MICROSOFT FABRIC

Bas Land

Your first Lakehouse using Microsoft

DATA:Scotland 2024

bridgeall

Tabular Editor

MySQL™

DATAmasterminds

Waterstons

ADVANCING ANALYTICS

quorum

hippo

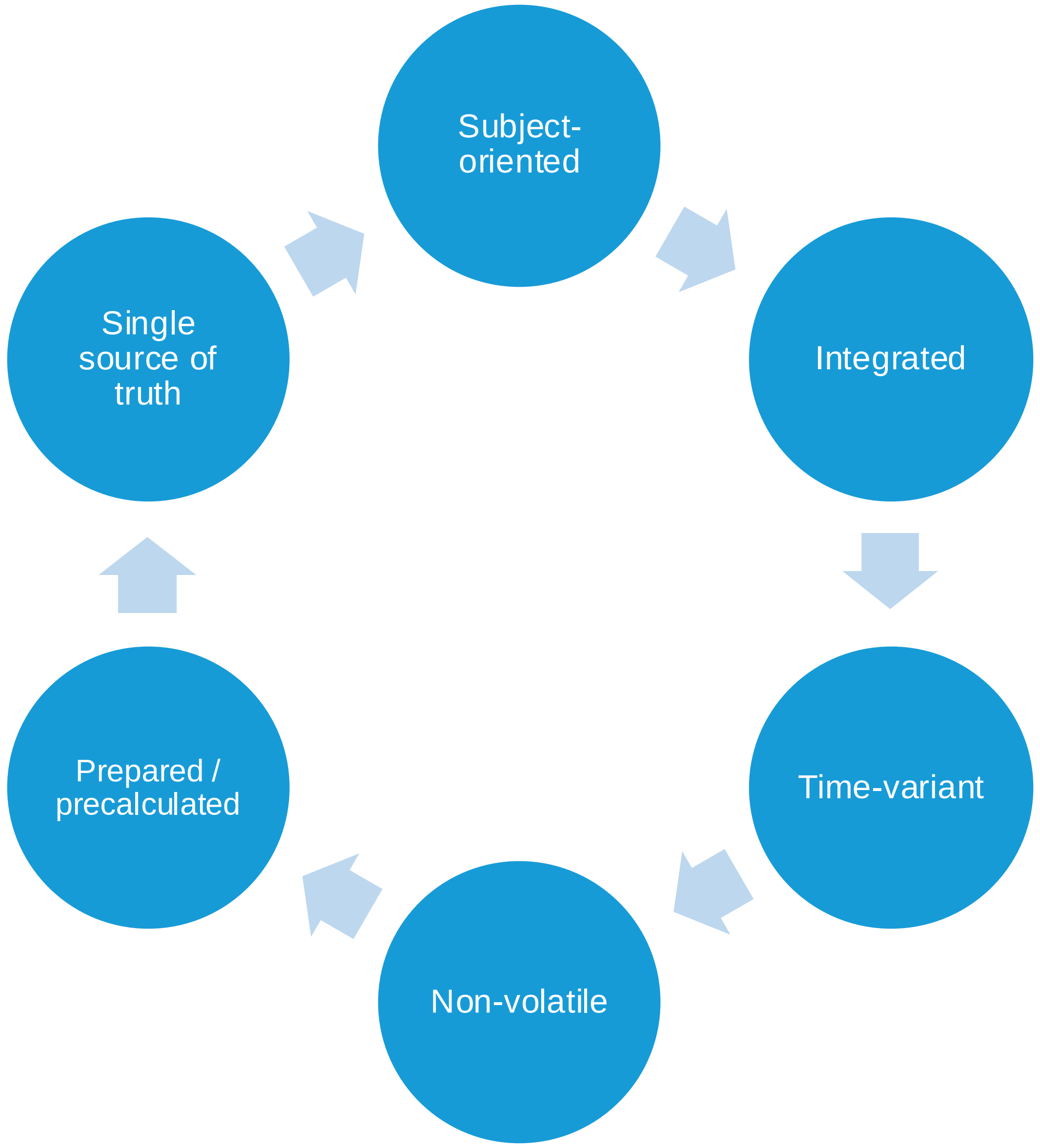
aim

Today

- Core concepts for data warehousing in Fabric
- Your first Fabric Lakehouse:
  - Rulebook
  - ELT process
  - End-to-end solution design with examples
- Takeaways

Core concepts

General design principles



Lakehouse: under the hood

Lakehouse: under the hood

Ingest

Pipelines
Notebooks
Dataflows

Store

OneLake
Data lake blob storage

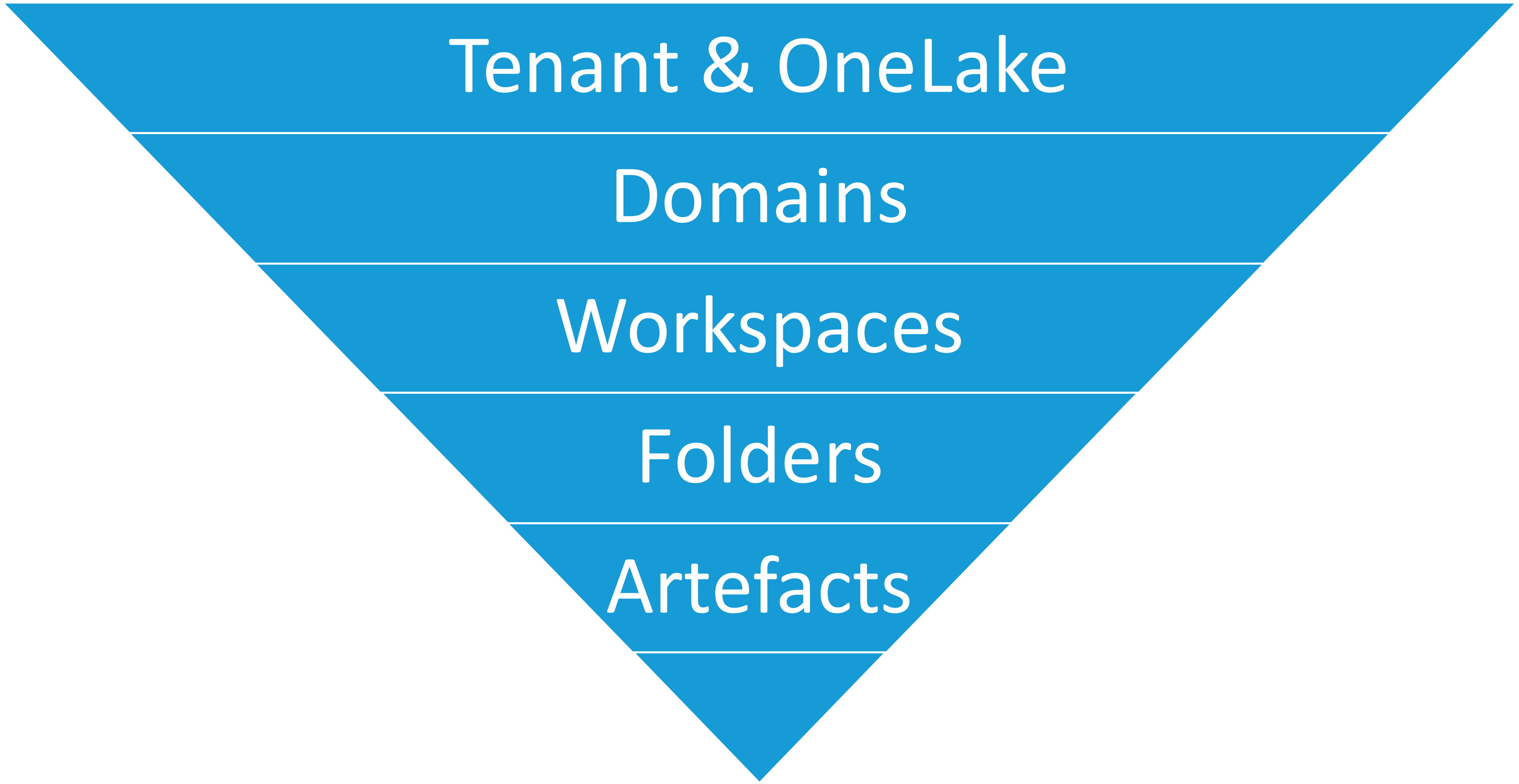
Process

Notebooks
(PySpark)

