North Insights

FABRIC ARCHITECTURE

Ásgeir Gunnarsson
@bidgeir



ÁSGEIR GUNNARSSON

WHO AM I

- Data Platform MVP
- Co-owner of Data Lab ApS
- In BI since 2007
- Co-author of Pro Microsoft Power BI Administration
- Co-organizer of Power BI Next Step, Data Platform Next Step and Power BI Cruise in Denmark

- E-mail: asgeir@thesmalldatalab.com
- Twitter: @bidgeir
- LinkedIn: linkedin.com/in/asgeirgun



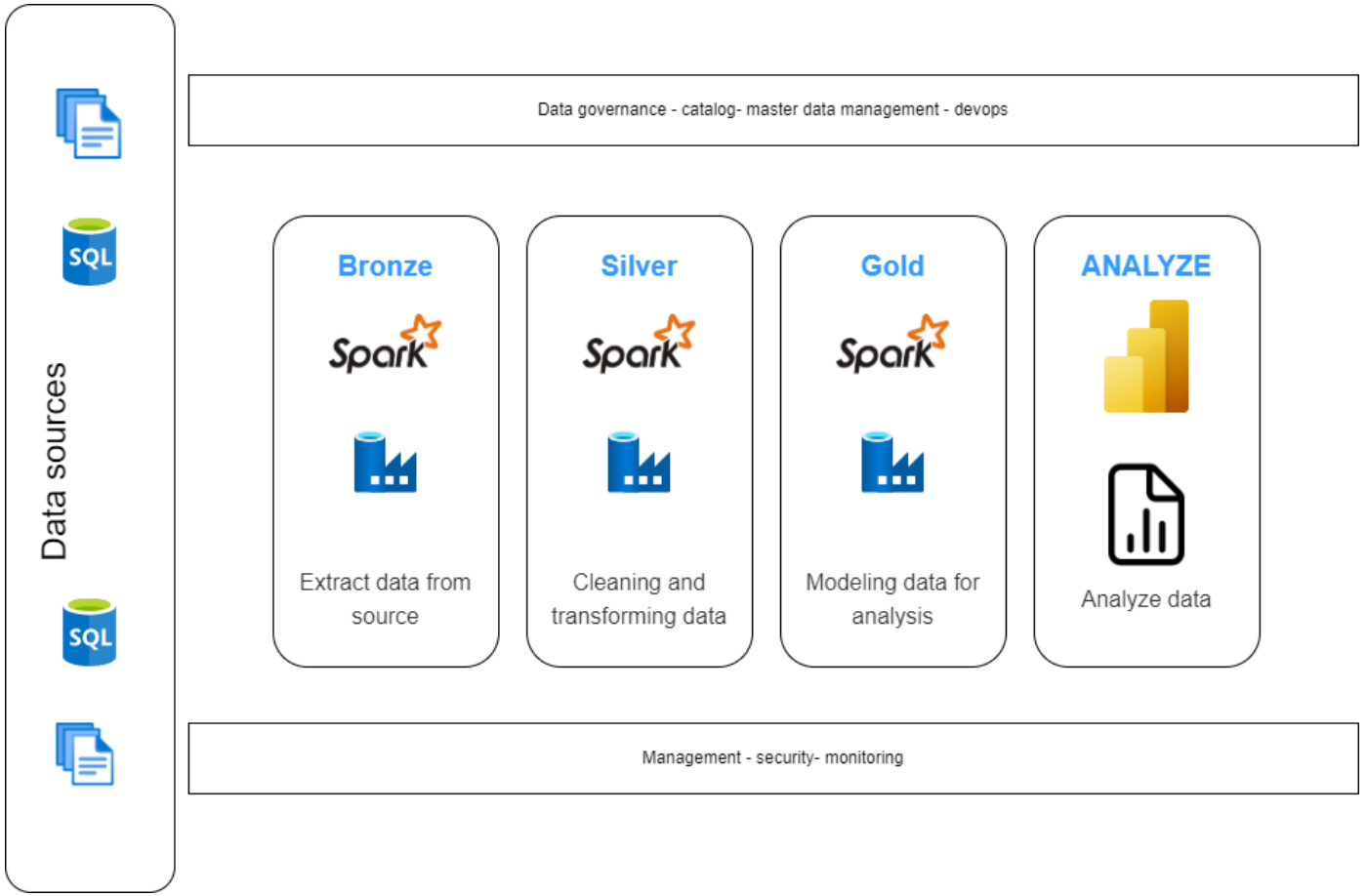
AGENDA

- Why this session
- Architecture patterns
- Workspace strategy
- DevOps impact
- Migration/Re-implementation from SQL

