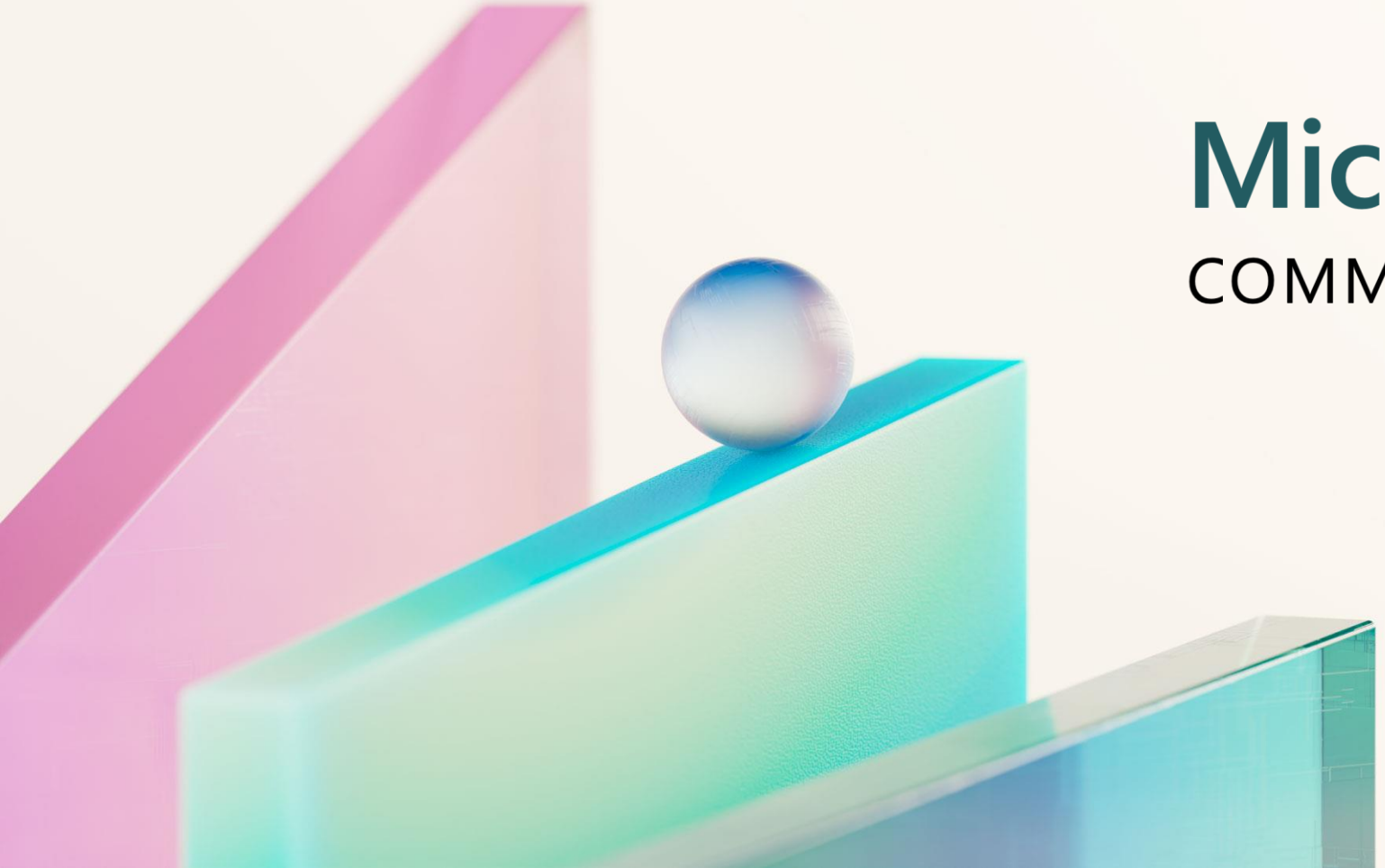
Microsoft

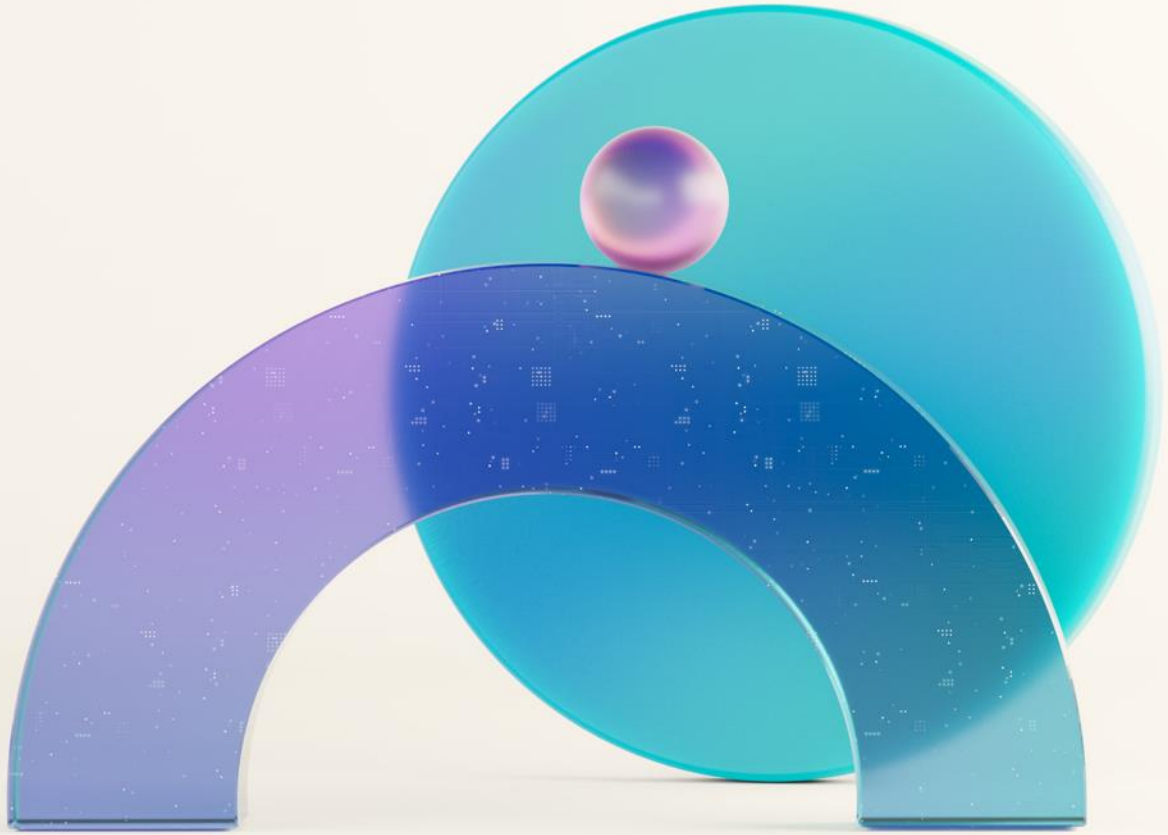


Microsoft Fabric

COMMUNITY CONFERENCE

Microsoft

Microsoft Fabric
COMMUNITY CONFERENCE



Drive Efficiency and Reliability Using Metadata-Driven Frameworks

March 26-28, 2024

Erwin de Kreuk

Erwin de Kreuk

- Principal Consultant
- Lead Data & Analytics InSpark

Let's connect

Microsoft® MVP Most Valuable Professional

@erwindekreuk

linkedin.com/in/erwindekreuk

erwindekreuk.com

github.com/edkreuk

https://sessionize.com/erwin-de-kreuk/

Objectives

- Data platform challenges
- Why
- Medallion Architecture
- Parameters
- Framework
- Logging
- Recap

Data platform Challenges

'From data source to data model to report

- Data Catalog
- Data Lineage
- Data Monitoring
- Incremental Loading
- Version Control
- Historical Data
- Data Orchestration
- Data Modeling
- Data Privacy
- Data Governance
- Data Enrichment
- Data Security
- Data Quality
- Large number of different data sources
- Medallion Architecture

What would help

Simplicity in connecting data sources

Fast result in hours

Focus on business value instead of data integration

Meta Data Driven

- Standardized data pipelines, Notebooks, orchestration and Way of Work

Overview of data process flows

Integration of solution like:

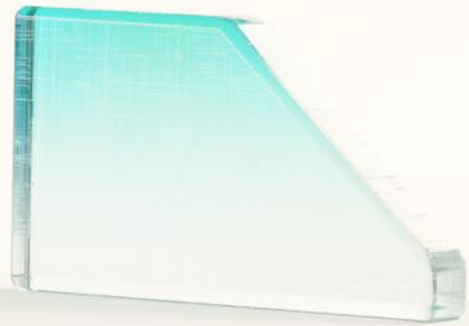
- Data Privacy
- Data Quality
- Data Governance

From data “Spaghetti to Lasagna”

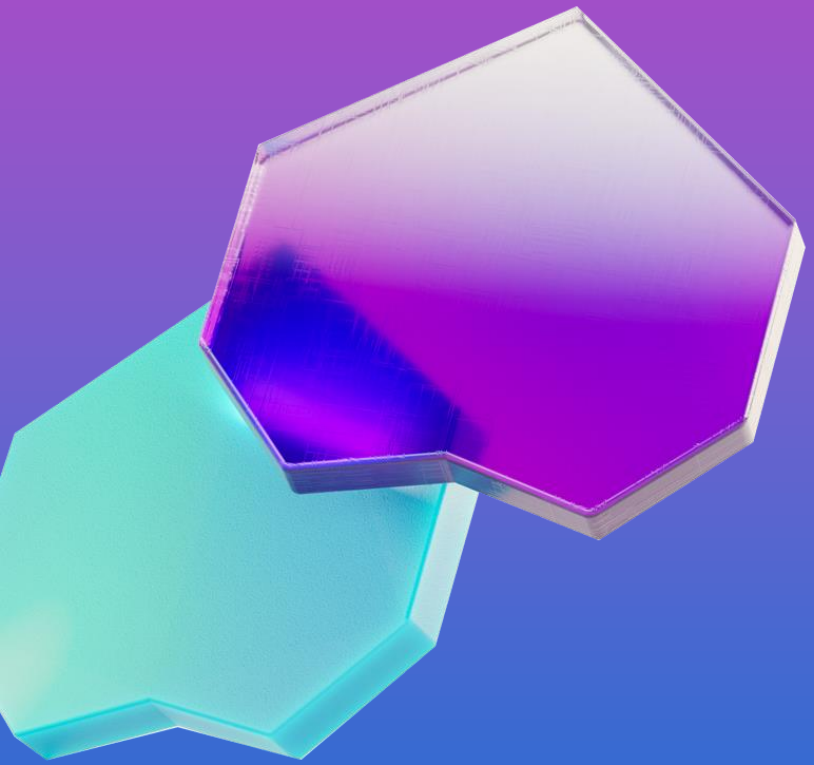
- Uniform data architecture

I am the Chief Data Information Officer
and don’t want to be the Chief Integration Officer

