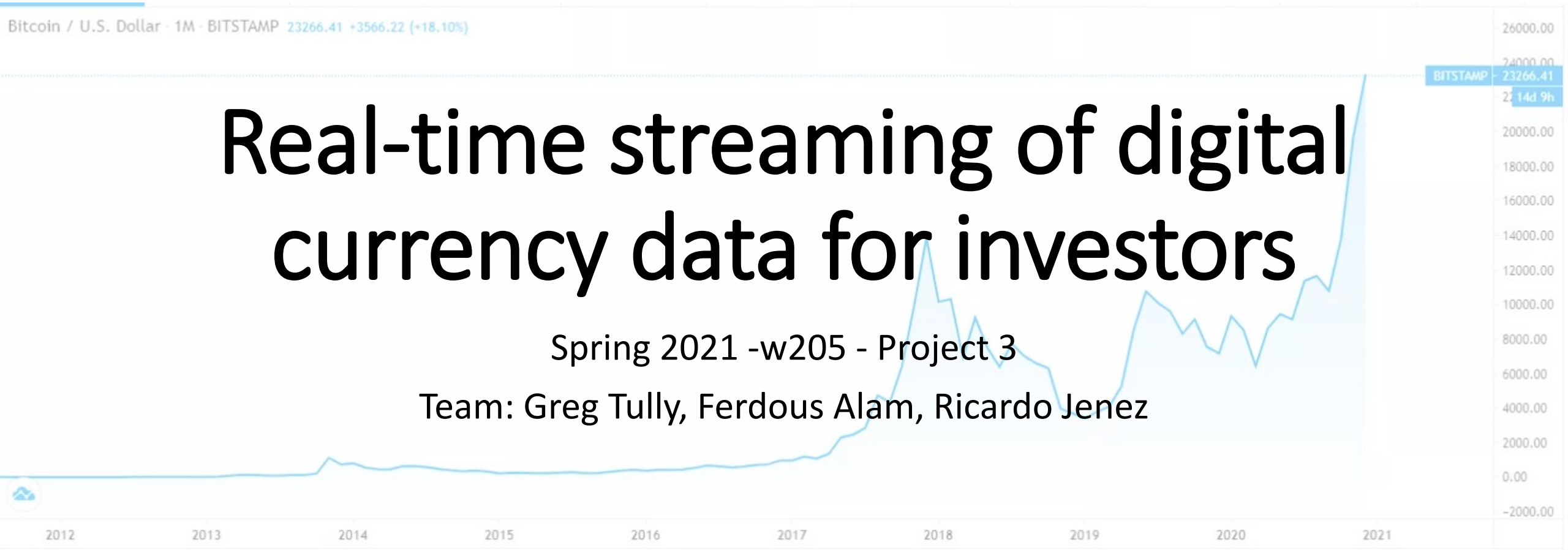


Real-time streaming of digital currency data for investors

Spring 2021 -w205 - Project 3

Team: Greg Tully, Ferdous Alam, Ricardo Jenez

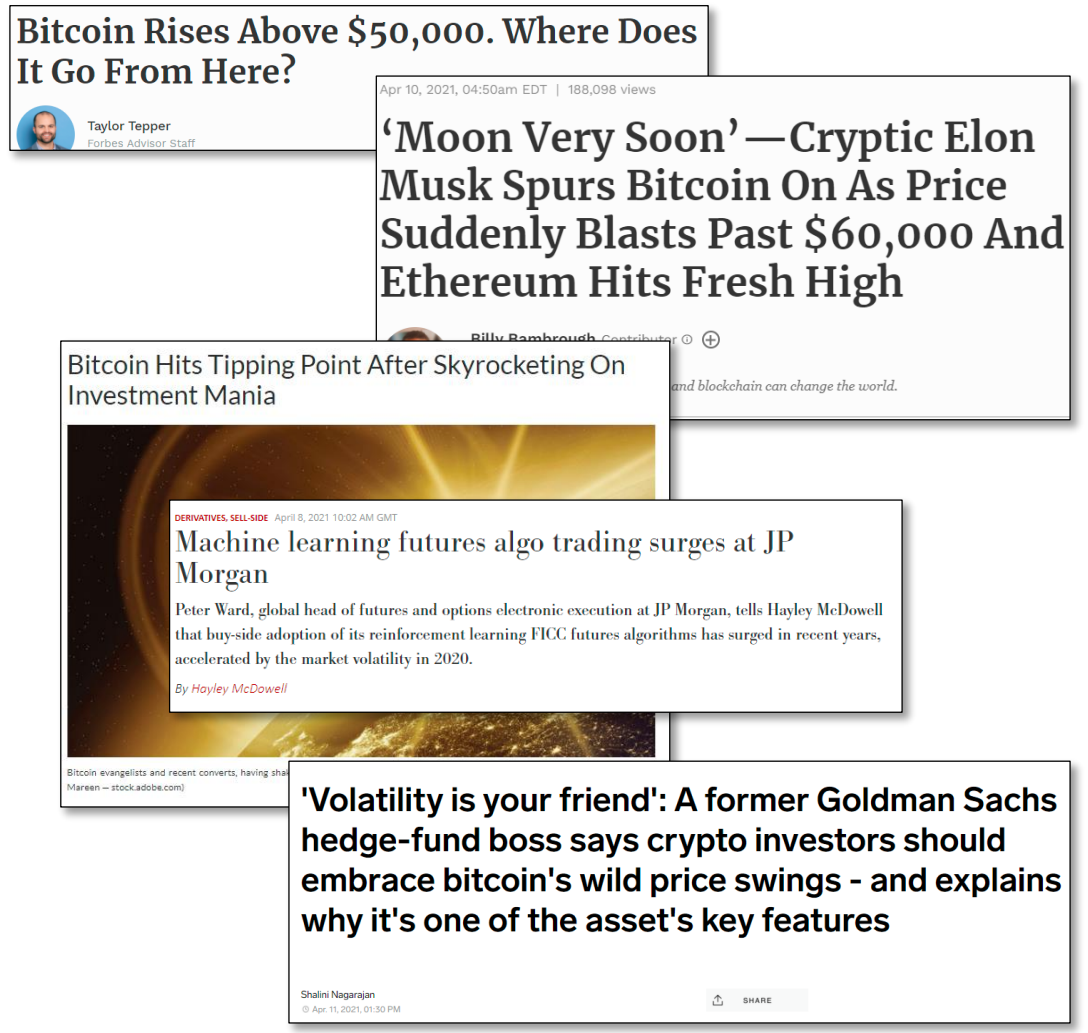
Research Question

- **Situation: Digital currencies have taken off**
  - Bitcoin launched in 2009 with almost no value and today it is worth nearly $60,000!

- **Complication**: With so much hype around digital currencies, **prices can be highly volatile.**
  - In the first weekend of 2021, bitcoin rose 20%. The next Sunday, it fell 20%

- **Challenge**: Savvy investors can exploit market volatility to generate significant returns; however, **they need real-time financial data at their fingertips to monitor performance and identify potential trading signals**

To address this challenge, our team set out to build a **streaming digital currency data pipeline** so that an investor can **have full access to real-time data for decision-making**