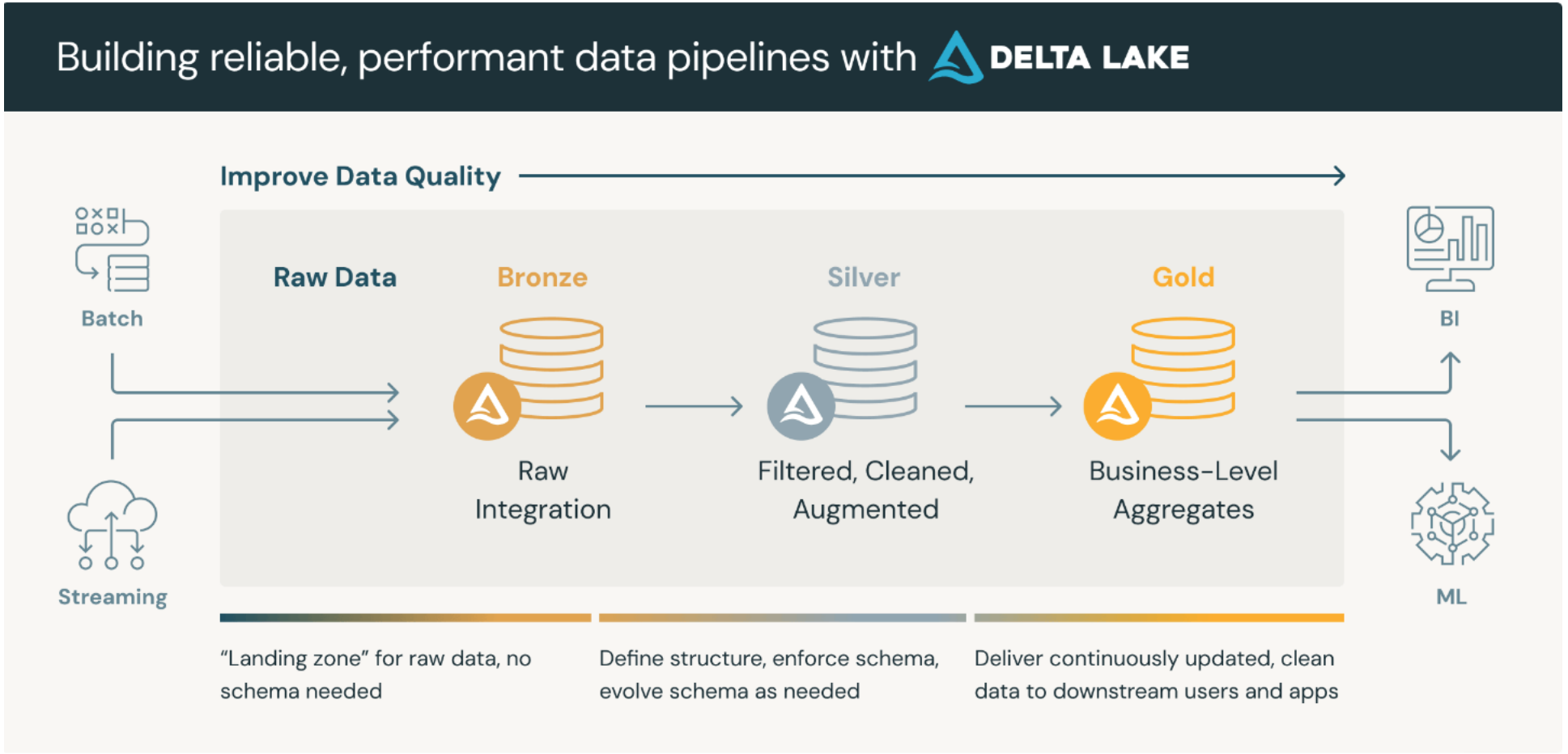


LAKEHOUSE PATTERNS

- Data loaded in as is (Bronze)
- Data transformed via Spark (Silver)
- Data loaded into the Lakehouse via Spark (Gold)
- Data mart exposed via SQL endpoints or lakehouse tables



LAKEHOUSE PATTERNS



MEDALLION ARCHITECTURE

- Let's assume some version of the medallion architecture
- What do you need to be aware of in Fabric

A **medallion architecture** is a data design pattern used to logically organize data in a lakehouse, with the goal of incrementally and progressively improving the structure and quality of data as it flows through each layer of the architecture (from Bronze ⇒ Silver ⇒ Gold layer tables). Medallion architectures are sometimes also referred to as "multi-hop" architectures.

