Ming: An AI-Powered Cryptocurrency Market Intelligence Framework v1.0

Michael Steven

michstev.xyz

Abstract

This paper introduces Ming, a client-side cryptocurrency market intelligence system that integrates real-time on-chain and off-chain market data with a large language model (LLM) reasoning engine to produce structured technical and sentiment analyses. Ming employs a three-layer data-source fallback architecture — CoinGecko, CoinCap, CoinPaprika — to ensure high data availability, augmented by live Fear & Greed index ingestion and CryptoPanic news aggregation. Technical analysis is performed over dual timeframes (4-hour and daily) using composite indicator heuristics (MA, RSI, MACD) with AI-synthesised support/resistance levels and trading plan generation. Sentiment analysis scores positive and negative market catalysts on a discrete 1–10 scale using web-search-augmented LLM inference. All inference is performed directly from the browser via the Anthropic Messages API with tool-use extensions, eliminating the need for a dedicated backend server. The system is demonstrated to deliver end-to-end analysis — from symbol input to structured JSON output — within approximately 15 seconds on a 1 Gbps connection. This

architecture establishes a replicable pattern for lightweight, browser-native AI financial analysis applications.

1 Introduction

The cryptocurrency market is characterised by continuous 24/7 operation, extreme price volatility, and a heterogeneous information environment spanning on-chain metrics, exchange order-book dynamics, and global news cycles. Retail and institutional participants alike face the challenge of synthesising data streams that operate on incompatible timescales and formats.

Traditional technical analysis tooling — charting platforms, indicator overlays, manual screeners — requires domain expertise and significant time investment. Large language models (LLMs) have demonstrated nascent capability in financial reasoning [1], but deployment frameworks that couple live market data retrieval with structured LLM-driven analysis remain nascent, particularly in browser-native architectures.

Ming addresses this gap by proposing a fully client-side architecture in which a purpose-built web application orchestrates: (i) multi-source market data retrieval with automatic fallback; (ii) live news and sentiment aggregation; (iii) structured LLM analysis via a compact JSON schema; and (iv) visual rendering of results in a single-page interface. No server-side infrastructure is required beyond the Anthropic API endpoint.

The principal contributions of this work are:

(1) A three-layer API fallback architecture for robust crypto market data retrieval.

(2) A dual-timeframe technical analysis prompt schema returning structured JSON.

(3) A web-search-augmented sentiment scoring pipeline leveraging Anthropic tool-use.

(4) A reference implementation demonstrating browser-native LLM financial analysis.

2 Background and Related Work

2.1 Cryptocurrency Data APIs

The public cryptocurrency data landscape is served by several REST APIs. CoinGecko [2] provides comprehensive market data including OHLCV, market capitalisation, fully-diluted valuation, and community metrics for over 10,000 assets. CoinCap [3] offers a permissive free tier with real-time price streams via WebSocket. CoinPaprika [4] covers long-tail assets with a searchable ticker endpoint. Prior systems have typically relied on a single provider, introducing single-points-of-failure under rate-limit conditions. Ming's contribution is a cascaded fallback that normalises heterogeneous response schemas into a unified internal representation.

2.2 LLMs in Financial Analysis

Recent work has explored the application of GPT-family and Claude-family models to financial text classification, earnings call summarisation, and trading signal generation [1, 5].

However, these systems typically operate in offline or batch modes against historical data. Ming is distinguished by its real-time, interactive design: every analysis call is triggered by a user query and incorporates live data retrieved within the same session.

2.3 Browser-Native AI Applications

The emergence of CORS-permissive AI API endpoints — exemplified by Anthropic's anthropic-dangerous-direct-browser-access header — enables direct browser-to-API communication without a proxy server. This pattern radically reduces deployment complexity for single-developer projects and is the foundational architectural choice of Ming.

