

# VARIANT PERCEPTION ENGINE

Architecture & Technology Stack

*Systematic discovery of hidden investment opportunities through  
data-driven variant perception analysis*

**Document Version:** 1.0

**Date:** December 2024

**Classification:** Technical Architecture

# EXECUTIVE SUMMARY

The Variant Perception Engine is designed to systematically identify investment opportunities where market consensus diverges from underlying reality. By integrating macro signals, policy developments, and alternative data sources, the system surfaces asymmetric risk/reward setups before they become apparent to the broader market.

## CORE INVESTMENT PRINCIPLES

- 1. **Variant Perception:** Outperformance requires non-consensus insight that is both contrarian AND correct
- 2. **Second-Derivative Thinking:** Focus on where rate of change is accelerating/decelerating
- 3. **Margin of Safety:** Buy at discount to intrinsic value for downside protection
- 4. **Cycle Awareness:** Mean reversion governs extremes—identify regime, adjust positioning

# SYSTEM ARCHITECTURE

The system operates through four integrated layers, each building on the previous to transform raw data into actionable investment opportunities.

## Layer 1: Data Ingestion

The foundation layer captures structured and unstructured data across three primary domains:

Domain	Sources	Key Signals
Macro Signals	FRED, Treasury Direct, BLS, CME	Yields, spreads, PMIs, rate expectations
Policy & Regulatory	Congress.gov, Federal Register, SEC	Legislation, rule proposals, enforcement
Market Data	Yahoo/Polygon, SEC EDGAR, OpenInsider	Prices, 13F filings, insider transactions
Alternative Data	Job boards, app stores, credit cards	Hiring trends, engagement, consumer spend

## Layer 2: Processing & Signal Extraction

Raw data is transformed into quantifiable signals through NLP analysis, regime detection, and consensus extraction algorithms. This layer leverages LLM capabilities for:

- Text-to-signal conversion from earnings transcripts, filings, and news
- Regime detection across macro cycles (early/mid/late cycle classification)
- Consensus extraction: identifying what expectations are priced into current valuations
- Sentiment analysis with financial domain fine-tuning

## Layer 3: Variant Perception Engine

The core algorithmic layer that identifies divergences between market consensus and underlying data signals. The engine scores opportunities based on:

Factor	Weight	Description
Signal Divergence	30%	Gap between alt-data signals and consensus expectations
Catalyst Proximity	20%	Time to event that could reprice the security
Asymmetry Score	25%	Upside/downside ratio under variant vs consensus scenarios

Macro Regime Fit	15%	Alignment with current cycle positioning
Data Quality	10%	Confidence in underlying signal sources

**Layer 4: Idea Generation & Output**

Scored opportunities are packaged with complete thesis documentation including variant view, catalysts, kill criteria, and position sizing recommendations. Each output includes falsifiable hypotheses for ongoing monitoring.

# HIDDEN OPPORTUNITY PATTERNS

The system is designed to detect five primary patterns where market pricing systematically diverges from underlying value:

## **Pattern 1: Credit-Equity Divergence**

When CDS spreads tighten but equity remains flat/down, credit markets see improving fundamentals that equity hasn't priced. The lagging asset class presents the opportunity.

## **Pattern 2: Insider Accumulation in Neglected Names**

Cluster buying by management combined with low analyst coverage and beaten-down price creates asymmetric setups. Management knows more than market.

## **Pattern 3: Policy-Created Structural Shifts**

Regulatory and legislative changes create winners/losers before market prices the impact. Early identification of beneficiaries provides edge.

## **Pattern 4: Second-Derivative Inflections**

Markets focus on direction; edge comes from identifying when rate of decline is slowing or rate of growth is accelerating. First-derivative thinkers miss these inflections.

## **Pattern 5: Macro Regime Mismatch**

Securities trading as if in recession while macro indicators show early cycle recovery. Multiple expansion opportunity as perception catches up to reality.

# TECHNOLOGY STACK

The recommended technology stack balances performance, scalability, and development velocity while optimizing for financial data processing requirements.