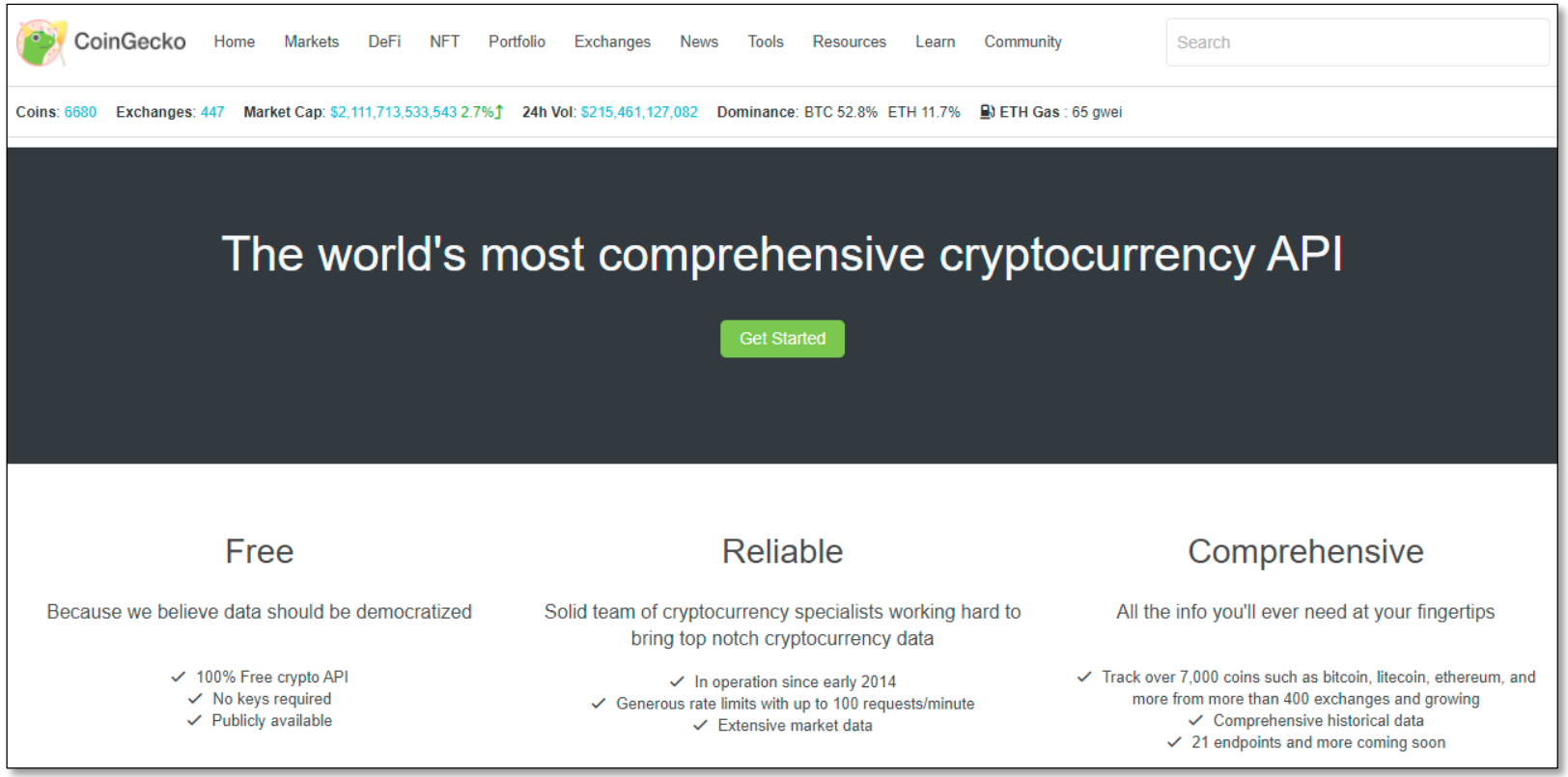


Access digital currency data in real-time with

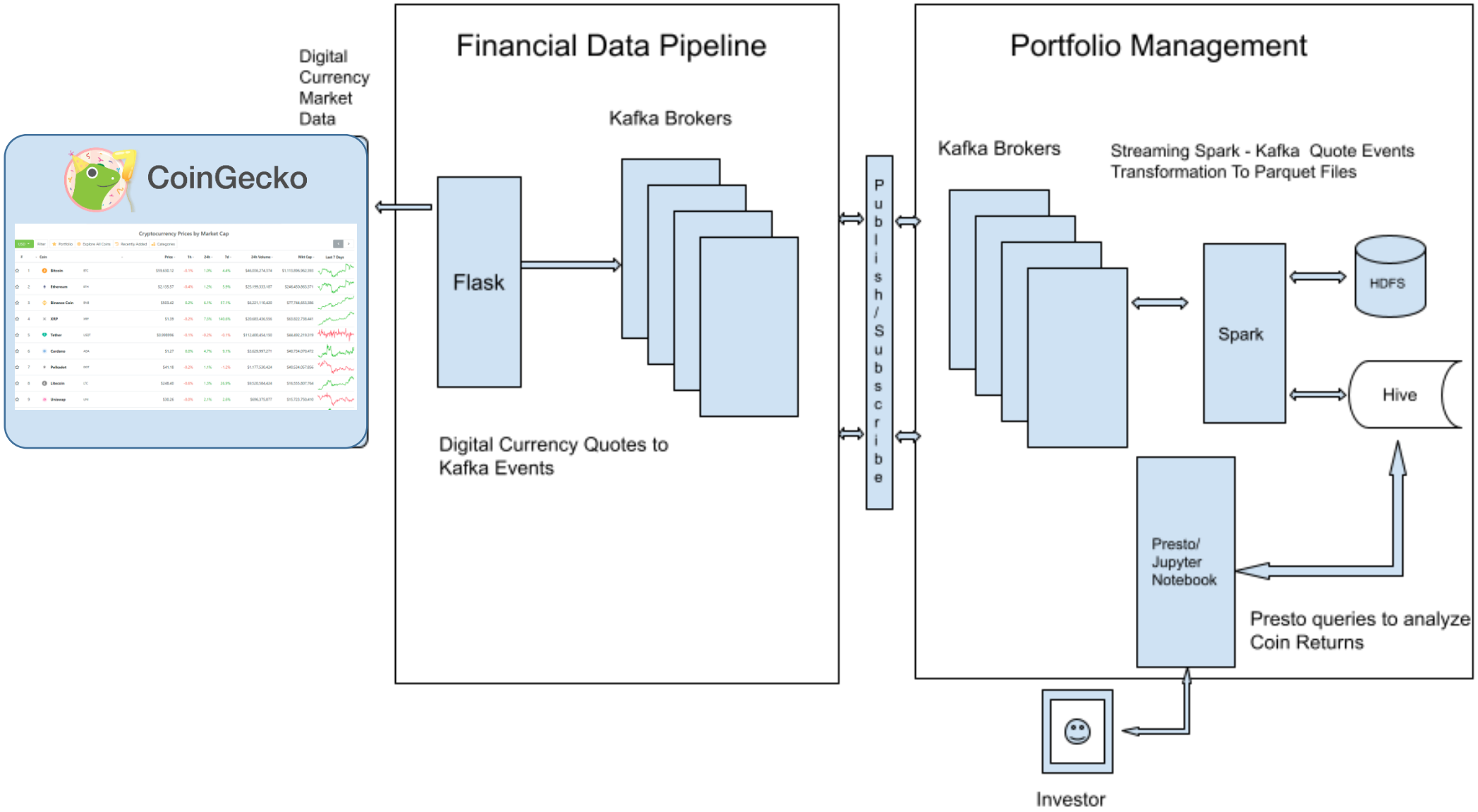
CoinGecko API

- **CoinGecko provides digital currency data** such as live quotes, trading volume, and historical data on more than 7,000 coins, such as bitcoin, litecoin, ethereum