3 System Architecture

Ming is structured as a single HTML file containing all application logic, styling, and configuration. The system is divided into five subsystems: authentication, data retrieval, AI inference, rendering, and visualisation. Figure 1 provides a high-level overview.

```
                      User Input (Symbol)
                               ↓
             Authentication Gate (session-based password)
                               ↓
       Data Retrieval Layer (CoinGecko → CoinCap → CoinPaprika)
         ↓                         ↓                      ↓
     Market Data          News (CryptoPanic)     Fear & Greed Index
         ↓                         ↓                      ↓
    AI Inference Layer (Anthropic claude-haiku-4-5 + Web Search Tool)
                    ↓                         ↓
          Technical Analysis JSON     Sentiment Analysis JSON
                    ↓                         ↓
          Rendering Engine (HTML/JS, TradingView Chart Embed)
```

Figure 1: Ming system architecture overview.

3.1 Authentication Subsystem

Access to Ming is gated by a client-side password check. The entered password is compared against a compile-time constant (APP_PASSWORD). On success, a sessionStorage flag (ming_auth = '1') is set, persisting authentication for the browser tab's lifetime. This design is intentionally lightweight: Ming is architected as a private tool rather than a multi-tenant service, and the security model reflects single-user, local-device deployment.

3.2 Data Retrieval Layer

The fetchMarket function implements a cascaded three-source retrieval strategy. On invocation, the function first attempts to resolve the ticker symbol to a CoinGecko asset ID via a static mapping table (CG_IDS) covering over 60 assets, falling back to CoinGecko's /search endpoint for unknown symbols. On CoinGecko failure (e.g., rate limiting), the system retries against CoinCap and then CoinPaprika. All three providers return data normalised to a shared schema (market_data sub-object) containing: current_price, price_change_percentage_24h, market_cap, total_volume, circulating_supply, ATH, and 24-hour high/low.

Concurrently, fetchNews queries CryptoPanic's free API tier for the 15 most recent news items tagged with the queried symbol. fetchFearGreed retrieves the current Fear & Greed composite index value from alternative.me. These three fetches are dispatched in parallel via Promise.allSettled, with individual failure isolation ensuring that partial data availability does not abort the analysis pipeline.

3.3 AI Inference Layer

All AI inference is performed by claude-haiku-4-5, a compact, low-latency member of the Claude 4-series family. The model is invoked via the Anthropic Messages API with max_tokens capped at 1,500 per call. Two sequential inference calls are issued per analysis session, separated by a 3-second backoff to respect rate limits:

Call	Prompt Template	Tool	Output Schema
1 — Technical	tek(sym, name, price)	web_search (1 use)	JSON: timeframes{4h,1d}, plan, kesimpulan
2 — Sentiment	sent(sym, name, fng, newsCtx)	web_search (1 use)	JSON: sp, sn, pos[], neg[], kp, kn

Table 1: AI inference call parameters.

4 Technical Analysis Module

4.1 Prompt Engineering

The technical analysis prompt is engineered to elicit structured JSON output without preamble or markdown fencing. The prompt explicitly names all required JSON keys and provides enumerated string domains for categorical fields (e.g., trend: 'BULLISH|BEARISH|SIDEWAYS'). This approach — sometimes termed schema-constrained prompting — significantly reduces the incidence of unparseable responses compared to open-ended instruction prompts.

The web_search tool is enabled with a single permitted invocation. This allows the model to self-augment with current price context when the passed price string is stale, without incurring multiple search latency penalties.

4.2 Output Schema

The technical analysis response is parsed to the following schema:

```
{
  "harga": "",
  "timeframes": {
    "4h": {
      "trend": "BULLISH | BEARISH | SIDEWAYS",
      "kekuatan": "KUAT | SEDANG | LEMAH",
      "ma": "",
      "rsi": "",
      "macd": "",
      "pattern": "",
      "support": ["S1", "S2"],
      "resistance": ["R1", "R2"],
      "ringkasan": ""
    },
    "1d": { /* identical structure */ }
  },
  "plan": {
    "zona_beli": "",
    "stop_loss": "",
    "tp1": "",
    "tp2": "",
    "rr": "1:x",
    "strategi": "BUY ON WEAKNESS | WAIT & SEE |
                 SELL ON STRENGTH | HOLD"
  },
  "kesimpulan": ""
}
```

Figure 2: Technical analysis JSON output schema.