Help me to simplify and automate my Data Orchestration

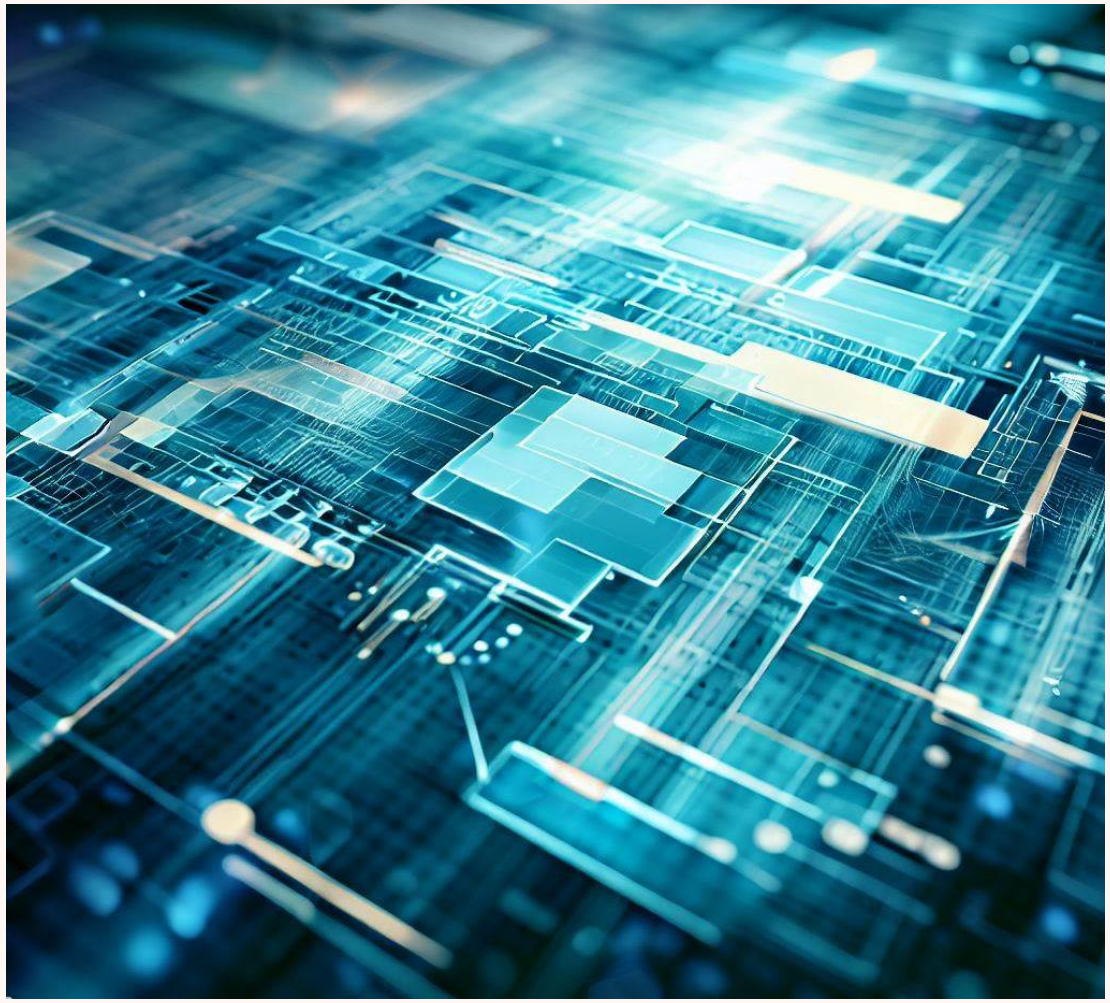


WHY



- Automation
- Flexibility
- Scalability
- Traceability

Out-of-the-Box Framework

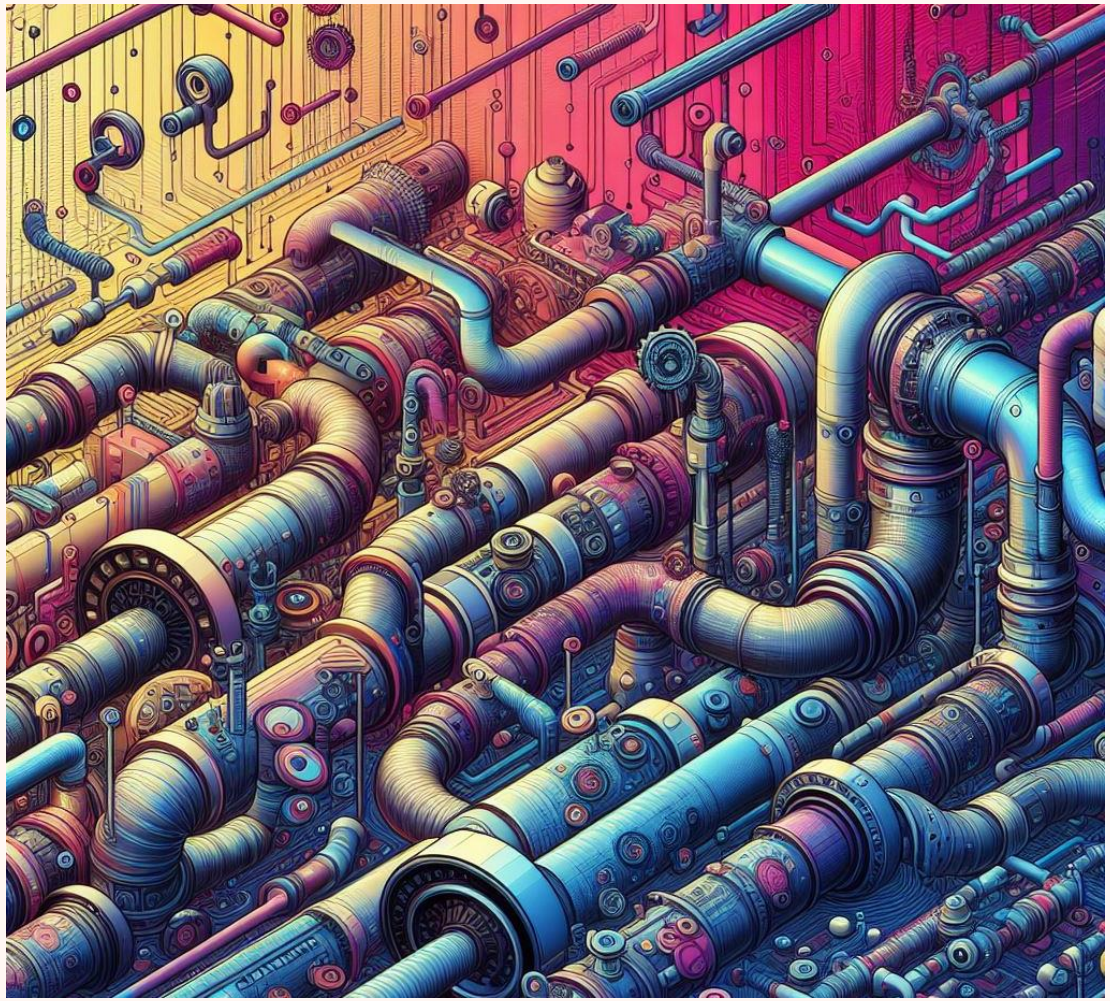


- Ready-to-use.
- Rapid implementation.
- Limited customization.
- Lower development effort.
- Lower upfront costs.
- Ongoing support and updates.

Custom-Made Framework

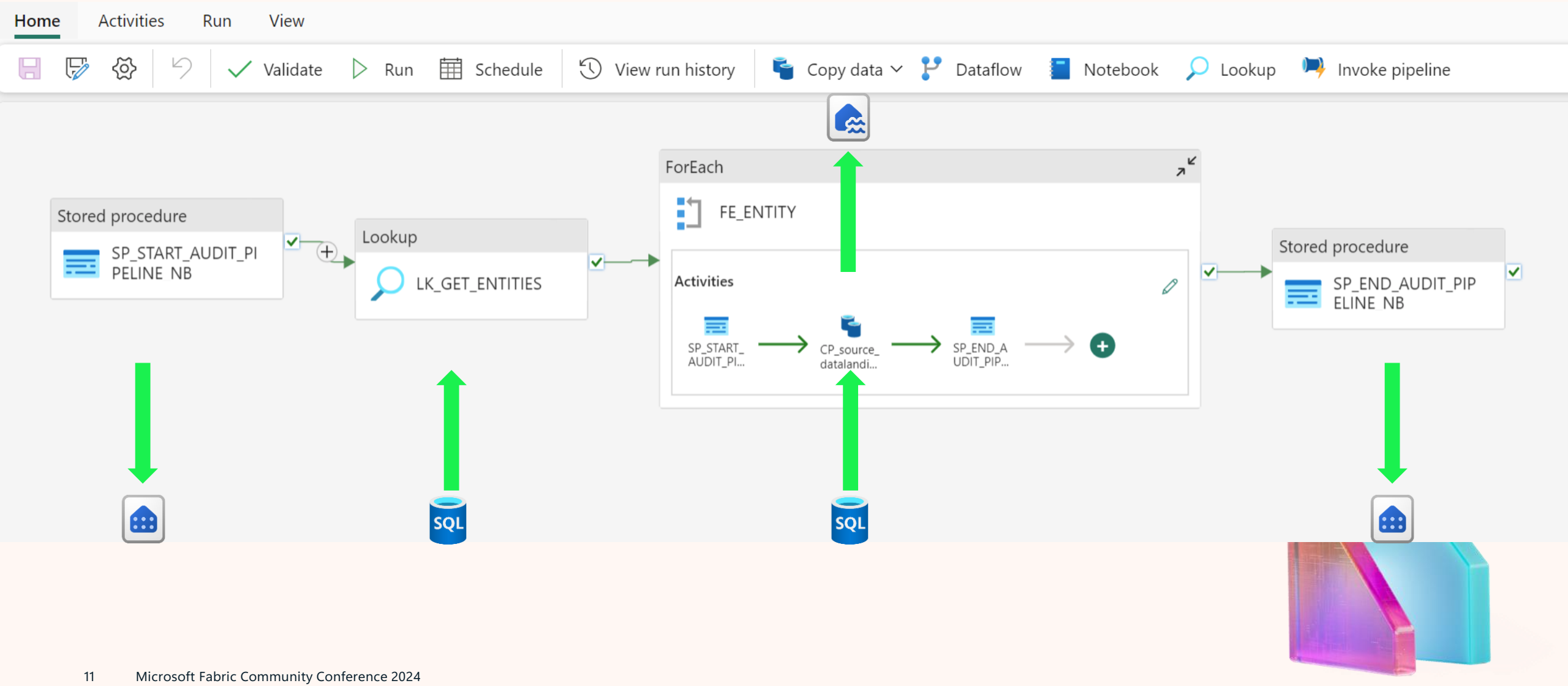
- Tailored to specific needs.
- Full control over design and features.
- Higher development effort.
- Flexibility and extensibility.
- Higher upfront costs.

Custom-Made Framework



- 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

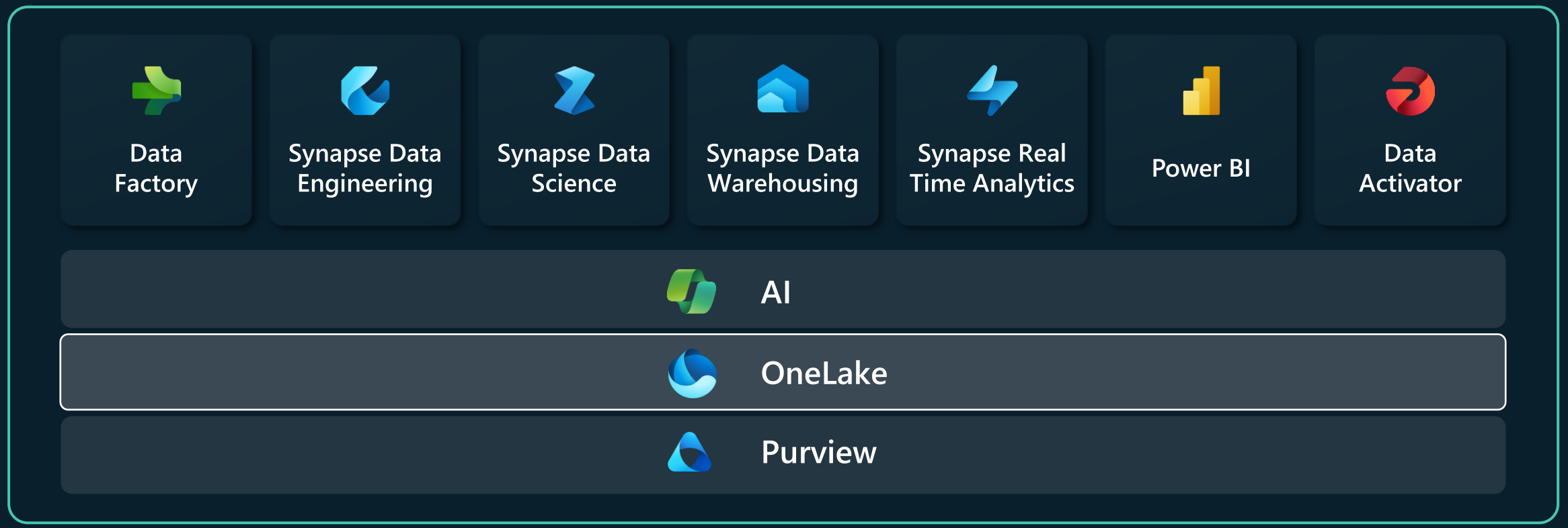
FINAL GOAL for today's session



Who is already using Parameters?

Microsoft Fabric

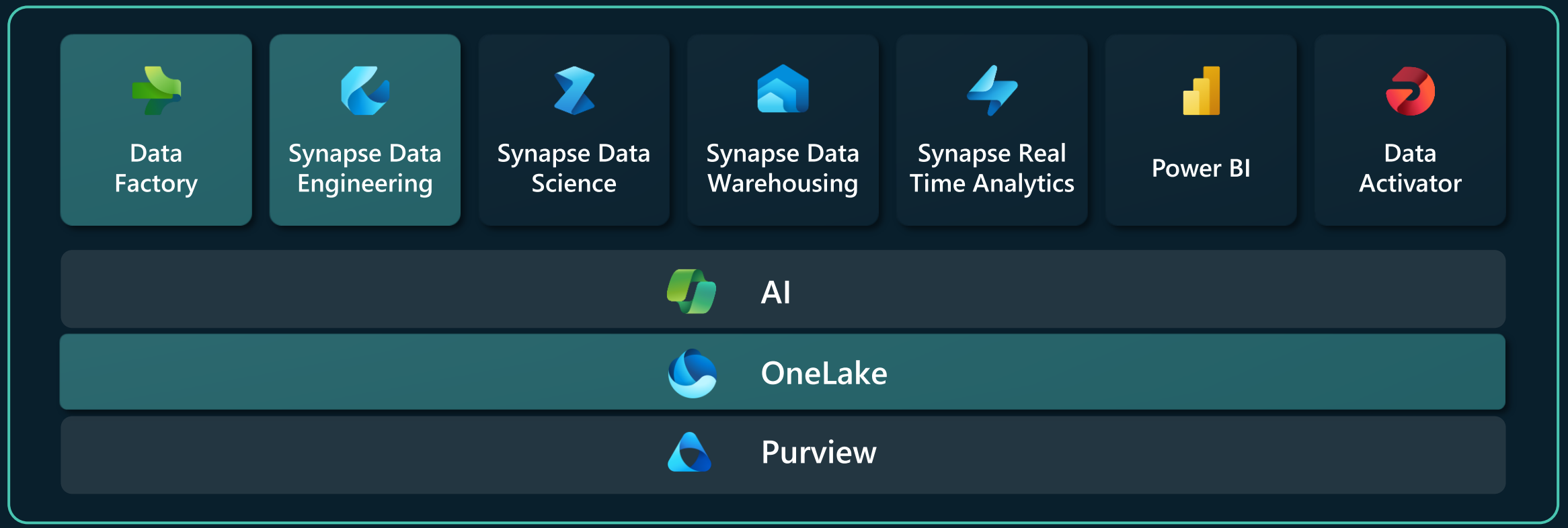
The unified data platform for the era of AI



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

Microsoft Fabric

The unified data platform for the era of AI

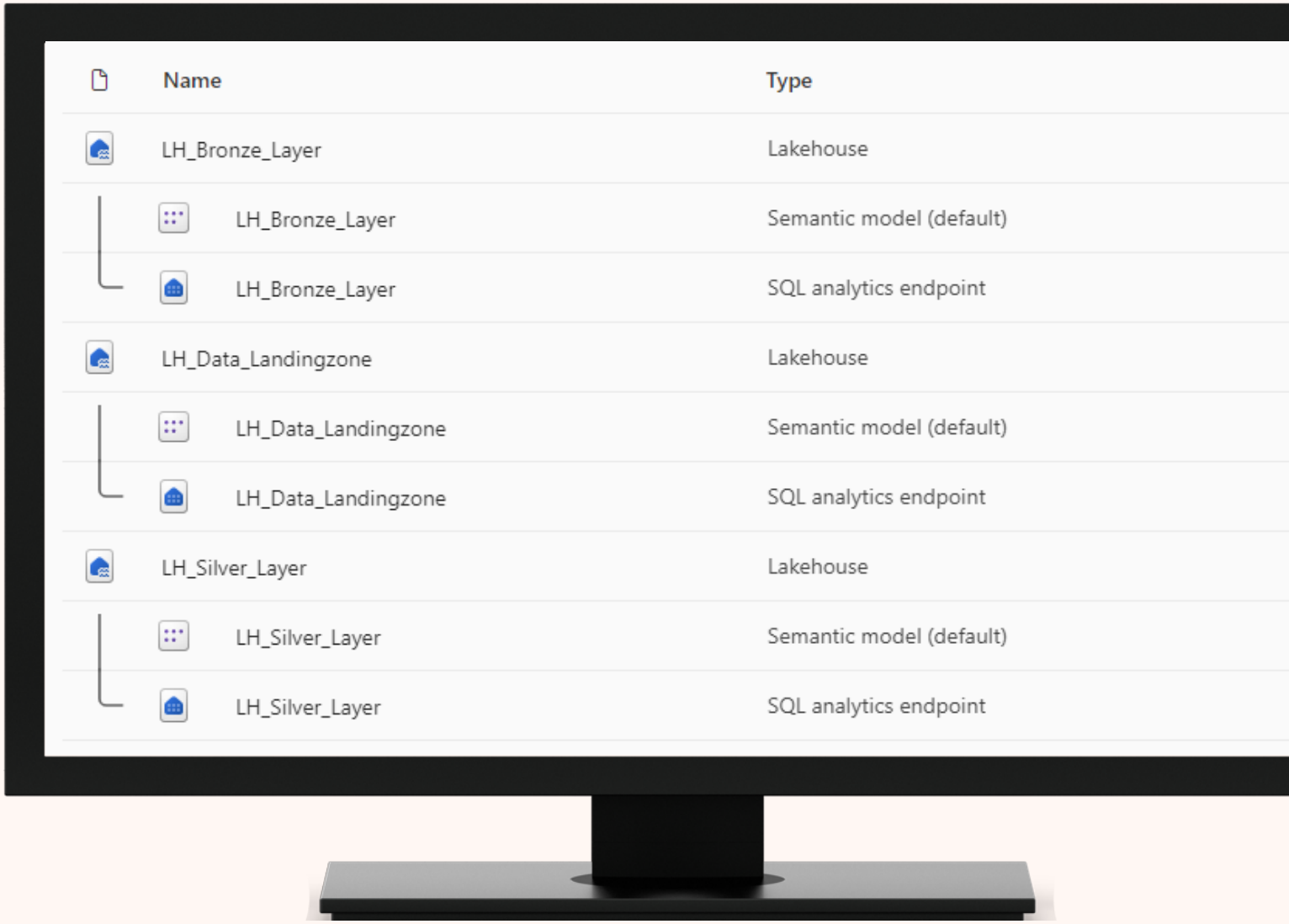


Unified architecture | Unified experience | Unified governance | Unified business model