## Frontend Layer

Component	Technology	Rationale
UI Framework	React 18+	Component architecture, large ecosystem, excellent for data-heavy dashboards
Styling	Tailwind CSS	Utility-first approach enables rapid iteration on complex financial UIs
State Management	Zustand / TanStack Query	Lightweight state + server state caching for real-time data
Charting	Recharts + D3.js	Recharts for standard charts, D3 for custom financial visualizations
Hosting	Vercel	Edge deployment, automatic scaling, excellent DX

## Backend Layer

Component	Technology	Rationale
API Framework	FastAPI (Python)	Async support, auto-documentation, native ML/data science integration
Task Queue	Celery + Redis	Background processing for data pipelines, scheduled jobs
Orchestration	Dagster	Modern data orchestration with observability, asset-based pipelines
Caching	Redis	Session management, rate limiting, real-time data caching
Hosting	Railway / Render	Simple deployment, auto-scaling, cost-effective for startups

## Data Layer

Component	Technology	Rationale
Primary Database	PostgreSQL	Robust ACID compliance, excellent for financial data integrity

Time-Series	TimescaleDB	PostgreSQL extension optimized for time-series queries on price/macro data
Vector Database	Pinecone / Weaviate	Semantic search on filings, transcripts, research documents
Document Store	S3 + DuckDB	Raw file storage + analytical queries on parquet files

**AI/ML Layer**

Component	Technology	Rationale
LLM Provider	Claude API (Anthropic)	Superior reasoning for financial analysis, long context for filings
RAG Framework	LangChain / LlamaIndex	Document grounding, retrieval pipelines, structured extraction
Embeddings	OpenAI / Cohere	High-quality embeddings for semantic search on financial text
Sentiment Models	FinBERT (fine-tuned)	Domain-specific sentiment classification for financial text

# DATA SOURCES & APIs

The following data sources provide the raw inputs for signal generation. Priority is given to free/low-cost sources for initial development, with premium sources for production scaling.

## Free / Low-Cost Sources

Source	Data Type	Update Freq	Access
FRED API	Macro indicators, yields, spreads	Daily	Free API key
SEC EDGAR	Filings, 13F, insider transactions	Real-time	Free, rate limited
Yahoo Finance	Prices, fundamentals, options	Real-time*	Free (unofficial)
Congress.gov	Legislation tracking	Daily	Free API
OpenInsider	Insider transactions	Daily	Web scrape

## Premium Sources (Production)

Source	Data Type	Est. Cost	Priority
Polygon.io	Market data, options, crypto	\$29-199/mo	High
Quandl/Nasdaq	Alternative data, fundamentals	\$50-500/mo	Medium
Thinknum	Job postings, web traffic, apps	Enterprise	High
Similarweb	Web traffic, engagement	Enterprise	Medium



# IMPLEMENTATION ROADMAP

A phased approach enables rapid validation while building toward full capability. Each phase delivers usable functionality while laying groundwork for the next.

## Phase 1: Foundation (Weeks 1-4)

- Set up PostgreSQL + TimescaleDB infrastructure
- Build data ingestion pipelines (FRED, SEC EDGAR, Yahoo Finance)
- Create basic React dashboard with opportunity card UI
- Implement manual opportunity entry and tracking

## Phase 2: Signal Processing (Weeks 5-8)

- Integrate LLM for transcript/filing analysis
- Build consensus extraction from analyst data
- Implement insider transaction scoring
- Add macro regime classification

## Phase 3: Scoring Engine (Weeks 9-12)

- Develop variant perception scoring algorithm
- Build asymmetry calculation models
- Implement catalyst detection and tracking
- Create kill criteria monitoring system

## Phase 4: Production (Weeks 13-16)

- Add real-time alerting on new opportunities
- Implement portfolio-level analytics
- Build backtesting framework for signal validation
- Deploy production infrastructure with monitoring

## Estimated Monthly Costs (Production)

Item	Service	Monthly
Frontend Hosting	Vercel Pro	\$20

Backend Hosting	Railway	\$50-100
Database	Supabase / Neon	\$25-75
Vector DB	Pinecone Starter	\$70
LLM API	Claude API	\$100-300
Market Data	Polygon.io	\$29-199
	<b>Total Range</b>	<b>\$294-764/mo</b>