CORE SCIENTIFIC

Order Book Data Processing
(Proof of Concept)

Durai Rajamanickam
02/25/2022

CORE SCIENTIFIC

Table of Contents

Scope of Work

Requirement

- Implement a proof-of-concept data pipeline to collect and analyze cryptocurrency orderbook

Data Pipelines

- Create a data pipeline/ETL to ingest and persist order book data across two (2) different exchanges for both BTC/USD and ETH/USD markets:
- Every 60 seconds poll each exchange API for the current order book and persist raw order book data
- For each poll, extract $100k of bid and ask order book data for each exchange

Live Reporting

- What is the average mid price per market?
- Which exchange would we prefer to execute a $50k buy or sell order on? At what price?

Architecture and Design Consideration

Architecture Consideration:

- ❖ Elastic Scalability
- ❖ Performance
- ❖ Cost of Operations (DevOps, DataOps)
- ❖ Ease of Development and Integration
- ❖ Ease of Maintenance
- ❖ Adaptable to future change

Design Considerations:

Data Processing approach:

- Enable Live reporting – Real time analysis
- ETL is required to normalize data coming from various exchanges. Each exchange reports orderbook at various lengths, and fields.
- Use Analytics warehouse to process large volume of data. (considering 60 sec continued stream of data)
- Ability to enable Batch processing for Historical/Timeseries analysis

Data Pipelines

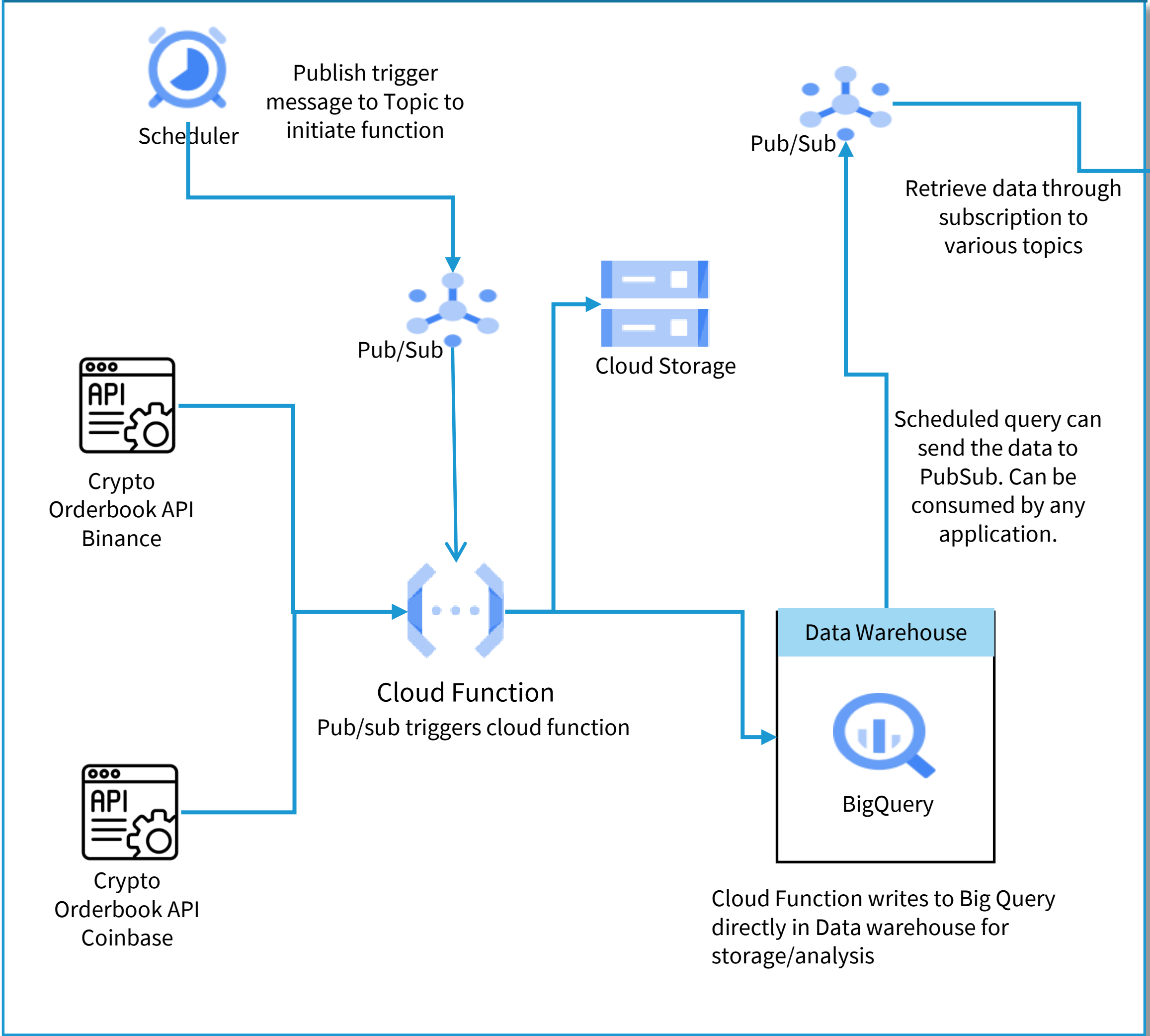
- Use data pipeline that is scalable and real time (considering 60 sec data feed). High throughput, and dynamic scaling required to meet the requirements.
- Perform data normalization (can be done at various scale) for PoC, two exchanges were considered to retrieve order book, So ETL is performed at the data collection function.
- Extract $100k of bid and ask order book data for each exchange
- Use scheduler to poll the exchange's orderbook API end point every 60 seconds

