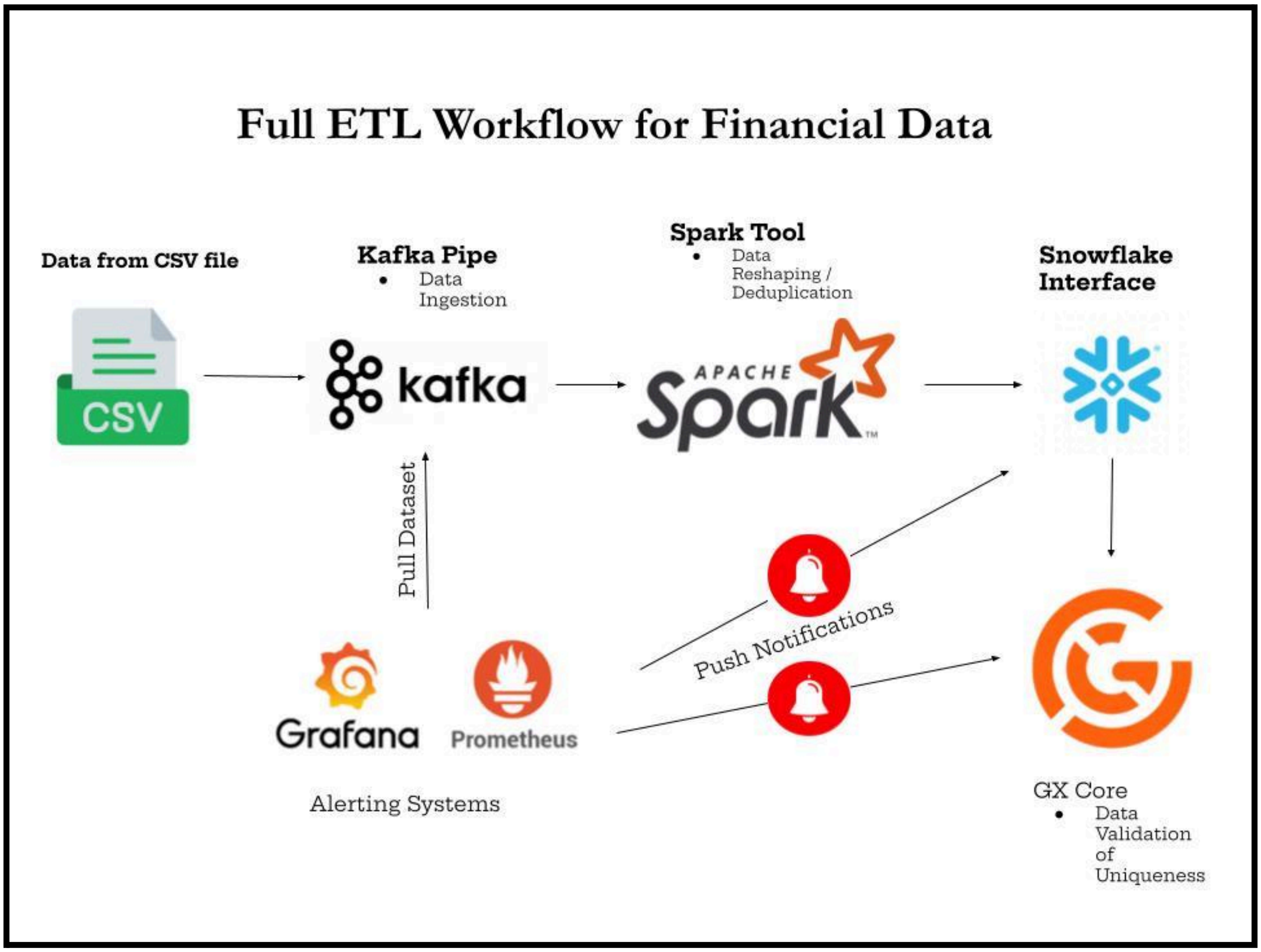
Foundation Setup:

1. Downloaded Packages for Spark and Kafka. Installed necessary packages, including PySpark
2. Created an account on Snowflake and set up a database for this project
3. Set up the consumer and producer consoles on Kafka
4. Configured Prometheus and Grafana using Homebrew
5. Set up Great Expectations Core and set up an expectation for my Snowflake table.

Pipeline Development:

MAIN PIPELINE (Diagram):



1. Data from CSV File:

a. Found 2,847 rows of Data from Data.gov on End-of-Day Pricing from NYSE Stocks. (I was not able to find sufficient data to match 1 TB per day, as it interfered with my computer storage.)

b. Column Names: Symbol, Date, Open, High, Low, Close, Volume. (Given time constraints, I was only able to find data with 1 timestamp: Date, and not bitemporal data with 2 timestamps.)

c. Downloaded data as a CSV File

2. Apache Kafka:
   a. Configured basic settings, including adjusting the default data directory of the broker Zookeeper, and adjusting the kafka-logs in server properties
   b. Started both Zookeeper and Server in my Kafka Folder:
      i. bin/zookeeper-server-start.sh config/zookeeper.properties
      ii. bin/kafka-server-start.sh config/server.properties
   c. Created Kafka Topic: earnings_topic
   d. Set up the Consumer and Producer Kafka System
   e. Ingested my dataset into Kafka from CSV, through Python3 (producers.py)
3. Apache Spark:
   a. Used Spark for Data Reshaping and Data Deduplication
   b. Built a schema with all Column Names and Data types to be sent to the Snowflake database
   c. Dropped duplicates of the dataframe using pyspark
4. Snowflake
   a. Wrote a parsed dataframe on the Snowflake account
   b. (Code was done in Python3 - consumers.py)
5. Great Expectations Core
   a. Connected the data source to Snowflake
   b. Set a preset expectation
   c. Due to time constraints, I was unable to validate this expectation with my batch of data; therefore, I left it at declaring the expectation.
6. Prometheus and Grafana
   a. Used for processing latencies and triggering alerts.
   b. Configured Prometheus and Grafana; however was not able to implement it in the pipeline due to time constraints.