



From Manual to Automated: Master Metadata-Driven Design in Fabric



From Manual to Automated: Master Metadata-Driven Design in
Fabric

Erwin de Kreuk

Technology Lead Data, InSpark, Netherlands

Let's connect

-  [@erwindekreuk.bsky.social](https://bsky.social/@erwindekreuk)
-  linkedin.com/in/erwindekreuk
-  erwindekreuk.com
-  github.com/edkreuk
-  <https://sessionize.com/erwin-de-kreuk/>
-  Dutchfabricusergroup.com



THANK YOU SPONSORS!

Platinum



Gold

Lucient



b.telligent
smart data. smart decisions.



HEDDA.IO

COHESITY

Bronze

paiqo

The Platform & AI Company



redgate

devart



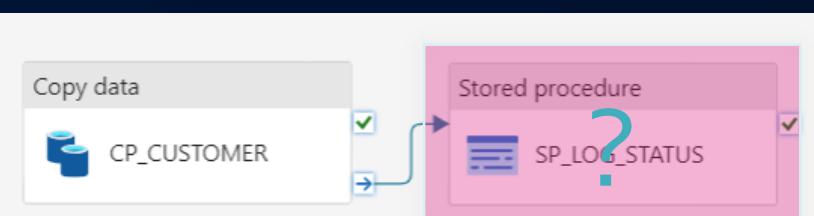
Power BI Camp
www.linearis.at

Problem statement

The screenshot shows the Power BI FabricDemo interface. On the left, there's a sidebar with various navigation options like Home, Create, Browse, OneLake data hub, Apps, Metrics, Monitor, Learn, Real-Time hub, Workspaces, and FabricDemo. The main area displays a list of data pipelines under the 'FabricDemo > Chaos' folder. The pipelines listed are:

Name	Type
PL_EXTRACT_AW_CUSTOMER	Data pipeline
PL_EXTRACT_AW_PRODUCT_CATEGORY	Data pipeline
PL_EXTRACT_AW_PRODUCT_DESCRIPTION	Data pipeline
PL_EXTRACT_AW_PRODUCT_DESCRIPTION_LINEAGE	Data pipeline
PL_EXTRACT_AW_PRODUCT_MODEL	Data pipeline
PL_EXTRACT_AW_PRODUCT_MODEL_PRODUCT_DESCRIPTION	Data pipeline
PL_EXTRACT_AW_PRODUCTS	Data pipeline
PL_EXTRACT_AW_SALES_ORDER_DETAIL	Data pipeline
PL_EXTRACT_AW_SALES_ORDER_HEADER	Data pipeline
PL_EXTRACT_AW_SALES_ORDER_LINE	Data pipeline
PL_EXTRACT_WWI_ADDRESS	Data pipeline
PL_EXTRACT_WWI_CITIES	Data pipeline
PL_EXTRACT_WWI_COUNTRIES	Data pipeline
PL_EXTRACT_WWI_DELIVERY_METHODS	Data pipeline
PL_EXTRACT_WWI_PAYMENT_METHODS	Data pipeline
PL_EXTRACT_WWI_PEOPLE	Data pipeline
PL_EXTRACT_WWI_STATE_PROVINCES	Data pipeline
PL_EXTRACT_WWI_TRANSACTION_TYPES	Data pipeline

Content of each pipeline



I will not repeat myself
I will not repeat myself



DON'T REPEAT YOURSELF

Repetition is the root of all software evil.

Agenda



- Metadata Driven Framework
- Data Platform Engineering
- Data Integration
- Data Processing
- Data Monitoring
- Q & A

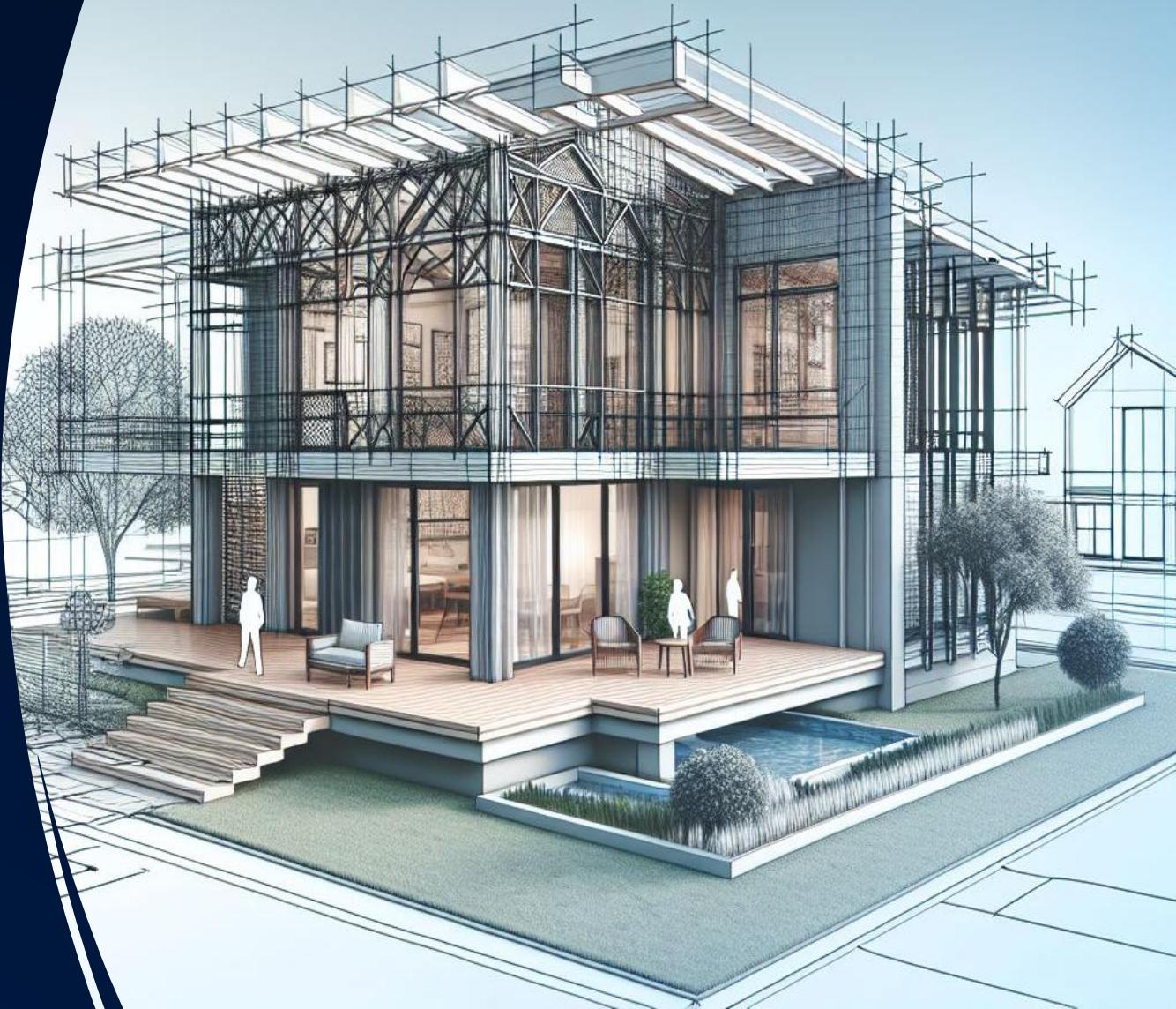
Out-of-the-Box Framework

- 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





Microsoft Fabric

The unified data platform for AI transformation



Data
Factory



Analytics



Databases



Real-Time
Intelligence



Power BI



Industry
Solutions



Partner
workloads



AI



OneLake



Microsoft Purview



The unified data platform for AI transformation



Data
Factory



Analytics



Databases



Real-Time
Intelligence



Power BI



Industry
Solutions



Partner
workloads



AI

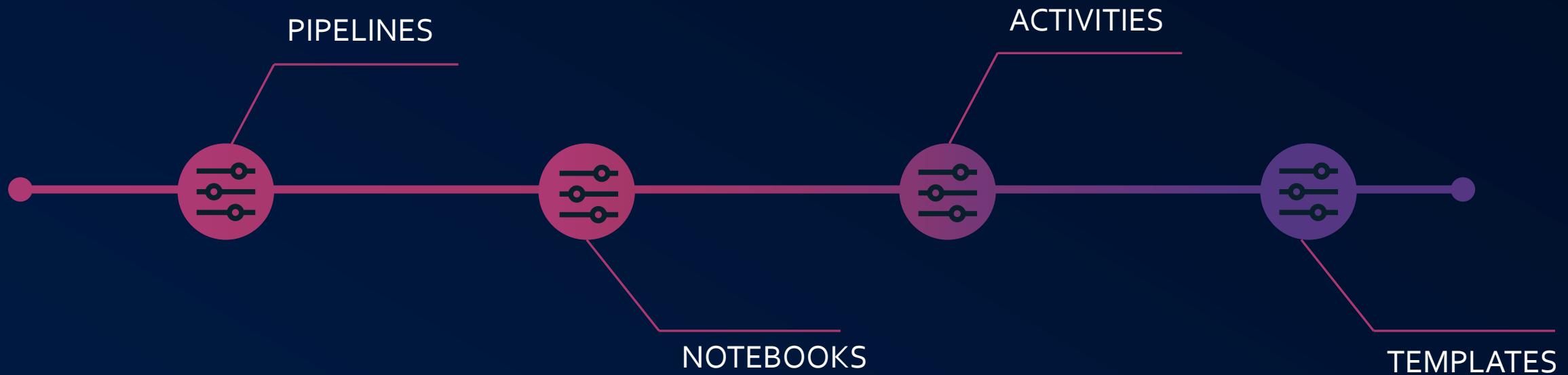


OneLake



Microsoft Purview

Parameters



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

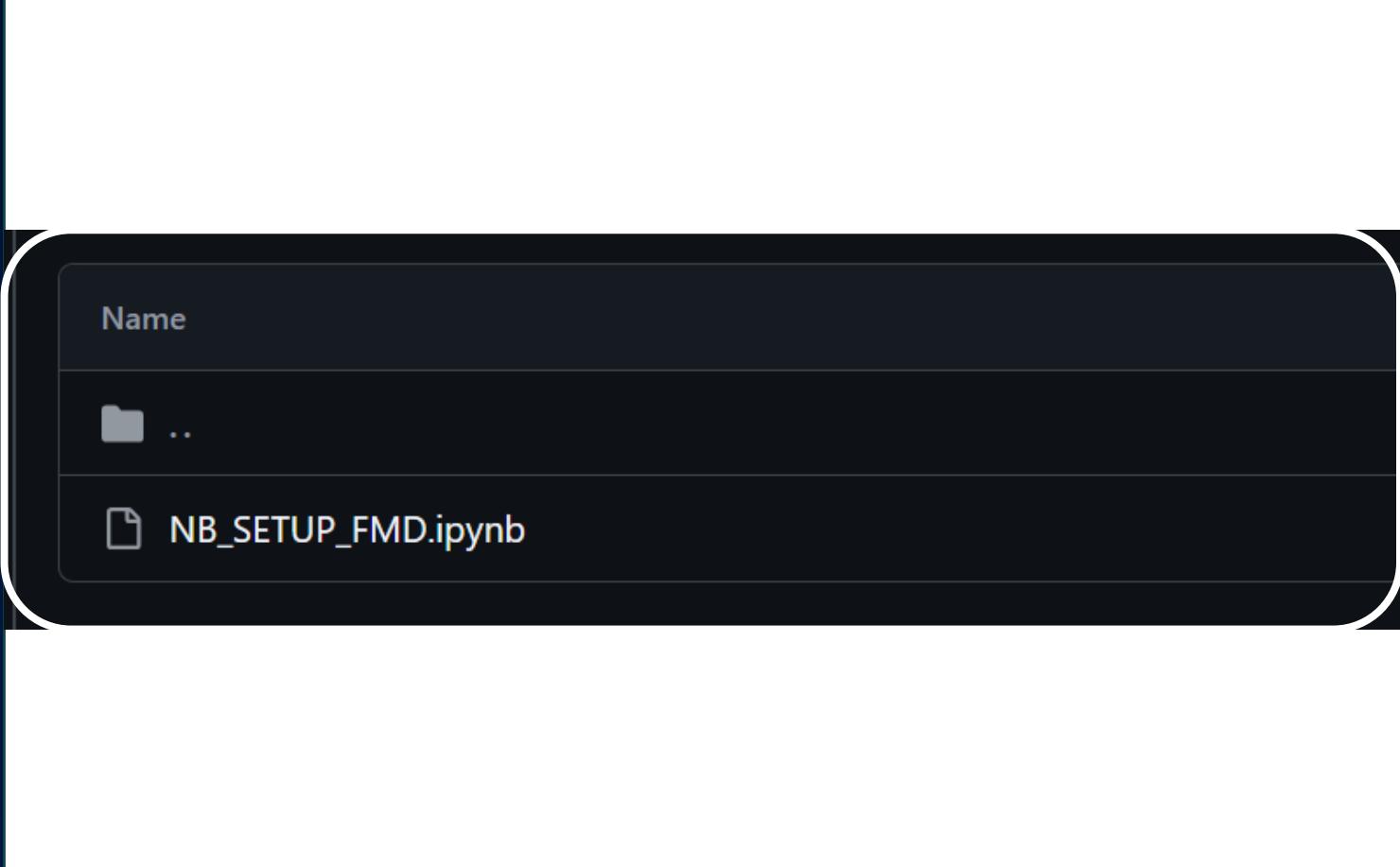
Modern data platforms demand agility, scalability, and consistency. FMD simplifies these challenges by enabling:

- **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



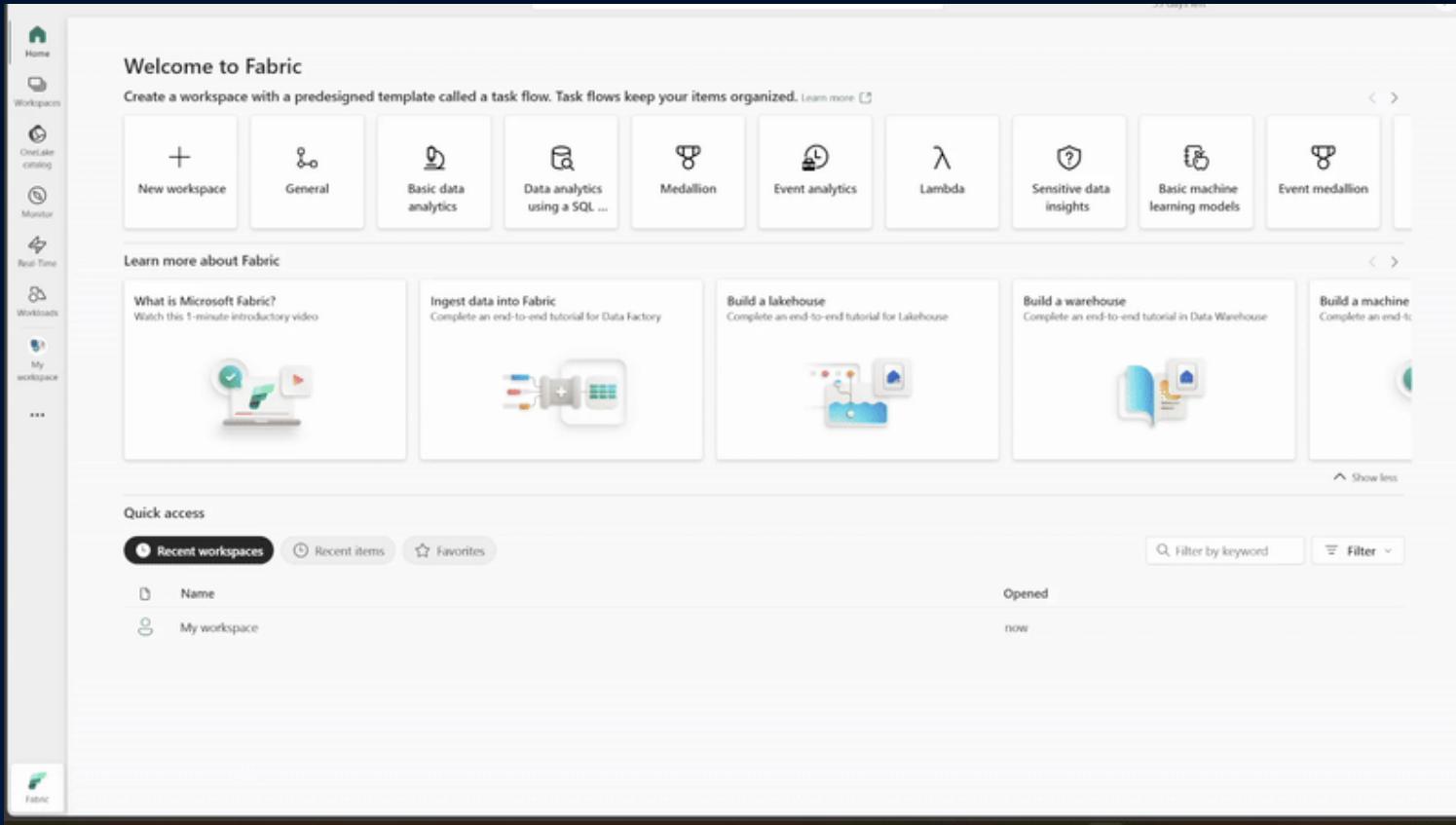
Fabric Metadata Driven Framework

- Download Notebook from Github



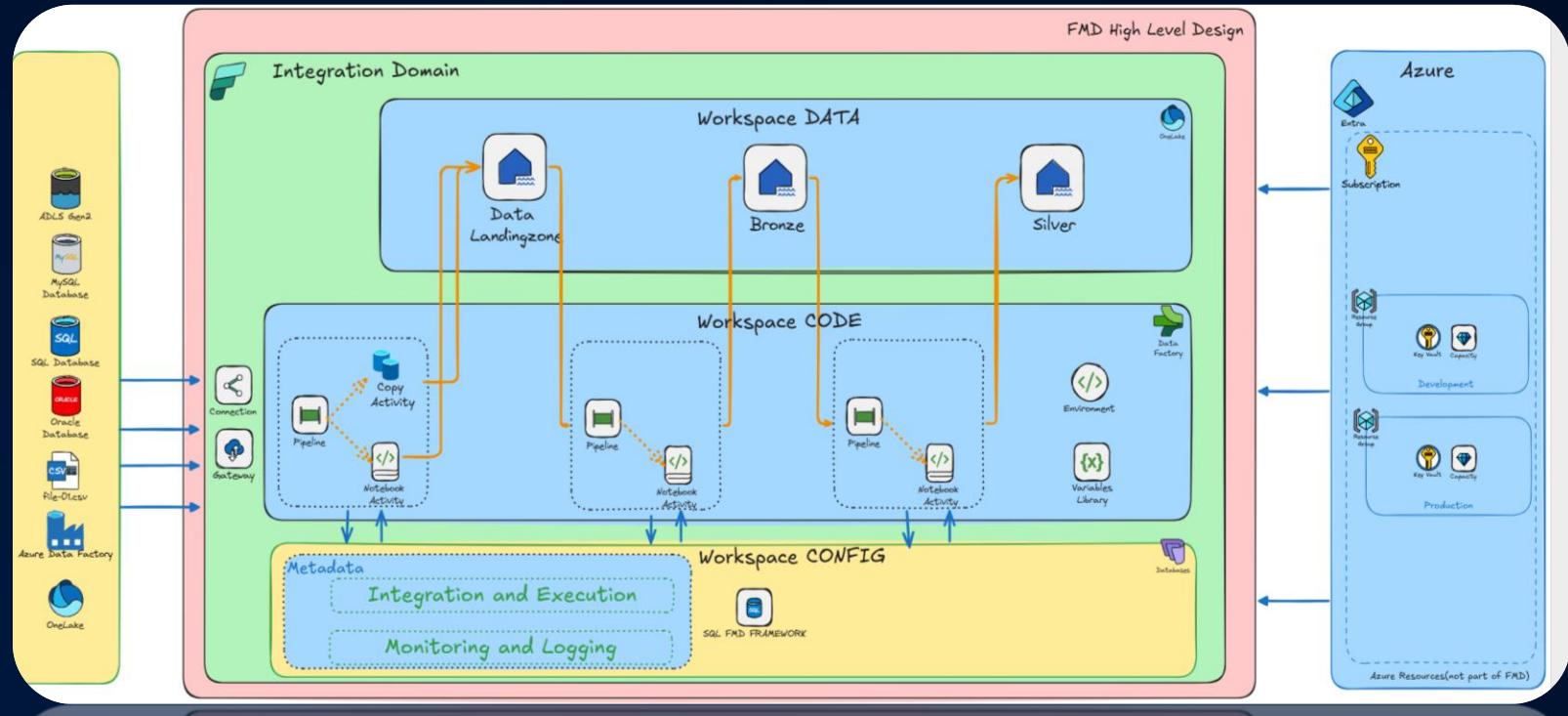
Fabric Metadata Driven Framework

- 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

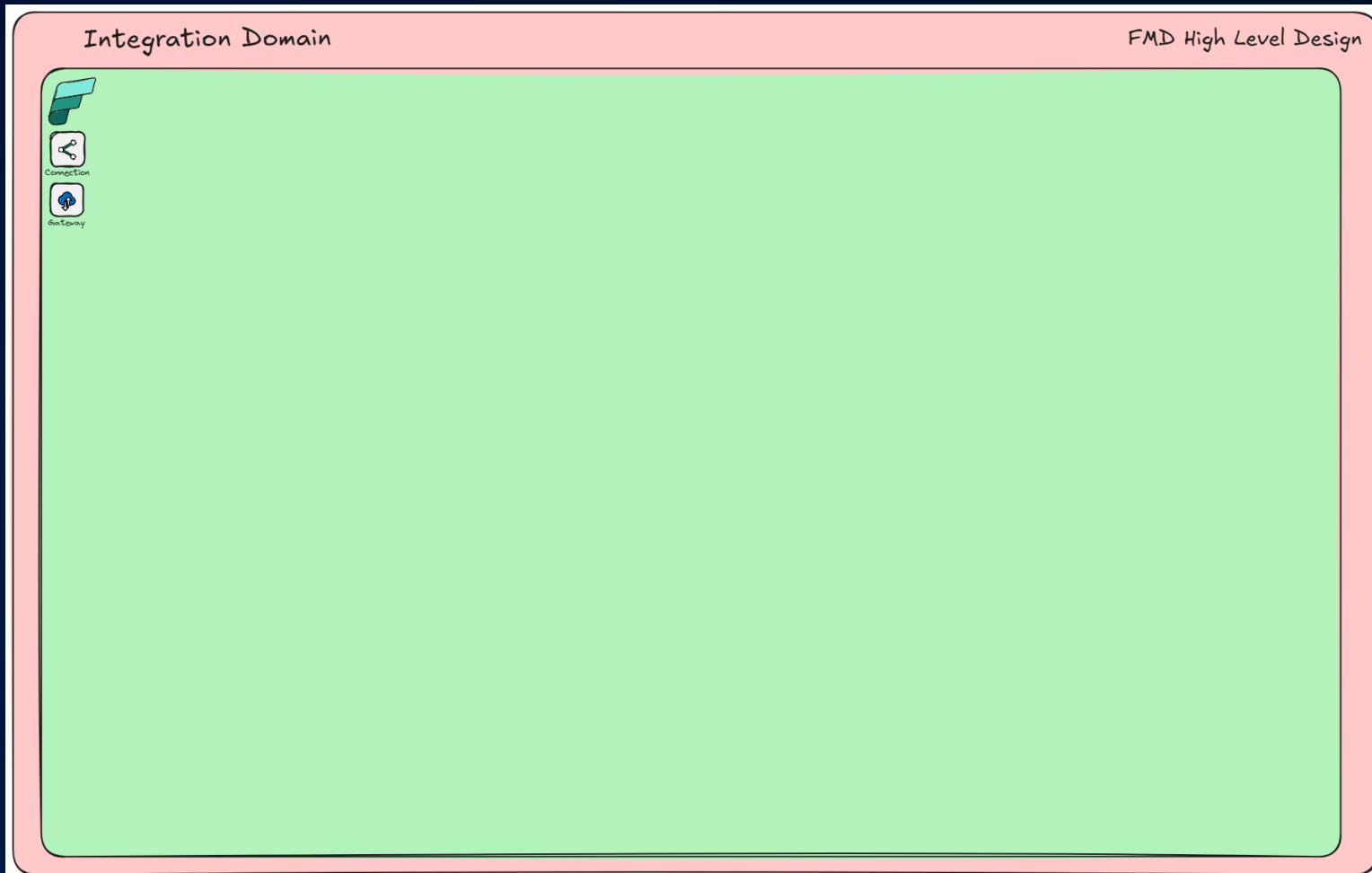


Data Platform Engineering

Data Platform Engineering