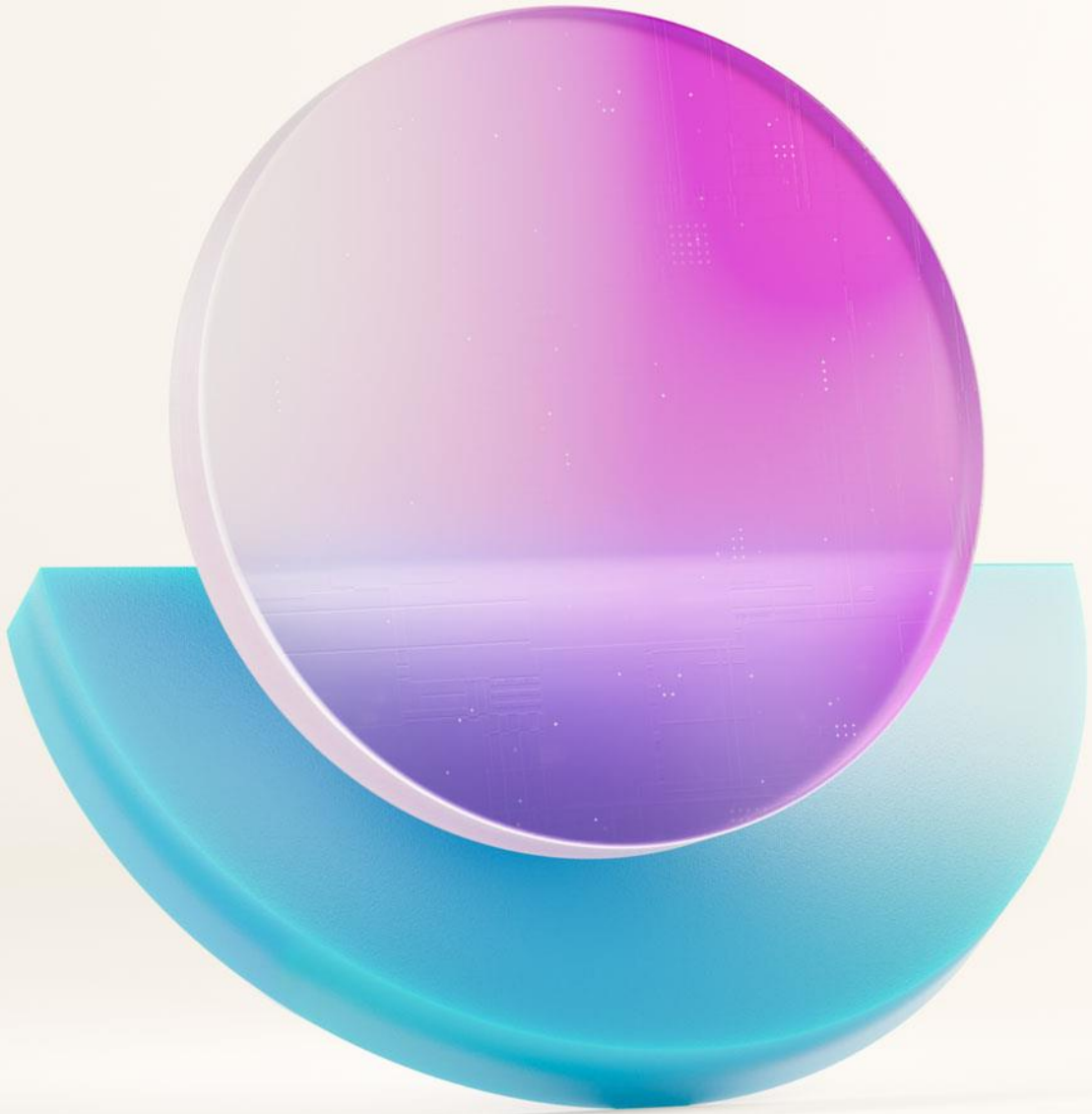
Lakehouse

- The foundation of Microsoft Fabric is a **Lakehouse**, which is built on top of the **OneLake** scalable storage layer and uses **Apache Spark and SQL** compute engines for big data processing.

- **Lakehouses use Spark and SQL engines** to process large-scale data and support machine learning or predictive modeling analytics.
- **Lakehouse data is organized in a *schema-on-read format***, which means you define the schema as needed rather than having a predefined schema.
- **Lakehouses support ACID** (Atomicity, Consistency, Isolation, Durability) transactions through Delta Lake formatted tables for data consistency and integrity.
- **Lakehouses are a single location** for data engineers, data scientists, and data analysts to access and use data.



Microsoft Fabric
COMMUNITY CONFERENCE



Medallion architecture

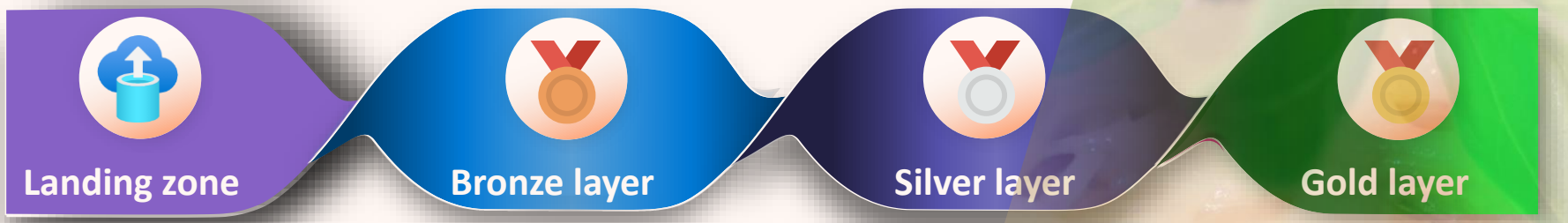
Who is using a Medallion architecture?

‘Uniform data architecture’
From data “Spaghetti to Lasagna”

Medallion Architecture

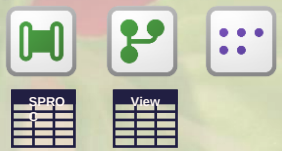
'Data processing in different stages'

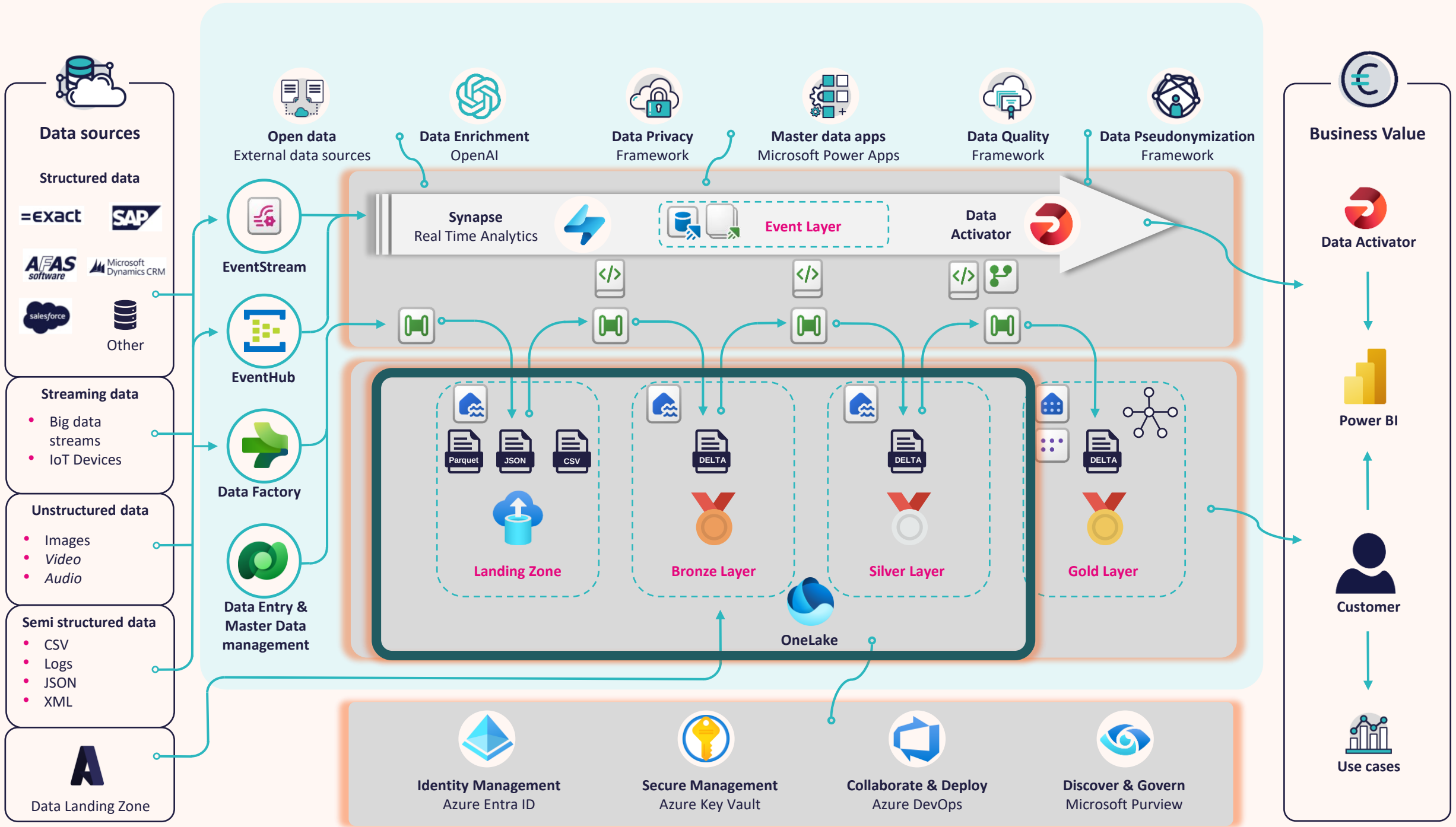
Stages



Medallion Architecture

'Data processing in different stages'

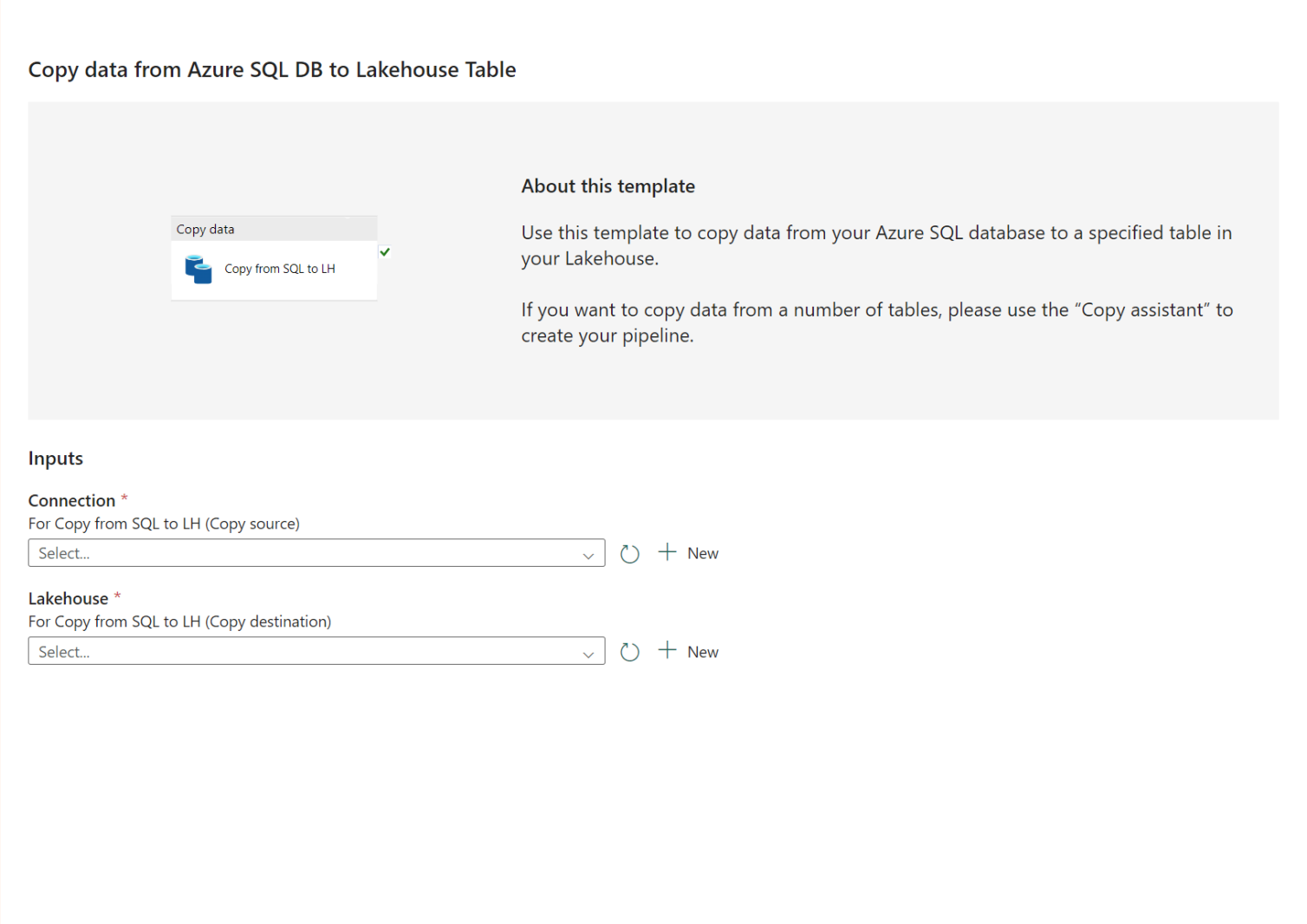
Stage:	Definition:	Filetype:	Files/Tables:	Fabric:
Gold layer	• Dimensions & Facts (Star Schema) • Historical Analysis • Business rules • Documentation • Aggregated data • Logical table names			
Silver layer	• Historical Data (Type 1 or 2) • Data quality rules • Data Cleansing • Validated data • No business 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			