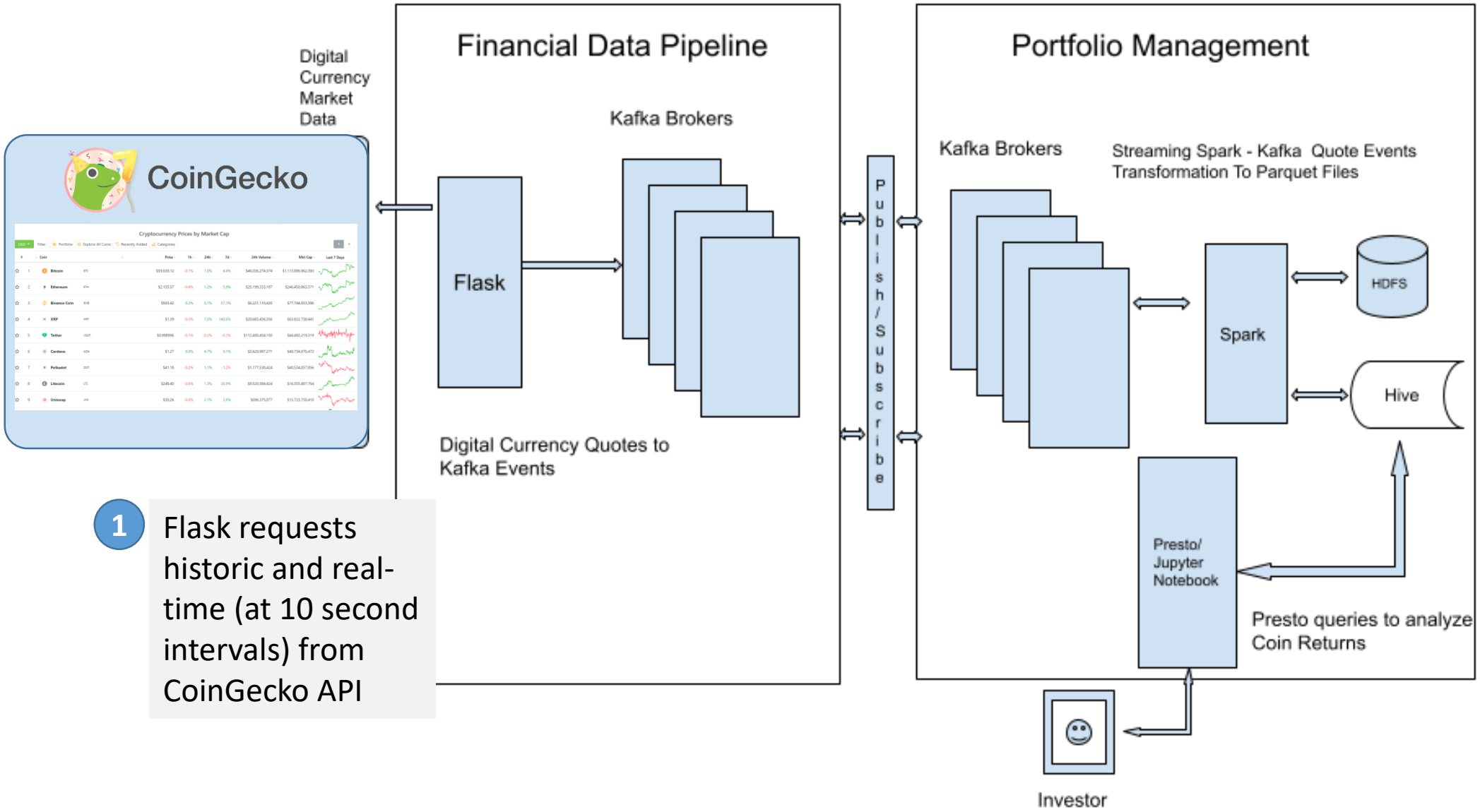
- The **CoinGecko API is free** and allows up to **10 calls per second** which is more than enough to set up a real-time streaming pipeline



