**Technical Design Document for Stocks Strategic Dashboard**

**Project Name:** Stocks Strategic Dashboard

**Objective:** To design and implement a data pipeline that extracts, transforms, loads (ETL), and visualizes stock market data, providing insights and patterns to investors and business analysts.

**1. Overview**

This document outlines the technical design for the Stocks Strategic Dashboard project. The project aims to automate the extraction of stock market data, transform it into a structured format, store it in a database, and visualize the results through an interactive dashboard. The following sections provide details on the architecture, components, technologies, and workflows involved.

**2. System Architecture**

**A diagram of software development

Description automatically generated**

The system architecture consists of the following key components:

* **API Source (AlphaVantage API)**
* **ETL Orchestration (Apache Airflow)**
* **Containerization (Docker)**
* **Database (PostgreSQL)**
* **Visualization (Power BI)**
* **Scripting Language (Python)**

Each component plays a crucial role in the overall data pipeline, as illustrated below:

**Flow Diagram:**

* **Extract**: API Source → Apache Airflow
* **Transform**: Apache Airflow → Docker
* **Load**: Docker → PostgreSQL
* **Visualize**: PostgreSQL → Power BI

**3. Components and Technologies**

**3.1 API Source (AlphaVantage API)**

* **Purpose**: To provide real-time stock market data.
* **Integration**: The data is fetched using HTTP GET requests with an API key provided by AlphaVantage.
* **Data Format**: CSV format is used to retrieve data.

**3.2 Apache Airflow (ETL Orchestration)**

* **Purpose**: To manage and schedule ETL jobs.
* **Role**: Apache Airflow is responsible for orchestrating the workflow, from data extraction to loading it into the database.
* **Features**:
  + DAGs (Directed Acyclic Graphs) for managing task dependencies.
  + Custom Python operators to extract data from the API and load it into PostgreSQL.

**3.3 Docker (Containerization)**

* **Purpose**: To ensure a consistent and replicable environment across development, testing, and production.
* **Role**: Docker containers package the entire application stack, including Apache Airflow, PostgreSQL, and Python scripts.
* **Benefits**: Simplifies deployment and scaling, maintains environment consistency.

**3.4 PostgreSQL (Database)**

* **Purpose**: To store transformed data in a structured format optimized for querying and analysis.
* **Data Model**:
  + **Star Schema**: The data is organized into a fact table (holding business values like stock prices) and dimension tables (e.g., date dimension).
* **Integration**: Data is loaded into PostgreSQL using custom Python scripts executed via Apache Airflow.

**3.5 Power BI (Visualization)**

* **Purpose**: To create interactive dashboards and reports for end-users.
* **Data Connection**: Power BI connects to the PostgreSQL database to fetch and visualize data.
* **Visualizations**: Includes graphs, charts, and other visuals to display stock patterns and trends.

**3.6 Python (Scripting Language)**

* **Purpose**: Used for scripting the data extraction, transformation, and loading processes.
* **Role**:
  + Extracts data from the AlphaVantage API.
  + Performs data transformations required for the star schema.
  + Interacts with PostgreSQL and Power BI.

**4. Infrastructure and Deployment**

**4.1 In-House Local System**

* The entire pipeline is hosted on an in-house local system, ensuring control over data security and performance.

**4.2 Docker for Deployment**

* **Containerization**: The application stack is containerized using Docker, making it easy to deploy and manage across different environments.
* **Components in Docker**: Apache Airflow, PostgreSQL, Python scripts.

**5. Data Pipeline Workflow**

1. **Data Extraction**
   * **API Request**: The pipeline starts with a request to the AlphaVantage API using a unique API key.
   * **Response Handling**: The data is received in CSV format and saved locally.
2. **Data Transformation**
   * **Python Scripts**: Custom Python scripts process the raw data.
   * **Transformation**: The data is cleaned, structured, and organized into a star schema.
   * **Data Loading**: The transformed data is then loaded into the PostgreSQL database.
3. **Data Visualization**
   * **Power BI Connection**: Power BI connects to the PostgreSQL database.
   * **Dashboard Creation**: Visuals are created to represent stock patterns and trends, which are then presented to end-users.

**6. External Integrations**

* **AlphaVantage API**: The primary external data source for stock market data.
* **Power BI**: External tool used for data visualization, connected to PostgreSQL.

**7. Development Process**

**7.1 Agile Methodology**

* **Iterations**: The project follows an Agile methodology with iterative development, regular reviews, and updates.
* **Sprints**: Each sprint focuses on a specific phase of the ETL process, from extraction to visualization.

**7.2 Implementation Phases**

* **Phase 1: Data Extraction**
  + Tools: Python, API Integration.
  + Output: Raw CSV data.
* **Phase 2: Data Transformation and Loading**
  + Tools: Python, Apache Airflow, PostgreSQL.
  + Output: Structured data in PostgreSQL.
* **Phase 3: Data Visualization**
  + Tools: Power BI.
  + Output: Interactive dashboards.

**8. Conclusion**

The Stocks Strategic Dashboard project is designed to provide a seamless data pipeline that automates the extraction, transformation, and visualization of stock market data. By using a combination of powerful tools like Apache Airflow, Docker, PostgreSQL, and Power BI, the project aims to deliver valuable insights to investors and business analysts, helping them make informed decisions.