Data sources
Structured data
Other
Streaming data
Big data streams
IoT Devices
Unstructured data
Images
Video
Audio
Semi structured data
CSV
Logs
JSON
XML
Data Landing Zone
Open data
External data sources
Data Enrichment
OpenAI
Data Privacy
Framework
Master data apps
Microsoft Power Apps
Data Quality
Framework
Data Pseudonymization
Framework
EventStream
EventHub
Data Factory
Data Entry & Master Data management
Synapse
Real Time Analytics
Event Layer
Data Activator
Parquet
JSON
CSV
DELTA
DELTA
DELTA
Landing Zone
Bronze Layer
Silver Layer
Gold Layer
OneLake
Identity Management
Azure Entra ID
Secure Management
Azure Key Vault
Collaborate & Deploy
Azure DevOps
Discover & Govern
Microsoft Purview
Business Value
Data Activator
Power BI
Customer
Use cases

Understand Parameters

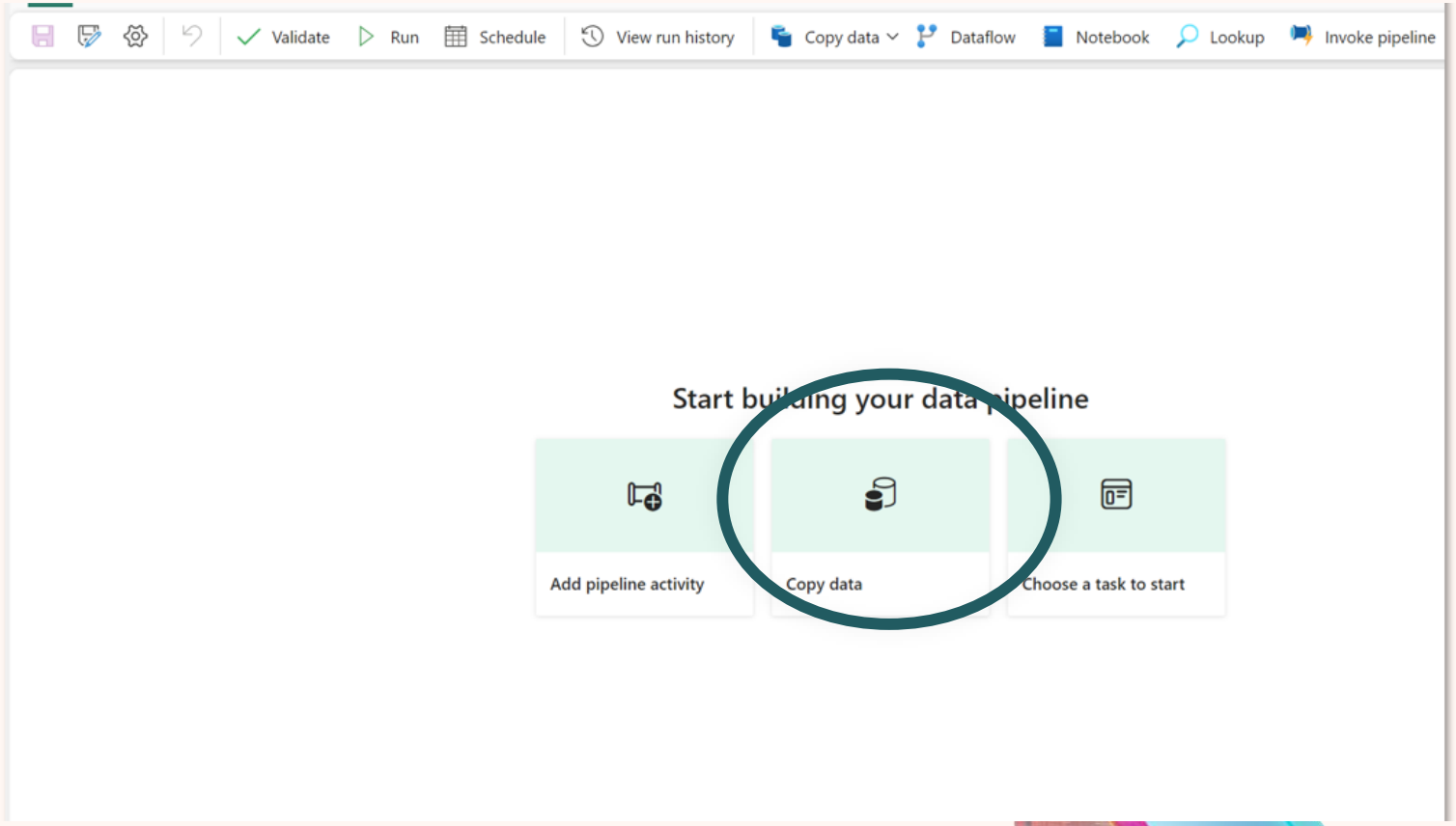
Templates

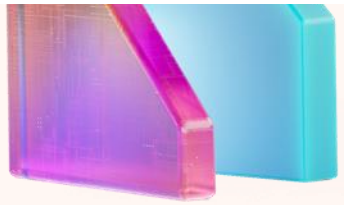
- Templates are pre-defined pipelines that allow you to get started quickly with Data Factory.
- These templates help to reduce development time by providing an easy way to create pipelines.
- Templates are available for common data integration scenarios.
- Templates can be customized to meet specific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



Understand Parameters

Copy Assistant

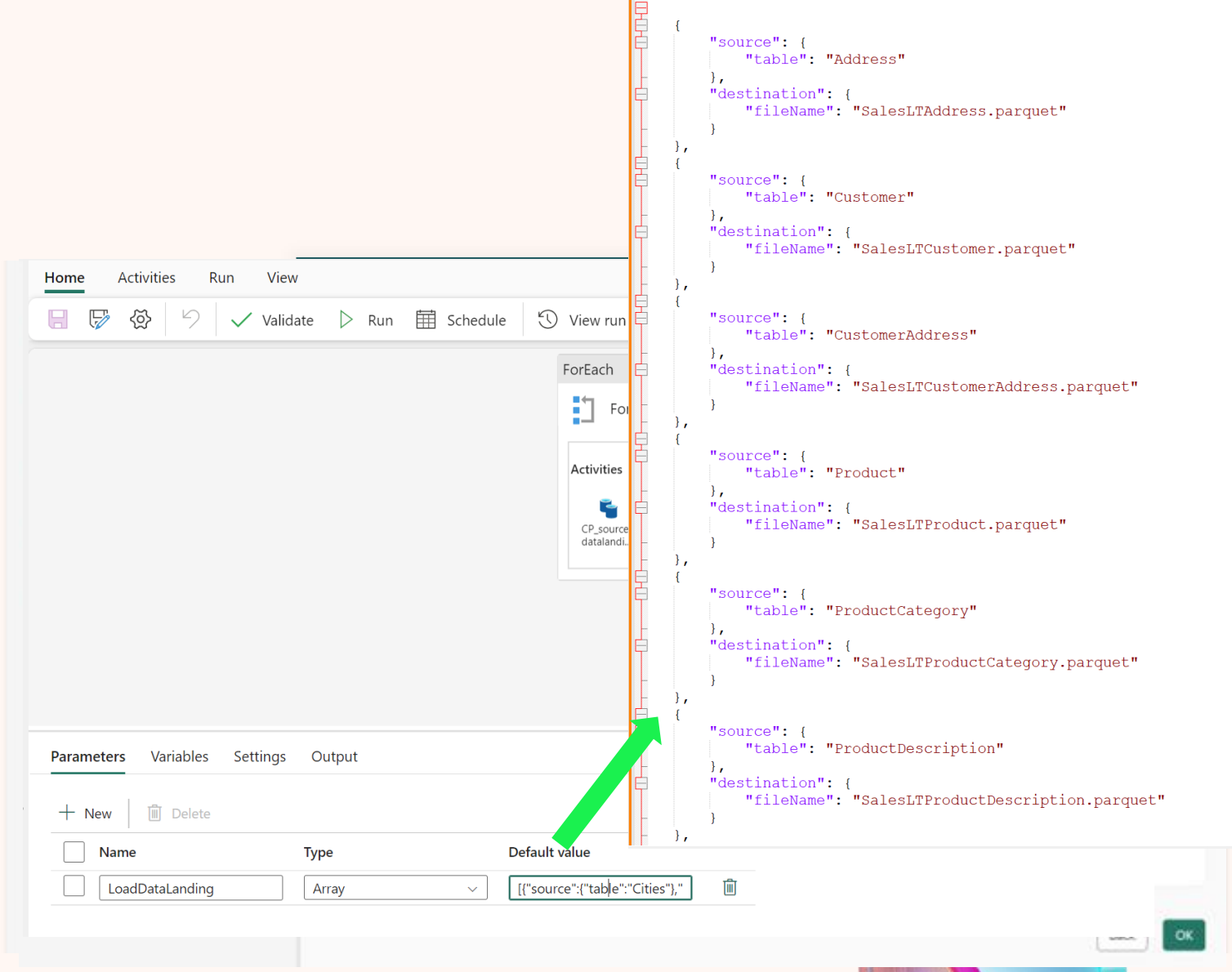




Understand Parameters

Copy Assistant

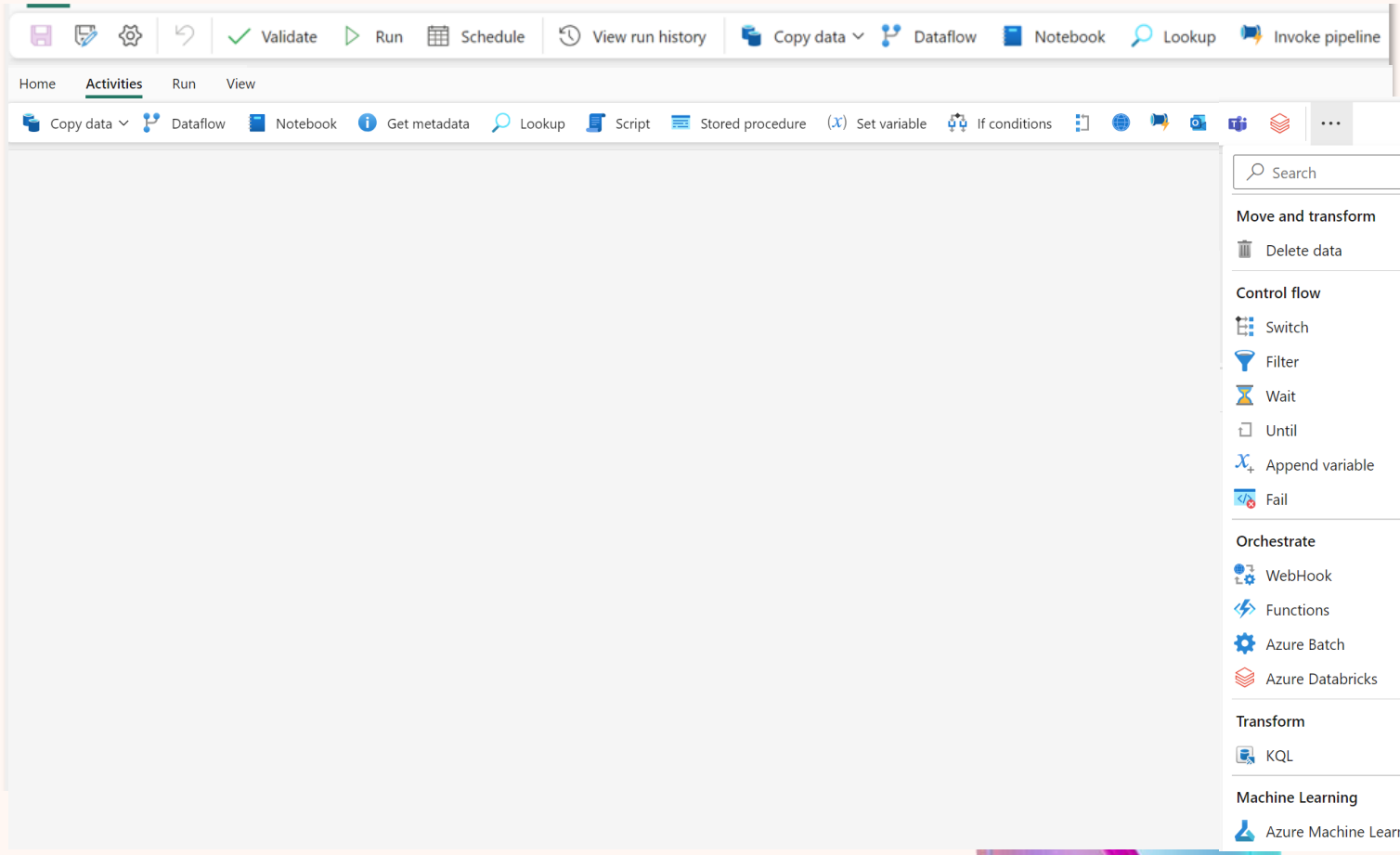
- Select Data Source
- Select Tables
- Select Destination
- Select Lakehouse
- Select Filetypes
- Data is stored as Delta Parquet