https://www.databricks.com/glossary/medallion-architecture

WORKSPACE STRATEGY

- All-in-one
- One per stage
- Split by type
- One per workload

WORKSPACE STRATEGY

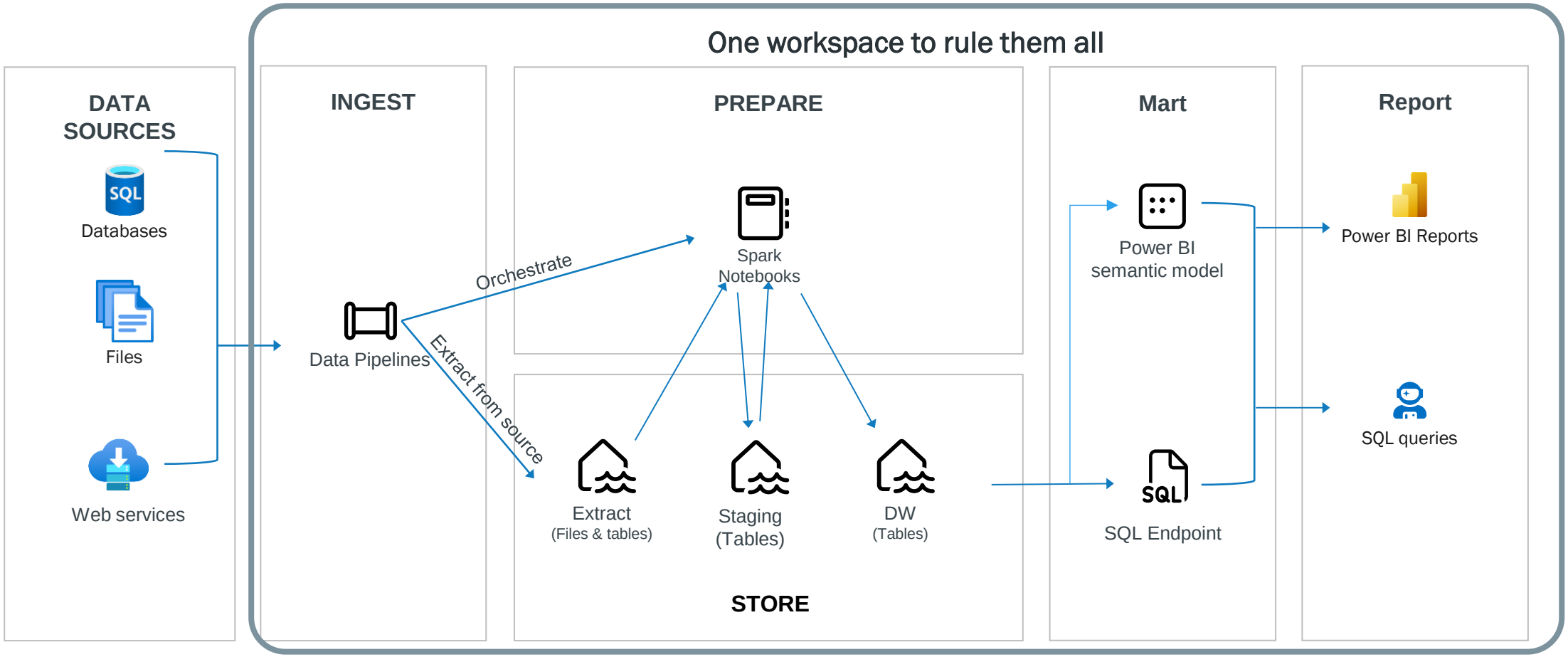
- Which factors determine the workspace strategy?
- How many workspaces do you need?
- How few workspaces can you get away with?

INFLUENCING FACTORS

- Number of environments (dev, test, pre-prod, prod etc.)
- Number of layers (landing, bronze, silver, curated, gold, mart, reports etc.)
- Size and composition of the development team
- CI/CD strategy
- Isolation of workloads (Data Factory, Lakehouse (Spark), Warehouse (SQL) etc.)
- Isolation of data

WORKSPACE STRATEGY

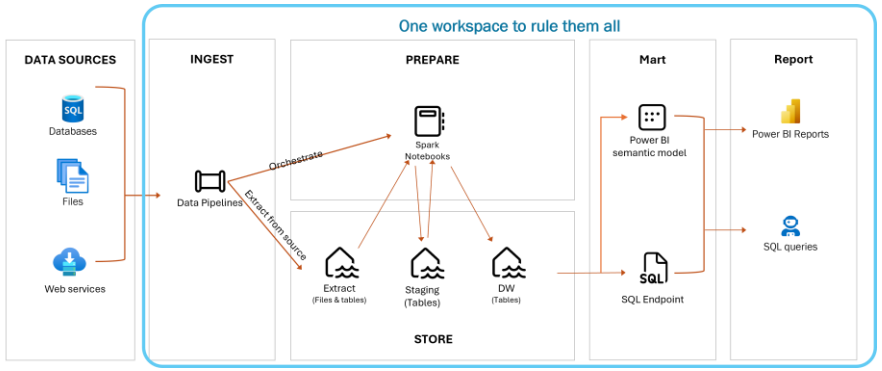
- Possible to use 1 workspace
  - But not very realistic



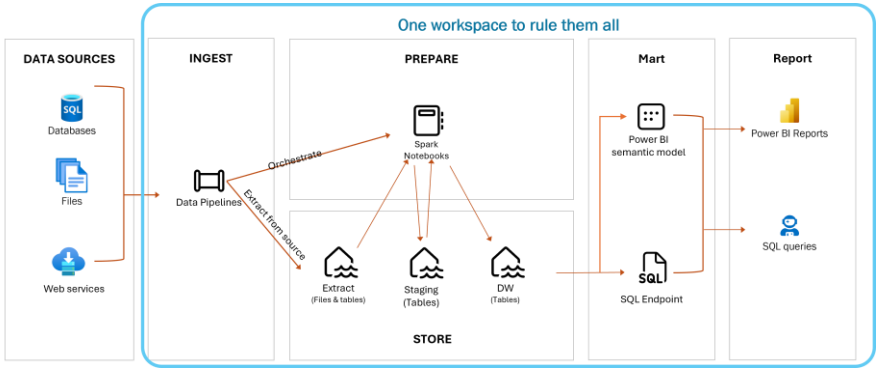
WORKSPACE STRATEGY

- Possible to use 1 workspace per environment (dev, test, prod osv.)
  - If we don´t need to isolate workloads or layers