Live Reporting

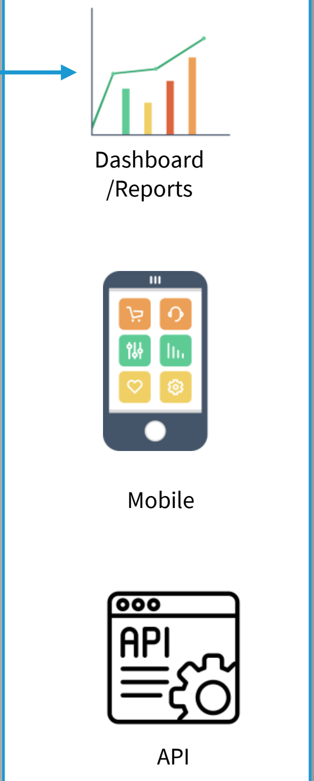
- Enable analytics warehouse to meet live reporting requirements to find the average mid-price. [Moving average if timestamp is not used]
- Provide granularity at exchange and type of trade level between asks and bids. [Persist data at granular level]

Solution design for Proof Of Concept

Solution Design – Data flow details


Scheduler
Publish trigger message to Topic to initiate function
Pub/Sub
Cloud Storage
Pub/Sub
Retrieve data through subscription to various topics
Crypto Orderbook API Binance
Scheduled query can send the data to PubSub. Can be consumed by any application.
Cloud Function
Pub/sub triggers cloud function
Data Warehouse
BigQuery
Crypto Orderbook API Coinbase
Cloud Function writes to Big Query directly in Data warehouse for storage/analysis

