Unleashing the Potential of Metadata-Driven ELT Framework

Erwin de Kreuk
Principal Consultant - Lead Data & Analytics
InSpark

MVP Microsoft® Most Valuable Professional

Erwin de Kreuk

Principal Consultant – Lead Data & Analytics
InSpark

Erwin de Kreuk
Principal Consultant – Lead Data & Analytics
InSpark

@erwindekreuk

linkedin.com/in/erwindekreuk

erwindekreuk.com

github.com/edkreuk

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

Microsoft® Most Valuable Professional

Let's connect

We Are InSpark

We help organizations **accelerating** their **digital transformation** with impactful Microsoft solutions & expertise

INSPARK

THANK YOU

Data Community Austria

Platinum

smart casual datadesign

redgate

TIMEXTENDER

Gold

b.telligent
smart data. smart decisions.

Lucient

Measure Killer

Spot by NetApp

Silver

veeam

Bronze

ACP CUBIDO Digital Solutions

Power BI Camp
powered by linearis.at

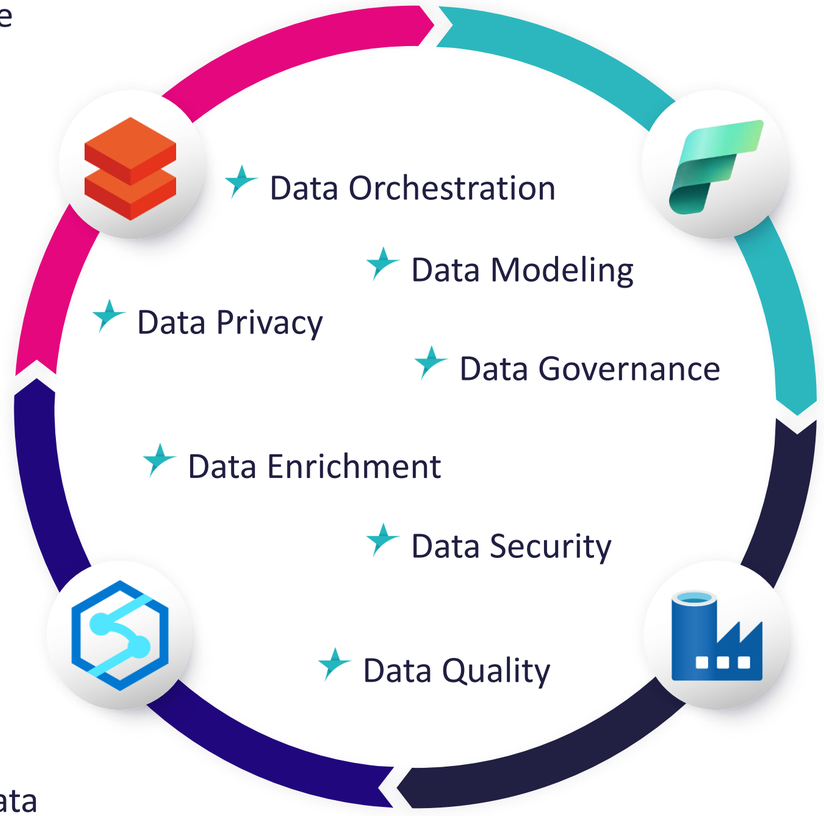
Savory DATA

SOLISYON

Data platform Challenges

'From data source to data model' to report

- Data Lineage
- Data Catalog
- Data Monitoring
- Capacity Control
- Incremental Loading
- Data Alerts
- Version Control
- Creation of Data Pipelines
- Historical Data
- 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

Metadata driven ELT Framework

Objectives of today

- Why
- Out-of-the-box Framework in Synapse
- Custom-made Framework in Fabric
- Recap

Why

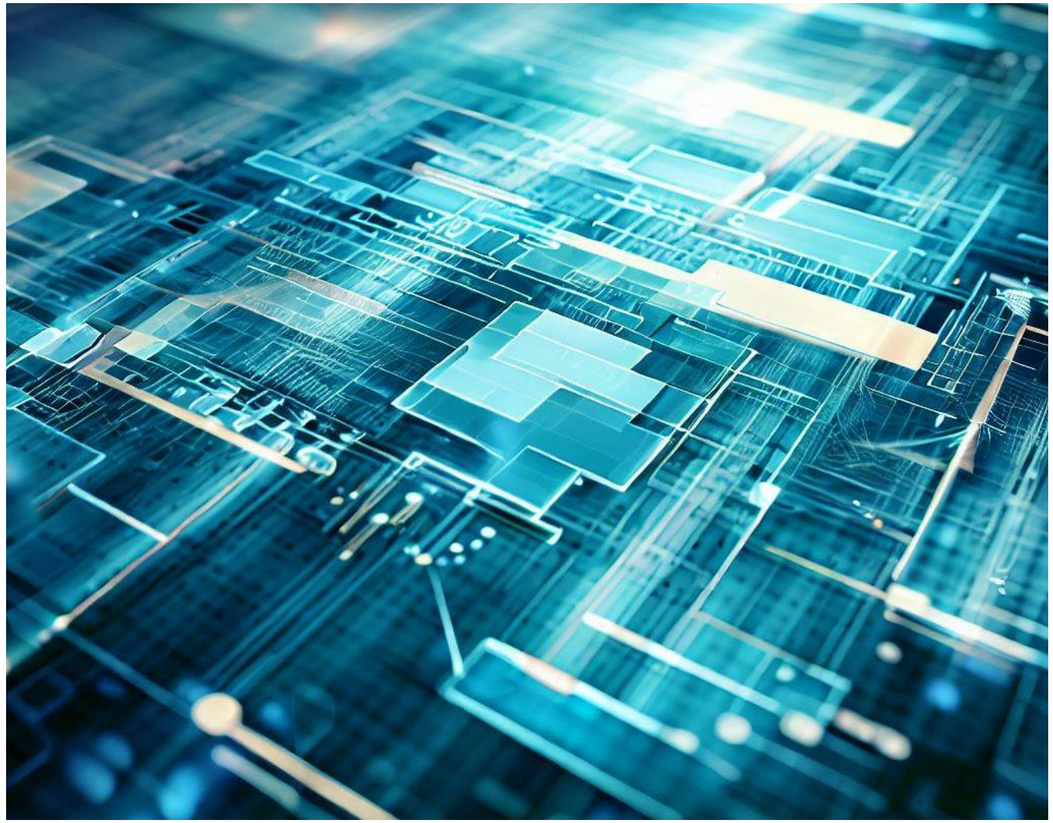
- **Automation**: It allows for the automation of data pipeline creation and management, reducing manual effort. For example, data engineers can use metadata to generate code, schedule tasks, monitor performance, and handle errors.
- **Flexibility**: Enables easy modifications and adaptations to changes in data sources, formats, or business requirements. For example, data engineers can use metadata to update data pipelines without changing the underlying code, or to switch between different data platforms or tools.
- **Scalability**: Facilitates the scaling of data pipelines as data volumes or complexity increase. For example, data engineers can use metadata to distribute data processing across multiple nodes, or to optimize data storage and retrieval.
- **Traceability**: Enhances traceability and auditability of data as it moves through the pipeline. For example, data engineers can use metadata to track data provenance, lineage, and quality, or to provide documentation and reports.
- **Quality Assurance**: Improves data quality by enabling better monitoring and validation of data at each stage. For example, data engineers can use metadata to define data quality rules, metrics, and thresholds, or to perform data cleansing and transformation.

INSPARK

Out-of-the-Box Framework

Out-of-the-Box

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



INSPARK

Custom-Made Framework

Custom-Made

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

INSPARK

Out-of-the-Box Framework

- Available by default in Azure Synapse Analytics and Azure Data Factory
- Azure SQL Database as requirement (Linked Service)
- Almost next, next, finish

Metadata-driven copy task

You will get parameterized pipelines which can read metadata from an external store to load data at a large scale.

INSPARK

Metadata-driven Framework

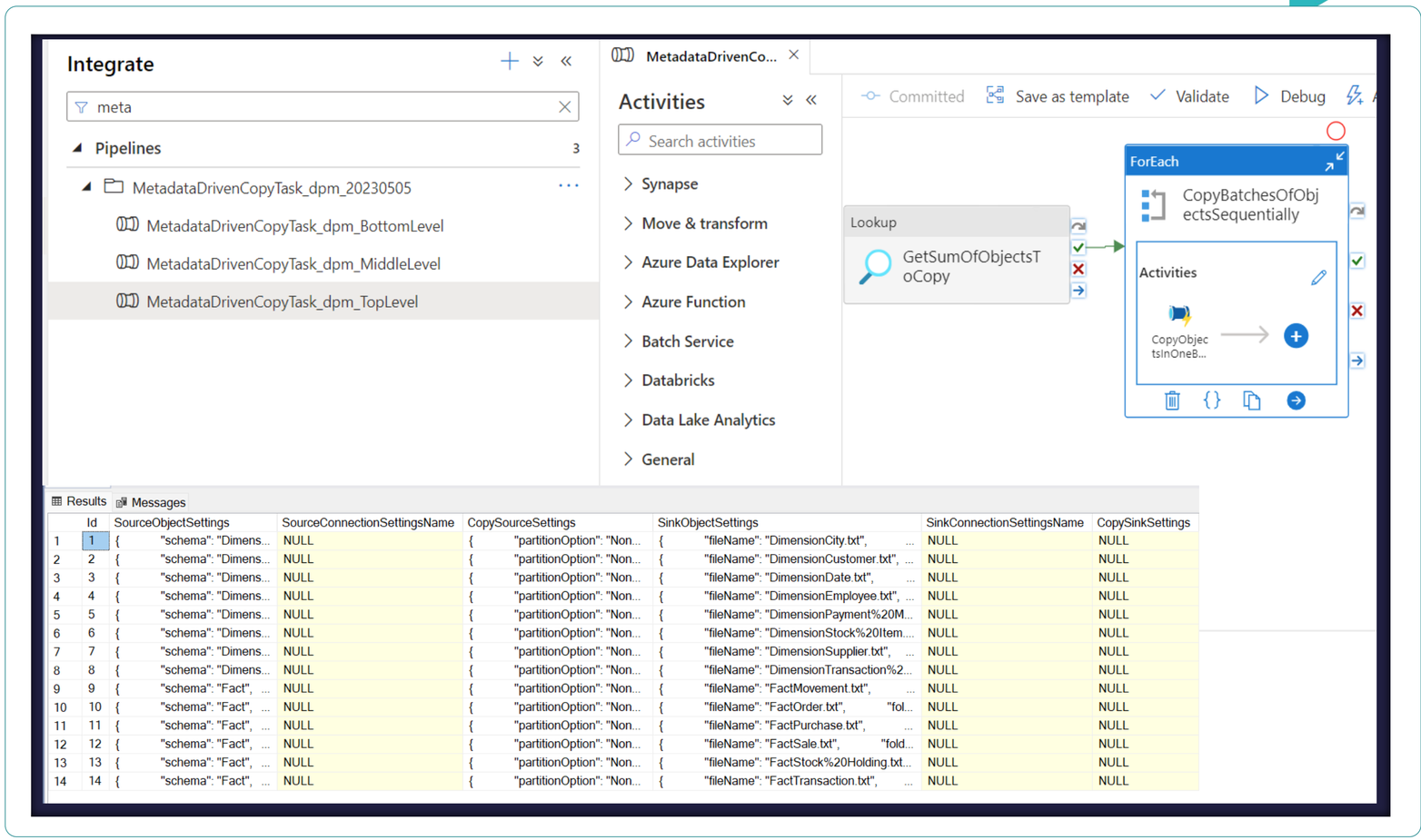
Out-of-the-Box Framework

- Connect to table data store
- Define Schema and Table name
- Define Schedule
- Connect to Source
- Select Tables
- Define full or Incremental
- Define Destination
- Create tables in datastore