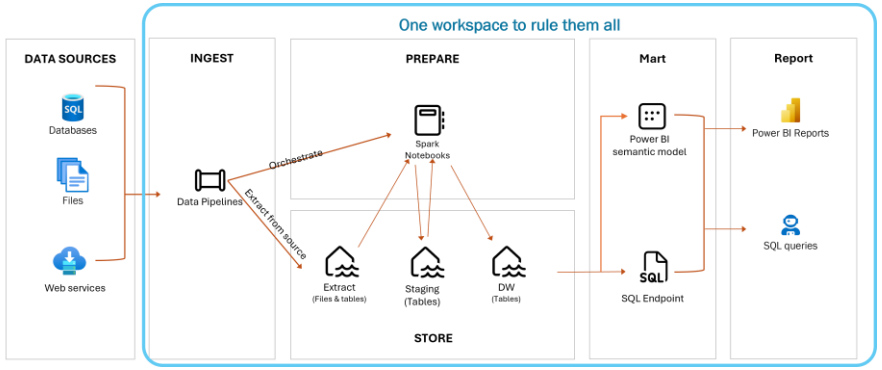
Development



Test

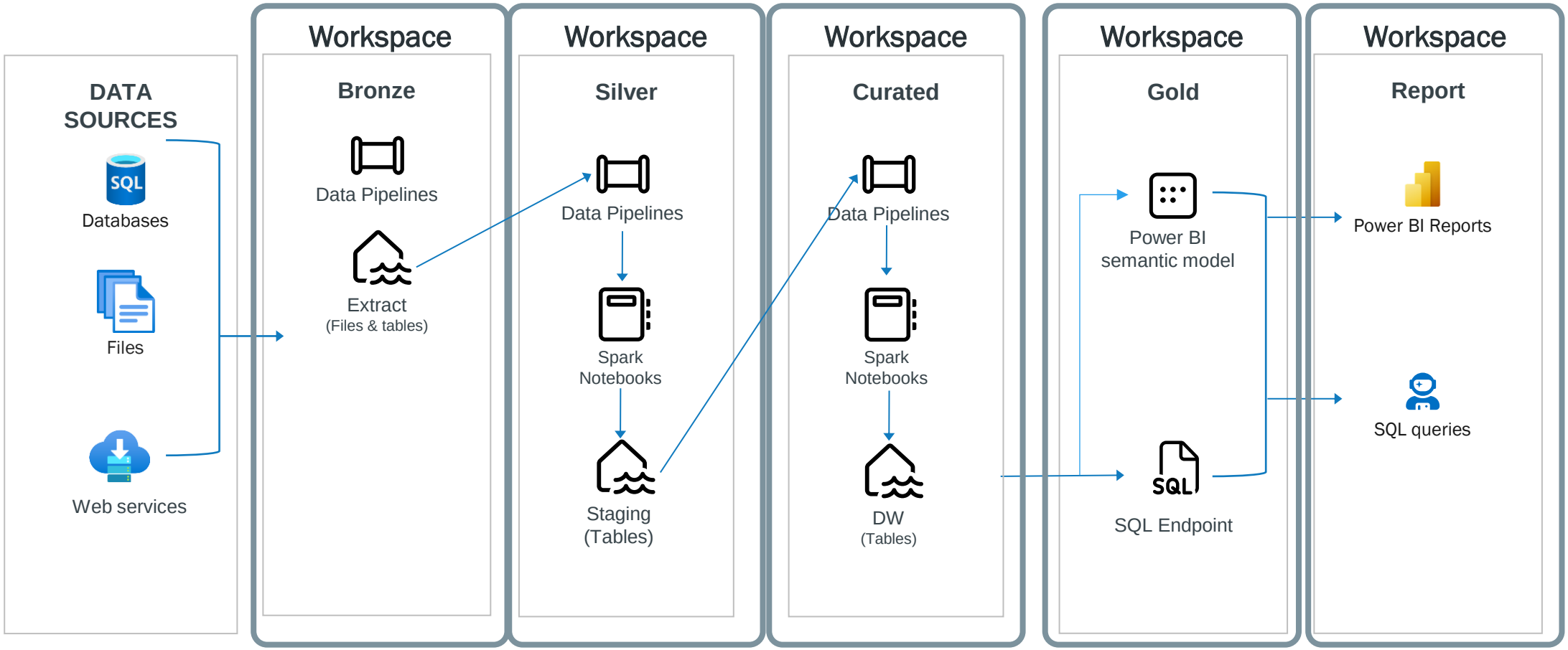


Production



WORKSPACE STRATEGY

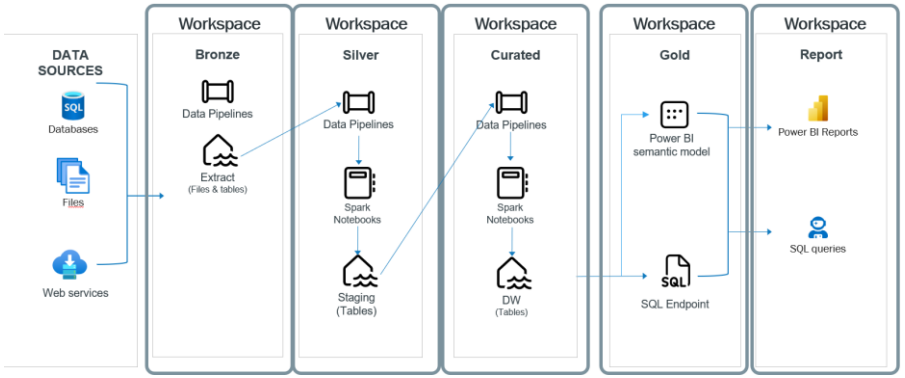
- Possible to use 1 workspace per layer (bronze, silver, curated, gold etc.)
  - The orchestrating Data Factory pipeline or notebook would live with