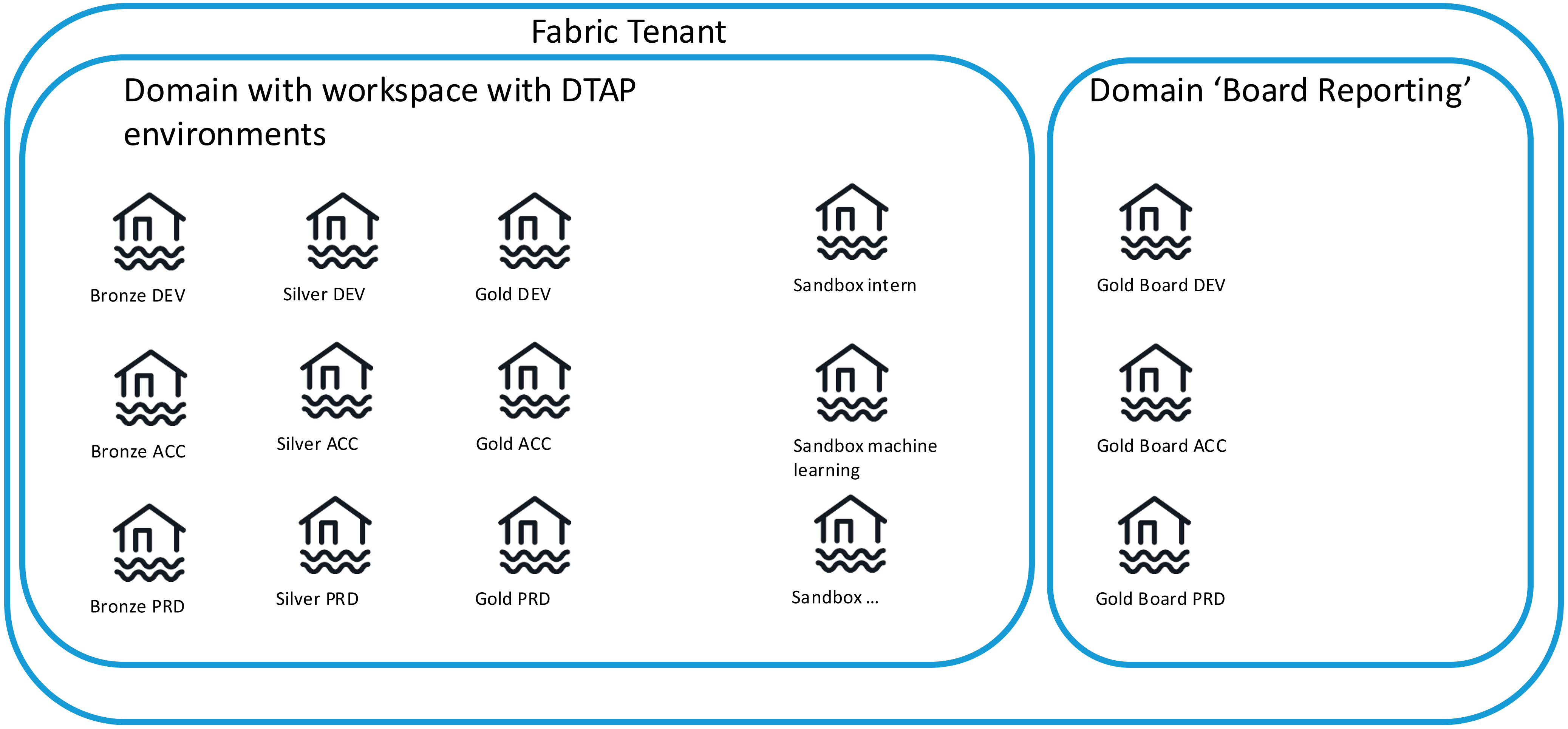
Fabric concepts

1. Tenant design
2. Storage
3. Compute

Fabric: tenant design

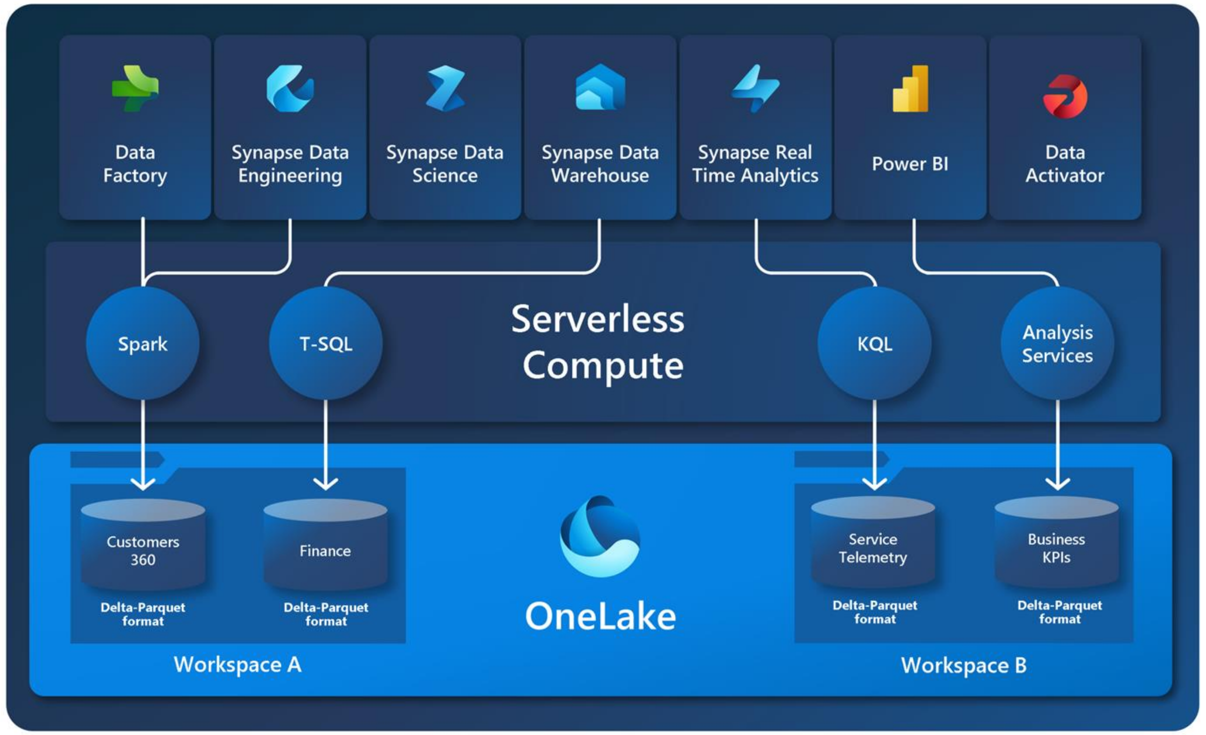


Fabric: tenant design



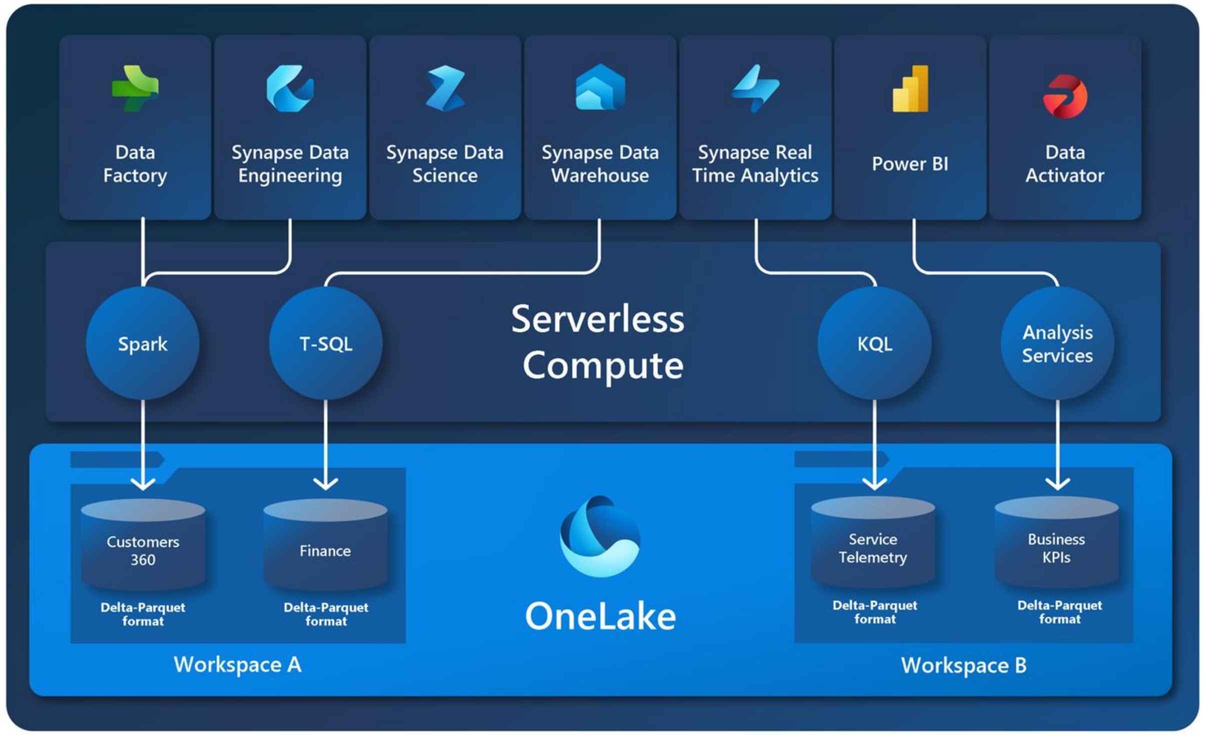
Fabric: storage

- Storage in Fabric is called 'OneLake'
- It's data lake storage (blob), with one OneLake per organisation (tenant)
- Storage is cheap! ± 2,2 cent per GB per month



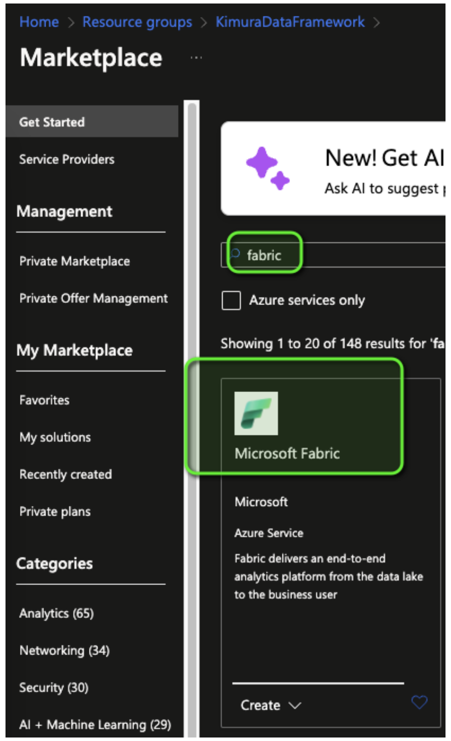
Fabric: compute

- Compute in Fabric is Serverless, and included in your SaaS capacity
- For data engineering purposes, you get a Spark cluster that is preconfigured and can be configured to your custom needs
- Compute = expensive. Be smart about it ☺



Buying the Fabric compute capacity

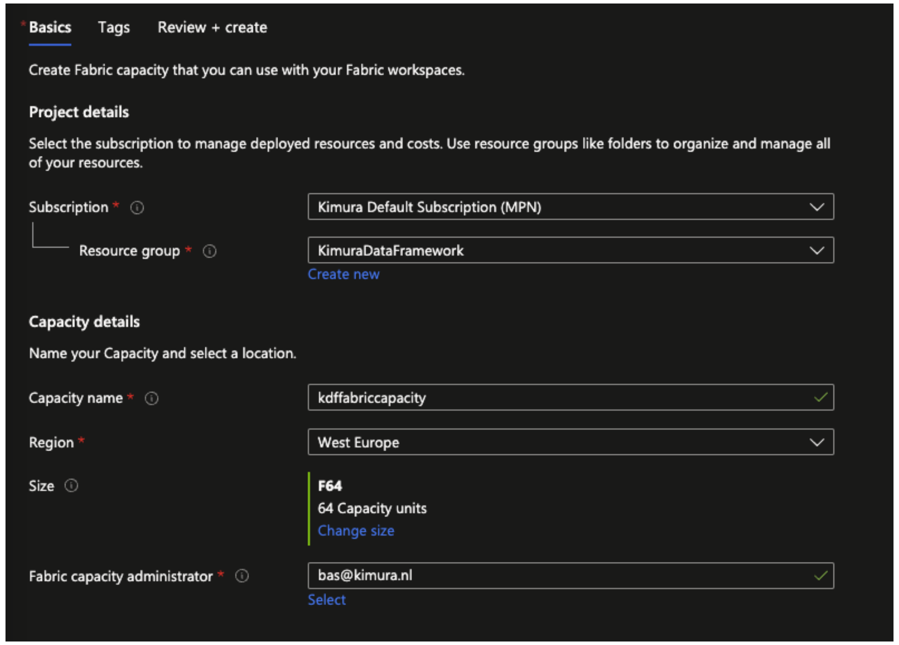
- portal.azure.com
- I'd create at least a new Resource Group, maybe even a Subscription (depending on organisation requirements)



Home > Resource groups > KimuraDataFramework >

Marketplace ...

Fabric delivers an end-to-end analytics platform from the data lake to the business user



Create Fabric capacity that you can use with your Fabric workspaces.

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all of your resources.

Subscription * : Kimura Default Subscription (MPN)

Resource group * : KimuraDataFramework

Create new

Capacity details

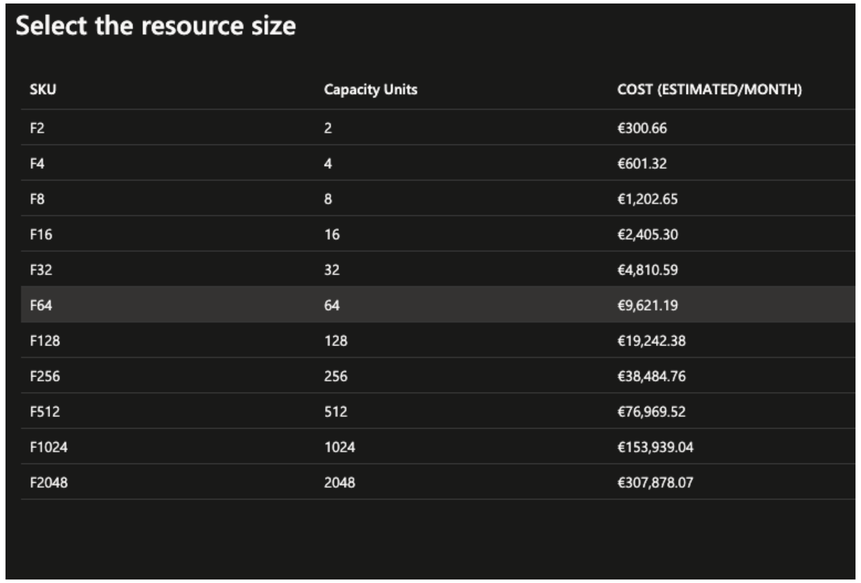
Name your Capacity and select a location.

