



Build Your AI Foundation with Fabric Metadata-driven Framework

Erwin de Kreuk

Technology Lead Data, InSpark, Netherlands



 @erwindekreuk.bsky.social

 [linkedin.com/in/erwindekreuk](https://www.linkedin.com/in/erwindekreuk)

 erwindekreuk.com

 github.com/edkreuk

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

 [Dutchfabricusergroup.com](https://dutchfabricusergroup.com)



FABCON | **SQLCON**

ATLANTA26

JOIN THE
CONVERSATION

#FABCONSQLCON26

AI Reality Check: Why Your Results Lag Behind

🚨 If you're still saying "AI wasn't that good when I tried it..."

This is your wake-up call.

⚡ AI Has Accelerated Exponentially

- What you tested last year... last month... even last week..., is already outdated.

💡 The Real Issue Isn't AI ,It's the Data Foundation

AI doesn't fail because the model is weak.

AI fails because the data platform is.



Still Saying "But I Tried AI and it wasn't that good..."

THIS IS YOUR WAKE-UP CALL!



Legacy Data Spaghetti

Unified Data Lasagna

No Clean Data

No Clean Data

No Real-Time Signals

No Governance

No Unified Foundation

No Data Observability

- ✓ Clean Data
- ✓ Real-Time Insights
- ✓ Governance
- ✓ Unified Foundation
- ✓ Data Observability

You can't scale AI on Legacy Data Spaghetti.

Lasagna Beats Spaghetti for AI Success! ✓

- ✓ Clean Data
- ✓ Real-Time Insights
- ✓ Governance
- ✓ Unified Foundation



AI Reality Check: Why Your Results Lag Behind

- ✓ Modern AI Needs a Modern Data Foundation
Fabric-native. Unified. Governed. Real-time. AI-ready.



A vibrant illustration of a chef in a kitchen. The chef, wearing a white hat and a red scarf, stands on the right, smiling and holding a wooden spoon. The kitchen counter is filled with various items: a laptop displaying 'AI' with a brain icon, a pot of food on a stove, a tray of colorful vegetables, and several beakers with colored liquids. In the background, a glowing brain is surrounded by data charts and graphs, suggesting a connection between AI and culinary arts.

Let's Get Started!

Time to Cook Up Some Insights!

Fabric Data Platform



Developer



Developer



Developer

Data Sources

Data Integration

Data Processing

Data Serving

Data Intelligence

Data Visualization



Users



Users



Users

Data Observability

Data Governance

Data Platform Engineering

Data Security



Microsoft Fabric

From Data Chaos to AI-Ready Intelligence

Data Platform Engineering

Building the Kitchen

A strong platform is the prerequisite for everything that follows.

Data Integration - Preparing the Ingredients

Integration turns scattered data into consistent, usable inputs.

Data Processing - Cooking the Data

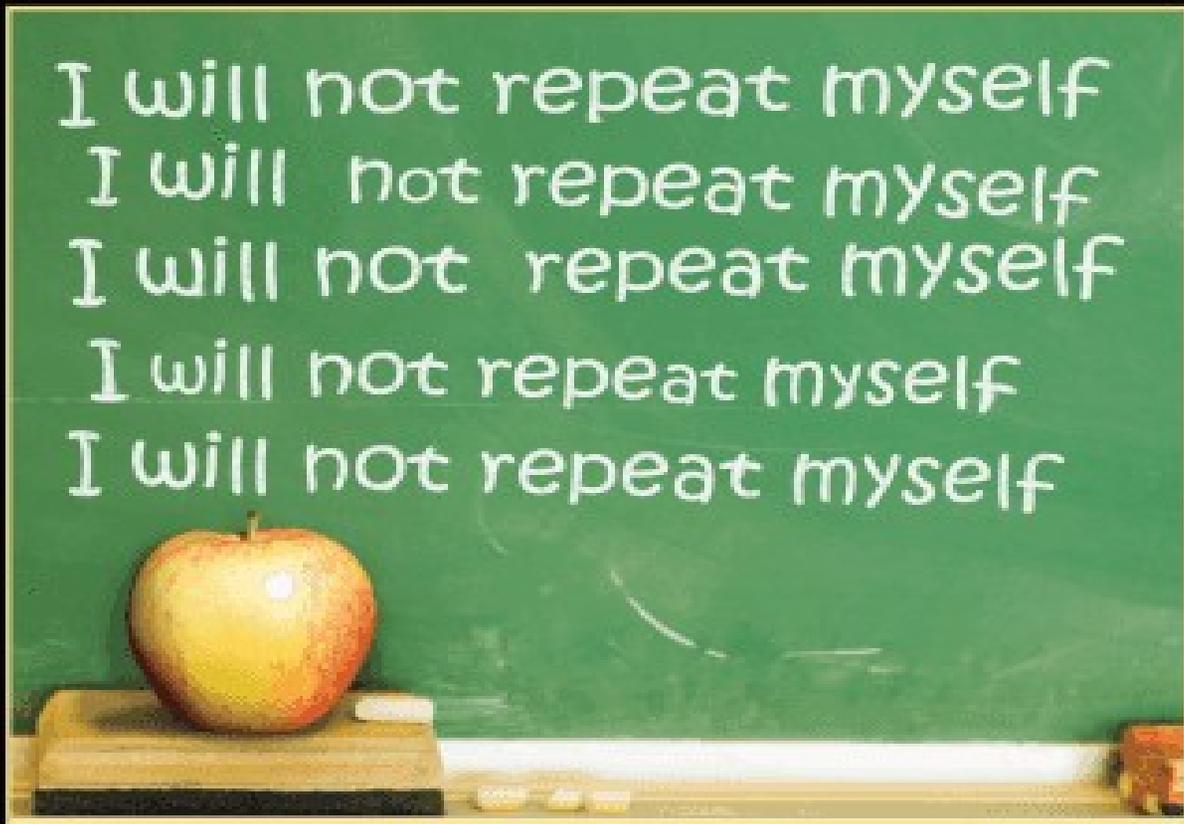
Processing is where raw data becomes real value.

Data Observability - Quality Control at Scale

From Spaghetti to Lasagna ✓

Metadata-Driven Frameworks





DON'T REPEAT YOURSELF

Repetition is the root of all software evil.

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
- Metadata => Fabric SQL Database / Json /
- Microsoft Fabric but also on Azure Synapse Analytics and Azure Data Factory
- Based on a Uniform Data Architecture



What about the Third option

- Fabric Native Solution





The unified data platform for AI transformation



Data
Factory



Analytics



Databases



Real-Time
Intelligence



IQ



Power BI

Fabric Platform



Copilot



OneLake



Governance



The unified data platform for AI transformation



Data
Factory



Analytics



Databases



Real-Time
Intelligence



IQ



Power BI

Fabric Platform



Copilot



OneLake



Governance

Fabric Metadata Driven Framework



- Meets most of your needs
- Full control over design and features.
- Less development effort.
- Flexibility and extensibility.
- Lower upfront costs.
- 100 % Fabric Native Solution

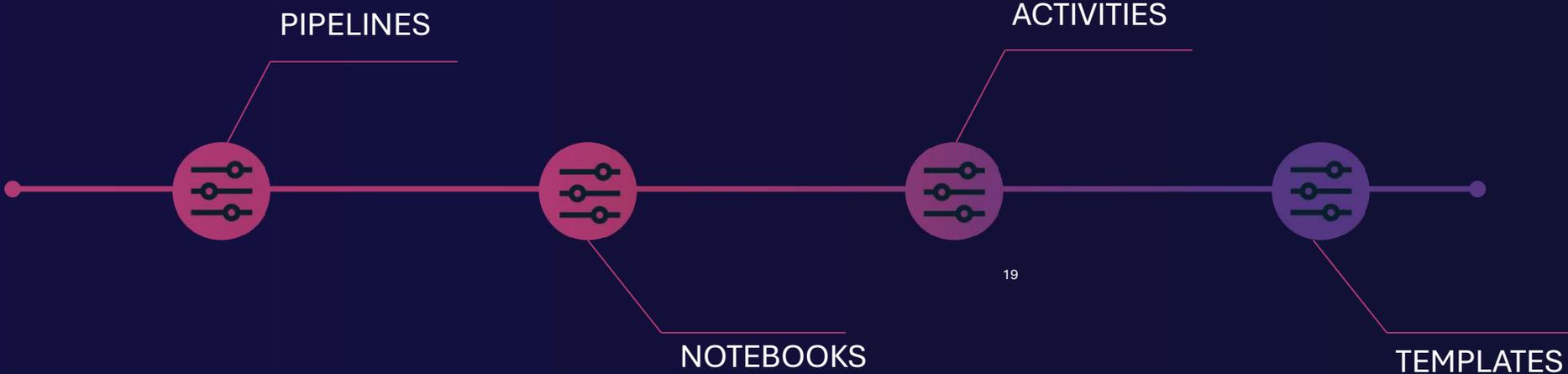
Fabric Metadata Driven Framework



Modern data platforms demand agility, scalability, and consistency. FMD simplifies these challenges by enabling:

- Dynamic, metadata-driven pipelines
- Consistent orchestration across ingestion, processing, and publishing
- Centralized configuration for all data entities
- Alignment with Microsoft Fabric Lakehouse & Medallion Architecture
- Reduced engineering effort through reusable patterns
- Faster delivery with standardized, tested components