Streaming pipeline set up



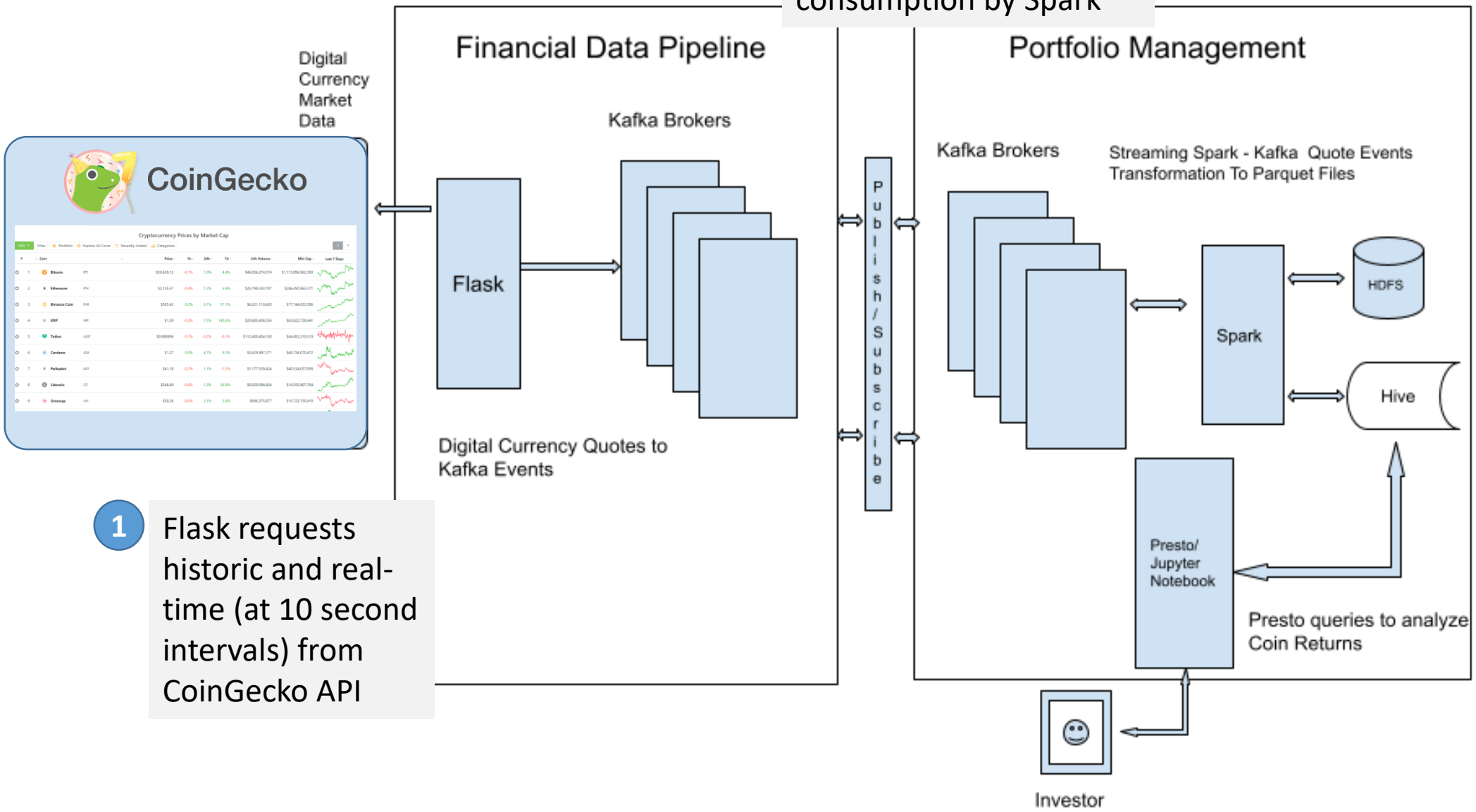
Real-time data stream: Financial Data Pipeline



