Amongus @ FinMMEval 2026 Task 3: Agentic Mk1 with Structured Multi-Agent Trading and TimesFM Forecasts

Pawaris Panyasombat[1], Thanavin Denkavin[1] and Natthawut Kertkeidkachorn[2]

[1] *Sirindhorn International Institute of Technology, Thailand*
[2] *Japan Advanced Institute of Science and Technology, Ishikawa, Japan*

Abstract
This paper describes Agentic Mk1, our submission to FinMMEval 2026 Task 3: Financial Decision Making. The task evaluates daily trading endpoints that receive market prices, recent news, momentum labels, historical prices, and optional SEC filing excerpts, and must return one of three actions: BUY, HOLD, or SELL. Our system combines deterministic technical-analysis features, zero-shot time-series forecasts from TimesFM, and a three-stage large-language-model decision pipeline. The language-model layer uses Qwen2.5-7B-Instruct with constrained structured generation to produce a market analysis, a risk assessment, and a final trade decision. The deterministic layer maintains per-symbol rolling state, technical indicators, regime labels, stop-loss and take-profit rules, and safe fallback behavior. The camera-ready implementation follows the official Task 3 action semantics exactly: BUY maps to long exposure, HOLD maps to flat exposure, and SELL maps to short exposure. We report the frozen official Task 3 snapshot for camera-ready reporting and local ablation results over the released Task 3 BTC and TSLA data.

Keywords
financial decision making, trading agents, large language models, time-series forecasting, technical analysis, FinMMEval

1. Introduction

Financial decision-making benchmarks increasingly test not only factual financial knowledge, but also the ability of trading systems to integrate noisy signals under operational constraints. FinMMEval 2026 introduces a multilingual and multimodal financial evaluation suite spanning question answering, summarization, and decision making [1]. We participated in Task 3, Financial Decision Making, where each submitted endpoint is queried once per trading day and must return one of three discrete actions: BUY, HOLD, or SELL [2, 3].

Our submission, Agentic Mk1, is a lightweight but stateful trading endpoint. The system is motivated by a practical observation: language models are useful for synthesizing heterogeneous evidence, but live trading endpoints require deterministic safeguards. We therefore treat the large language model (LLM) as an interpretive layer rather than as an unconstrained policy. Technical indicators, position accounting, allowed-action filtering, stop-loss and take-profit rules, and fallback logic are computed outside the model. The LLM receives these stationary and normalized features, together with news snippets, filings, momentum labels, recent action memory, and an enriched TimesFM forecast summary.

The main contributions of this system paper are as follows:

- We describe a three-agent LLM chain for daily trading decisions: market analyst, risk manager, and final decision layer.
- We combine LLM reasoning with deterministic regime classification based on SMA distance, RSI, Bollinger-band position, MACD direction, volatility, and recent momentum.
- We enrich each request with a 32-day TimesFM forecast summarized into slope, trend fit, average return, segment trajectory, bullish-day ratio, and forecast extrema.

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany
EMAIL: panyasombatpawaris@gmail.com (P. Panyasombat); thanavindenkavin@gmail.com (T. Denkavin); natt@jaist.ac.jp (N. Kertkeidkachorn)

- We implement robust endpoint behavior through schema-constrained generation, per-symbol state isolation, rolling price buffers, forced risk exits, and safe HOLD defaults.

2. Task Overview

FinMMEval Task 3 is a live, news-driven trading workflow. For each accepted participant endpoint, the organizer sends a daily JSON request containing the date, current close price, symbol, recent news, momentum information, recent price history, and optionally 10-K or 10-Q excerpts. The endpoint must return a JSON object containing a single action from the set {BUY, HOLD, SELL}. In the official setup, BUY maps to a long position, HOLD maps to a flat position, and SELL maps to a short position. A new action fully replaces the previous day's position, and performance is evaluated over a live evaluation window. The primary metric is cumulative return, with secondary metrics including Sharpe ratio, maximum drawdown, daily volatility, and annualized volatility [2, 1, 3, 4].

