

# AI-Powered Predictive Trading Analytics DB System

David Alexander Colorado Rodríguez

Andres Felipe Martin Rodriguez



Databases II, Systems Engineering Curriculum Project, Universidad Distrital Francisco José de Caldas, Colombia, Bogotá D.C.

## 1. Introduction

Modern financial trading demands speed, accuracy, and adaptability. This project presents an innovative hybrid database architecture that leverages PostgreSQL, MongoDB, and Snowflake—integrated through Apache Kafka—to support real-time data processing, high-frequency transactions, and complex analytics. Designed for scalability, regulatory compliance, and cost efficiency, the system achieves a 98.7% trade execution success rate, reduces storage costs by 33%, and cuts alert latency by 37%. This architecture offers a future-ready solution for AI-driven predictive trading platforms.

## 2. Goal

To design and implement a comprehensive trading analytics platform that leverages artificial intelligence to identify optimal trading opportunities across global markets while providing customizable strategy evaluation and automated trade execution. The system aims to deliver real-time insights for traders of varying experience levels while maintaining high performance, security, and regulatory compliance.

## 3. Analysis

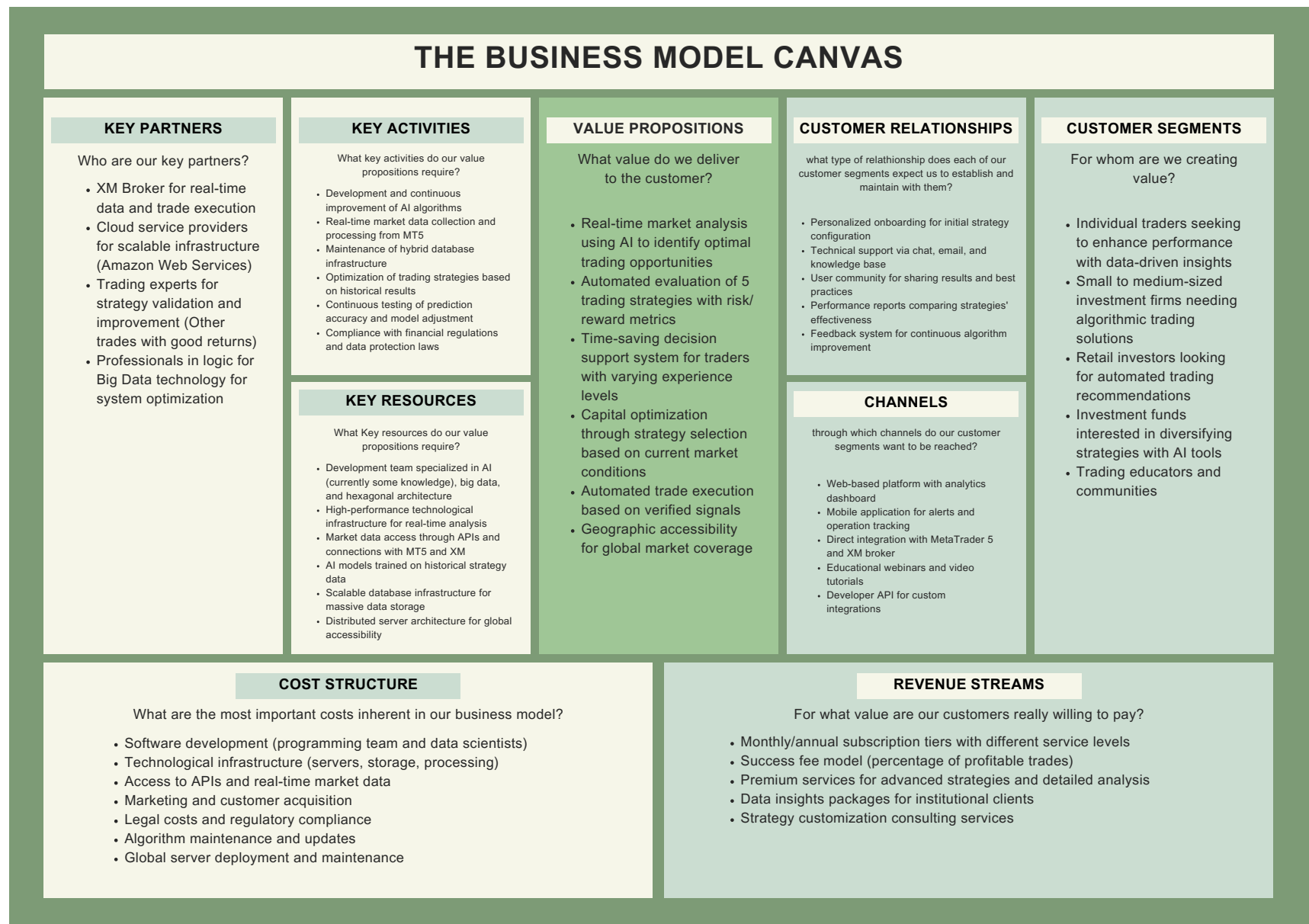


Figure 1: Business Model Canvas

Designing a system capable of handling real-time trading data, high-frequency transactions, and complex analytics presented significant challenges. One of the core difficulties was achieving low-latency processing while maintaining transactional integrity, especially during periods of extreme market volatility where data volume spikes dramatically.

Another major challenge was ensuring consistent data synchronization across systems without sacrificing performance. Delays or inconsistencies between real-time data ingestion and analytical layers risked inaccurate insights or flawed trading decisions. Balancing consistency with speed required careful planning and real-time monitoring.

Lastly, regulatory compliance added layers of complexity, especially with global data residency, privacy, and audit requirements. The system had to implement fine-grained controls and ensure secure, traceable data handling—without compromising on responsiveness or scalability. These challenges shaped a robust, resilient architecture built to meet the demands of modern financial systems.