Capacity name * : kdffabriccapacity

Region * : West Europe

Size : **F64**
64 Capacity units
Change size

Fabric capacity administrator * : bas@kimura.nl
Select

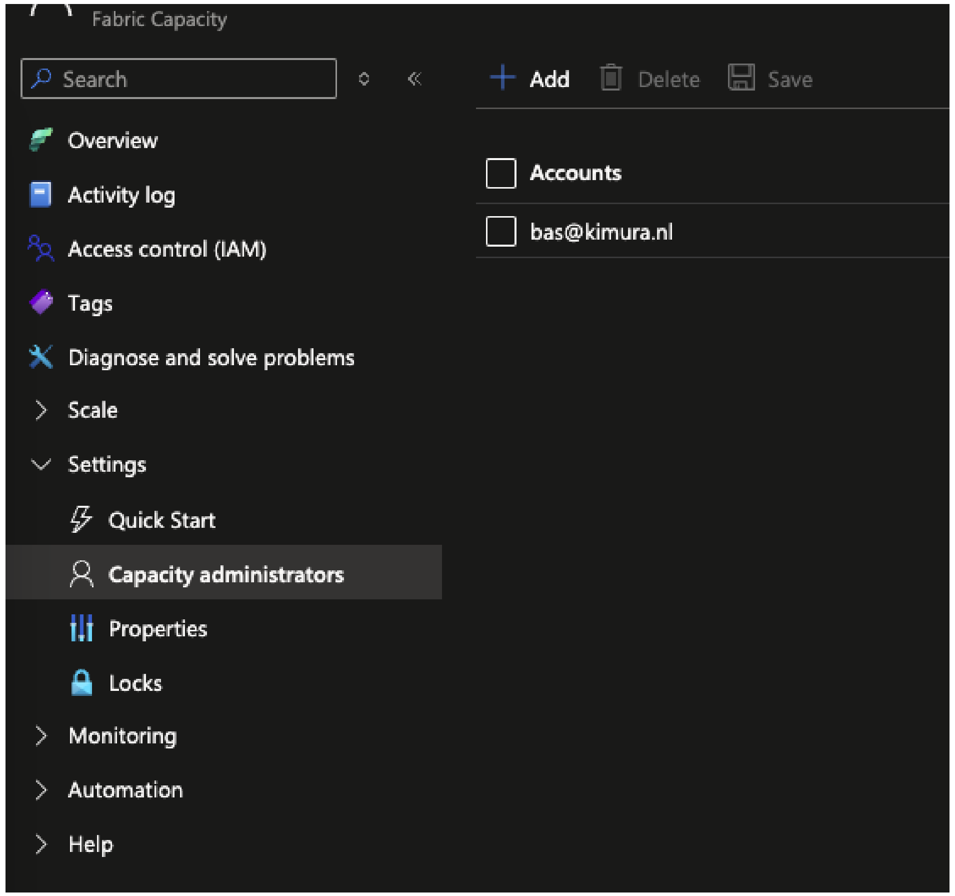


Select the resource size

SKU	Capacity Units	COST (ESTIMATED/MONTH)
F2	2	€300.66
F4	4	€601.32
F8	8	€1,202.65
F16	16	€2,405.30
F32	32	€4,810.59
F64	64	€9,621.19
F128	128	€19,242.38
F256	256	€38,484.76
F512	512	€76,969.52
F1024	1024	€153,939.04
F2048	2048	€307,878.07

Buying the Fabric compute capacity

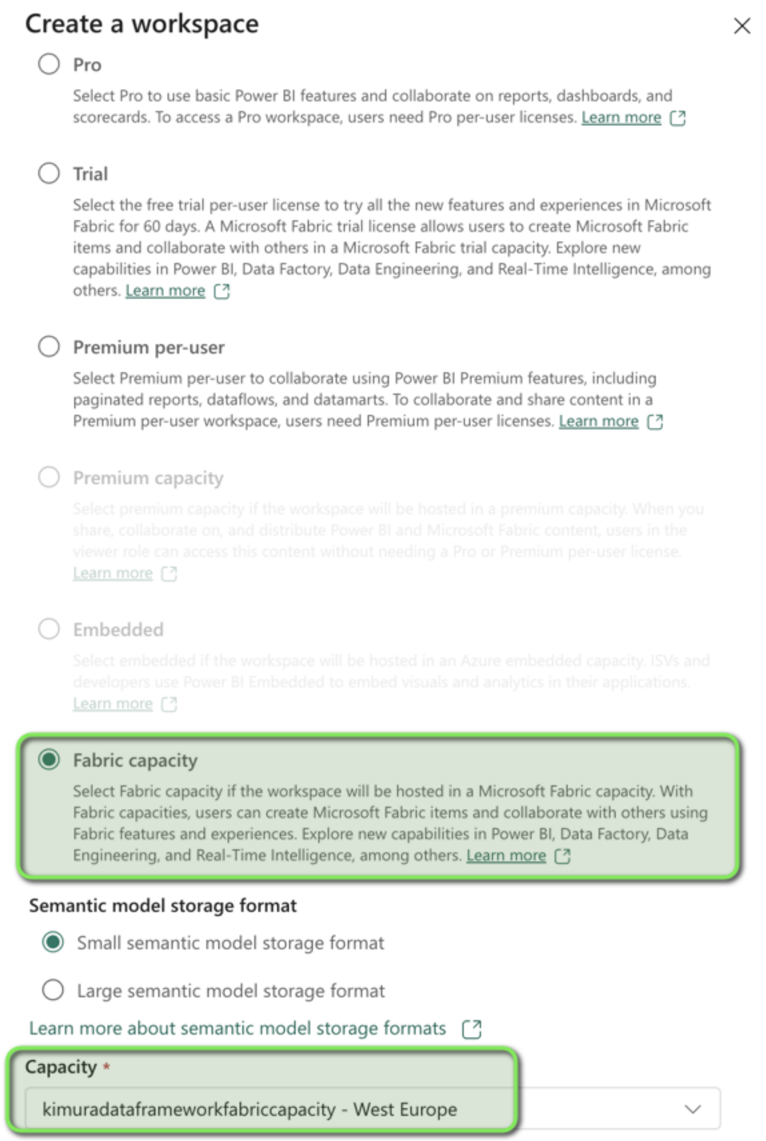
- After deployment, configure the Capacity Administrator:
- Unfortunately, no Entra ID groups…



Accounts
bas@kimura.nl

Creating the first Fabric Workspace

- Head over to app.powerbi.com
- Log in with the same account that is the Capacity Administrator
- Create a new workspace,
- Now, pay attention to the License Type


Create a workspace

Pro

Select Pro to use basic Power BI features and collaborate on reports, dashboards, and scorecards. To access a Pro workspace, users need Pro per-user licenses. Learn more

Trial

Select the free trial per-user license to try all the new features and experiences in Microsoft Fabric for 60 days. A Microsoft Fabric trial license allows users to create Microsoft Fabric items and collaborate with others in a Microsoft Fabric trial capacity. Explore new capabilities in Power BI, Data Factory, Data Engineering, and Real-Time Intelligence, among others. Learn more

Premium per-user

Select Premium per-user to collaborate using Power BI Premium features, including paginated reports, dataflows, and datamarts. To collaborate and share content in a Premium per-user workspace, users need Premium per-user licenses. Learn more

Premium capacity

Select premium capacity if the workspace will be hosted in a premium capacity. When you share, collaborate on, and distribute Power BI and Microsoft Fabric content, users in the viewer role can access this content without needing a Pro or Premium per-user license. Learn more

Embedded

Select embedded if the workspace will be hosted in an Azure embedded capacity. ISVs and developers use Power BI Embedded to embed visuals and analytics in their applications. Learn more

[✓ Fabric capacity]

Select Fabric capacity if the workspace will be hosted in a Microsoft Fabric capacity. With Fabric capacities, users can create Microsoft Fabric items and collaborate with others using Fabric features and experiences. Explore new capabilities in Power BI, Data Factory, Data Engineering, and Real-Time Intelligence, among others. Learn more

Semantic model storage format

[✓ Small semantic model storage format]

