DATUMO SQL Day

Building scalable and automated Databricks-oriented Data Platform from scratch

Wojciech Pratkowiecki

DATUMO SQL Day

17th edition SQLDay Conference

12-14 May 2025, WROCŁAW + ONLINE

Platinum sponsors

DATUMO Woodler

Gold sponsors

lingaro VOLVO TECHNOLOGY INNOVATION DATA KNOWLEDGE tidk softwareone Dataedo

Silver sponsors

Capgemini DATA RISE LAB elitmind C&F kursy FABRIC

Wojciech Pratkowiecki

Project leader
Datumo

Agenda

- The story
- The challenge
- The solution
- Lessons learned

The story

Leader of CEE-pharmaceutical market with departments and subsidiaries across Europe

Production department

Regulatory department

Strategy department

R&D department

1. **Unify and centralize data pipelines and analysis**
2. **Standardize data management and processing**
3. **Provide tailored Data Platform to each tenant**
4. **Data mesh approach for datasets exchange**

The story

DATUMO

SQL Day

Tenant 1

Microsoft SQL Server

ORACLE

B

CSV

Tenant 2

Dynamics 365

ORACLE

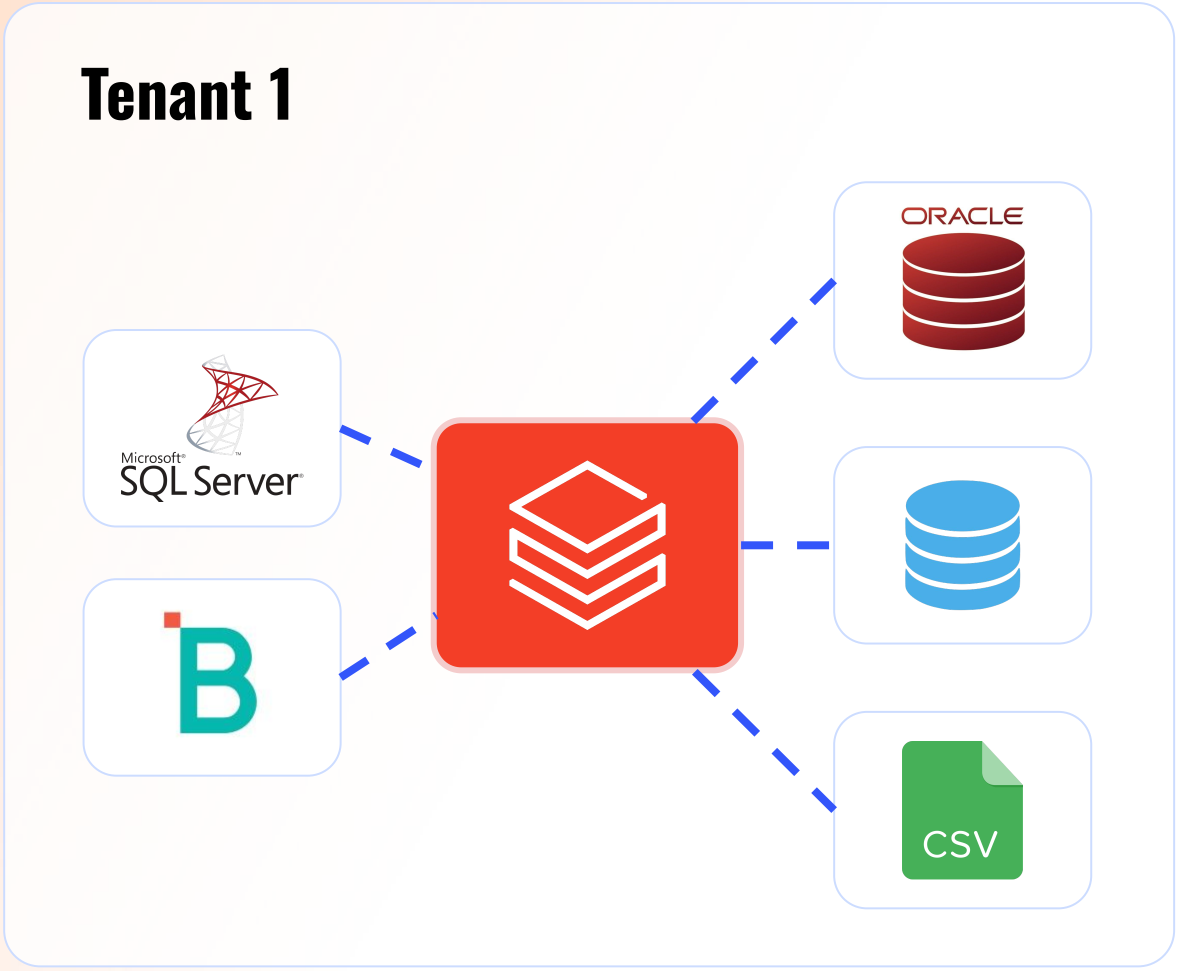
Informix

Tenant 3

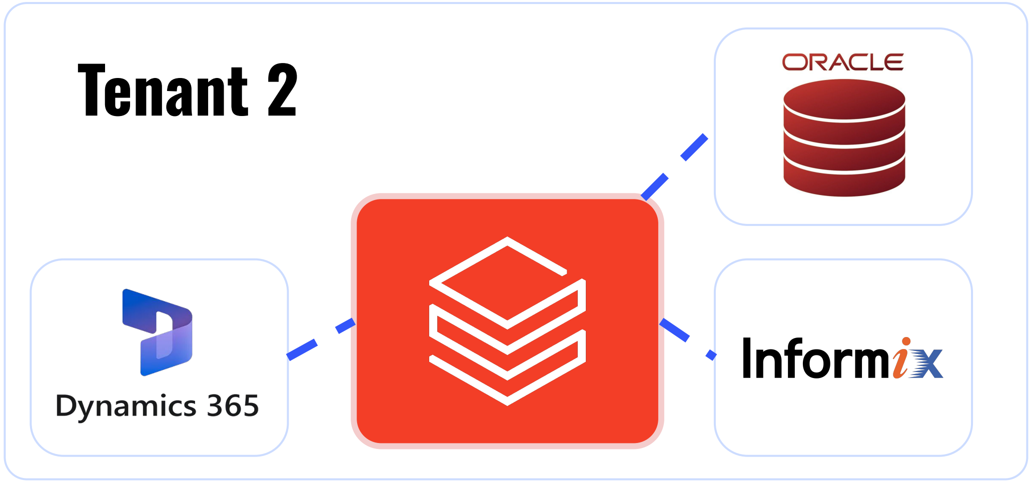
IQVIA™

d.