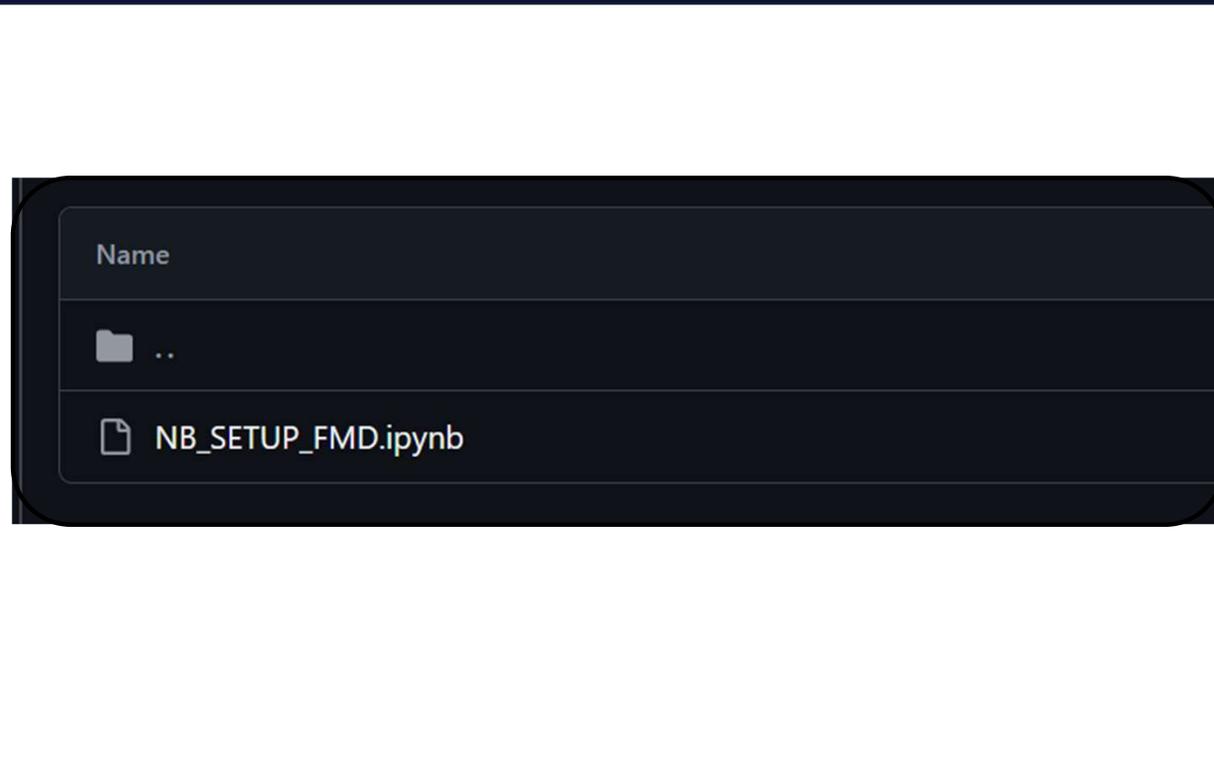
Parameters



Fabric Metadata Driven Framework



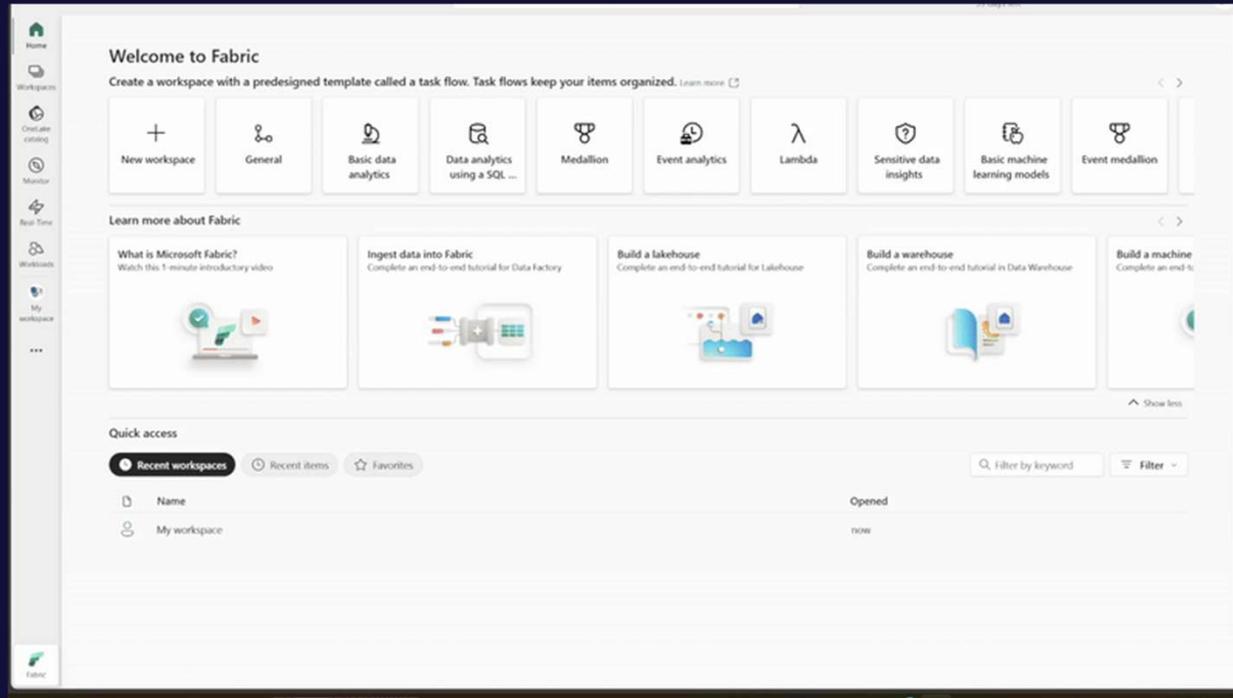
- Download Notebook from Github



Fabric Metadata Driven Framework



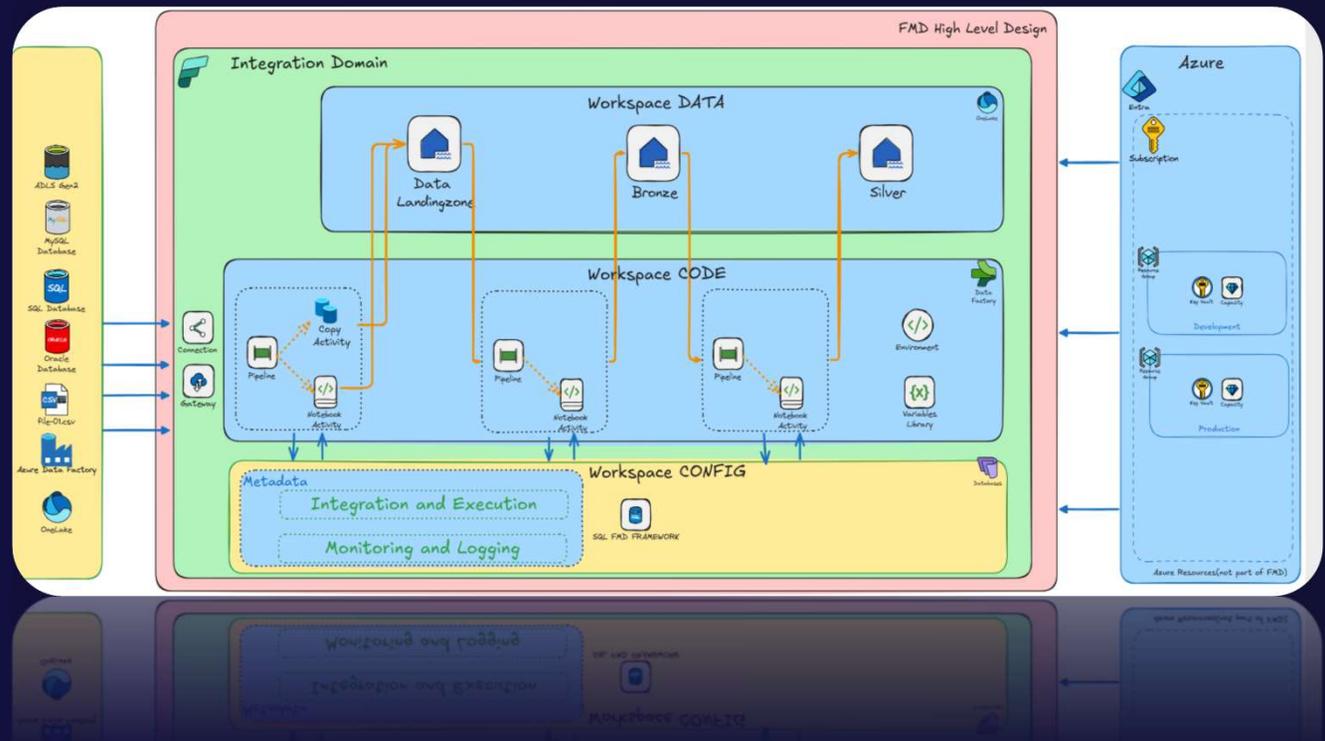
- Download Notebook from Github
- Create Workspace Name is up to you:
 - **FMD_XXX_CONFIGURATION**
- Assign a capacity
- Import Notebook to this Workspace



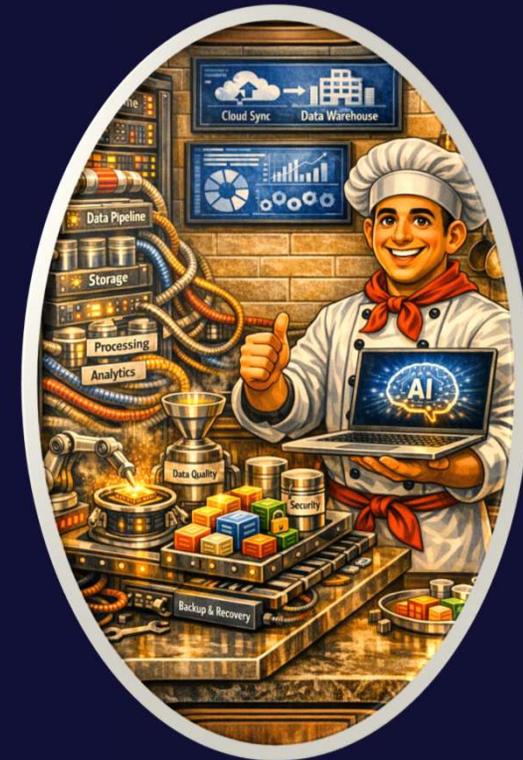
Fabric Metadata Driven Framework



- Open Notebook
- NB_SETUP_FMD.ipynb



*A strong platform enables speed,
reliability, and innovation at scale.*



Building the Kitchen

Data Platform Engineering

FABCON | SQLCON

ATLANTA26

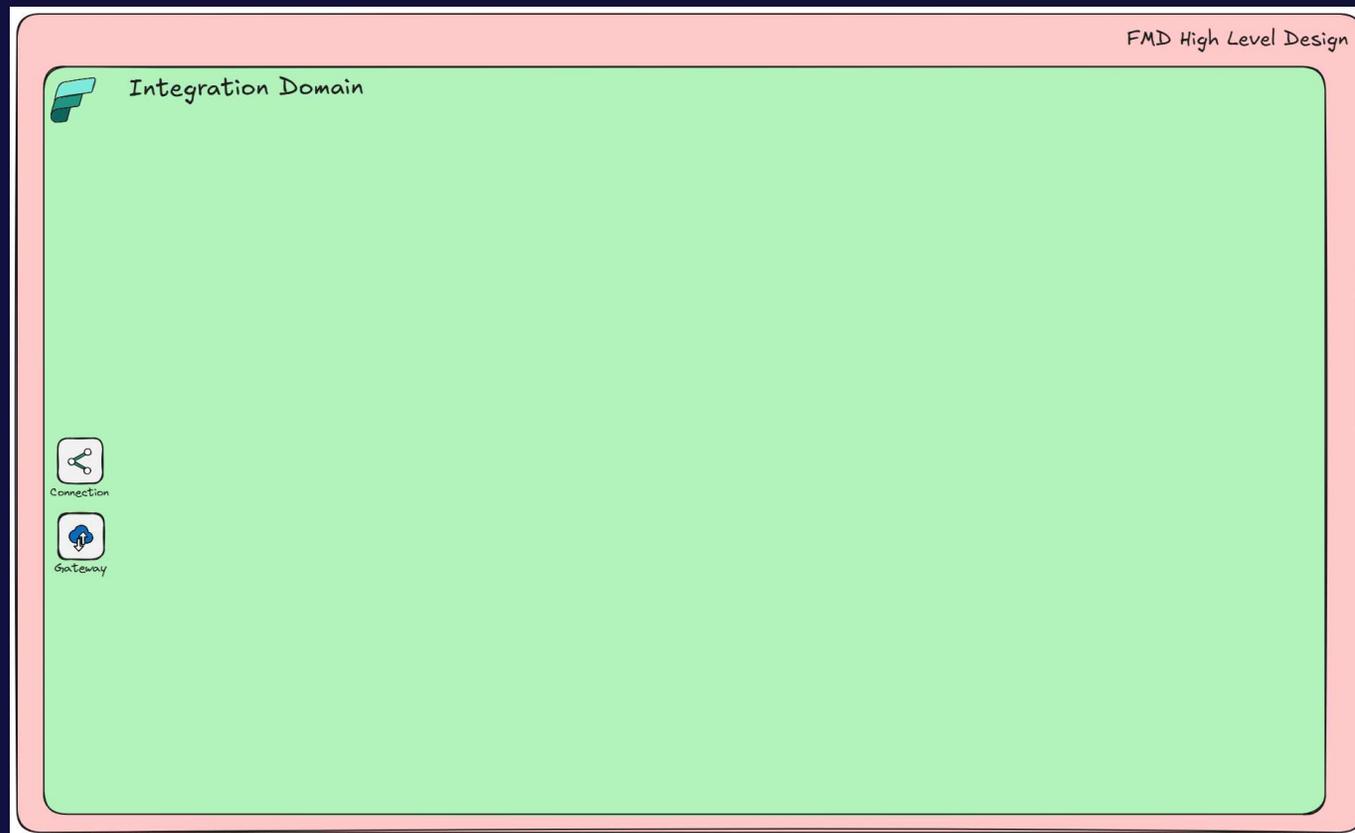
JOIN THE
CONVERSATION

#FABCONSQLCON26

Data Platform Engineering

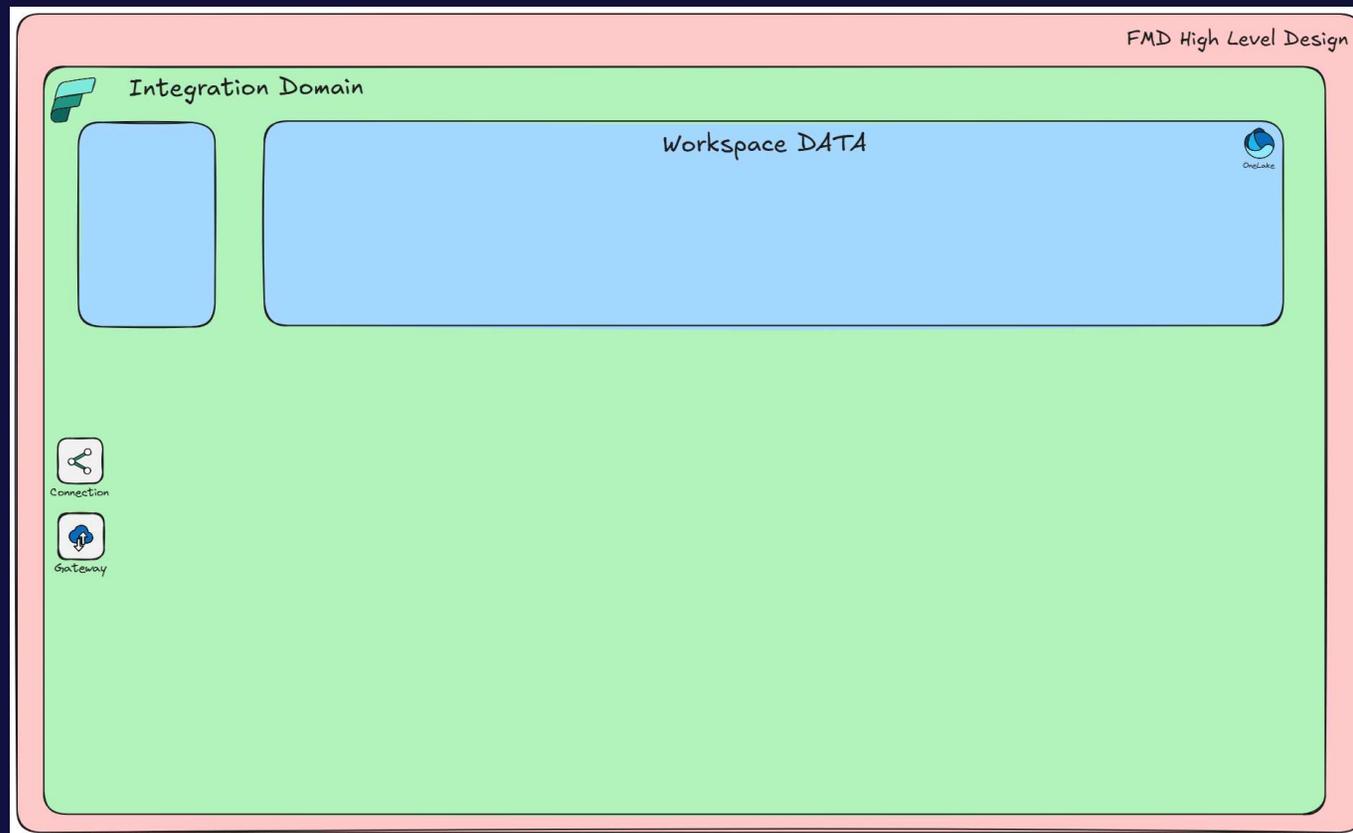


- Azure subscription
- Fabric Capacity (Trial)
- Key Vault optional
- Fabric Administrator role