INSPARK

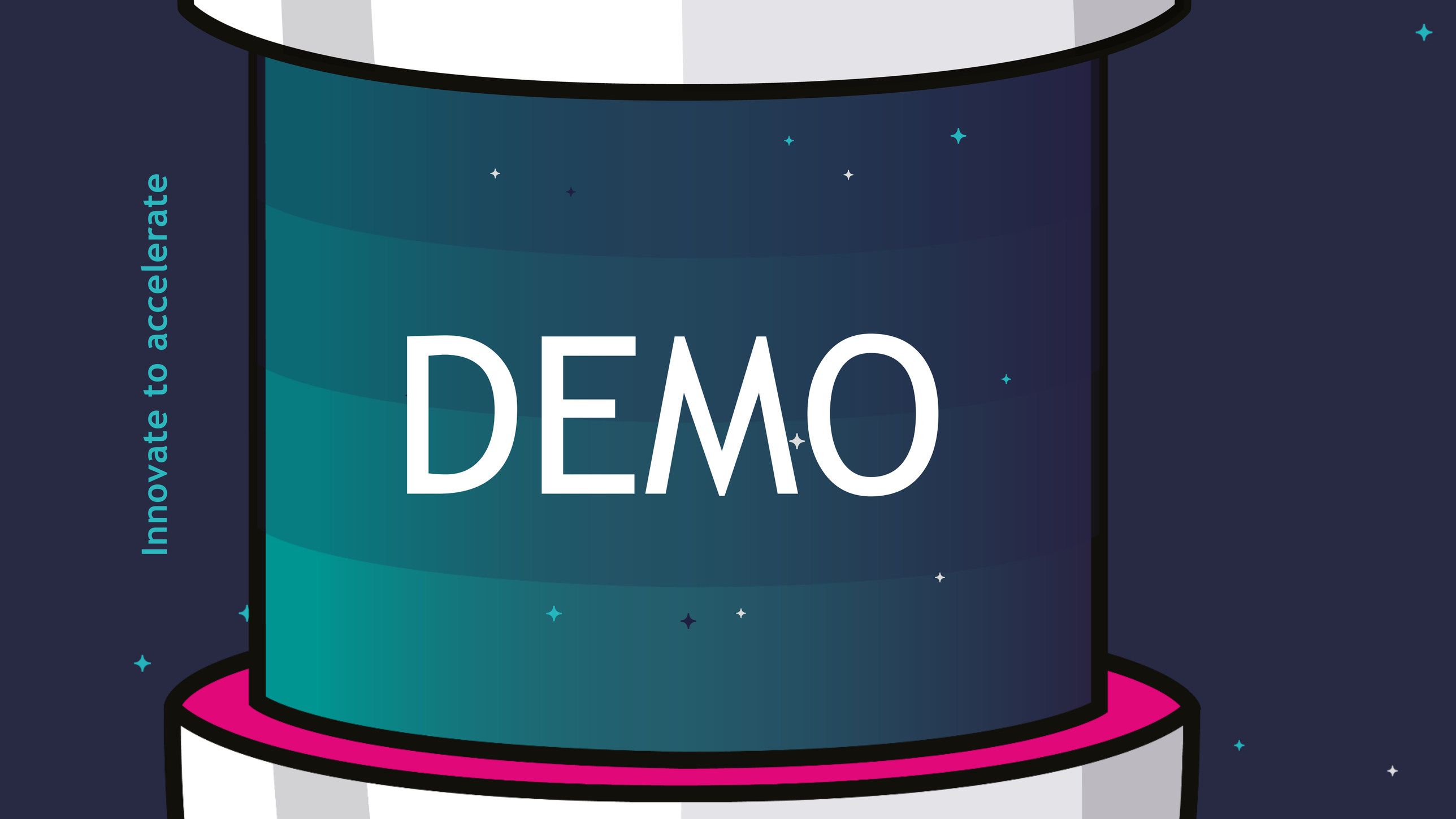
Out-of-the-Box Framework

- Execute pipeline
- Top
  - This pipeline will calculate the total number of objects (tables etc.) required to be copied in this run
- Middle
  - This pipeline will copy one batch of objects.
- Bottom
  - This pipeline will copy objects from one group



	Id	SourceObjectSettings	SourceConnectionSettingsName	CopySourceSettings	SinkObjectSettings	SinkConnectionSettingsName	CopySinkSettings
1	1	{ "schema": "Dimens...	NULL	{ "partitionOption": "Non...	{ "fileName": "DimensionCity.txt", ...	NULL	NULL
2	2	{ "schema": "Dimens...	NULL	{ "partitionOption": "Non...	{ "fileName": "DimensionCustomer.txt", ...	NULL	NULL
3	3	{ "schema": "Dimens...	NULL	{ "partitionOption": "Non...	{ "fileName": "DimensionDate.txt", ...	NULL	NULL
4	4	{ "schema": "Dimens...	NULL	{ "partitionOption": "Non...	{ "fileName": "DimensionEmployee.txt", ...	NULL	NULL
5	5	{ "schema": "Dimens...	NULL	{ "partitionOption": "Non...	{ "fileName": "DimensionPayment%20M...	NULL	NULL
6	6	{ "schema": "Dimens...	NULL	{ "partitionOption": "Non...	{ "fileName": "DimensionStock%20Item....	NULL	NULL
7	7	{ "schema": "Dimens...	NULL	{ "partitionOption": "Non...	{ "fileName": "DimensionSupplier.txt", ...	NULL	NULL
8	8	{ "schema": "Dimens...	NULL	{ "partitionOption": "Non...	{ "fileName": "DimensionTransaction%2...	NULL	NULL
9	9	{ "schema": "Fact", ...	NULL	{ "partitionOption": "Non...	{ "fileName": "FactMovement.txt", ...	NULL	NULL
10	10	{ "schema": "Fact", ...	NULL	{ "partitionOption": "Non...	{ "fileName": "FactOrder.txt", "fol...	NULL	NULL
11	11	{ "schema": "Fact", ...	NULL	{ "partitionOption": "Non...	{ "fileName": "FactPurchase.txt", ...	NULL	NULL
12	12	{ "schema": "Fact", ...	NULL	{ "partitionOption": "Non...	{ "fileName": "FactSale.txt", "fold...	NULL	NULL
13	13	{ "schema": "Fact", ...	NULL	{ "partitionOption": "Non...	{ "fileName": "FactStock%20Holding.txt...	NULL	NULL
14	14	{ "schema": "Fact", ...	NULL	{ "partitionOption": "Non...	{ "fileName": "FactTransaction.txt", ...	NULL	NULL

INSPARK


Innovate to accelerate
DEMO

Metadata-driven Framework

Custom-Made Framework

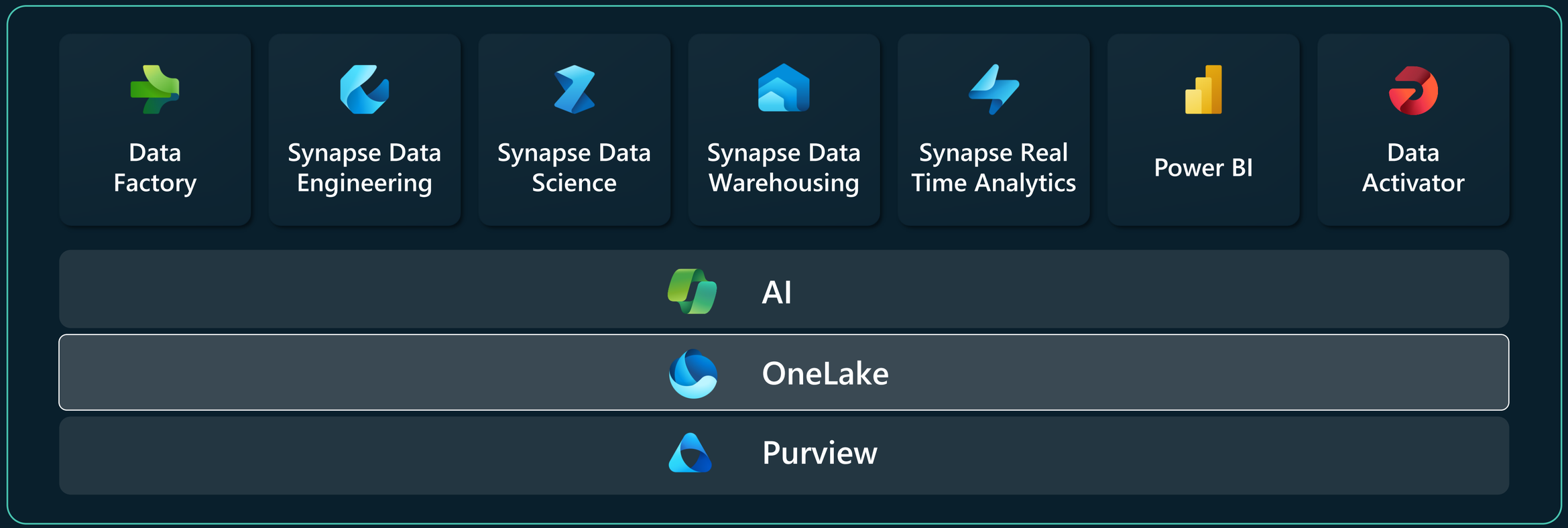
- Based on parameters
- Meta data => Azure SQL Database / Json / ……
- Microsoft Fabric but also on Azure Synapse Analytics and Azure Data Factory
- Based on the Medaillon Architecture

Microsoft Fabric

Who is already using Parameters?