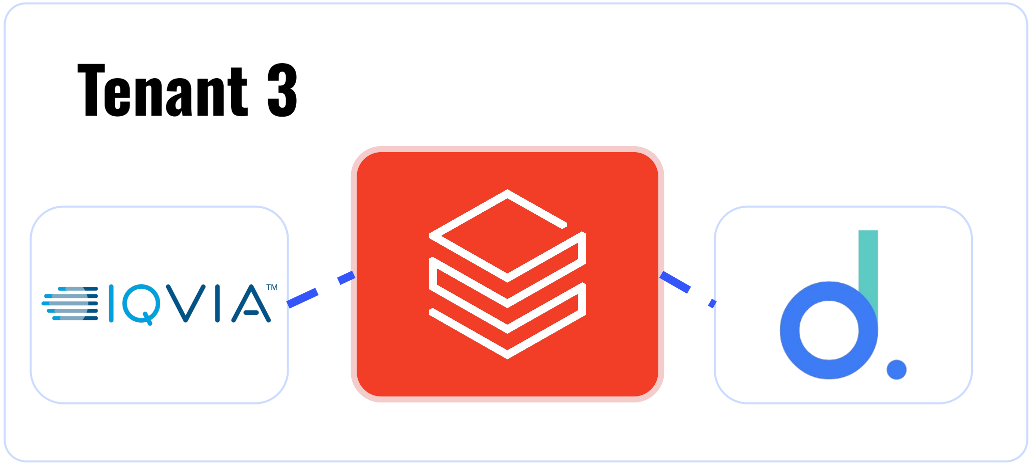
The story



Tenant 1
Microsoft SQL Server
ORACLE
CSV

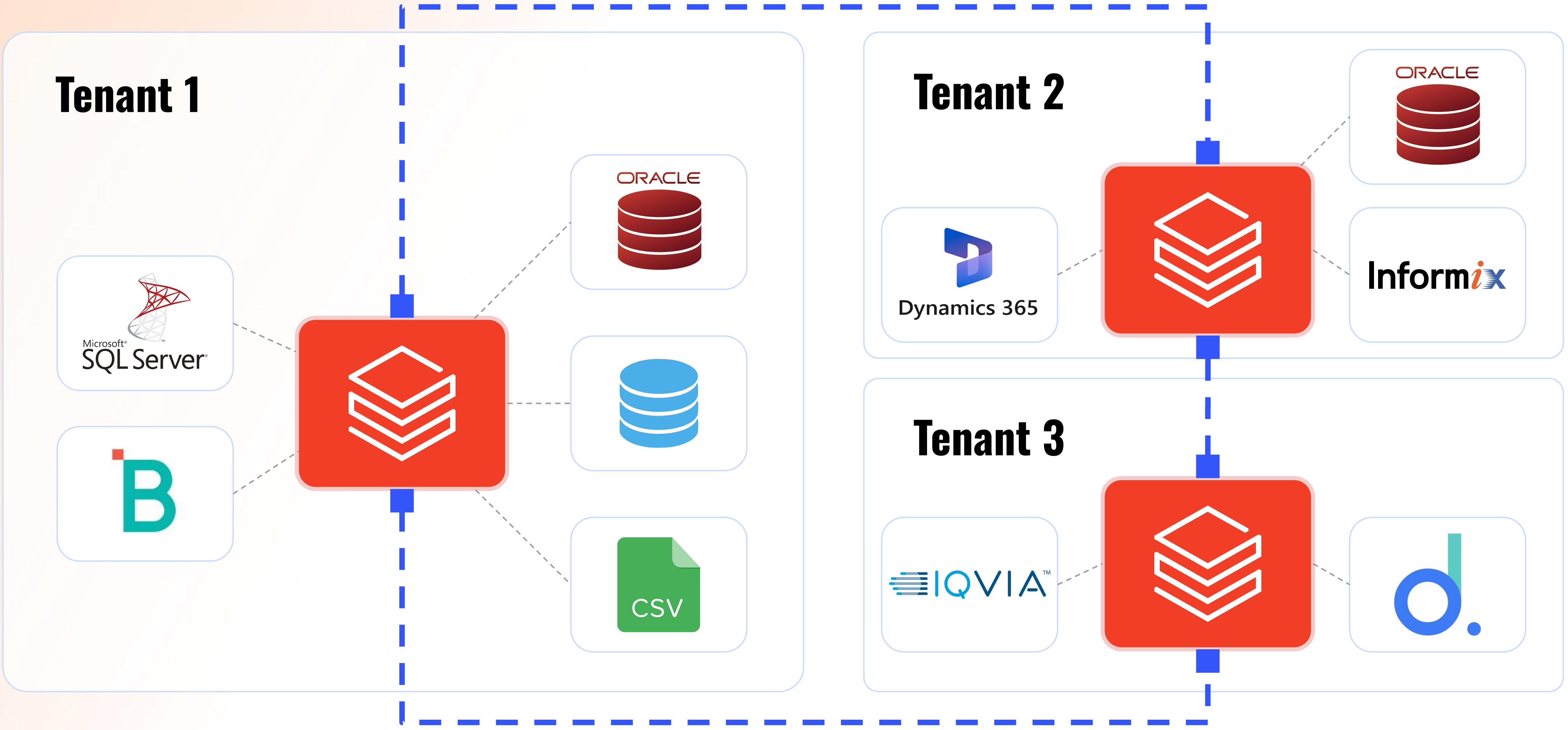


Tenant 2
Dynamics 365
ORACLE
Informix



Tenant 3
IQVIA

The story

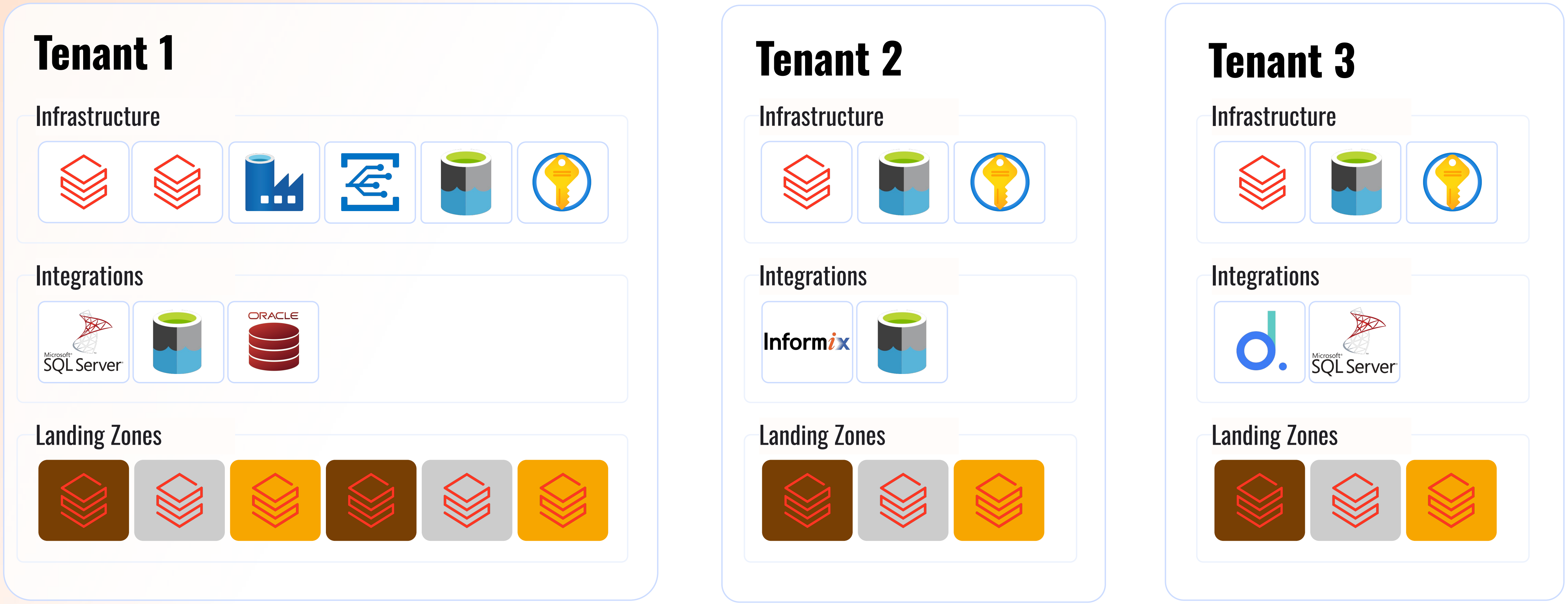

Tenant 1
Microsoft SQL Server
ORACLE
CSV
Tenant 2
Dynamics 365
ORACLE
Informix
Tenant 3
IQVIA

The challenge

- Datasets are spread across various systems
- Introduced platform must be standardized and scalable
- All the resources should be implemented in modular approach
- Tenant-specific adjustments to each of platform’s elements are required
- All of the Data Platform resources and processes need to be automatized
- Significant value for technical and non-technical users is expected

DATUMO SQL Day

The challenge



The solution

Functional Framework

Infrastructure Framework

Datastructure Framework

Pipelines Framework

CI/CD

The solution

DATUMO SQL Day

Functional Framework