This evaluation setting differs from static prediction tasks in three ways. First, the model must operate as a service with latency and availability constraints. Second, decisions have sequential consequences because prior positions influence risk. Third, the endpoint must remain robust to sparse daily context. The organizer request may include only a short price history, which is insufficient for indicators such as 50-day moving averages. Agentic Mk1 addresses these issues with stateful per-symbol buffers and deterministic safety behavior.

3. System Overview

Agentic Mk1 is implemented as a FastAPI service. At startup, it loads Qwen2.5-7B-Instruct [5] in 4-bit quantized form, compiles constrained generators for three Pydantic output schemas, and optionally loads TimesFM [6]. The service exposes a health endpoint and a `/trading_action/` endpoint compatible with the Task 3 request format.

The full decision pipeline is shown in Figure 1. Each request is parsed, merged into a per-symbol rolling price buffer, transformed into technical indicators, enriched with a time-series forecast, passed through a three-agent LLM chain, filtered by deterministic trading rules, written into a short-term memory buffer, and returned as a single competition action.

3.1. Base Model and Structured Generation

The LLM component uses Qwen2.5-7B-Instruct. We selected this model because it provides strong instruction following and structured-data reasoning while remaining small enough to serve on a single consumer GPU under 4-bit quantization. The model is loaded through Hugging Face Transformers with `bitsandbytes` 4-bit NF4 quantization [7]. To reduce malformed outputs, we use Outlines-style constrained generation [8] with Pydantic schemas for the three model-produced objects:

- **MarketAnalysis**: news bias, macro view, volatility level, momentum signal, and summary.
- **RiskAssessment**: position state, risk level, preferred action, conviction, and risk note.
- **TradeDecision**: final action, conviction, and reason.

If structured generation fails, the system logs the failure and falls back to deterministic rules. The endpoint returns HOLD on unrecoverable errors, matching the competition-safe default behavior.

3.2. Per-Symbol State

The initial prototype used a single global position state. We revised this into a dictionary keyed by symbol, so BTC and TSLA requests cannot contaminate one another. Each symbol state stores the current position sign, average entry price, peak price since buy, bars since buy, and a rolling price

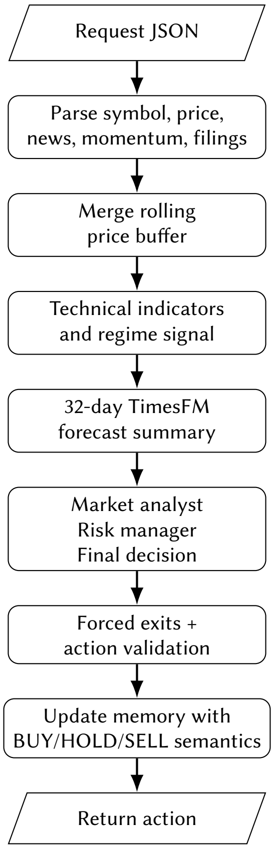

Figure 1: Agentic Mk1 inference pipeline for one Task 3 request.

buffer. A separate rolling memory stores compact daily decision records for each symbol. This memory is passed to the risk and decision prompts as `last_trades`.

This design matters because Task 3 can send requests for different assets. A shared state could incorrectly apply a BTC position to TSLA or vice versa. Per-symbol state also allows technical indicators to mature independently.

4. Feature Engineering

4.1. Rolling Price Buffer

The organizer request may include a short recent price history. This is useful for immediate context but insufficient for longer-window indicators such as SMA-50, MACD, and volatility estimates. Agentic Mk1 therefore keeps an accumulated price buffer per symbol. Incoming rows are merged by date, request prices override previous values on conflicts, and the buffer is capped at 500 rows. At inference time, the last 50 rows are used for technical analysis.