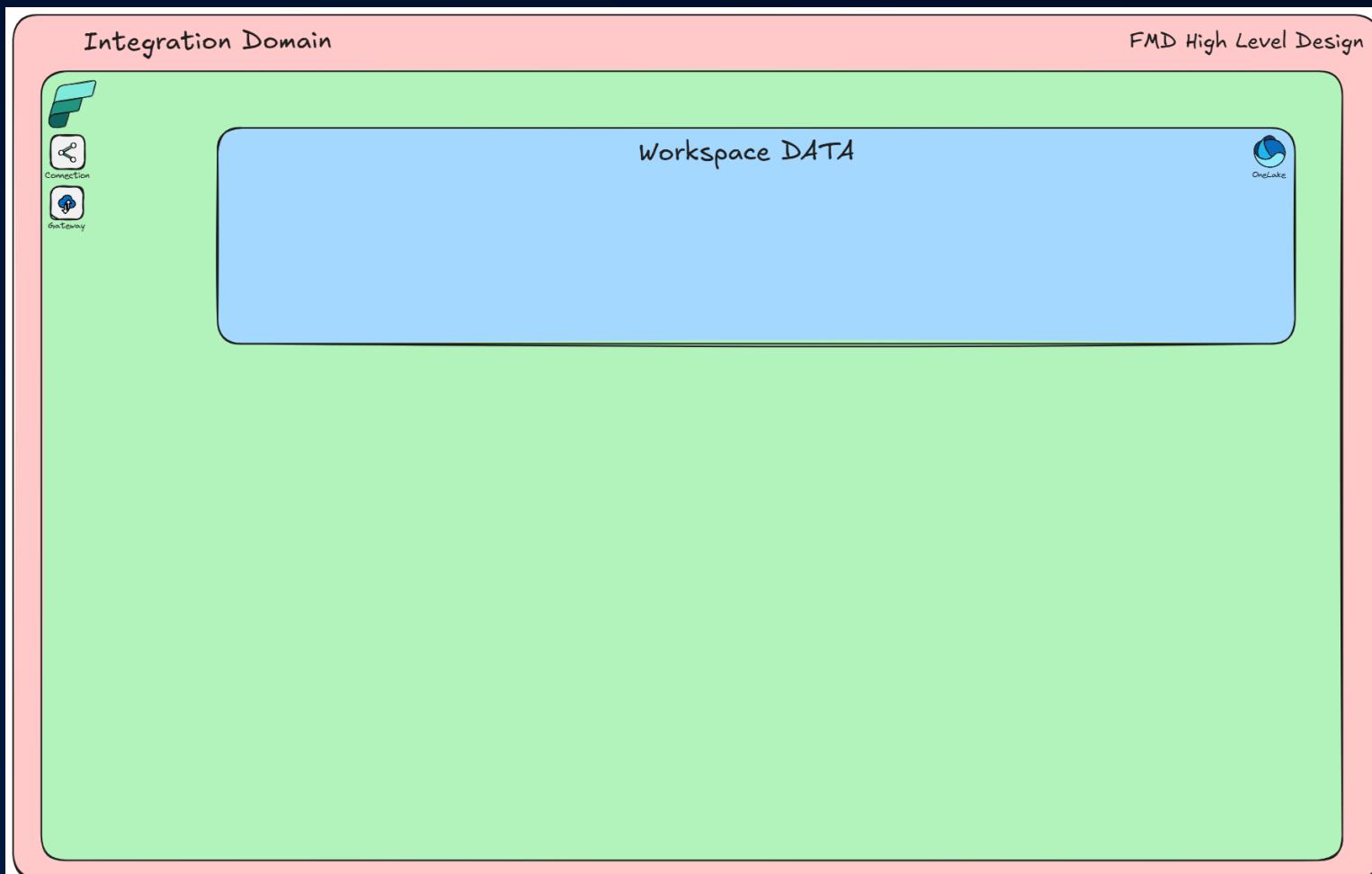


- Azure subscription
- Fabric Capacity (Trial)
- Key Vault optional
- Fabric Administrator role



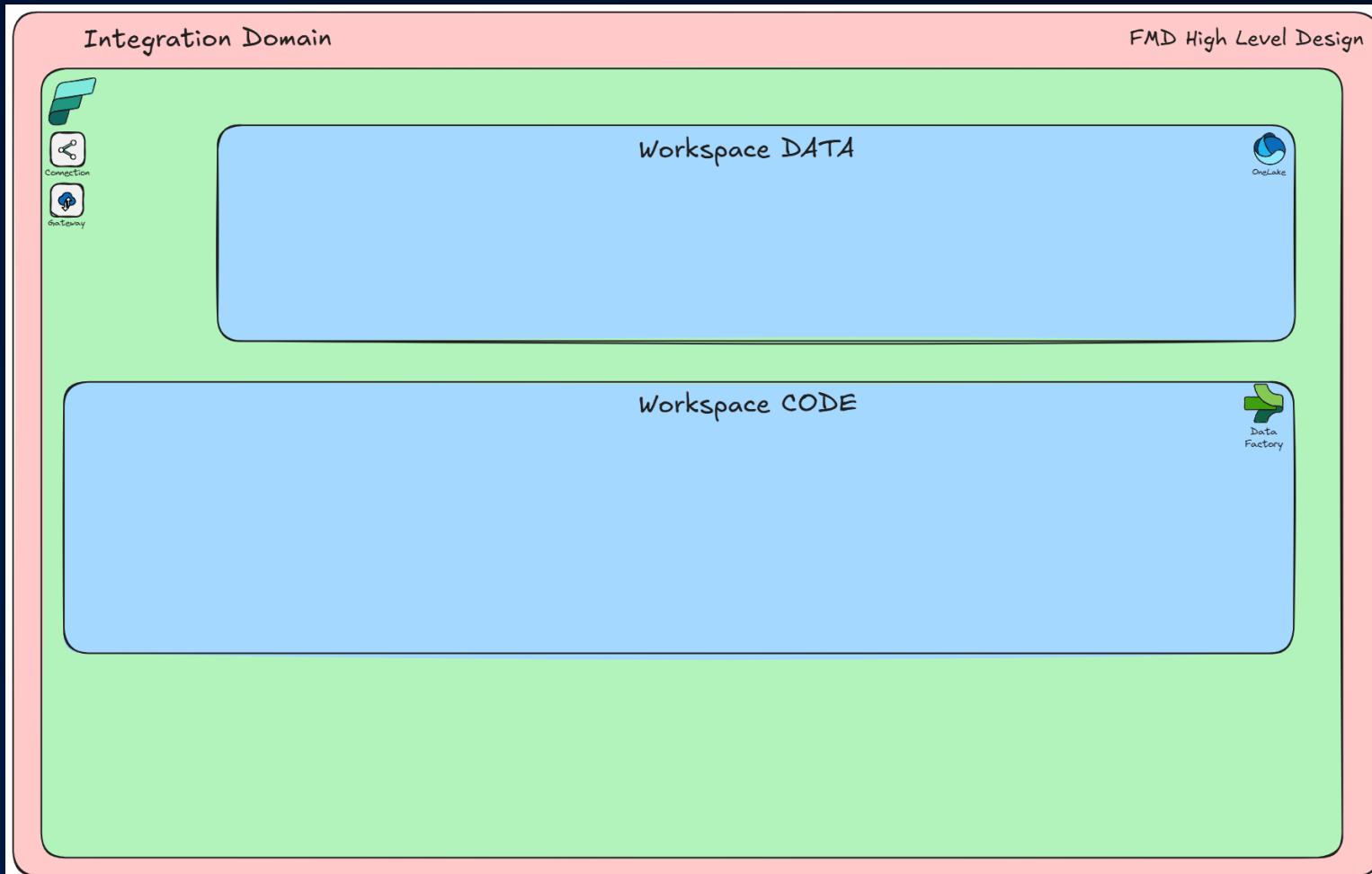
Data Platform Engineering

- Workspace DATA:
- Lakehouses



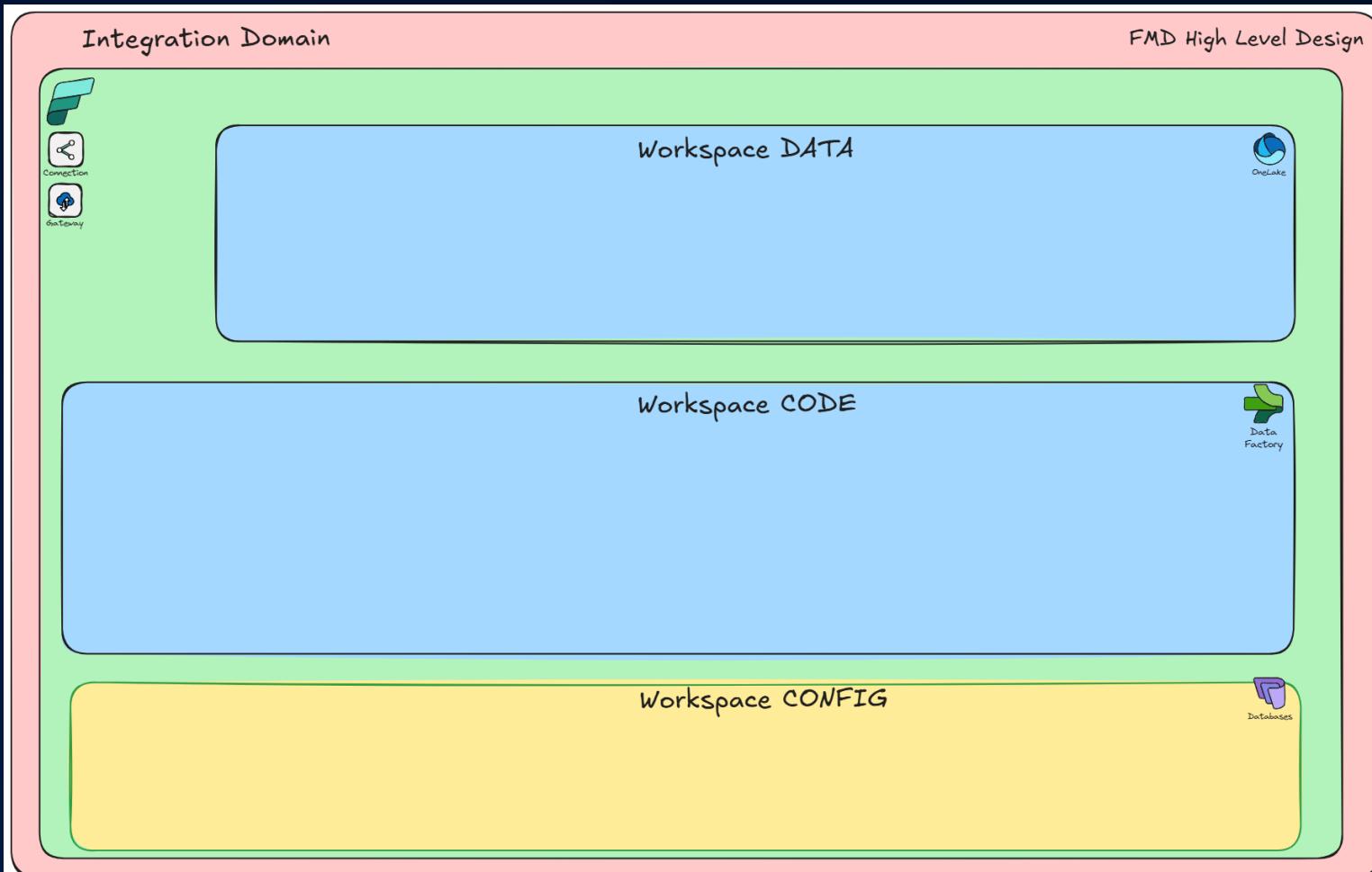
Data Platform Engineering

- Workspace CODE
- Data Pipelines
- Notebooks
- Environments
- Variable Libraries



