

AIAlgoTradeHits.com

Fintech AI Data Architecture & Implementation Specification

Version:	2.0
Platform:	Google Cloud Platform (GCP)
AI Engine:	Vertex AI + Gemini 3
Date:	November 2025
Classification:	Technical Specification

Executive Summary

AIAlgoTradeHits.com is a comprehensive cross-asset, AI-driven analytics, alerting, and intelligence platform delivering institutional-grade insights across equities, ETFs, cryptocurrency, forex, commodities, and interest-rate markets. Built entirely on Google Cloud Platform, the system leverages Vertex AI, Gemini 3 LLM, BigQuery, and Cloud Run to create a next-generation fintech trading intelligence engine.

Platform Vision

Component	Technology	Purpose
Data Warehouse	BigQuery	Multi-layer data architecture (Bronze/Silver/Gold)
AI/ML Engine	Vertex AI	Predictive models, pattern recognition, regime classification
LLM Intelligence	Gemini 3 Pro	Natural language analysis, RAG-powered commentary
Document Intelligence	Document AI + RAG	Research document processing, knowledge retrieval
Real-Time Serving	Cloud Run + Functions	API endpoints, scheduled data collection
Frontend	React + TradingView	Professional trading interface

Part 1: Data Architecture

1.1 Multi-Layer Data Architecture

The platform implements a rigorous four-layer data architecture inspired by institutional quant trading desks:

BRONZE LAYER - Raw, Immutable Market Data: Exact API responses preserved, full lineage and audit trail, compliance-grade immutability

SILVER LAYER - Cleaned & Standardized Data: Normalized timestamps (UTC), missing data handling, anomaly removal, session calendar alignment

GOLD LAYER - Engineered Features & Intelligence: 70+ Technical Indicators, pattern recognition signals, cross-asset correlations, volatility regime classification, ML-ready feature vectors

WIDE TABLES - Materialized Fast-Access Layer: Flattened serving tables, real-time UI/API access, LLM RAG retrieval optimized, ML inference ready

1.2 BigQuery Table Structure

1.2.1 Asset Class Tables (6 Categories)

Table Name	Asset Class	Timeframes	Data Source
stocks_historical_daily	US Equities	Daily, Weekly	TwelveData
stocks_hourly	US Equities	Hourly, 5-min	TwelveData
cryptos_historical_daily	Cryptocurrency	Daily, Weekly	Kraken/TwelveData
crypto_hourly_data	Cryptocurrency	Hourly, 5-min	Kraken
etfs_historical_daily	ETFs	Daily, Weekly	TwelveData
forex_historical_daily	FX Pairs	Daily, Weekly	TwelveData
indices_historical_daily	Market Indices	Daily, Weekly	TwelveData
commodities_historical_daily	Commodities	Daily, Weekly	TwelveData

Core OHLCV Fields (Required)

Field Name	Type	Description	Status
symbol	STRING	Asset ticker symbol	Available
datetime	TIMESTAMP	Candle timestamp (UTC)	Available
open	FLOAT64	Opening price	Available

high	FLOAT64	High price	Available
low	FLOAT64	Low price	Available
close	FLOAT64	Closing price	Available
volume	FLOAT64	Trading volume	Available
candle_body_pct	FLOAT64	$(\text{close} - \text{open}) / \text{open}$	Add
candle_range_pct	FLOAT64	$(\text{high} - \text{low}) / \text{open}$	Add

RSI Indicators

Field Name	Type	Description	Status
rsi_7	FLOAT64	RSI 7-period	Available
rsi_14	FLOAT64	RSI 14-period	Available
rsi_21	FLOAT64	RSI 21-period	Available
rsi_slope	FLOAT64	RSI change rate	Add
rsi_zscore	FLOAT64	RSI z-score (100-week)	Add
rsi_overbought_flag	INT64	RSI > 70 flag	Add
rsi_oversold_flag	INT64	RSI < 30 flag	Add

Part 2: AI & Machine Learning Architecture

2.1 Vertex AI Integration

2.1.1 Model Types & Use Cases

Model Type	Vertex AI Service	Use Case
Gradient Boosting	AutoML Tables	Regime classification, direction prediction
Time Series	Vertex AI Forecasting	Price prediction, volatility forecasting
LSTM/Transformer	Custom Training	Sequential pattern learning
Vision Models	AutoML Vision	Chart pattern recognition
LLM	Gemini 3 Pro	Commentary generation, analysis

2.2 Gemini 3 LLM Integration

2.2.1 LLM Use Cases

Use Case	Input	Output	Frequency
Market Commentary	Indicators, patterns, regime	Research-style analysis	Real-time
Alert Explanations	Alert trigger data, context	Natural language reasoning	Per alert
Pattern Analysis	Chart patterns, structure	Professional interpretation	On detection
Risk Assessment	Cross-asset correlations	Risk narrative	Hourly
Q&A Interface	User questions + RAG context	Informed responses	On-demand

2.2.2 RAG (Retrieval-Augmented Generation) Architecture

The RAG pipeline consists of: 1) USER QUERY / TRIGGER, 2) CONTEXT RETRIEVAL from BigQuery Market Data, Vector DB Research Embeddings, and Document Store, 3) PROMPT CONSTRUCTION with System Instructions + Retrieved Context + User Query, 4) GEMINI 3 PRO for grounded response generation with citation and hallucination prevention, 5) STRUCTURED OUTPUT with market analysis, key insights, risk warnings, and actionable recommendations.