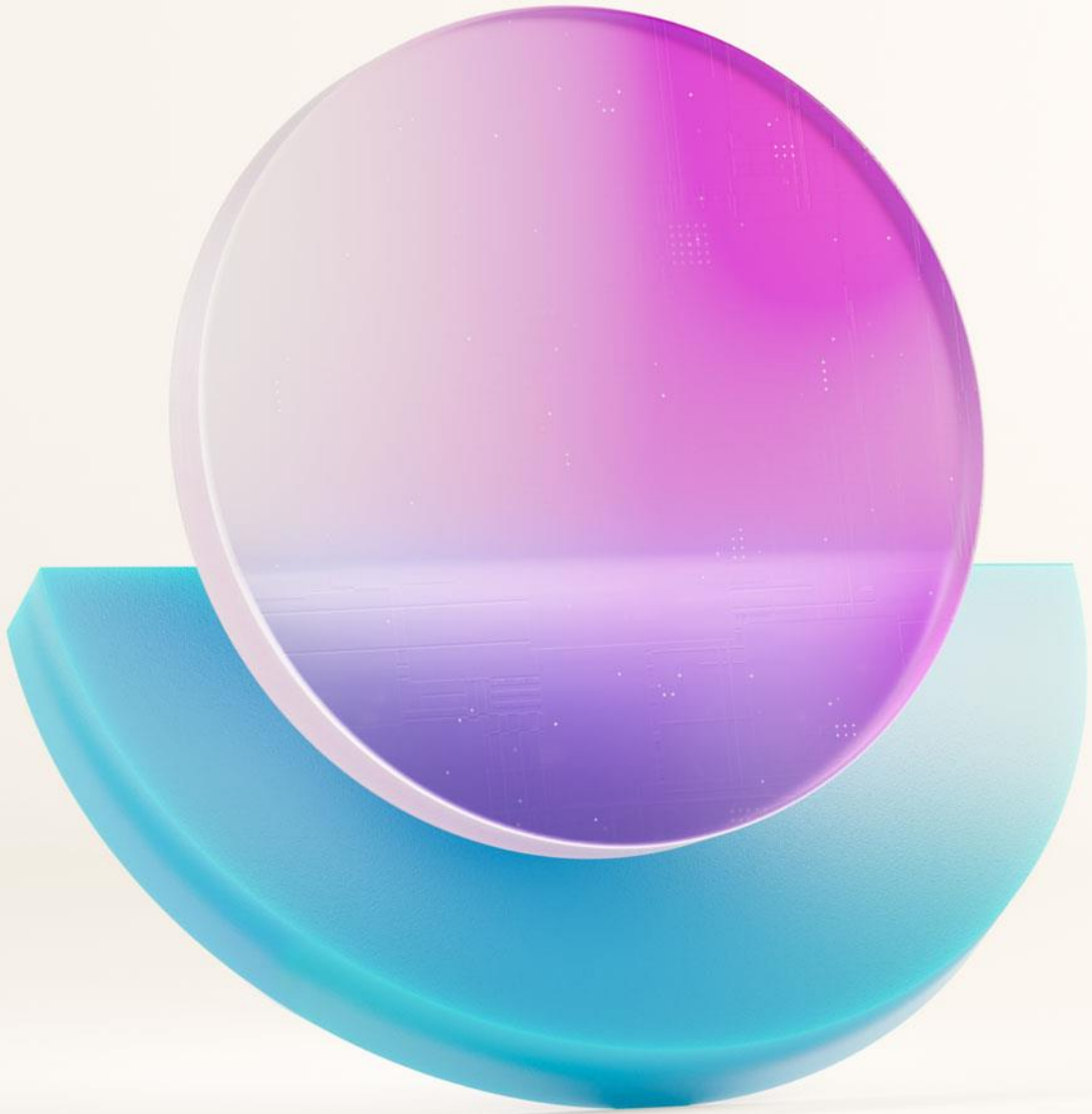
Understand Parameters

Blank Canvas

- Or just start from a blank Canvas

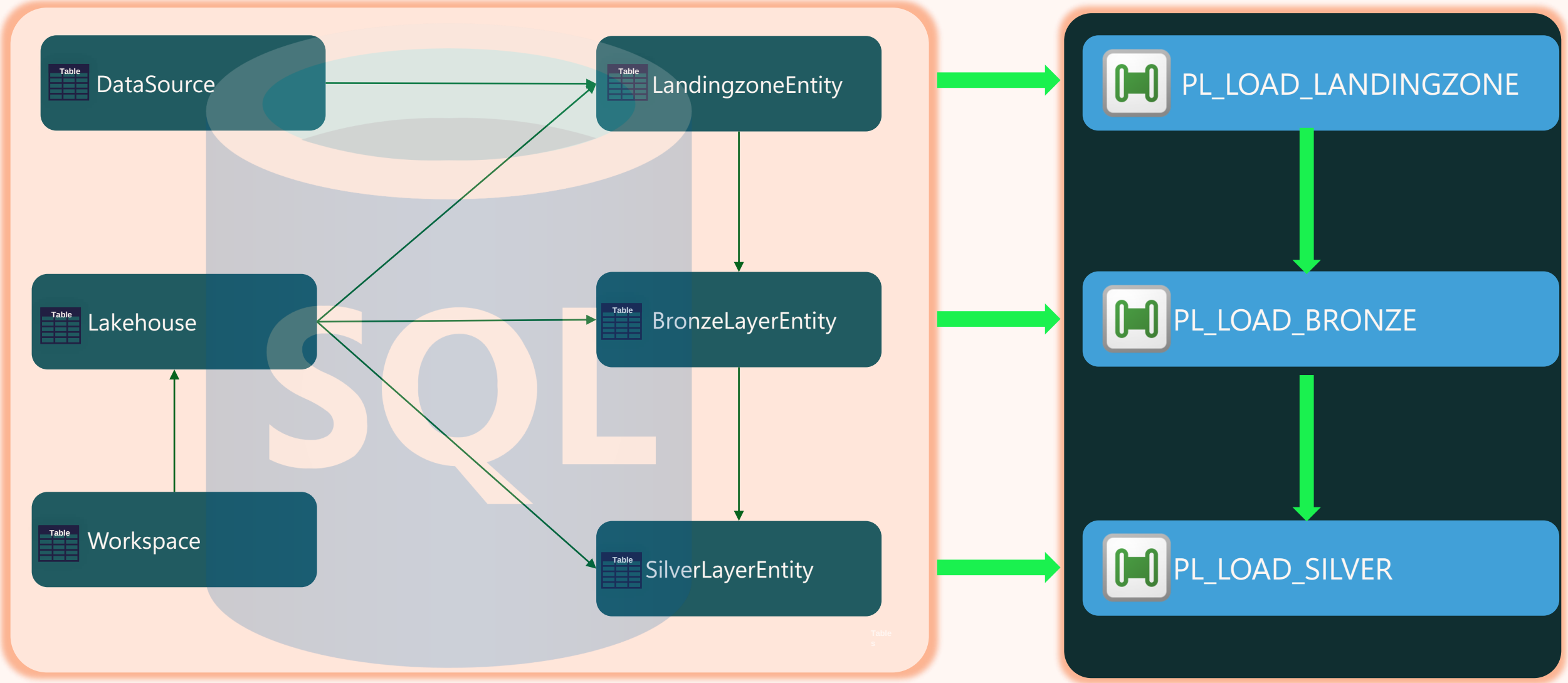


Microsoft Fabric
COMMUNITY CONFERENCE

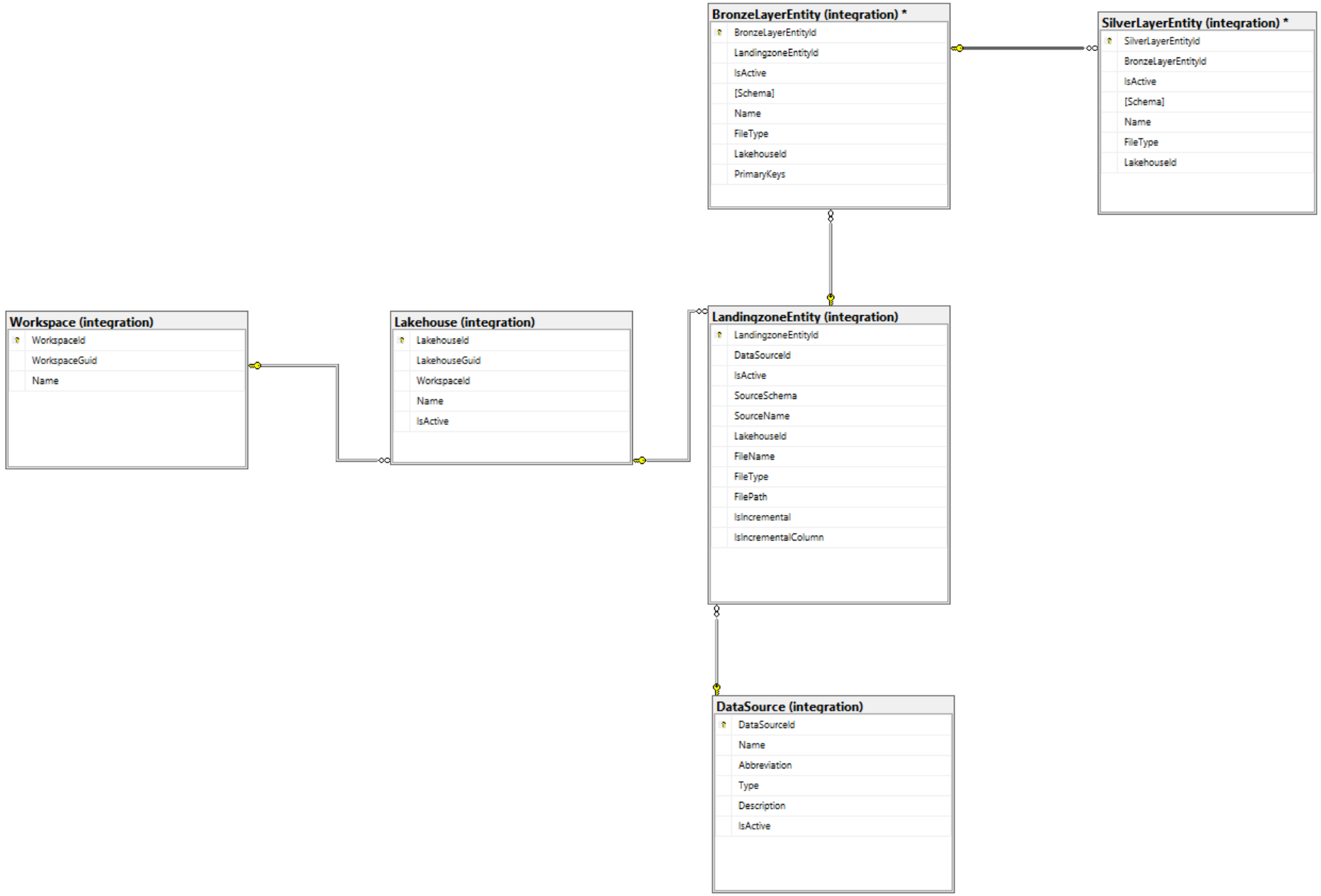


Framework in Azure SQL

Framework

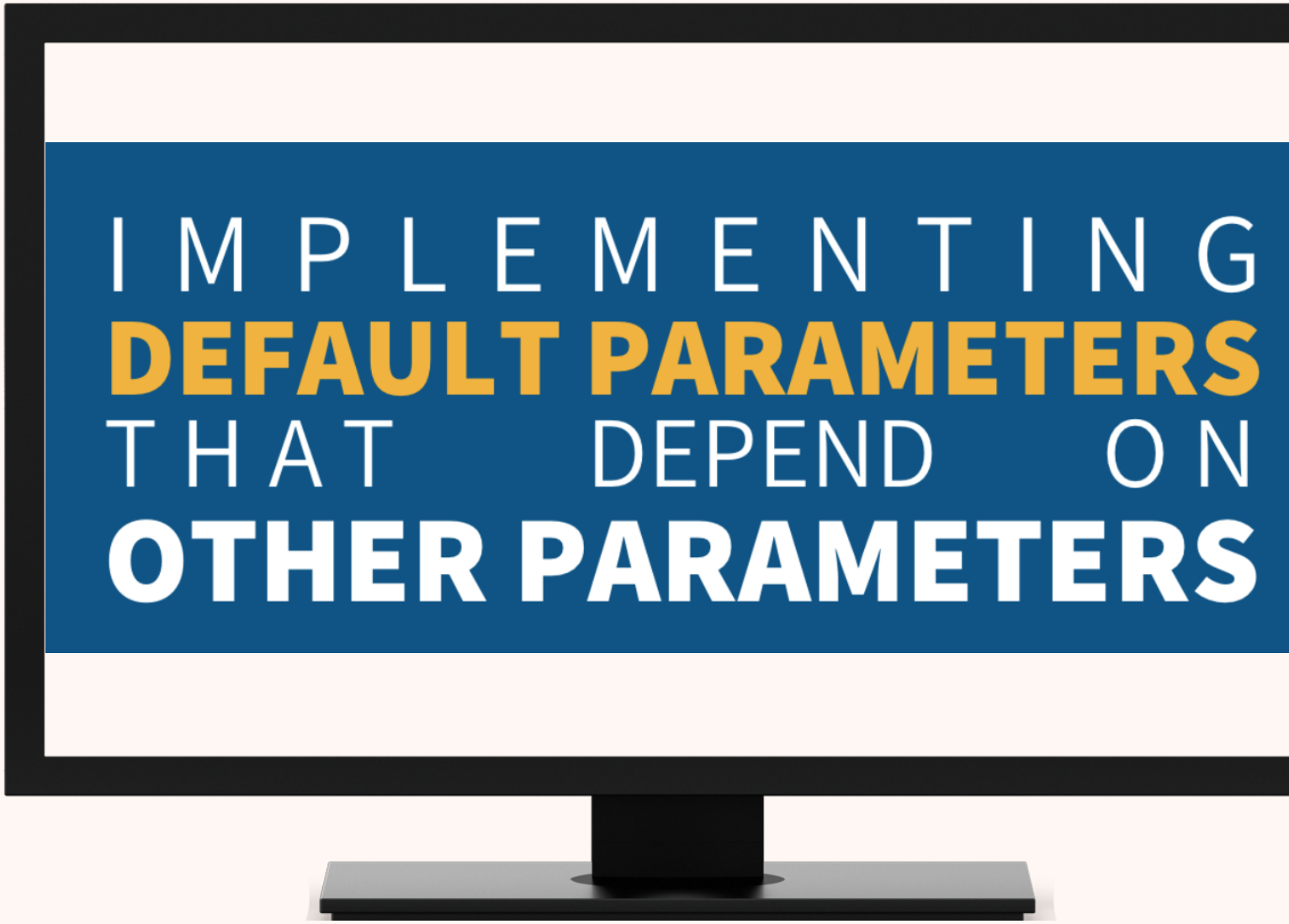

DataSource
LandingzoneEntity
Lakehouse
BronzeLayerEntity
Workspace
SilverLayerEntity
PL_LOAD_LANDINGZONE
PL_LOAD_BRONZE
PL_LOAD_SILVER

Framework


BronzeLayerEntity (integration) *
BronzeLayerEntityId
LandingzoneEntityId
IsActive
[Schema]
Name
FileType
LakehouseId
PrimaryKeys
SilverLayerEntity (integration) *
SilverLayerEntityId
BronzeLayerEntityId
IsActive
[Schema]
Name
FileType
LakehouseId
Workspace (integration)
WorkspaceId
WorkspaceGuid
Name
Lakehouse (integration)
LakehouseId
LakehouseGuid
WorkspaceId
Name
IsActive
LandingzoneEntity (integration)
LandingzoneEntityId
DataSourceId
IsActive
SourceSchema
SourceName
LakehouseId
FileName
FileType
FilePath
IsIncremental
IsIncrementalColumn
DataSource (integration)
DataSourceId
Name
Abbreviation
Type
Description
IsActive