1 Flask requests historic and real-time (at 10 second intervals) from CoinGecko API

Real-time data stream: Financial Data Pipeline

2 Events from Flask are logged (aka produced) to Kafka broker, ready for consumption by Spark

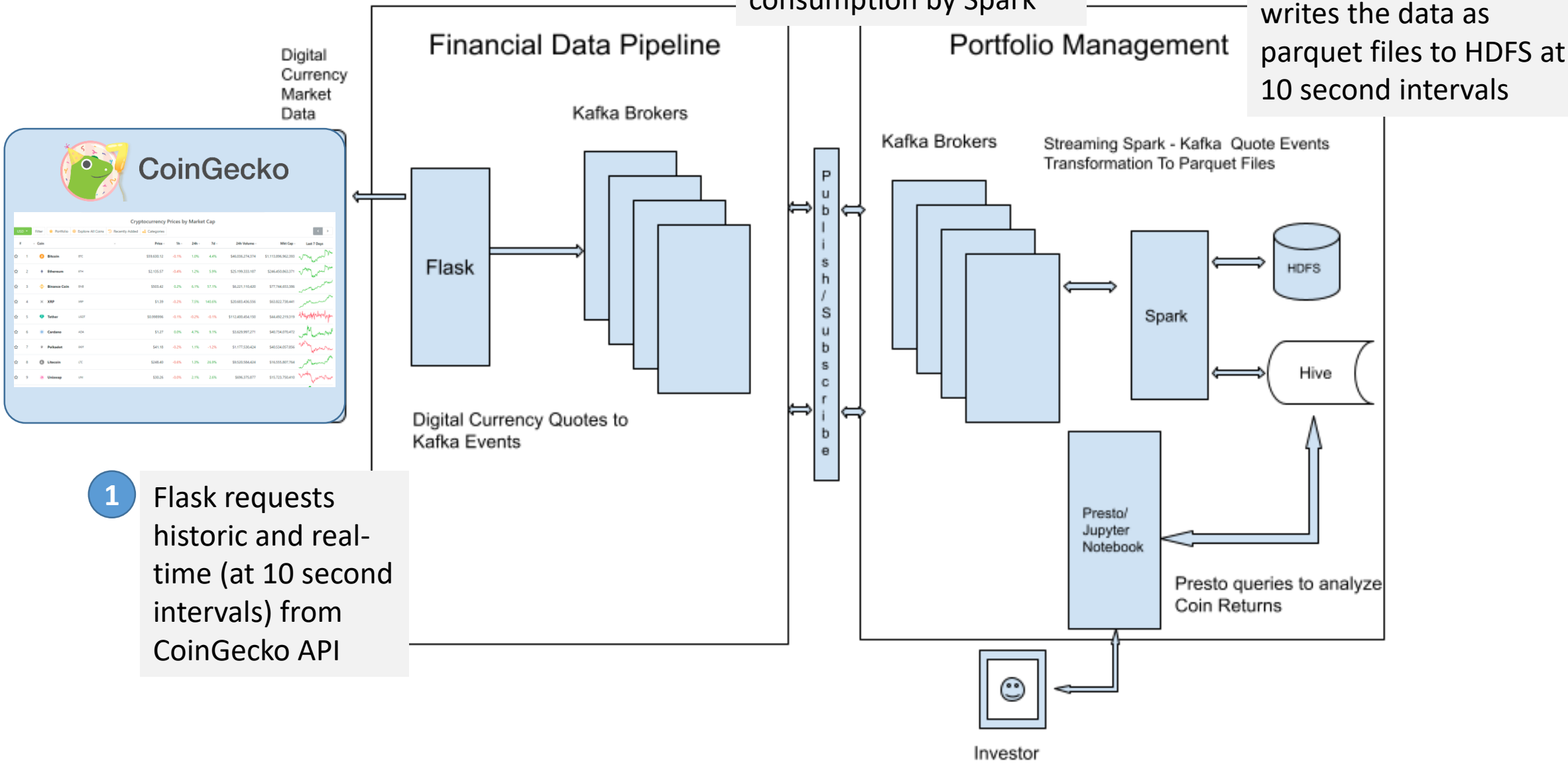


1 Flask requests historic and real-time (at 10 second intervals) from CoinGecko API

Real-time data stream: Portfolio Management

2 Events from Flask are logged (aka produced) to Kafka broker, ready for consumption by Spark

3 Spark reads JSON files from Kafka broker, defines schema and writes the data as parquet files to HDFS at 10 second intervals