- ADLS integration with Autoloader and native Spark
- JDBC sources integration in incremental/full mode
- Tenant-specific implementation for integrated sources
- Common functions used for Silver/Gold layer creation
- Utilities for analysis and notebooks development
- Data Quality facilities

The solution

Infrastructure Framework

- Terraform IaC implementation
- Blueprints of multi-service platform components
- Modules defining adjusted templates for services
- Simple reproduction across environments
- Access management
- Automated management of all platform resources

The solution

Datastructure Framework

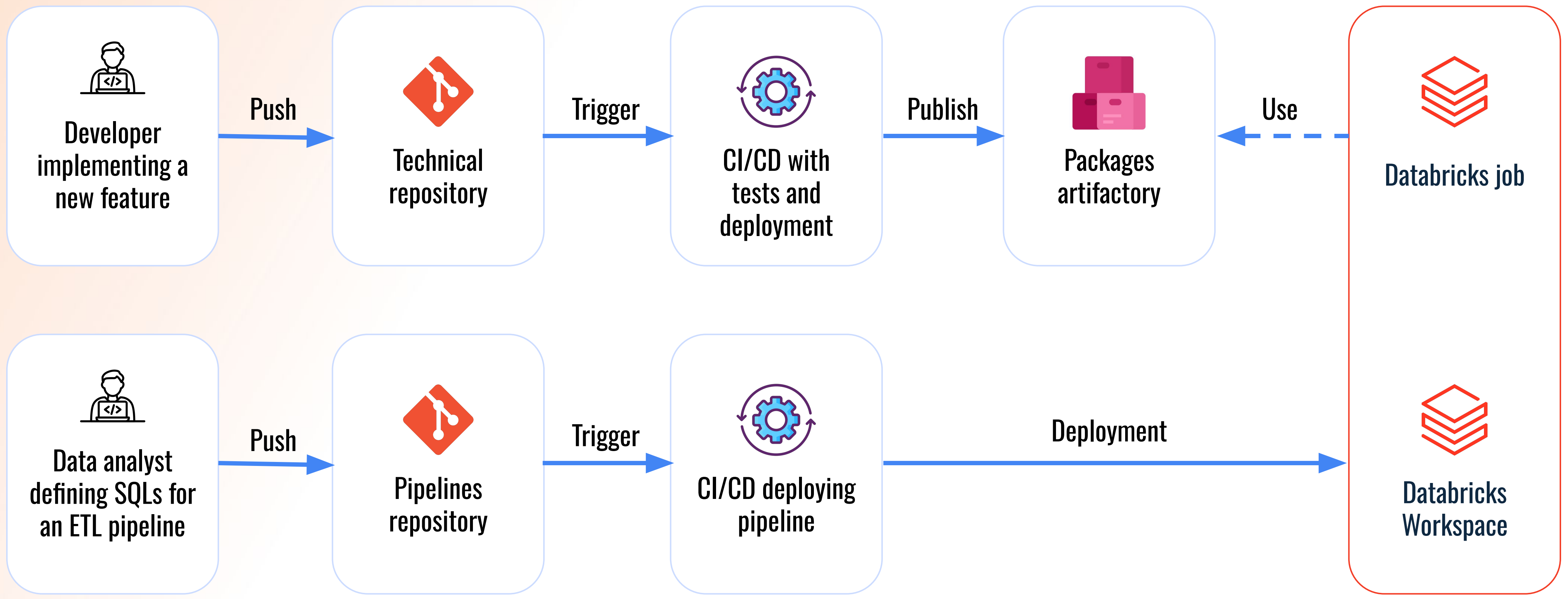
- Terraform modules for Unity Catalog resources management
- Definitions of particular catalogs/schemas
- Modules defining templates for catalogs, schemas, tables, external locations etc.
- Table’s schema evolution management
- UC resources access management
- Automated management of all UC resources