Part 3: Implementation Phases

3.1 Phase 1: Trader-Focused AI Alert Engine

Focus: Real-time alerts for equities and cryptocurrency

MVP Symbol Universe

Category	Examples	Count
US Equities	AAPL, NVDA, JPM, LLY, MSFT, GOOGL, AMZN, META, TSLA	50
ETFs	SPY, QQQ, IWM, DIA, VTI, VOO, GLD	30
Cryptocurrency	BTC/USD, ETH/USD, SOL/USD, XRP/USD	20
Forex	EUR/USD, GBP/USD, USD/JPY	20
Indices	SPX, NDX, DJI, VIX	14
Commodities	XAU/USD, XAG/USD, CL, NG	16

Phase 1 Deliverables:

- 1. Real-Time Alert System:** Trend flip alerts, momentum reversal signals, volatility expansion warnings, breakout/breakdown detection, pattern confirmation alerts
- 2. AI-Enhanced Dashboards:** Regime classification display, volatility heatmaps, correlation monitors, sentiment indicators
- 3. LLM Commentary (Basic):** Alert explanation generation, daily market summaries, pattern interpretation

Part 4: GCP Infrastructure

4.1 Service Architecture

GCP PROJECT: cryptobot-462709

The infrastructure consists of: Cloud Run (Frontend + API), BigQuery (Data Warehouse), Vertex AI (ML + Gemini 3), Cloud Functions (Data Fetch), Cloud Storage (Documents), and Cloud Scheduler (Triggers).

4.2 Cost Optimization Strategy

Service	Optimization	Monthly Estimate
BigQuery	Partitioned tables, clustering	\$50-100
Cloud Functions	Efficient scheduling, cold start optimization	\$130
Cloud Run	Auto-scaling, min instances	\$50-100
Vertex AI	Batch predictions, model caching	\$200-500
Gemini 3	Caching, prompt optimization	\$100-300
Total Estimate		\$530-1,130/month

Part 5: Missing Fields Implementation Plan

Priority 1: Critical ML Features (Add Immediately)

- Log returns: $\text{weekly_log_return} = \ln(\text{close_t} / \text{close_t-1})$
- Multi-lag returns: `return_2w`, `return_4w`
- RSI enhancements: `rsi_slope`, `rsi_zscore`, `rsi_overbought_flag`, `rsi_oversold_flag`
- MACD enhancements: `macd_hist`, `macd_cross_flag`
- EMA suite: `ema_5`, `ema_10`, `ema_20`, `ema_50`, `ema_100`, `ema_200`
- MA distance features: `close_vs_sma20_pct`, `close_vs_sma50_pct`, `close_vs_sma200_pct`, `close_vs_ema20_pct`
- EMA slopes: `ema_slope_20`, `ema_slope_50`

Priority 2: Enhanced Features (Add Next Sprint)

- ATR enhancements: `atr_pct`, `atr_zscore`, `atr_slope`
- Volume enhancements: `volume_zscore`, `volume_ratio`

- ADX trend: adx, plus_di, minus_di
- Candle geometry: candle_body_pct, candle_range_pct

Part 6: Implementation Roadmap

Week 1-2: Data Schema Enhancement

- Add Priority 1 missing fields to BigQuery tables
- Update data fetching scripts with new calculations
- Backfill historical data with new indicators

Week 3-4: ML Pipeline Setup

- Configure Vertex AI Feature Store
- Create initial training pipelines
- Deploy baseline models

Week 5-6: LLM Integration

- Implement RAG pipeline with BigQuery
- Configure Gemini 3 Pro prompts
- Build commentary generation endpoints

Week 7-8: Testing & Optimization

- Backtest ML models
- Evaluate LLM output quality
- Performance optimization

Appendix A: Technical Indicator Formulas

A.1 Log Return

`weekly_log_return = ln(close_t) - ln(close_t-1)`

A.2 RSI Slope

`rsi_slope = rsi_t - rsi_t-1`

A.3 RSI Z-Score

`rsi_zscore = (rsi - mean_100w) / std_100w`

A.4 MACD Cross Flag

`+1 if macd crosses above signal, -1 if below, 0 otherwise`

A.5 MA Distance

`close_vs_sma20_pct = (close / sma_20 - 1) * 100`

A.6 EMA Slope

`ema_slope_20 = ema_20_t - ema_20_t-1`

A.7 ATR Percentage

`atr_pct = (atr_14 / close) * 100`

A.8 Volume Z-Score

`volume_zscore = (volume - mean_20) / std_20`

Document Control

Version	Date	Author	Changes
1.0	Nov 2025	System	Initial draft
2.0	Nov 2025	Claude/AI	Combined Phase 1 + AI/LLM specs

End of Document