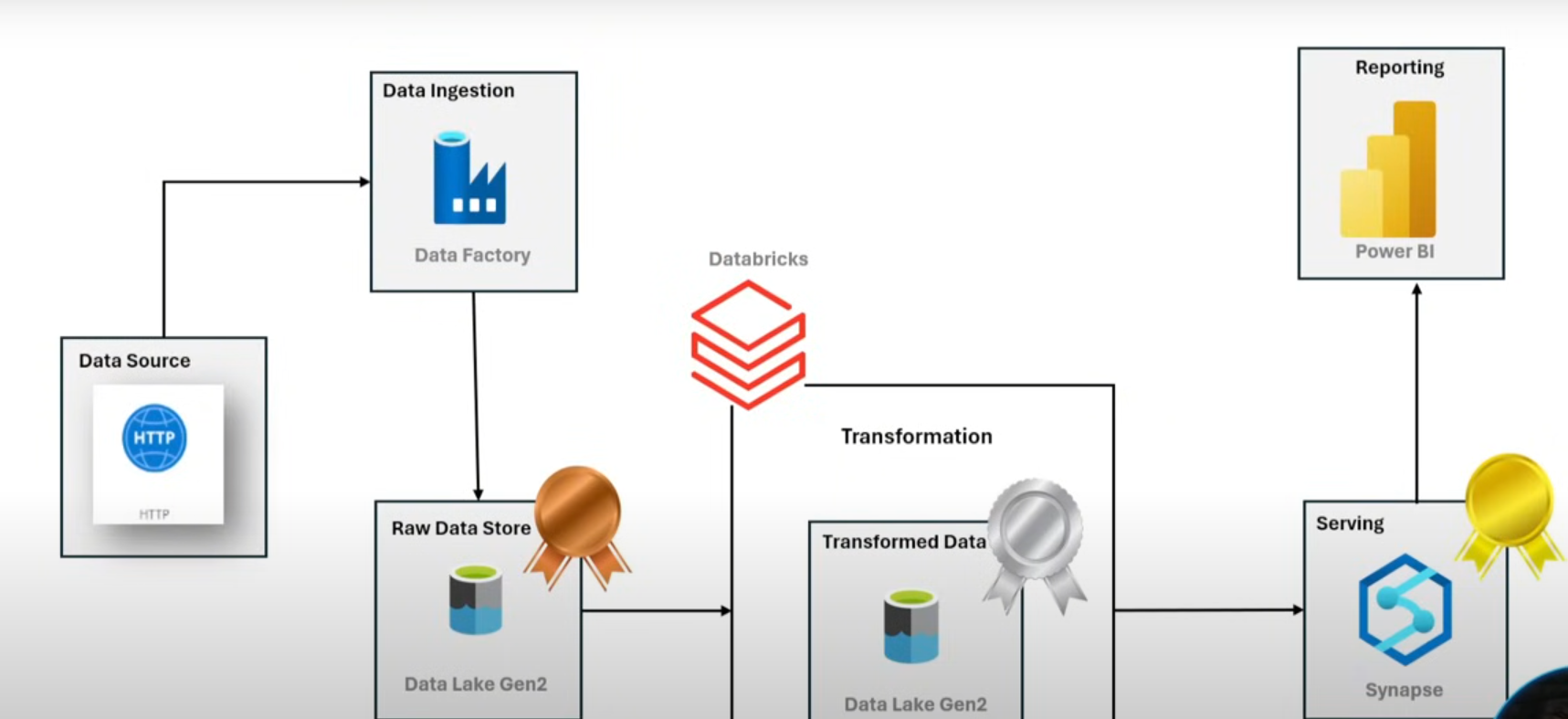
**Azure Data Engineering Project Documentation**

**Data Architecture**



**Data Source:** Fetching data from API

**Azure Data Factory:** Orchestration tool used to pull source data and land it in the bronze layer. Activities performed include:

* Building static and dynamic pipelines
* Using parameters
* Implementing loops
* Utilizing lookup activities

**Medallion Architecture**

Data traverses through three different zones:

1. **Raw Layer (Bronze):** Stores data as-is from the source.
2. **Transform Layer (Silver):** Data is transformed and pushed from the bronze layer to the silver layer.
3. **Serving Layer (Gold):** Data is served to stakeholders (analysts, data scientists) using Power BI dashboards.