either the source or target layer (push or pull)



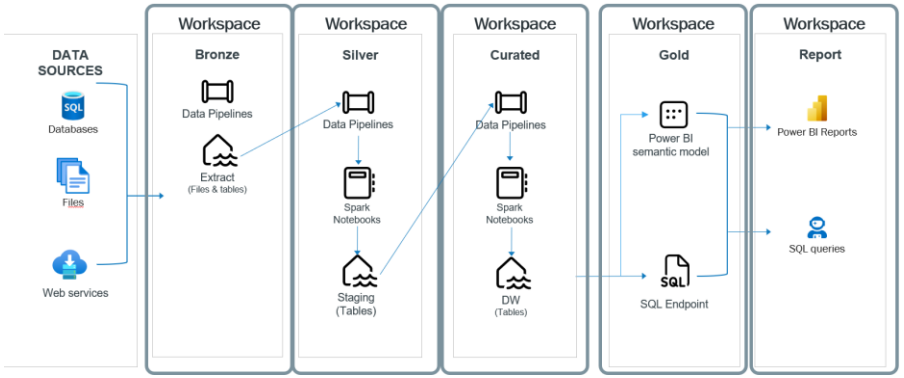
WORKSPACE STRATEGY

- Possible to use 1 workspace per environment per layer (bronze, silver, curated, gold etc.)
  - The orchestrating Data Factory pipeline or notebook would live with either the source or target layer (push or pull)