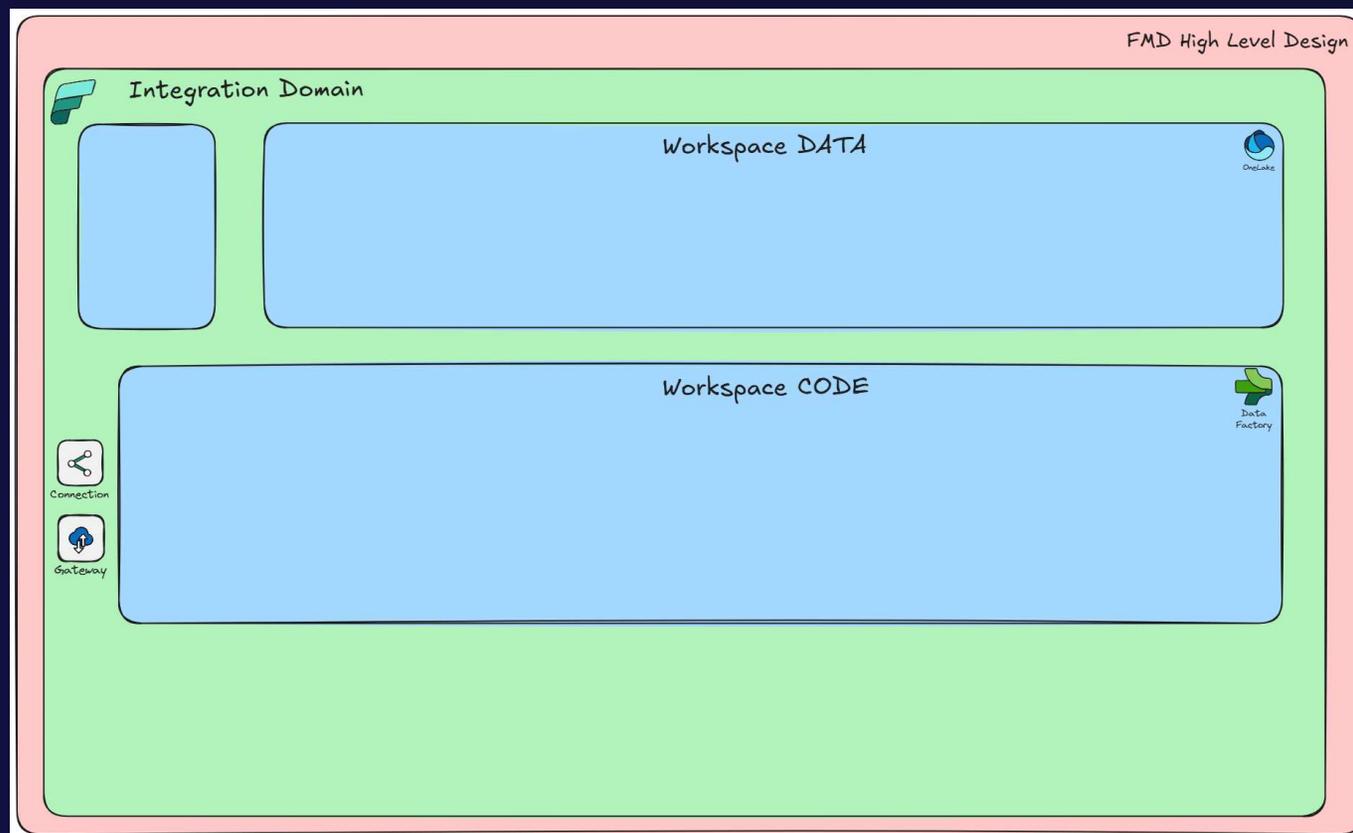
Data Platform Engineering

- Workspace DATA:
- Lakehouses



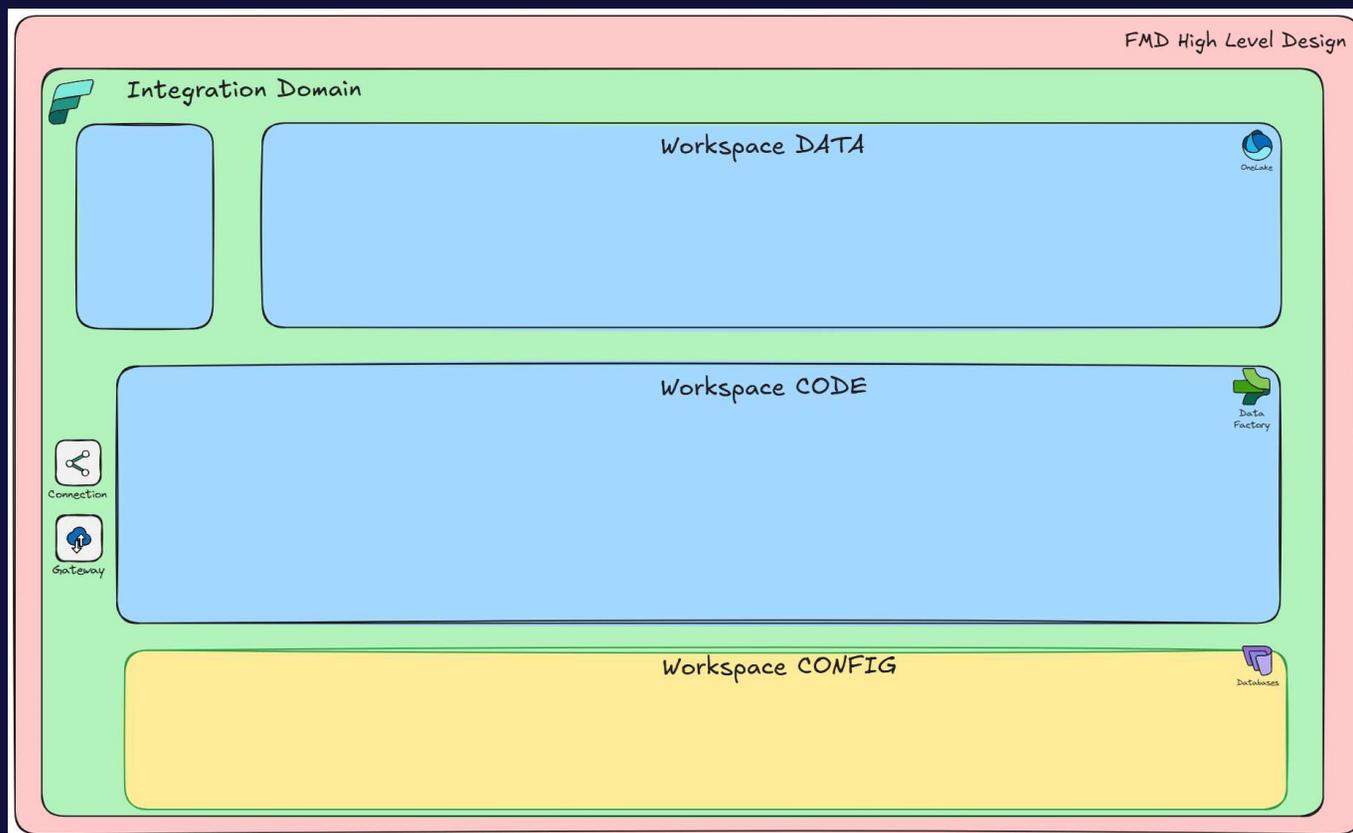
Data Platform Engineering

- Workspace CODE
- Data Pipelines
- Notebooks
- Variable Libraries



Data Platform Engineering

- Workspace CONFIG
- Fabric database
 - Metadata database
- Environments

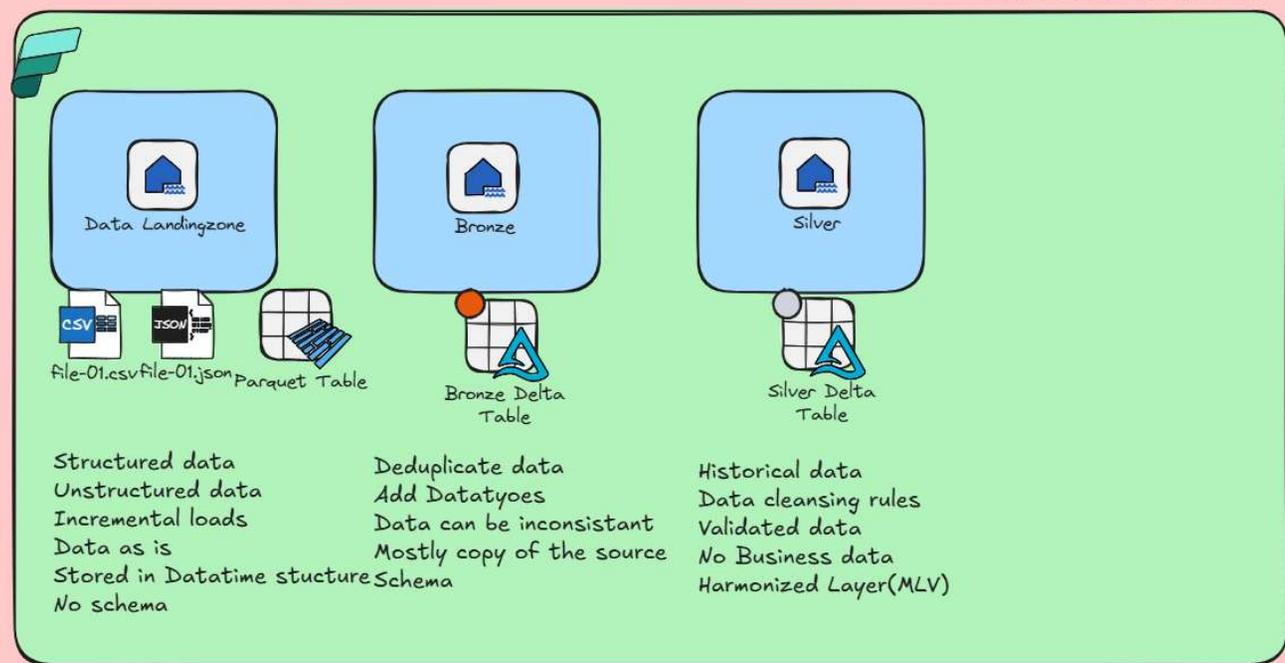


Item deployment

- Lakehouse
- Medallion Architecture
 - Data Landingzone
 - Bronze
 - Silver

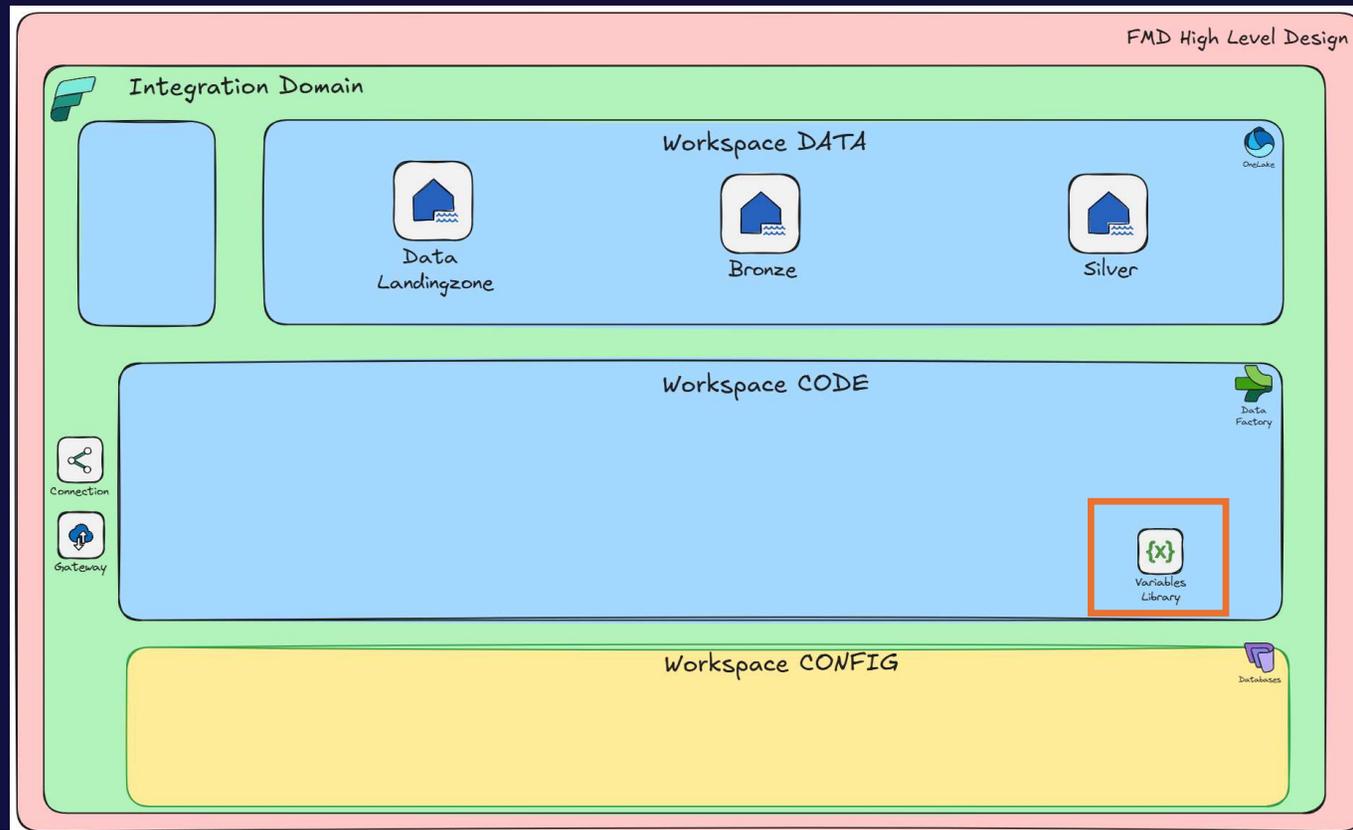
FMD High Level Design