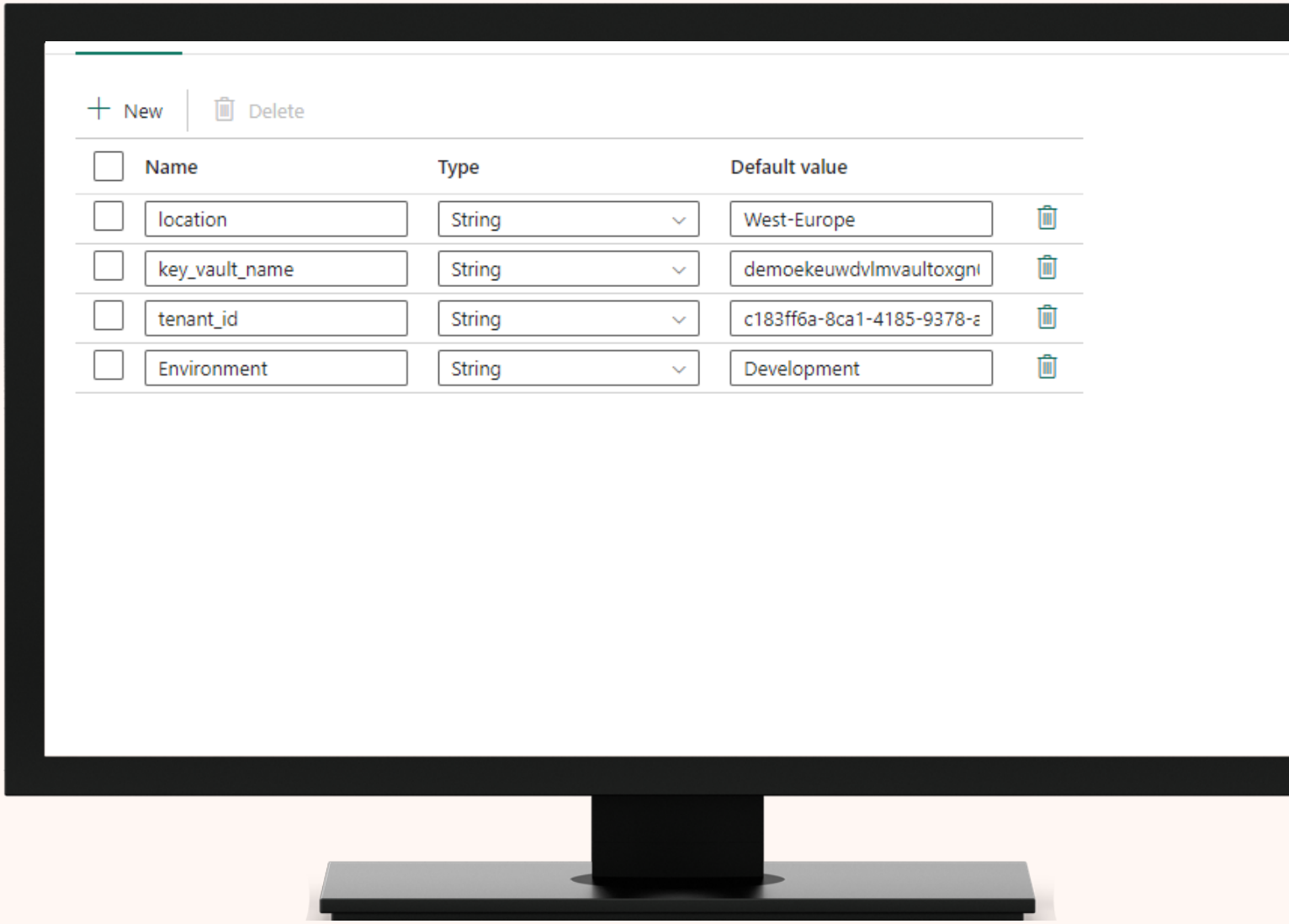
Parameters

- Pipeline Parameters
- Notebook Parameters
- Copy Activity Parameters



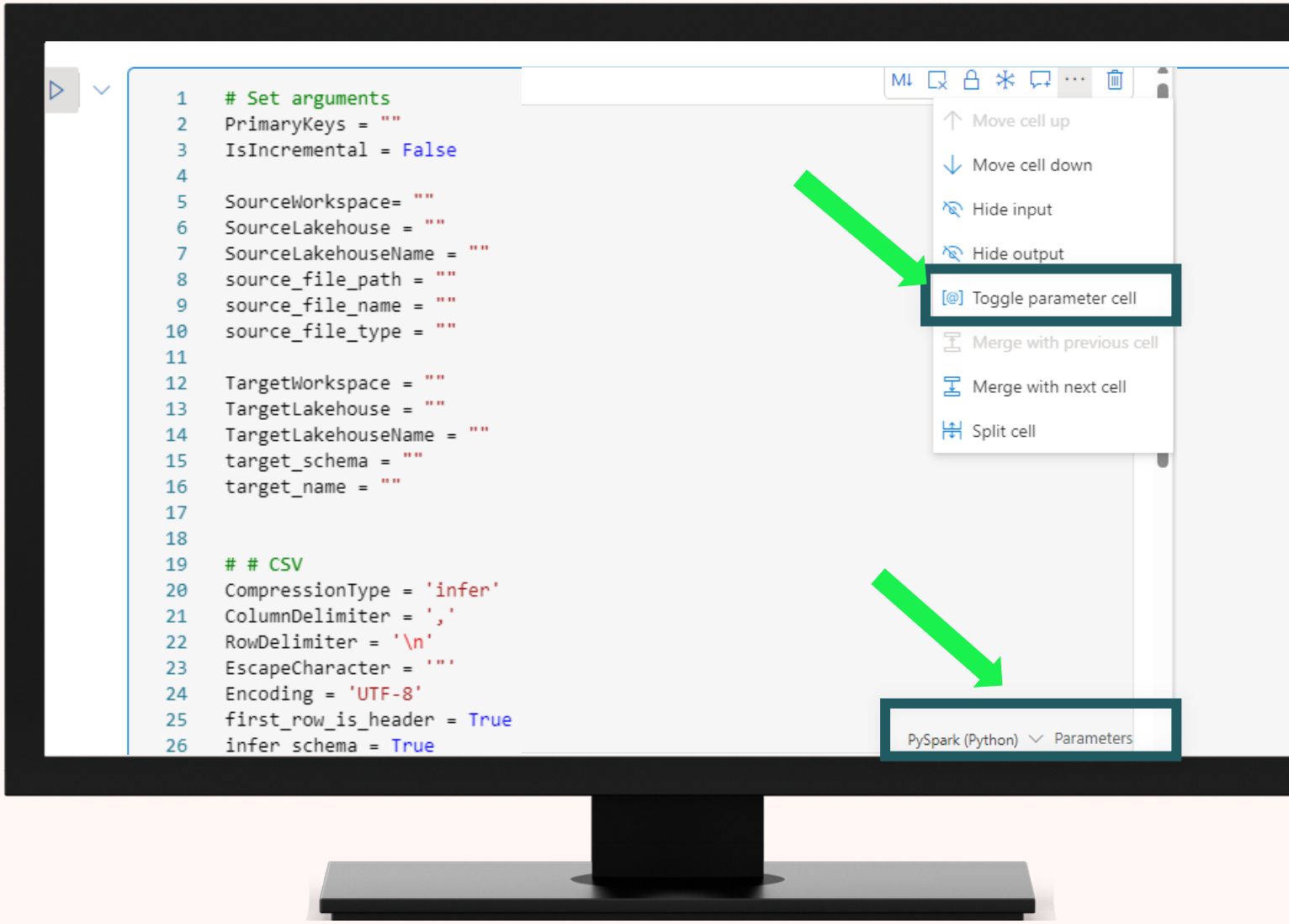
Define Pipeline Parameters

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



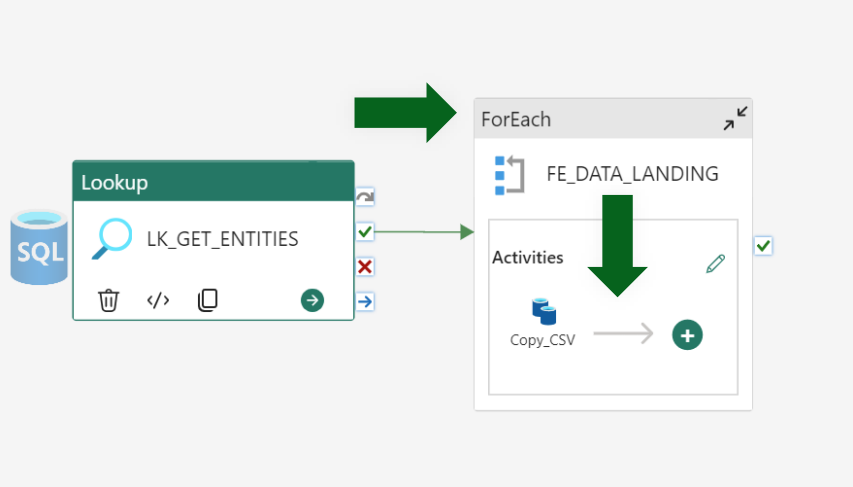
Define Notebook Parameters

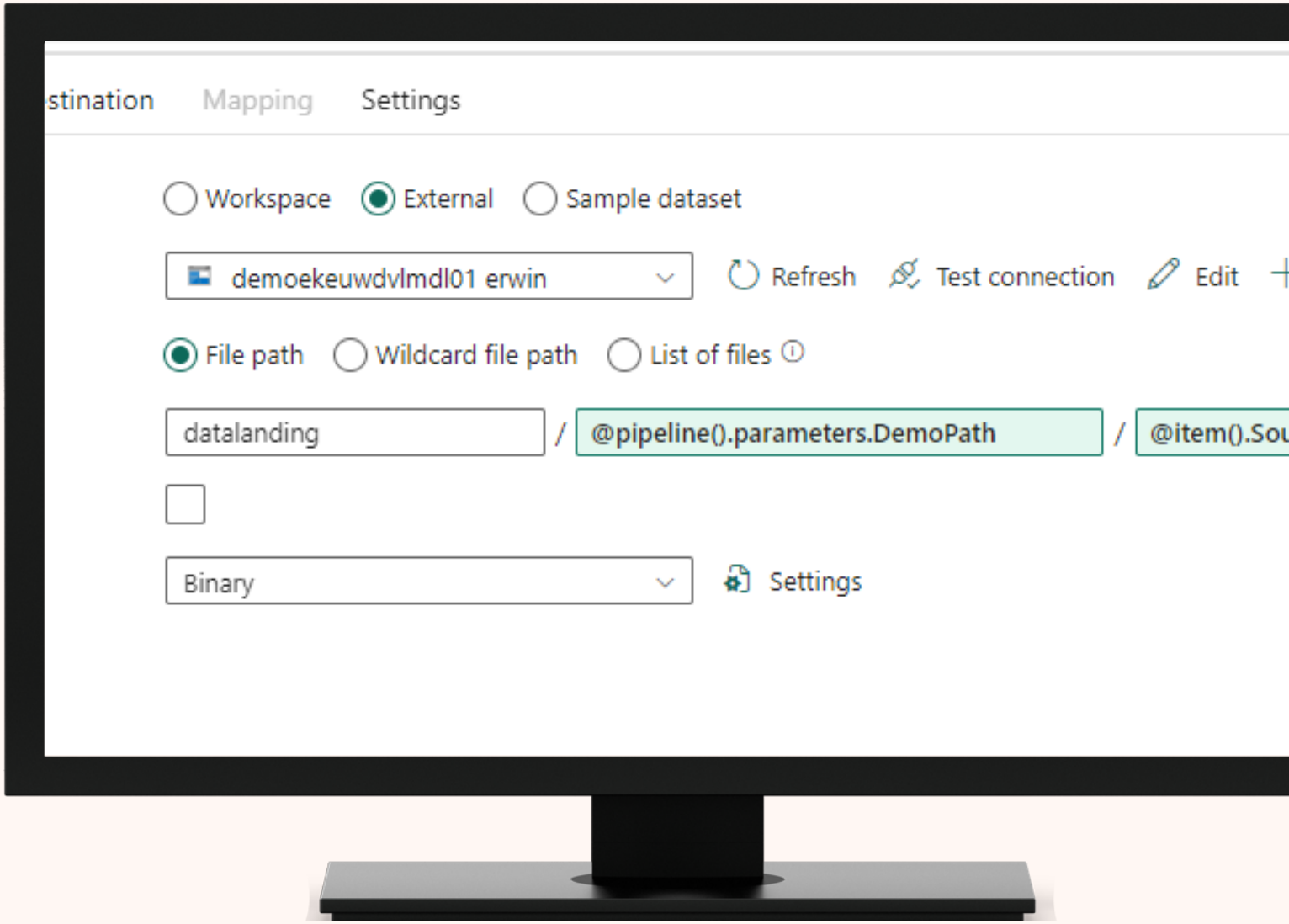
- Pass Parameters from Data Pipeline to Notebook
  - Toggle parameter cell



Copy Activity Parameters

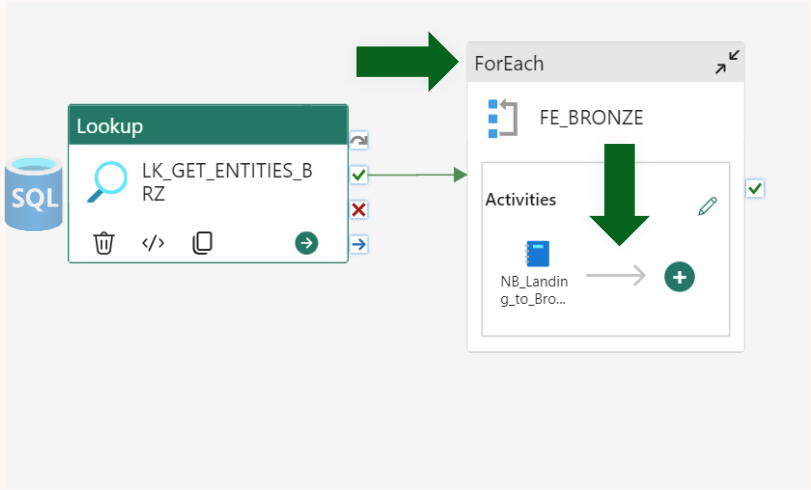
- Pass Parameters from Pipeline to Copy activity
- Use Parameters from For Each Activity