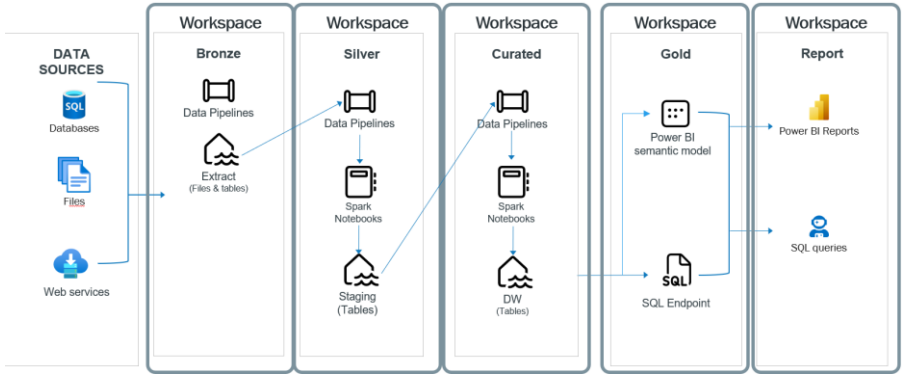
Development



Test

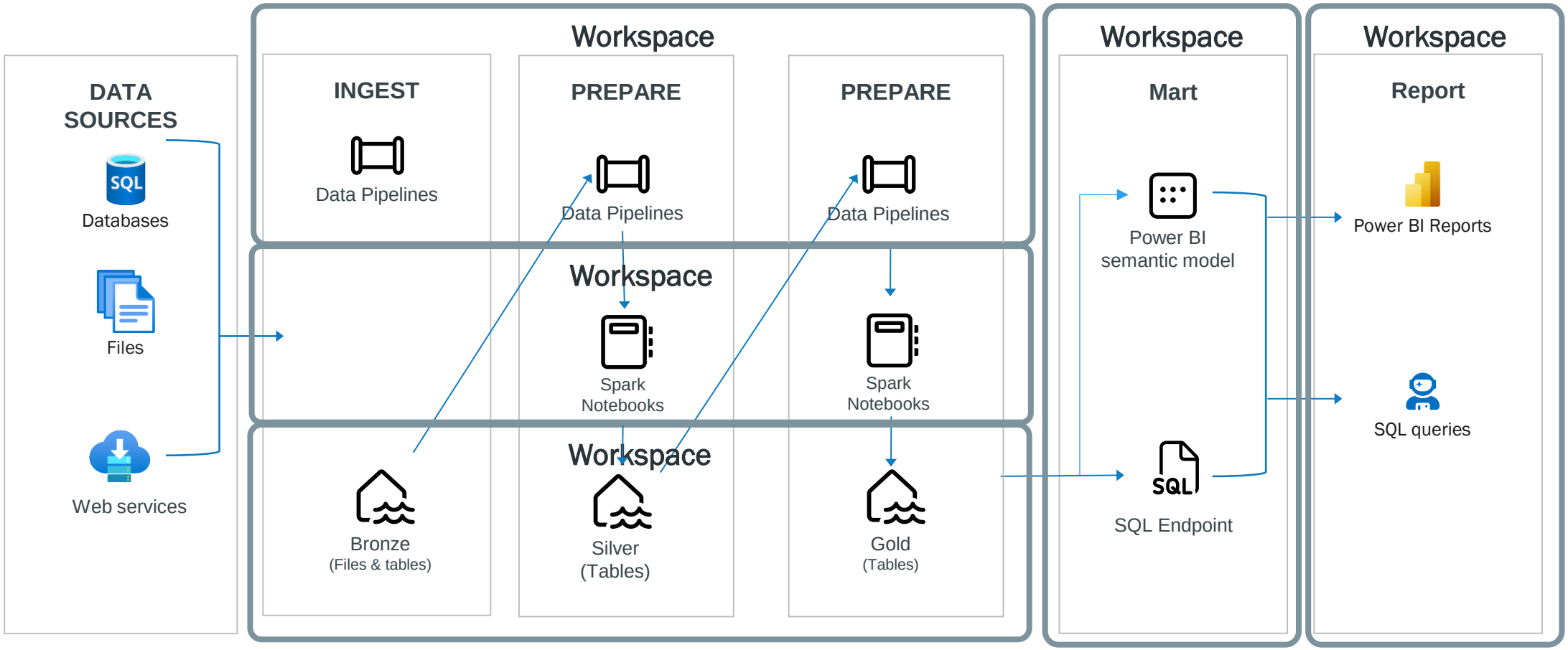


Production



WORKSPACE STRATEGY

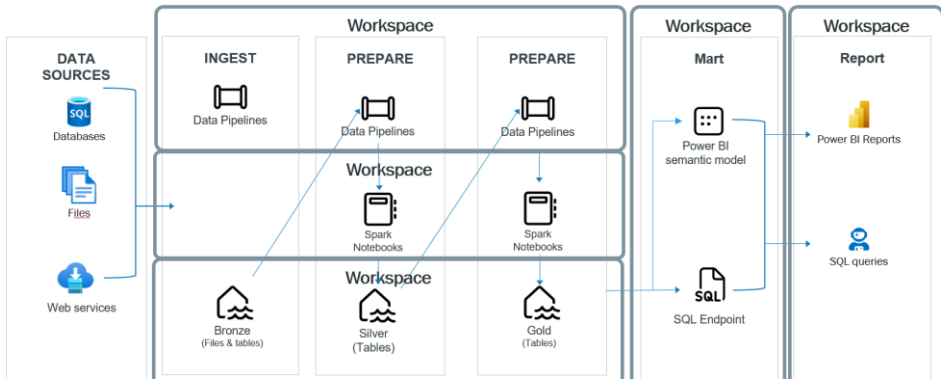
- Possible to use 1 workspace per workload (DF, Spark, Lakehouse etc.)



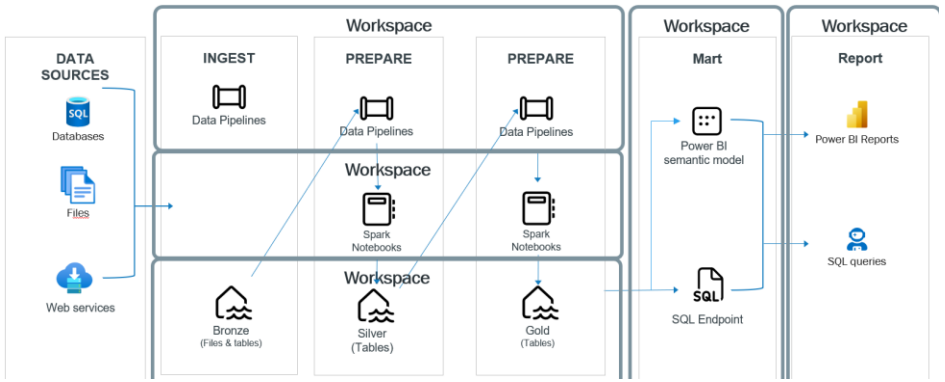
WORKSPACE STRATEGY

- Possible to use 1 workspace per environment per workload (DF, Spark, Lakehouse etc.)