The solution

Pipelines Framework

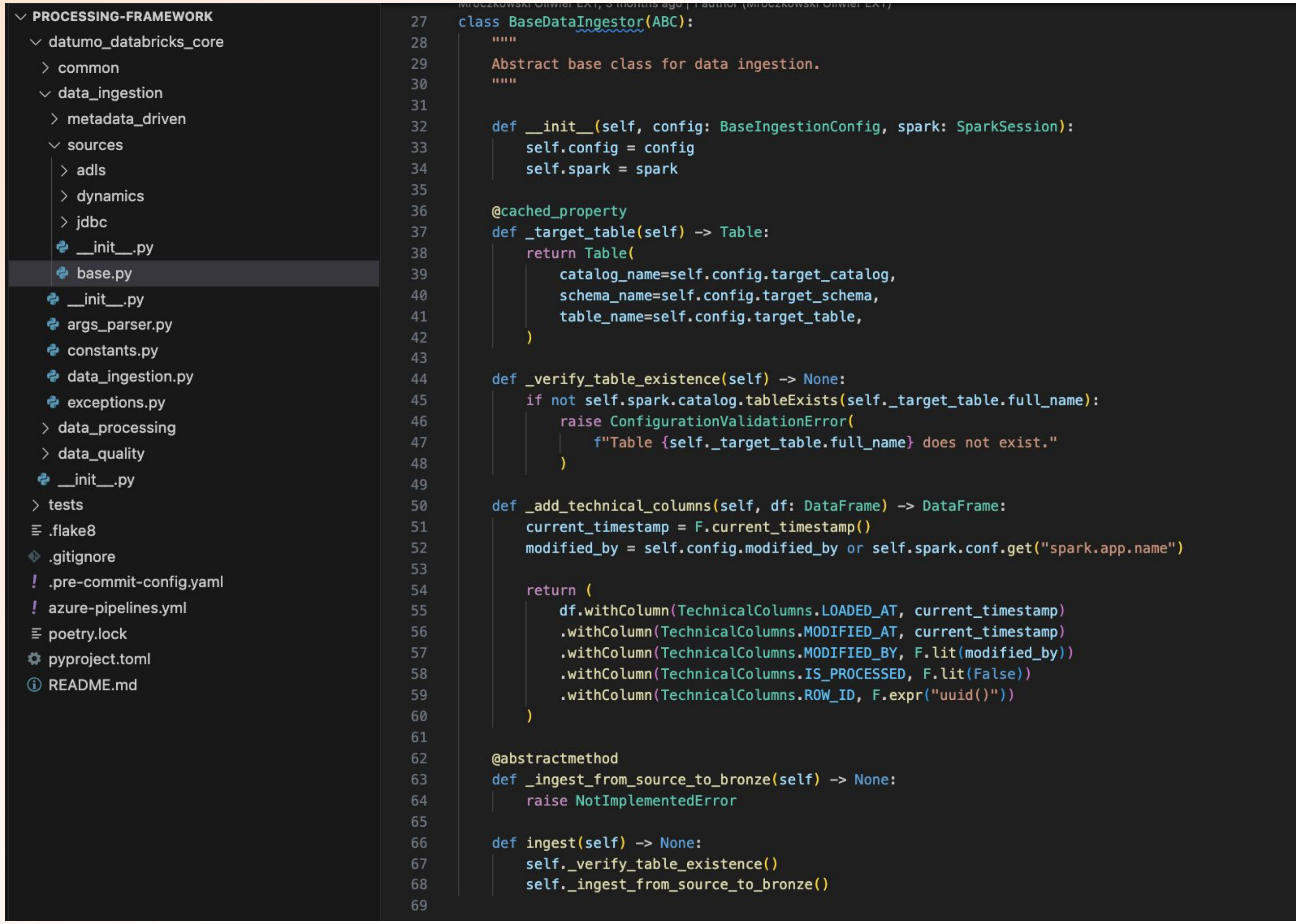
- Data pipelines as Databricks Workflows
- Management of cross-environment deployment
- Implemented and managed with Databricks Asset Bundles
- Custom configuration of jobs/clusters across environments
- CLI and templates simplifying pipeline creation - single YAML file to define a pipeline
- Automated deployment of Workflows to Workspaces

The solution



The solution

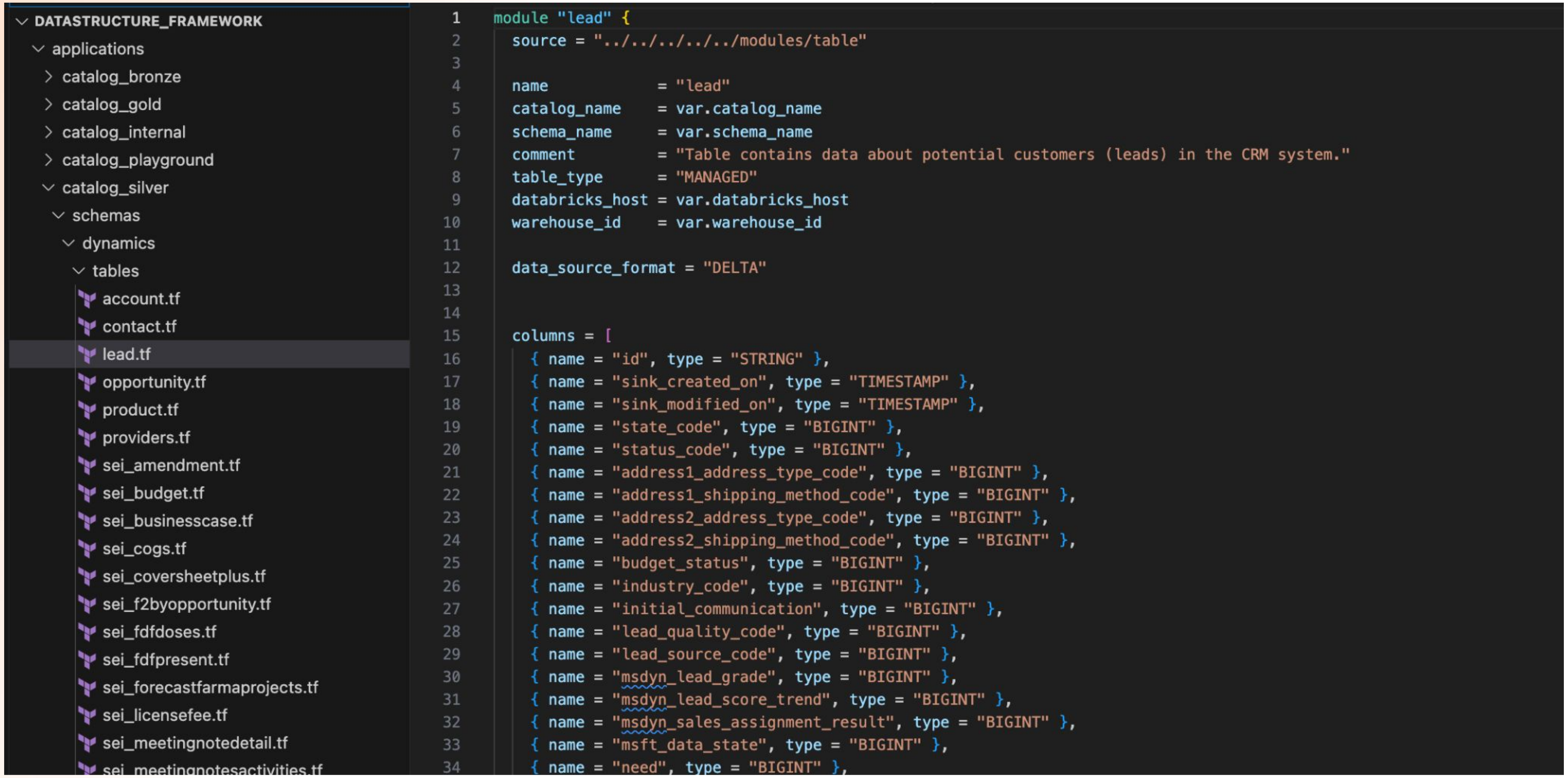
DATUMO SQL Day



```
PROCESSING-FRAMEWORK
  datumo_databricks_core
    common
    data_ingestion
      metadata_driven
      sources
        adls
        dynamics
        jdbc
        __init__.py
        base.py
      __init__.py
      args_parser.py
      constants.py
      data_ingestion.py
      exceptions.py
    data_processing
    data_quality
    __init__.py
  tests
  .flake8
  .gitignore
  .pre-commit-config.yaml
  azure-pipelines.yml
  poetry.lock
  pyproject.toml
  README.md
```

```python
class BaseDataIngestor(ABC):
    """
    Abstract base class for data ingestion.
    """

    def __init__(self, config: BaseIngestionConfig, spark: SparkSession):
        self.config = config
        self.spark = spark

    @cached_property
    def _target_table(self) -> Table:
        return Table(
            catalog_name=self.config.target_catalog,
            schema_name=self.config.target_schema,
            table_name=self.config.target_table,
        )

    def _verify_table_existence(self) -> None:
        if not self.spark.catalog.tableExists(self._target_table.full_name):
            raise ConfigurationValidationError(
                f"Table {self._target_table.full_name} does not exist."
            )

    def _add_technical_columns(self, df: DataFrame) -> DataFrame:
        current_timestamp = F.current_timestamp()
        modified_by = self.config.modified_by or self.spark.conf.get("spark.app.name")

        return (
            df.withColumn(TechnicalColumns.LOADED_AT, current_timestamp)
            .withColumn(TechnicalColumns.MODIFIED_AT, current_timestamp)
            .withColumn(TechnicalColumns.MODIFIED_BY, F.lit(modified_by))
            .withColumn(TechnicalColumns.IS_PROCESSED, F.lit(False))
            .withColumn(TechnicalColumns.ROW_ID, F.expr("uuid()"))
        )

    @abstractmethod
    def _ingest_from_source_to_bronze(self) -> None:
        raise NotImplementedError

    def ingest(self) -> None:
        self._verify_table_existence()
        self._ingest_from_source_to_bronze()
```