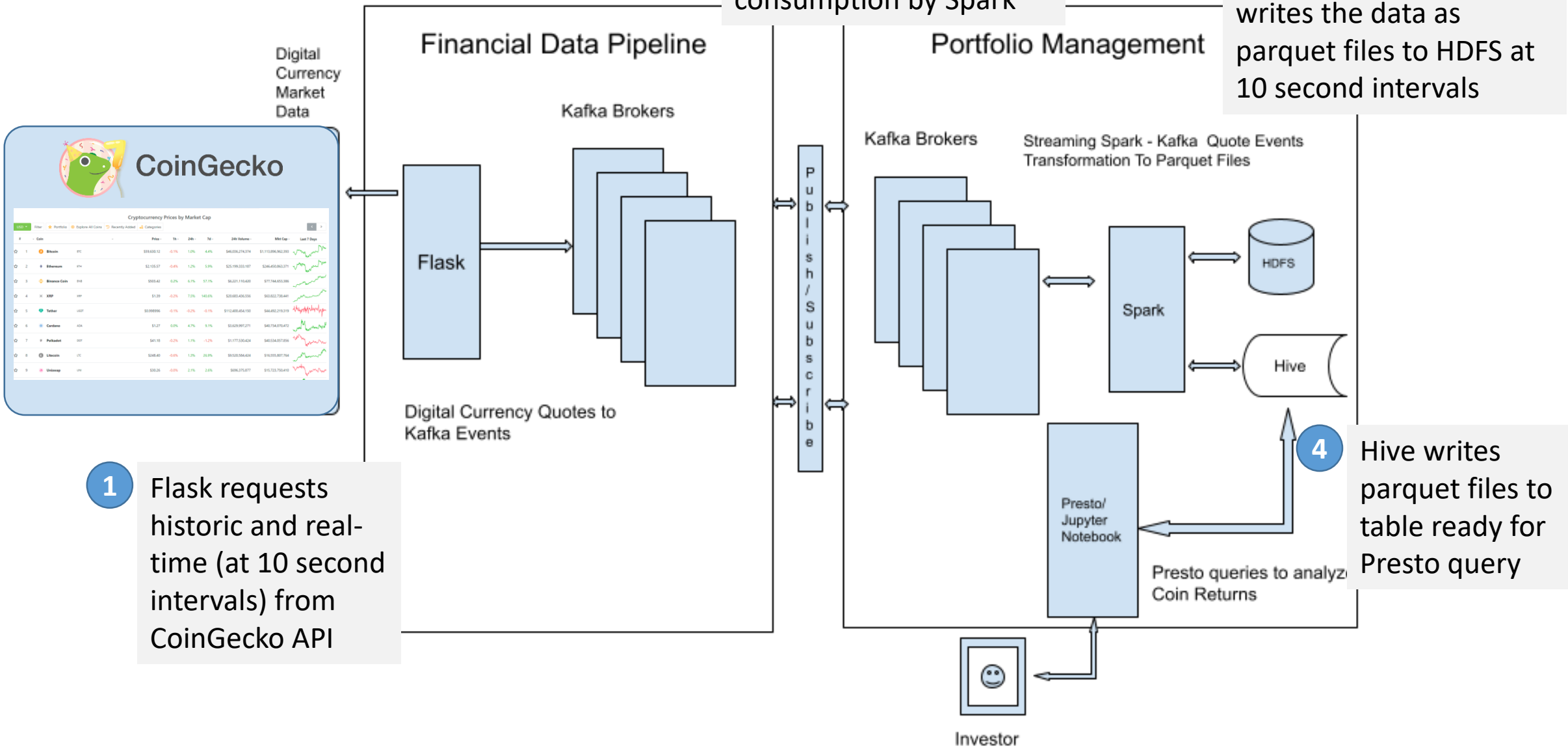


1 Flask requests historic and real-time (at 10 second intervals) from CoinGecko API

Real-time data stream: Portfolio Management

2 Events from Flask are logged (aka produced) to Kafka broker, ready for consumption by Spark