FMD Lakehouse Overview



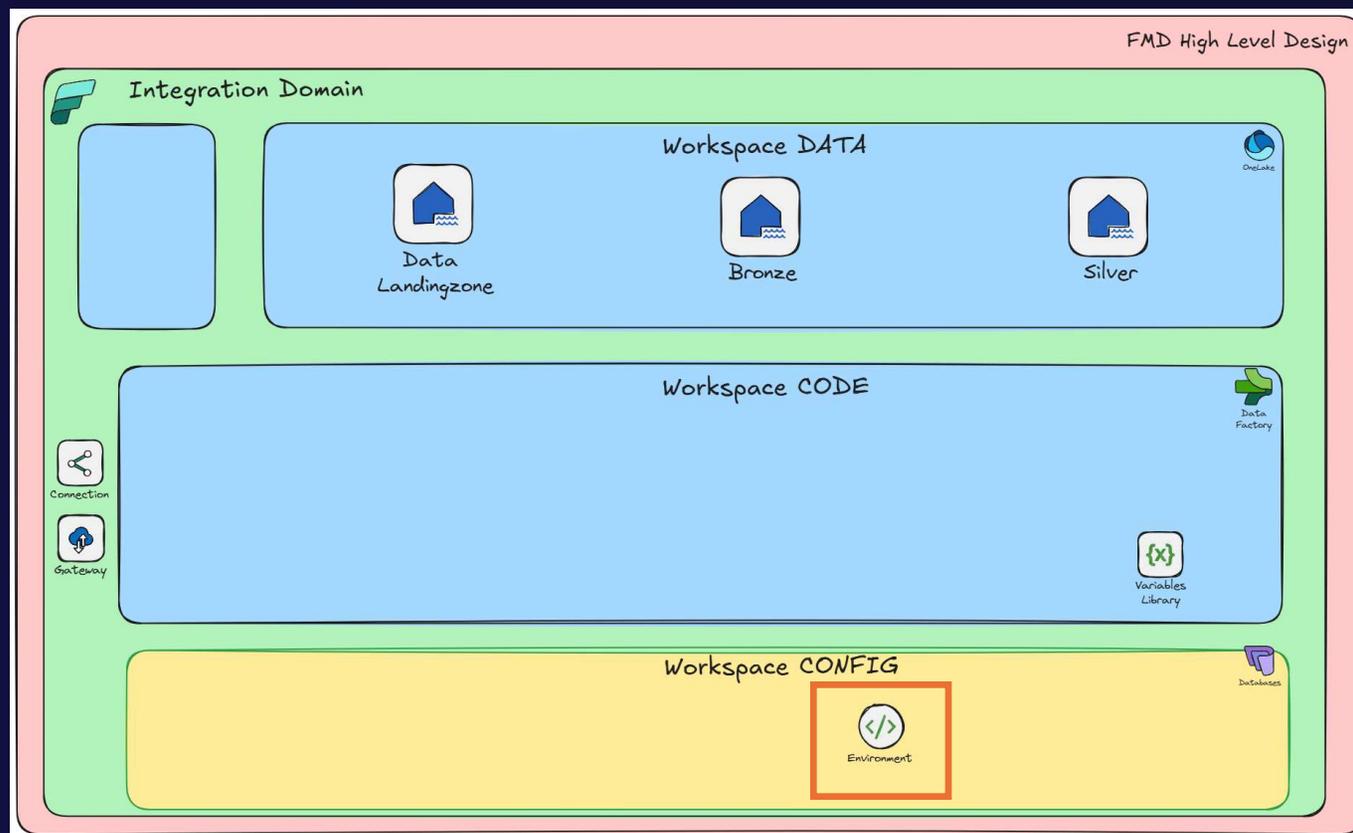
Item deployment

- Variable Library



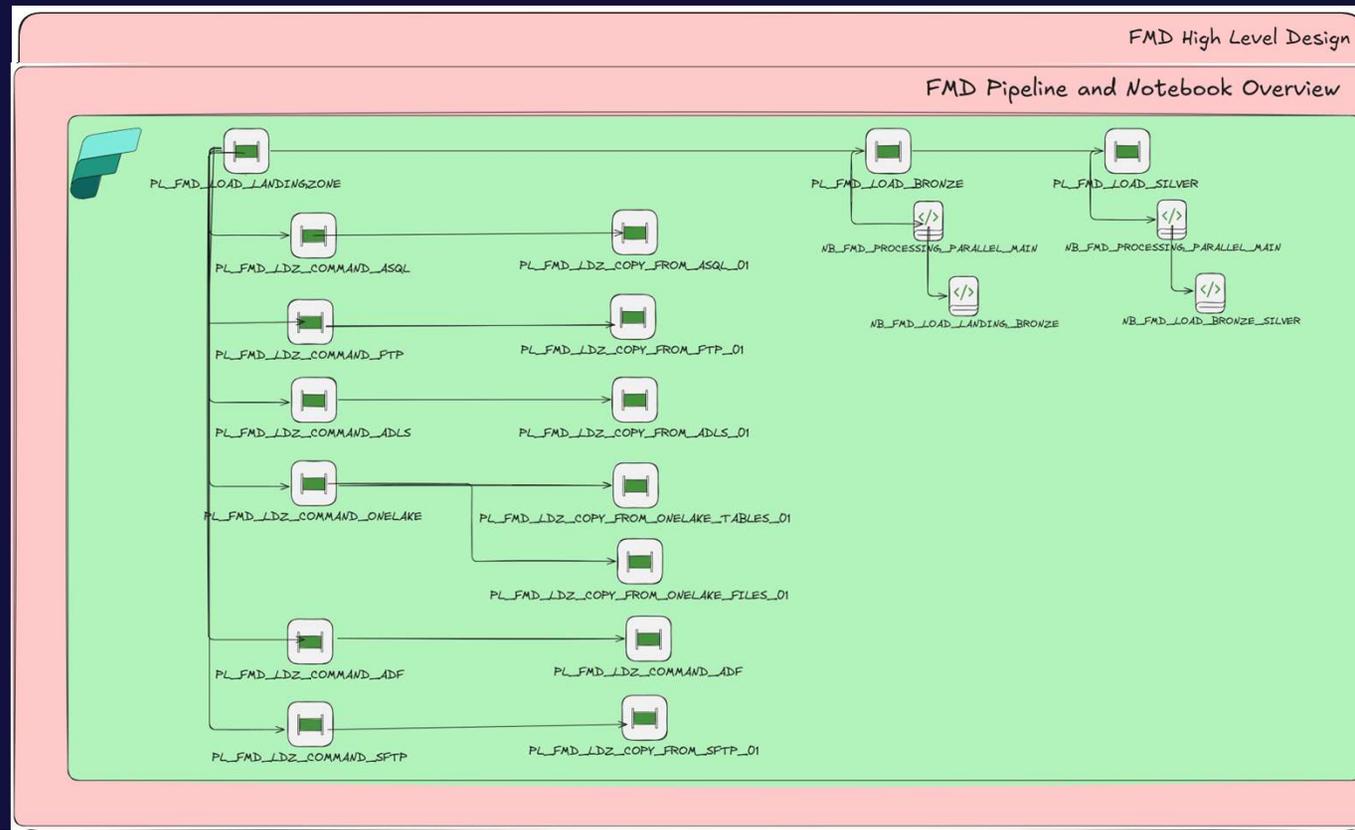
Item deployment

- Variable Library
- Environment



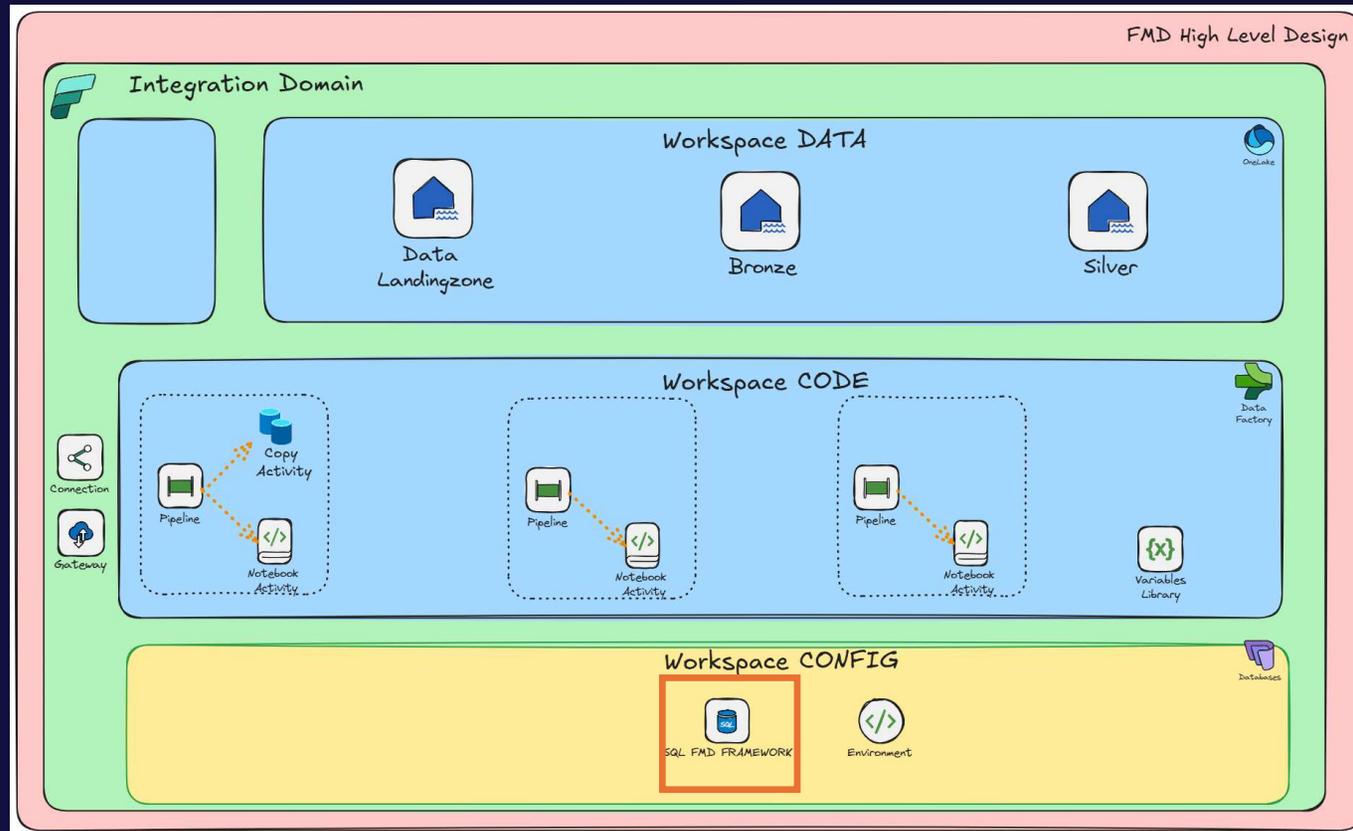
Item deployment

- Variable Library
- Environment
- Data Pipelines
- Notebooks



Item deployment

- Variable Library
- Environment
- Data Pipelines
- Notebooks
- Fabric Database



Demo

FABCON | SQLCON

ATLANTA26

JOIN THE
CONVERSATION

#FABCONSQLCON26

Unified data is created at ingestion, not at the end.