When fewer than 10 total rows are available, the endpoint returns HOLD. When fewer than 50 rows are available, the endpoint continues but logs that technical indicators are partially reliable. NaN indicator values are replaced by neutral defaults such as RSI 50, Bollinger position 0.5, and zero SMA distance.

4.2. Technical Indicators

The technical-analysis module computes:

- distance from 10-, 20-, and 50-day simple moving averages;
- macro trend, defined by whether price is above or below SMA-50;

- 14-day RSI and RSI zone;
- Bollinger-band percentile and zone;
- MACD-vs-signal direction;
- 14-day realized volatility;
- 10-day price momentum.

These indicators are converted into a discrete regime signal using fixed thresholds. RSI below 35 is treated as oversold, and RSI above 65 is treated as overbought. The macro trend is bullish when the close is above SMA-50 and bearish when it is below SMA-50. Negative momentum is defined as 10-day momentum below -5%. The regime `bull_momentum_buy_ok` requires price above SMA-50 together with bullish MACD direction. The regime `overbought_momentum_fading` requires RSI above 65 and bearish MACD direction. The regime `strong_bear_avoid_buy` requires a bearish macro trend, 10-day momentum below -5%, and RSI above 40. The regime `bear_oversold_bounce` requires RSI below 35 and improving MACD direction. For risk exits, the dynamic stop threshold is $\min(\max(5\%, 2 \times \text{volatility}_{14d}), 25\%)$, and take-profit is triggered when unrealized long-position return is at least 10% together with overbought or fading momentum. This regime label is included in all three LLM prompts and also controls deterministic forced exits.

4.3. TimesFM Forecast Summary

Agentic Mk1 optionally loads TimesFM 2.5, a decoder-only foundation model for time-series forecasting. Given the most recent close-price window, the system forecasts a 32-day horizon. Instead of passing all forecasted prices directly into the LLM, we summarize the forecast into stationary features:

- linear slope in dollars per day and R^2 over the 32-day forecast;
- average forecast return versus the current price;
- early, middle, and late segment returns, using forecast days 1–10, 11–20, and 21–32 respectively;
- trajectory shape, such as accelerating, peak-then-fade, or dip-then-recover;
- number and percentage of forecast days above the current price;
- peak and trough returns and their forecast days.

This summary is intentionally descriptive. The final action is not determined by TimesFM alone; instead, the forecast acts as an additional signal for the market analyst prompt. If TimesFM is unavailable or fails, the system continues without the signal.

5. Agent Chain

5.1. Market Analyst

The market analyst receives technical indicators, the regime signal, position metrics, news snippets, filing context, and the TimesFM summary. Its prompt instructs it to interpret stationary metrics rather than recalculate raw prices. This reduces arithmetic burden on the LLM and focuses the model on synthesis: whether signals are bullish, bearish, or mixed, and whether the macro view is a bull or bear market.

5.2. Risk Manager

The risk manager receives the current portfolio state, position metrics, allowed actions, bars since buy, technical indicators, regime signal, and recent decision memory. Its task is to protect capital while identifying asymmetric entries. The prompt includes hard constraints: never recommend actions outside `allowed_actions`; choose HOLD when a dynamic stop is triggered on a long position, because HOLD exits to flat in the official evaluator; favor buys for oversold bounce and bull momentum regimes; and avoid repeated losing patterns observed in `last_trades`.

5.3. Final Decision Layer

The final decision layer receives both prior agent outputs and the same structured technical context. Its prompt specifies a strict priority ordering: forced stop-loss exits first, take-profit exits second, then bull or oversold-bounce buy regimes, then macro-bear conditions, and finally conviction-based filtering. If the final action is outside the allowed set, the system rewrites it to HOLD.

The LLM output is subsequently passed through deterministic exit enforcement. Thus, even if the LLM misses a stop-loss or take-profit condition, the endpoint still executes the safety rule when the action is allowed.

