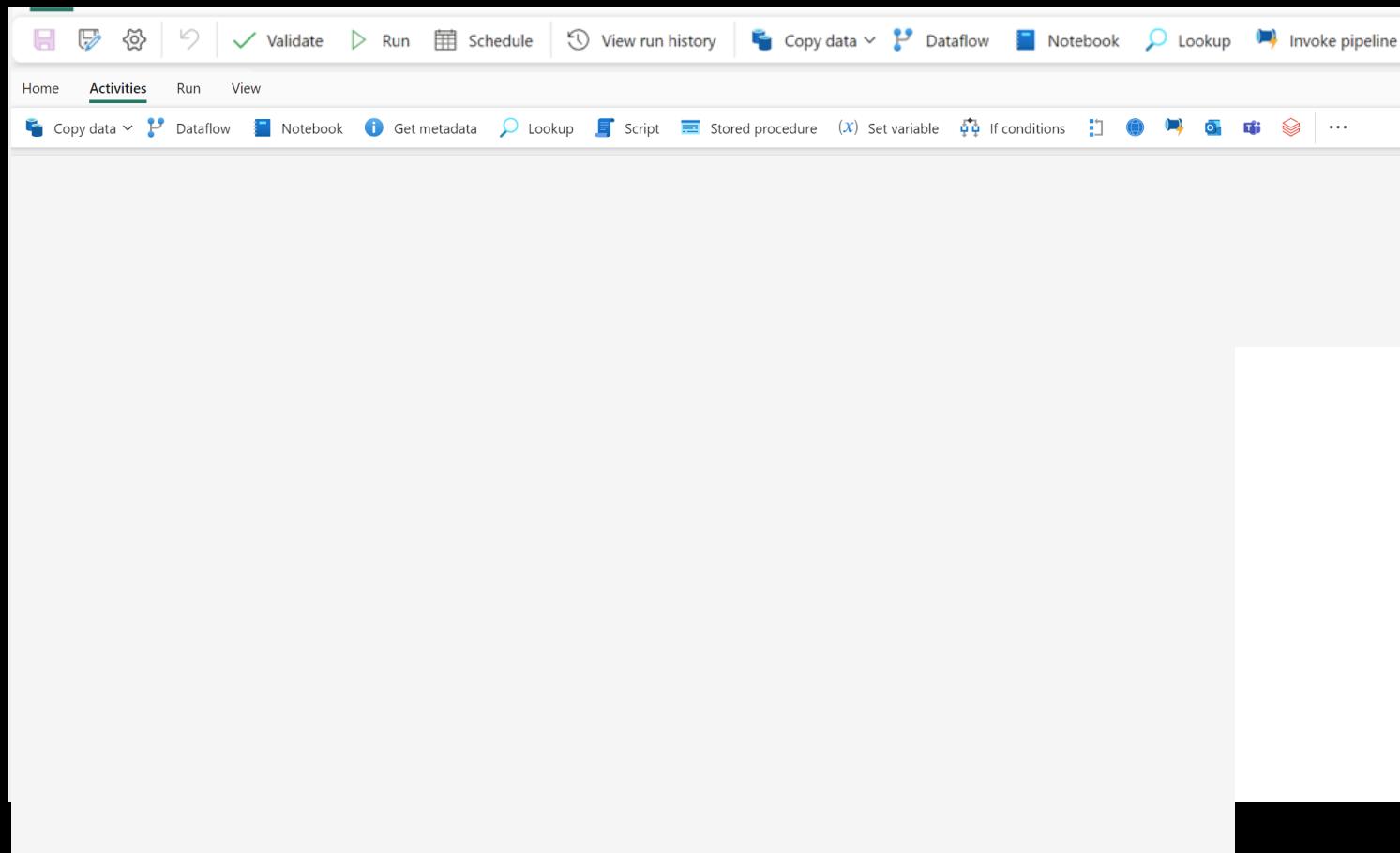
```
{  
    "source": {  
        "table": "Address"  
    },  
    "destination": {  
        "fileName": "SalesLTAddress.parquet"  
    }  
},  
{  
    "source": {  
        "table": "Customer"  
    },  
    "destination": {  
        "fileName": "SalesLTCustomer.parquet"  
    }  
},  
{  
    "source": {  
        "table": "CustomerAddress"  
    },  
    "destination": {  
        "fileName": "SalesLTCustomerAddress.parquet"  
    }  
},  
{  
    "source": {  
        "table": "Product"  
    },  
    "destination": {  
        "fileName": "SalesLTProduct.parquet"  
    }  
},  
{  
    "source": {  
        "table": "ProductCategory"  
    },  
    "destination": {  
        "fileName": "SalesLTProductCategory.parquet"  
    }  
},  
{  
    "source": {  
        "table": "ProductDescription"  
    },  
    "destination": {  
        "fileName": "SalesLTProductDescription.parquet"  
    }  
},  
{  
    "source": {  
        "table": "Cities"  
    },  
    "destination": {  
        "fileName": "SalesLTAddress.parquet"  
    }  
}  
}
```

Below the JSON, the 'Parameters' tab of the configuration pane is selected. It shows a single parameter named 'LoadDataLanding' of type 'Array'. A green arrow points from the 'Default value' field, which contains the JSON array shown above, towards the JSON configuration block.

Parameters	Variables	Settings	Output
+ New <input type="button" value="Delete"/>			
<input type="checkbox"/> Name	Type	Default value	
<input type="checkbox"/> LoadDataLanding	Array	[{"source": {"table": "Cities"}, "destination": {"fileName": "SalesLTAddress.parquet"}}]	<input type="button" value="Delete"/>

