High Level Design Document

Introduction

This High Level Design (HLD) document outlines the architecture and core components of **FinSight - Real-Time Financial Analytics Engine**. FinSight is a backend system designed for high-performance processing, analysis, and visualization of large-scale financial transactions, leveraging Java Streams, multithreading, Spring Boot, PostgreSQL, and a Dockerized microservices architecture.

1. System Architecture Overview

Architecture Description:
FinSight is structured as a set of Dockerized microservices, each responsible for a core function: data ingestion, analytics processing, API serving, and data storage. Services communicate via REST APIs and are secured using JWT authentication.

Module/Component	Role/Responsibility
Ingestion Service	Receives and validates incoming transaction data
Analytics Engine	Processes data streams, performs real-time analytics
API Gateway	Exposes REST APIs, handles authentication & routing
Data Store (PostgreSQL)	Persists transactions, analytics results, and metadata
Visualization Service	Provides endpoints for analytics visualization
Auth Service	Manages JWT-based authentication and authorization

2. Component Interactions

Interaction Sequence
1. Client sends transaction data to Ingestion Service via API Gateway
2. Ingestion Service validates and forwards data to Analytics Engine
3. Analytics Engine processes data using Java Streams & multithreading, stores results in Data Store
4. Visualization Service retrieves analytics data from Data Store via secure API
5. All API requests are authenticated and authorized via Auth Service (JWT)

3. Data Flow Overview

Source	Destination	Data Type	Flow Description
Client	Ingestion Service	Transaction Data (JSON)	Incoming financial transactions

Ingestion Service	Analytics Engine	Validated Transactions	Streamed for real-time processing
Analytics Engine	Data Store	Analytics Results	Persisted analytics and raw data
Visualization	Data Store	Query/Response	Fetches analytics for visualization endpoints
API Gateway	All Services	Auth Tokens, Requests	Secures and routes all API traffic

4. Technology Stack

Layer/Function	Technology/Framework
Language	Java 17+
Framework	Spring Boot
Data Processing	Java Streams, Multithreading
Database	PostgreSQL
API Security	JWT (JSON Web Tokens)
Containerization	Docker
Build/Dependency Mgmt	Maven
API Design	RESTful APIs

5. Scalability, Reliability & Security

- **Scalability:**
  - Microservices architecture enables independent scaling of ingestion, analytics, and visualization services.
  - Stateless services and containerization (Docker) support horizontal scaling and orchestration.
- **Reliability:**
  - Advanced error handling and retry mechanisms in all services.
  - Persistent storage in PostgreSQL ensures data durability.
- **Security:**
  - All APIs secured with JWT-based authentication and authorization.
  - Input validation and strict access controls at API Gateway and service levels.

End of Document