Systems of Engagement

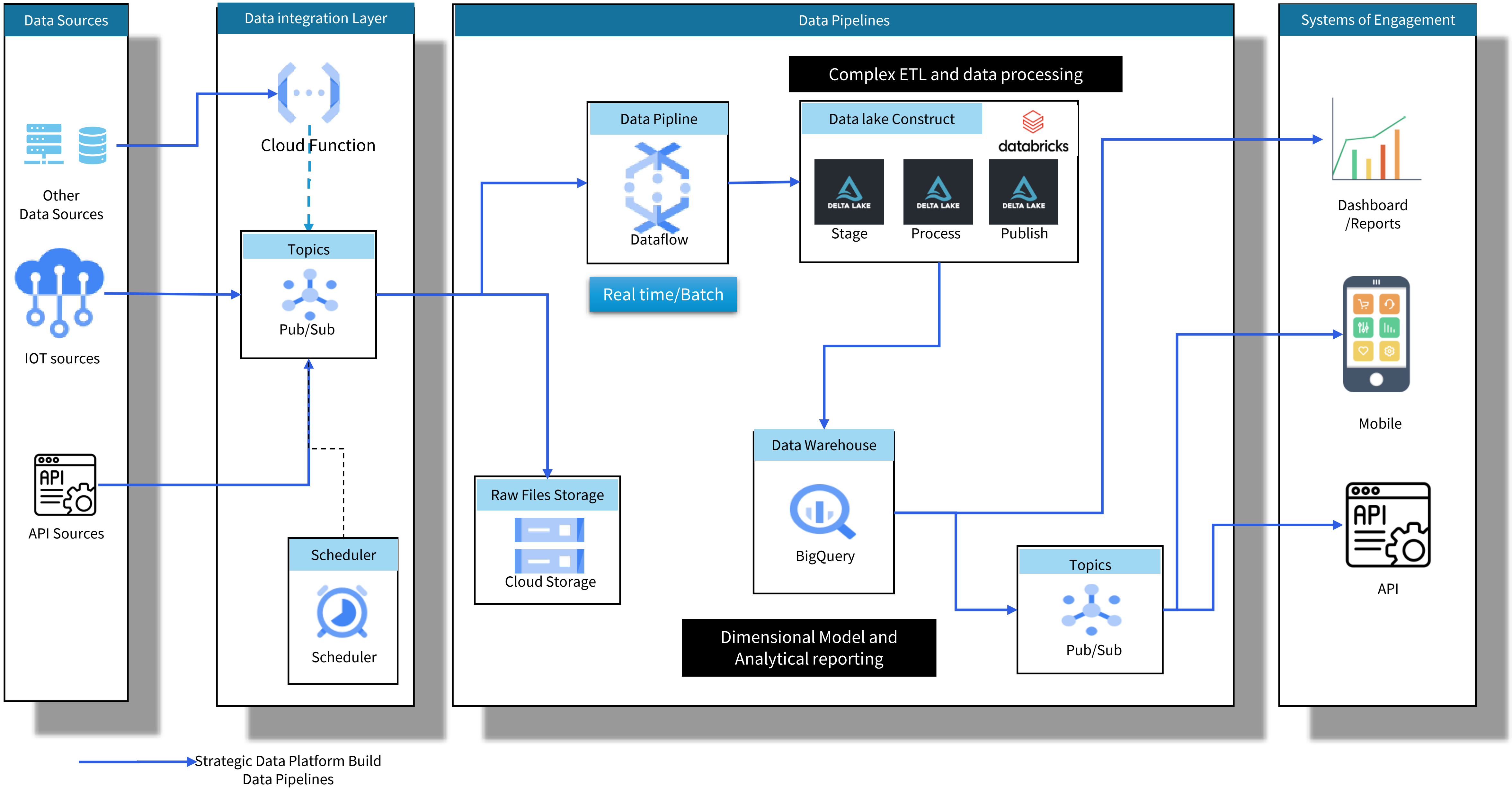


Functionality and Design decisions

- Core Scientific uses GCP and AWS, The PoC is leveraging GCP managed services.
- Use Orderbook API from Binance and Coinbase exchanges for the PoC
- Each exchange sends a slightly different structure of order book. Build modularized ETL framework to clean up the order book and standardize the data (Embedded in Cloud Function for PoC purpose)
- Use Cloud Scheduler (GCP component) to trigger the cloud function. The scheduled sends a message in a 60-sec interval to topic that triggers the function to pull data from exchanges.
- Use on-demand scalable cloud function. Build framework to extract the data in order to add more exchanges in the future.
- Once the function is triggered, the process will retrieve order book data through API.
- Send standardized "order book" with the following data structure [side(bid, asks), price, quantity, cost, timestamp, exchange] to BigQuery (Data Warehouse) and Cloud Storage. (This program extracts only trade values between $40K to $100 K for PoC purpose)
- From Big query we can analyze and extract mid-price ranges and other questions like when we can likely purchase 50k crypto tokens.

→ Proof Of Concept Data pipeline

Solution Architecture – Data platform



DEMO

CORE SCIENTIFIC

Conclusion

Following are achieved in this proof of concept.

- ✓ Collected data from various exchanges
- ✓ Sent data to storage accounts (for raw data persistent)
- ✓ Achieved data standardized from various exchanges with timestamp.
- ✓ Enabling analytics using data warehouse (Big query)
- ✓ Enabled polling for every 60 sec.
- ✓ Enabled real time access to polled data from exchanges.
- ✓ Expandable architecture to realize enterprise data platform.

CORE SCIENTIFIC

Thank You