Understand Parameters

- Or just start from a blank Canvas

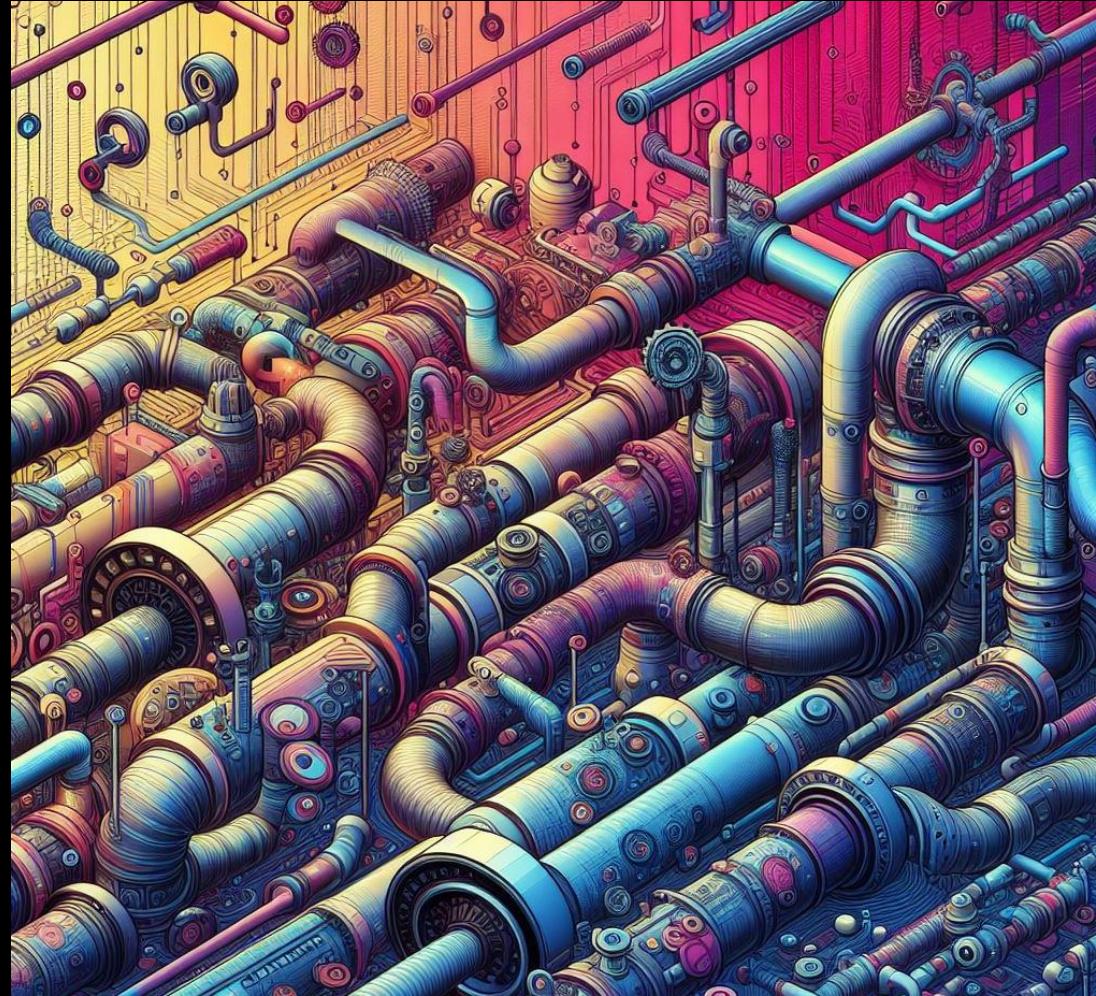


Parameters

- Pipeline Parameters
- Notebook Parameters
- Copy Activity Parameters

IMPLEMENTING
DEFAULT PARAMETERS
THAT DEPEND ON
OTHER PARAMETERS

Custom-Made Framework



Frameworks



Automation



Flexibility

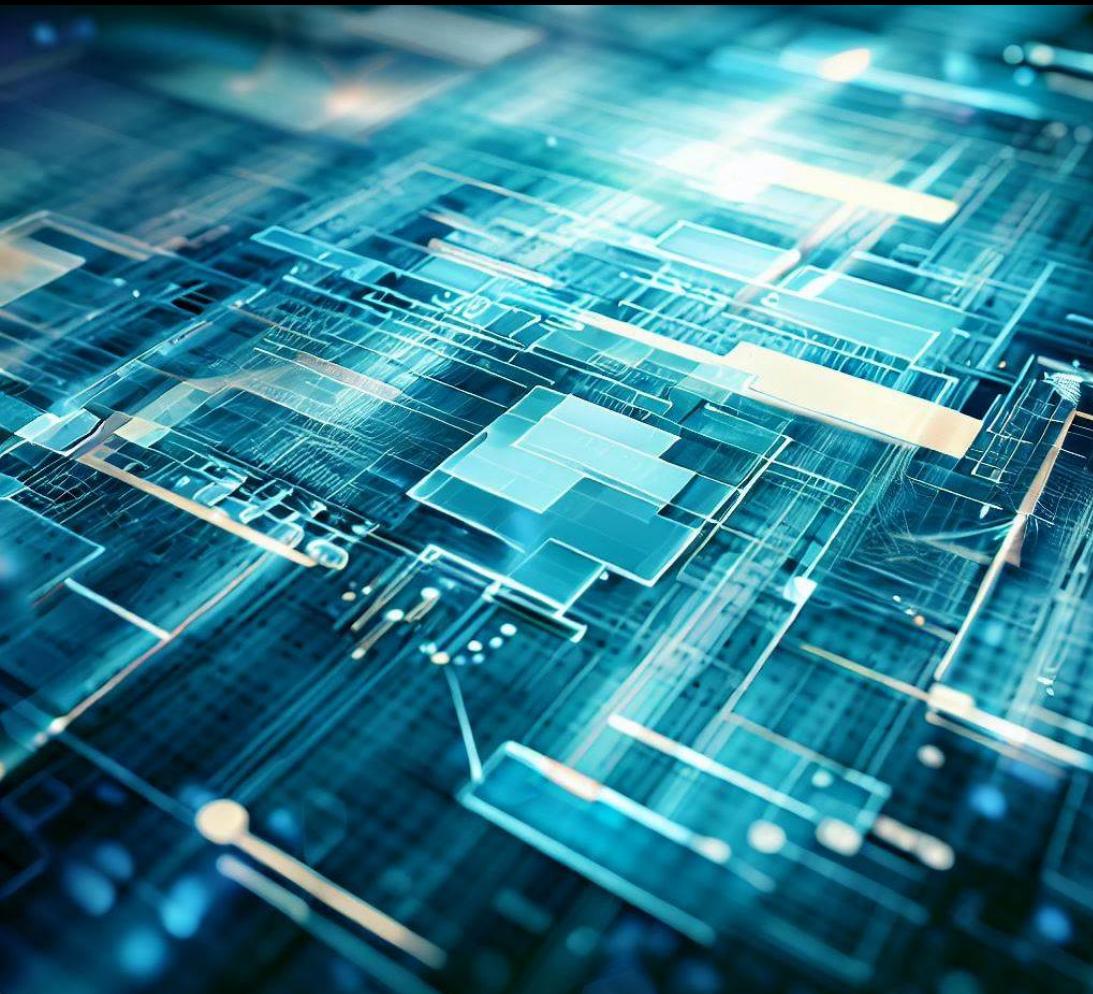


Scalability



Traceability

Out-of-the-Box Framework



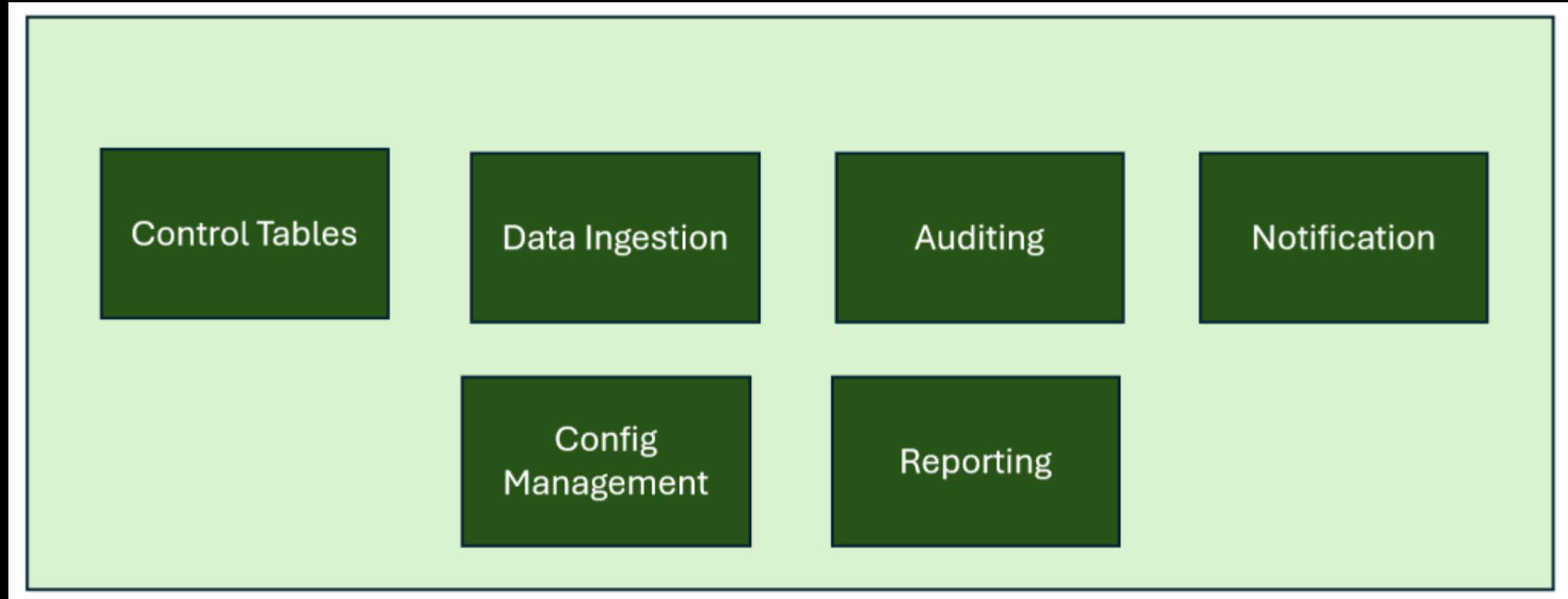
- Ready-to-use.
- Rapid implementation.
- Limited customization.
- Lower development effort.
- Lower upfront costs.
- Ongoing support and updates.

Custom-Made Framework

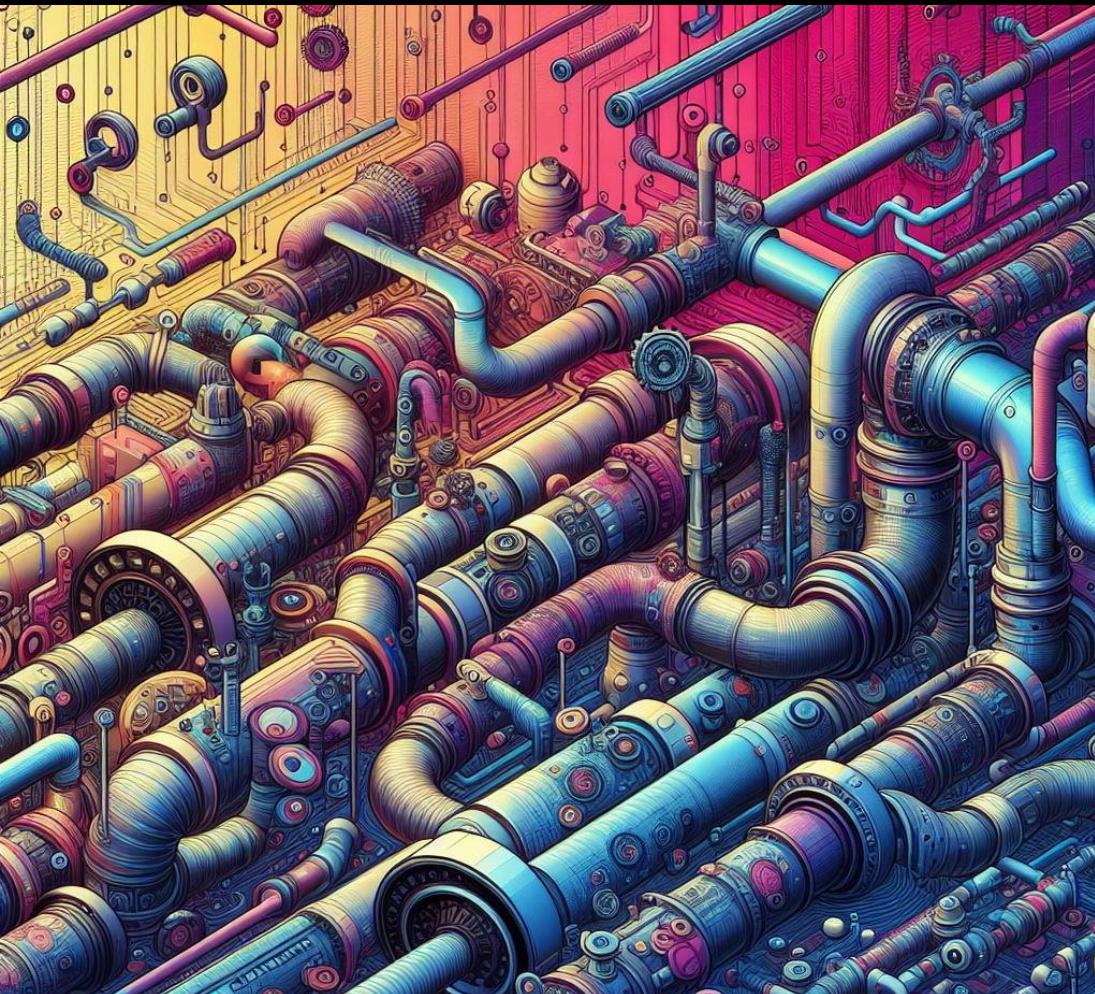


- Tailored to specific needs.
- Full control over design and features.
- Higher development effort.
- Flexibility and extensibility.
- Higher upfront costs.