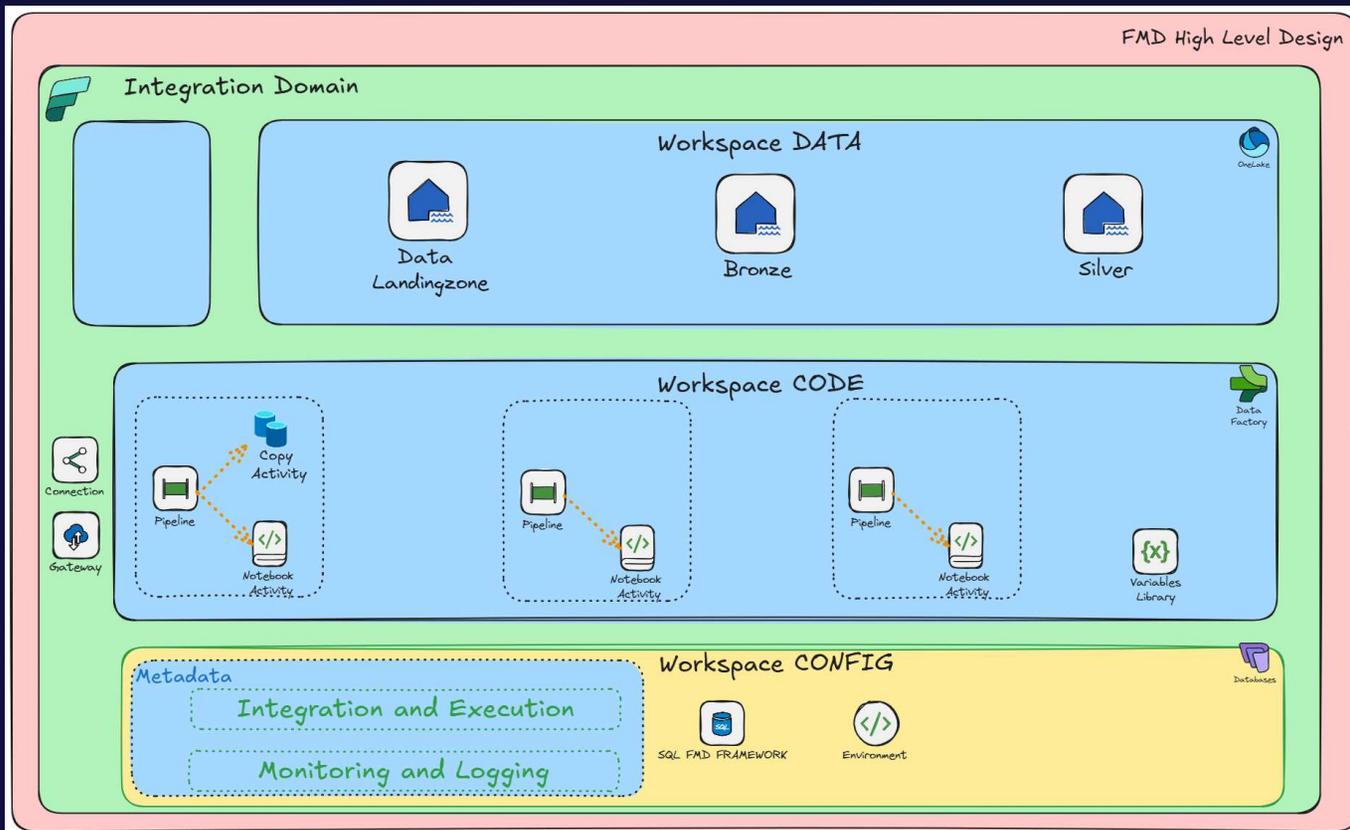


Preparing and Combining Ingredients

Data Integration

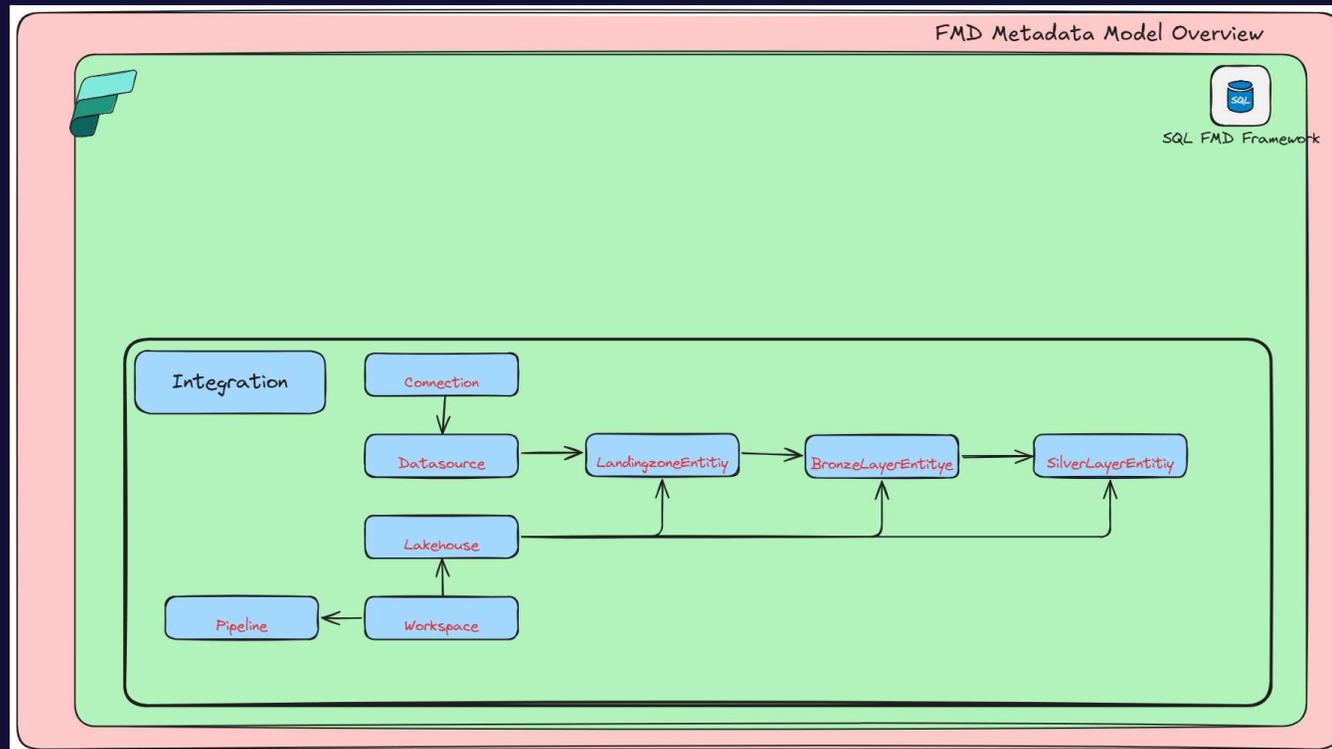
Data Integration

- Fabric database for metadata



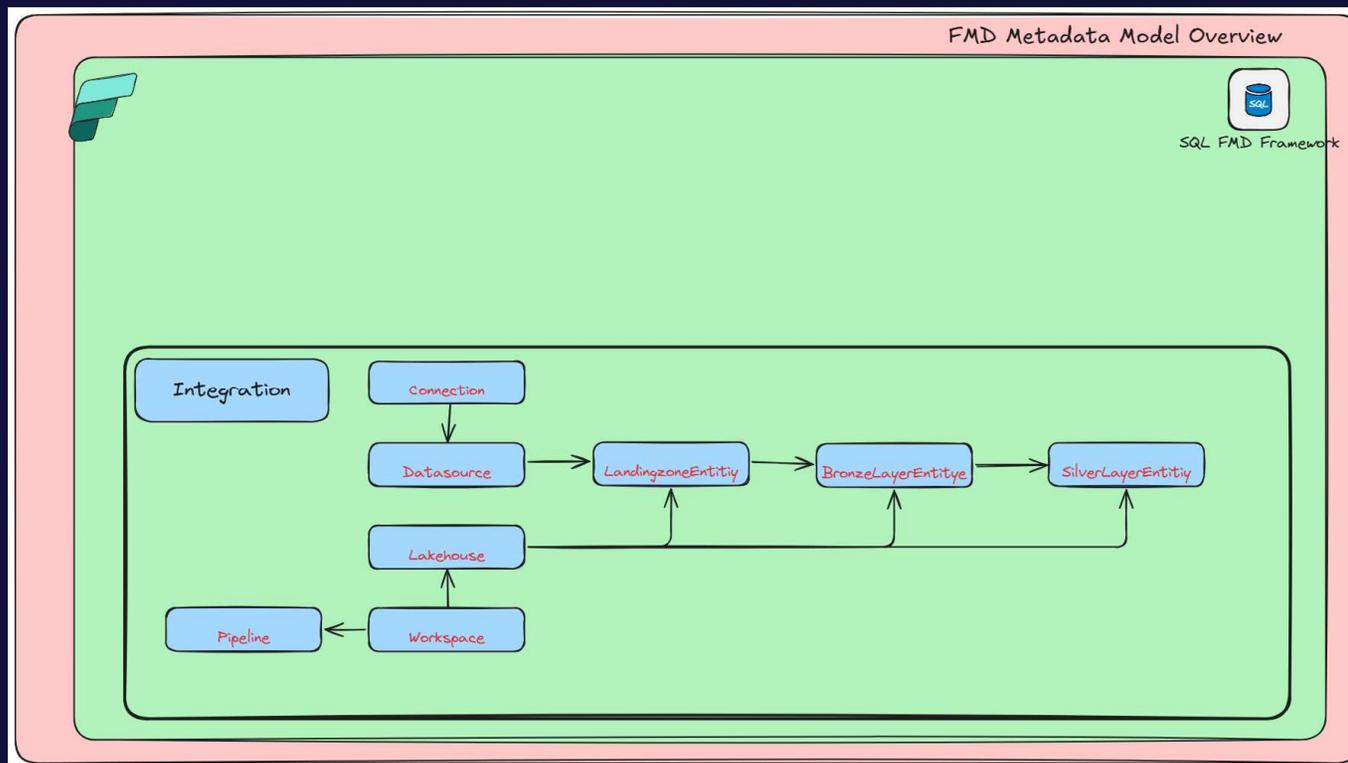
Data Integration

- Fabric database for metadata
- Integration
 - Tables
 - Views
 - Stored Procedures



Data Integration

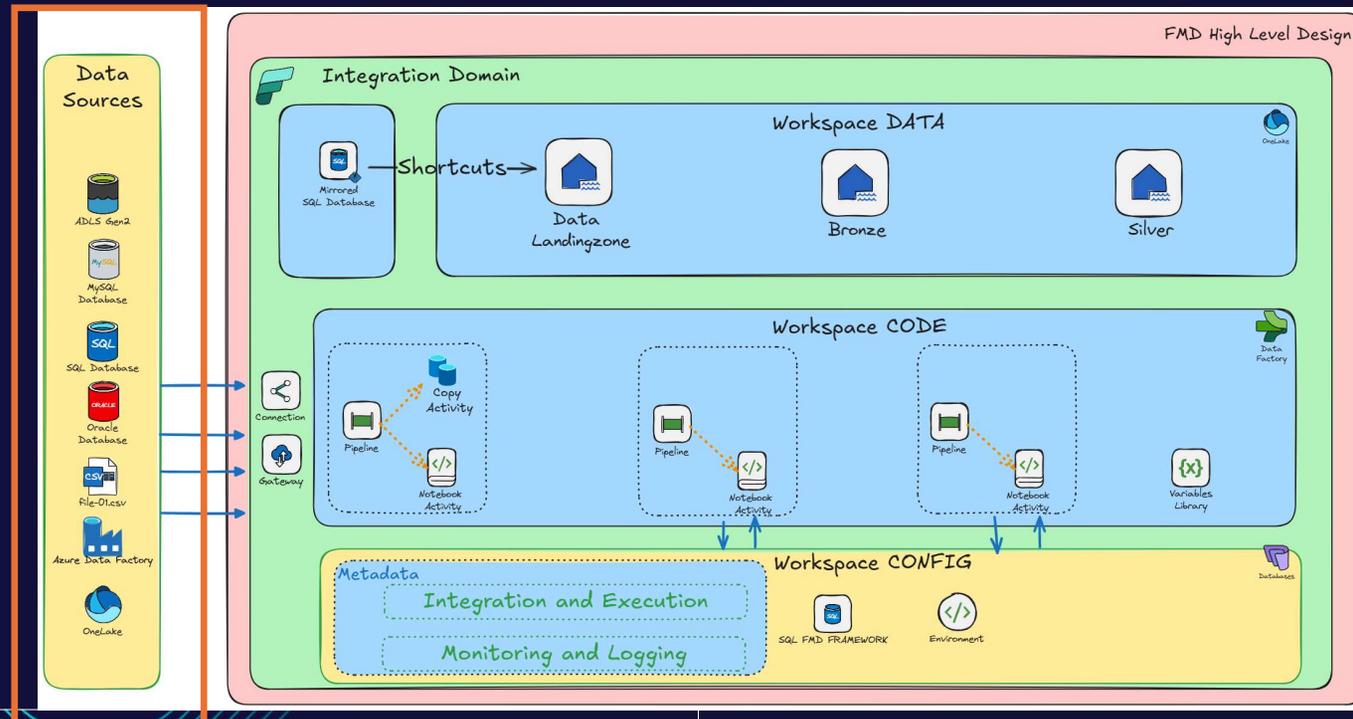
- Execution
 - Tables
 - Views
 - Stored Procedures
- Logging
 - Tables
 - Views
 - Stored Procedures



Data Sources

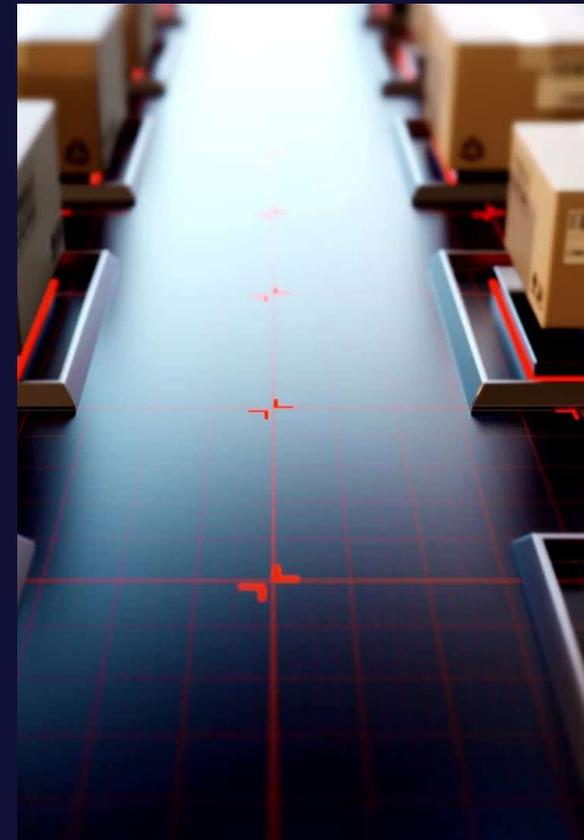
- SQL Server
- Data Lake Gen 2
- Onelake Tables
- Onelake Files
- FTP / SFTP
- Oracle
- Custom Notebooks(API)
- ADF (Connect to ADF pipeline)

All other connections can easily be added
Every connection needs a new Data Pipeline



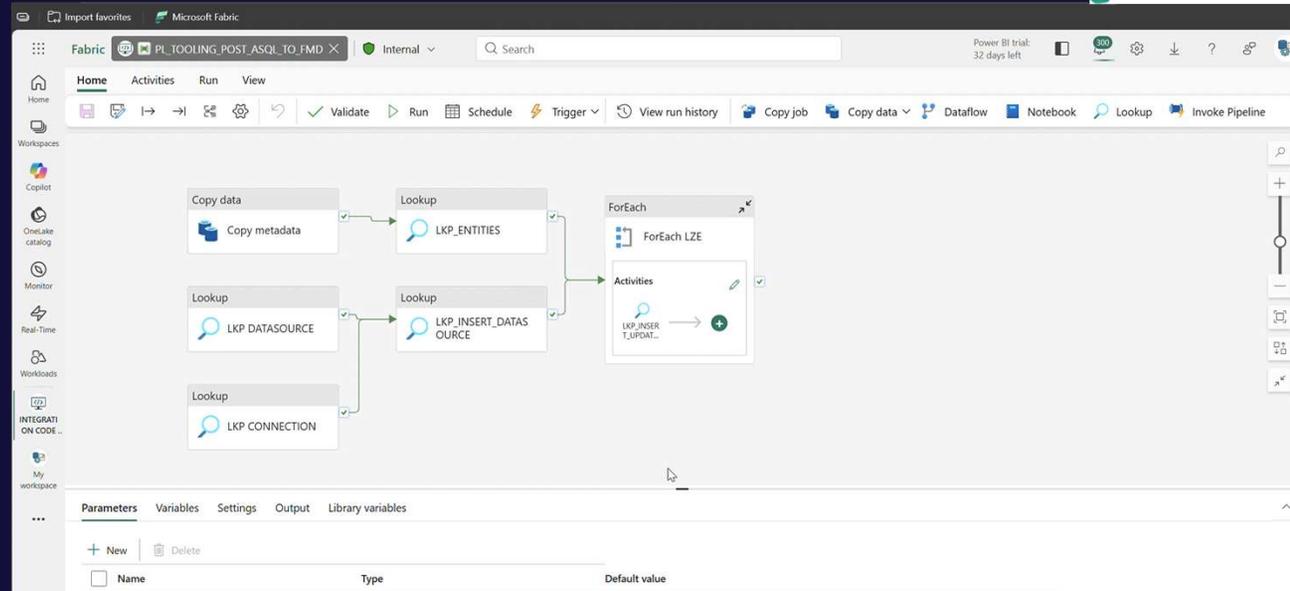
How to get data easily into Fabric Metadata Driven data model

- Metadata driven ingestion Pipeline
- Manual



Metadata driven ingestion Pipeline

- ConnectionGuid
- DatasourceName
- Namespace
- DatasourceType
- Tables(schema+table)



<input type="checkbox"/>	Name	Type	Default value
<input type="checkbox"/>	ConnectionGuid	S...	cf673e6a-13f6-4ebb-9cbb-4ba4ab390818
<input type="checkbox"/>	Data_WorkspaceGuid	S...	3e43b742-3e41-4ec0-a414-03adf83c08e7
<input type="checkbox"/>	DatasourceName	S...	WideWorldImporters
<input type="checkbox"/>	DatasourceNamespace	S...	wwi
<input type="checkbox"/>	DatasourceType	S...	ASQL_01
<input type="checkbox"/>	Tables	S...	('Warehouse.StockItems','Warehouse.PackageTypes','Purchasing.PurchaseOrders','Purchasing.PurchaseOrderLines','Sales.vCustomers','Sales.CustomerCategories','Sales.Or...

ONSQLCON26

Metadata driven ingestion Pipeline

The screenshot displays the Microsoft Fabric Data Pipeline Designer interface. The pipeline consists of the following activities:

- Copy data**: Copy metadata
- Lookup**: LKP_ENTITIES
- Lookup**: LKP_DATASOURCE
- Lookup**: LKP_INSERT_DATASOURCE
- Lookup**: LKP_CONNECTION
- ForEach**: ForEach LZE
- Activities**: LKP_INSERT_UPDATE...

The pipeline flow is: Copy metadata → LKP_ENTITIES → ForEach LZE. LKP_DATASOURCE and LKP_INSERT_DATASOURCE also feed into ForEach LZE. LKP_CONNECTION feeds into LKP_INSERT_UPDATE...