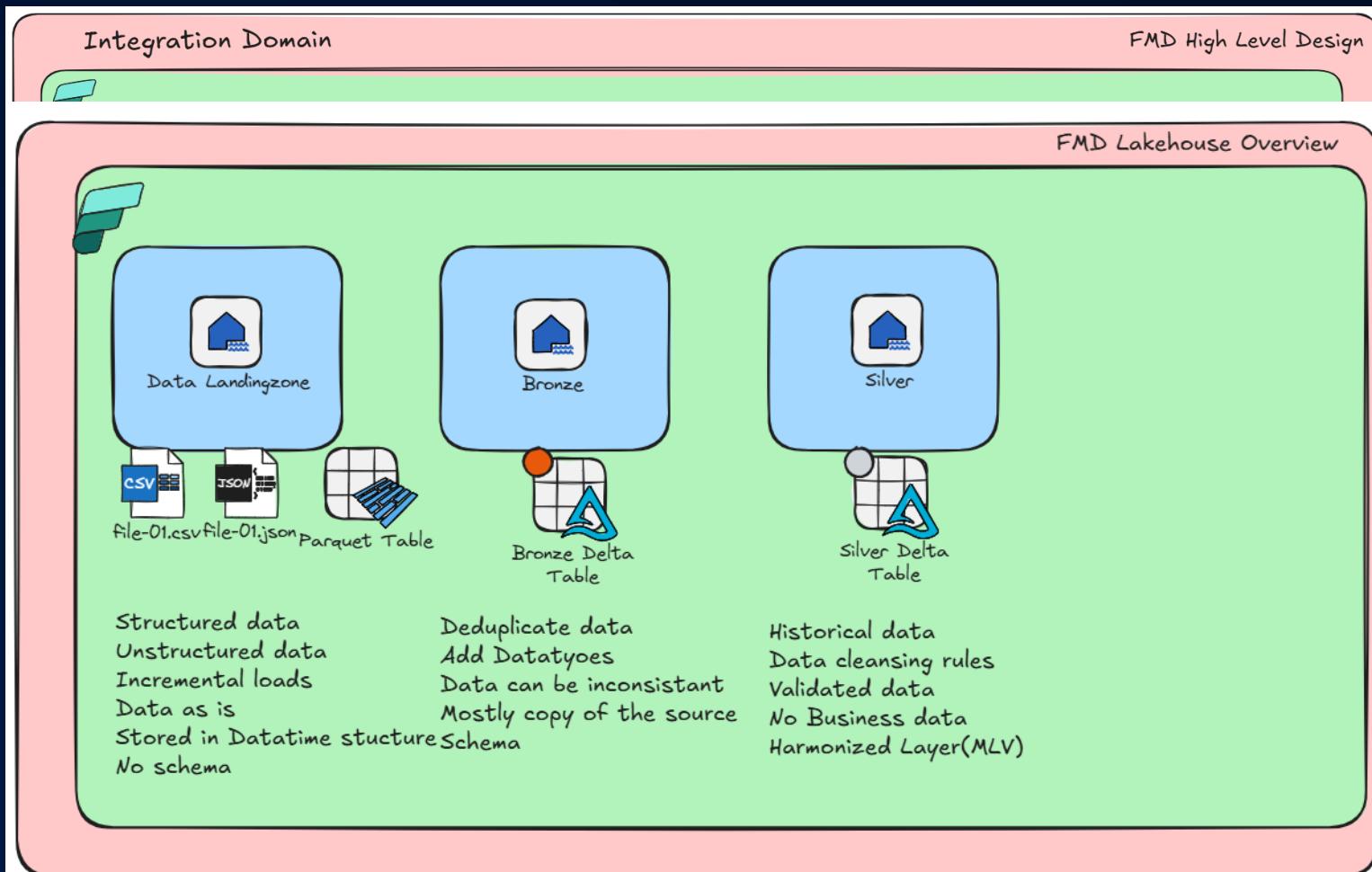
Data Platform Engineering

- Workspace CONFIG
- Fabric database
 - Metadata database



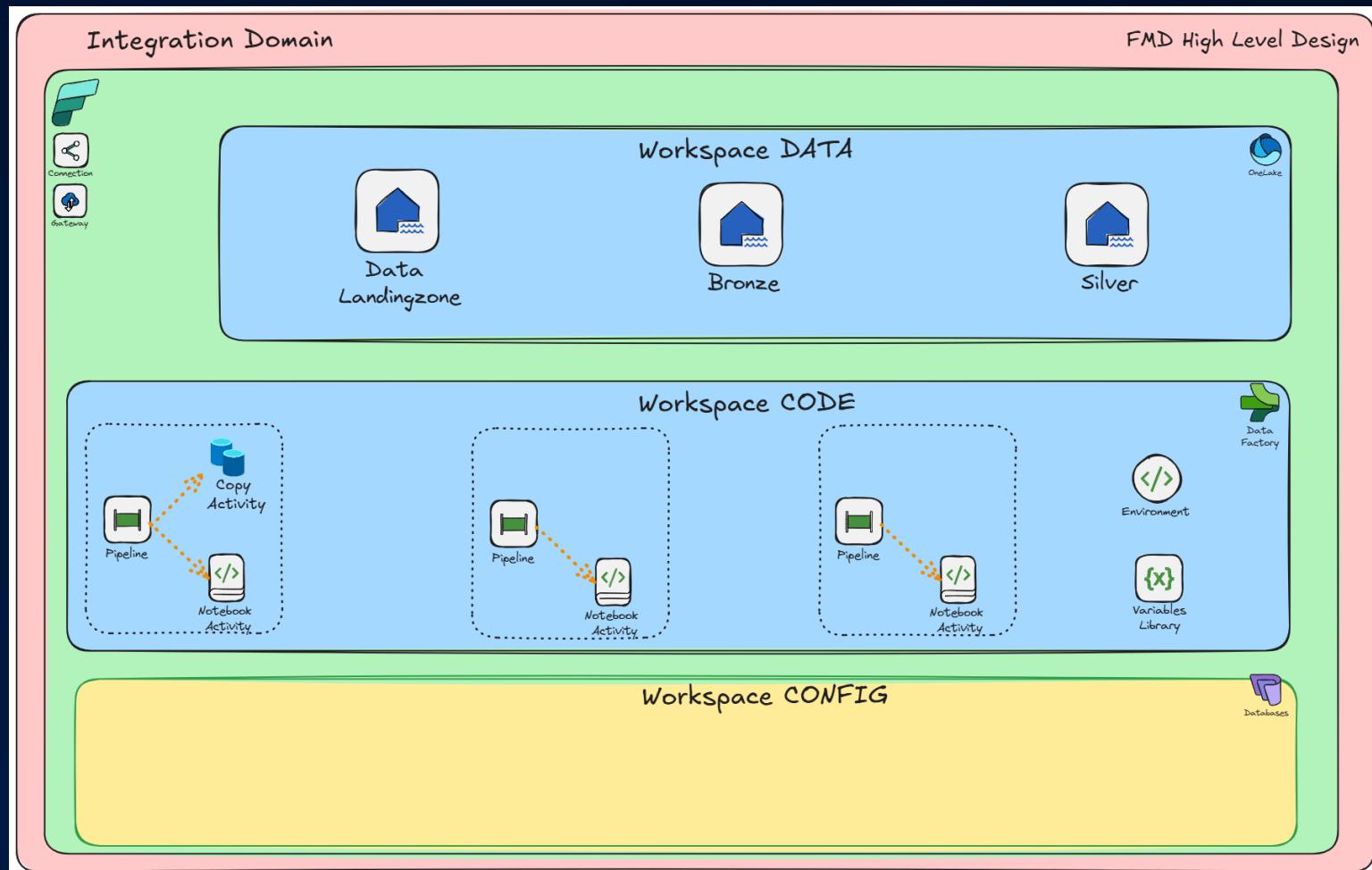
Item deployment

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



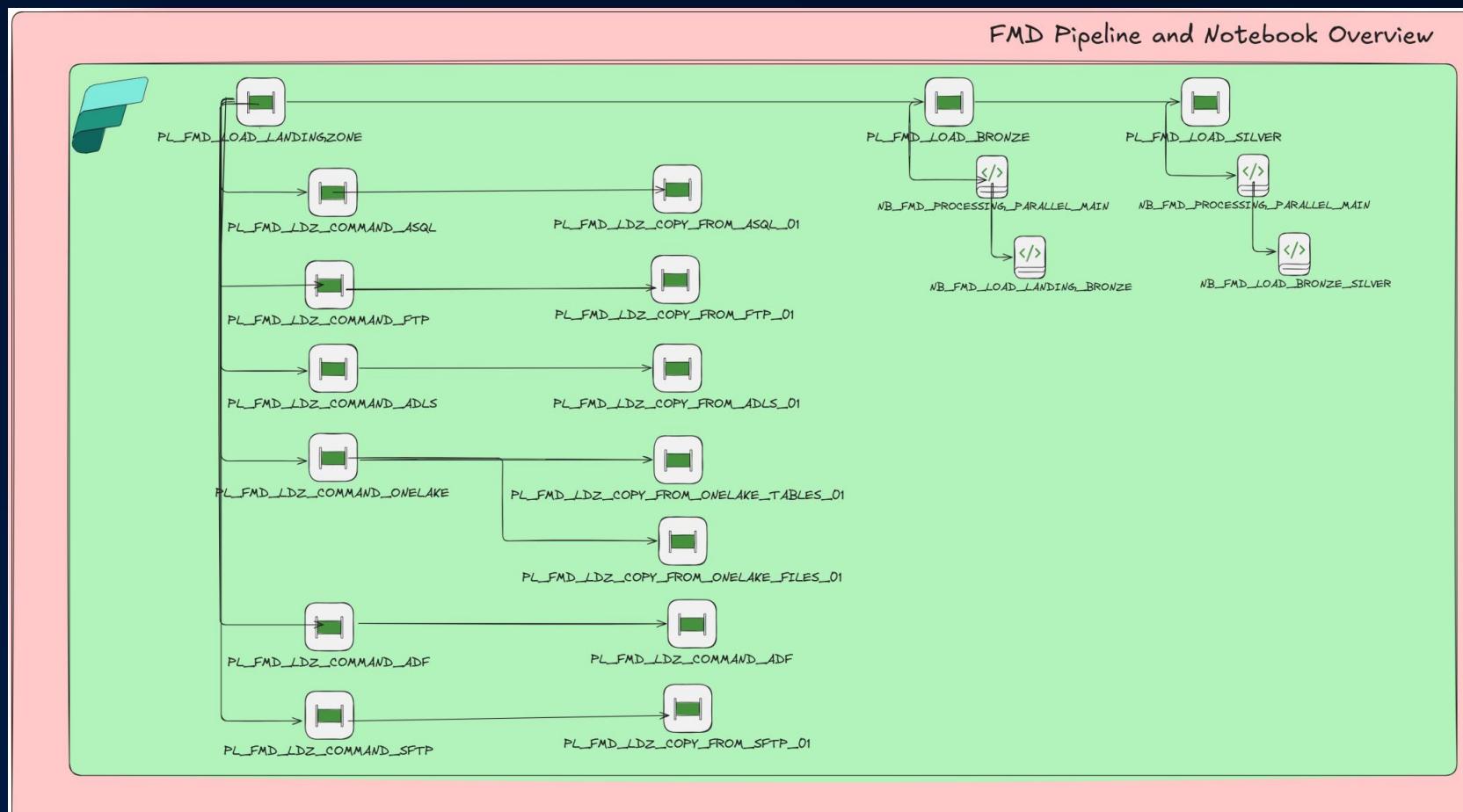
Item deployment

- Variable Library
- Environment



Item deployment

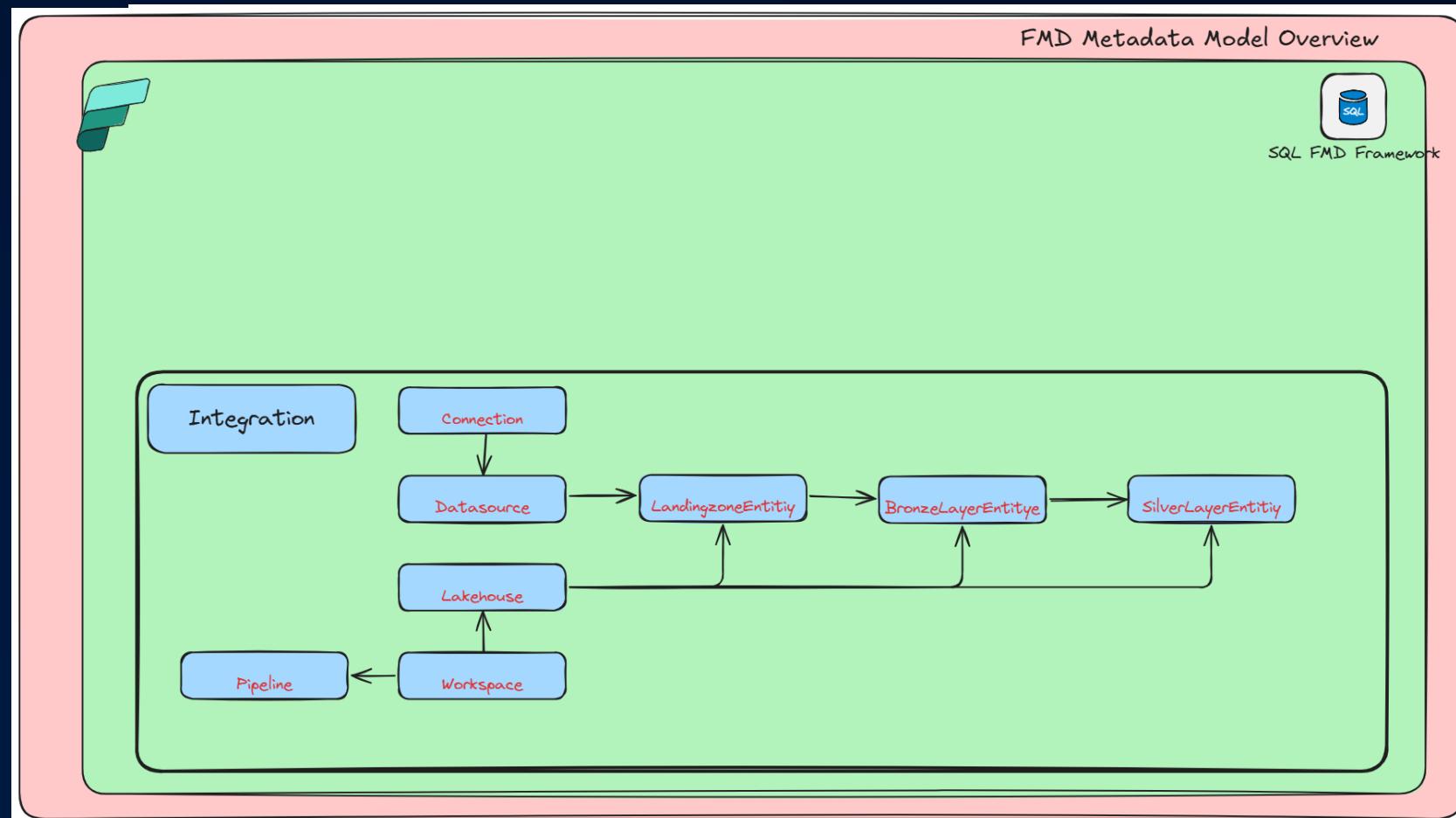