6. Risk Controls

Agentic Mk1 includes several deterministic risk controls:

- **Allowed-action filtering**: actions are restricted by current cash, position, minimum hold bars, and stop-loss conditions.
- **Minimum buy conviction**: weak buy signals are converted to HOLD outside explicitly favorable regimes.
- **Cash reserve**: simulated local backtesting keeps a cash reserve when sizing long positions.
- **Dynamic trailing stop**: stop-loss distance scales with recent realized volatility, bounded by a minimum and maximum threshold.
- **Take-profit logic**: profitable positions are exited when gains coincide with overbought or fading-momentum signals.
- **Safe fallback**: malformed LLM output or unexpected endpoint exceptions return HOLD.

In the camera-ready implementation, BUY sets the symbol state to long, SELL sets it to short, and HOLD sets the symbol state to flat. HOLD also clears the internal entry price, peak price, bars-since-entry counter, and stop-loss state. This reconciles the code with the official Task 3 evaluator: each action fully replaces the previous day's exposure, so the risk manager cannot apply trailing-stop logic to a position that the evaluator has already closed. The earlier prototype preserved internal position state across HOLD outputs for bookkeeping, but that behavior was a mismatch with the official semantics and has been removed.

7. Implementation Details

The service is implemented in Python. The API layer uses FastAPI and Pydantic request and response models. We explicitly reparse the raw JSON body to recover digit-prefixed keys such as `10k` and `10q`, which cannot be directly represented as normal Python field names. The model layer uses Transformers, PyTorch, `bitsandbytes`, and Outlines. The forecasting layer uses TimesFM when installed; otherwise, it logs that the forecast signal is disabled and continues.

The endpoint performs the following steps for each request:

1. identify the first requested symbol and current close price;
2. merge recent price history into the symbol's accumulated buffer;
3. compute technical indicators and substitute neutral defaults for NaNs;
4. compute a regime signal and position metrics;
5. generate a TimesFM forecast summary if the model is available;
6. run the market, risk, and decision structured generators;
7. apply fallback logic if any LLM stage fails;
8. enforce forced exits and validate the action;
9. update live state and rolling memory;
10. return the final uppercase action.

Table 1
Frozen official Task 3 snapshot for camera-ready reporting. Return values are shown after transaction costs; no-fee values are given in parentheses.

Agent	Asset	Days	Closed	Return	Vs. B&H	Sharpe	Win Rate	Final Balance
OhAzdaha	BTC	47	0	+2.38% (+2.62%)	+29.63%	1.47	100.00%	$102,375.71
OhAzdaha	TSLA	33	14	-2.42% (-2.18%)	+12.41%	-2.95	50.00%	$97,584.03

Table 2
Local full-model backtest metrics over the released Task 3 data.

Asset	Days	Return	Sharpe	MaxDD	DailyVol	AnnVol	B&H	Alpha	Win Rate
BTC	253	+46.52%	1.62	20.00%	1.61%	25.61%	-36.45%	+82.97%	56.2%
TSLA	252	-10.56%	-0.52	20.92%	1.16%	18.41%	+2.28%	-12.84%	31.6%

8. Experimental Setup

We evaluated the prototype locally using the released Hugging Face Task 3 BTC and TSLA data for backtesting and validation. The simulation mode iterates over sorted daily rows using a 50-day rolling window. A decision made at close t determines exposure over t to $t + 1$. Exposure is +1 for BUY, 0 for HOLD, and -1 for SELL. Transaction fees are applied when exposure changes, and a no-fee equity curve is also recorded. The harness tracks equity, fees, trade history, maximum drawdown, Sharpe ratio, daily volatility, annualized volatility, buy-and-hold return, and net alpha. The same technical-analysis, forecasting, memory, fallback, and exit-enforcement functions are shared between simulation mode and live API mode.

The deployed configuration used:

- base LLM: Qwen2.5-7B-Instruct;
- quantization: 4-bit NF4;
- forecasting model: TimesFM 2.5 200M PyTorch when available;
- history window for technical analysis: 50 daily bars;
- rolling LLM memory: 10 daily decision records;
- take-profit threshold: 10%;
- minimum stop-loss threshold: 5%;
- dynamic stop threshold: $\min(\max(5\%, 2 \times \text{volatility}_{14d}), 25\%)$;
- fallback action: HOLD.