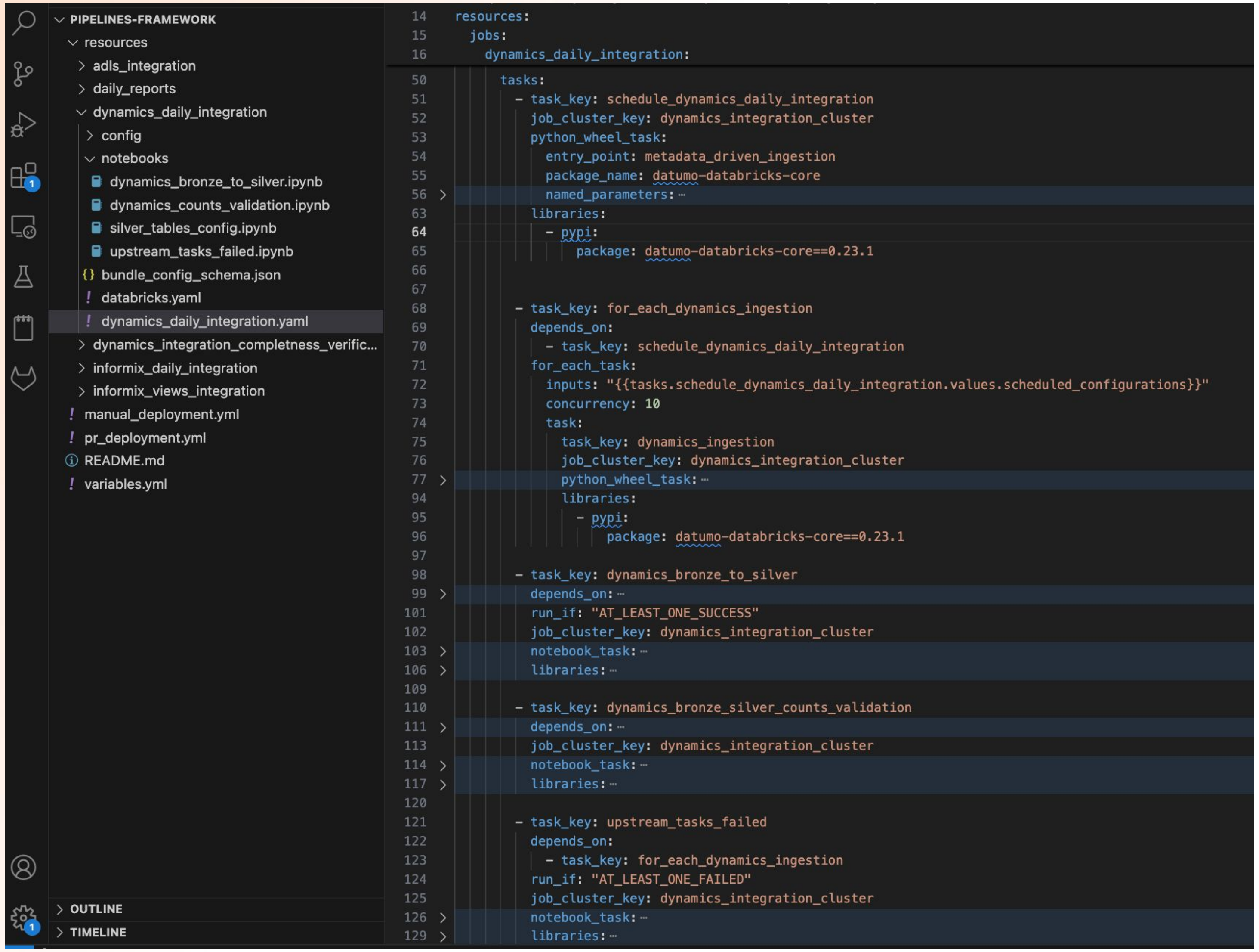
The solution



```
DATASTRUCTURE_FRAMEWORK
  applications
    catalog_bronze
    catalog_gold
    catalog_internal
    catalog_playground
    catalog_silver
      schemas
        dynamics
          tables
            account.tf
            contact.tf
            lead.tf
            opportunity.tf
            product.tf
            providers.tf
            sei_amendment.tf
            sei_budget.tf
            sei_businesscase.tf
            sei_cogs.tf
            sei_coversheetplus.tf
            sei_f2byopportunity.tf
            sei_fdfdoses.tf
            sei_fdfpresent.tf
            sei_forecastfarmaprojects.tf
            sei_licensefee.tf
            sei_meetingnotedetail.tf
            sei_meetingnotesactivities.tf
```

```
module "lead" {
  source = "../../../../../modules/table"

  name            = "lead"
  catalog_name    = var.catalog_name
  schema_name     = var.schema_name
  comment         = "Table contains data about potential customers (leads) in the CRM system."
  table_type      = "MANAGED"
  databricks_host = var.databricks_host
  warehouse_id    = var.warehouse_id

  data_source_format = "DELTA"


  columns = [
    { name = "id", type = "STRING" },
    { name = "sink_created_on", type = "TIMESTAMP" },
    { name = "sink_modified_on", type = "TIMESTAMP" },
    { name = "state_code", type = "BIGINT" },
    { name = "status_code", type = "BIGINT" },
    { name = "address1_address_type_code", type = "BIGINT" },
    { name = "address1_shipping_method_code", type = "BIGINT" },
    { name = "address2_address_type_code", type = "BIGINT" },
    { name = "address2_shipping_method_code", type = "BIGINT" },
    { name = "budget_status", type = "BIGINT" },
    { name = "industry_code", type = "BIGINT" },
    { name = "initial_communication", type = "BIGINT" },
    { name = "lead_quality_code", type = "BIGINT" },
    { name = "lead_source_code", type = "BIGINT" },
    { name = "msdyn_lead_grade", type = "BIGINT" },
    { name = "msdyn_lead_score_trend", type = "BIGINT" },
    { name = "msdyn_sales_assignment_result", type = "BIGINT" },
    { name = "msft_data_state", type = "BIGINT" },
    { name = "need", type = "BIGINT" },
```