- Variable Library
 - Environment
 - Data Pipelines
 - Notebooks



Data Integration

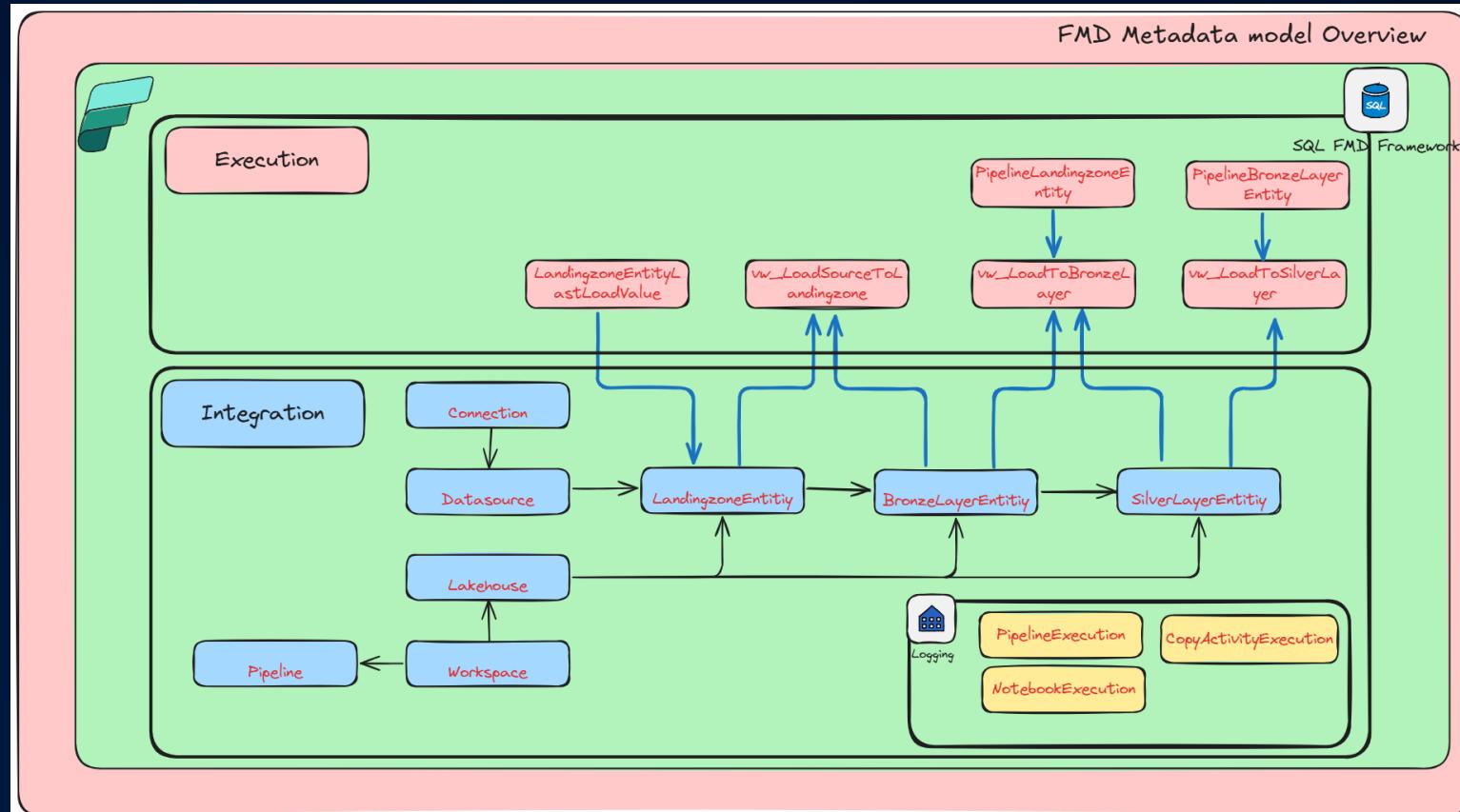
Data Integration

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



Data Integration

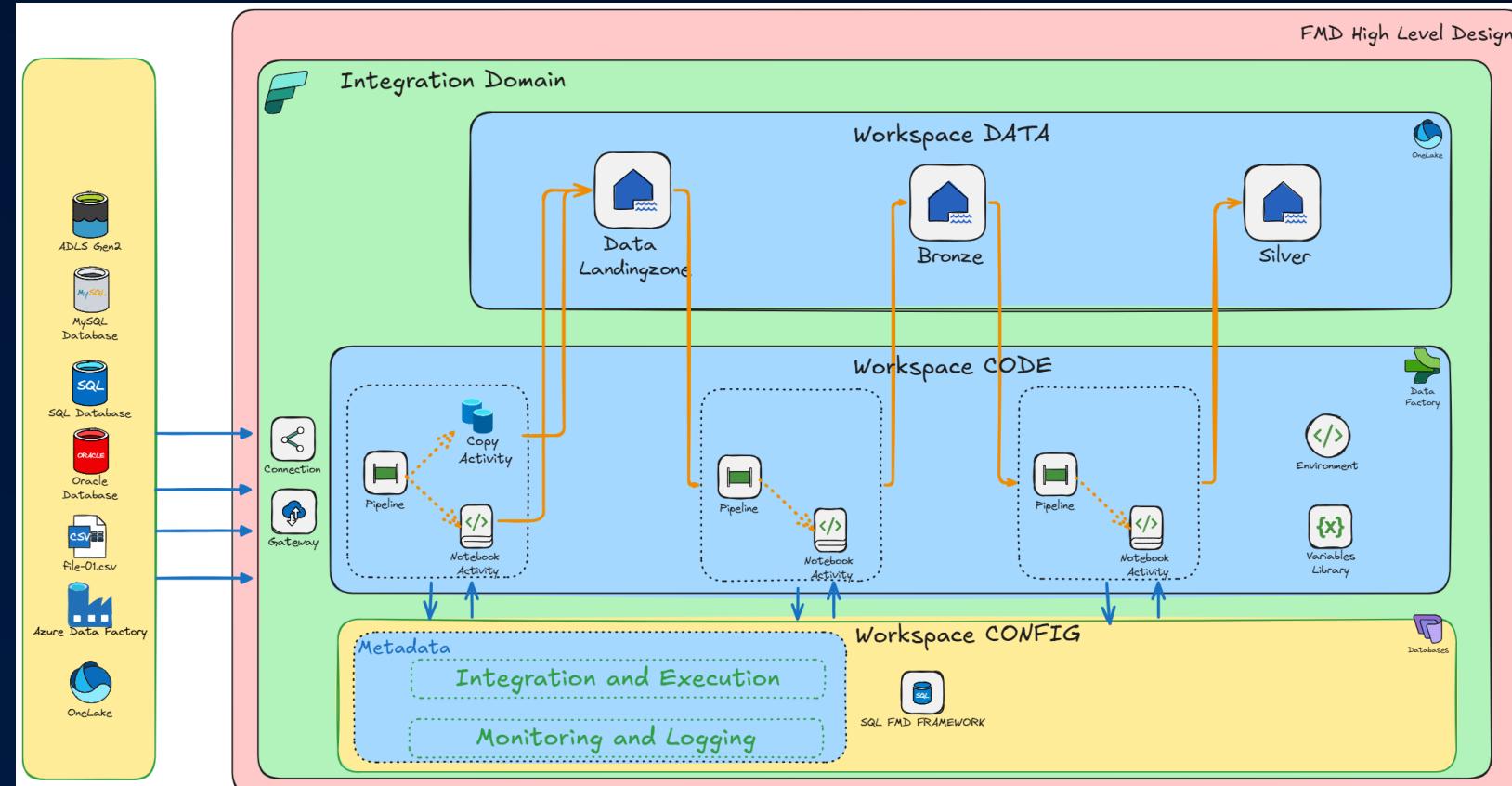
- Execution
 - Tables
 - Views
 - Stored Procedures
- Logging
 - Tables
 - Views
 - Stored Procedures