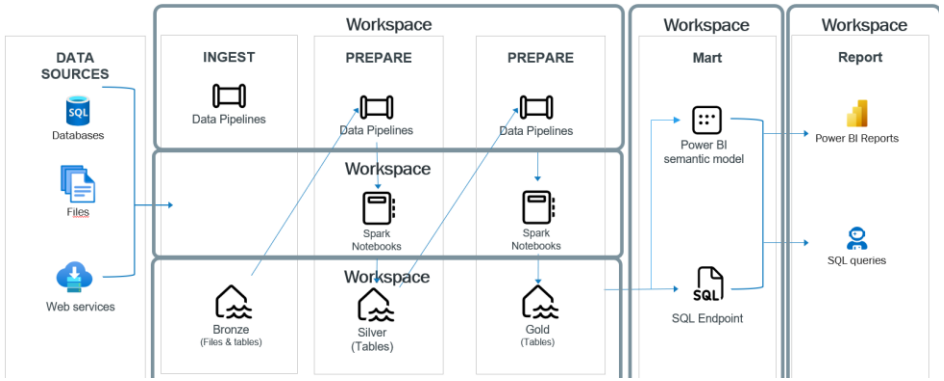
Development



Test



Production



WHERE TO STORE BUSINESS LOGIC

- Notebooks
- SQL Endpoint views
- Spark views
- Files or datastore outside of Fabric

LOGIC IN NOTEBOOKS

- Very explicit
- Can write the logic in verity of languages
- Might end up with a lot of notebooks
- Need to handle the default lakehouse issue

LOGIC IN SQL ENDPOINT VIEWS

- Very explicit
- Can write the logic in the familiar T-SQL language
- Can orchestrate/move the data via Data Factory or notebooks
  - If notebooks you need to deal with the unstable (currently) connector for Spark
- Might end up with a lot of views

LOGIC IN SPARK VIEWS

- Very explicit
- Can write the logic in verity of languages
- Can orchestrate/move the data via notebooks
- Might end up with a lot of views
- Views not visible in the user interface at all

LOGIC IN FILES OR DATASTORE

- Logic separated from workload
- Possible to have subject matter experts maintain the logic
- Not always easy to maintain the logic in files, lists or databases
- Needs a very good process so that the logic is entered correctly
- Needs good cooperation between data engineers and subject matter experts

IMPACT OF DEVOPS

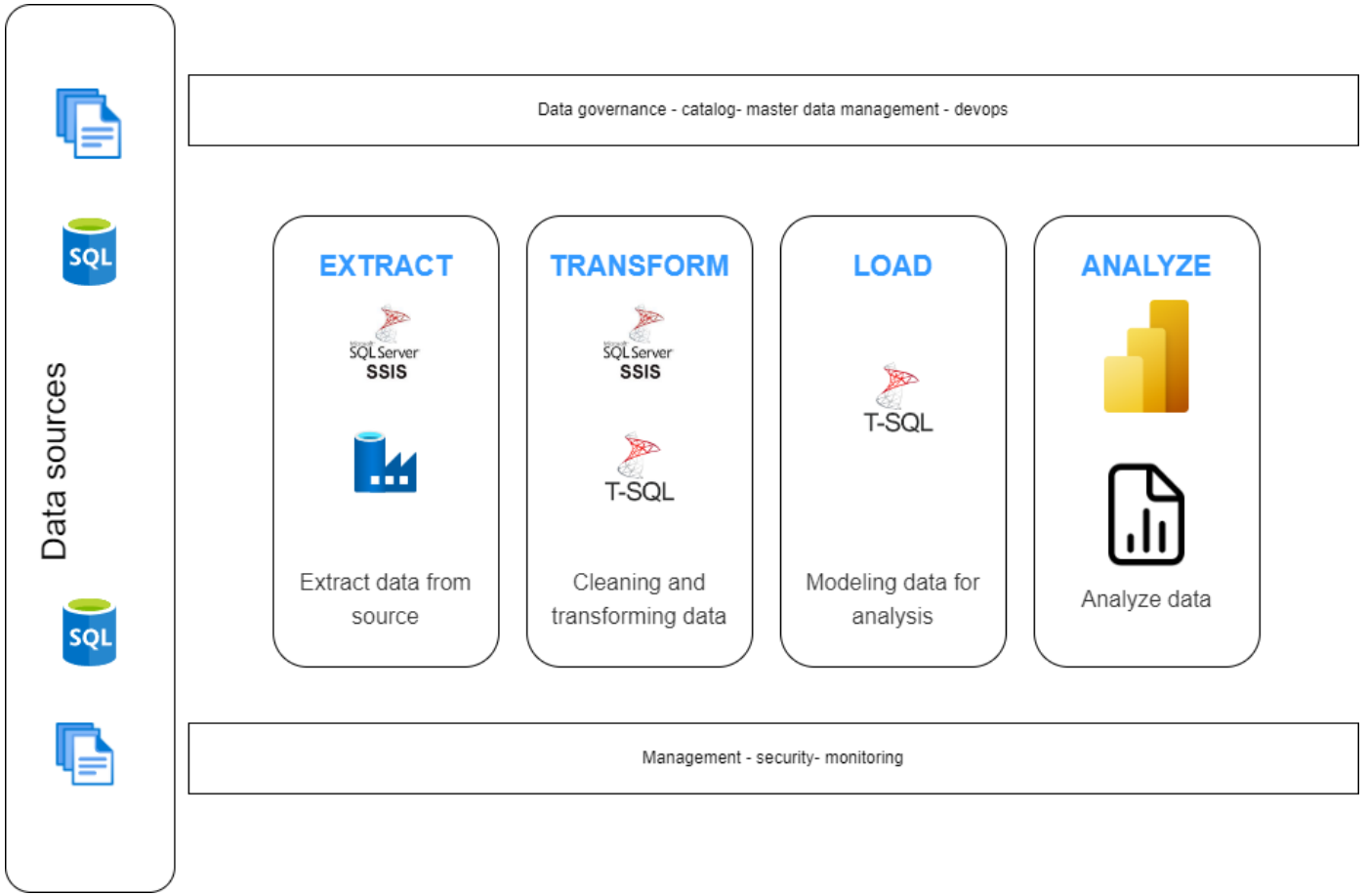
- Strategy for branching
  - No branches
  - Feature workspaces
  - One workspace for each developer
- How does that impact your pattern