9. Results

For camera-ready reporting, we use the frozen official Task 3 snapshot released by the organizers [2]. The final evaluation window differs by asset: BTC was evaluated from 2026-05-11 to 2026-07-04, while TSLA was evaluated from 2026-05-11 to 2026-06-26. The ranking metric is cumulative return under the Long/Short setting, averaged across BTC and TSLA. Organizer-side payload issue dates are treated uniformly as HOLD, i.e., flat exposure, for all agents. The reported BTC and TSLA returns in Table 1 follow this final official snapshot.

Table 2 reports the local backtest metrics requested by the reviewers for the full model. These results are not substituted for the official leaderboard snapshot; they are included to make the local harness output explicit and reproducible.

10. Ablation Results

We ran endpoint-level ablations against the same released data and reporting harness. Each variant changes one component while keeping the official BUY/HOLD/SELL exposure mapping fixed. The

Table 3
Local ablation results from the released Task 3 BTC and TSLA data. Actions show counts of BUY/HOLD/SELL decisions.

Asset	Variant	Days	Return	Sharpe	MaxDD	B&H	Alpha	Actions
BTC	fullmodel	253	+46.52%	1.62	20.00%	-36.45%	+82.97%	37/158/58
BTC	noforcedexits	253	+51.08%	1.53	13.75%	-36.45%	+87.53%	44/147/62
BTC	nomemory	253	+16.52%	1.04	7.35%	-36.45%	+52.97%	47/206/0
BTC	nopricebuffer	253	-11.06%	-0.26	22.85%	-36.45%	+25.38%	45/145/63
BTC	notimesfm	253	-1.06%	0.07	15.65%	-36.45%	+35.38%	35/140/78
TSLA	fullmodel	252	-10.56%	-0.52	20.92%	+2.28%	-12.84%	33/190/29
TSLA	noforcedexits	253	-0.19%	0.09	16.24%	+2.28%	-2.47%	33/164/56
TSLA	nomemory	253	+8.43%	0.59	14.26%	+2.28%	+6.15%	44/209/0
TSLA	nopricebuffer	166	-3.51%	-0.13	16.81%	-7.29%	+3.78%	30/105/31
TSLA	notimesfm	253	+14.27%	0.76	7.88%	+2.28%	+11.98%	36/167/50

`no_timesfm` variant removes the forecast summary, `no_memory` removes recent decision memory, `no_forced_exits` disables deterministic stop-loss and take-profit overrides, and `no_price_buffer` computes indicators from request history without the accumulated preload buffer.

The ablations show that memory and forecast context interact differently by asset. On BTC, the full model outperforms `notimesfm`, `nomemory`, and `nopricebuffer`, indicating that both forecast context and the longer price buffer were useful in the local window. On TSLA, `notimesfm` and `nomemory` outperform the full model, suggesting that the forecast and recent action memory can overfit during equity-specific reversals. The `no_forced_exits` variant improves local returns for both assets, but we retain forced exits in the main system because they provide deterministic protection against malformed outputs and clarify risk behavior under the official HOLD-as-flat semantics.

11. Error Analysis

During development, we observed several likely failure modes. First, short histories produce unreliable RSI, Bollinger, and SMA features. The rolling price buffer mitigates this after sufficient live operation, but the earliest days remain weaker. Second, LLM outputs can fail schema validation or under-specify actions; constrained generation and fallback rules reduce but do not eliminate this risk. Third, forecast models can extrapolate recent trends into unstable market regimes; this is why TimesFM is treated as a soft context feature rather than as a direct policy. Fourth, live endpoint operation requires robust handling of missing filings, null fields, and unexpected symbol order.