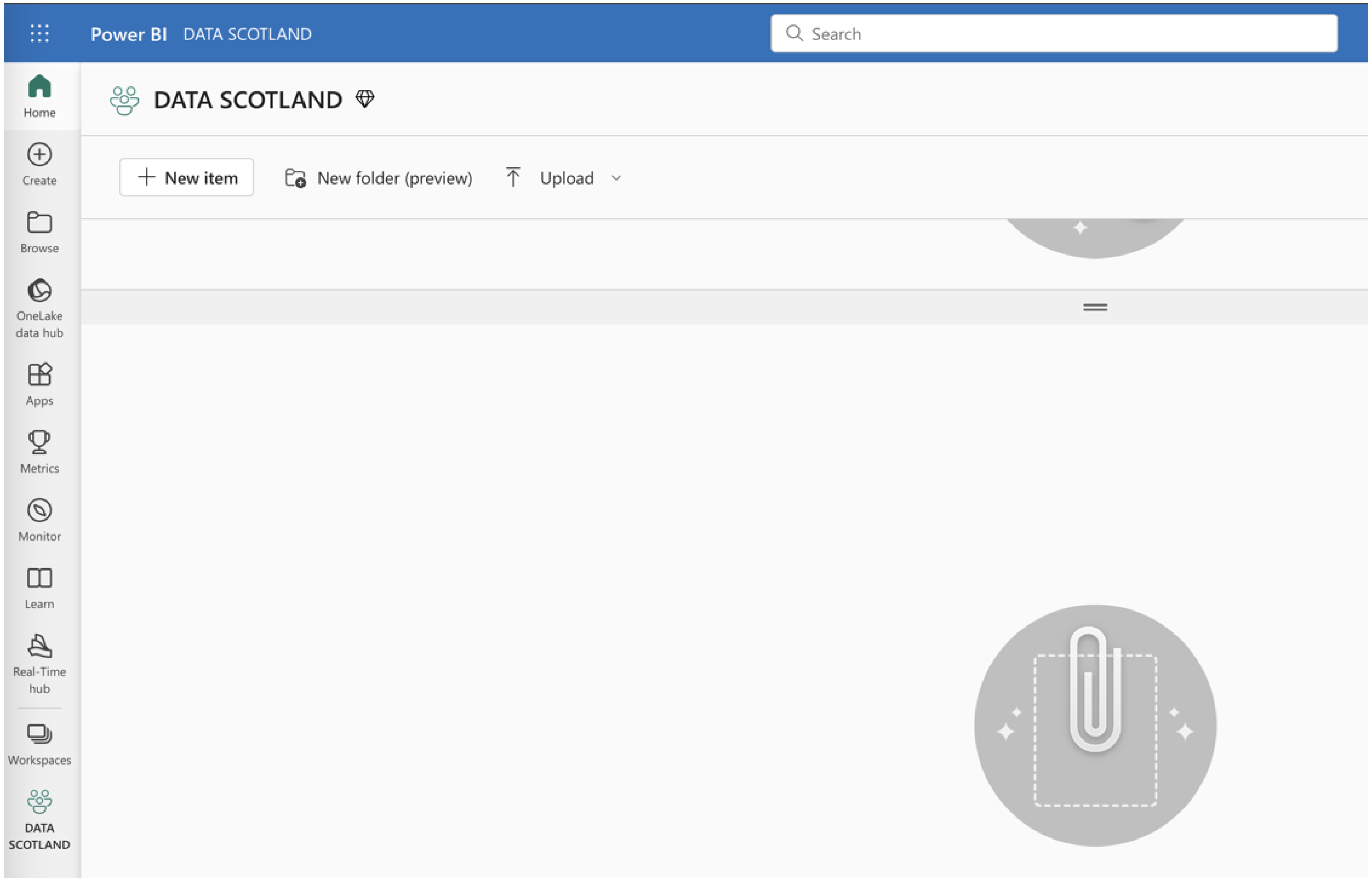
Large semantic model storage format

Learn more about semantic model storage formats

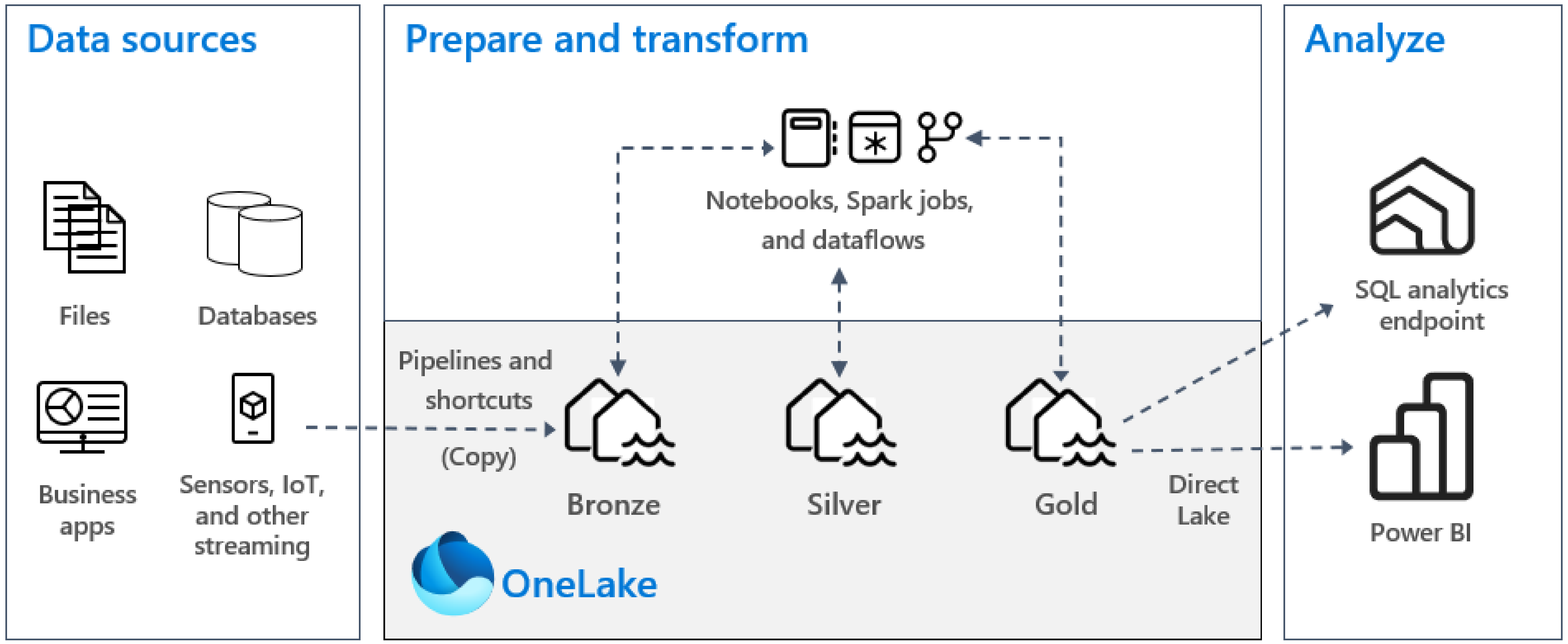
Capacity *

kimuradataframeworkfabriccapacity - West Europe

Creating the first Fabric Workspace

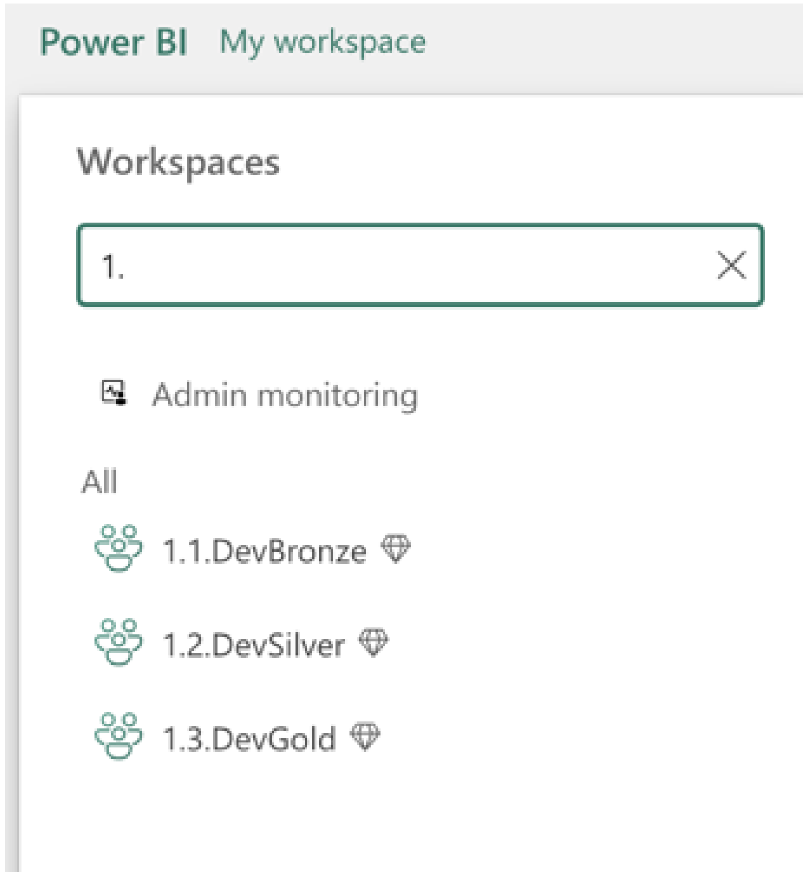


Medallion architecture



Workspaces and Medallion

- One workspace per Medallion stage
- One workspace per DTAP environment



Your first Fabric Lakehouse

The scenario

- Client: small B2B professional services company
- Main software: Moneybird, Odoo
- Data domains: Finance, Projects & Timesheets

The rulebook

- Data Factory Pipelines orchestration
- PySpark notebooks for processing
- Reusable code for everything
- Lakehouse for storage

Prerequisites

Setting up:

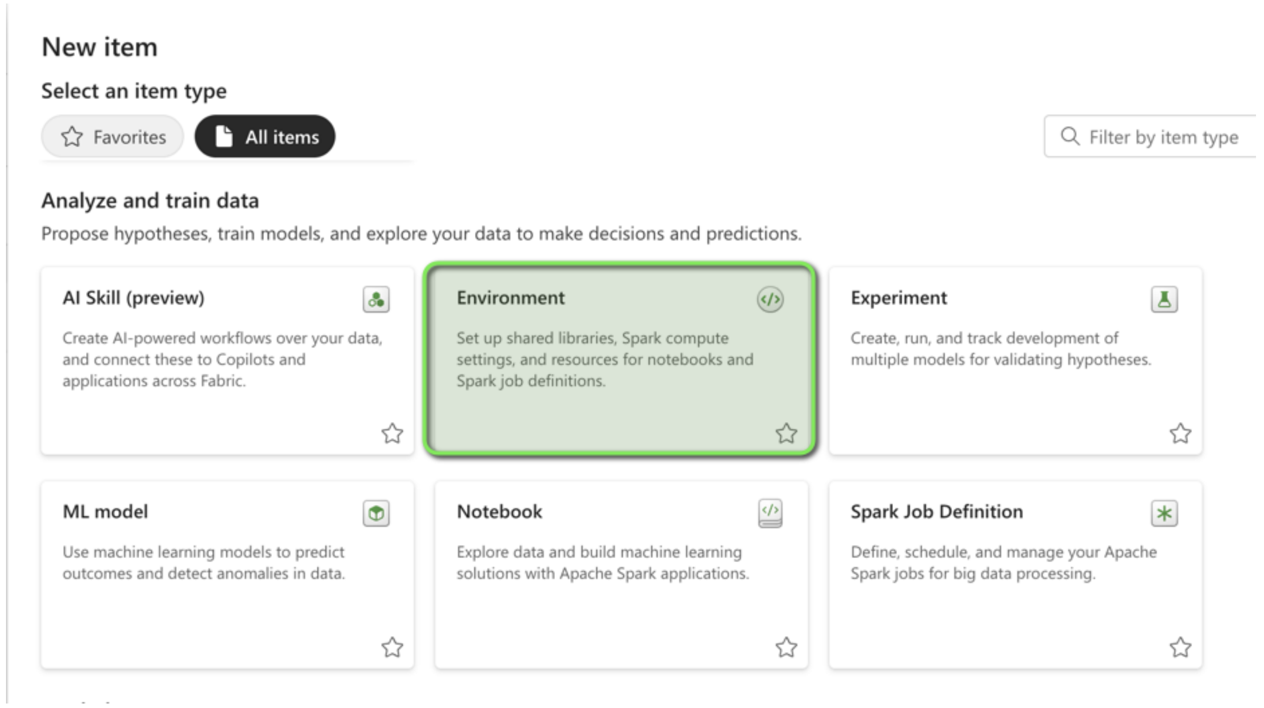
- Fabric compute capacity (done)
- Fabric workspaces (done)
- Configuring Spark (we skip that for today)
- Setting up the Environment (**important**!)

Environments in Fabric

- Fabric contains a lot of Python packages natively
- You can also add packages using the PyPI (Python Package Index)
- And, you can write and include your own code
- This is done using Environments

Environments in Fabric

- Create a new environment:



New item

Select an item type

Analyze and train data

Propose hypotheses, train models, and explore your data to make decisions and predictions.

AI Skill (preview) — Create AI-powered workflows over your data, and connect these to Copilots and applications across Fabric.

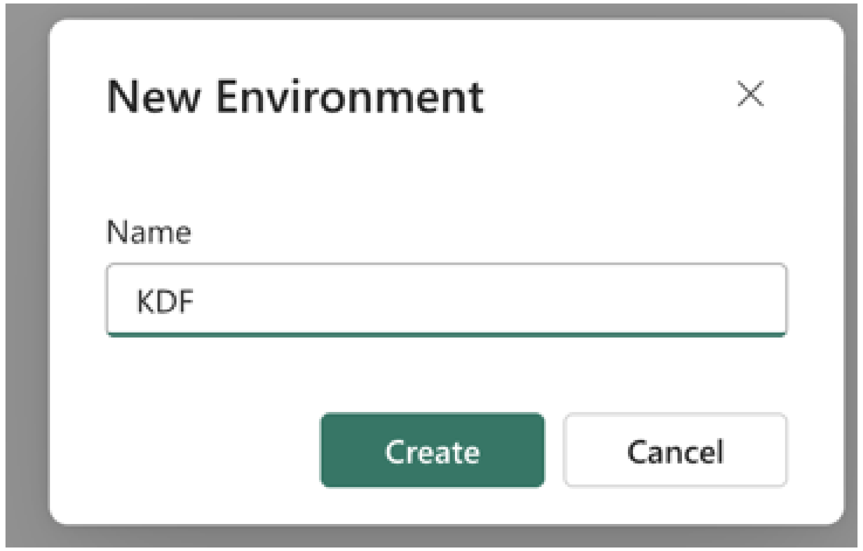
Environment — Set up shared libraries, Spark compute settings, and resources for notebooks and Spark job definitions.

Experiment — Create, run, and track development of multiple models for validating hypotheses.

ML model — Use machine learning models to predict outcomes and detect anomalies in data.

Notebook — Explore data and build machine learning solutions with Apache Spark applications.