Below the pipeline, the **Parameters** tab is active, showing a table of parameters:

Name	Type	Default value
ConnectionGuid	String	cf673e6a-13f6-4ebb-9cbb-4ba4ab390818
Data_WorkspaceGuid	String	3e43b742-3e41-4ec0-a414-03adf83c08e7
DatasourceName	String	WideWorldImporters
DatasourceNamespace	String	wwi
DatasourceType	String	ASQL_01
Tables	String	('Warehouse.StockItems','Warehouse.PackageTypes','Purchasing.PurchaseOrders','Purcha:

How to add a new connection and entity

```

set @ConnectionGuid='11a8e5fe-fbca-4822-9ba4-9162cf56e6dd'
set @Name='CON_FMD_CONTOSO'
set @Type='ASQL_01'

EXECUTE @RC = [integration].[sp_UpsertConnection]
    @ConnectionGuid
    ,@Name
    ,@Type
  
```

```

set @ConnectionId='Id of connection you want to connect to'
set @DataSourceId=0
set @Name='Databasename'
set @Namespace='loaction in Landingzone'
set @Type='SQL'
set @Description=''

EXECUTE @RC = [integration].[sp_UpsertDataSource]
    @ConnectionId
    ,@DataSourceId
    ,@Name
    ,@Namespace
    ,@Type
    ,@Description
  
```

```

SET @DataSourceId=''
SET @WorkspaceGuid='3e43b742-3e41-4ec0-a414-03adf83c08e7'
SET @SourceSchema='dbo'
SET @SourceName='Employees'
SET @SourceCustomSelect=''
SET @FileName='dbo_Employees'
SET @FilePath='FMD'
SET @FileType='parquet'
SET @IsIncremental=''
SET @IsIncrementalColumn=''
SET @CustomNotebookName=''
SET @PrimaryKeys='EmployeeID'

EXECUTE @RC = [integration].[sp_UpsertLandingzoneBronzeSilver]
    @DataSourceId
    ,@WorkspaceGuid
    ,@SourceSchema
    ,@SourceName
    ,@SourceCustomSelect
    ,@FileName
    ,@FilePath
    ,@FileType
    ,@IsIncremental
    ,@IsIncrementalColumn
    ,@CustomNotebookName
    ,@PrimaryKeys
  
```

Add Connection

Add Datasource

Add Entity

Landingzone Entity Table

- Source Schema
- Source Name
- SourceCustomSelect
- File Type
- IsIncremental
- Incremental Column

123 DataSourceId	123 LakehouseId	480 SourceSchema	480 SourceName	480 FileName	480 FileType	480 FilePath	0/1 IsIncremental	480 IsIncrementalColumn
1	2	SalesLT	Product	SalesLT_Product	parquet	fmd	False	
1	2	SalesLT	Customer	SalesLT_Customer	parquet	fmd	True	ModifiedDate
2	2	Sales	Invoices	Sales_Invoices	parquet	fmd	False	NULL
3	2	dbo	Employees	dbo_Employees	parquet	fmd	False	
4	2	demo1	customers.csv	customers	csv	fmd	False	NULL
8	2	dl	people	people	parquet	fmd	False	NULL
2	2	Purchasing	PurchaseOrders	Purchasing_PurchaseOrders	parquet	fmd	False	NULL
2	2	Purchasing	PurchaseOrderLines	Purchasing_PurchaseOrderLines	parquet	fmd	False	NULL
2	2	Sales	CustomerCategories	Sales_CustomerCategories	parquet	fmd	False	NULL
2	2	Sales	OrderLines	Sales_OrderLines	parquet	fmd	False	NULL
2	2	Sales	Orders	Sales_Orders	parquet	fmd	False	NULL
2	2	Sales	BuyingGroups	Sales_BuyingGroups	parquet	fmd	False	NULL
2	2	Sales	InvoiceLines	Sales_InvoiceLines	parquet	fmd	False	NULL

BronzeLayer Entity Table

- Schema
- Name
- Primarykeys
- CleansingRules

ABC_LandingzoneEntityId	123_LakehouseId	ABC_Schema	ABC_Name	ABC_PrimaryKeys	ABC_FileType	ABC_CleansingRules
1	3	SalesLT	Product	ProductID	Delta	NULL
2	3	SalesLT	Customer	CustomerID	Delta	NULL
3	3	Sales	Invoices	InvoiceID	Delta	NULL
4	3	dbo	Employees	EmployeeID	Delta	NULL
5	3	demo1	customers	Customerid	Delta	NULL
10	3	di	people	Userid	Delta	NULL
19	3	Purchasing	PurchaseOrders	PurchaseOrderID	Delta	NULL
20	3	Purchasing	PurchaseOrderLines	PurchaseOrderLineID	Delta	NULL
21	3	Sales	CustomerCategories	CustomerCategoryID	Delta	NULL
22	3	Sales	OrderLines	OrderLineID	Delta	NULL
23	3	Sales	Orders	OrderID	Delta	NULL
25	3	Sales	InvoiceLines	InvoiceLineID	Delta	NULL
24	3	Sales	BuyingGroups	BuyingGroupID	Delta	NULL

Silver Layer Entity Table

- Schema
- Name
- CleansingRules

ABC Schema	ABC Name	ABC FileType	ABC CleansingRules
SalesLT	Product	delta	NULL
SalesLT	Customer	delta	NULL
Sales	Invoices	delta	NULL
dbo	Employees	delta	NULL
demo1	customers	delta	NULL
dl	people	delta	NULL
Purchasing	PurchaseOrderLines	delta	NULL
Purchasing	PurchaseOrders	delta	NULL
Sales	CustomerCategories	delta	NULL
Sales	OrderLines	delta	NULL
Sales	InvoiceLines	delta	NULL
Sales	Orders	delta	NULL
Sales	BuyingGroups	delta	NULL

Demo

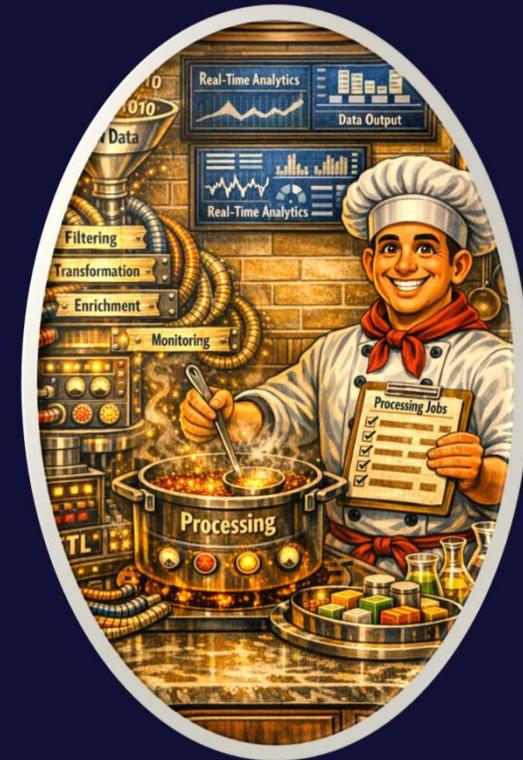
FABCON | SQLCON

ATLANTA26

JOIN THE
CONVERSATION

#FABCONSQLCON26

Processing is where data turns into value.



Cooking the Disk

Data Processing

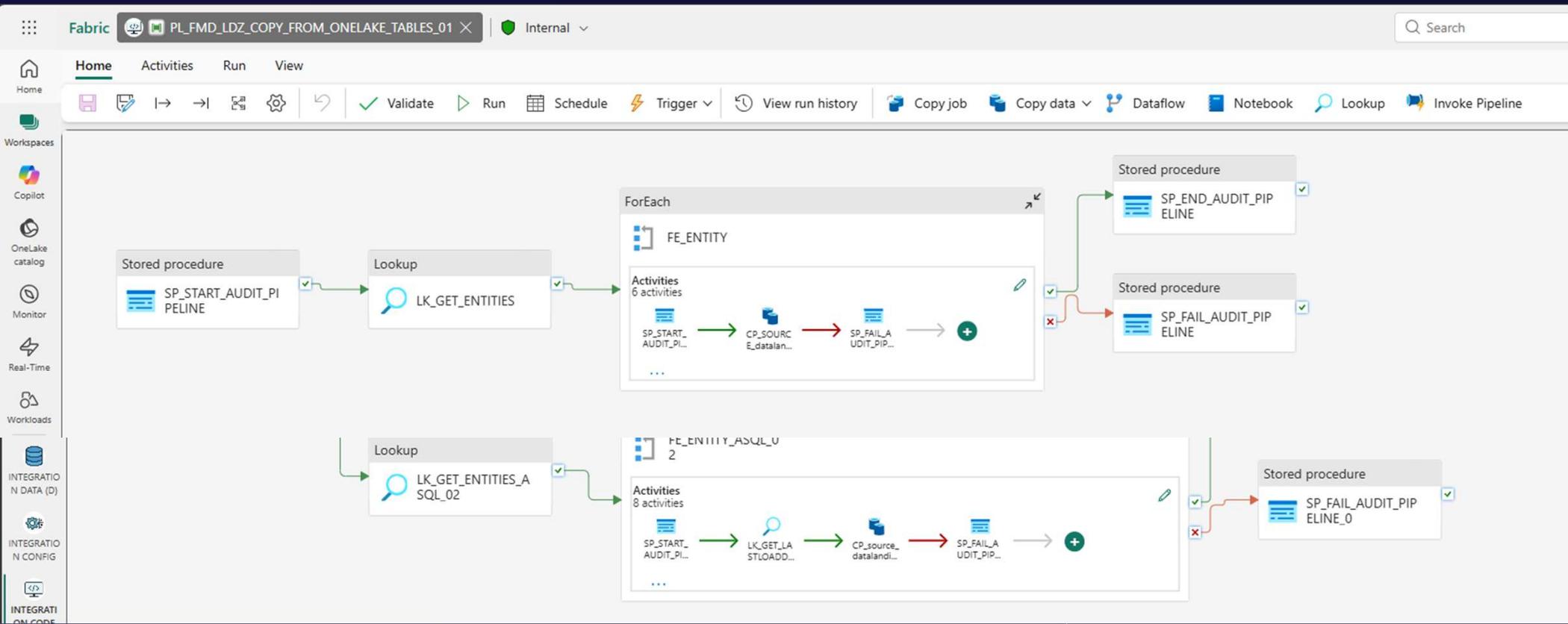
FABCON | SQLCON

ATLANTA26

JOIN THE
CONVERSATION

#FABCONSQLCON26

Data Processing (source to Data Landingzone)



Data Processing (Data LDZ to Bronze/Silver)

Fabric PL_FMD_LOAD_BRONZE Internal

Home Activities Run View

Validate

Stored procedure SP_START_AUDIT_PIPELINE → Lookup LK_GET_ENRZ

Parameters

```

1 from json import loads, dumps
2 import uuid
3 from datetime import datetime
4 NotebookExecutionId = str(uuid.uuid4())
5

```

PySpark (Python)

```

1 variable_settings=notebookutils.variableLibrary.getLibrary("VAR_FMD")

```

PySpark (Python)

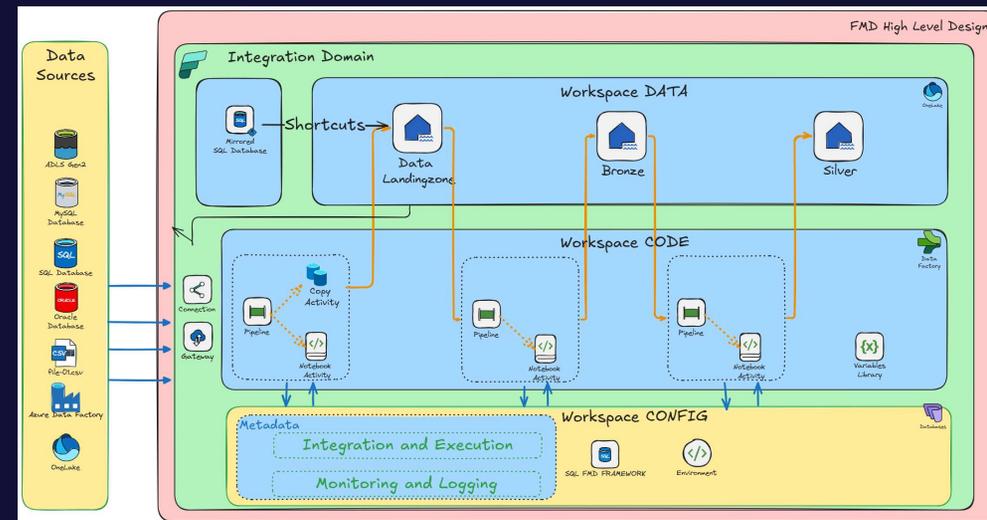
```

1 Path = ""
2 useRootDefaultLakehouse= True
3 PipelineRunGuid = ""
4 PipelineGuid = ""
5 TriggerGuid = ""
6 TriggerTime = ""

```

Data Processing

- Processing at Scale (Run Multiple)
- Data Quality
- Data Cleansing
- Queing



Data Processing Cleansing rules (BRZ/SLV)

- NB_FMD_DQ_CLEANSING
 - Built in Rules
 - normalize_text
 - split
 - fill_nulls
 - parse_datetime

```
df_norm = normalize_text(
df,
columns=["title"],
args={"case": "lower", "empty_as_null": False}
)
```

```
df_norm = normalize_text(
df,
columns=["name"],
args={"case": "title", "collapse_spaces": False}
)
```

```
DECLARE @RC int
DECLARE @SilverLayerEntityId int
DECLARE @CleansingRules nvarchar(max)

-- TODO: Set parameter values here.

EXECUTE @RC = [integration].[sp_UpsertSilverCleansingRule]
24
,[{"function": "normalize_text", "columns": "CustomerName", "parameters": {"case": "upper"}}]
GO
```

Messages	Results	Search								
	<table border="1"> <thead> <tr> <th>123</th> <th>SilverLayerEntityId</th> <th>ABC</th> <th>CleansingRules</th> </tr> </thead> <tbody> <tr> <td>123</td> <td>SilverLayerEntityId</td> <td>ABC</td> <td>CleansingRules</td> </tr> </tbody> </table>	123	SilverLayerEntityId	ABC	CleansingRules	123	SilverLayerEntityId	ABC	CleansingRules	<input type="text" value="Search"/>
123	SilverLayerEntityId	ABC	CleansingRules							
123	SilverLayerEntityId	ABC	CleansingRules							

Fabric Metadata-driven Framework

Demo

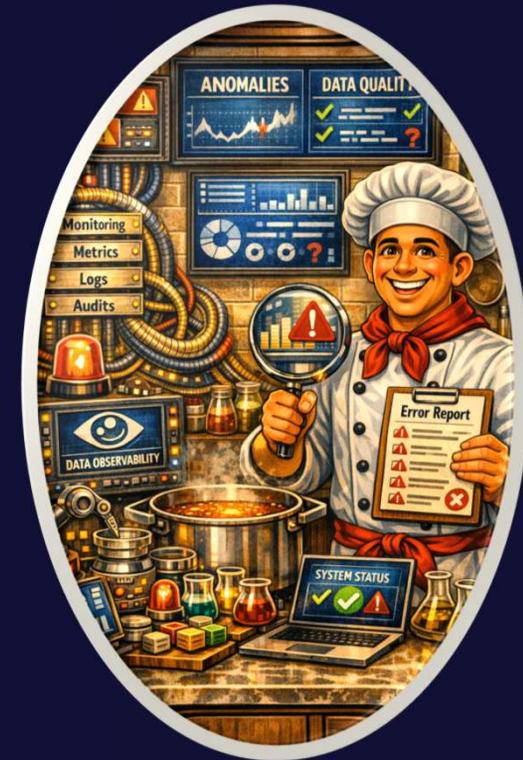
FABCON | SQLCON

ATLANTA26

JOIN THE
CONVERSATION

#FABCONSQLCON26

You can't scale what you can't see.