3 Spark reads JSON files from Kafka broker, defines schema and writes the data as parquet files to HDFS at 10 second intervals



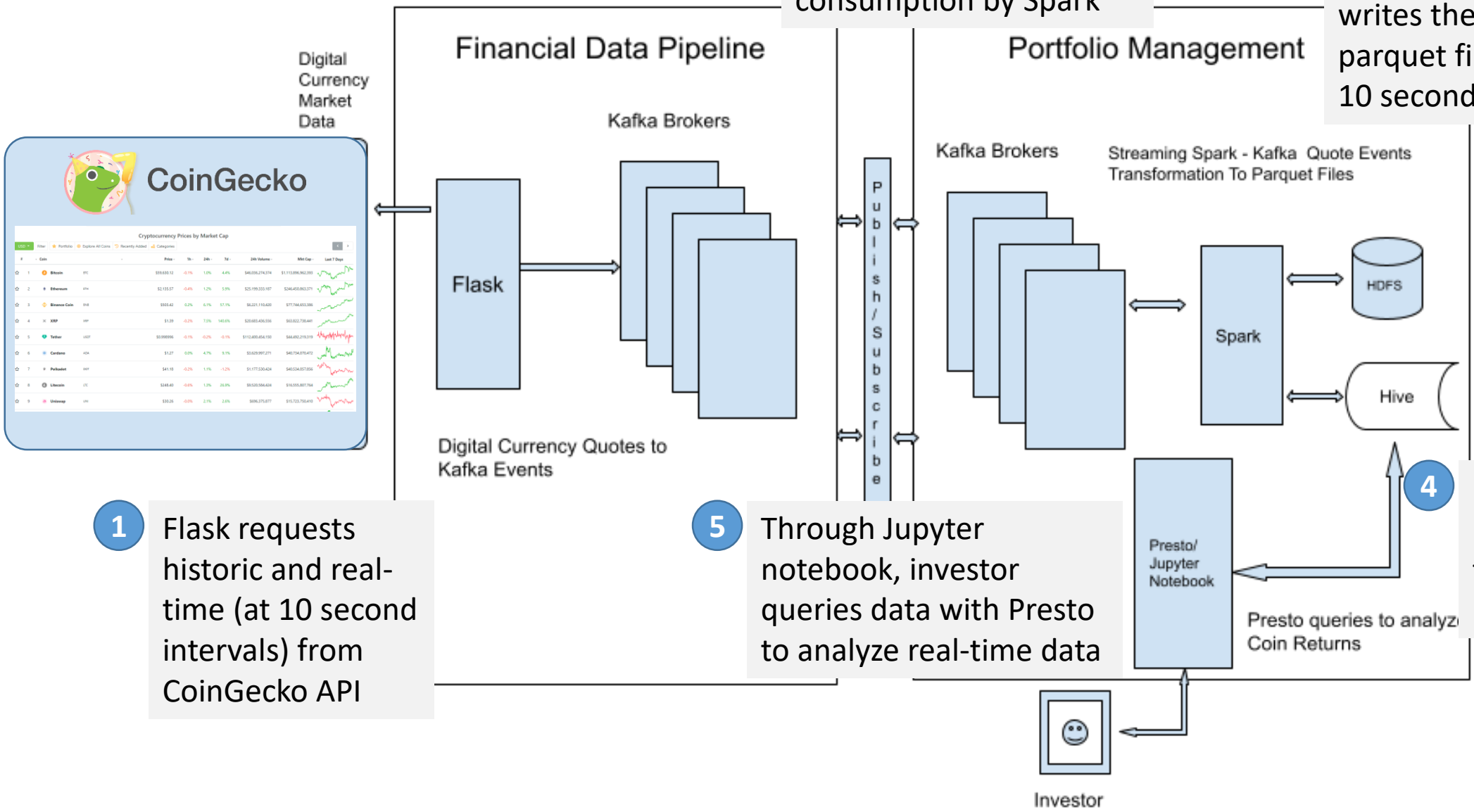
1 Flask requests historic and real-time (at 10 second intervals) from CoinGecko API

