**PROJECT PROPOSAL BY MICHAEL ZAVINOUSKI**

*Market Structure Analyzer*

[**Project Abstract** 3](#_Toc190958552)

[**Conceptual Design** 3](#_Toc190958553)

[**Background** 3](#_Toc190958554)

[**Proof of Concept** 3](#_Toc190958555)

[**Required Resources** 3](#_Toc190958556)

**[Class Diagram](#_Class_Diagram)** [4](#_Class_Diagram)

**[Use Case Diagram](#_Class_Diagram)** [5](#_Class_Diagram)

**[Sequence Diagram](#_Sequence_Diagram)** [6](#_Sequence_Diagram)

# **Project Abstract**

*In this project, I propose developing a market structure analyzer, a python-based platform that allows traders to read visualized data of market orders, spreads, and volumes. Behind the visualization, a computation of quotes, trades, depth, and order-flow imbalance is completed through python. Users will be able to check the order flow of a specific stock. The intended users are day-traders who would like to perform live technical analysis. The goal is to allow for day-trader and quantitative traders to perform a quick analysis to determine which trade to take.*

# **Background**

The Market Structure Analyzer ingests real-time Level-1/Level-2 market data from exchange WebSocket feeds, normalizes it, stores it in column format, and computes microstructure metrics (spread, depth, order-flow imbalance, short-horizon impact). Storage would be on S3 (Parquet) with certain metadata in AWS Glue so the data can be queried in Athena; dashboards/alerts and liquidity demand in near-real time. Public market-data WebSockets from major venues (e.g., Binance Spot/Derivatives, Coinbase Exchange) provide trades and order-book updates suitable for research and monitoring. (AI was used to turn bullet points into paragraph form here.)

<https://github.com/aws-samples/quicksight-crypto-dashboard>

* QuickSight templates/JSON for market dashboards (e.g., spread, OBI, impact).
* Data pipeline setups: Athena table DDLs, Glue extraction jobs, S3 Parquet partitioning examples.

<https://github.com/ezhulenev/orderbook-dynamics>

* Use this as a guide to formulate features, metrics, or for advanced analytics/batch ML downstream, but not as a direct ingestion or dashboard pipeline.

<https://github.com/dyn4mik3/OrderBook>

* Reference L2/L3 book state handling: matching, book updates, depth snapshots.
* Book-reconstruction code for simulation, which helps in metrics like NBBO spread, OBI, and market depth calculation.