Data Sources

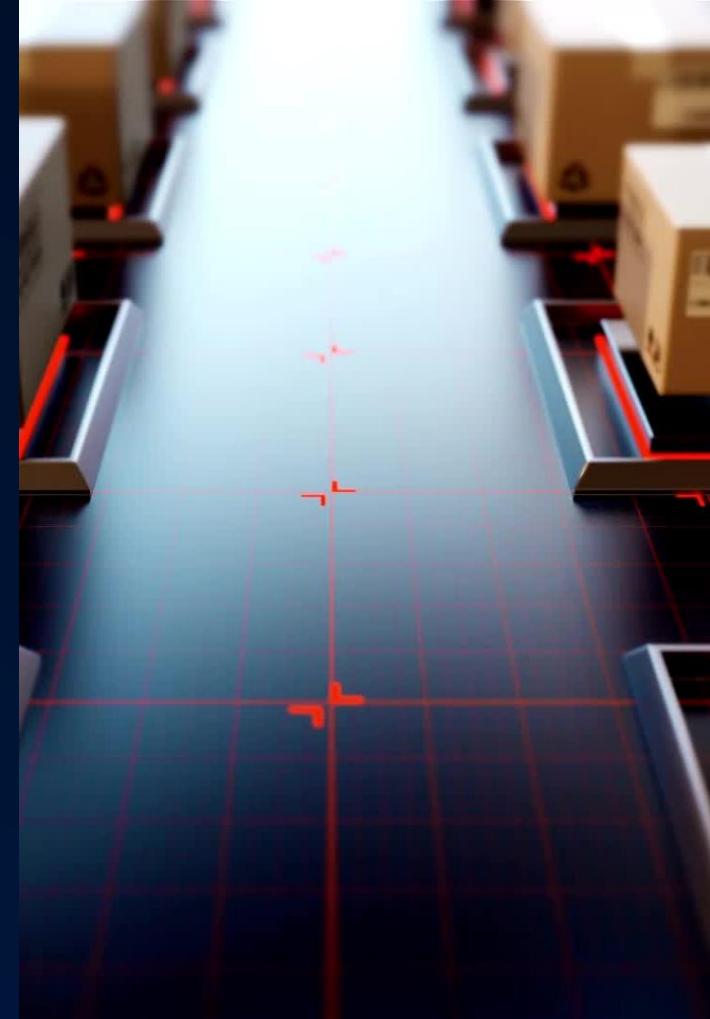
- SQL Server
- Data Lake Gen 2
- Onelake Tables
- Onelake Files
- FTP / SFTP
- Oracle
- Custom Notebooks(API)
- ADF (Connect to ADF pipeline)

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



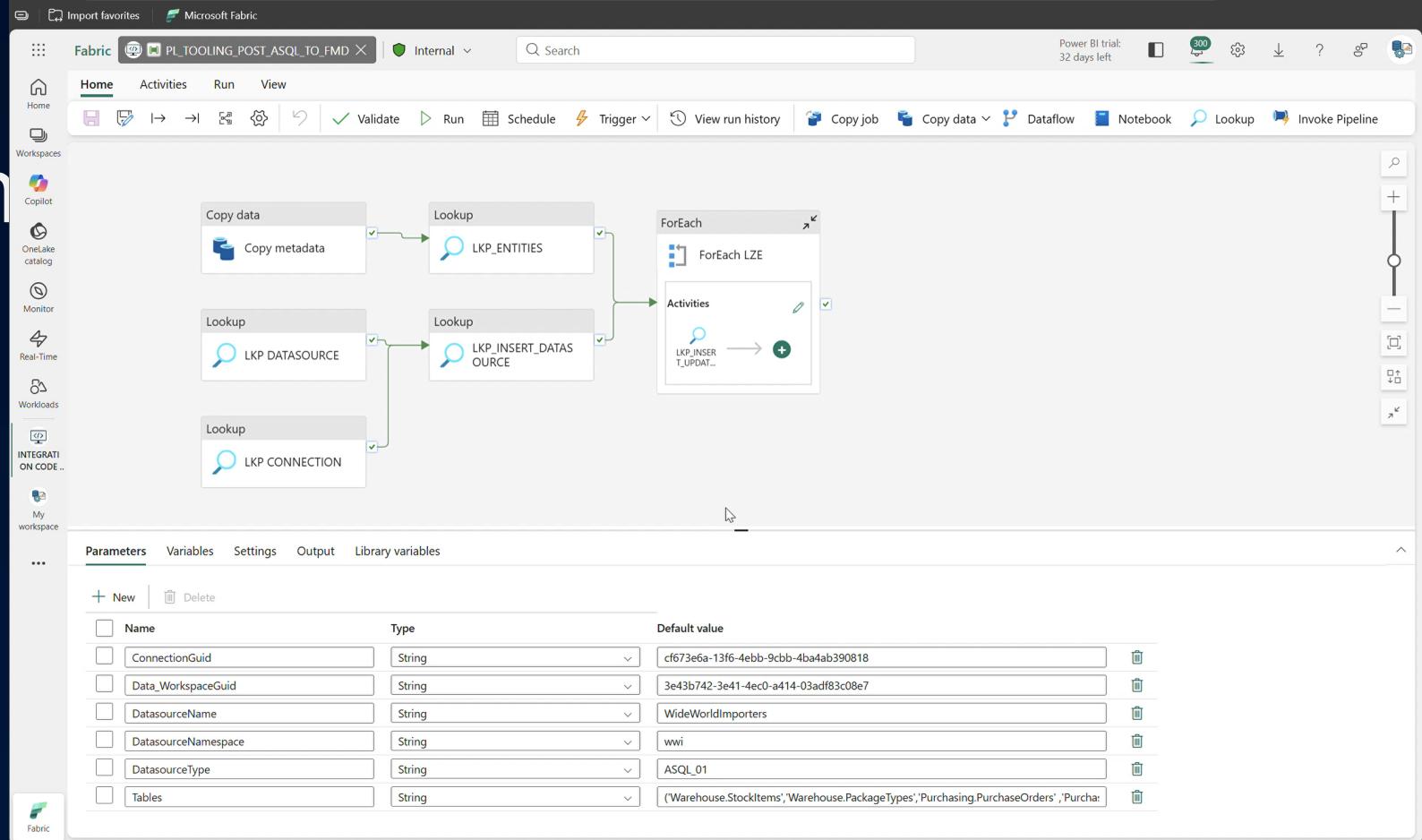
How to get data easily into Fabric Metadata Driven data model

- Metadata driven ingestion Pipeline
- Manual



Metadata driven

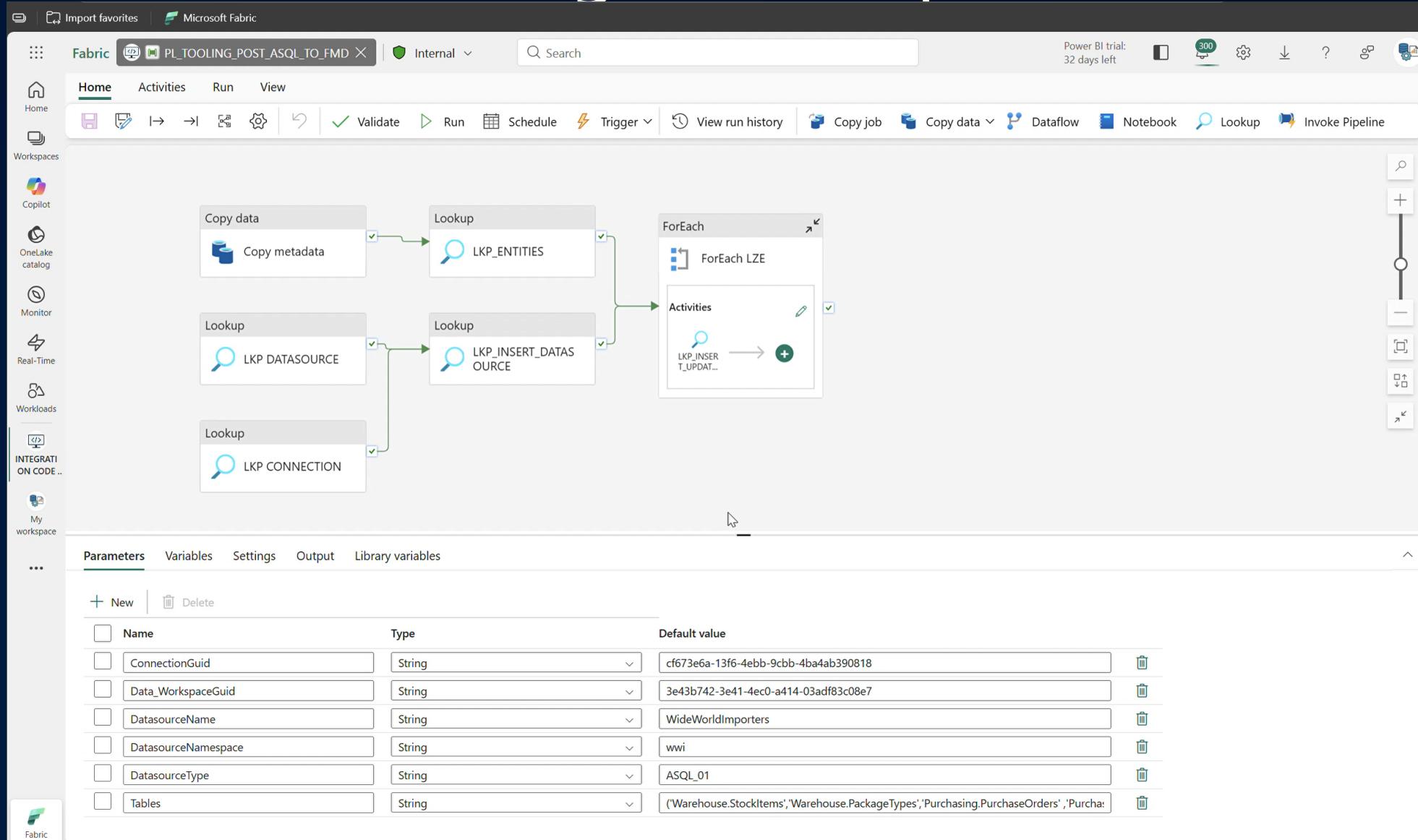
- ConnectionGuid
- DatasourceName
- Namespace
- DatasourceType
- Tables(schema+table)



New | Delete

	Name	Type	Default value
	ConnectionGuid	S...	cf673e6a-13f6-4ebb-9ccb-4ba4ab390818
	Data_WorkspaceGuid	S...	3e43b742-3e41-4ec0-a414-03adf83c08e7
	DatasourceName	S...	WideWorldImporters
	DatasourceNamespace	S...	wwi
	DatasourceType	S...	ASQL_01
	Tables	S...	('Warehouse.StockItems','Warehouse.PackageTypes','Purchasing.PurchaseOrders', 'Purchasing.PurchaseOrderLines', 'Sales.vCustomers','Sales.CustomerCategories','Sales.Orc')

Metadata driven ingestion Pipeline



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='location 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

