**Project: Live Cryptocurrency Price Tracking and Analysis**

**Goal of the Project**

To track live cryptocurrency prices in near real-time and analyze short-term trends and volatility, simulating what fintech platforms or financial analysts would monitor to support trading decisions and market insights.

**1. Research CoinGecko Public API Structure & Rate Limits**

**Structure:**

* **Simple**: Fetches the current price of one or more cryptocurrencies. Example: https://api.coingecko.com/api/v3/simple/price?ids=bitcoin&vs\_currencies=usd
* **Complex**: Retrieves comprehensive market data for multiple coins, including price, market cap, and volume. Example: https://api.coingecko.com/api/v3/coins/markets?vs\_currency=usd&ids=bitcoin&order=market\_cap\_desc&per\_page=1&page=1&sparkline=false&price\_change\_percentage=24h

**Rate Limits:**

Free plan has 5-15 calls per minute. We intend on 2 calls per minute (i.e., every 30 seconds).

**2. Design Overall Architecture**

Flowchart using draw.io.

**3. Define Schema for Price Data**

(coin, price, timestamp, etc.)

**Initial Collection from Source:**

* **coin**: The unique identifier or name of the cryptocurrency (e.g., 'bitcoin', 'ethereum').
* **price**: The price of the cryptocurrency in a specific currency (e.g., USD).
* **timestamp**: The timestamp when the data was fetched (UTC).
* **market\_cap**: The market capitalization of the cryptocurrency.
* **volume\_24h**: 24-hour trading volume of the cryptocurrency.
* **high\_24h**: The highest price in the last 24 hours.
* **low\_24h**: The lowest price in the last 24 hours.
* **last\_updated**: The timestamp when the data was last updated.

**Added Spark Fields:**

* **price\_change\_1min**: (current price - price\_1min\_ago) / price\_1min\_ago \* 100
* **price\_change\_5min**: (current price - price\_5min\_ago) / price\_5min\_ago \* 100
* **SMA**: (Simple Moving Average)
* **EMA**: (Exponential Moving Average)
* **volatility**: (rolling) over a window of 5 or 10 minutes
* **top\_5\_gainers\_last\_5min**
* **top\_5\_losers\_last\_5min**

**4. Technical Constraints**

* You're okay storing data on local disk / Delta Lake.
* You batch stream every 30s instead of real-time clusters.
* Airflow unusable since Spark scripts are running on local machine and Airflow is hosted on Kubernetes Cluster.
* **Decision made**: Remove Airflow.

**5. Plan Data Enrichment Logic**

Every time data is pulled (every 30s), we store that record - raw and processed. Nothing gets overwritten. Then we compute:

* change (1min, 5min)
* SMA, EMA
* Volatility
* Top gainers/losers

We use Spark Structured Streaming with time windows and watermarking, which keeps just enough data in memory to do all these.