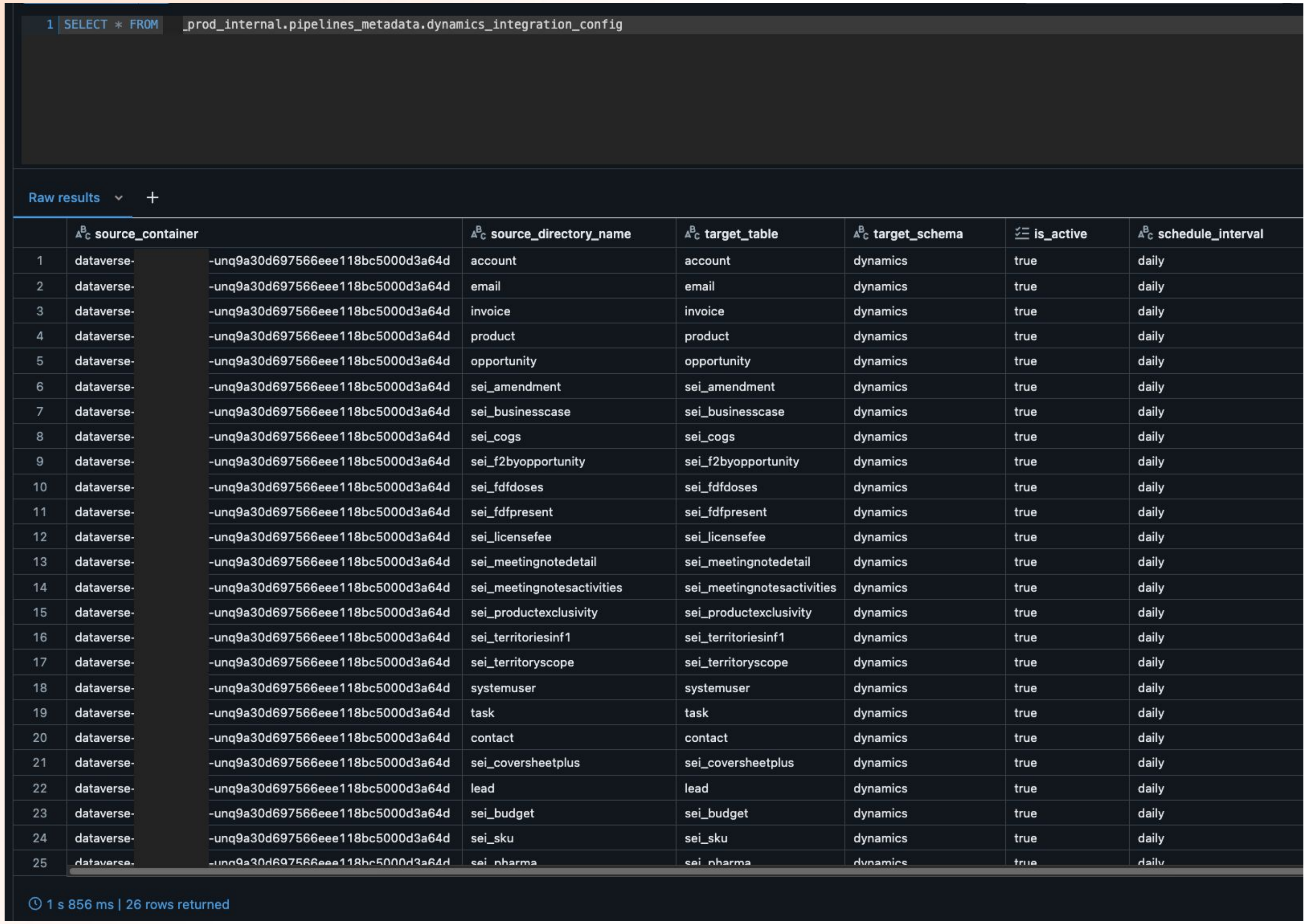
The solution



```
PIPELINES-FRAMEWORK
  resources
    adls_integration
    daily_reports
    dynamics_daily_integration
      config
      notebooks
        dynamics_bronze_to_silver.ipynb
        dynamics_counts_validation.ipynb
        silver_tables_config.ipynb
        upstream_tasks_failed.ipynb
      bundle_config_schema.json
      databricks.yaml
      dynamics_daily_integration.yaml
    dynamics_integration_completness_verific...
    informix_daily_integration
    informix_views_integration
  manual_deployment.yml
  pr_deployment.yml
  README.md
  variables.yml
OUTLINE
TIMELINE
```

```yaml
resources:
  jobs:
    dynamics_daily_integration:
      tasks:
        - task_key: schedule_dynamics_daily_integration
          job_cluster_key: dynamics_integration_cluster
          python_wheel_task:
            entry_point: metadata_driven_ingestion
            package_name: datumo-databricks-core
            named_parameters: ⋯
          libraries:
            - pypi:
                package: datumo-databricks-core==0.23.1


        - task_key: for_each_dynamics_ingestion
          depends_on:
            - task_key: schedule_dynamics_daily_integration
          for_each_task:
            inputs: "{{tasks.schedule_dynamics_daily_integration.values.scheduled_configurations}}"
            concurrency: 10
            task:
              task_key: dynamics_ingestion
              job_cluster_key: dynamics_integration_cluster
              python_wheel_task: ⋯
              libraries:
                - pypi:
                    package: datumo-databricks-core==0.23.1

        - task_key: dynamics_bronze_to_silver
          depends_on: ⋯
          run_if: "AT_LEAST_ONE_SUCCESS"
          job_cluster_key: dynamics_integration_cluster
          notebook_task: ⋯
          libraries: ⋯

        - task_key: dynamics_bronze_silver_counts_validation
          depends_on: ⋯
          job_cluster_key: dynamics_integration_cluster
          notebook_task: ⋯
          libraries: ⋯

        - task_key: upstream_tasks_failed
          depends_on:
            - task_key: for_each_dynamics_ingestion
          run_if: "AT_LEAST_ONE_FAILED"
          job_cluster_key: dynamics_integration_cluster
          notebook_task: ⋯
          libraries: ⋯
```