## 4. Design

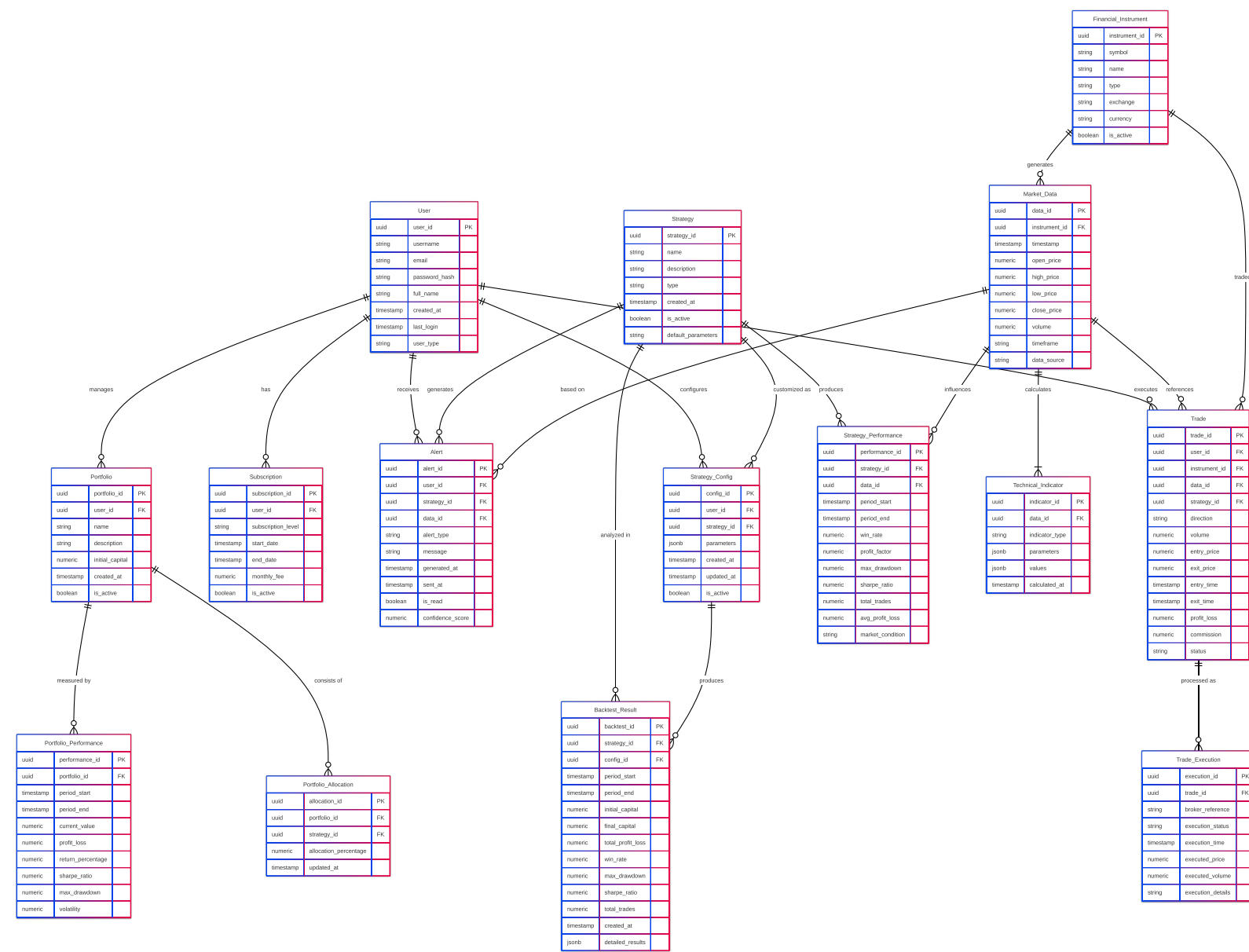


Figure 2: Entity-Relation Model

During the design stage, one of the biggest challenges was balancing the conflicting requirements of speed, reliability, and compliance. Financial systems demand strict ACID compliance for transactions, while also needing sub-second performance for real-time data. Designing a structure that could handle both without compromising one or the other required early domain separation and deep workload analysis.

A key process was the selection and alignment of database technologies to specific system domains. PostgreSQL was chosen for transactional operations, MongoDB for high-velocity time-series data, and Snowflake for complex analytics. Defining clear boundaries between these systems through domain-driven design helped reduce complexity and allowed each component to be optimized independently. Integrating these systems required thoughtful planning of data flow, consistency models, and synchronization strategies. Apache Kafka was introduced early in the design as the backbone for event-driven communication. This ensured minimal latency across components while allowing for future scalability. Overall, the design phase was guided by performance modeling, regulatory constraints, and a clear focus on operational resilience.

## 5. Outcomes

### Performance Gains

The implemented hybrid architecture delivered measurable improvements across all key system metrics. Transaction success rates exceeded 98.7%, alert latency dropped by 37%, and storage costs were reduced by 33%. These performance gains directly supported faster trading decisions and more reliable user experiences, especially during volatile market conditions.

### Scalability and Responsiveness

By distributing workloads across PostgreSQL, MongoDB, and Snowflake, the system achieved high scalability and responsiveness. Real-time data ingestion reached over 1 million events per second, while complex analytics maintained sub-second response times. Apache Kafka ensured synchronized data flow, allowing users to access consistent insights across transactional and analytical domains.

### Compliance and Resilience

Operationally, the platform improved system availability to 99.997% and simplified compliance with global financial regulations. Features like geographic data residency, automated audit trails, and row-level security not only met strict regulatory standards but also enhanced trust and transparency for users. Overall, the implementation proved that a hybrid, domain-driven architecture can outperform traditional monolithic systems in speed, resilience, and compliance.