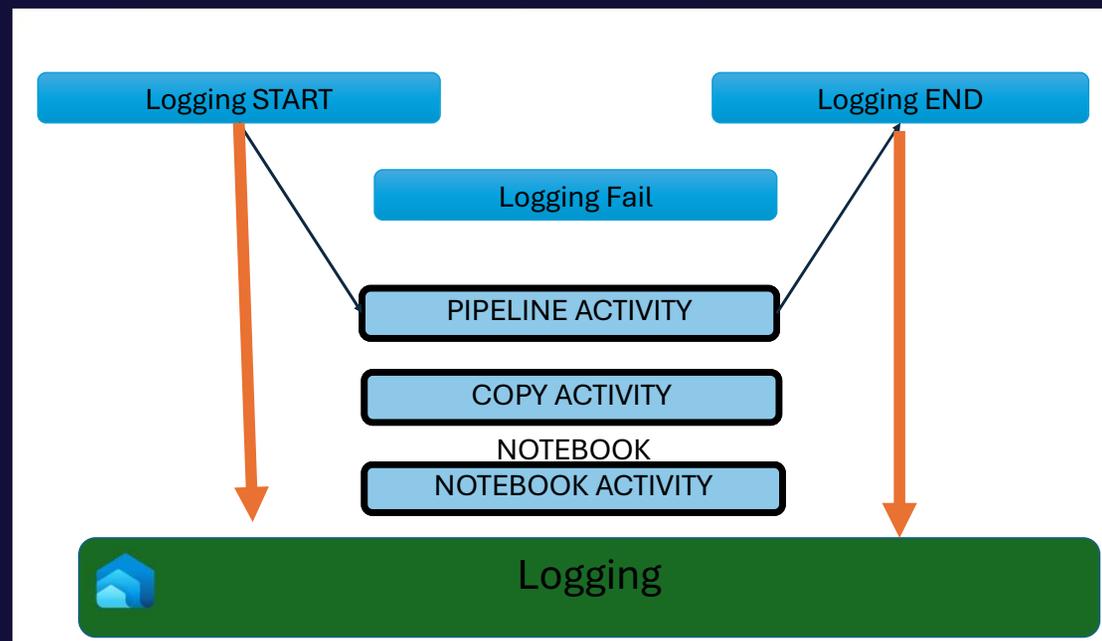


, Monitoring, and Quality Control

Data Observability

Data Observability

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



Data Observability

- Add Information about pipelines
- Adding System Variables

General Settings

Data store type Workspace External

Warehouse

Stored procedure name *

Stored procedure parameters

← Import + New Delete

<input type="checkbox"/>	Name	Type
<input type="checkbox"/>	<input type="text" value="LogData"/>	<input type="text" value="String"/>
<input type="checkbox"/>	<input type="text" value="LogType"/>	<input type="text" value="String"/>
<input type="checkbox"/>	<input type="text" value="PipelineGuid"/>	<input type="text" value="Guid"/>
<input type="checkbox"/>	<input type="text" value="PipelineName"/>	<input type="text" value="String"/>
<input type="checkbox"/>	<input type="text" value="PipelineParameters"/>	<input type="text" value="String"/>
<input type="checkbox"/>	<input type="text" value="PipelineParentRunGuid"/>	<input type="text" value="Guid"/>
<input type="checkbox"/>	<input type="text" value="PipelineRunGuid"/>	<input type="text" value="Guid"/>
<input type="checkbox"/>	<input type="text" value="TriggerGuid"/>	<input type="text" value="Guid"/>
<input type="checkbox"/>	<input type="text" value="TriggerTime"/>	<input type="text" value="DateTime"/>
<input type="checkbox"/>	<input type="text" value="TriggerType"/>	<input type="text" value="String"/>
<input type="checkbox"/>	<input type="text" value="WorkspaceGuid"/>	<input type="text" value="Guid"/>

Pipeline expression builder

Add dynamic content below using any combination of expressions, functions and system variables.

```
@pipeline().Pipeline
```

Clear contents

Parameters System variables Functions Variables

Search

Pipeline ID
ID of the pipeline

Pipeline Name
Name of the pipeline

Pipeline group ID
ID of the group to which the pipeline run belongs

Pipeline run ID
ID of the specific pipeline run

Pipeline trigger ID
ID of the trigger that invokes the pipeline

Pipeline trigger time
Time when the trigger that invoked the pipeline. The trigger time is the actual fired time, not the sched...

Pipeline trigger type
Type of the trigger that invoked the pipeline (Manual, Scheduler)

Pipeline triggered by pipeline ID
ID of the pipeline that triggered this pipeline. Applicable when a pipeline run is triggered by an Execut...

Pipeline triggered by pipeline name
Name of the pipeline that triggered this pipeline. Applicable when a pipeline run is triggered by an Exe...

Pipeline triggered by pipeline run ID
Run ID of the pipeline that triggered this pipeline. Applicable when a pipeline run is triggered by an Exe...

Workspace ID
ID of the workspace the pipeline run is running within

Data Observability

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

```

"status": "Succeeded",
"result": {
  "runId": "10883ea1-cac8-4d1b-92d3-03f8ca451054",
  "runStatus": "Succeeded",
  "sessionId": "188bc363-7872-453e-9369-8a18c115f6e8",
  "sparkPool": "1acclc9e-b8d2-4f1d-ba36-5ba8e5088067",
  "error": null,
  "lastCheckedOn": "2025-03-17T18:29:37.44Z",
  "metadata": {
    "isForPipeline": null,
    "runStartTime": "2025-03-17T18:25:13.2077314Z",
    "runEndTime": "2025-03-17T18:29:37.4521264Z"
  }
},
"highConcurrencyModeStatus": null,
"exitValue": "[{'TableName': 'Sales_Orders_0_0', 'exitVal': \"{'CopyOutput':
},
"message": "Notebook execution is in Succeeded state, runId: 10883ea1-cac8-4d1b-9
"SparkMonitoringURL": "https://app.powerbi.com/workloads/de-ds/sparkmonitor/fcd42
"executionDuration": 266

```

Pipeline expression builder

Add dynamic content below using any combination of expressions, functions and system variables.

```

{
  "Action": "End",
  @{{activity('NB_Landing_to_Bronze').output.result.exitValue}}
}

```

Data Observability



PipelineExecution X

Data preview - PipelineExecution

Showing 1000 rows

Search

ID	UID Workspace...	UID PipelineRun...	UID PipelinePare...	UID PipelineGuid	ABC PipelineName	ABC PipelinePara...	ABC TriggerType	UID TriggerGuid	TriggerTime	ABC LogType	LogDateTime	ABC LogData
1	1ACC1C9E-B8D2...	C19BCD8F-6F46...	NULL	67B39D07-7CE2...	PL_FMD_LOAD_...	NULL	Manual	AA34BEA8-E884...	2025-03-17 18:1...	EndPipeline	2025-03-17 18:3...	{ "Action": "End" }
2	1ACC1C9E-B8D2...	1C939008-FF26-...	NULL	DC3E6A24-2155...	PL_FMD_LOAD_S...	NULL	Manual	6C182932-0447-...	2025-03-17 18:3...	EndPipeline	2025-03-17 18:3...	{ "Action": "En...
3	1ACC1C9E-B8D2...	1C939008-FF26-...	NULL	DC3E6A24-2155...	PL_FMD_LOAD_S...	NULL	Manual	6C182932-0447-...	2025-03-17 18:3...	StartPipeline	2025-03-17 18:3...	{ "Action": "Start...
4	1ACC1C9E-B8D2...	2BB79DC5-CB19...	NULL	69523C39-B819-...	PL_FMD_LOAD_B...	NULL	Manual	AA45F0C5-24D3...	2025-03-17 18:2...	StartPipelineActi...	2025-03-17 18:2...	{ "Action": "Start...
5	1ACC1C9E-B8D2...	C19BCD8F-6F46...	NULL	67B39D07-7CE2...	PL_FMD_LOAD_...	NULL	Manual	AA34BEA8-E884...	2025-03-17 18:1...	EndPipeline	2025-03-17 18:2...	{ "Action": "Err...
6	1ACC1C9E-B8D2...	2BB79DC5-CB19...	NULL	69523C39-B819-...	PL_FMD_LOAD_B...	NULL	Manual	AA45F0C5-24D3...	2025-03-17 18:2...	EndPipelineActiv...	2025-03-17 18:3...	{ "Action": "En...
7	1ACC1C9E-B8D2...	CF5D046E-7D58...	NULL	C0737F04-B4E9-...	PL_FMD_LDZ_CO...	NULL	Manual	F26FF0CD-71A9-...	2025-03-17 18:1...	EndPipeline	2025-03-17 18:2...	{ "Action": "En...
8	1ACC1C9E-B8D2...	D238165C-FCDF...	NULL	B8236D27-F8C2...	PL_FMD_LDZ_CO...	NULL	Manual	800DB4CE-CA74...	2025-03-17 18:1...	EndPipeline	2025-03-17 18:2...	{ "Action": "En...
9	1ACC1C9E-B8D2...	D238165C-FCDF...	NULL	B8236D27-F8C2...	PL_FMD_LDZ_CO...	Invoices	Manual	800DB4CE-CA74...	2025-03-17 18:1...	EndCopyActivity	2025-03-17 18:2...	{ "Action": "En...
10	1ACC1C9E-B8D2...	D238165C-FCDF...	NULL	B8236D27-F8C2...	PL_FMD_LDZ_CO...	InvoiceLines	Manual	800DB4CE-CA74...	2025-03-17 18:1...	EndCopyActivity	2025-03-17 18:2...	{ "Action": "En...
11	1ACC1C9E-B8D2...	D238165C-FCDF...	NULL	B8236D27-F8C2...	PL_FMD_LDZ_CO...	OrderLines	Manual	800DB4CE-CA74...	2025-03-17 18:1...	EndCopyActivity	2025-03-17 18:2...	{ "Action": "En...
12	1ACC1C9E-B8D2...	D238165C-FCDF...	NULL	B8236D27-F8C2...	PL_FMD_LDZ_CO...	Orders	Manual	800DB4CE-CA74...	2025-03-17 18:1...	EndCopyActivity	2025-03-17 18:2...	{ "Action": "En...
13	1ACC1C9E-B8D2...	D238165C-FCDF...	NULL	B8236D27-F8C2...	PL_FMD_LDZ_CO...	BuyingGroups	Manual	800DB4CE-CA74...	2025-03-17 18:1...	EndCopyActivity	2025-03-17 18:2...	{ "Action": "En...
14	1ACC1C9E-B8D2...	D238165C-FCDF...	NULL	B8236D27-F8C2...	PL_FMD_LDZ_CO...	PurchaseOrderLi...	Manual	800DB4CE-CA74...	2025-03-17 18:1...	EndCopyActivity	2025-03-17 18:2...	{ "Action": "En...

Conclusion

	Out of the Box Framework	Custom Made Framework	Fabric Metadata-driven Framework
Ready to use	✓	✗	✓
Rapid Implementation	✓	✗	✓
Lower development effort	✓	✗	✓ ✗
Lower upfront cost	✓	✗	✓ ✗
Support and Updated	✓	✗	✓ ✗
Tailored to specific needs.	✗	✓	✗
Full control over design and features.	✗	✓	✓
Easy to extend.	✗	✓	✓