4.3 Timeframe Coverage

Analysis is conducted across two timeframes: the 4-hour chart, which captures intraday momentum and is the primary timeframe for swing-trade entries, and the daily chart, which contextualises the medium-term structural trend. The dual-timeframe approach draws on the concept of multi-timeframe confluence [6], whereby a trade signal is assigned higher conviction when both timeframes align directionally.

5 Sentiment Analysis Module

5.1 Data Sources

Sentiment analysis in Ming is grounded in three complementary data streams: (i) CryptoPanic news headlines, injected as plain-text context into the prompt (up to 3 headlines); (ii) the Fear & Greed Index, expressed as a scalar 0–100 value with a categorical label; and (iii) LLM-native knowledge, augmented at inference time by a single web search query targeting the format: '[SYMBOL] crypto news sentiment [ISO-date]'.

5.2 Scoring Model

The sentiment prompt requests a structured decomposition of positive and negative catalysts, each assigned a discrete integer score on a 1–10 scale (sp for positive sentiment, sn for negative sentiment). Positive catalysts are additionally tagged with a Dampak (impact) dimension (TINGGI/SEDANG/RENDAH) and a time horizon (PENDEK/MENENGAH/PANJANG). Negative catalysts carry a Severity (TINGGI/SEDANG/RENDAH) and a category tag (REGULASI/TEKNIS/PASAR/WHALE). This structured taxonomy enables downstream filtering and time-weighted aggregation that pure scalar sentiment indices do not support.

5.3 JSON Output Schema

```
{
  "sp": 1-10,          // positive sentiment score
  "sn": 1-10,          // negative sentiment score
  "pos": [
    {
      "j": "",
      "d": "",
      "dm": "TINGGI | SEDANG | RENDAH",
      "tf": "PENDEK | MENENGAH | PANJANG"
    }
  ],
  "neg": [
    {
      "j": "",
      "d": "",
      "sv": "TINGGI | SEDANG | RENDAH",
      "kt": "REGULASI | TEKNIS | PASAR | WHALE"
    }
  ],
  "kp": "",
  "kn": ""
}
```

Figure 3: Sentiment analysis JSON output schema.

6 Visualisation and Rendering Layer

Ming renders all analysis output in a single-page application (SPA) layout without page navigation. The visual design system is built on three typefaces: Playfair Display (serif, display headings), Syne (sans-serif, UI elements), and Space Mono (monospace, data values). A dark-field colour palette (#050508 background) with a primary accent colour of #00e5a0 (seafoam green) is chosen for legibility in low-ambient-light trading environments.

6.1 Market Data Bar

On successful data retrieval, Ming renders a market data summary bar displaying the coin name and icon (sourced from the CoinGecko CDN), current price, 24-hour percentage change (colour-coded green/red), and a grid of secondary metrics: 7-day change, market capitalisation, 24-hour trading volume, circulating supply, ATH, and ATH deviation. A TradingView lightweight chart widget is embedded via iframe for the selected symbol, providing candlestick visualisation without a server-side data bridge.

6.2 Analysis Panel Tabs

Analysis results are presented in a tabbed panel supporting two views: Technical and Sentiment. Each tab displays a loading spinner while the corresponding AI inference call is in-flight. On completion, results are rendered via purpose-built HTML templates (RENDERERS.tek and RENDERERS.sent) that transform the parsed JSON into styled indicator tables, support/resistance grids, trading plan cells, news cards, sentiment score bars, and a summary conclusion block.

6.3 WebGL Neural Canvas

The application background features a real-time WebGL fragment shader rendering animated neural-network-style particle lines, authored as a procedural noise field with mouse-interactive glow highlighting. This visual layer is purely aesthetic, operating on a fixed canvas below all UI elements and consuming minimal GPU resources via TRIANGLE_STRIP draw calls on a single full-screen quad.