Microsoft Fabric

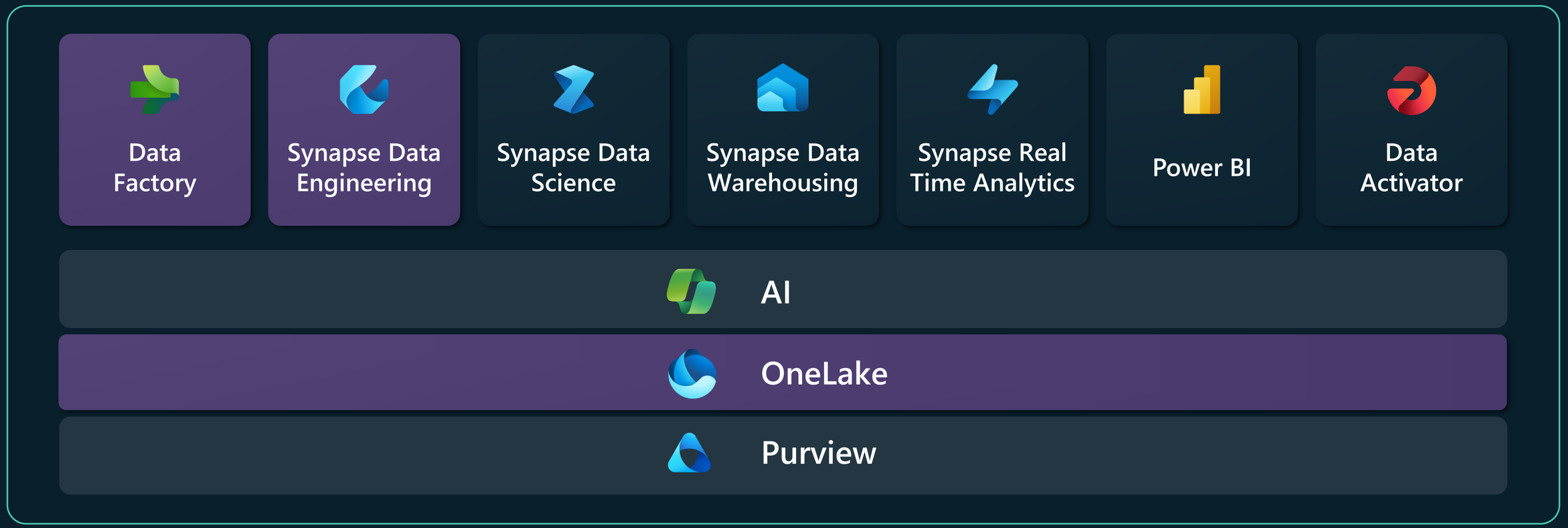
The unified data platform for the era of AI



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

Medaillon Architecture

'Data processing in different stages'

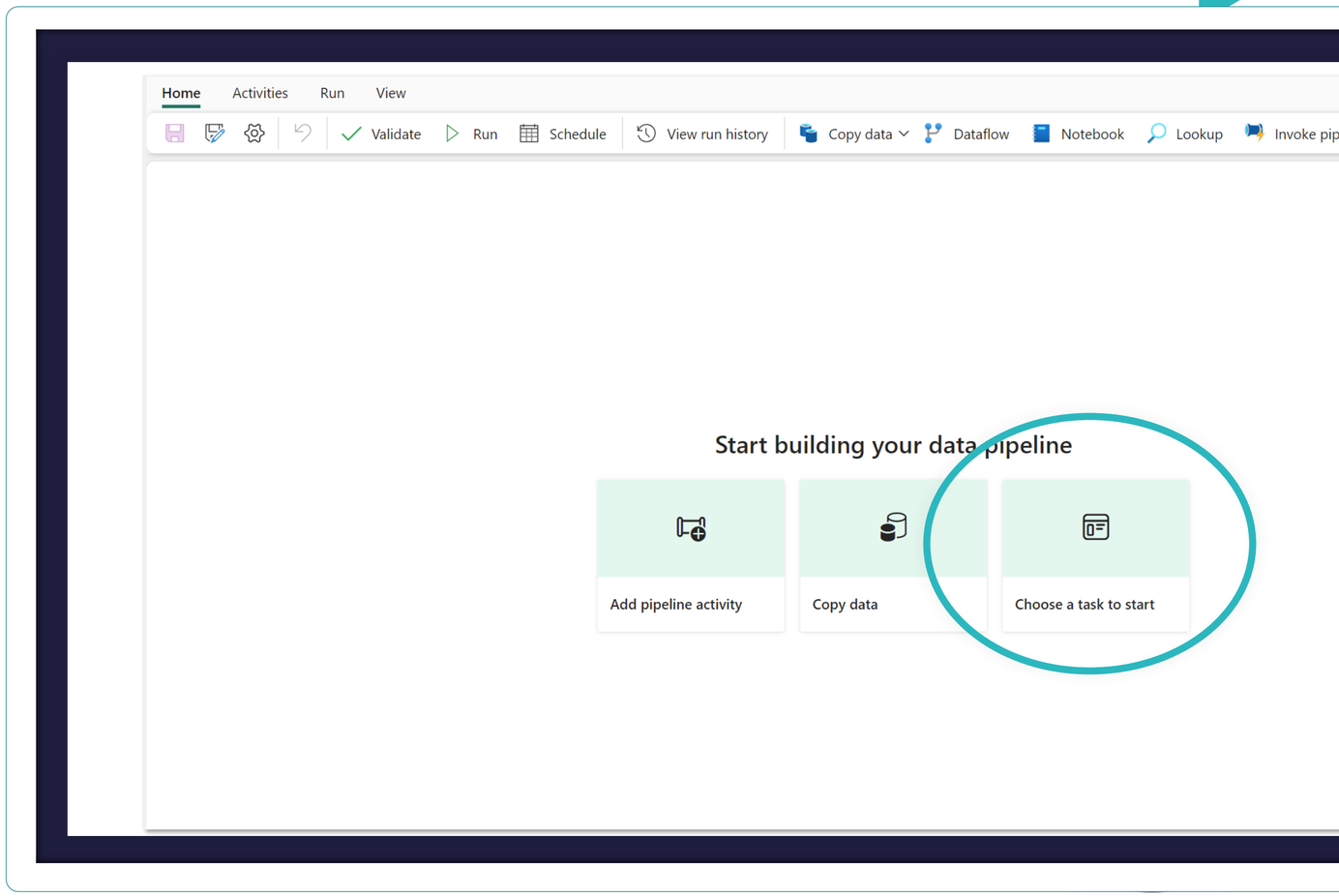
Features

- **Landing zone**
  - Structured data
  - Unstructured data
  - Incremental loads
  - Data as is
- **Bronze layer**
  - Deduplicate data
  - Add datatypes
  - Data can be inconsistent
- **Silver layer**
  - Historical Data
  - Data quality rules
  - Data Cleansing
  - Validated data
  - No business model/data
- **Gold layer**
  - Dimensions & Facts (Star Schema)
  - Historical Analysis
  - Business rules
  - Documentation
  - Aggregated data
  - Logical table names