The solution



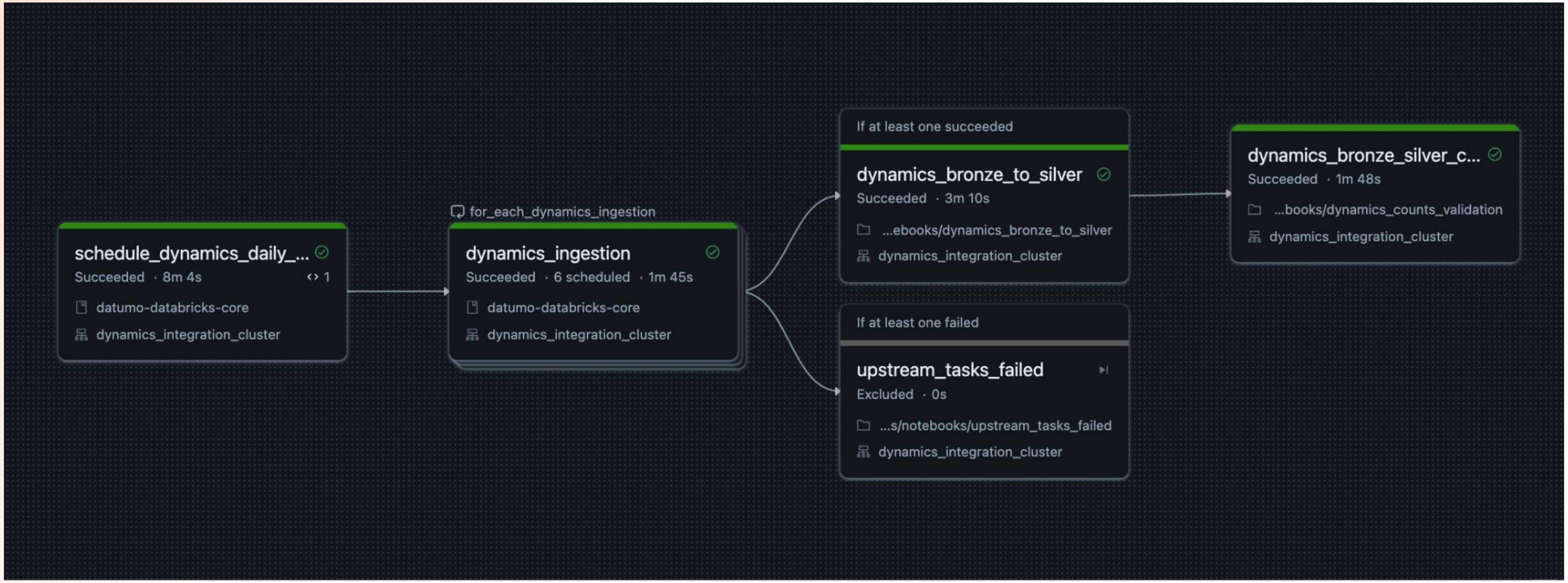
```
SELECT * FROM [redacted]_prod_internal.pipelines_metadata.dynamics_integration_config
```

Raw results

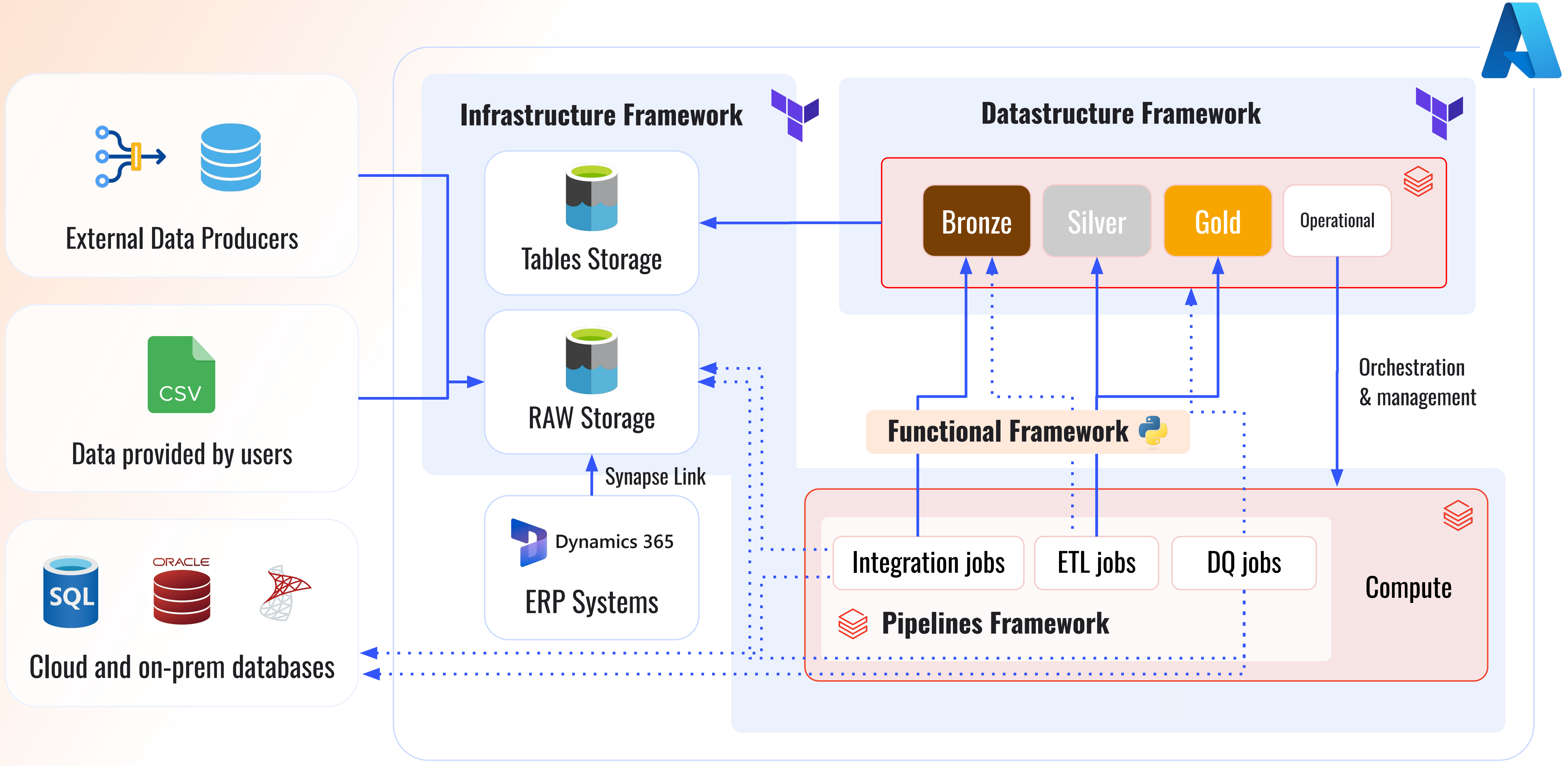
	source_container	source_directory_name	target_table	target_schema	is_active	schedule_interval
1	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	account	account	dynamics	true	daily
2	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	email	email	dynamics	true	daily
3	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	invoice	invoice	dynamics	true	daily
4	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	product	product	dynamics	true	daily
5	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	opportunity	opportunity	dynamics	true	daily
6	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	sei_amendment	sei_amendment	dynamics	true	daily
7	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	sei_businesscase	sei_businesscase	dynamics	true	daily
8	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	sei_cogs	sei_cogs	dynamics	true	daily
9	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	sei_f2byopportunity	sei_f2byopportunity	dynamics	true	daily
10	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	sei_fdfdoses	sei_fdfdoses	dynamics	true	daily
11	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	sei_fdfpresent	sei_fdfpresent	dynamics	true	daily
12	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	sei_licensefee	sei_licensefee	dynamics	true	daily
13	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	sei_meetingnotedetail	sei_meetingnotedetail	dynamics	true	daily
14	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	sei_meetingnotesactivities	sei_meetingnotesactivities	dynamics	true	daily
15	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	sei_productexclusivity	sei_productexclusivity	dynamics	true	daily
16	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	sei_territoriesinf1	sei_territoriesinf1	dynamics	true	daily
17	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	sei_territoryscope	sei_territoryscope	dynamics	true	daily
18	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	systemuser	systemuser	dynamics	true	daily
19	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	task	task	dynamics	true	daily
20	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	contact	contact	dynamics	true	daily
21	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	sei_coversheetplus	sei_coversheetplus	dynamics	true	daily
22	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	lead	lead	dynamics	true	daily
23	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	sei_budget	sei_budget	dynamics	true	daily
24	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	sei_sku	sei_sku	dynamics	true	daily
25	dataverse-[redacted]-unq9a30d697566eee118bc5000d3a64d	sei_pharma	sei_pharma	dynamics	true	daily