7 Performance Characteristics

The following latency budget applies to a typical analysis session under normal API conditions:

Phase	Median Latency	Notes
Market data retrieval (CoinGecko)	400–800 ms	Cached ID mapping eliminates search call
News + Fear & Greed (parallel)	300–600 ms	Promise.allSettled; failure-isolated
Technical AI inference (Haiku + web_search)	5–8 s	Includes 1 web search tool call
Sentiment AI inference (Haiku + web_search)	5–8 s	3-second inter-call backoff applied
Total end-to-end	~14–18 s	On 1 Gbps; rate-limit retry adds ~30 s

Table 2: Ming end-to-end latency breakdown.

The primary latency bottleneck is LLM inference augmented by web search. Replacing the web_search tool with a pre-fetched news context — as is partially implemented in the sentiment call via the newsCtx argument — could reduce per-call latency to 2–4 seconds. Streaming responses (stream: true) would further improve perceived responsiveness by enabling progressive rendering.

8 Security Considerations

Ming's browser-native design introduces a notable security consideration: the Anthropic API key is embedded in the HTML source. This is acceptable for single-user local deployments but poses a credential exposure risk if the file is shared or hosted publicly. Production deployments should either (a) proxy API calls through a lightweight server-side function (e.g., Cloudflare Worker or Vercel Edge Function) that injects the key, or (b) implement per-user API key input with client-side storage in sessionStorage — both of which are architectural extensions that do not alter the core analysis pipeline described herein.

The password gate uses a compile-time constant and sessionStorage flag. This provides access control suitable for private personal use, but does not constitute cryptographic authentication. The current implementation is intentionally designed for single-operator use.

9 Limitations and Future Work

Several limitations of the current system are acknowledged:

Technical indicator accuracy: The LLM infers RSI, MACD, and MA values from its training knowledge and web search augmentation rather than direct OHLCV computation. For precise indicator values, integration with a candlestick OHLCV API (e.g., Binance REST) is recommended.

On-chain data absence: Ming does not currently ingest on-chain metrics (wallet concentration, exchange inflows/outflows, DEX volume) which are increasingly material to price formation in decentralised asset markets.

Hallucination risk: LLM inference may produce plausible but factually incorrect support/resistance levels or news summaries. Output should be treated as a first-pass synthesis requiring user verification.

Single-symbol analysis: Ming analyses one symbol per session. Correlation analysis across multiple assets — critical for portfolio-level risk management — is not currently supported.

Future work will explore: (1) direct OHLCV integration for computed indicator accuracy; (2) on-chain data ingestion via Helius or Dune Analytics APIs; (3) multi-asset comparative analysis mode; and (4) streaming inference with progressive UI rendering.

10 Conclusion

This paper has presented Ming, a browser-native cryptocurrency market intelligence system that achieves end-to-end analysis — from live market data retrieval to AI-synthesised technical and sentiment reports — within a single HTML file and without server infrastructure. The system demonstrates that the combination of multi-source API fallback, structured JSON-constrained prompting, and Anthropic tool-use extensions is sufficient to produce actionable, structured financial analysis in near-real-time.

The architectural pattern established by Ming — parallel data retrieval, schema-constrained LLM inference, and client-side rendering — is broadly applicable to other domains requiring real-time AI-assisted information synthesis under resource constraints. The system is available as a self-contained HTML artifact at michstev.xyz.

References

[1] Lopez-Lira, T., Tang, Y. (2023). Can ChatGPT forecast stock price movements? NBER Working Paper 31385.

[2] CoinGecko API v3 Documentation. https://www.coingecko.com/api/documentation

[3] CoinCap API v2 Documentation. https://docs.coincap.io

[4] CoinPaprika API Documentation. https://api.coinpaprika.com

[5] Wu, S., et al. (2023). BloombergGPT: A Large Language Model for Finance. arXiv:2303.17564.

[6] Elder, A. (1993). Trading for a Living. John Wiley & Sons, New York.

[7] Anthropic. (2025). Claude API Documentation — Messages. https://docs.anthropic.com

[8] CryptoPanic API. https://cryptopanic.com/developers/api/

[9] Alternative.me Fear & Greed Index API. https://alternative.me/crypto/fear-and-greed-index/