INSPARK

Metadata-driven Framework

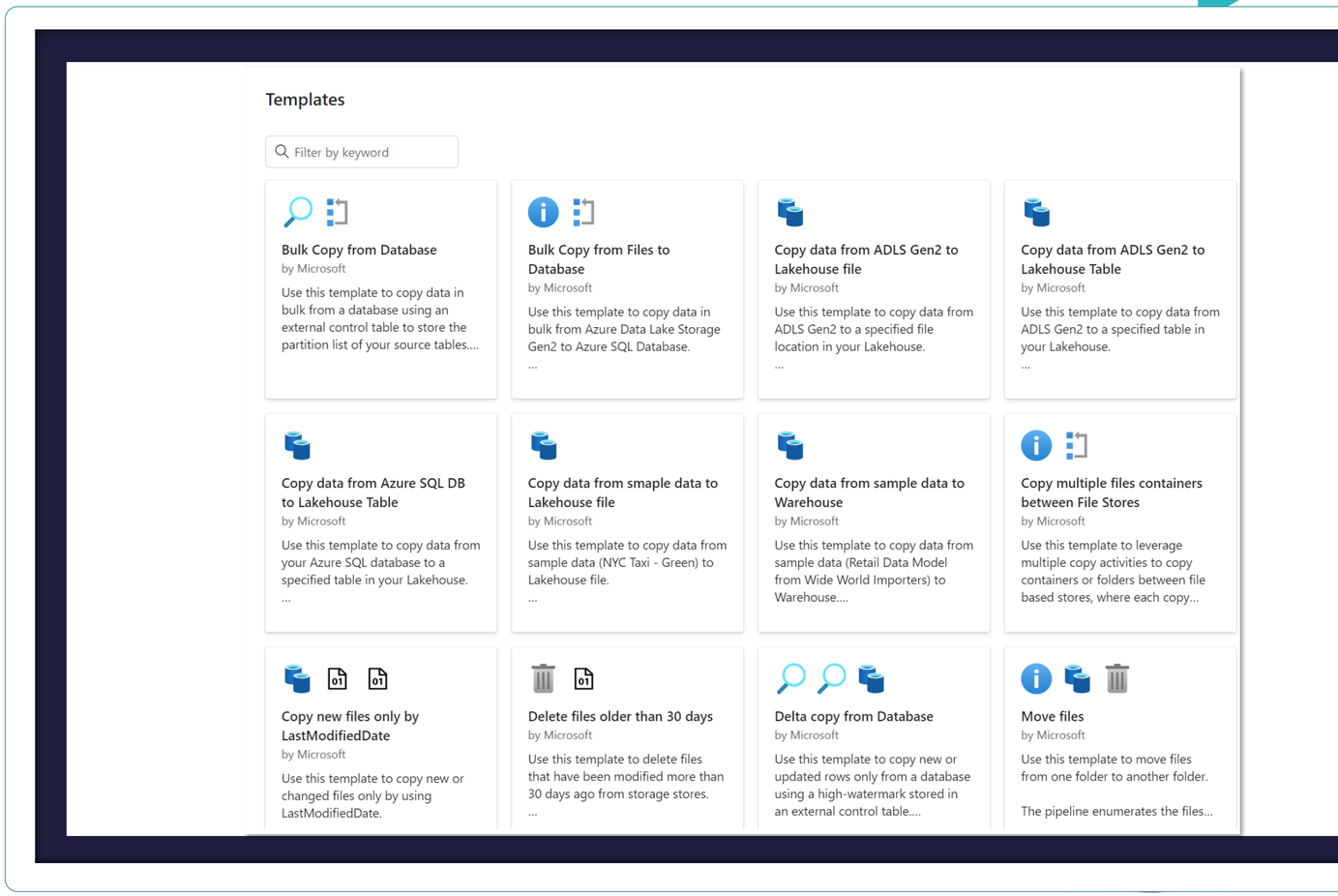
Understand Parameters



Metadata-driven Framework

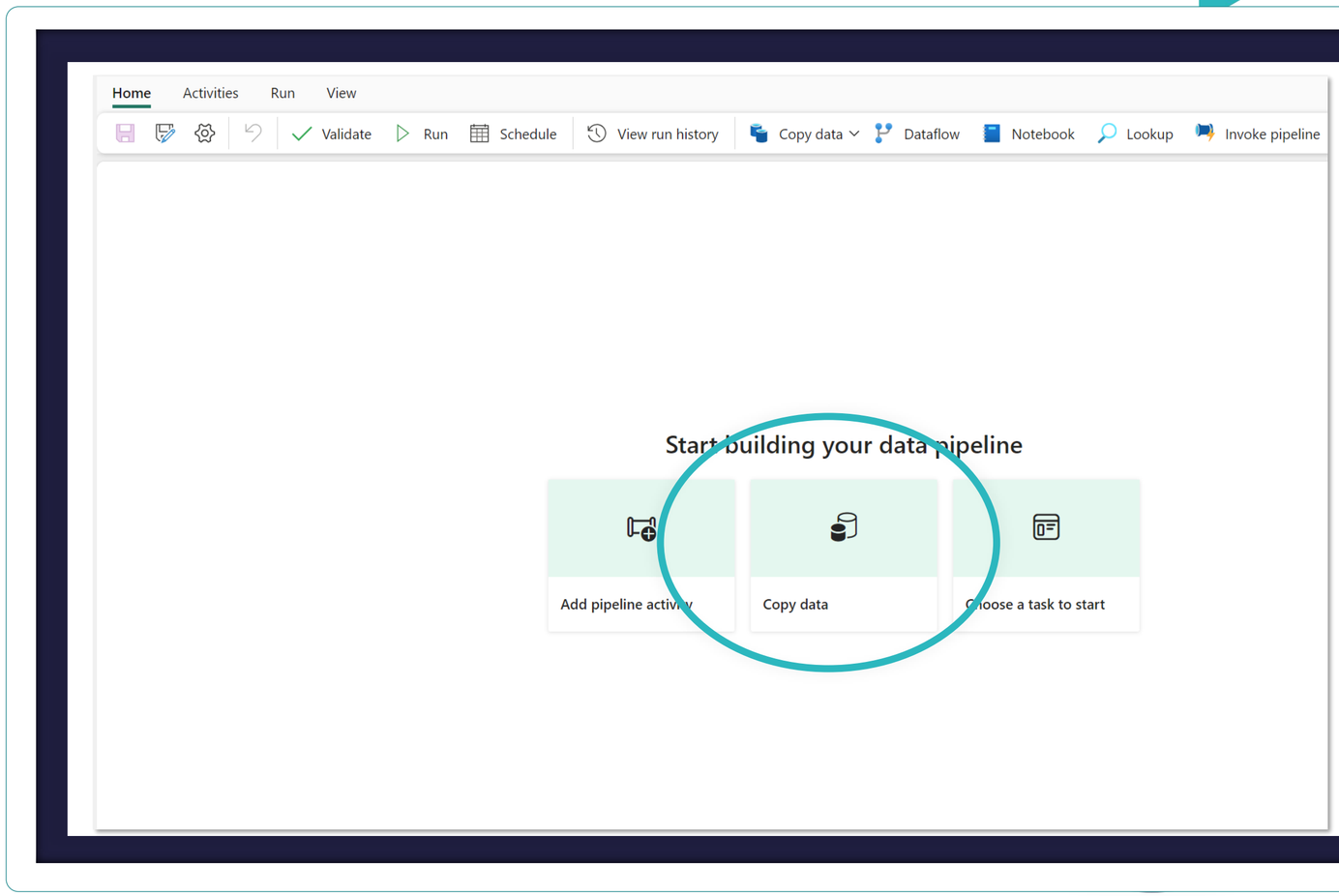
Templates

- 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



Metadata-driven Framework

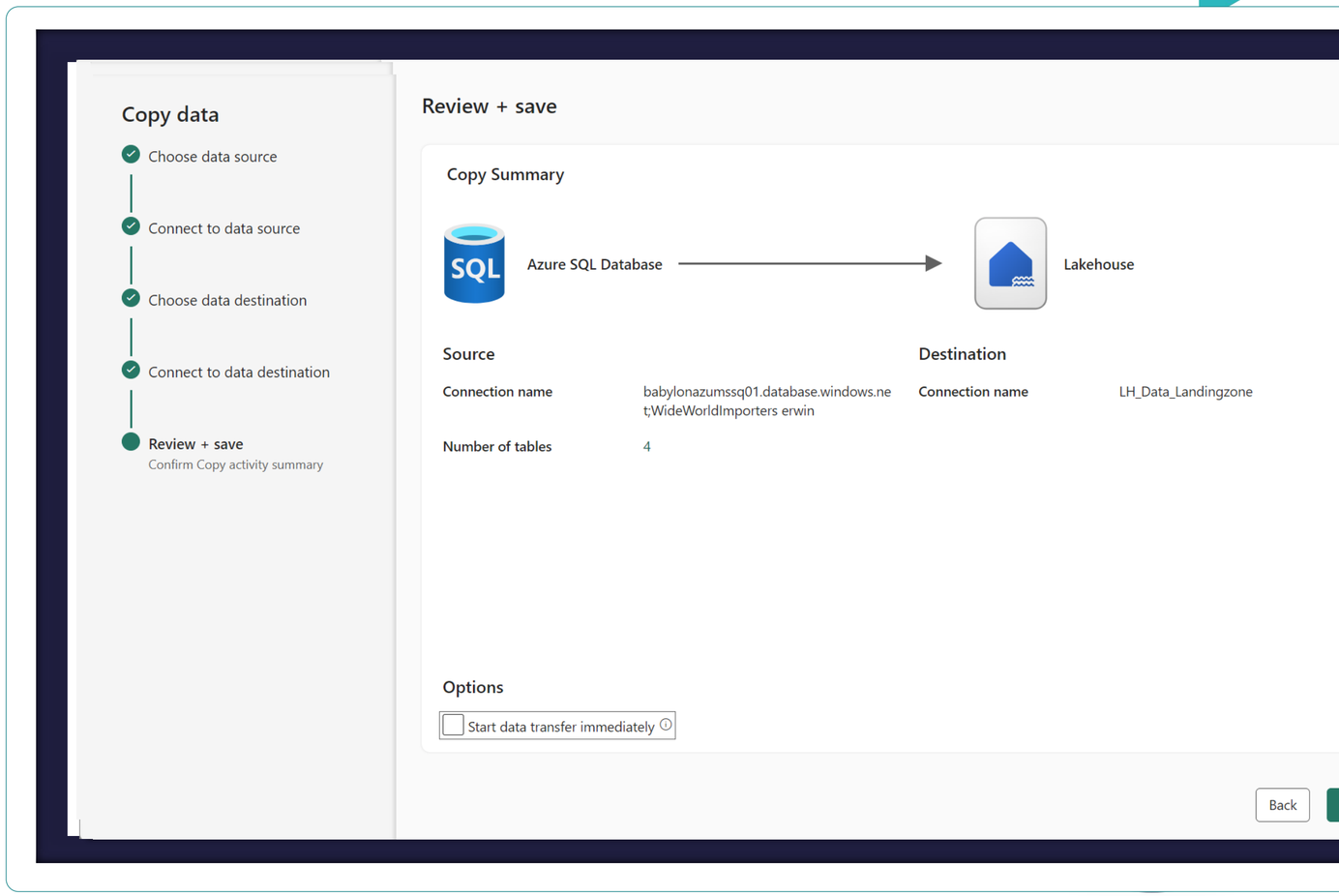
Understand Parameters



Metadata-driven Framework

Copy Assistant

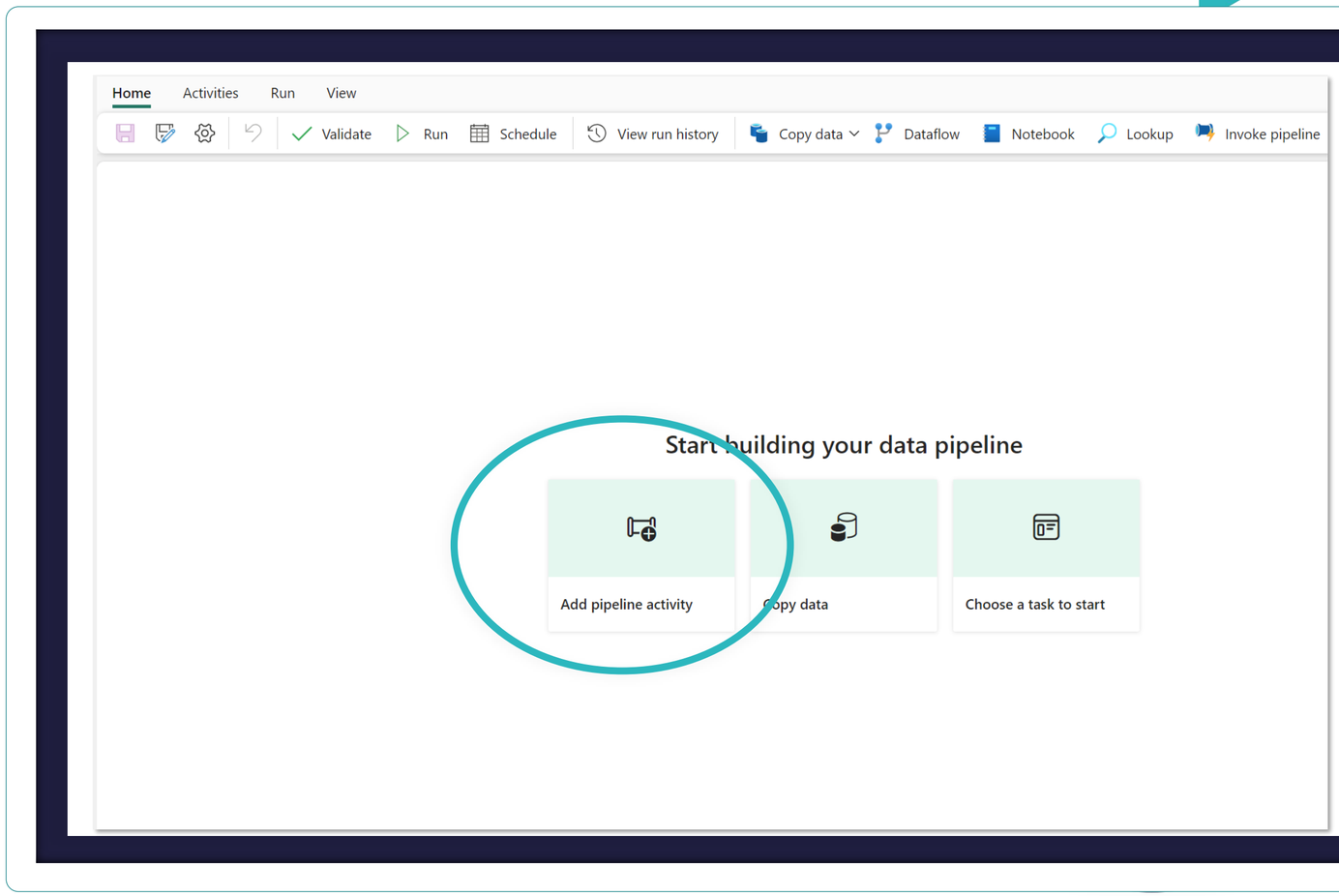
- Select Data Source
- Select Tables
- Select Destination
- Select Lakehouse
- Select Filetypes
- Data is stored as Delta Parquet