Custom-Made Framework

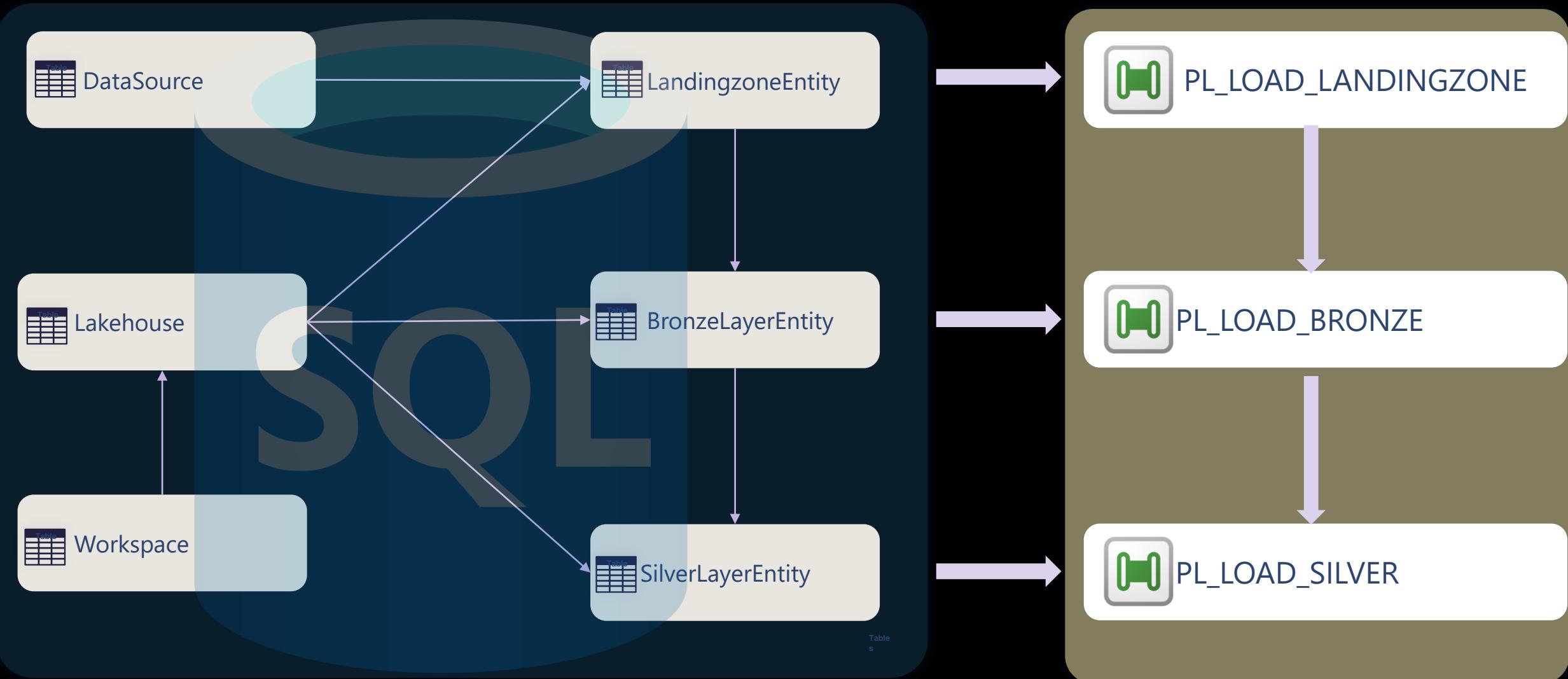


Custom-Made Framework



- 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

Framework



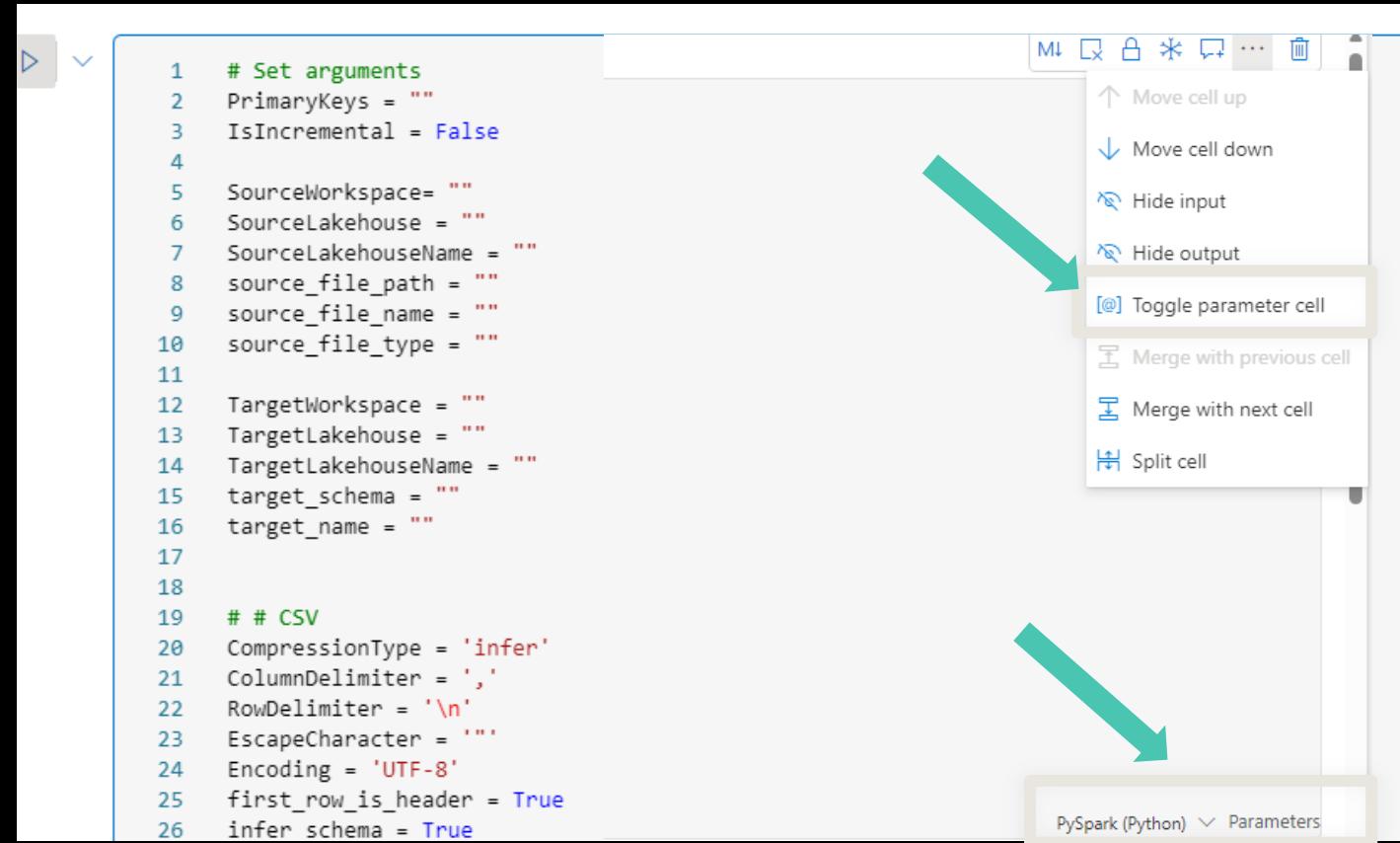
Define Pipeline Parameters

- Define Parameters
 - Pass through from Pipeline to Pipeline
 - Define Metadata
 - Versions
 - Key Vault
 - Secrets

<input type="checkbox"/> Name	Type	Default value	
<input type="checkbox"/> location	String	West-Europe	
<input type="checkbox"/> key_vault_name	String	demokeeuwdvlmvaultoxgnl	
<input type="checkbox"/> tenant_id	String	c183ff6a-8ca1-4185-9378-e	
<input type="checkbox"/> Environment	String	Development	