**Dataset:** Adventure Works

**Steps to Create Resources and Implement Pipelines**

**Creating a Resource Group**

* Use tags to categorize resources (optional).

**Create a Data Lake (Storage Account)**

1. Set redundancy to LRS (Local Redundant Storage).
2. Enable hierarchical namespace for Data Lake functionality.
3. Difference between Blob and Data Lake:
   * Blob: Does not support folder hierarchy.
   * Data Lake: Supports folder hierarchy.

**Create Azure Data Factory**

1. Navigate to the **Author** tab for pipeline creation.
2. Use the **Monitor** tab to check pipeline status and debug errors.
3. Use the **Manage** tab for repositories, GitHub, and DevOps connections.
4. Build blocks such as pipelines, activities, functions, notebooks.

**Create Containers in Data Lake**

1. Create containers for bronze, silver, and gold layers.
2. Load data into the bronze layer using linked services.

**Create Linked Services**

1. **First Linked Service:** HTTP
2. **Second Linked Service:** ADLS GEN2 (Data Lake Storage)
   * Test connections to ensure they are established correctly.

**Create Datasets**

* For the source:
  + Format: CSV
  + Linked service: HTTP
  + Provide relative URL and set the first row as the header.
* For the sink:
  + Format: CSV
  + Linked service: ADLS GEN2

**Static Pipeline Implementation**

1. Use **Copy Activity** to load data from the source to the sink.
2. Debug and publish the pipeline.

**Dynamic Pipeline Implementation**

1. Create parameterized datasets to handle multiple files.
2. Use JSON to iterate over file names and store them in a new container.
3. Implement **Lookup Activity** to view file outputs.

**Storage Access via Entra ID**

1. Navigate to Entra ID and create an app registration.
2. Store the client secret and assign roles for read/write access.

**Phase 2: Transformations and Databricks Integration**

**Databricks Setup**

1. Create a Databricks resource in Azure.
2. Launch the Databricks UI and create a compute cluster.
3. Use Databricks to ingest data from the raw layer into the silver layer with transformations.

**Assign Roles and Access**

1. Assign roles in the Azure portal for secure access.
2. Configure Databricks to connect to the storage account and process data dynamically.

**Phase 3: Serving Layer in Synapse Analytics**

**Synapse Workspace Setup**

1. Create a Synapse workspace in the resource group.
   * Use the existing storage account created earlier.
   * Synapse supports integration with Azure Data Factory, Spark, and SQL Data Warehouse.
2. Synapse Components:
   * **Azure Data Factory:** Integrate pipelines.
   * **Spark Pools:** Perform transformations.
   * **SQL Data Warehouse:** Create and manage dedicated and serverless databases.

**Data Warehouse Creation**

1. Create a dedicated SQL database for structured storage.
2. Use serverless options for lakehouse architecture (data stored in Data Lake with metadata).
3. Implement OPENROWSET() for querying raw data in the lake.

**Data Processing**

1. Create credentials to connect to the storage account.
   * Use managed identity, SAS tokens, or access keys.
2. Define external data sources and file formats:
   * External Data Source: Points to the container (e.g., silver container).
   * External File Format: Defines the file format (CSV, Parquet, JSON).
3. Create views for transformed data in the gold layer.
4. Use CTAS (Create Table As Select) to create external tables from views for long-term storage.

**Connecting Synapse to Power BI**

1. Establish a connection between Synapse and Power BI using SQL endpoints.
2. Build Power BI dashboards using data in the gold layer.

**Project Overview**

1. Loaded data into the **bronze layer** using HTTP.
2. Processed and transformed data in **Databricks** from the bronze to the silver layer.
3. Pulled data from the silver layer into Synapse, created views, and pushed data into the **gold layer**.
4. Established a connection with Power BI for visualization and reporting.

**Status**

* **Raw Data Ingestion:** Completed
* **Dynamic Data Processing:** Successfully implemented
* **Databricks Configuration:** Completed for raw to silver layer transformations
* **Serving Layer:** Completed with Synapse Analytics
* **Power BI Dashboards:** Successfully connected and operational