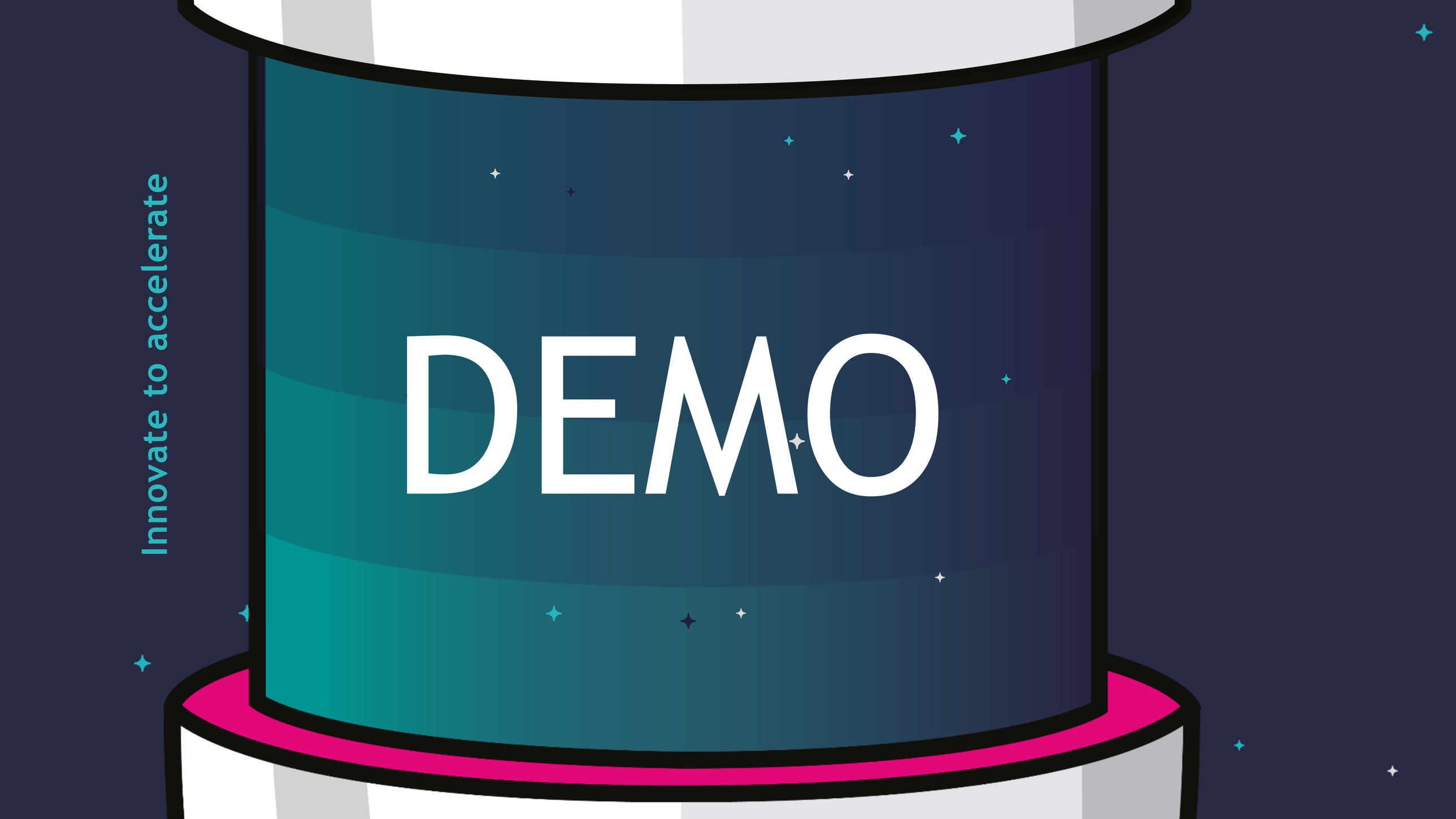


Metadata-driven Framework

Understand Parameters

- Or just start from blank canvas




Innovate to accelerate
DEMO

Metadata-driven Framework

Custom-Made Framework

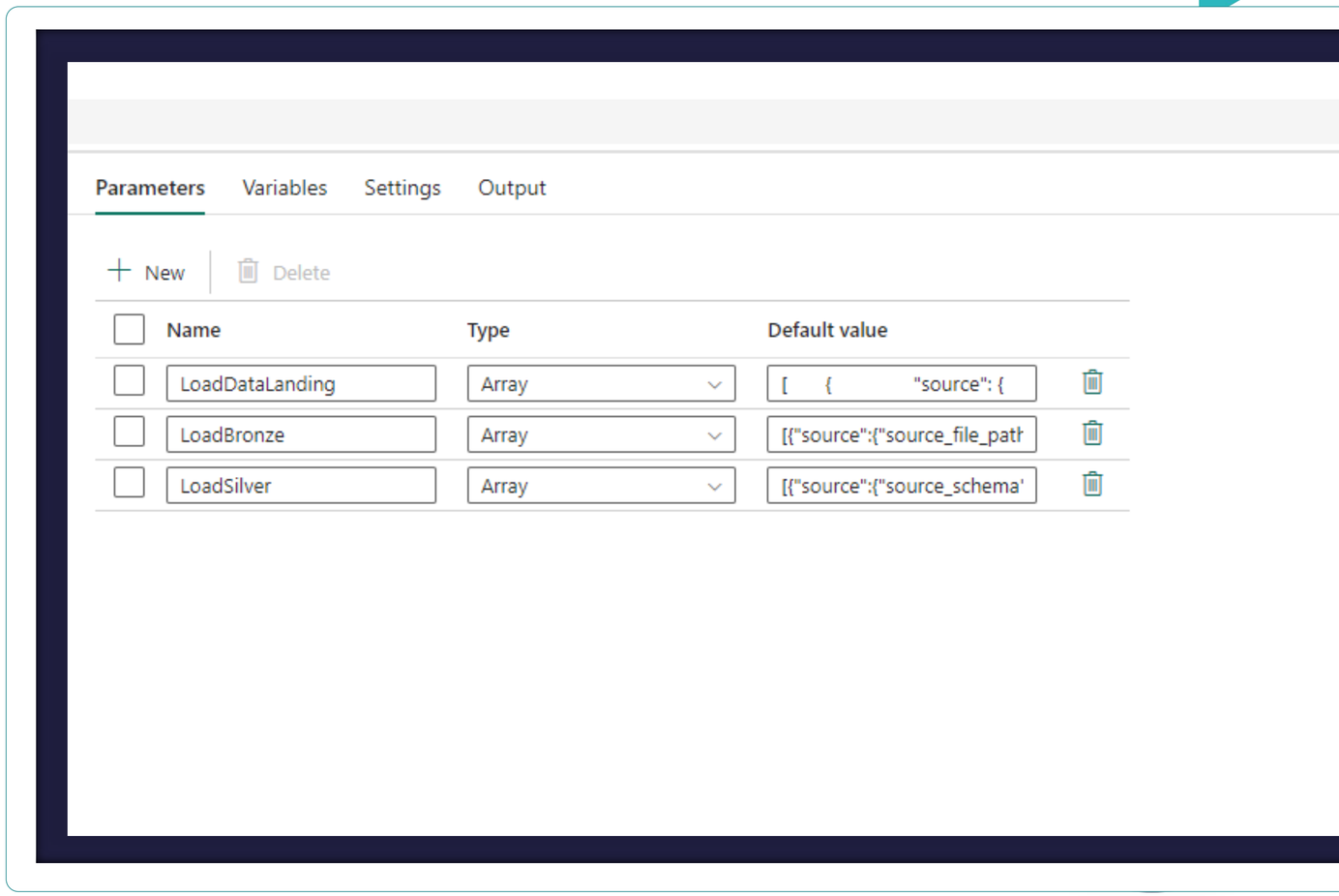
- Pipeline Parameters
- Notebook Parameters

INSPARK

Metadata-driven Framework

Pipeline Parameters

- Define Parameters
  - Pass through from Pipeline to Pipeline
  - Define Metadata
  - Versions