Notebook Parameters

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

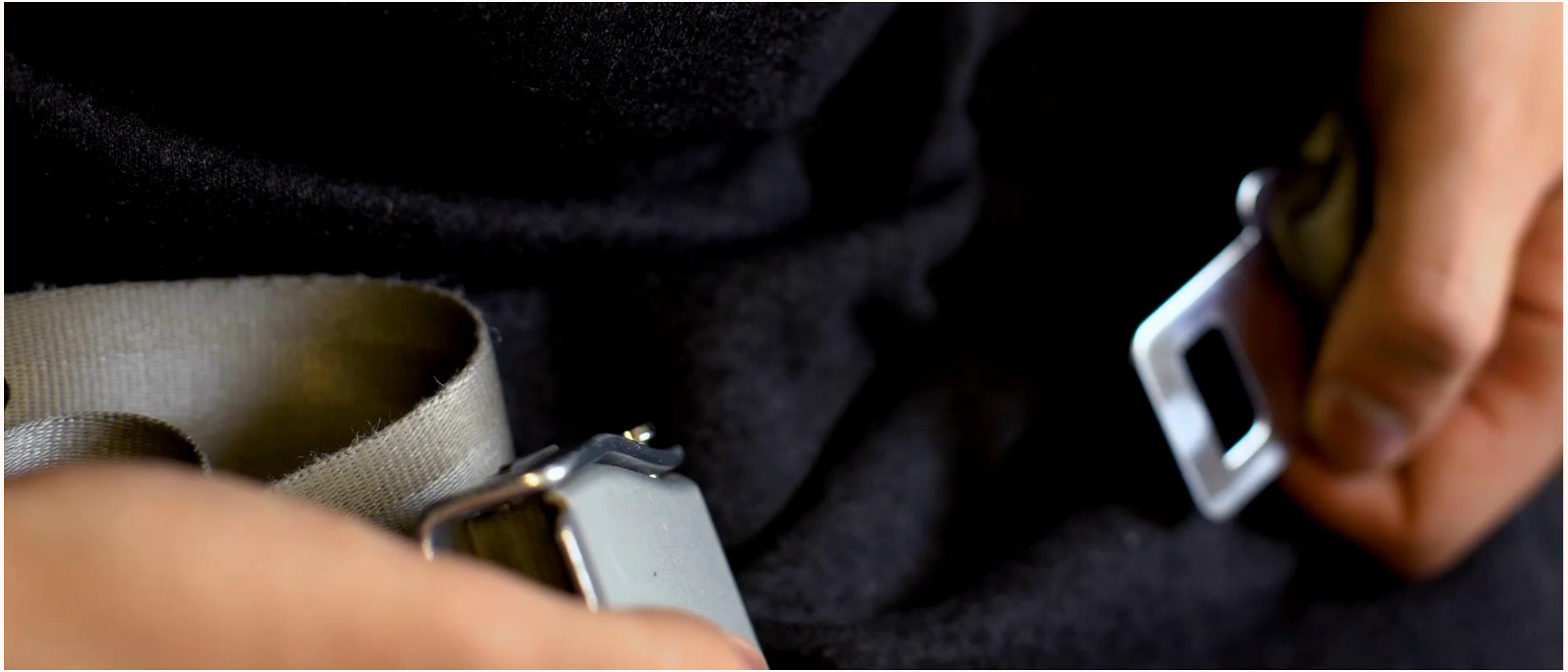


Name	Value
SourceLakehouse	@item().SourceLakehouseId
source_file_path	@item().SourceFilePath
source_file_name	@item().SourceFileName
PrimaryKeys	@item().PrimaryKeys
TargetLakehouse	@item().TargetLakehouseId
target_schema	@item().TargetSchema
target_name	@item().TargetName
SourceWorkspace	@item().SourceWorkspaceId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

Microsoft Fabric
COMMUNITY CONFERENCE

DEMO

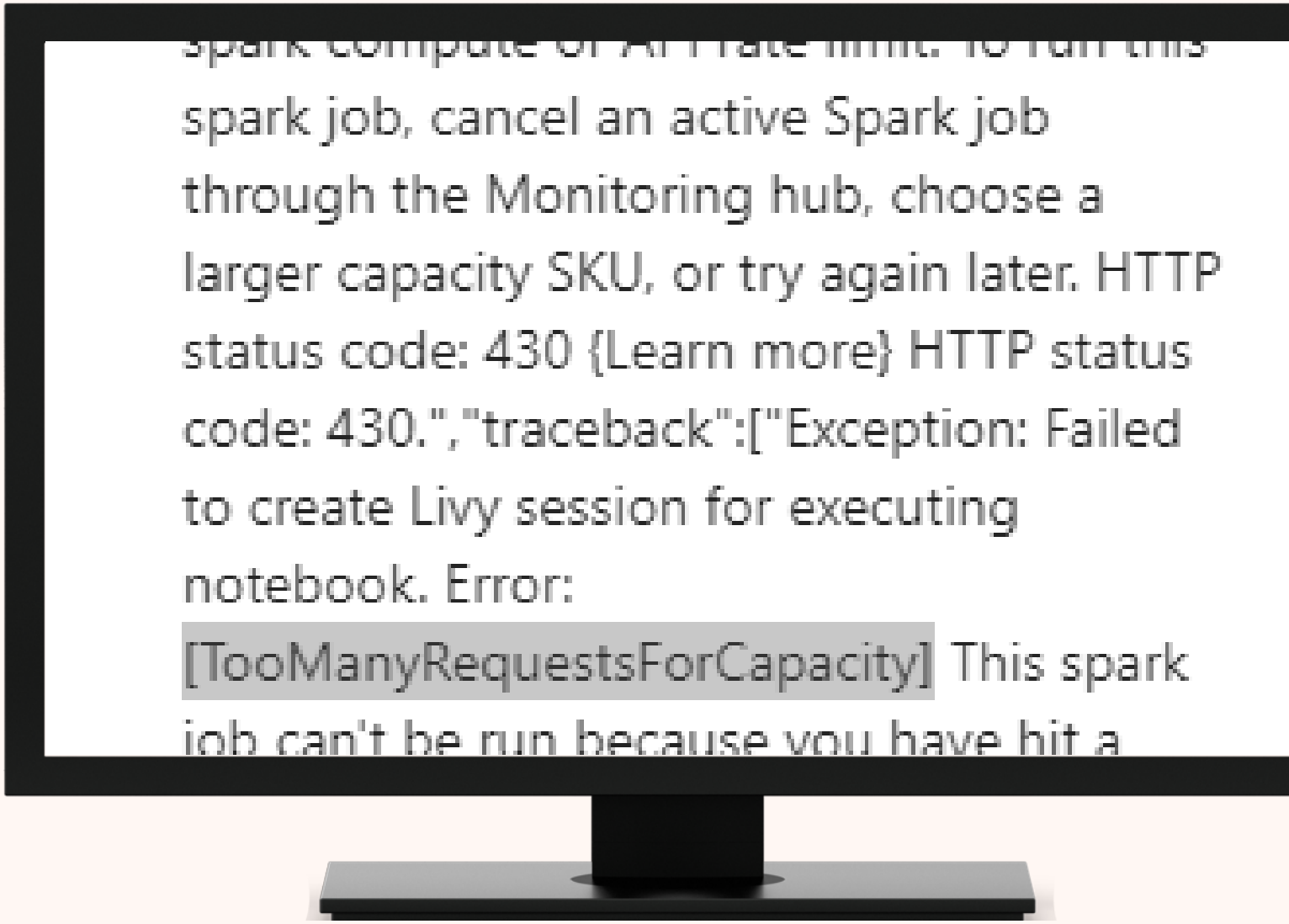
Fasten your seatbelts



‘Optimize the notebook processing’

Fasten your seatbelts

- Notebook Activity
  - TooManyRequestForCapacity
  - Missing High Concurrency in Notebook Activity
  - Default 8 Vcores

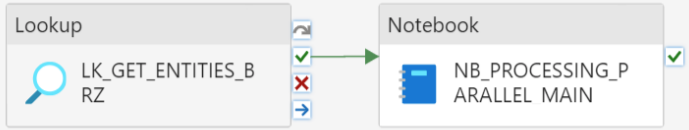


Fasten your seatbelts

- Notebook Activity
  - TooManyRequestForCapacity
  - Missing High Concurrency in Notebook Activity
  - Default 8 Vcores
- Mssparkutils.notebook.runMultiple
  - DAG (directed acyclic graph)
  - Notebook Name/Path
  - Arguments
  - Dependencies

```
DAG = {
    "activities": [
        {
            "name": "NotebookSample", # activity name, must be unique
            "path": "NotebookSample", # notebook path
            "timeoutPerCellInSeconds": 90, # max timeout for each cell, default to 90 seconds
            "args": {"p1": "changed value", "p2": 666}, # notebook parameters
        },
        {
            "name": "NotebookSimple2",
            "path": "NotebookSimple2",
            "timeoutPerCellInSeconds": 120,
            "args": {"p1": "changed value 2", "p2": 777}
        },
        {
            "name": "NotebookSample2.2",
            "path": "NotebookSample2",
            "timeoutPerCellInSeconds": 120,
            "args": {"p1": "changed value 3", "p2": 888},
            "retry": 1,
            "retryIntervalInSeconds": 10,
            "dependencies": ["NotebookSample"] # list of activity names that this activity depends on
        }
    ]
}

mssparkutils.notebook.runMultiple(DAG)
```

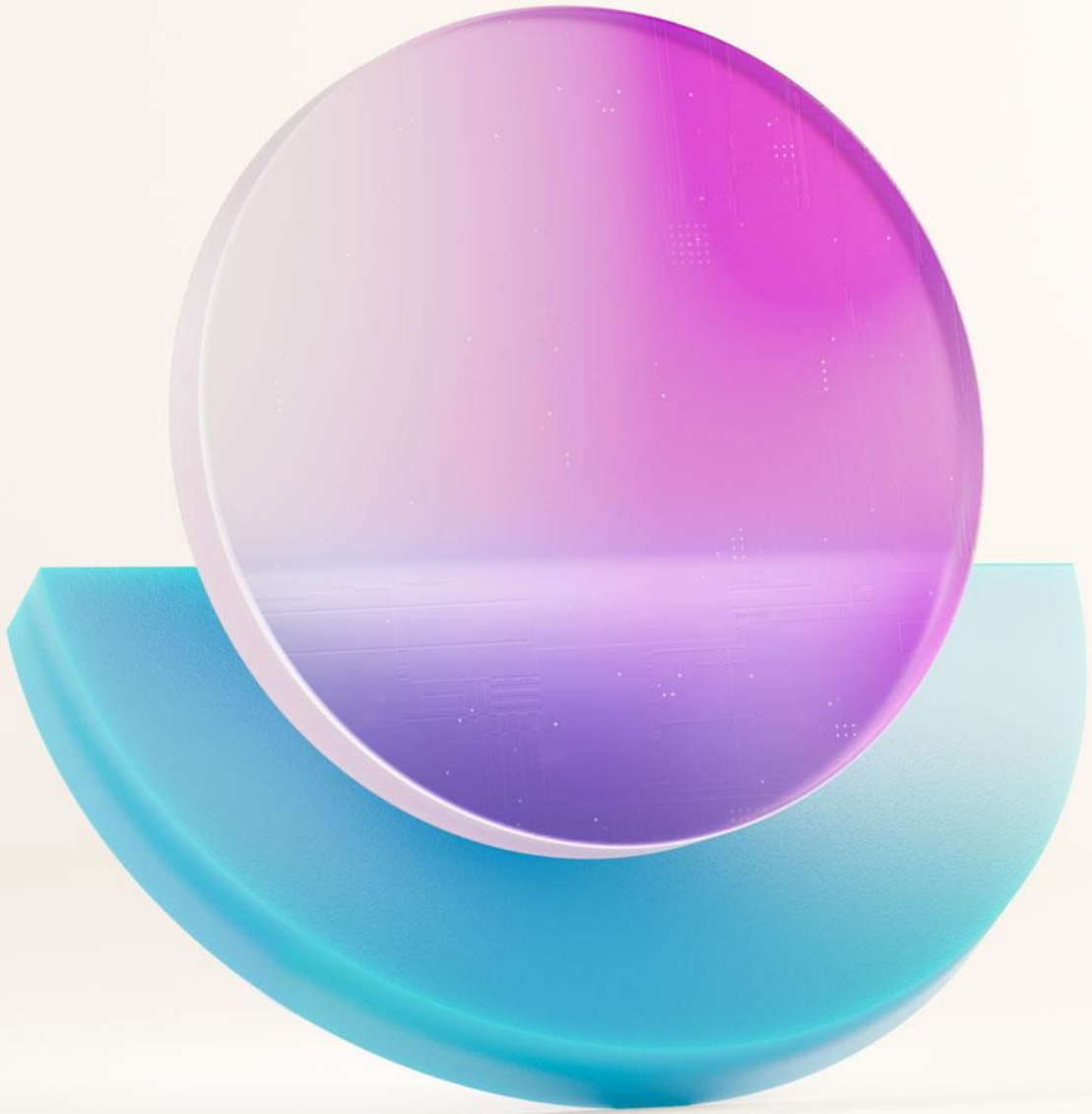


More details on my blog:
How to use mssparkutils.notebook.runMultiple in Notebooks in Microsoft Fabric? - Erwin & Data Analytics (erwindekreuk.com)