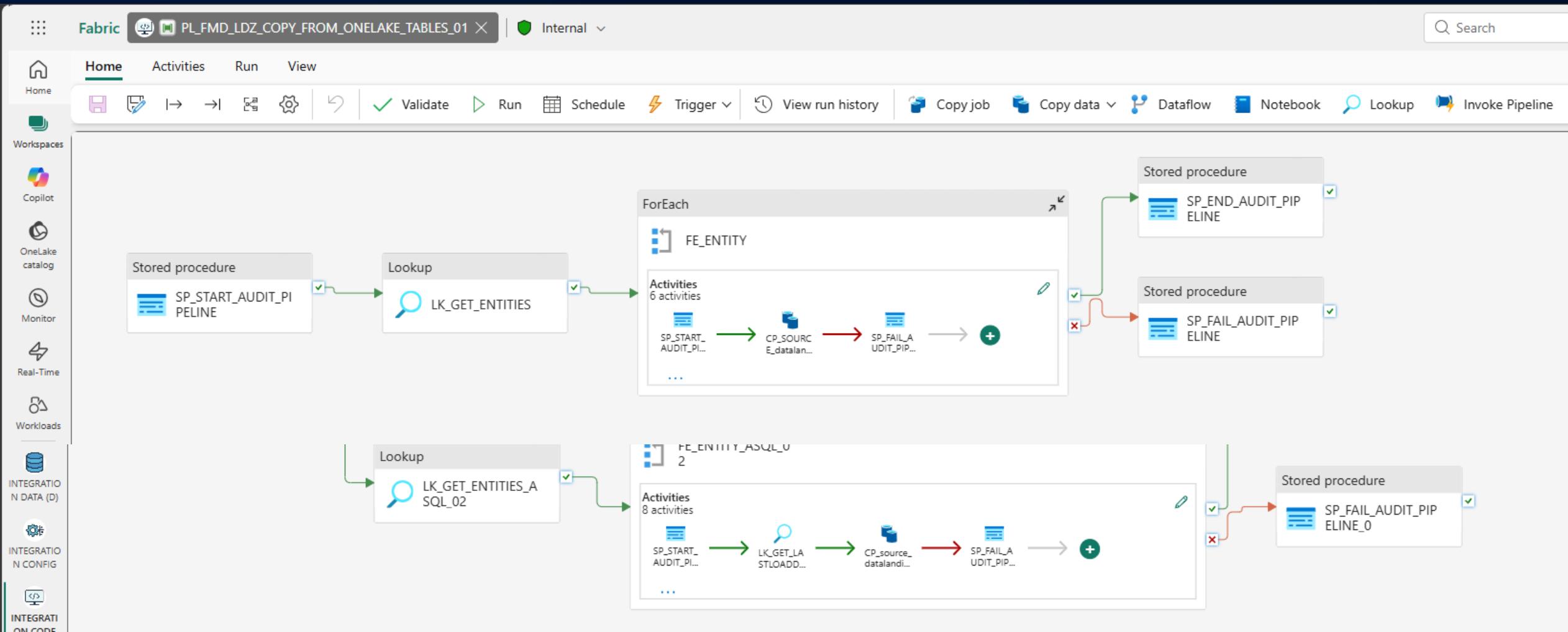
Add Datasource

Add Entity

Demo

Data Processing

Data Processing (source to Data Landingzone)



Data Processing (Data LDZ to Bronze/Silver)

Fabric  PL_FMD_LOAD_BRONZE X | Internal

Home Activities Run View      

Parameters

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

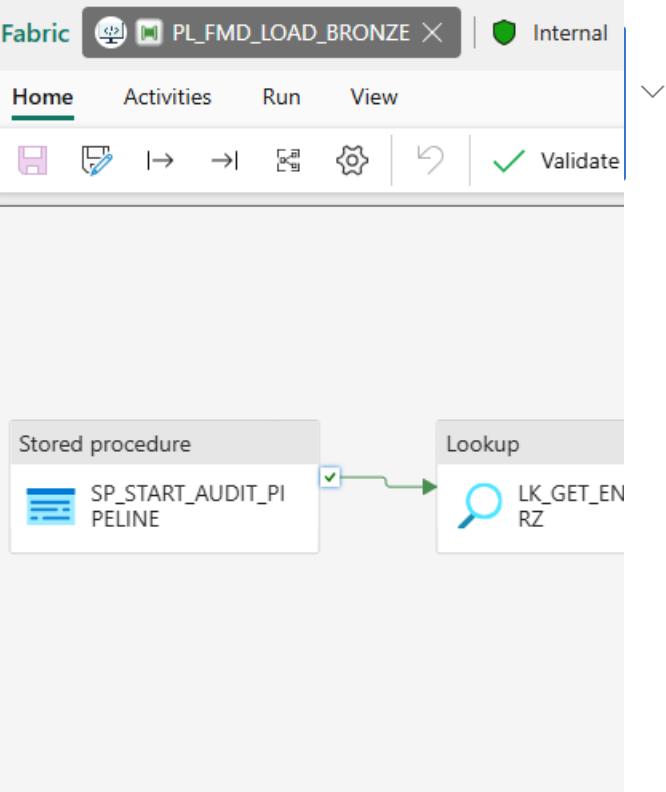
PySpark (Python) ▾

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

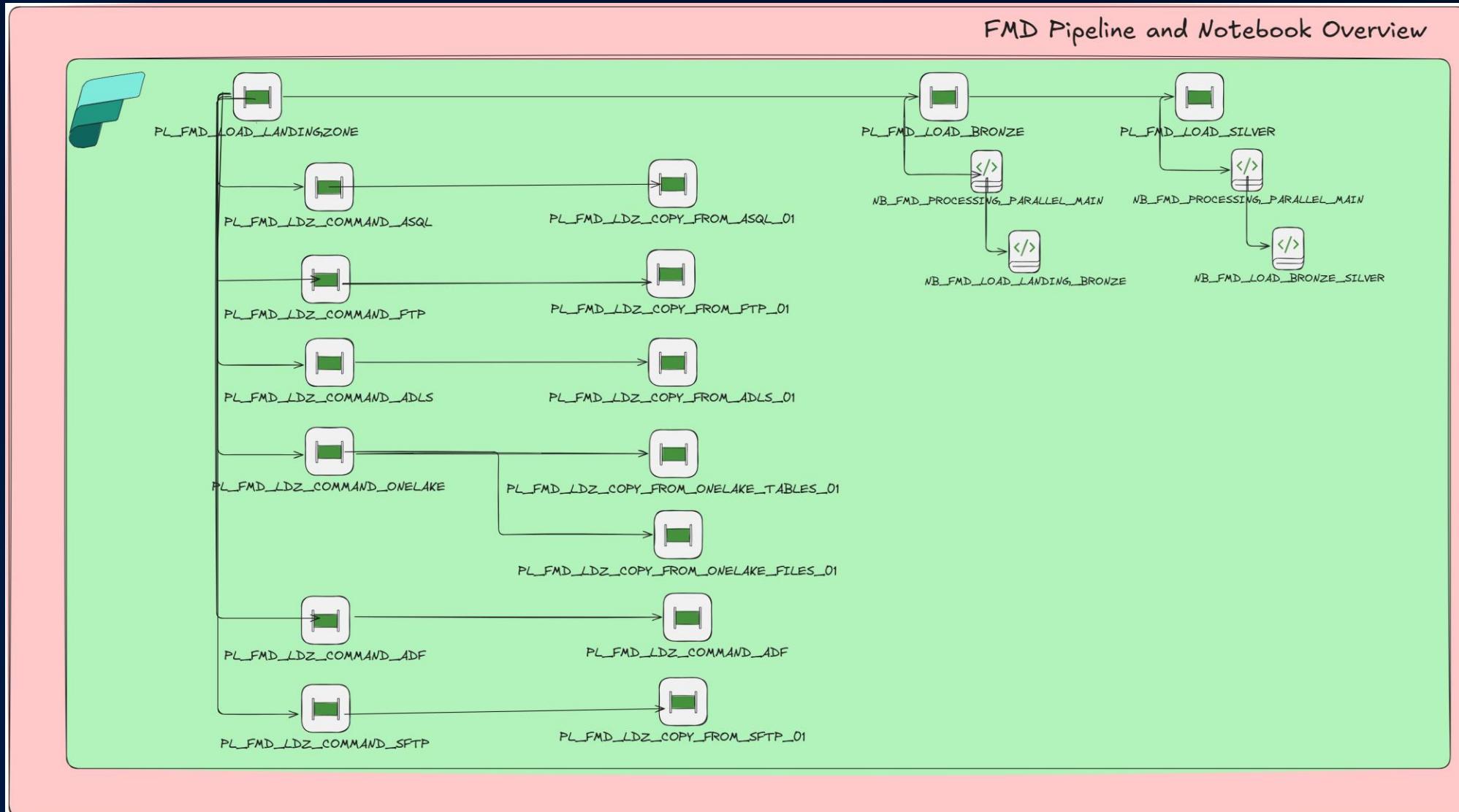
PySpark (Python) ▾

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

...



Data Processing



Data Processing Cleansing rules (BRZ/SLV)

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

```

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>SilverLayerEntityId</th> <th>CleansingRules</th> </tr> </thead> <tbody> <tr> <td>123</td> <td>{"function": "normalize_text", "columns": "CustomerName", "parameters": {"case": "upper"}}</td> </tr> <tr> <td>24</td> <td></td> </tr> </tbody> </table>	SilverLayerEntityId	CleansingRules	123	{"function": "normalize_text", "columns": "CustomerName", "parameters": {"case": "upper"}}	24		
SilverLayerEntityId	CleansingRules							
123	{"function": "normalize_text", "columns": "CustomerName", "parameters": {"case": "upper"}}							
24								

```

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

```

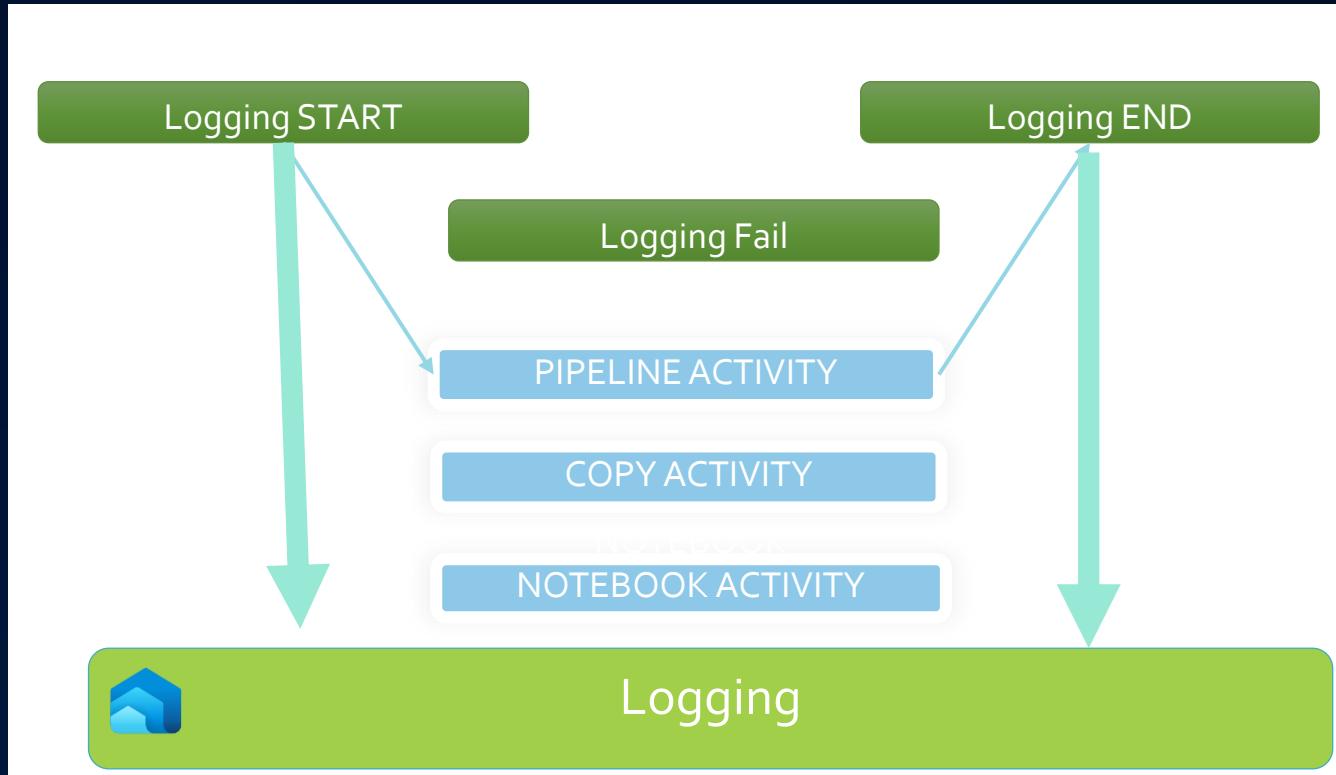
df_norm = normalize_text(
    df,
    columns=["name"],
    args={"case": "title", "collapse_spaces": False}
)
  