Spark Job Definition — Define, schedule, and manage your Apache Spark jobs for big data processing.

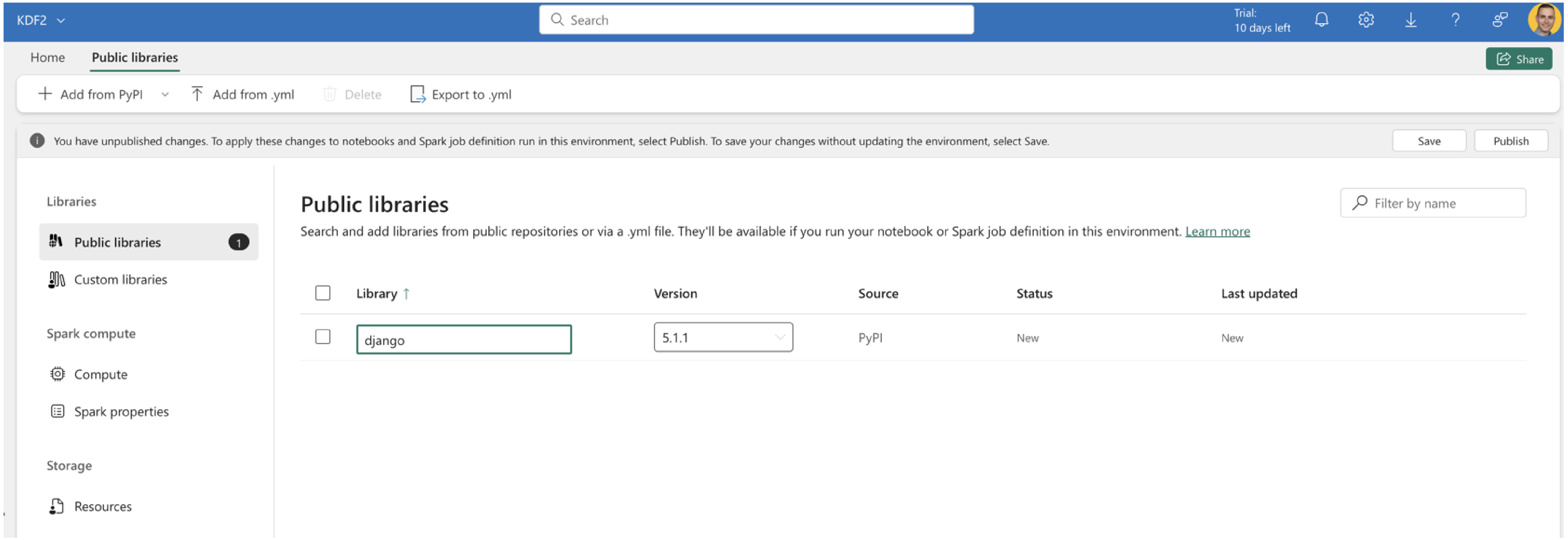


New Environment

Name: KDF

Environments in Fabric

- Add code from PyPI:



You have unpublished changes. To apply these changes to notebooks and Spark job definition run in this environment, select Publish. To save your changes without updating the environment, select Save.

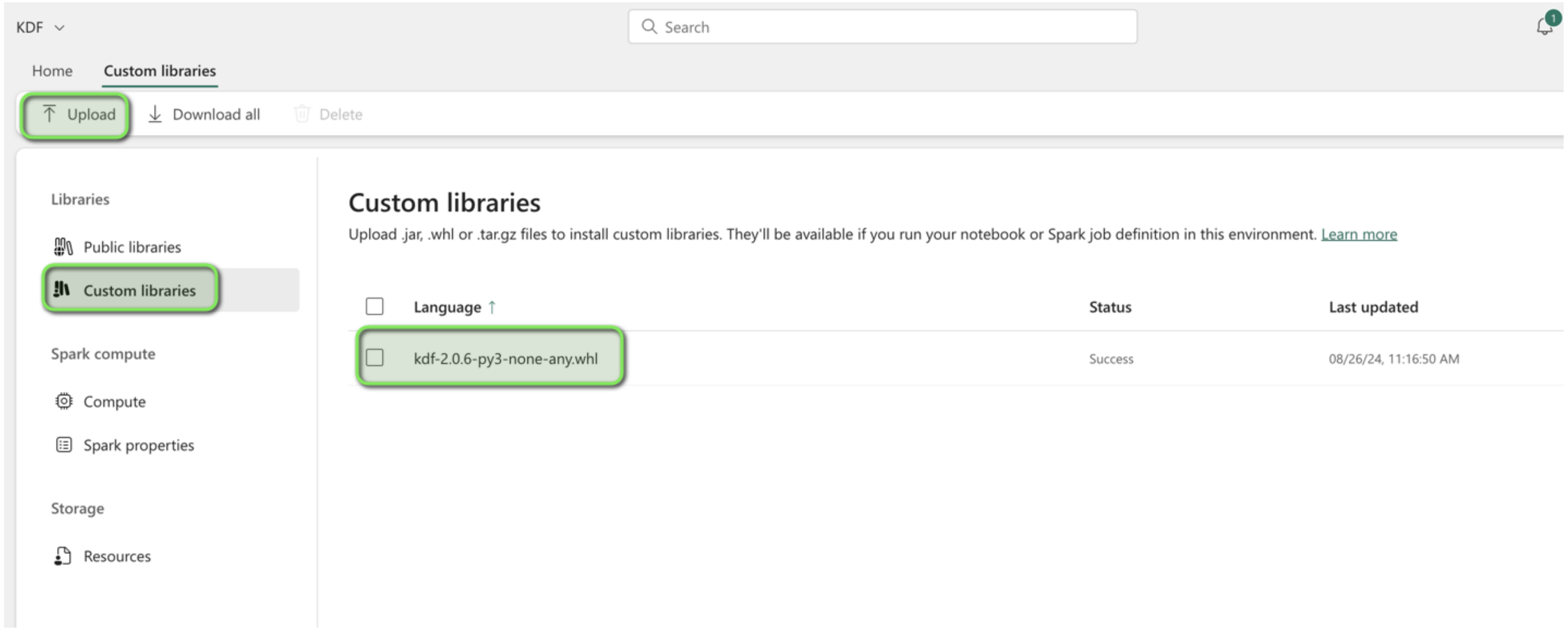
Public libraries

Search and add libraries from public repositories or via a .yml file. They'll be available if you run your notebook or Spark job definition in this environment. Learn more

Library	Version	Source	Status	Last updated
django	5.1.1	PyPI	New	New

Environments in Fabric

- Add your own code:

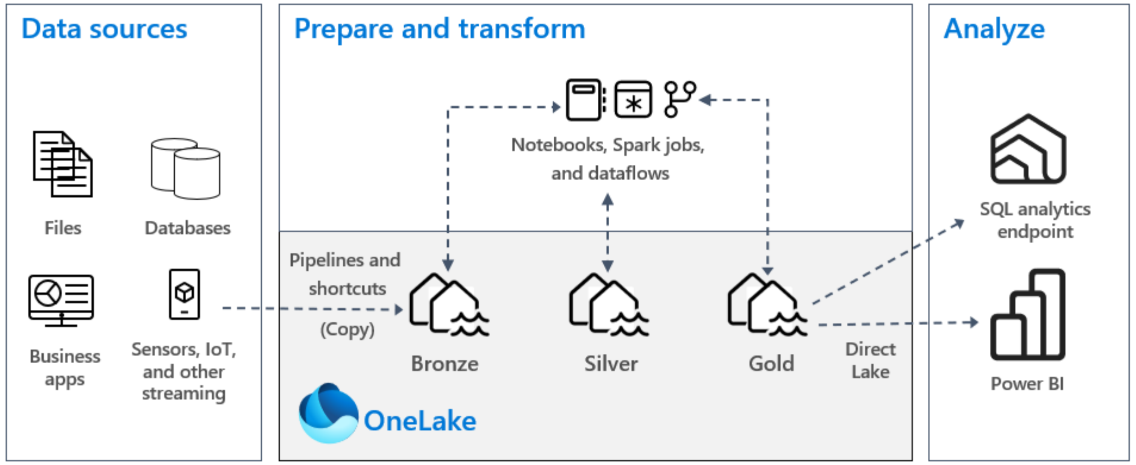


Custom libraries

Upload .jar, .whl or .tar.gz files to install custom libraries. They'll be available if you run your notebook or Spark job definition in this environment. Learn more

Language	Status	Last updated
kdf-2.0.6-py3-none-any.whl	Success	08/26/24, 11:16:50 AM

Three lakehouses

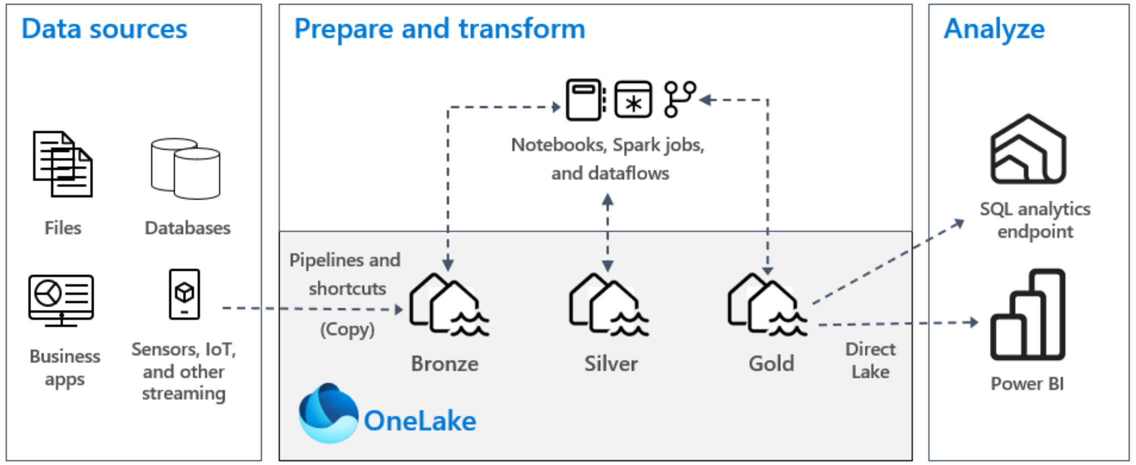


- Bronze: timestamped folders, (incremental loads), raw data (files; json, xml, csv, parquet etc)