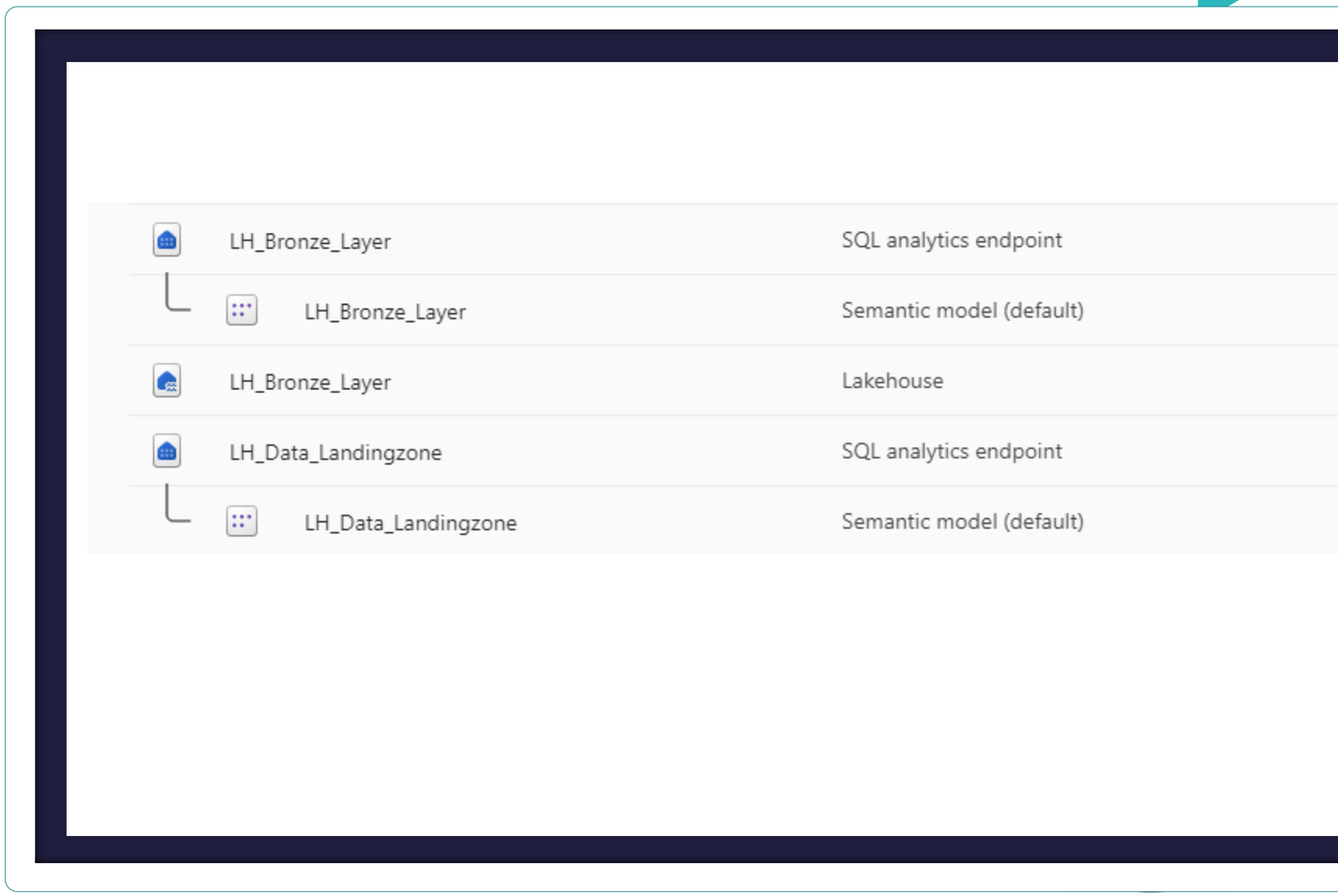
INSPARK

Metadata-driven Framework

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.

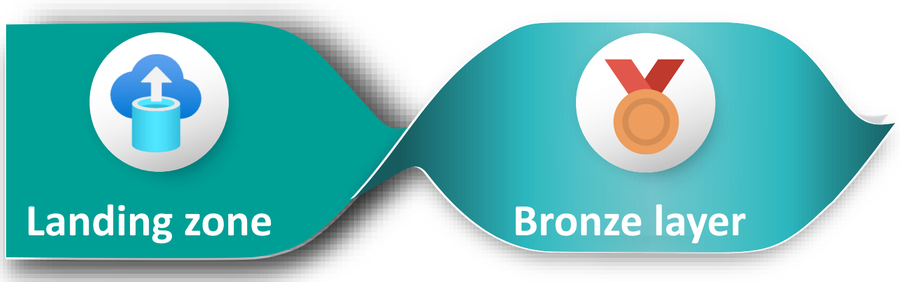
- **Lakehouses use Spark and SQL engines** to process large-scale data and support machine learning or predictive modeling analytics.
- **Lakehouse data is organized in a *schema-on-read format***, which means you define the schema as needed rather than having a predefined schema.
- **Lakehouses support 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.



Medaillon Architecture

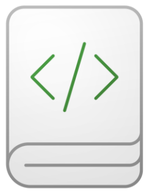
‘Data processing in different stages’

Stages



File types

Files and tables



INSPARK

Metadata-driven Framework

Notebook Parameters

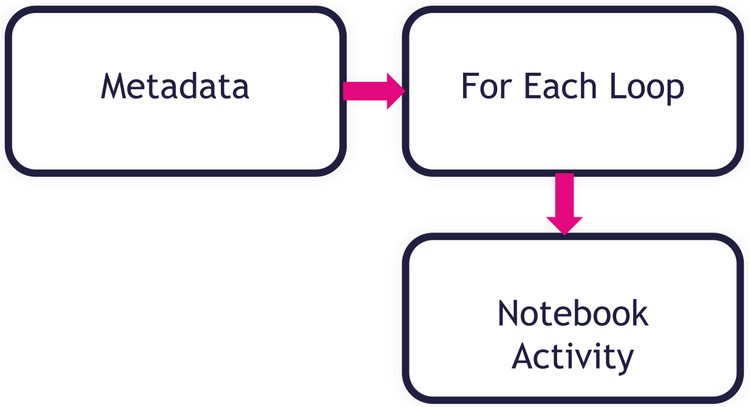
- Pass Parameters from Data Pipeline to Notebook
  - Toggle parameter cell

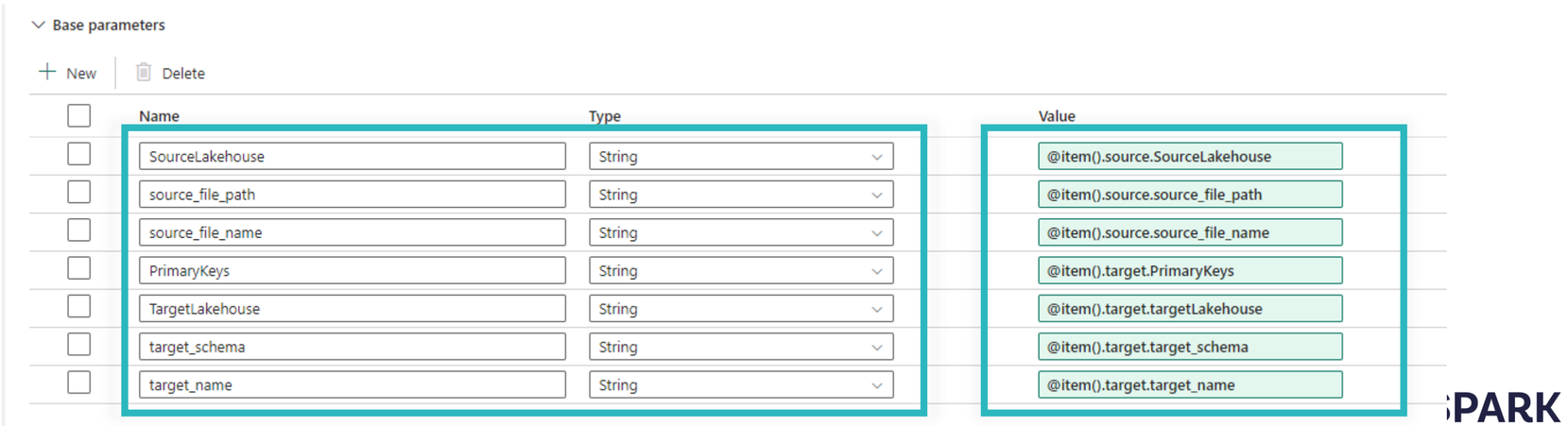
Metadata-driven Framework

Notebook Parameters

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

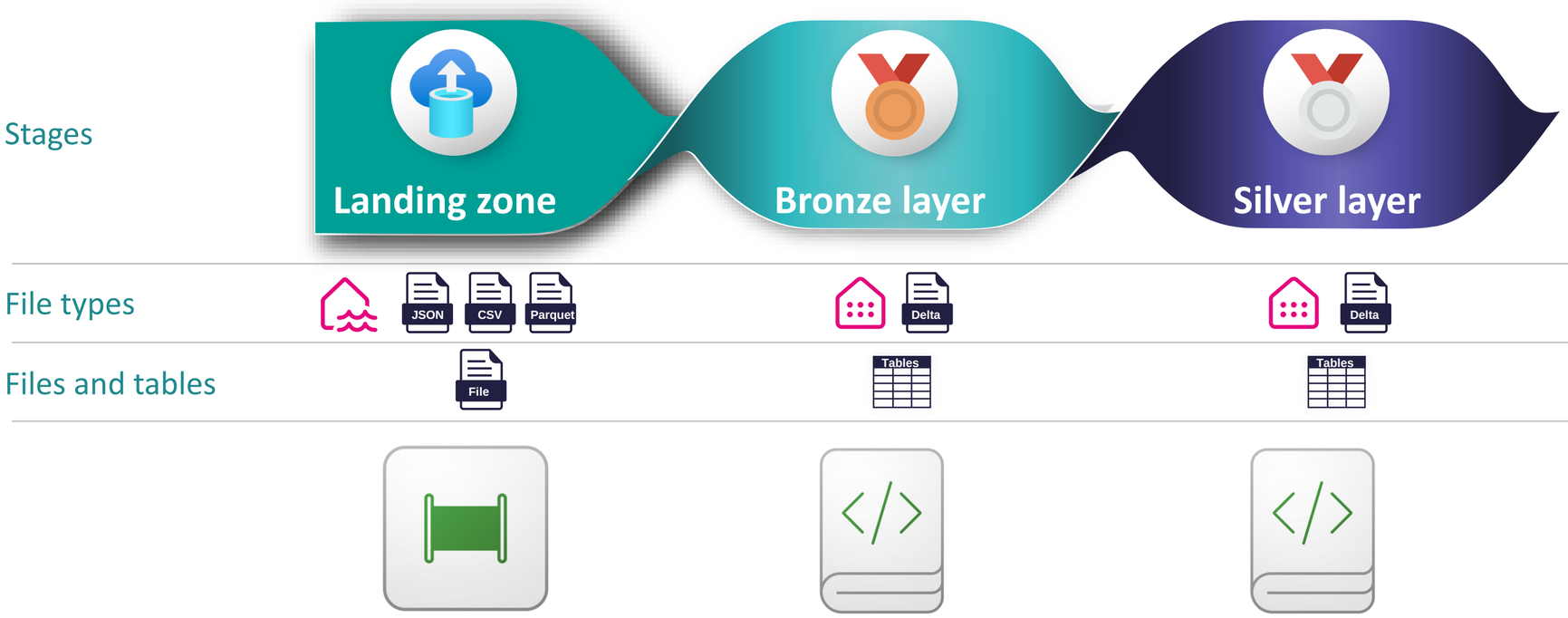
```
{
    "source": {
        "source_file_path": "WideWorldImporters",
        "source_file_name": "ApplicationPeople.parquet",
        "SourceLakehouse": "xxxxxxxxxxxxxxxxxxx"
    },
    "target": {
        "target_schema": "Application",
        "target_name" : "People",
        "PrimaryKeys" : "PersonID",
        "targetLakehouse" : "xxxxxxxxxxxxxxxxxxx"
    }
},
```