Define Notebook Parameters

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

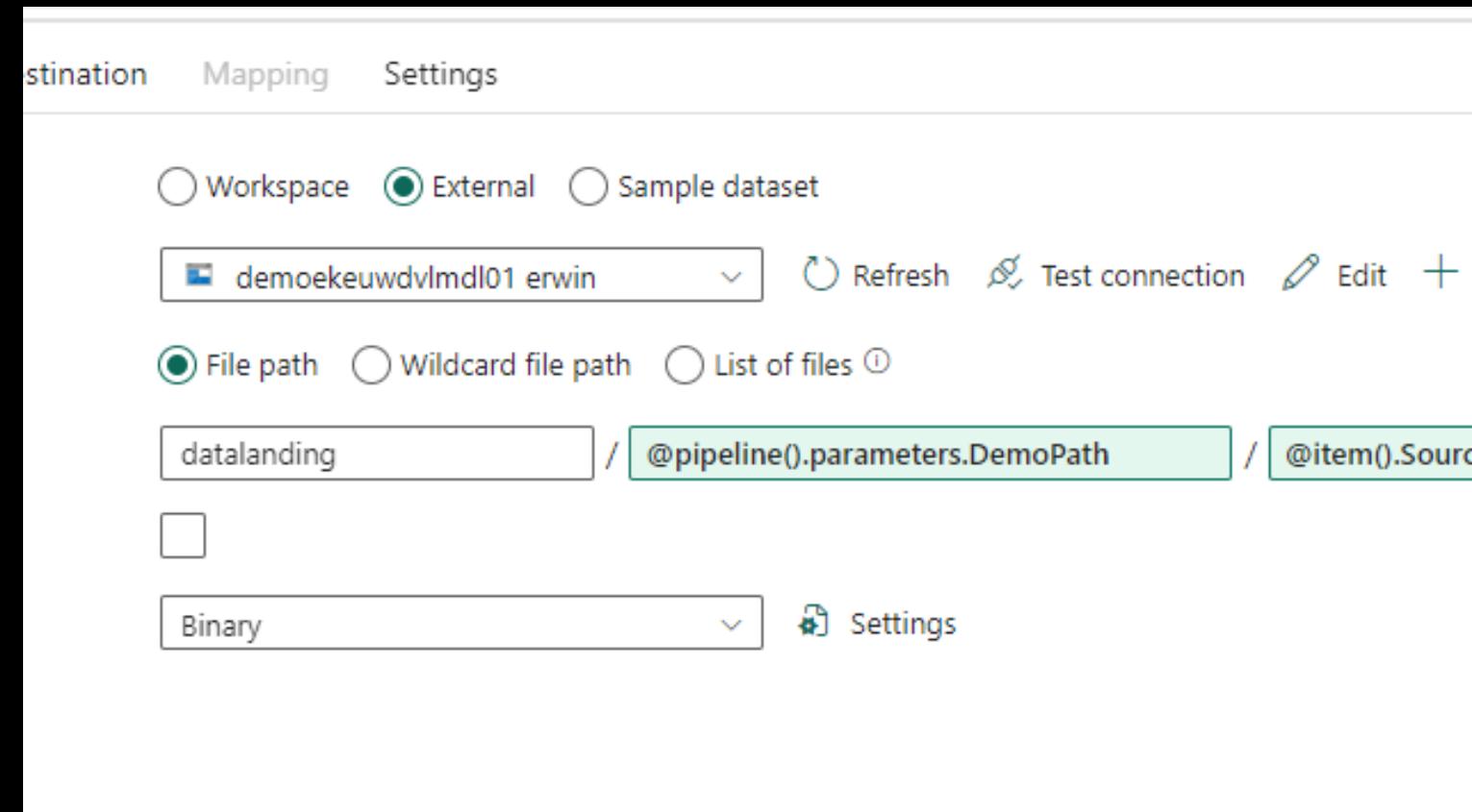
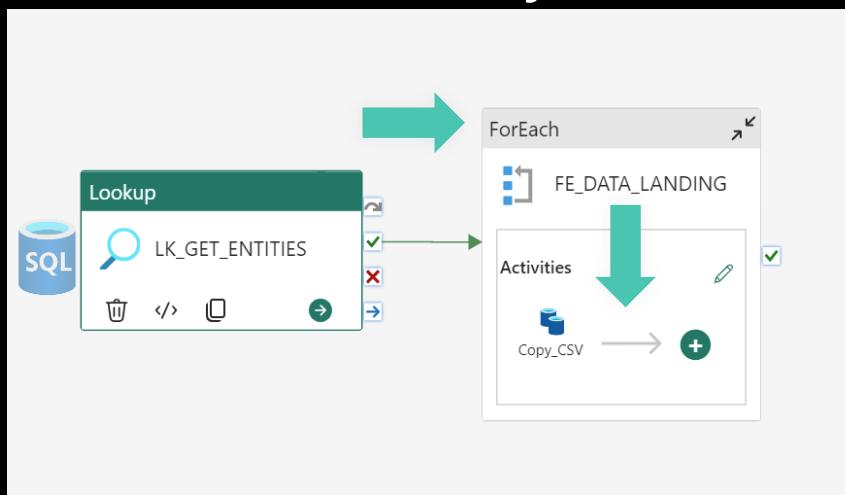


```
1 # Set arguments
2 PrimaryKeys = ""
3 IsIncremental = False
4
5 SourceWorkspace= ""
6 SourceLakehouse = ""
7 SourceLakehouseName = ""
8 source_file_path = ""
9 source_file_name = ""
10 source_file_type = ""
11
12 TargetWorkspace = ""
13 TargetLakehouse = ""
14 TargetLakehouseName = ""
15 target_schema = ""
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
```

PySpark (Python) ▾ Parameters

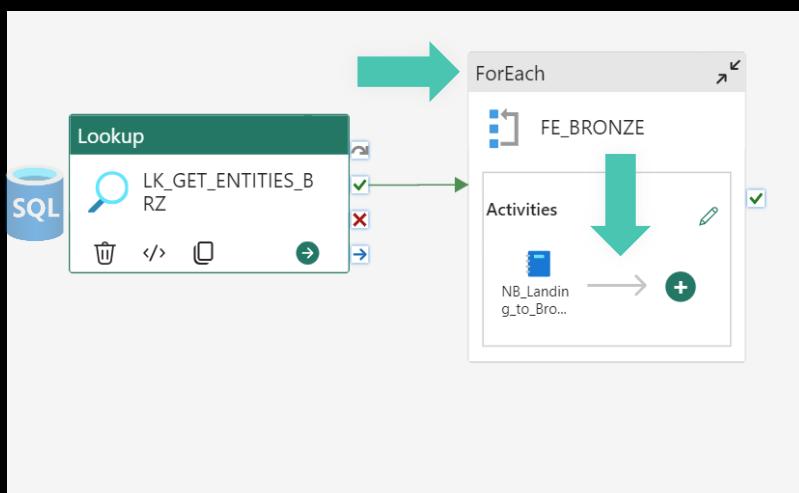
Copy Activity Parameters

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



Notebook Parameters

- Pass Parameters from Data Pipeline to Notebook
 - Set Base Parameters
 - Define Values