- Silver: historical archive, normalised schemas, delta tables

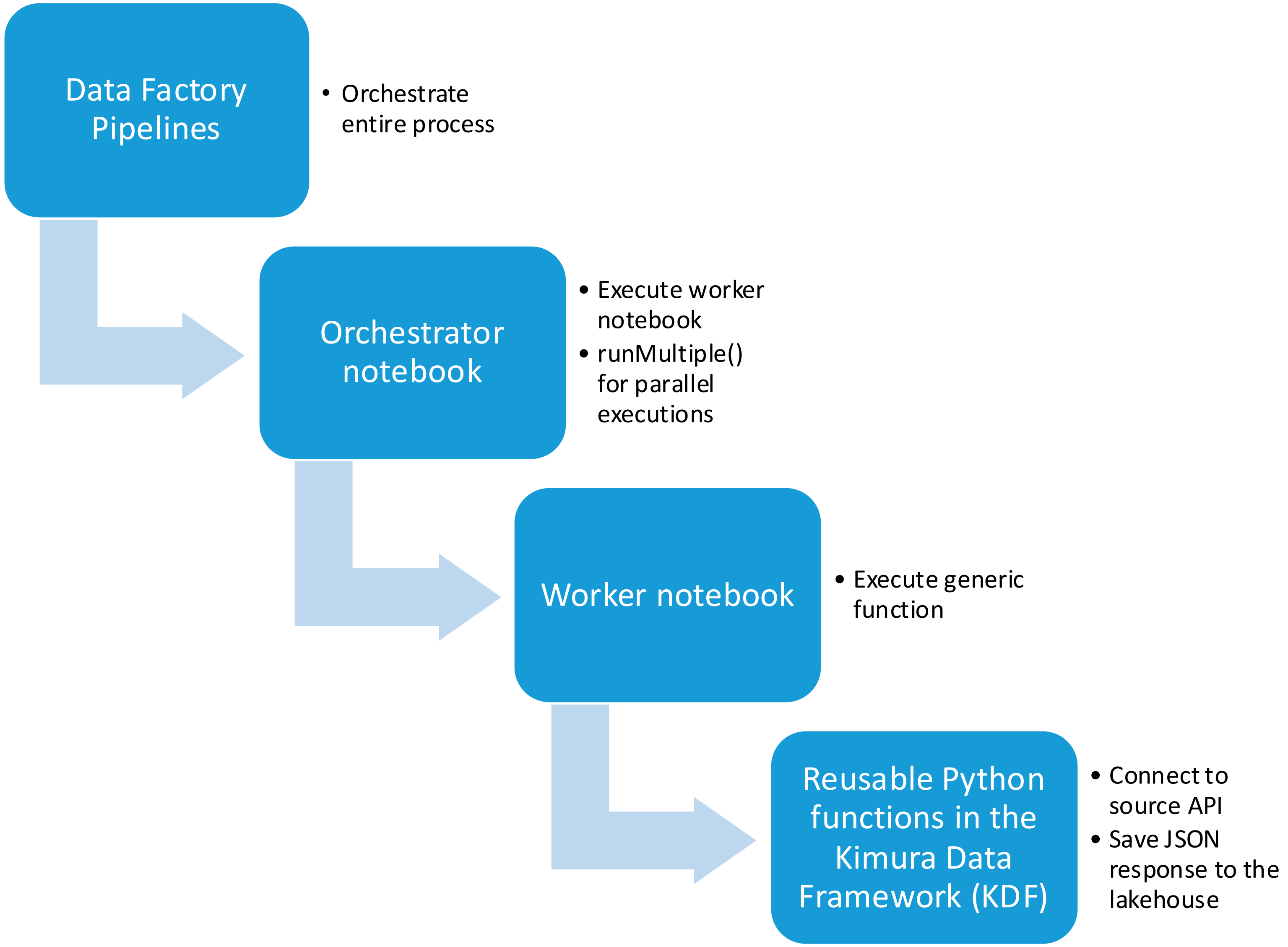
- Gold: prepared and modeled DWH tables, dims and facts, delta tables

Setting up the ELT process



- Keep in mind the Medallion architecture
- Start connecting to data, to extract Moneybird and Odoo to Bronze
- We do so using Python, executed in Fabric Notebooks, scheduled and run in Fabric Pipelines

ELT: Extract



Data Factory Pipelines
Orchestrator notebook
Worker notebook
Reusable Python functions in the Kimura Data Framework (KDF)

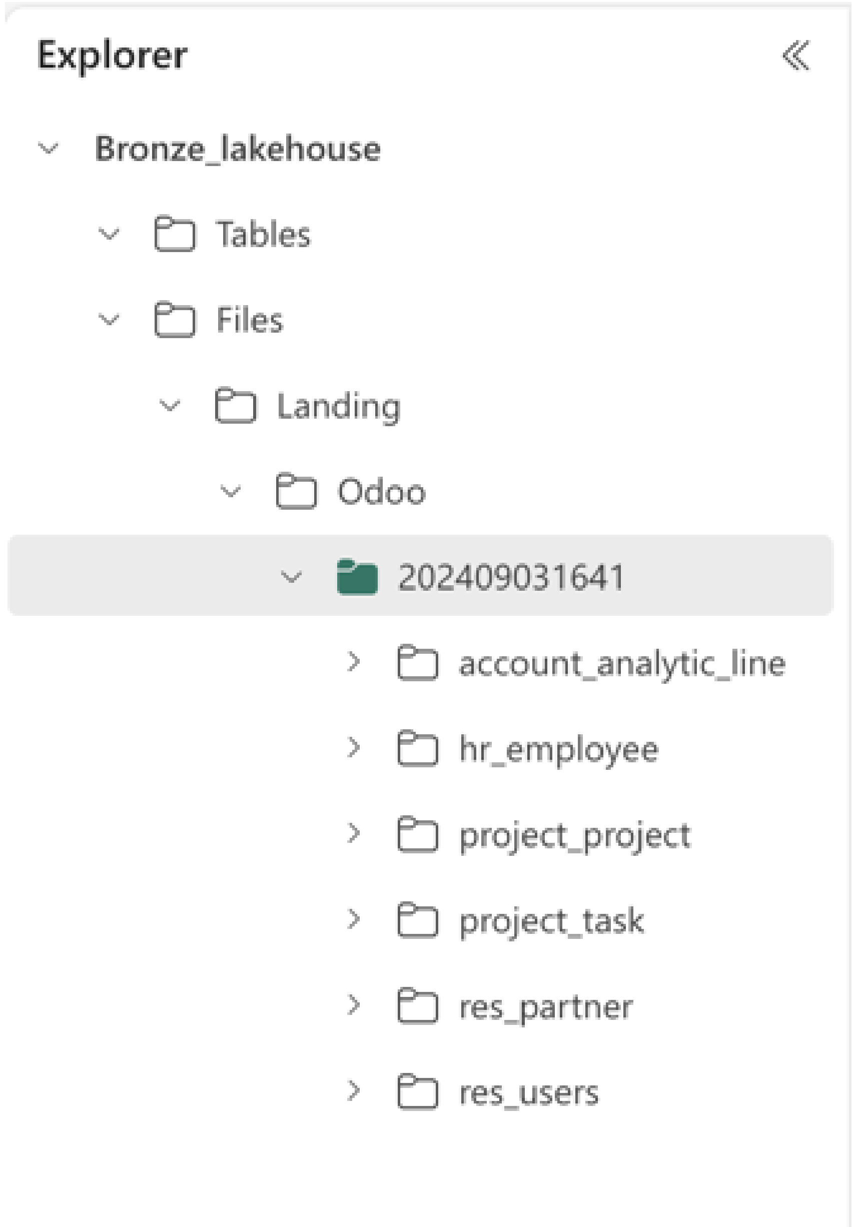
- Orchestrate entire process

- Execute worker notebook
- runMultiple() for parallel executions

- Execute generic function

- Connect to source API
- Save JSON response to the lakehouse

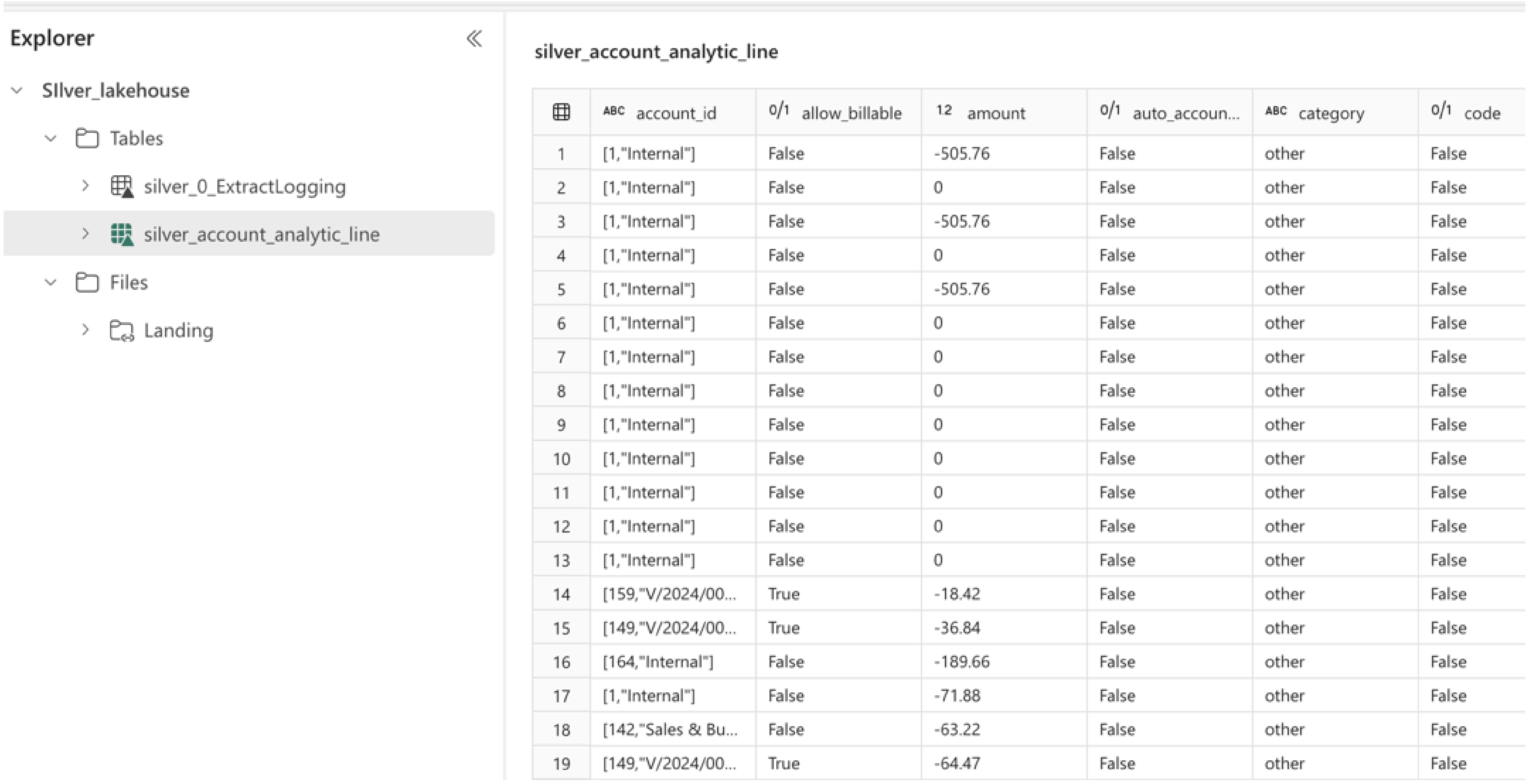
ELT: Bronze folder structure


Explorer

- Bronze_lakehouse
  - Tables
  - Files
    - Landing
      - Odoo
        - 202409031641
          - account_analytic_line
          - hr_employee
          - project_project
          - project_task
          - res_partner
          - res_users

- Root folder ‘Landing’ (for shortcuts!)
- Source system
- Timestamp
- Folder per table/endpoint
- Files managed by Fabric!