Fabric Metadata-driven Framework

Recap

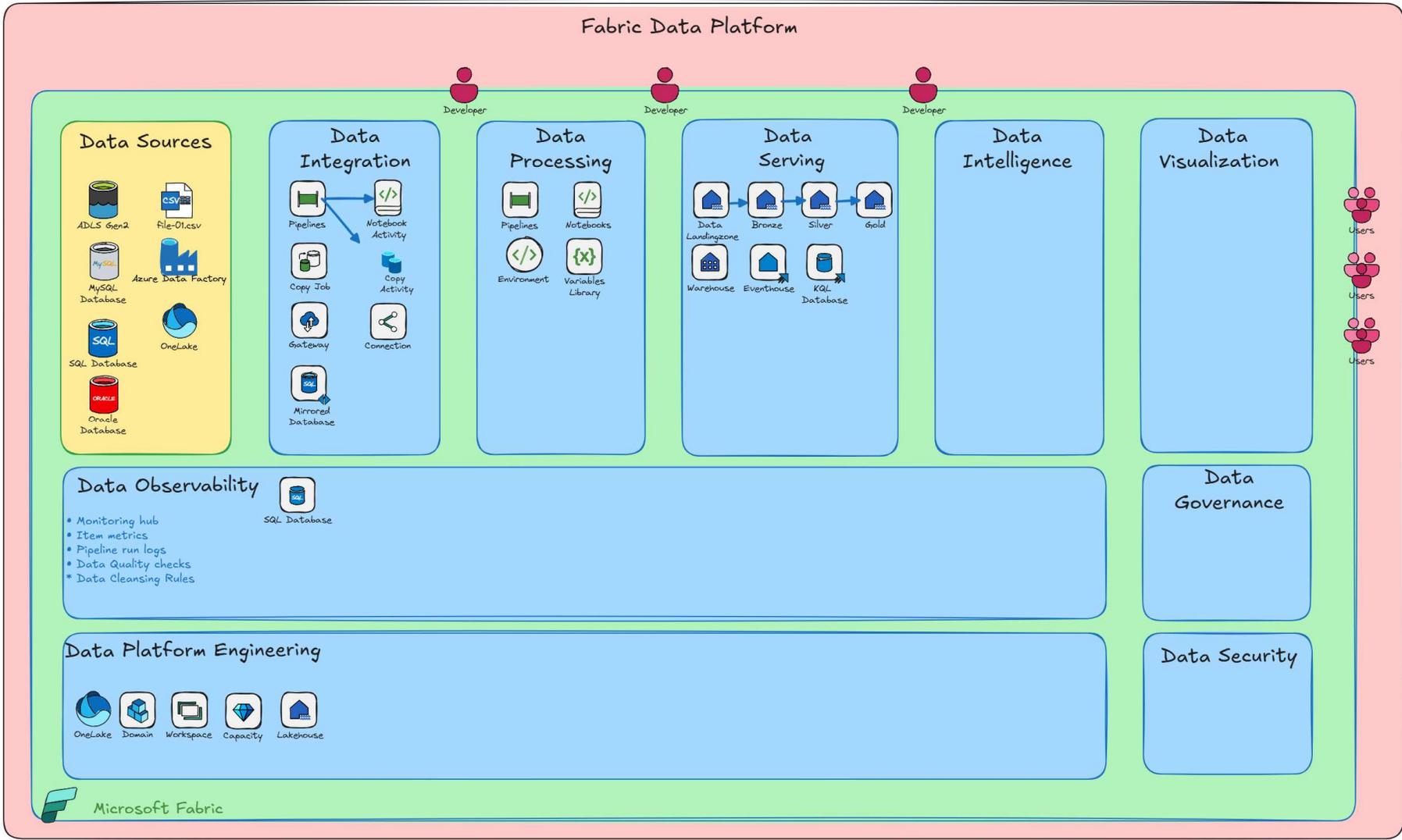
FABCON | SQLCON

ATLANTA26

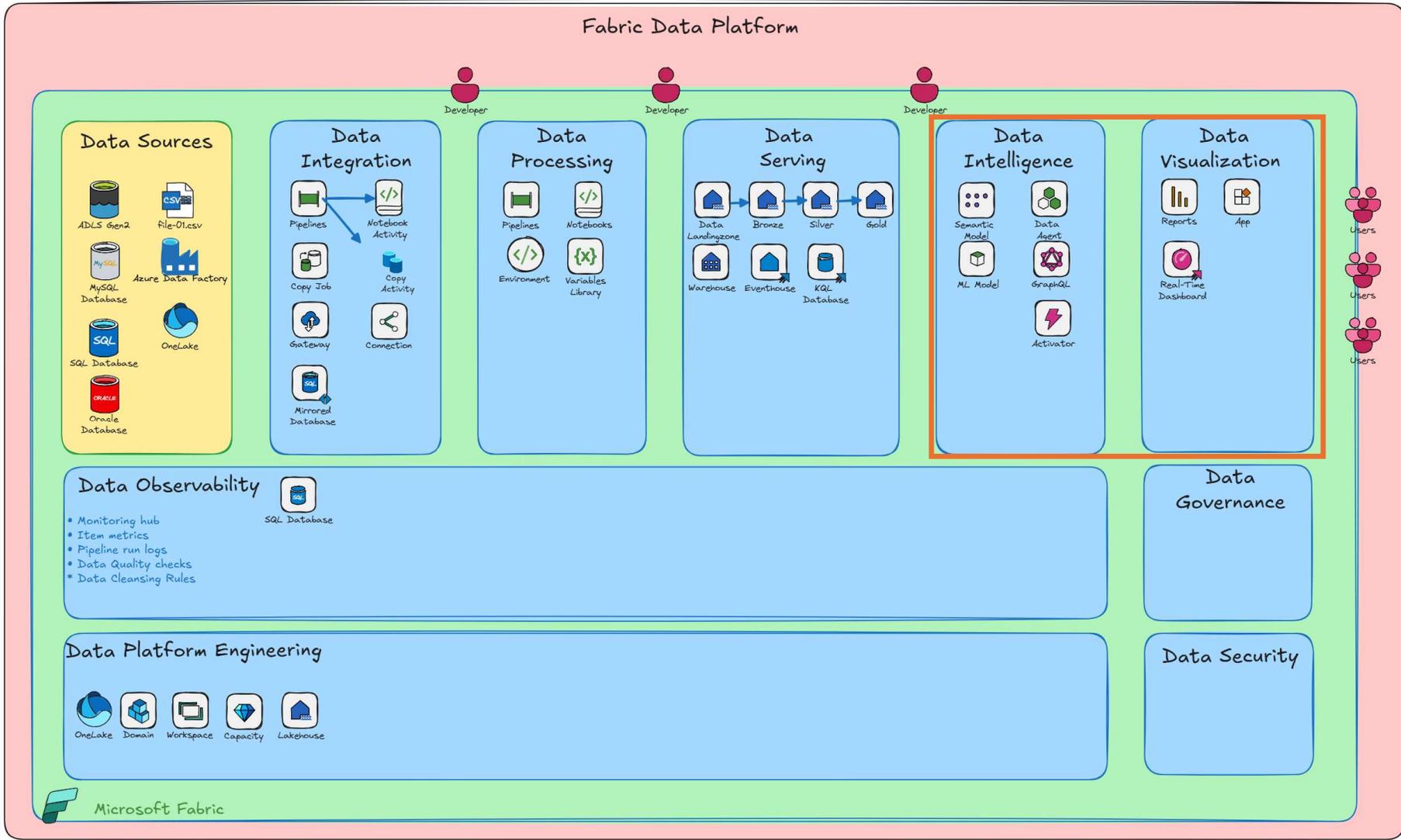
JOIN THE
CONVERSATION

#FABCONSQLCON26

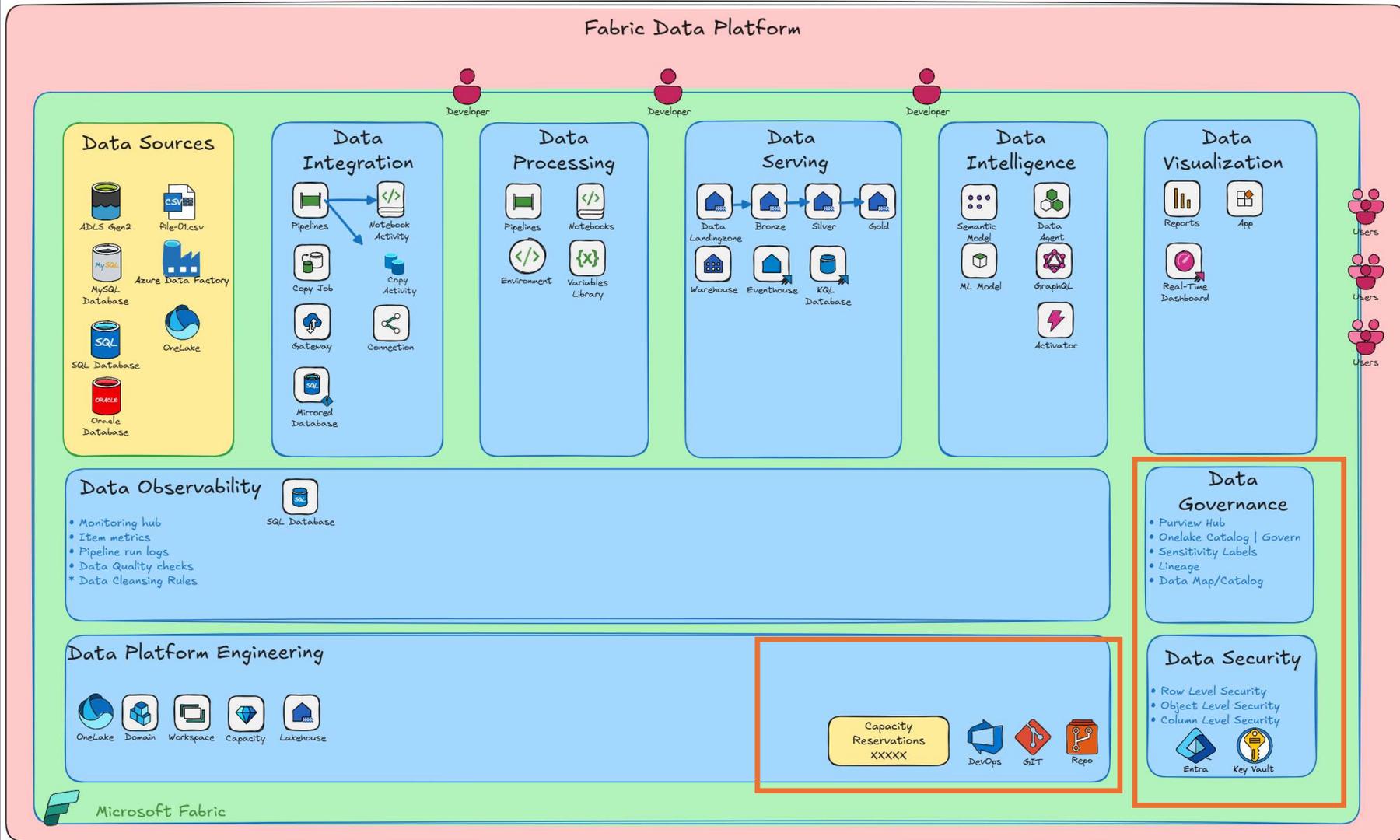
Fabric Data Platform



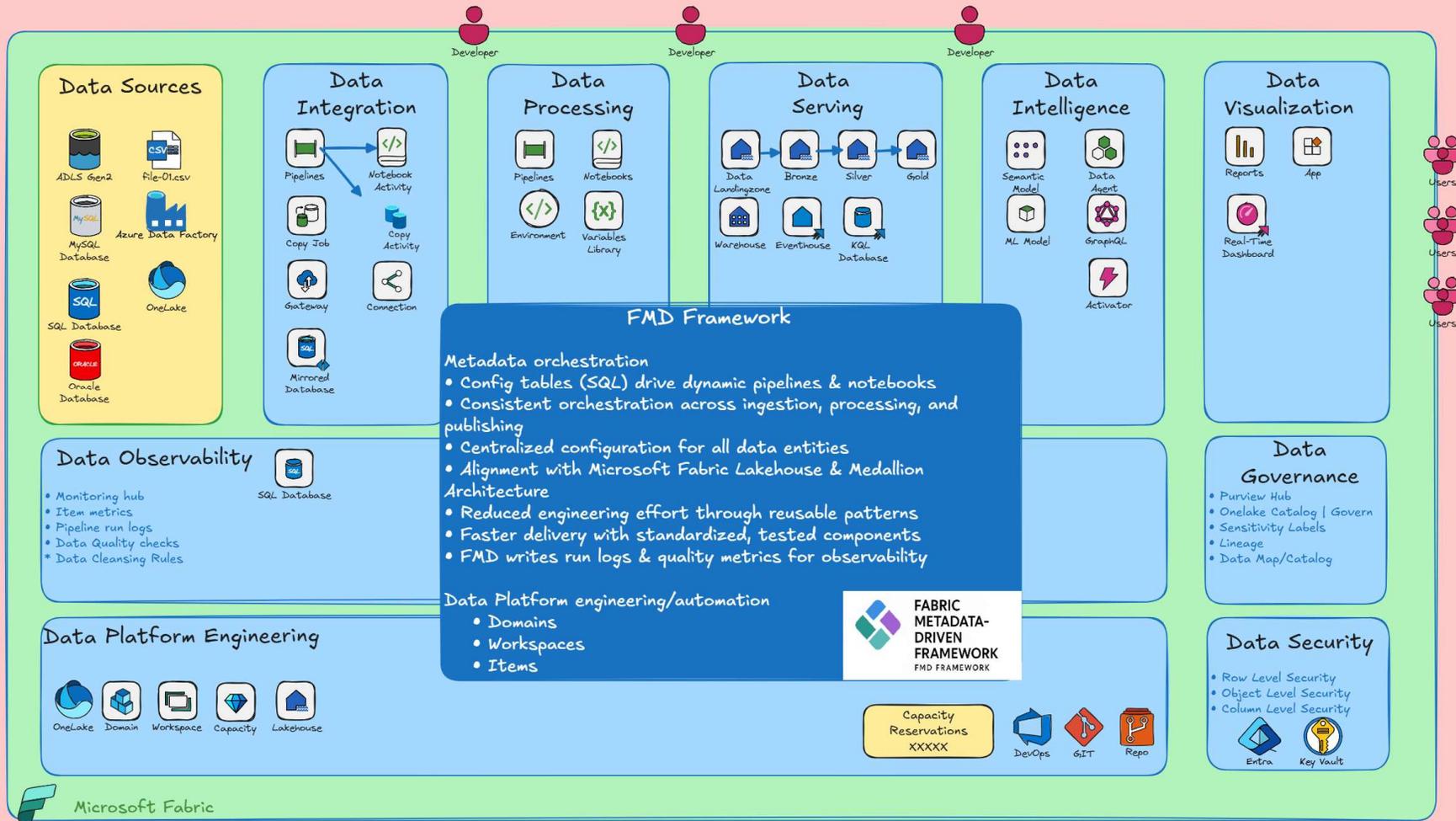
Fabric Data Platform



Fabric Data Platform



Fabric Data Platform

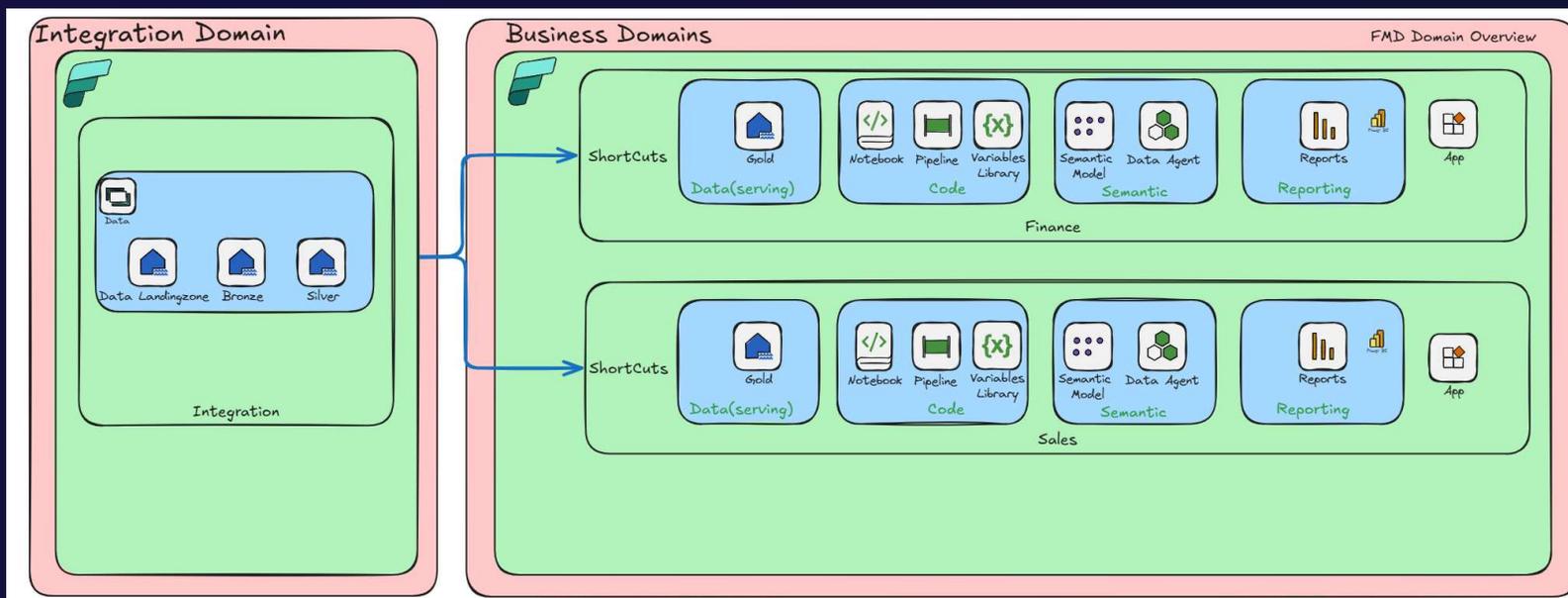


Your AI Foundation with Fabric is now Ready

- AI-ready data foundation built fully on Fabric, leveraging metadata-driven design for consistency, automation and reuse.
- **Standardized** ingestion and processing patterns ensure reliable, repeatable data flows that accelerate ML and AI development.
- **Centralized metadata model** enables dynamic configuration, reducing engineering overhead and enabling scalable AI workloads.
- **Medallion Architecture** provides clean, structured, high-quality data pipelines ideal for model training, RAG patterns, and generative AI scenarios.
- Unified Data Platform engineering workspace (notebooks, pipelines, Fabric databases, variables) streamlines data prep, feature engineering, and experimentation.
- **End-to-end observability** ensures full logging and monitoring, critical for AI governance, model audits, and responsible AI frameworks.
- **Native** Fabric capabilities like OneLake, Real-Time Intelligence, Data Factory, and Copilot create an integrated platform for building and operationalizing AI.
- **Extensible and future-proof foundation** positioned to support enterprise-scale AI, including automation, predictive analytics, and generative AI workloads.

FMD Business Domain Framework

- A modular extension of the Fabric Metadata-Driven Framework (FMD) designed to simplify, standardize, and automate deployment of Business Domains within Microsoft Fabric.
- The Business Domain Framework enables organizations to scale analytics through a governed, metadata-driven approach where each business domain (e.g., Sales, HR, Finance, Operations) is deployed consistently using reusable patterns, automated provisioning, and ready-to-use template assets.



Feedback / Ideas



FMD Framework

How was the session?

Let's connect
LinkedIn



THANK YOU