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

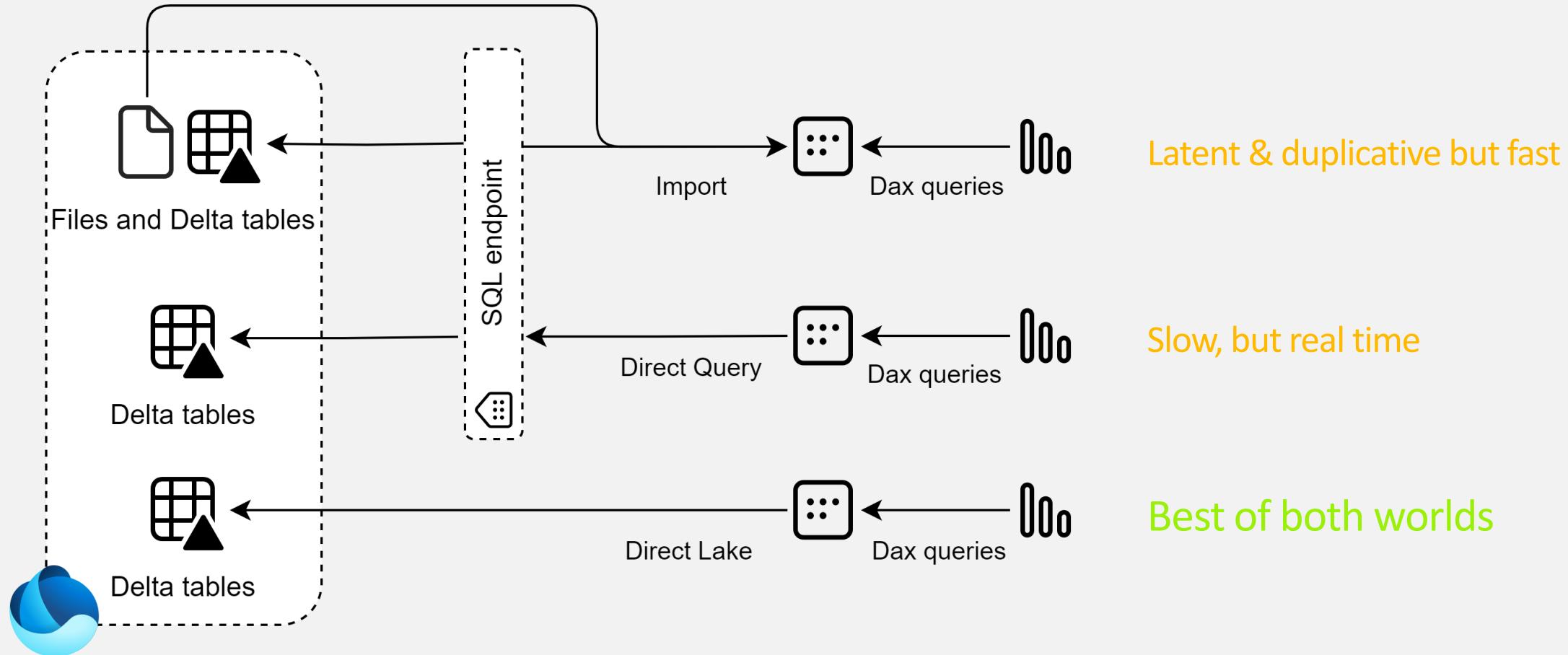
Demo





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*

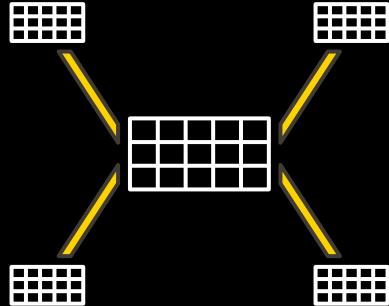


Data transformations should be done as far upstream as possible

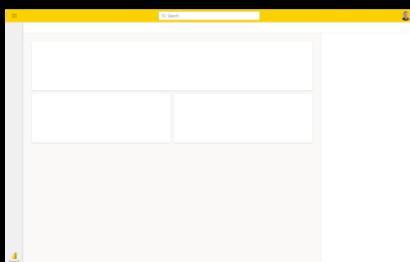


Data transformation directly in the Lakehouse unlocks “new” possibilities

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

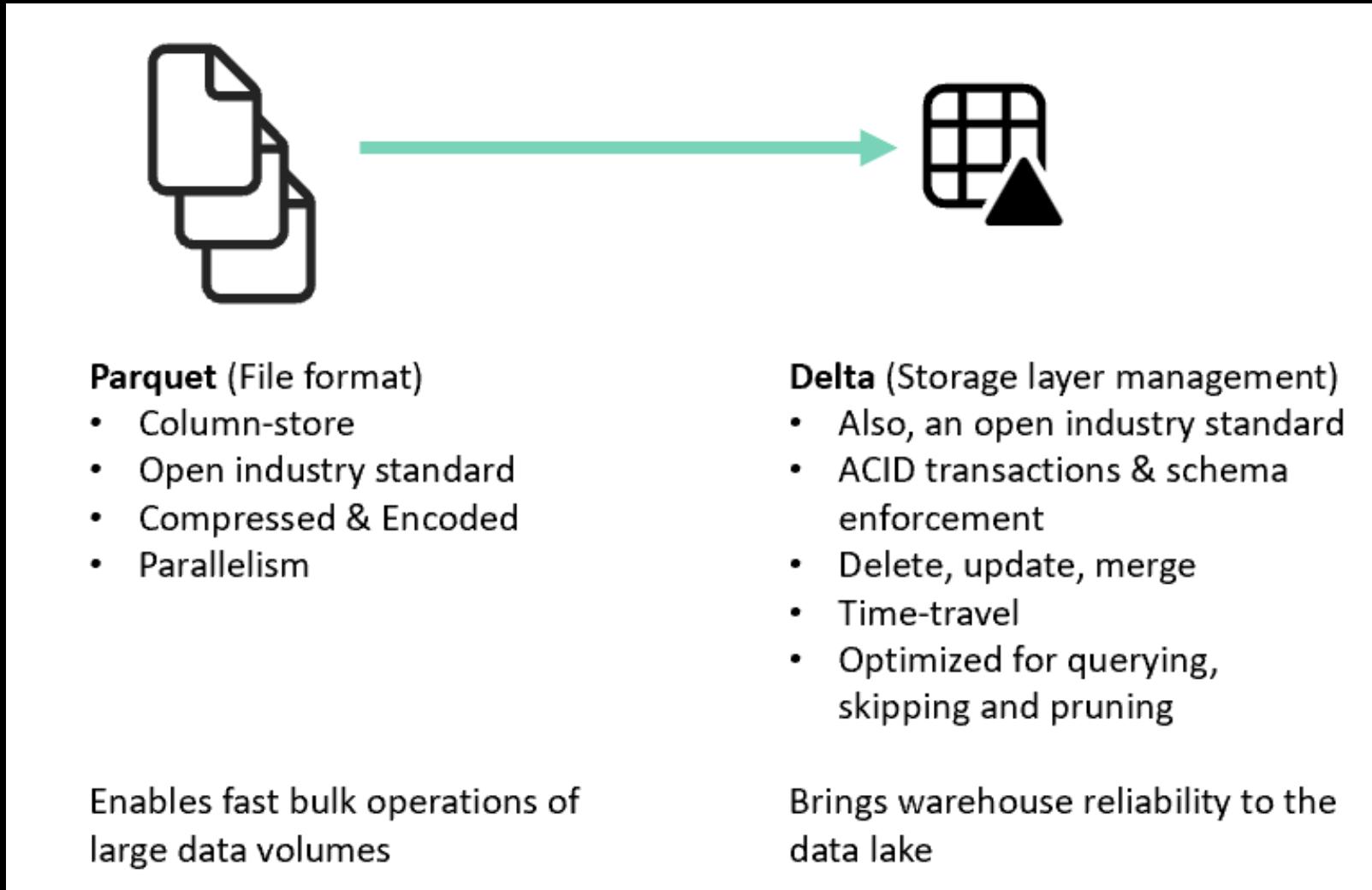
Demo





Direct Lake Internals & performance

Delta (Parquet)



From Z-order to V-order

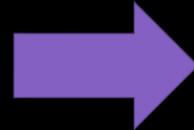
Yellow taxi (3 Billion rows)



416 GB



164 GB



V-order
60GB

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

A screenshot of a Windows File Explorer window. The address bar shows the path: C:\Program Files\Microsoft SQL Server\MSAS15.MSSQLSERVER\OLAP\Data\AdventureWorksTabular.0.db\DimCustomer (10).tbl\238.prt. The main pane displays a list of files related to the DimCustomer table, including various .idf and .idfm files. To the right of the file list, there is a detailed view of a single file named '238.prt'. This view is organized into sections: 'Finance' (highlighted in blue), 'Lake view' (selected), 'Table view', 'Tables' (expanded), and 'Revenue' (selected). The 'Tables' section lists several fact tables: ACR Adjustment Type, Adjustment Type, Budget, Business, Calendar, Consumed Revenue, Forecast, Forecast Type, Future Flag, and SCM. The 'Revenue' section shows a list of Parquet files: '_delta_log' (2 items, Folder), and five part files: part-00000-87858576-90b7-4aff-8c9e-69dcc52dbt (8.4 GB), part-00001-631fb085-0591-46b8-a0b5-0fec8f2255 (8.4 GB), part-00002-0469bb29-daaaf-4ecd-a3ee-bb90331a6 (8.4 GB), part-00003-27e6062b-4d55-4469-b285-7cb2a2f32 (8.4 GB), and part-00004-b12eea8e-f255-41fa-a943-a78def57ce (8.4 GB). A large teal arrow points from the left towards the 'Revenue' section.

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.idfmata	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.idfmata	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.idfmata	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.idfmata	1/29/2020 6:36 PM		

Finance

Lake view Table view

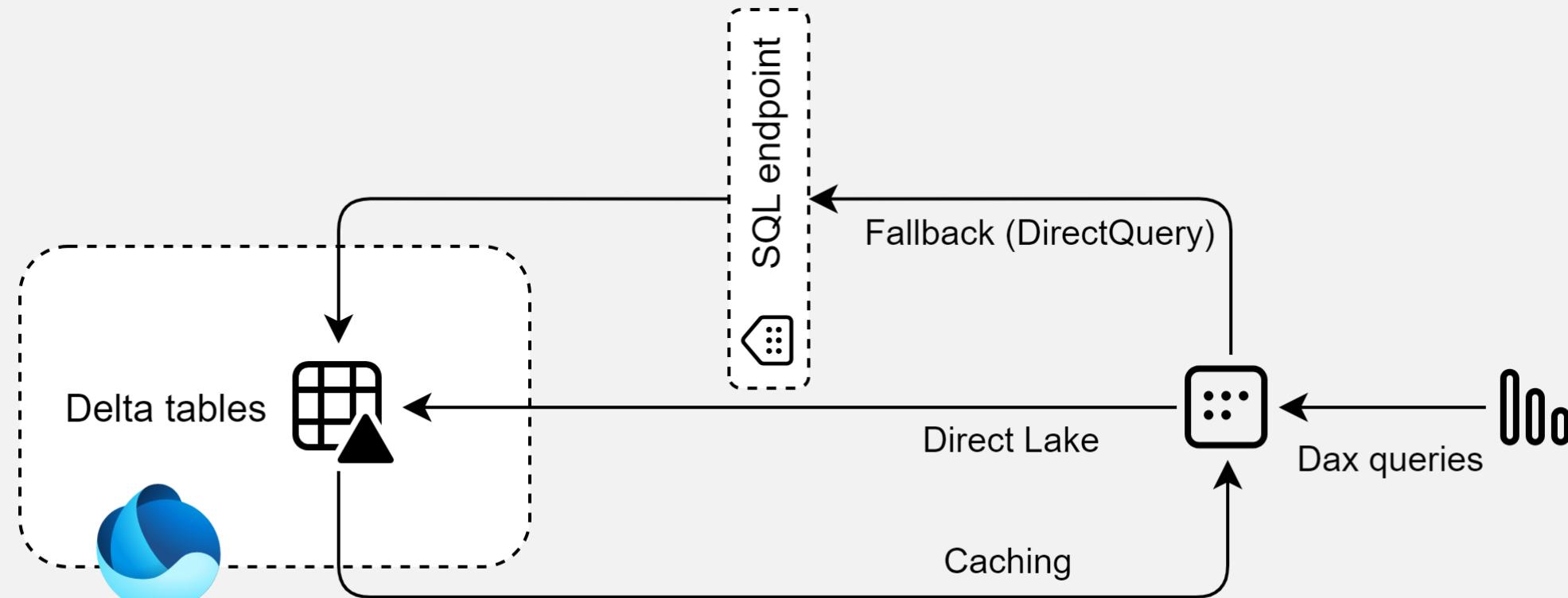
Tables

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

Tables > Revenue

Name ↑	Size	Type
_delta_log	2 items	Folder
part-00000-87858576-90b7-4aff-8c9e-69dcc52dbt	8.4 GB	PARQUET
part-00001-631fb085-0591-46b8-a0b5-0fec8f2255	8.4 GB	PARQUET
part-00002-0469bb29-daaaf-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