# 

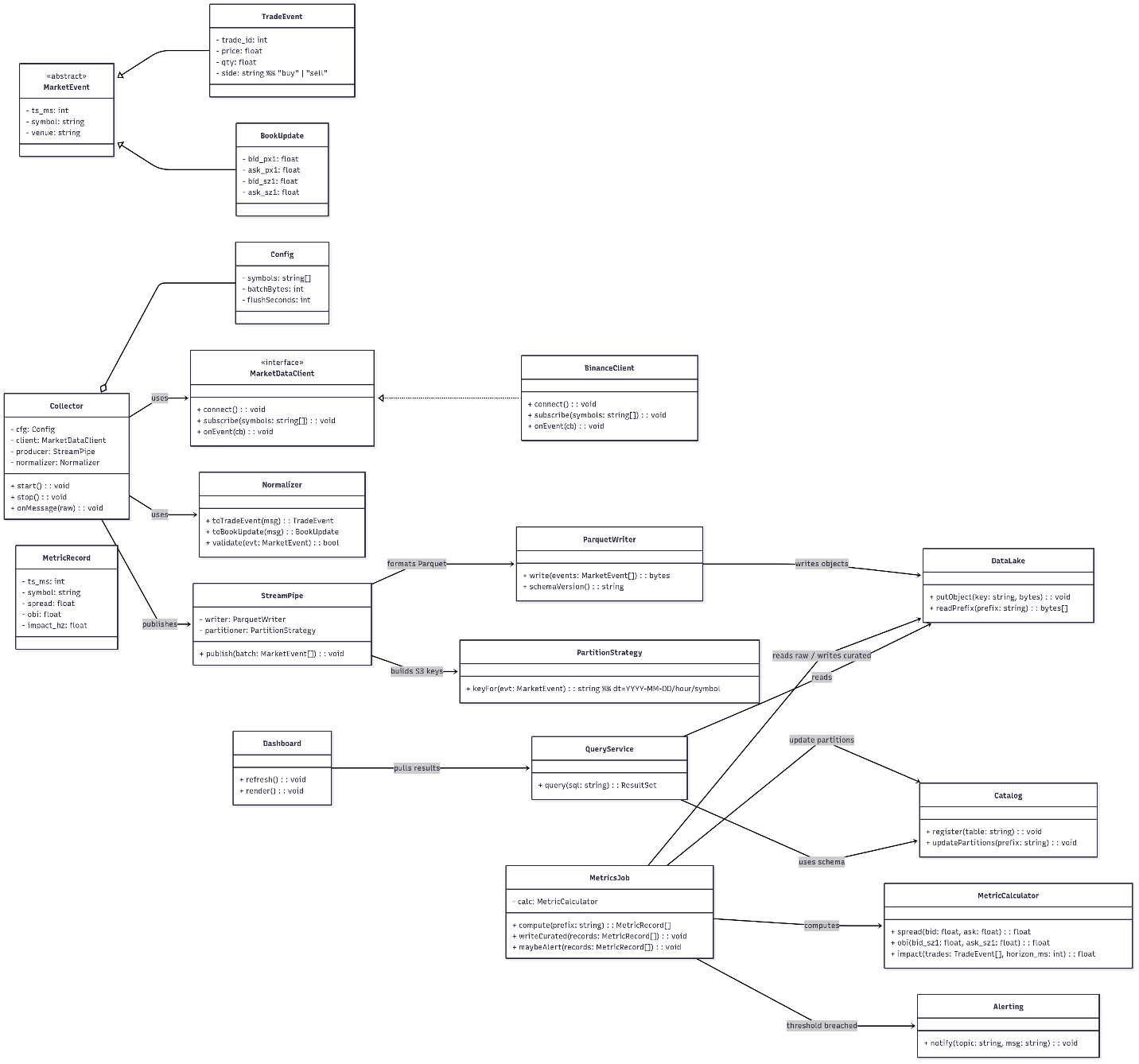
# **Proof of Concept**

<https://github.com/tul39810/FInal_project_tester>

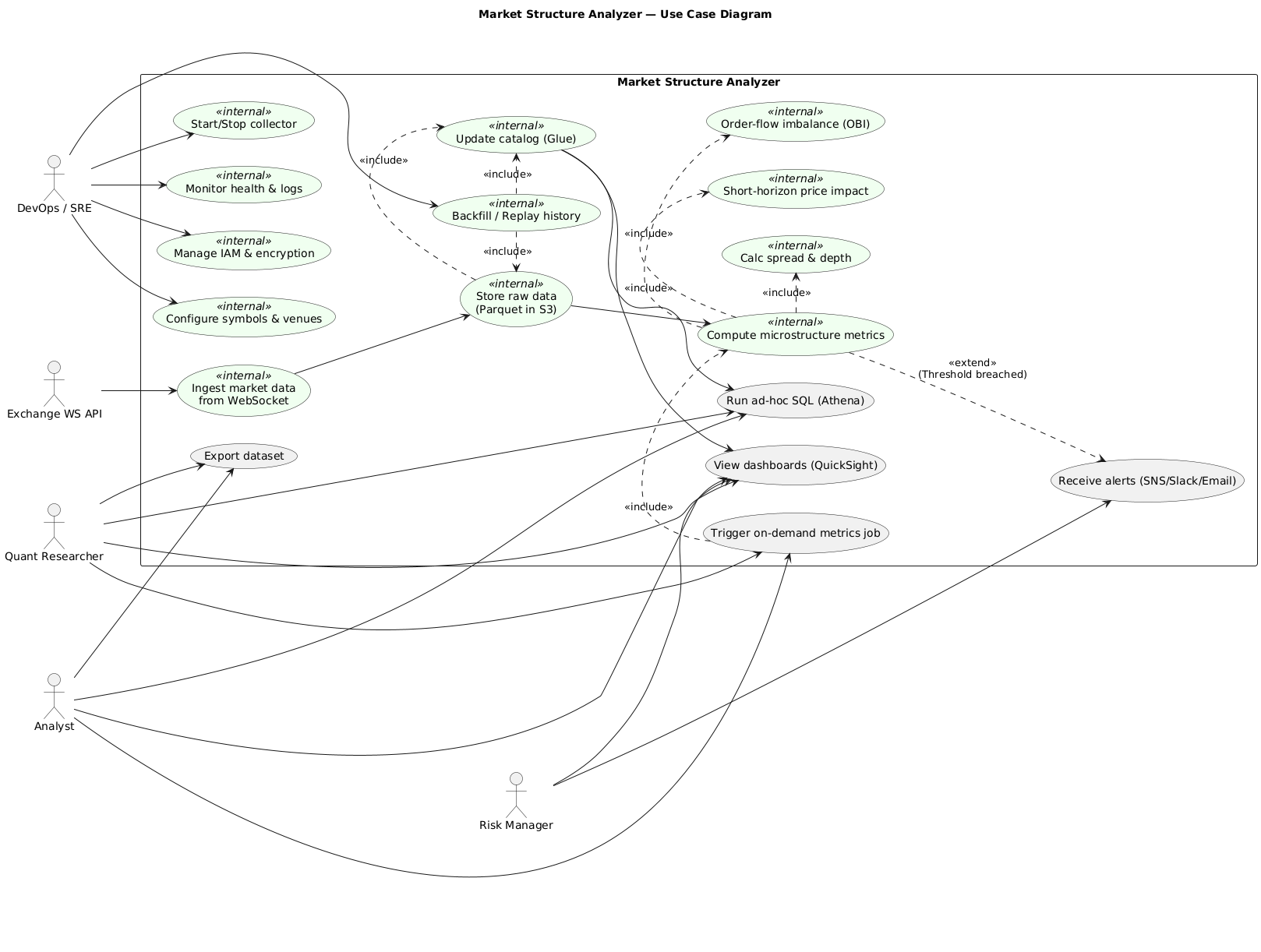
# **Required Resources**

Python, Websocket, Kinesis, AWS, Linux server (potentially), Glue/Athena, Quicksight (potentially)

# **Class Diagram**



# **Use Case Diagram**



# **Sequence Diagram**