1 s 856 ms | 26 rows returned

The solution


schedule_dynamics_daily_...
Succeeded · 8m 4s
<> 1
datumo-databricks-core
dynamics_integration_cluster
for_each_dynamics_ingestion
dynamics_ingestion
Succeeded · 6 scheduled · 1m 45s
datumo-databricks-core
dynamics_integration_cluster
If at least one succeeded
dynamics_bronze_to_silver
Succeeded · 3m 10s
...ebooks/dynamics_bronze_to_silver
dynamics_integration_cluster
If at least one failed
upstream_tasks_failed
Excluded · 0s
...s/notebooks/upstream_tasks_failed
dynamics_integration_cluster
dynamics_bronze_silver_c...
Succeeded · 1m 48s
...books/dynamics_counts_validation
dynamics_integration_cluster

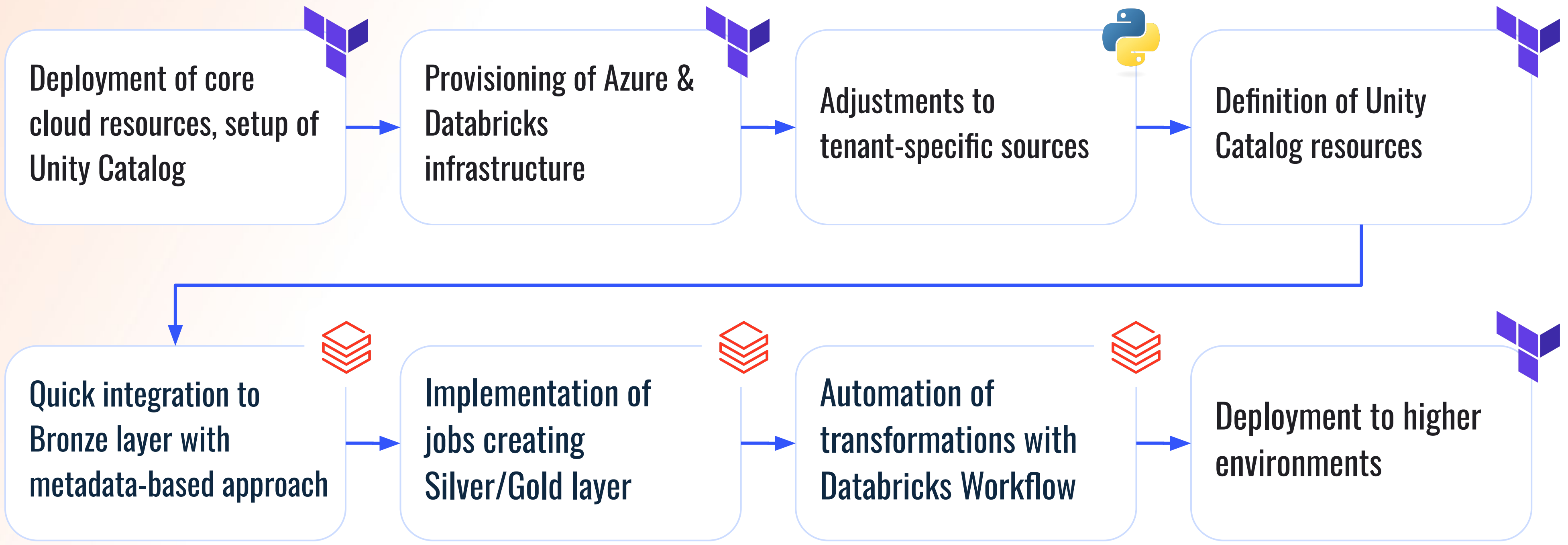
The solution



The solution

DATUMO SQL Day

Short Time To Market for each tenant


Deployment of core cloud resources, setup of Unity Catalog
Provisioning of Azure & Databricks infrastructure
Adjustments to tenant-specific sources
Definition of Unity Catalog resources
Quick integration to Bronze layer with metadata-based approach
Implementation of jobs creating Silver/Gold layer
Automation of transformations with Databricks Workflow
Deployment to higher environments

The solution

Provided value

Centralized data platform enabling cross-source system analysis

Standardized data processing and management across numerous tenants

Automated and extendable solution allowing quick integration of new sources

Defining advanced data pipelines utilizing powerful Databricks features made easy

Modular and scalable solution enabling quick provisioning of adjusted Data Platform

Lessons learned

- Define the data usage patterns with target users at the very beginning
- Short TTM introducing numerous novel services can be overwhelming
- Infrastructure and data structure automated management pays off
- Lack of date-based scheduling in Databricks hurts, but can be overcomed
- Strict schema management can be cumbersome, however smart scripts/notebooks can be an answer

DATUMO SQL Day

Q&A

DATUMO SQL Day

Thank you for your attention