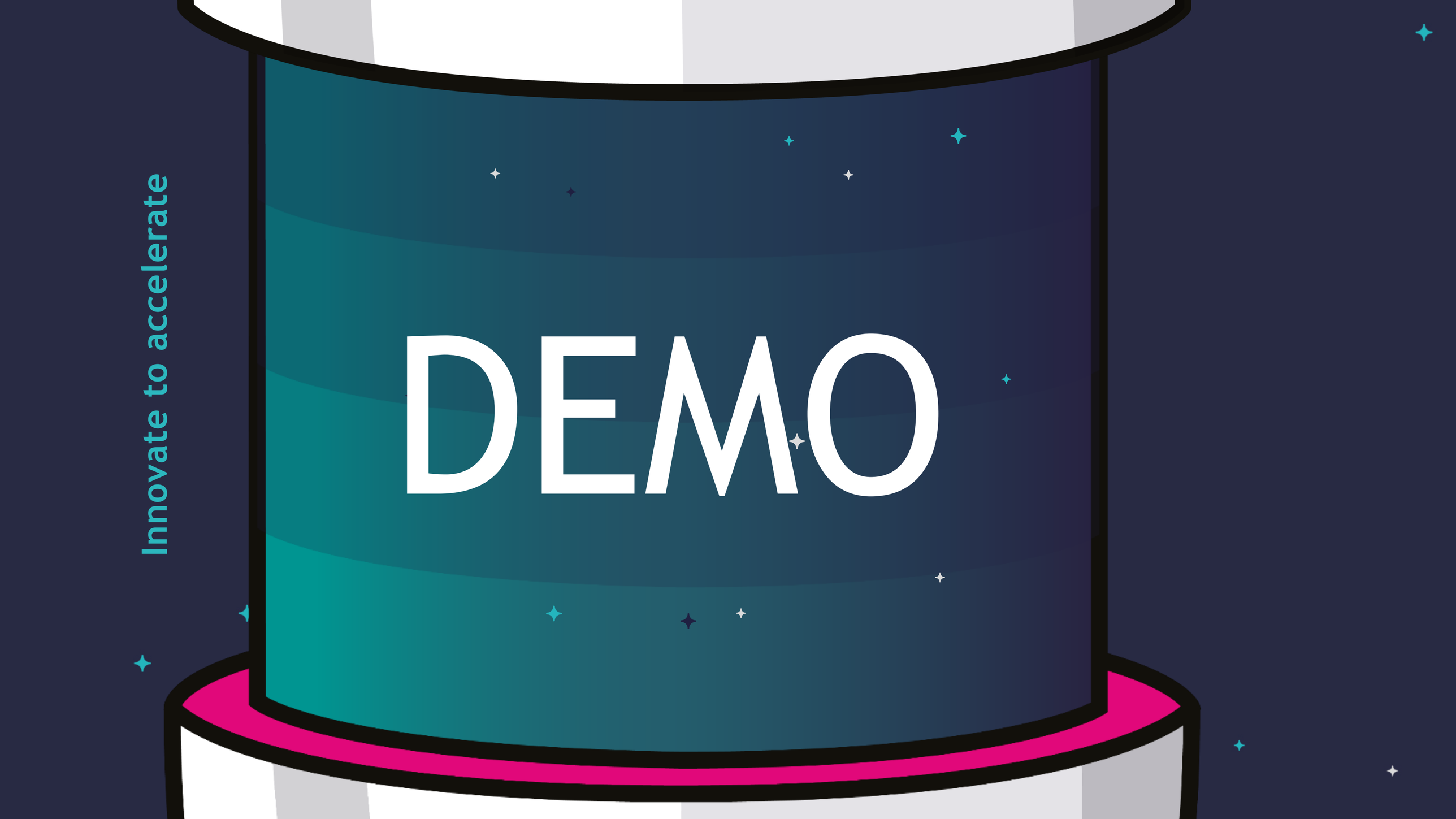


Medaillon Architecture

‘Data processing in different stages’