Microsoft Fabric
COMMUNITY CONFERENCE

DEMO

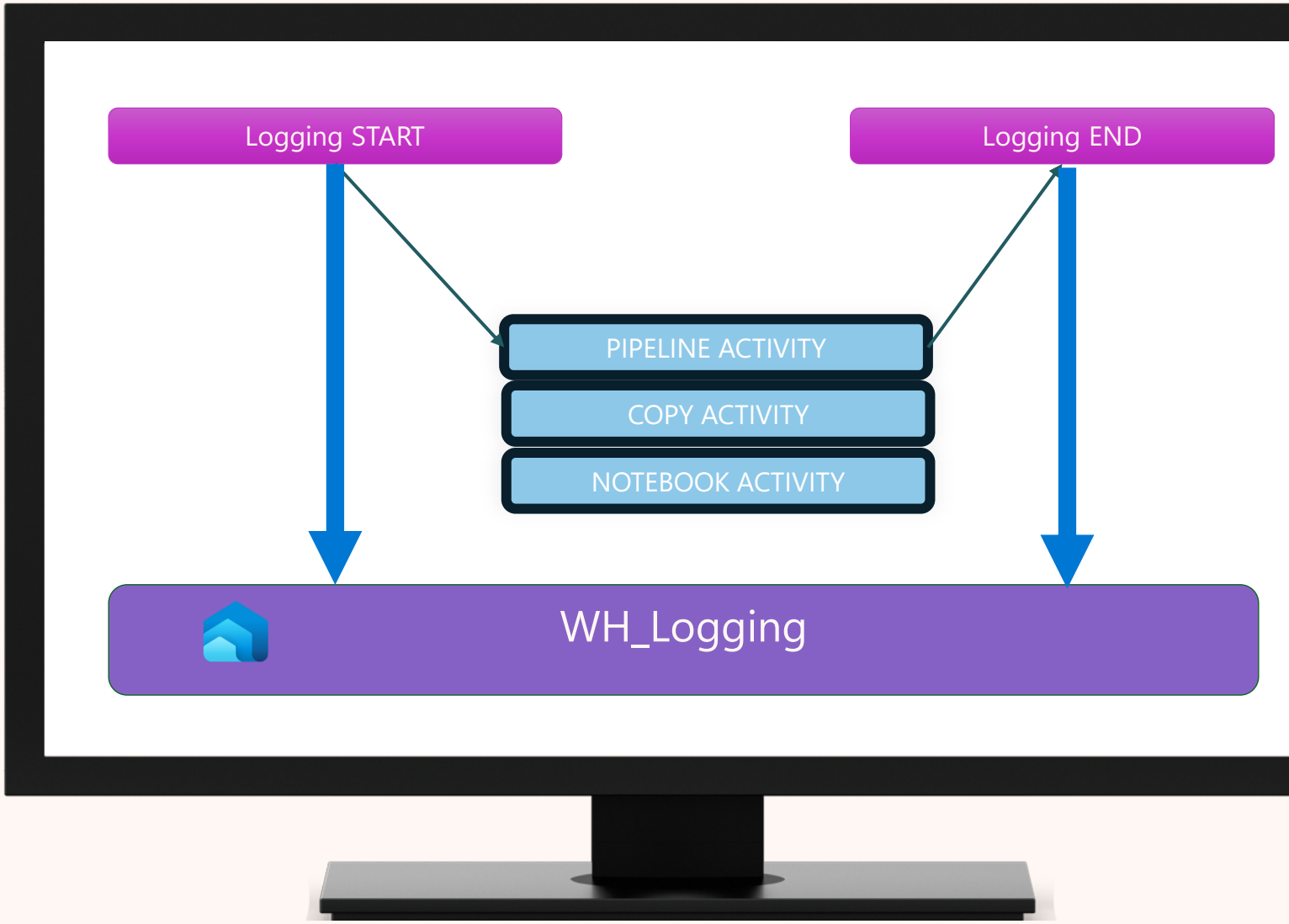
Microsoft Fabric
COMMUNITY CONFERENCE



Logging

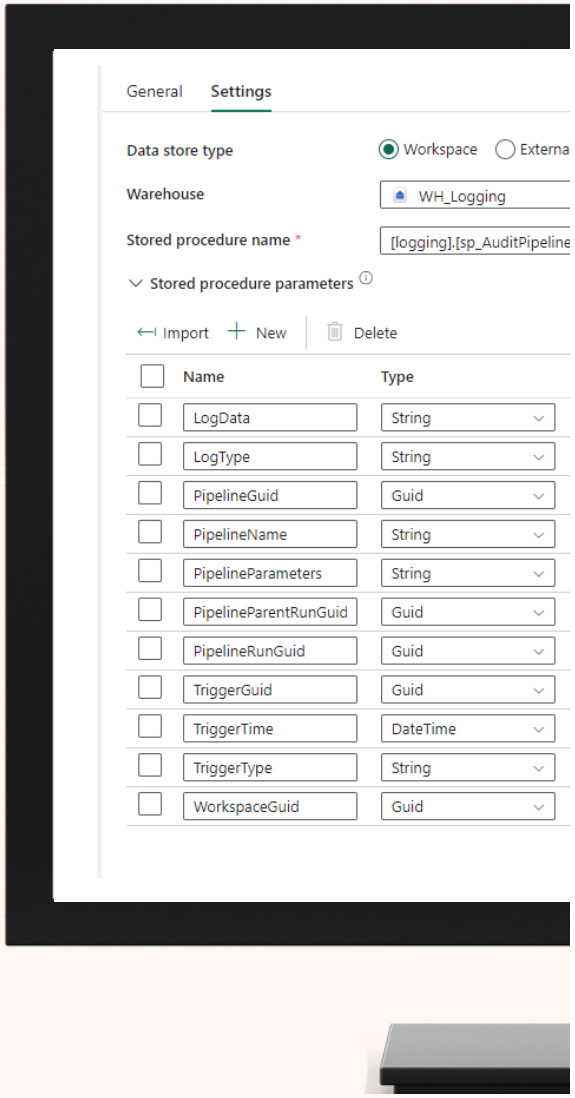
Logging

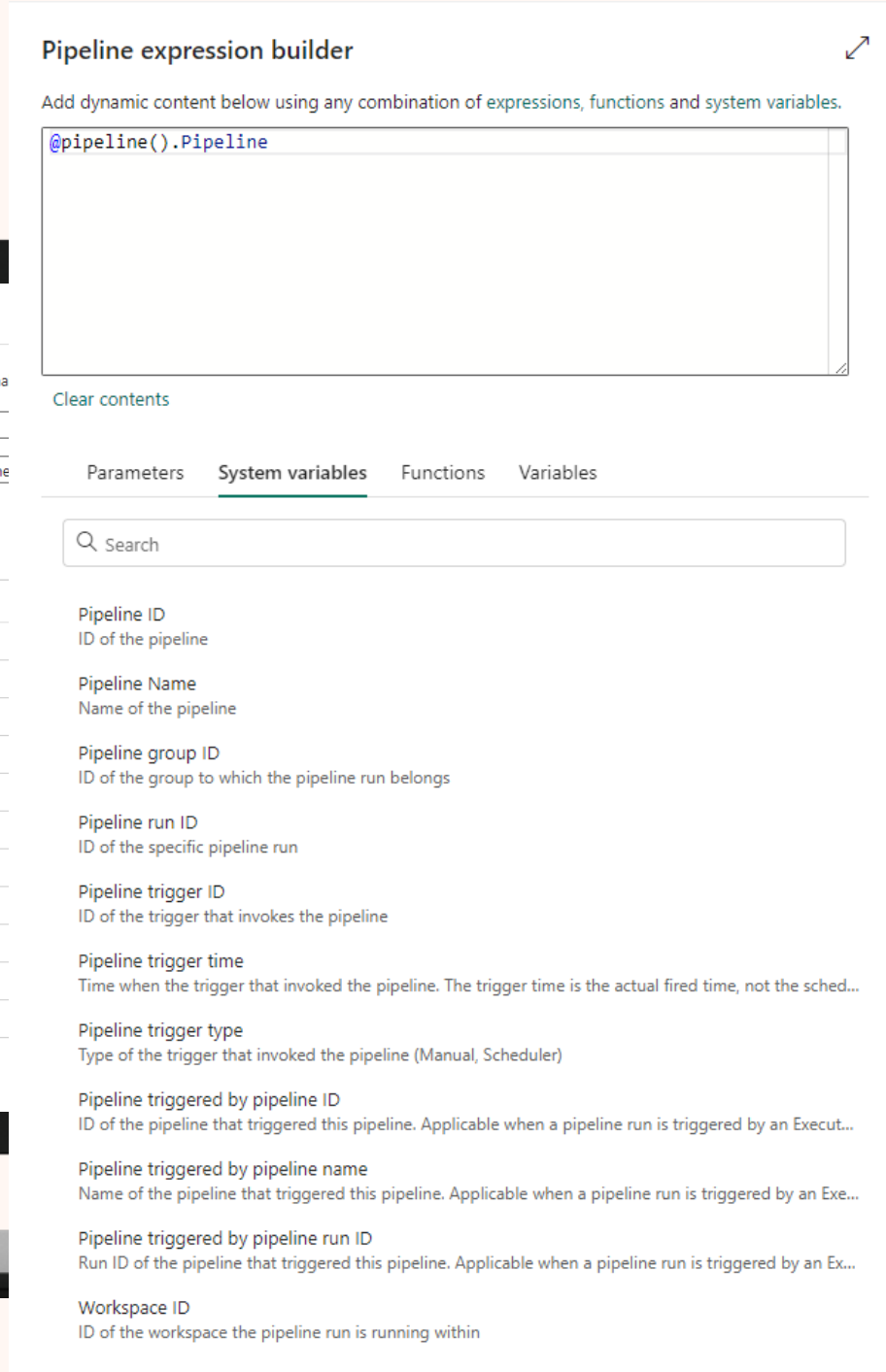
- Log Start and End Time of records
- Log Extracted Records
- Log Execution Failure



Logging

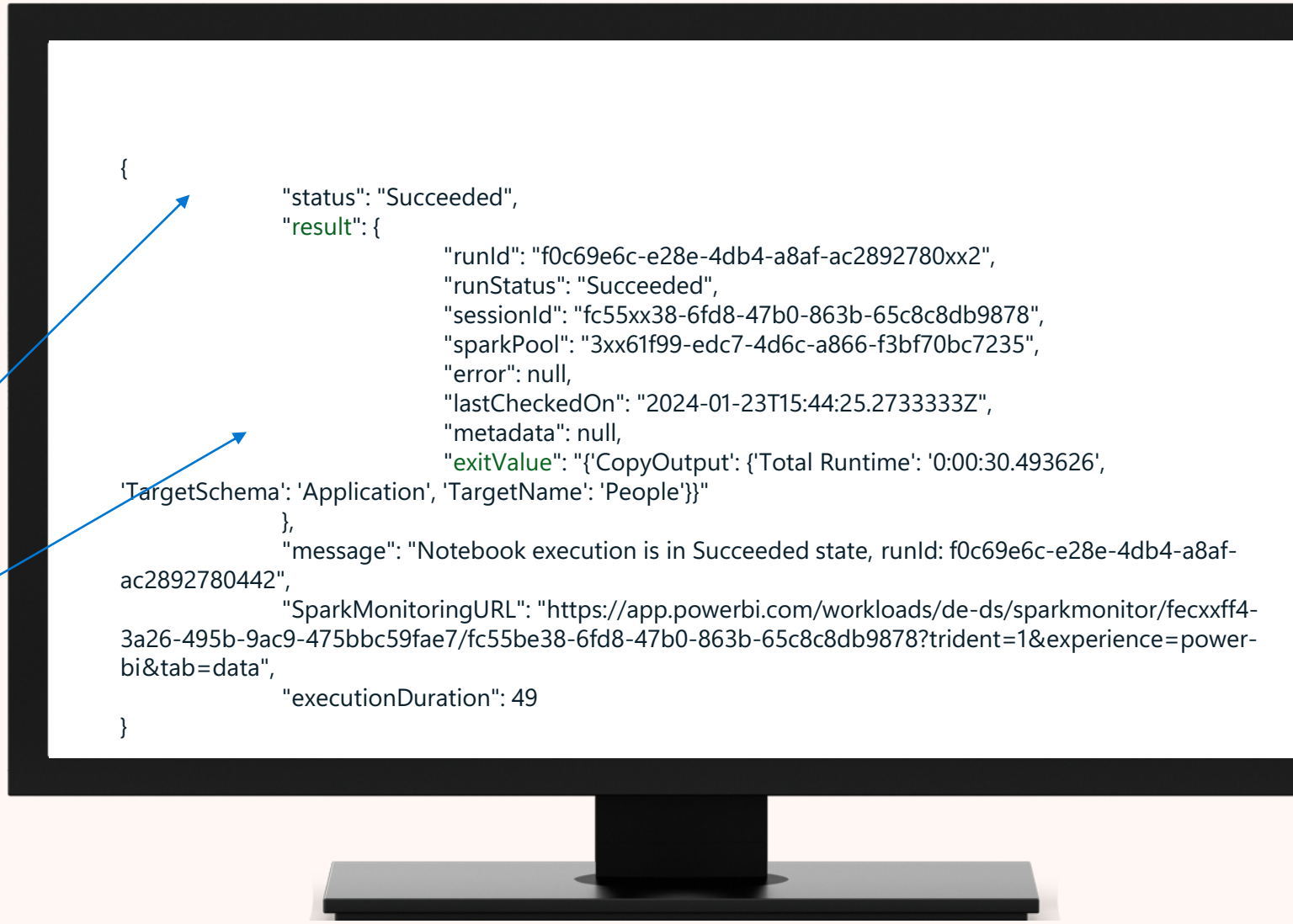
- Add Information about pipelines
- Adding System Variables





Logging

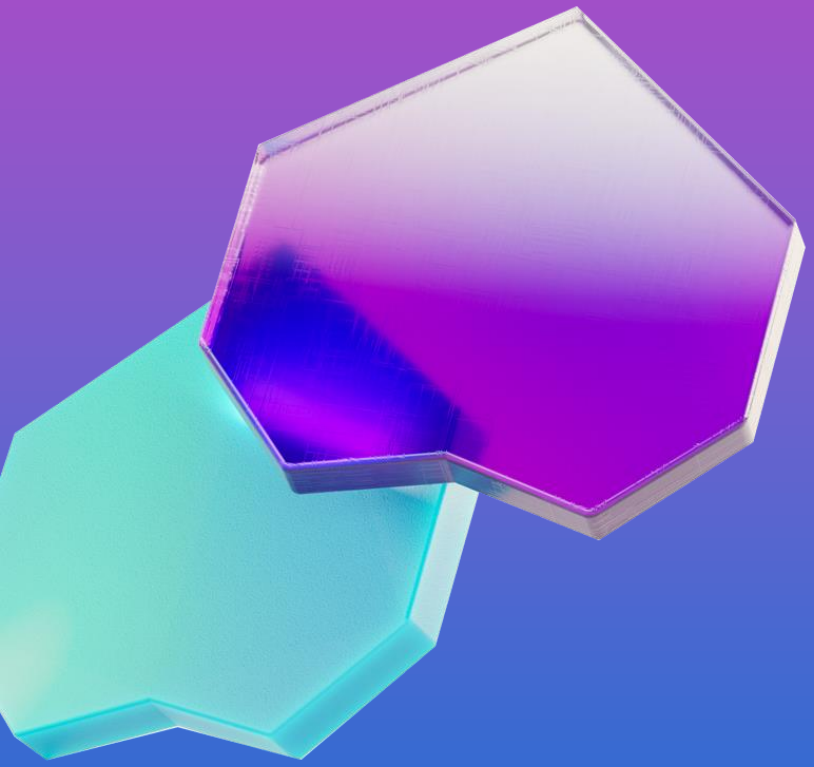
- Add Information about pipelines
- Adding System Variables
- Add Information about Notebooks



Microsoft Fabric
COMMUNITY CONFERENCE

DEMO

Recap



- Can we build Pipelines dynamically?
- Can we extract data from my sources based on MetaData?
- Can we load the active(current) or historical records to a Lakehouse?
- Can we build history from extracted data based on MetaData?

Challenges

- Out of the box fast and easy, less flexible
- Parameterize of connections with Azure Key Vault (Like ADF/Synapse with Linked Services)
- Need for High concurrency Notebook Activity in Pipeline (currently solved with the use of a Notebook executor or mssparkutils.notebook.runMultiple(DAG))
- Data gateway for Pipelines is coming
- Schedule can’t be Parameterized like in ADF/Synapse
- Build in retry to Notebook Activity

Questions?

Let’s connect

Microsoft

Microsoft Fabric
COMMUNITY CONFERENCE

@erwindekreuk
linkedin.com/in/erwindekreuk
erwindekreuk.com
github.com/edkreuk
https://sessionize.com/erwin-de-kreuk/

Thank you

Microsoft® Most Valuable Professional