```

Demo

Data Monitoring

Data Monitoring

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



Data Monitoring

- Add Information about pipelines
- Adding System Variables

The screenshot shows the 'Pipeline expression builder' interface. At the top, there's a header 'Pipeline expression builder' with a back arrow icon. Below it is a sub-header 'Add dynamic content below using any combination of expressions, functions and system variables.' A code editor window contains the expression '@pipeline().Pipeline'. Below the code editor are tabs for 'Parameters', 'System variables' (which is selected), 'Functions', and 'Variables'. A search bar is also present. To the left of the search bar, there's a table titled 'Stored procedure parameters' with columns 'Name' and 'Type'. The table lists various pipeline-related variables:

Name	Type
LogData	String
LogType	String
PipelineGuid	Guid
PipelineName	String
PipelineParameters	String
PipelineParentRunGuid	Guid
PipelineRunGuid	Guid
TriggerGuid	Guid
TriggerTime	DateTime
TriggerType	String
WorkspaceGuid	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 Ex...

Workspace ID

ID of the workspace the pipeline run is running within

Data Monitoring

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

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  
}
```



A screenshot of a JSON API response. The response is a single object with two main keys: "status" and "result". The "status" key has a value of "Succeeded". The "result" key contains detailed information about the pipeline run, including its ID, status, session ID, spark pool, error status, and metadata. A green rounded rectangle highlights the "result" object. Two orange arrows point from the "exitValue" field in the Pipeline expression builder code to the "exitValue" field in the JSON response. The "exitValue" field in the JSON response is part of a larger object under the "highConcurrencyModeStatus" key, which also includes "highConcurrencyModeStatus" and "exitValue". The "exitValue" field itself is a list of objects, each containing a "TableName" and an "exitVal". The "exitVal" field contains a JSON string starting with "\\"{ 'CopyOutput':".

```
{"status": "Succeeded",  
 "result": {  
   "runId": "10883eal-cac8-4d1b-92d3-03f8ca451054",  
   "runStatus": "Succeeded",  
   "sessionId": "188bc363-7872-453e-9369-8a18c115f6e8",  
   "sparkPool": "1acc1c9e-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: 10883eal-cac8-4d1b-92d3-03f8ca451054\",  
   \"SparkMonitoringURL\": \"https://app.powerbi.com/workloads/de-ds/sparkmonitor/fcd42\",  
   \"executionDuration\": 266  
}]
```

Data Monitoring

PipelineExecution X

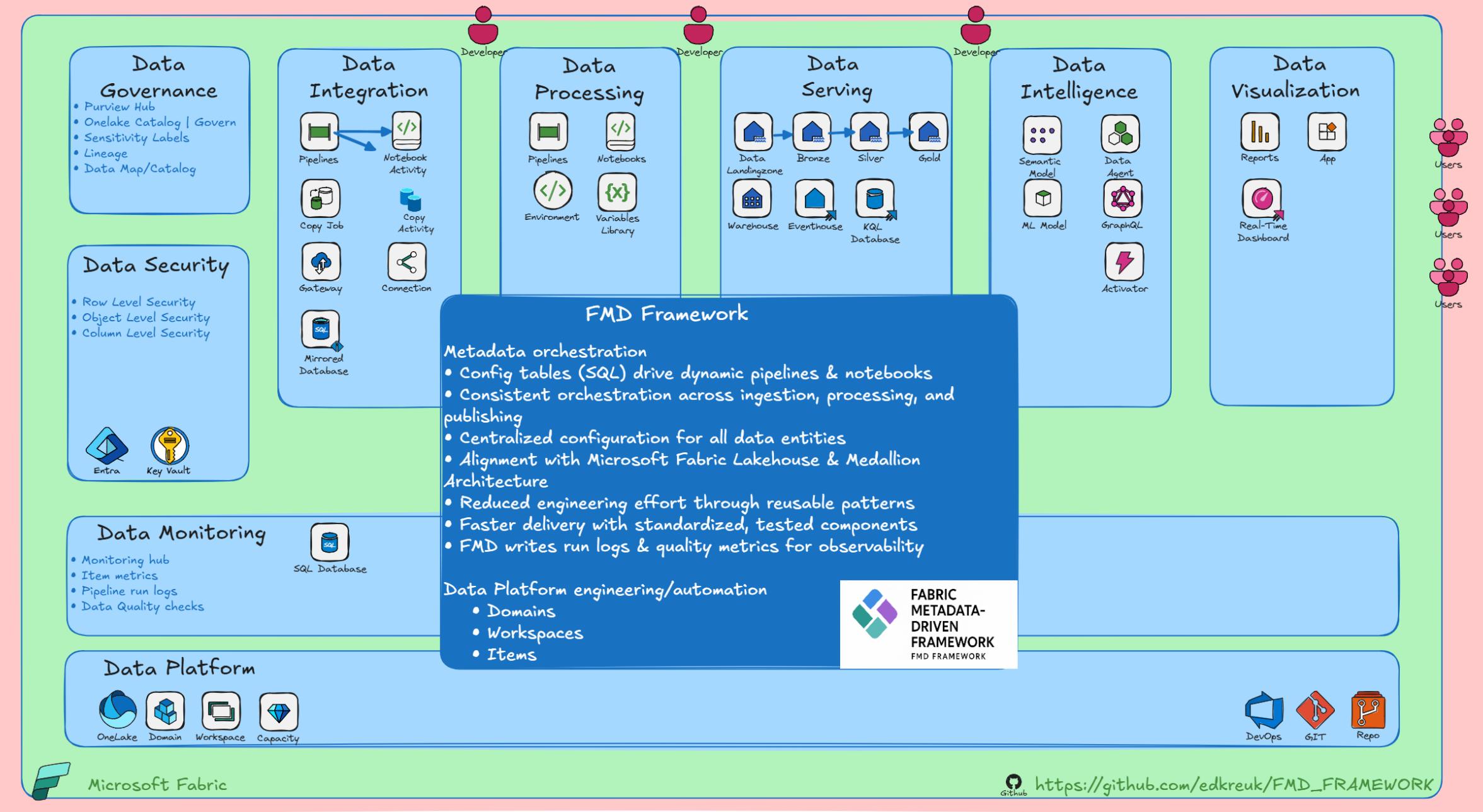
Data preview - PipelineExecution Showing 1000 rows Search

#	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...								
15	1ACC1C9E-B8DS...	DS38F02C-ECDE...	NULL		B8530D51-F8CS...	PL_FMD_LDZ_CO...	InvoiceLines		Manual	800DB4CE-CA74...	2025-03-17 18:1...	EndCopyActivity	2025-03-17 18:2...	{ "Action" : "En...								
16	1ACC1C9E-B8DS...	DS38F02C-ECDE...	NULL		B8530D51-F8CS...	PL_FMD_LDZ_CO...	OrderLines		Manual	800DB4CE-CA74...	2025-03-17 18:1...	EndCopyActivity	2025-03-17 18:2...	{ "Action" : "En...								
17	1ACC1C9E-B8DS...	DS38F02C-ECDE...	NULL		B8530D51-F8CS...	PL_FMD_LDZ_CO...	Orders		Manual	800DB4CE-CA74...	2025-03-17 18:1...	EndCopyActivity	2025-03-17 18:2...	{ "Action" : "En...								
18	1ACC1C9E-B8DS...	DS38F02C-ECDE...	NULL		B8530D51-F8CS...	PL_FMD_LDZ_CO...	BuyingGroups		Manual	800DB4CE-CA74...	2025-03-17 18:1...	EndCopyActivity	2025-03-17 18:2...	{ "Action" : "En...								
19	1ACC1C9E-B8DS...	DS38F02C-ECDE...	NULL		B8530D51-F8CS...	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



Key Learnings and Best Practices



Metadata -driven approach is a best practice for managing data pipelines

Medallion Lakehouse architecture is a proven framework for implementing Metadata -driven approach

Start small and gradually expand the Metadata -driven approach across the organization

Nitrogen Control Center



The screenshot shows a web browser window for the InSpark Cloud Management Portal at <https://cmp.inspark.nl/Nitrogen/Information>. The page title is "Cloud Management Portal" and the sub-page title is "Nitrogen > Information". The navigation bar includes "Import favorites", "EDK_NB_PROCESSL...", "Microsoft Fabric", and "Power BI". The top right corner shows the INSPARK logo and the user "Erwin De Kreuk(Labs)".

The left sidebar has a "Domain wizard" section active, showing "DM_WWL_DEMO" under "Domain". Other options include "Overview", "Tenant settings", "Insights", "Landing zone entities", "Bronze entities", "Silver entities", "Gold entities", "Promotions", "Load Planner Groups", and "Entity wizard".

The main content area has tabs: "Overview", "Product Information" (which is selected), "Enrollment", and "Deployment". The "Product Information" tab contains sections for "Nitrogen Explained" and "Key Features".

Nitrogen Explained: Nitrogen is a cutting-edge software solution that harnesses the power of the Medallion architecture to revolutionize the way you manage and leverage data. Designed specifically for the new Microsoft Fabric service, a groundbreaking addition to the Microsoft Azure platform, Nitrogen positions itself at the forefront of the data world's most exciting innovations. With the promise of a SaaS-managed data solution from Microsoft, Nitrogen is poised to elevate your data management to new heights.

Key Features:

1. Medallion Architecture: Nitrogen leverages the robust and proven Medallion architecture to provide a rock-solid foundation for all your data management needs. This architecture is renowned for its reliability, scalability, and efficiency, making Nitrogen a trusted choice for businesses of all sizes.
2. Microsoft Fabric Compatibility: Nitrogen is tailor-made for the new Microsoft Fabric service, making it seamlessly integrated with this groundbreaking platform. With Nitrogen, you can effortlessly harness the capabilities of Microsoft Fabric to optimize your data infrastructure.
3. Next-Generation Data Management: As Microsoft Fabric takes center stage in the data world, Nitrogen enables you to stay ahead of the curve. This SaaS-managed data solution from Microsoft offers unparalleled performance, security, and scalability, ensuring your data is always accessible and protected.
4. Streamlined Operations: Nitrogen simplifies data management by providing a user-friendly interface and a range of powerful tools. It enables your team to focus on extracting insights from your data rather than wrestling with complex data management tasks.
5. Scalability and Performance: Whether you're a small startup or a large enterprise, Nitrogen grows with your business. It offers the scalability and performance required to handle increasing data volumes and

Contact
Would you like to know more?
Feel free to reach out to learn more about this service. Simply complete the "contact me" form, and we'll get back to you as soon as possible.
[Contact me](#)

Feedback



From Manual to Automated: Master Metadata-Driven Design in Fabric

LINKS

FMD Framework



Center

Let's connect
LinkedIn