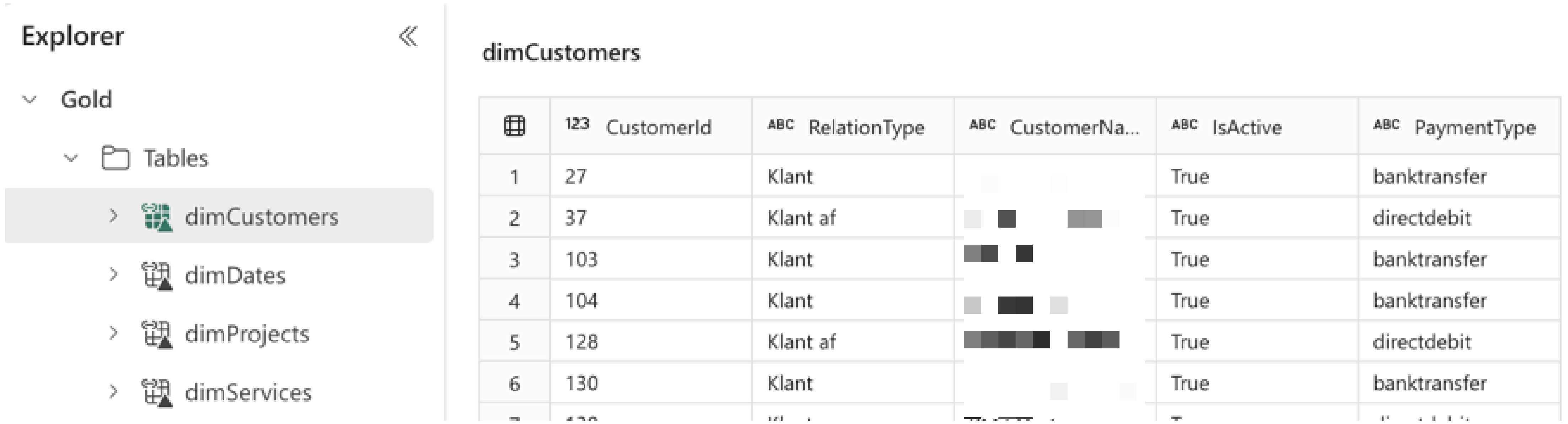
Silver lakehouse



silver_account_analytic_line

	account_id	allow_billable	amount	auto_accoun...	category	code
1	[1,"Internal"]	False	-505.76	False	other	False
2	[1,"Internal"]	False	0	False	other	False
3	[1,"Internal"]	False	-505.76	False	other	False
4	[1,"Internal"]	False	0	False	other	False
5	[1,"Internal"]	False	-505.76	False	other	False
6	[1,"Internal"]	False	0	False	other	False
7	[1,"Internal"]	False	0	False	other	False
8	[1,"Internal"]	False	0	False	other	False
9	[1,"Internal"]	False	0	False	other	False
10	[1,"Internal"]	False	0	False	other	False
11	[1,"Internal"]	False	0	False	other	False
12	[1,"Internal"]	False	0	False	other	False
13	[1,"Internal"]	False	0	False	other	False
14	[159,"V/2024/00...	True	-18.42	False	other	False
15	[149,"V/2024/00...	True	-36.84	False	other	False
16	[164,"Internal"]	False	-189.66	False	other	False
17	[1,"Internal"]	False	-71.88	False	other	False
18	[142,"Sales & Bu...	False	-63.22	False	other	False
19	[149,"V/2024/00...	True	-64.47	False	other	False

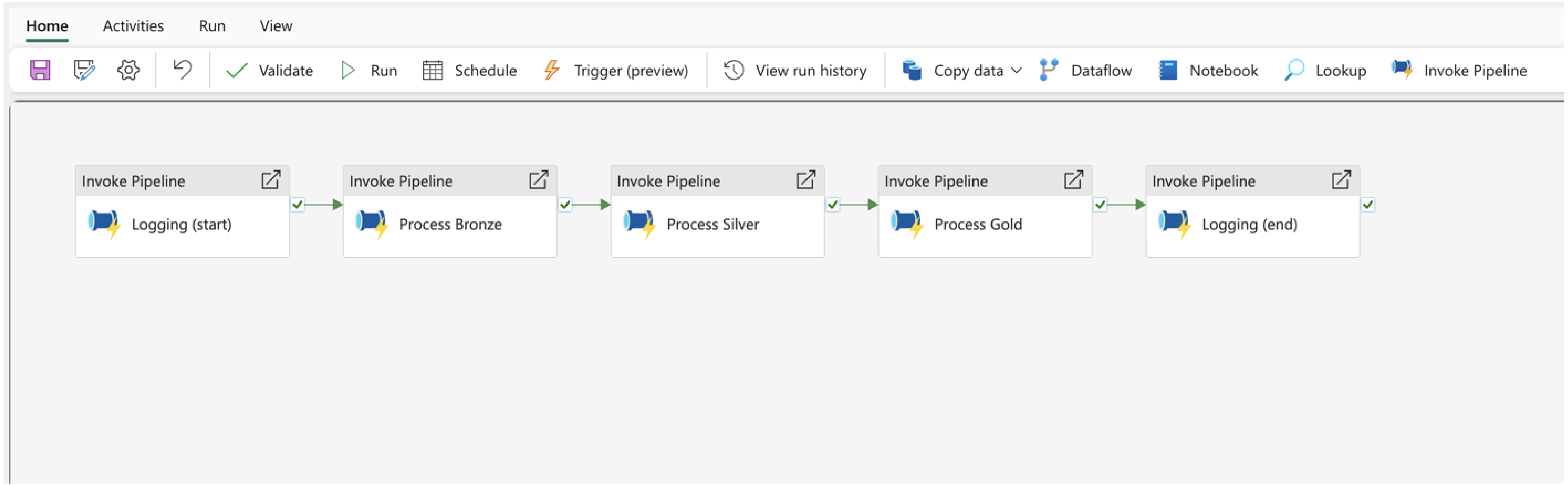
Gold lakehouse



dimCustomers

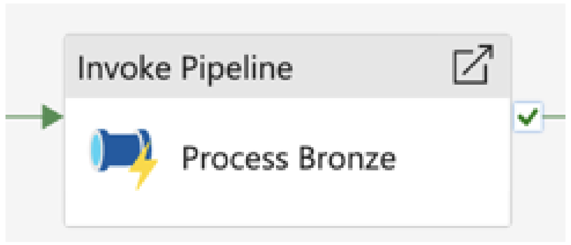
	CustomerId	RelationType	CustomerNa...	IsActive	PaymentType
1	27	Klant	[redacted]	True	banktransfer
2	37	Klant af	[redacted]	True	directdebit
3	103	Klant	[redacted]	True	banktransfer
4	104	Klant	[redacted]	True	banktransfer
5	128	Klant af	[redacted]	True	directdebit
6	130	Klant	[redacted]	True	banktransfer

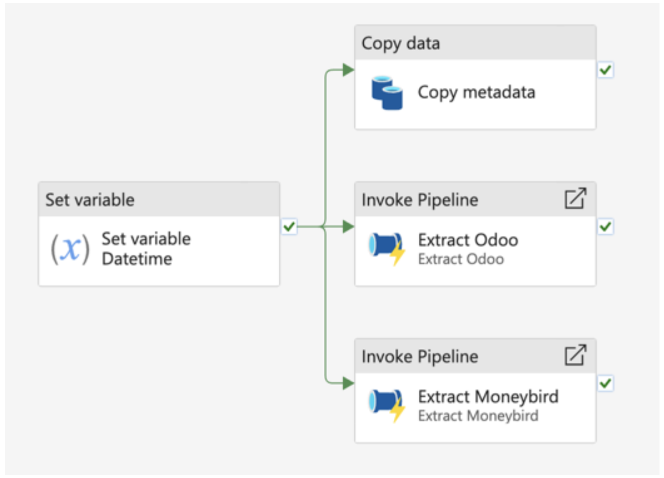
End-to-end solution with examples

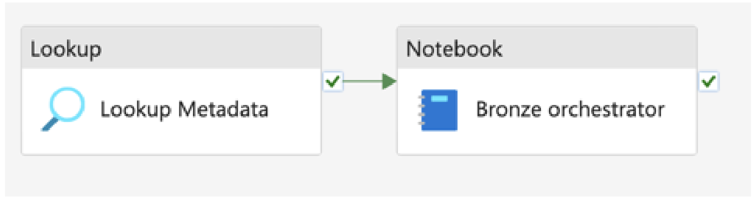


Invoke Pipeline — Logging (start)
Invoke Pipeline — Process Bronze
Invoke Pipeline — Process Silver
Invoke Pipeline — Process Gold
Invoke Pipeline — Logging (end)