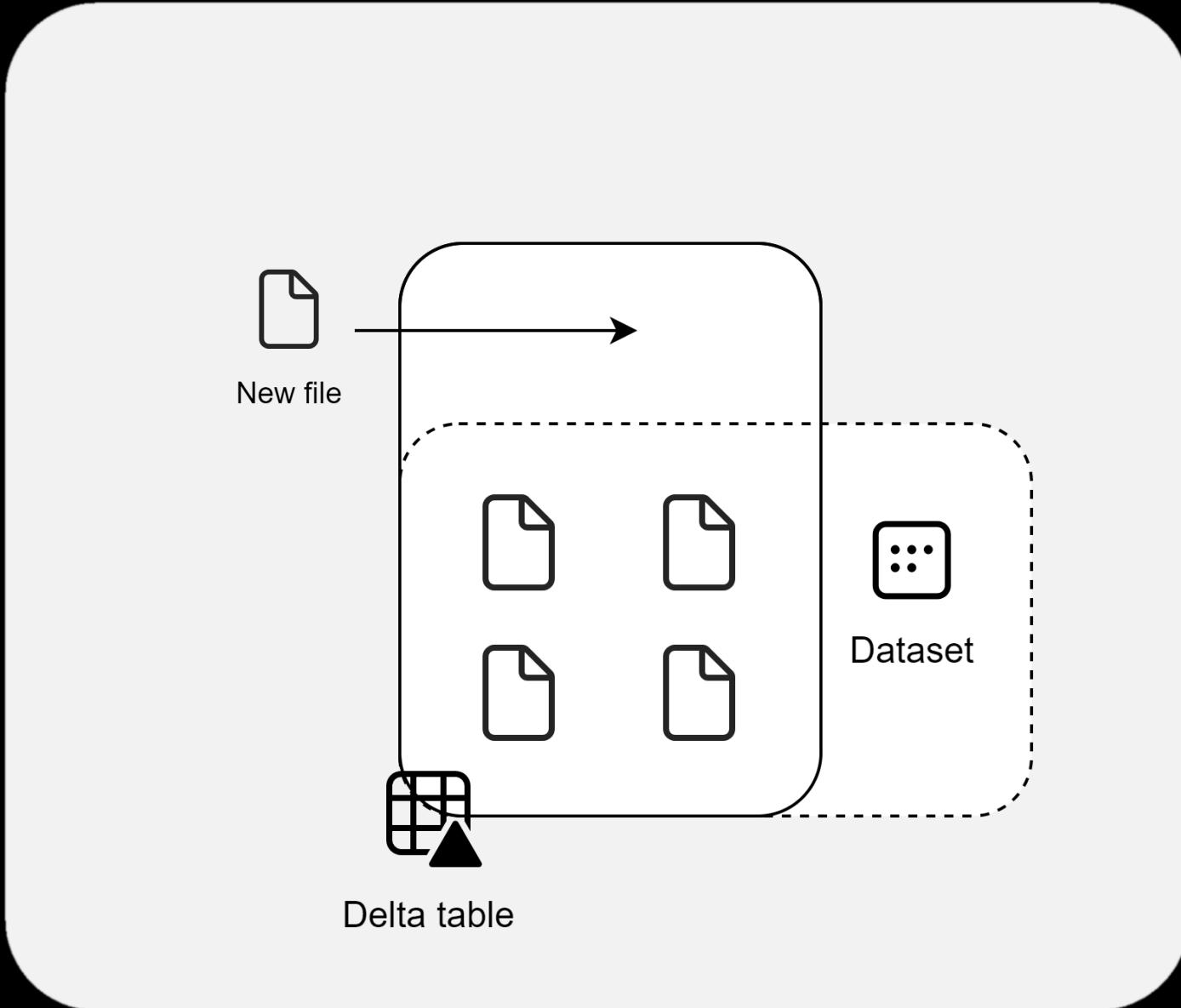
Item level on lakehouse

Introducing Framing

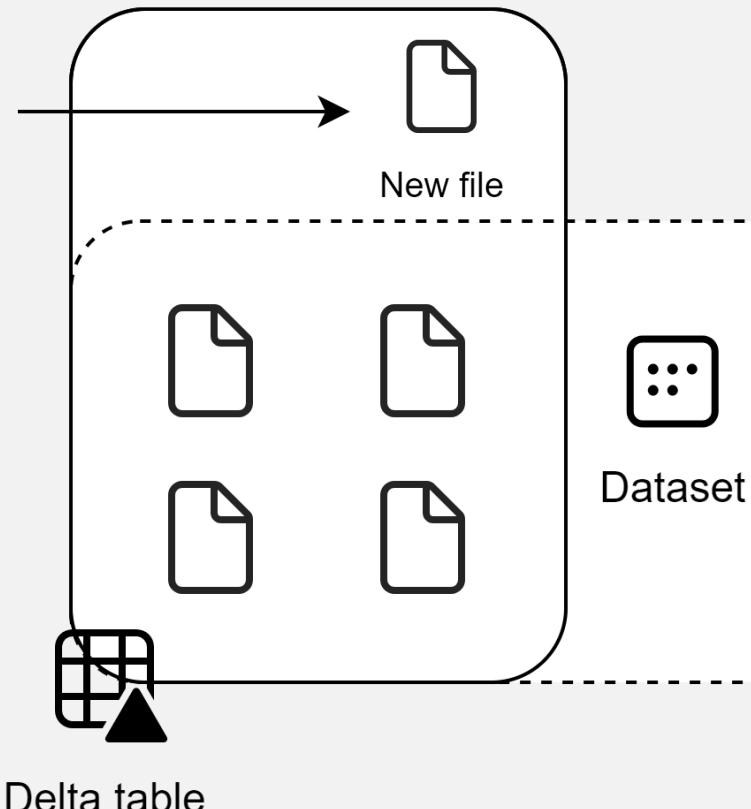
- Metadata **refresh** which does not actually load the data, but only the delta table definitions.



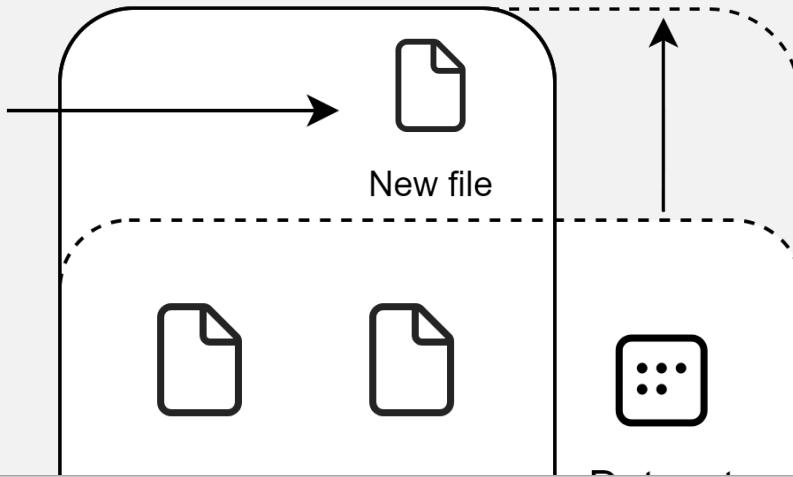
Framing



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

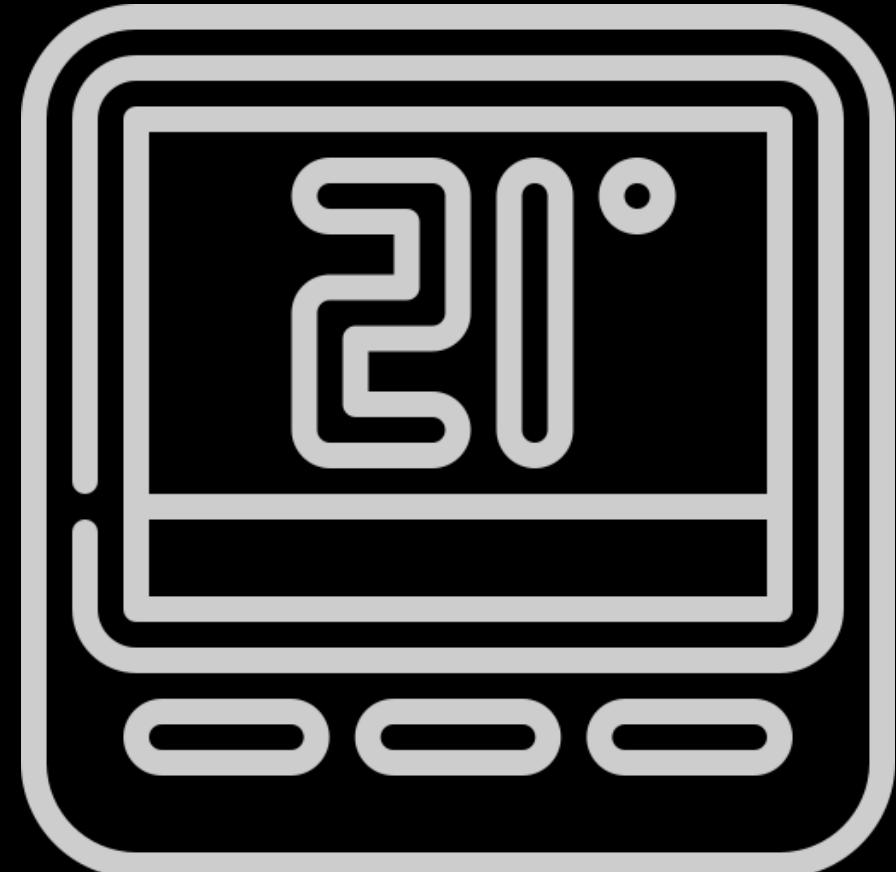
Demo



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

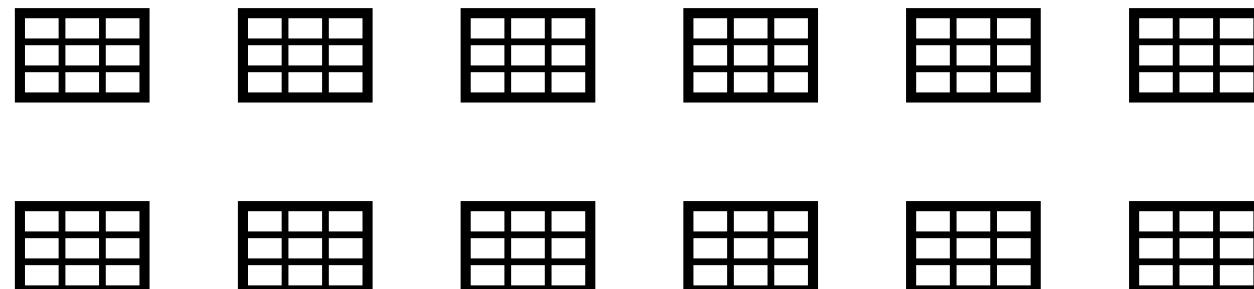
Capacity utilization:

Cool

Active memory



Storage



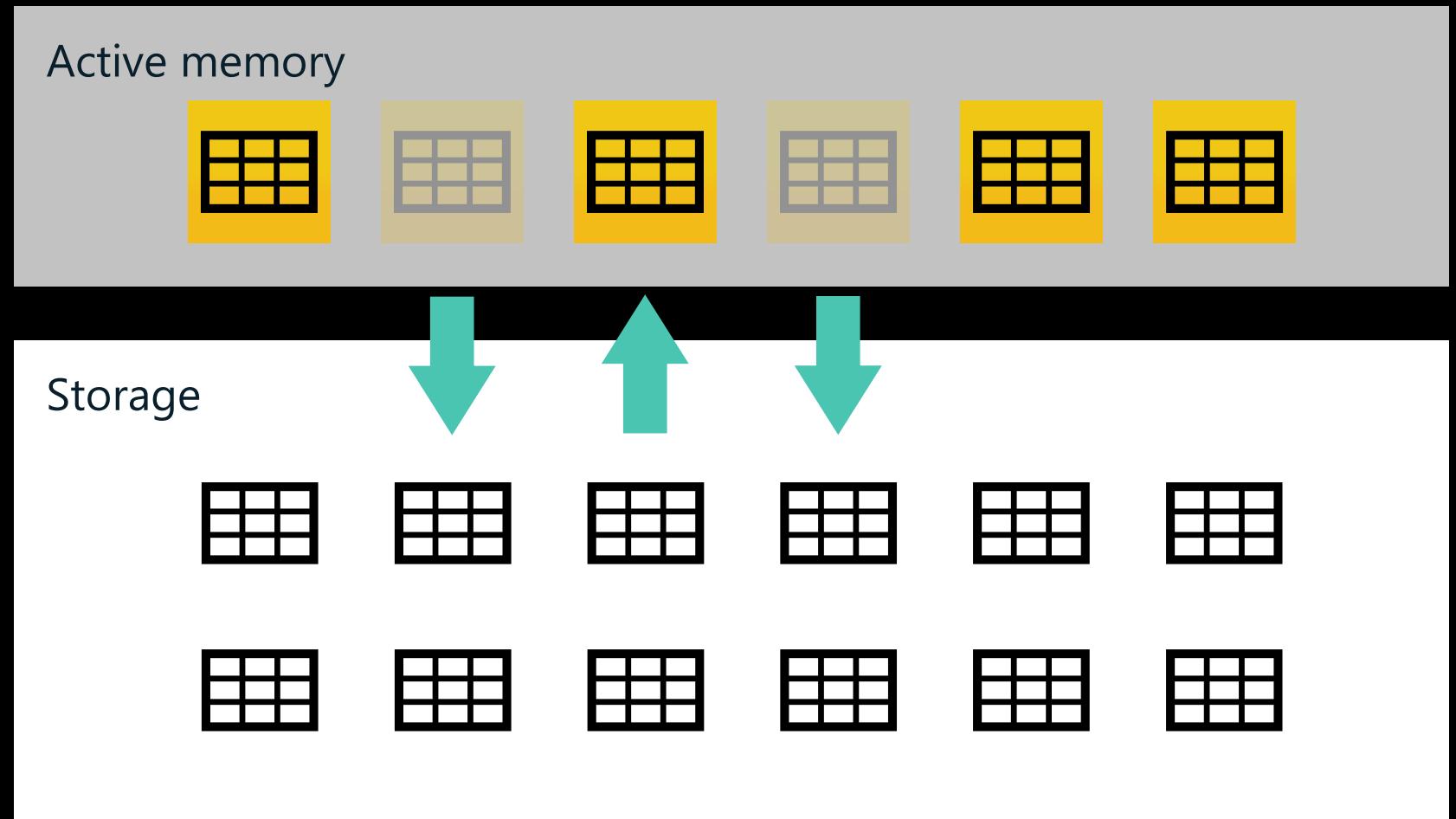
Eviction

Capacity: F64 / P1

Memory: 25 GB

Capacity utilization:

Warm



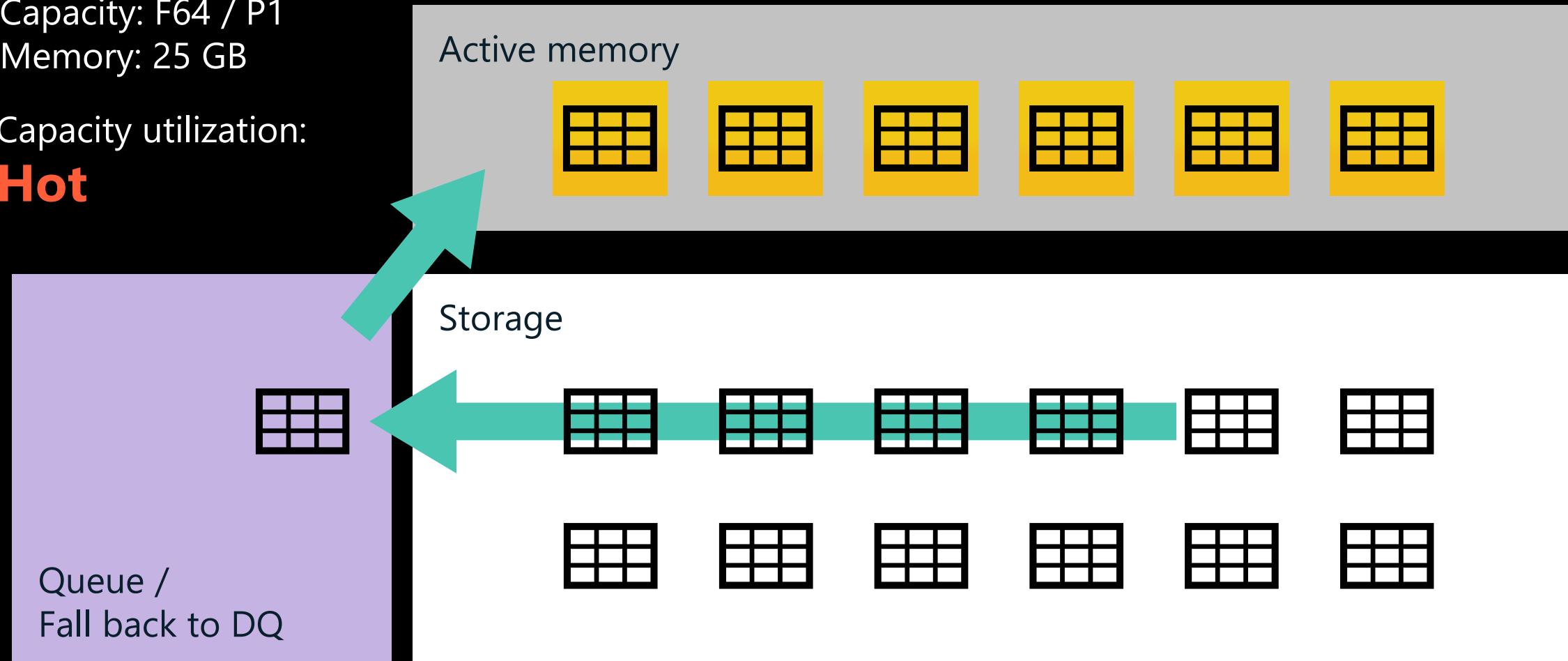
Eviction – queue / fall-back

Capacity: F64 / P1

Memory: 25 GB

Capacity utilization:

Hot



Temperature management

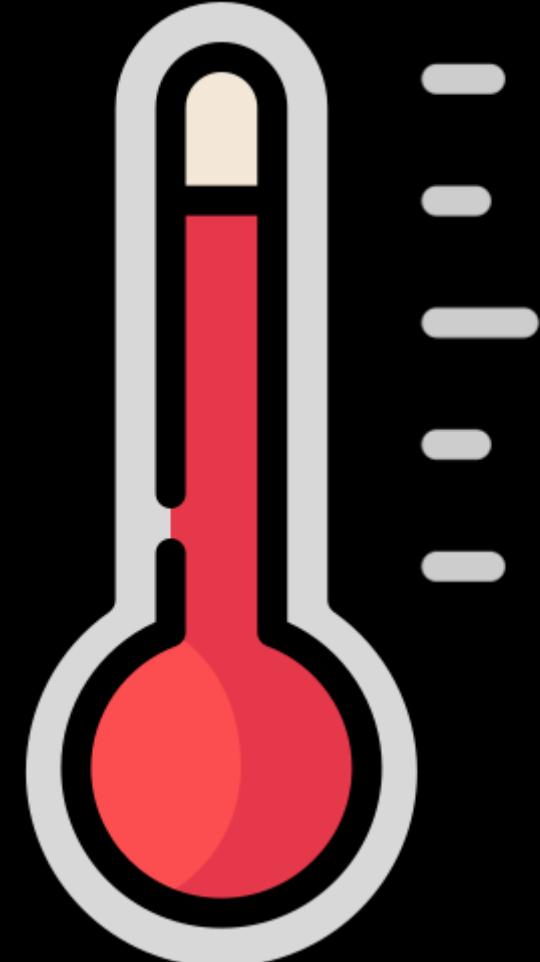
What will be evicted?

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

How can you influence that?

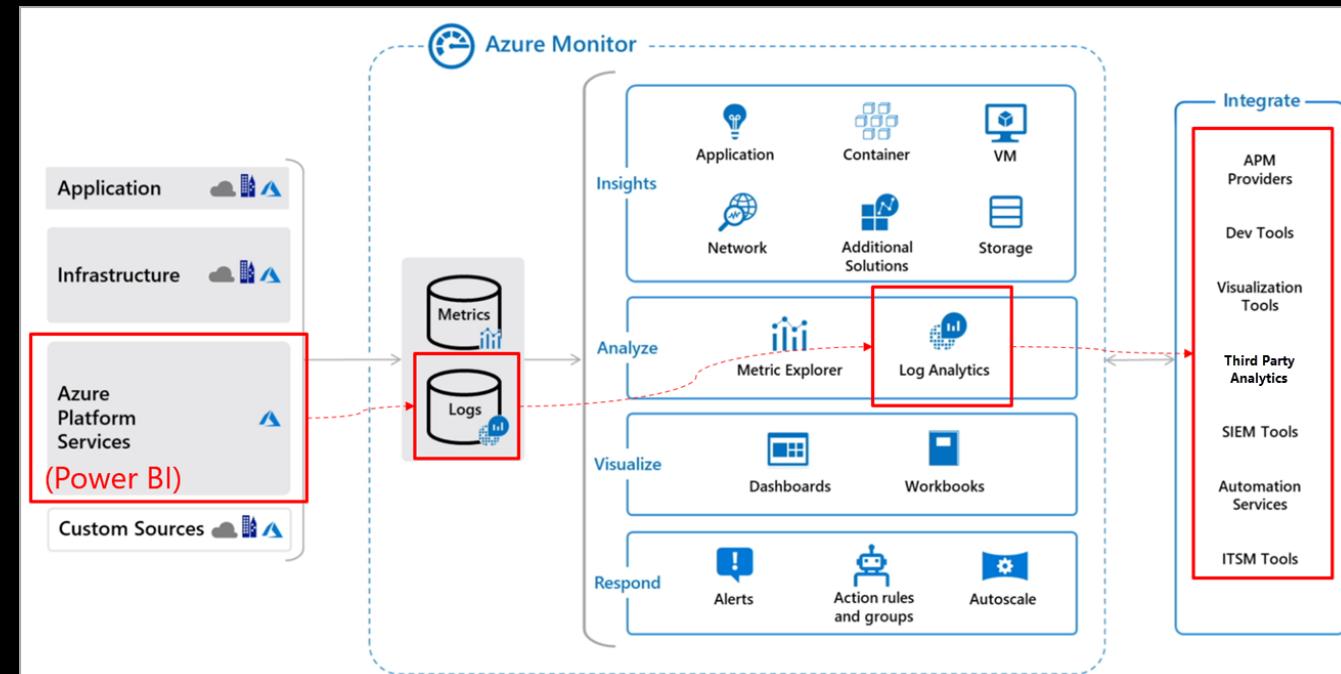
Consider setting up a process (notebook, other automated setup) to pro-actively execute queries to keep certain data **WARM**!