4 Hive writes parquet files to table ready for Presto query

Real-time data stream: Portfolio Management

2 Events from Flask are logged (aka produced) to Kafka broker, ready for consumption by Spark

3 Spark reads JSON files from Kafka broker, defines schema and writes the data as parquet files to HDFS at 10 second intervals



1 Flask requests historic and real-time (at 10 second intervals) from CoinGecko API

5 Through Jupyter notebook, investor queries data with Presto to analyze real-time data

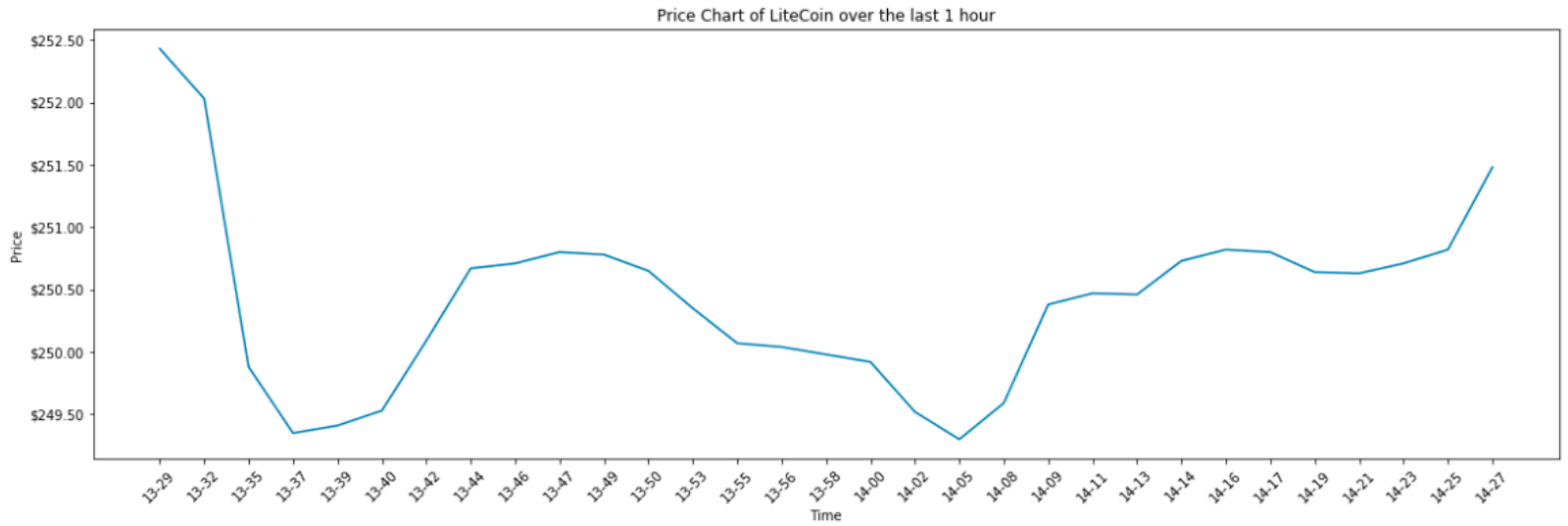
4 Hive writes parquet files to table ready for Presto query

Pipeline provides real-time data for investors & traders

Example Analytics: Current coin prices and Returns over any time period

	Coin ID	Start Date	End Date	Initial Price	Current Price	Number of Days in Period	Return
0	ethereum	2015-08-07	2021-04-11	USD 2.83	USD 2,139.06	2,074	75,441.92%
1	bitcoin	2013-04-28	2021-04-11	USD 135.30	USD 59,518.00	2,905	43,889.65%
2	litecoin	2018-07-08	2021-04-11	USD 84.50	USD 252.75	1,008	199.11%

Example Analytics: Trend of coin prices over last hour



Pipeline use cases:

- **Investors** can monitor coin prices and returns of investments
- **Algo traders** can use data to identify potential signals to trade on