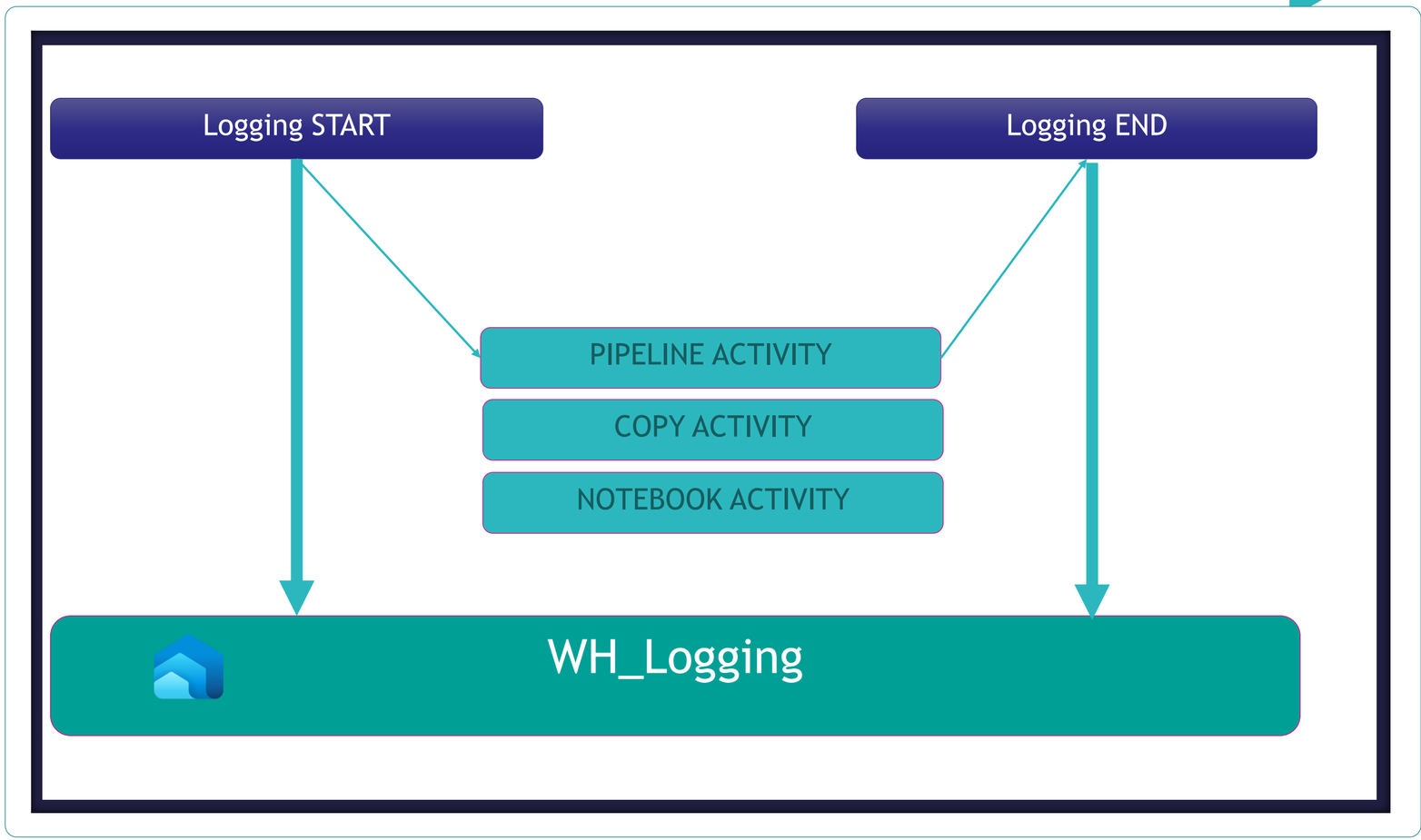
INSPARK


Innovate to accelerate
DEMO

Custom-Made Framework

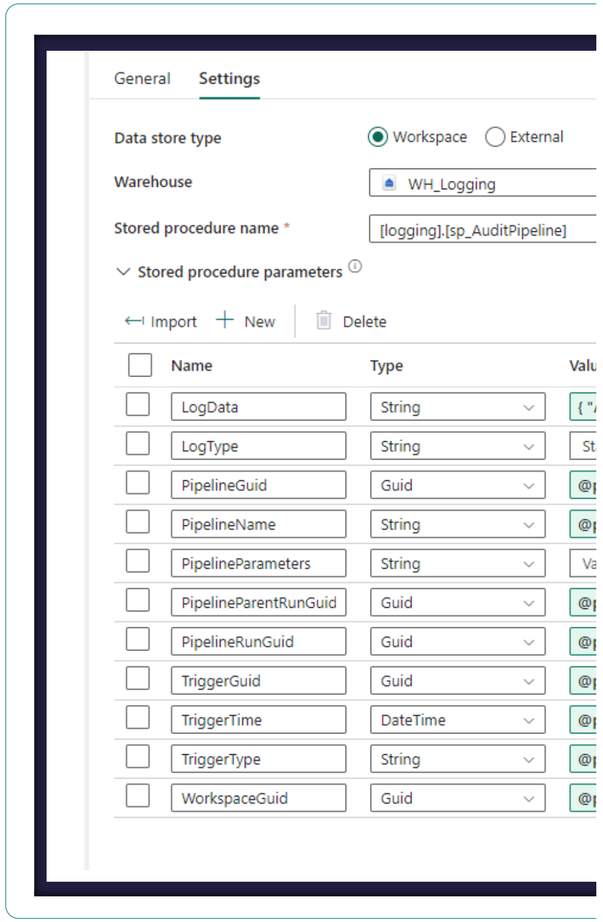
Logging

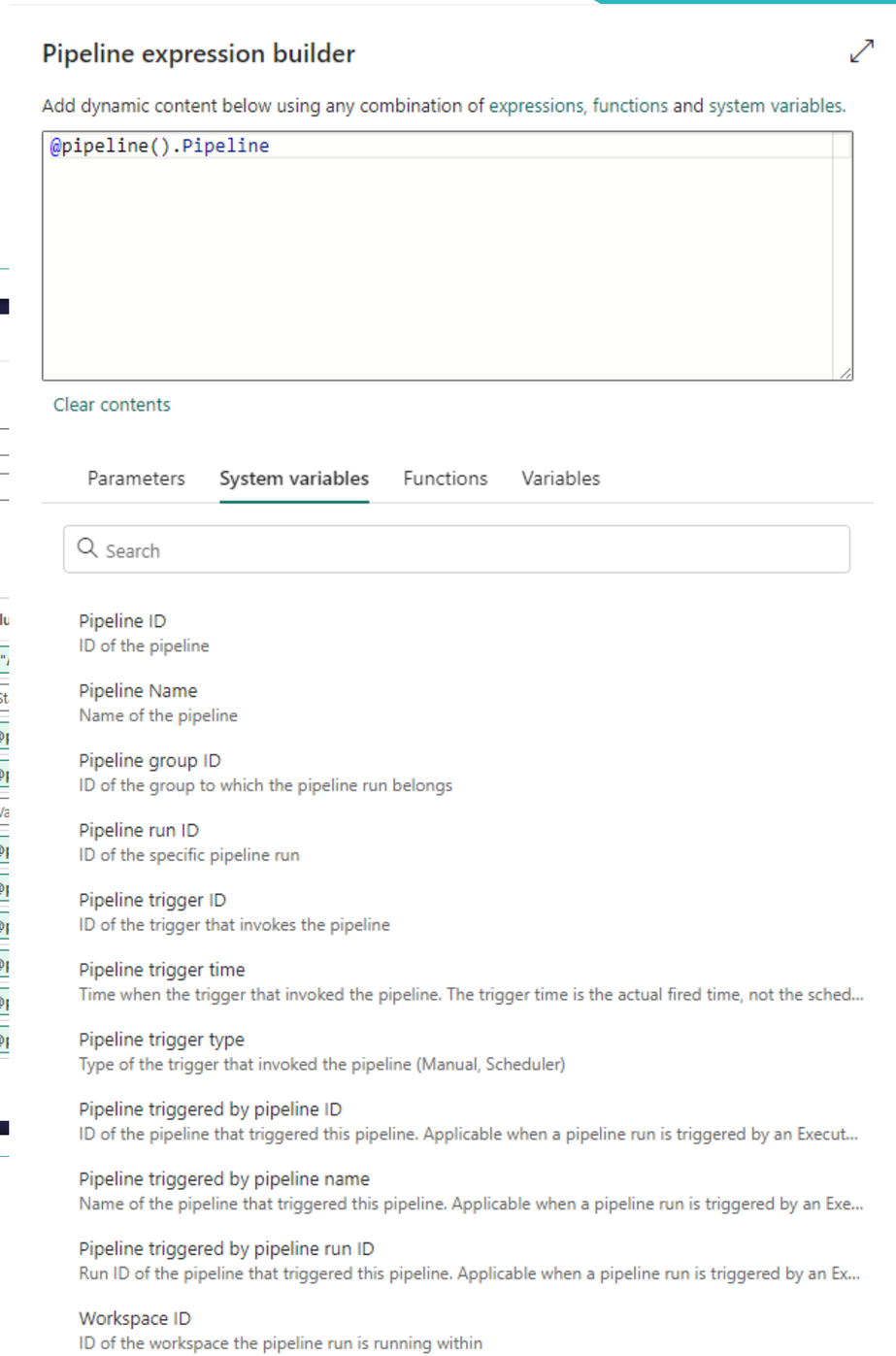
- Log Start and End Time of records
- Log Extracted Records
- Log Execution Failure



Logging

- Add Information about pipelines
- Adding System Variables

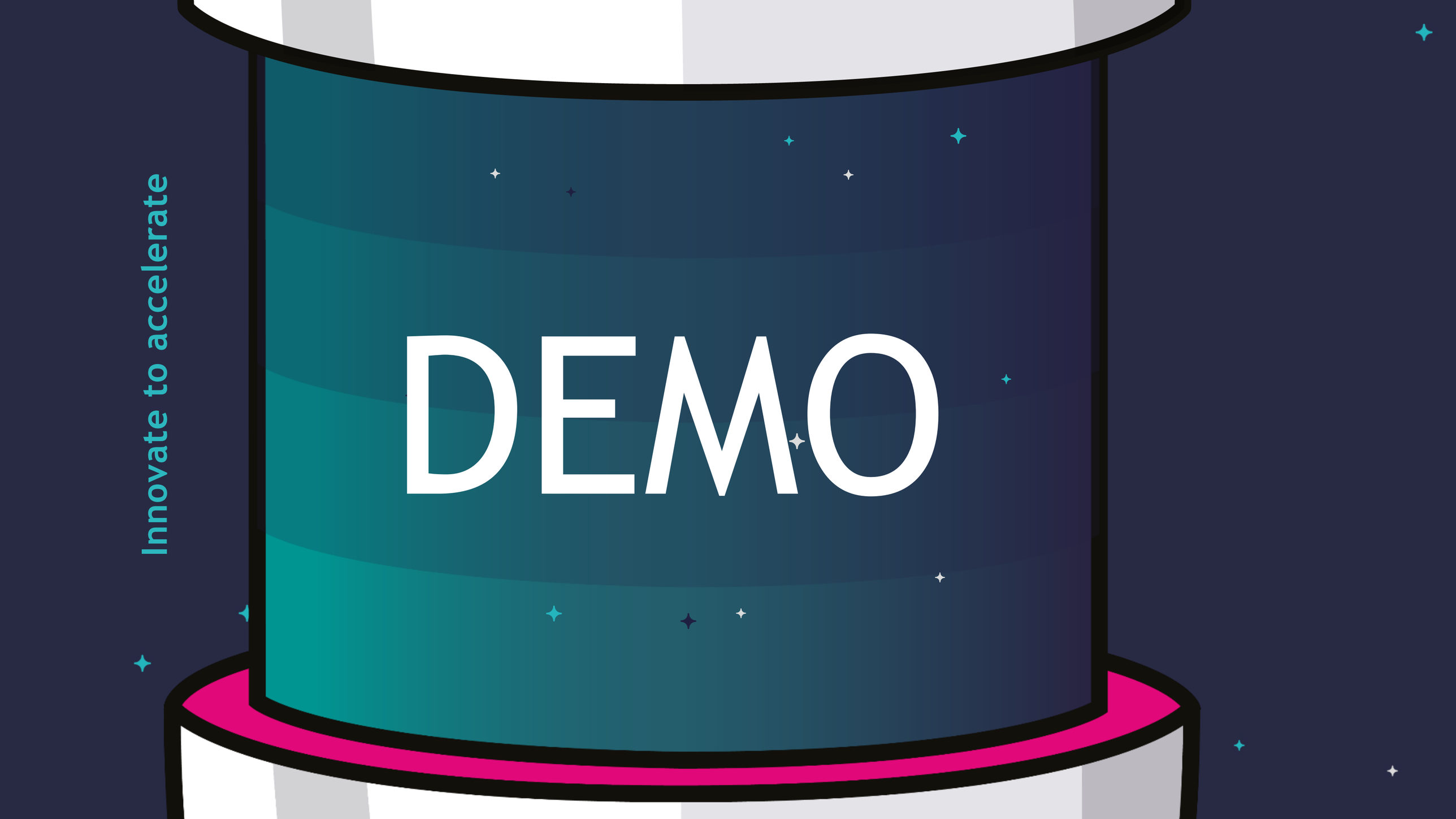




Logging

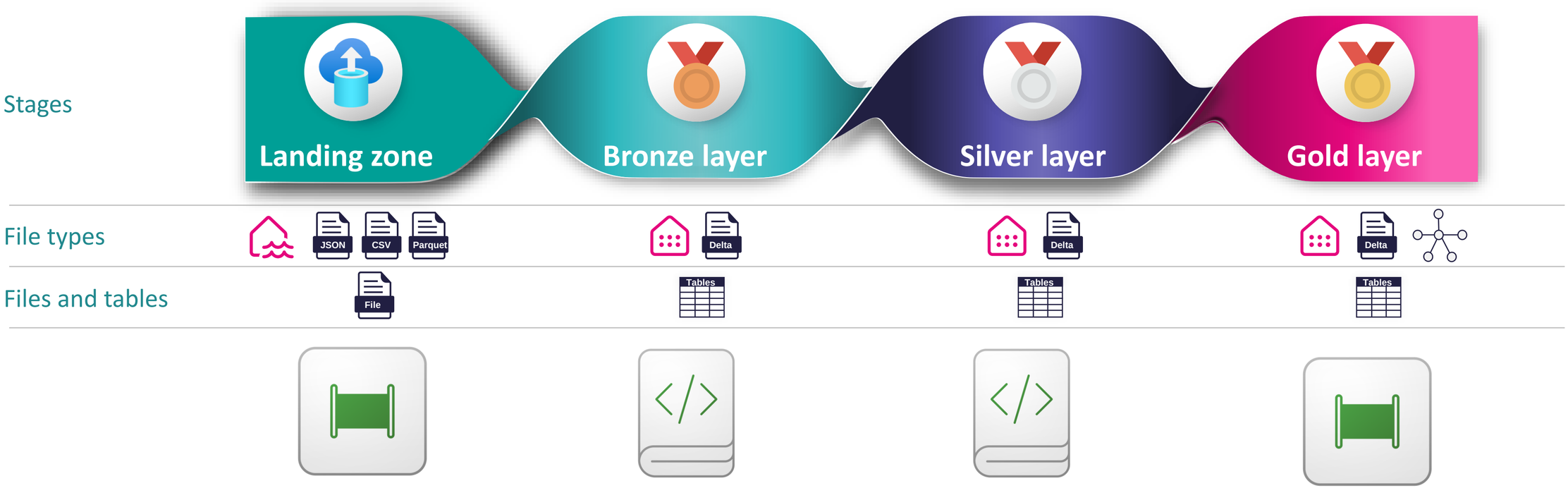
- Add Information about pipelines
- Adding System Variables
- Add Information about Notebooks


```
{
        "status": "Succeeded",
        "result": {
                "runId": "f0c69e6c-e28e-4db4-a8af-ac2892780xx2",
                "runStatus": "Succeeded",
                "sessionId": "fc55xx38-6fd8-47b0-863b-65c8c8db9878",
                "sparkPool": "3xx61f99-edc7-4d6c-a866-f3bf70bc7235",
                "error": null,
                "lastCheckedOn": "2024-01-23T15:44:25.2733333Z",
                "metadata": null,
                "exitValue": "{'CopyOutput': {'Total Runtime': '0:00:30.493626', 'TargetSchema':
'Application', 'TargetName': 'People'}}"
        },
        "message": "Notebook execution is in Succeeded state, runId: f0c69e6c-e28e-4db4-a8af-
ac2892780442",
        "SparkMonitoringURL": "https://app.powerbi.com/workloads/de-ds/sparkmonitor/fecxxff4-
3a26-495b-9ac9-475bbc59fae7/fc55be38-6fd8-47b0-863b-65c8c8db9878?trident=1&experience=power-
bi&tab=data",
        "executionDuration": 49
}
```


Innovate to accelerate
DEMO

Medaillon Architecture

'Data processing in different stages'



Recap

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

INSPARK

Custom-Made Framework

Recap

- 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
- Views can’t not be called from a Notebook (Silver to Gold)
- Schedule can’t be Parameterized like in ADF/Synapse
- Build in retry to Notebook Activity

INSPARK

Questions?

INSPARK

THANK YOU

Data Community Austria

Platinum

smart casual datadesign

redgate

TIMEXTENDER

Gold

b.telligent
smart data. smart decisions.

Lucient

Measure Killer

Spot by NetApp

Silver

veeam

Bronze

ACP CUBIDO Digital Solutions

Power BI Camp
powered by linearis.at

Savory DATA

SOLISYON

Thank you

Evaluations, evaluations...

https://evals.datagrillen.com/evals_vienna.aspx

Erwin de Kreuk
Principal Consultant – Lead Data & Analytics
InSpark

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

Let's connect

Microsoft® Most Valuable Professional