.



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

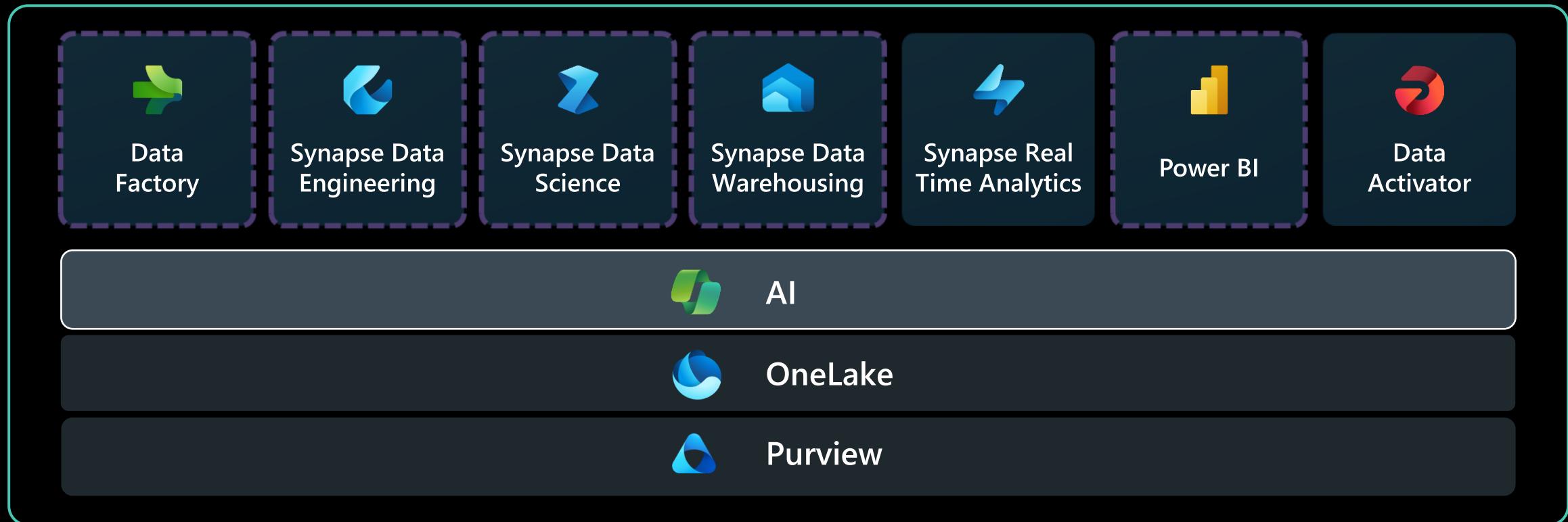
Autopilot refers to allowing AI to make all the decisions, while copilot involves AI assisting humans in making informed





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 Jupyter Notebook interface titled "Sales analysis | Data updated 1/12/23". The left sidebar displays a "Lakehouse explorer" with a tree view of datasets like Customer360, Sales, and Transaction. The main area has two code cells. The first cell contains AI-generated Python code for reading data from a PBIX file and merging it with sales data:

```
1 # Welcome to your new notebook
2 # Type here in the cell editor to add code!
3
4 # AI-generated code
5
6 import pandas as pd
7 customer_data = pbi.read_table('CustomerProfitabilitySamplePBIX', 'Customer')
8 sales_data = pbi.read_measure('CustomerProfitabilitySamplePBIX', 'Total Revenue', [('Customer', 'Name')])
9 customer_sales = customer_data.merge(sales_data, on='Name')
10 print(customer_sales.head())
11
12 # Command executed in 7 sec 427 ms by Sonia
```

The second cell shows the resulting DataFrame:

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

The right side of the interface includes a "Copilot Preview" panel with a sidebar for "Transform your data with Copilot" and a main area for generating code based on user input.

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 sidebar on the left includes "Lakehouse explorer", "Customer360", "Tables", "Customer", "CustomerFeedback...", "Inventory", "Product", "Sales", "Transaction", "Unidentified", "Files", "Sales", and "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_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 is a table titled "Country/Region Ab" with columns: Name Ab, City Ab, Industry ID #, and #. The table lists various US states and their corresponding industry types and IDs. To the right of the table is a preview pane showing the generated Python code.

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

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

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

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

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

AI-generated content can have mistakes. Make sure it's appropriate before using it. Read preview terms

Generative AI on your data

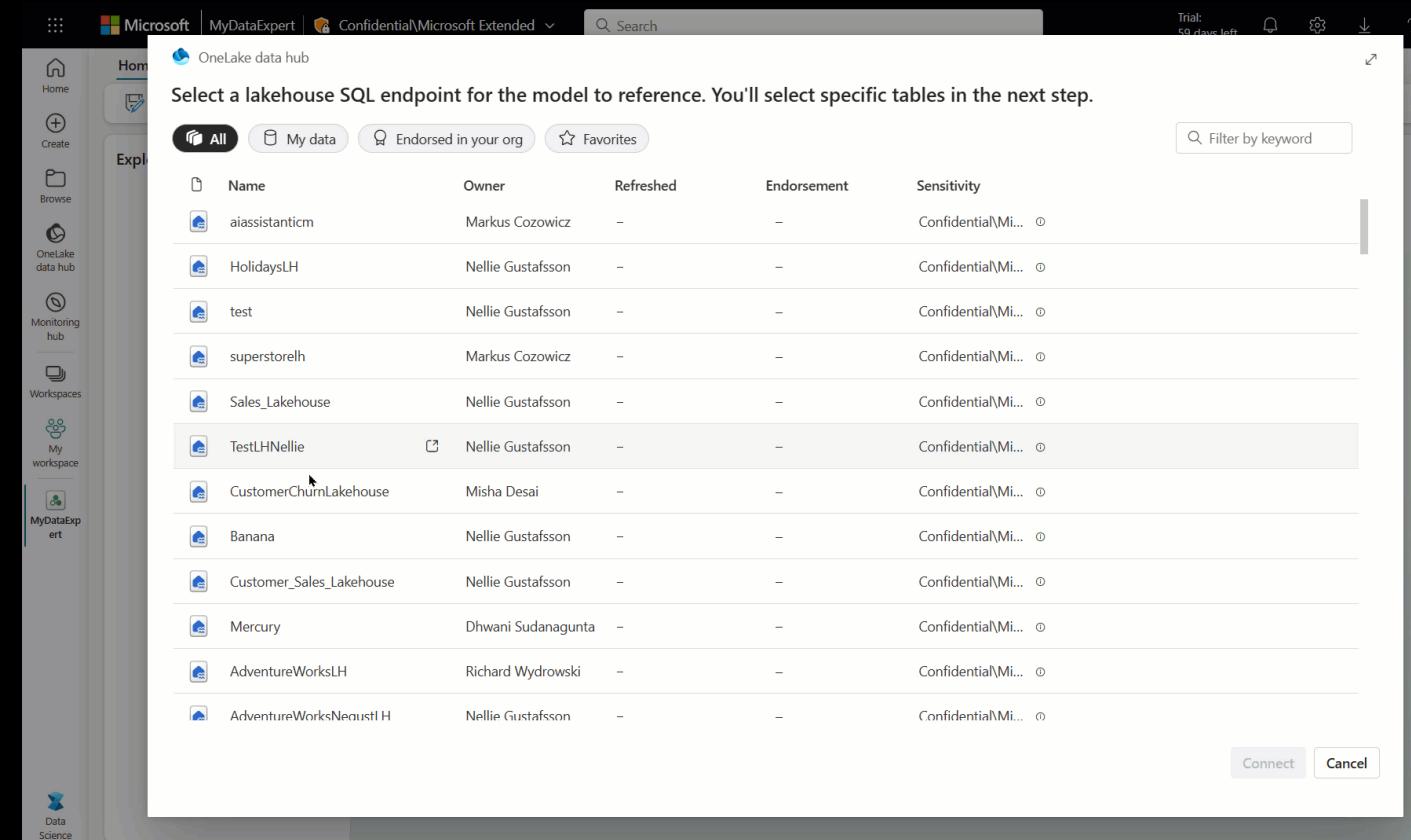
(Public Preview)

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



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. Below the navigation bar, the main title is "OneLake data hub". A sub-header reads: "Select a lakehouse SQL endpoint for the model to reference. You'll select specific tables in the next step." There are four filter buttons: "All" (selected), "My data", "Endorsed in your org", and "Favorites". A search bar with the placeholder "Filter by keyword" is also present. The main area is a table listing 14 lakehouse endpoints:

Name	Owner	Refreshed	Endorsement	Sensitivity
aiaassistantcm	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 (Coming soon)

The unified data platform for the era of AI



Unified
architecture

Unified
experience

Unified
governance

Unified
business model

Demo



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



Erwin de Kreuk



Marc Lelijveld