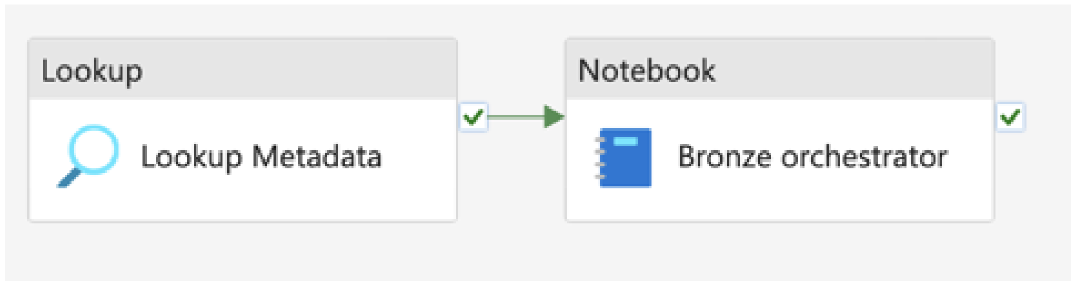
End-to-end solution with examples

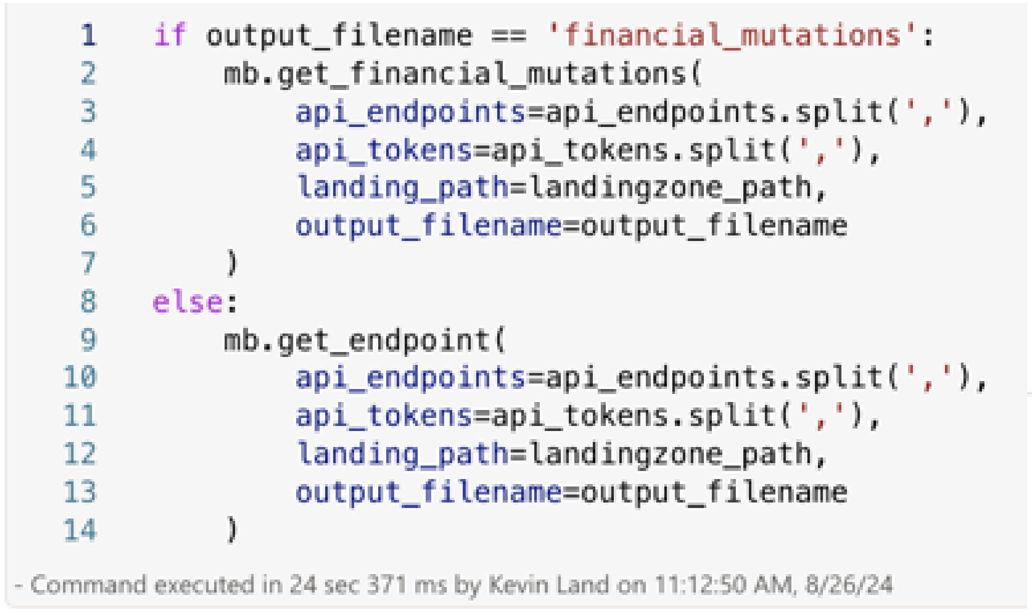






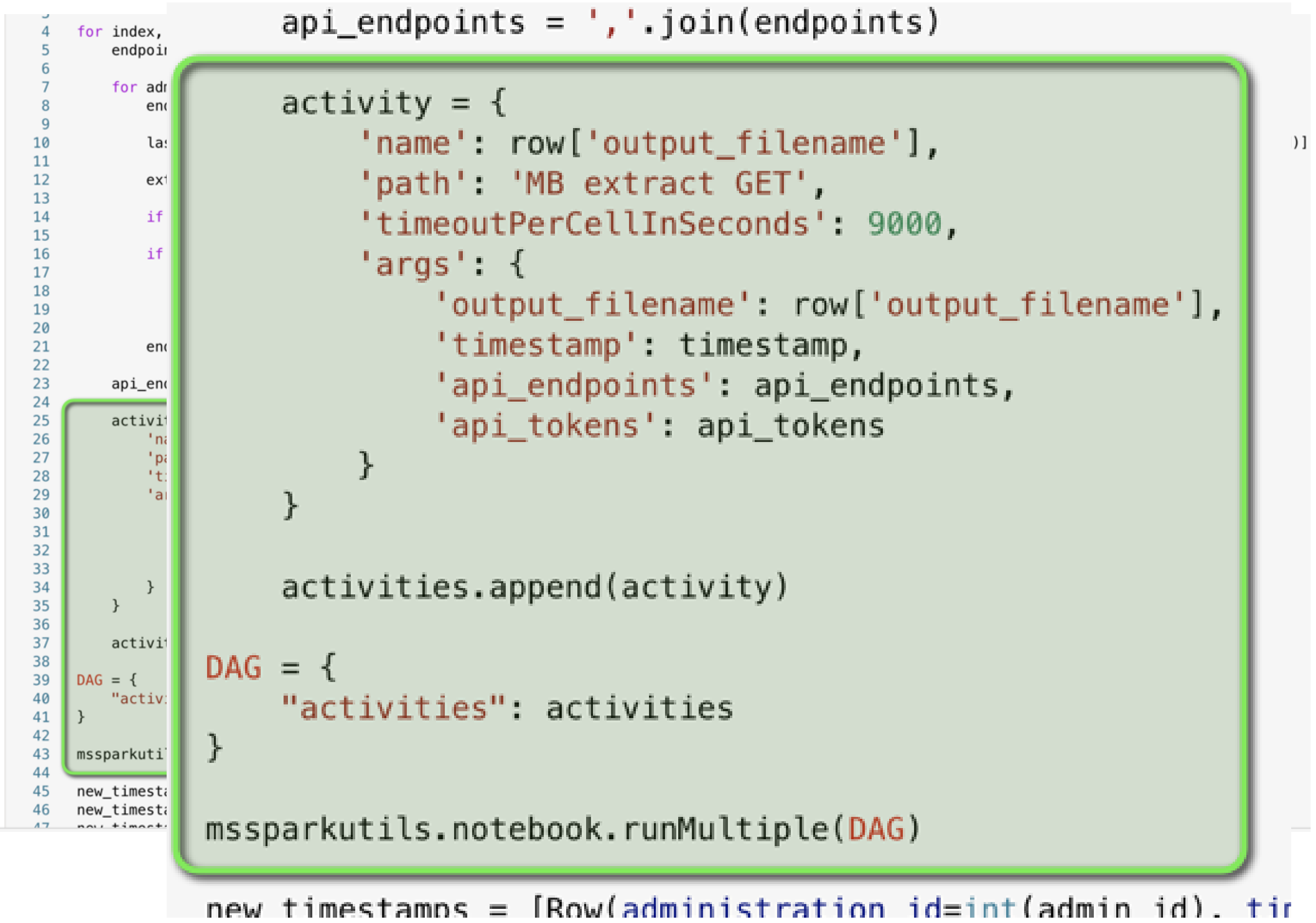
End-to-end solution with examples





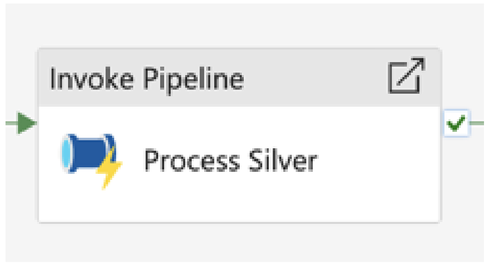
```python
if output_filename == 'financial_mutations':
    mb.get_financial_mutations(
        api_endpoints=api_endpoints.split(','),
        api_tokens=api_tokens.split(','),
        landing_path=landingzone_path,
        output_filename=output_filename
    )
else:
    mb.get_endpoint(
        api_endpoints=api_endpoints.split(','),
        api_tokens=api_tokens.split(','),
        landing_path=landingzone_path,
        output_filename=output_filename
    )
```

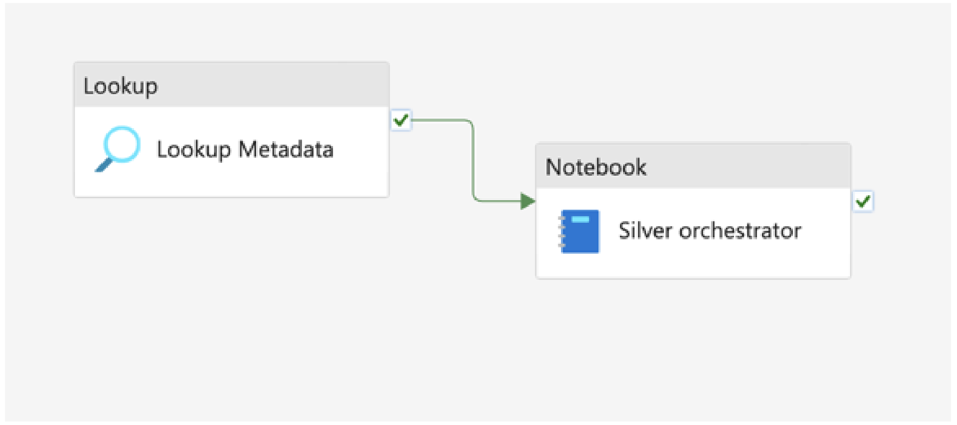
- Command executed in 24 sec 371 ms by Kevin Land on 11:12:50 AM, 8/26/24

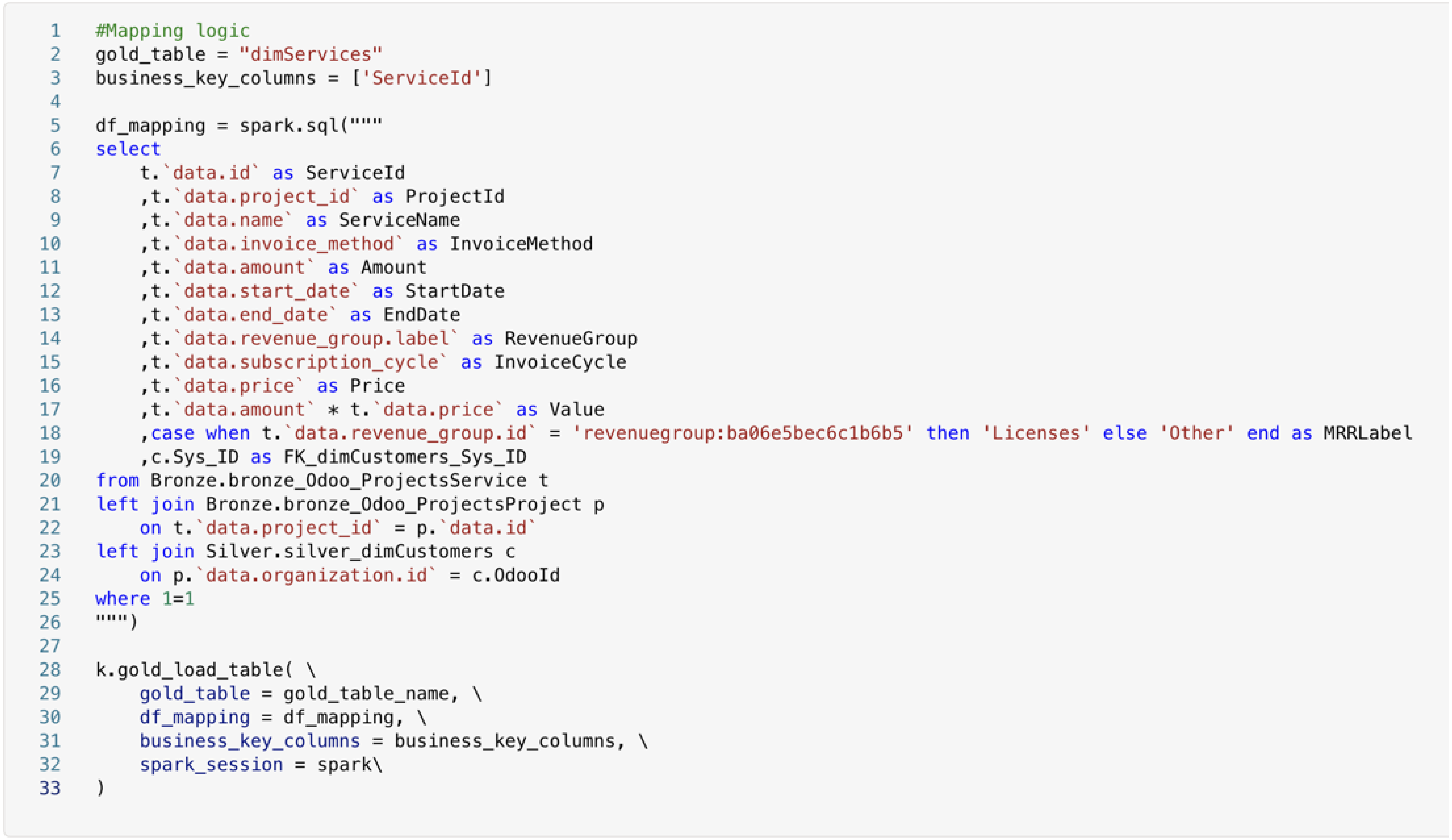


```python
    api_endpoints = ','.join(endpoints)

    activity = {
        'name': row['output_filename'],
        'path': 'MB extract GET',
        'timeoutPerCellInSeconds': 9000,
        'args': {
            'output_filename': row['output_filename'],
            'timestamp': timestamp,
            'api_endpoints': api_endpoints,
            'api_tokens': api_tokens
        }
    }

    activities.append(activity)

DAG = {
    "activities": activities
}

mssparkutils.notebook.runMultiple(DAG)

new_timestamps = [Row(administration_id=int(admin_id), ti
```

End-to-end solution with examples







```python
#Mapping logic
gold_table = "dimServices"
business_key_columns = ['ServiceId']

df_mapping = spark.sql("""
select
    t.`data.id` as ServiceId
    ,t.`data.project_id` as ProjectId
    ,t.`data.name` as ServiceName
    ,t.`data.invoice_method` as InvoiceMethod
    ,t.`data.amount` as Amount
    ,t.`data.start_date` as StartDate
    ,t.`data.end_date` as EndDate
    ,t.`data.revenue_group.label` as RevenueGroup
    ,t.`data.subscription_cycle` as InvoiceCycle
    ,t.`data.price` as Price
    ,t.`data.amount` * t.`data.price` as Value
    ,case when t.`data.revenue_group.id` = 'revenuegroup:ba06e5bec6c1b6b5' then 'Licenses' else 'Other' end as MRRLabel
    ,c.Sys_ID as FK_dimCustomers_Sys_ID
from Bronze.bronze_Odoo_ProjectsService t
left join Bronze.bronze_Odoo_ProjectsProject p
    on t.`data.project_id` = p.`data.id`
left join Silver.silver_dimCustomers c
    on p.`data.organization.id` = c.OdooId
where 1=1
""")

k.gold_load_table( \
    gold_table = gold_table_name, \
    df_mapping = df_mapping, \
    business_key_columns = business_key_columns, \
    spark_session = spark\
)
```

Recap

Recap

1. Keep core-concepts in mind
2. Setup a rulebook and abide by it
3. Use tools and reusable code!
4. Fabric lakehouses aren’t that scary ☺

Bas Land

- BI consultant since 2013
- Co-founder of two data companies:
  - Kimura Data Intelligence (consultancy)
  - DataChimp (SaaS analytics for accounting firms)

- Married to Anouk, we have a daxhund (☺) called Chester
- Sports: purple belt Brazilian jiu-jitsu, weight lifting, running

Kimura Data Intelligence B.V.

Fonteinkruid 6b
3931 WX, Woudenberg
The Netherlands

www.kimura.nl
info@kimura.nl

Bas Land
Managing Partner

bas@kimura.nl

DATA:Scotland
2024

Session Feedback

Event Feedback