WHAT IF I'M COMING FROM SQL

- Some things are harder
- Many things are easier
- Most thing are the same

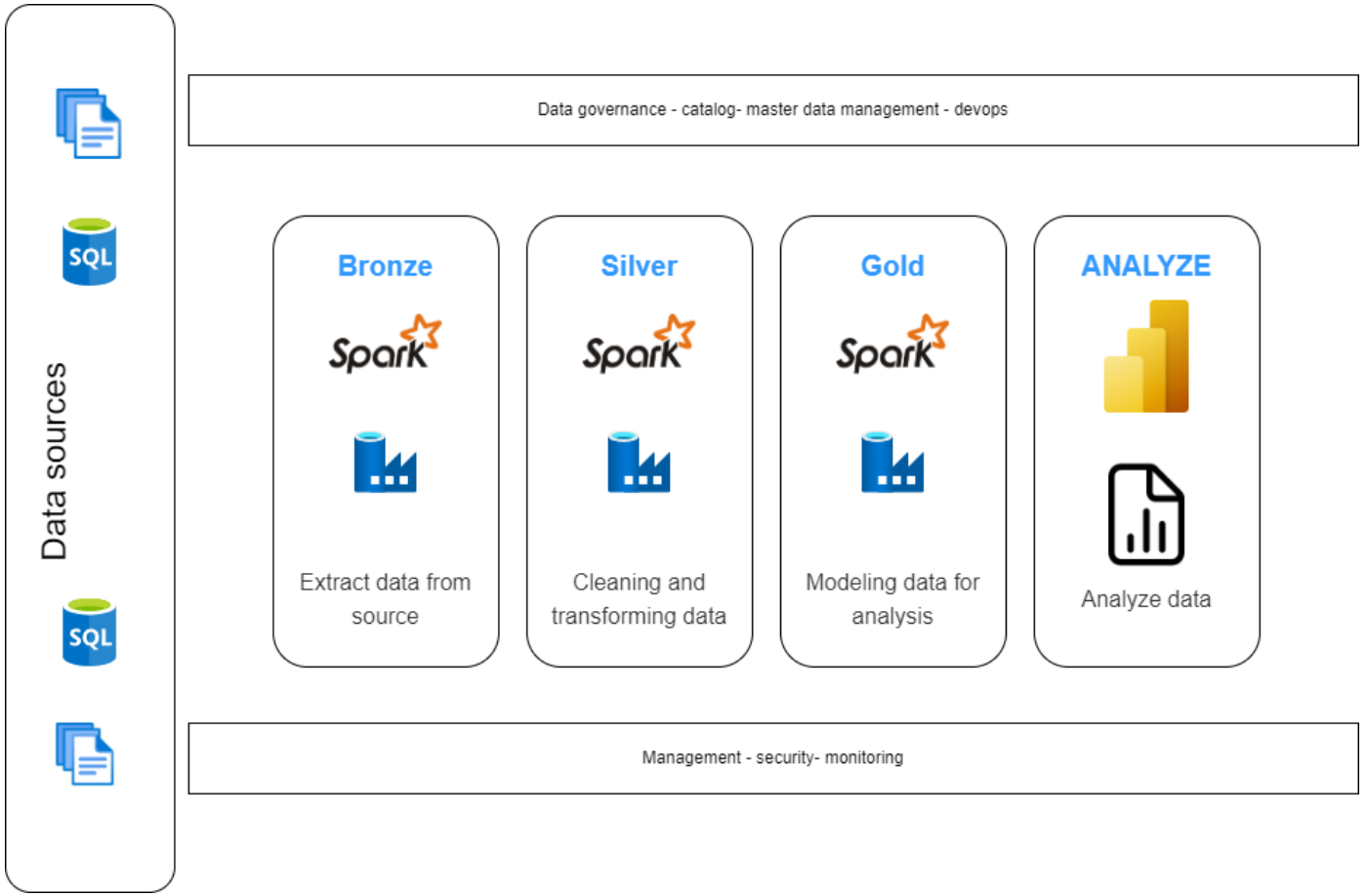
TYPICAL DATAWAREHOUSE IN SQL

- Data loaded in as is
- Data transformed via stored procedures or SSIS
- Data loaded into the Datawarehouse (schema or database)
- Data mart exposed via views or physical tables



DATA LAKEHOUSE

- Data loaded in as is (Bronze)
- Data transformed via Spark (Silver)
- Data loaded into the Lakehouse via Spark (Gold)
- Data mart exposed via SQL endpoints



CONCLUSIONS

- No one size fits all
- Think about the size of your team
- Think about how you want to do devops
- Think about the limitation of Fabric
- Examine the roadmap of Fabric