12. Reproducibility

The system is reproducible from the Python source code and configuration constants. The main external components are Qwen2.5-7B-Instruct, Hugging Face Transformers, PyTorch, `bitsandbytes`, Outlines, FastAPI, Pydantic, pandas, NumPy, and TimesFM. The endpoint uses no real-time external retrieval during inference. It uses only the information supplied by the organizer request, optional local historical preload files, and the endpoint's accumulated per-symbol state. In local ablation runs, the server preloads the released Hugging Face `TheFinAI/CLEF_Task3_Trading` rows before each request date and then merges the organizer-style request history. In live deployment, the endpoint may preload local BTC/TSLA history files if present, then continues from the organizer's rolling daily-request buffer.

Structured generation used maximum new-token limits of 300 for `MarketAnalysis`, 250 for `RiskAssessment`, and 200 for `TradeDecision`. The fallback Transformers path used greedy decoding with `do_sample=False`; temperature and top-p sampling were not set. No stochastic seed was required for the greedy path. We used one structured generation attempt per stage and then deterministic fallback on schema failure or model error. The deployed dependency log included Transformers 4.47.0, PyTorch 2.12.0.dev20260407+cu128, `bitsandbytes` 0.45.5, Outlines 1.2.12, TimesFM 2.0.0, CUDA 12.8

Table 4
System card summary for Agentic Mk1.

Item	Description
Task	FinMMEval 2026 Task 3: Financial Decision Making
Endpoint output	One of BUY, HOLD, SELL
Architecture	FastAPI endpoint with deterministic feature extraction, TimesFM forecast summarization, three structured LLM agents, and rule-based safety filters
Base LLM	Qwen2.5-7B-Instruct, loaded with 4-bit NF4 quantization
Forecast model	TimesFM 2.5 200M PyTorch when available; disabled gracefully otherwise
Training	No task-specific fine-tuning; zero-shot/prompted inference
External retrieval	None during inference
Data usage	Organizer request fields, optional local historical preload files, and process-local rolling state
Main risks	Forecast overconfidence, regime shifts, process-local state loss, sparse initial history, and LLM schema failures
Safety behavior	HOLD on unrecoverable errors; deterministic stop-loss/take-profit enforcement; action validation
Hardware	Single-GPU local deployment; tested with a 4-bit model configuration intended for approximately 9 GiB GPU memory plus CPU offload
Artifacts	Working-notes paper, BibTeX references, system card, results JSONL where applicable

runtime libraries, FastAPI 0.136.1, Pydantic 2.13.3, pandas 3.0.2, and NumPy 1.26.4. The model identifiers were `Qwen/Qwen2.5-7B-Instruct` and `google/timesfm-2.5-200m-pytorch`; no task-specific fine-tuning was applied.

The local simulation writes two JSONL artifacts: a trade-history file containing decisions, market and risk views, technical indicators, and position metrics; and a TimesFM forecast file containing daily forecast outputs and derived forecast returns. Logging records daily indicators, regime signals, allowed actions, trades, and summary portfolio statistics.

13. System Card and Submission Compliance

Table 4 summarizes the main system-card information requested by the FinMMEval working-notes packaging checklist. Agentic Mk1 is submitted only for Task 3, so the Task 1/2 disclosure requirement for dataset expansion or real-time retrieval is not applicable. For Task 3, the system does not call external search engines, market-data APIs, or news APIs during inference; all non-model input is provided by the organizer request or by local state accumulated from prior organizer requests.

13.1. License and Model Compliance

The implementation depends on open-source Python libraries and publicly released model weights. The paper cites the FinMMEval lab overview, the official Task 3 overview, the official FinMMEval lab page, and the Agent Market Arena / When Agents Trade source paper. We also cite the technical references for Qwen2.5, TimesFM, QLoRA-style quantization, and constrained generation. Any final submission should verify that the deployed model licenses and platform terms are compatible with the participant's institution and intended use.

14. Limitations

Agentic Mk1 remains a heuristic system. The LLM is not trained or fine-tuned specifically for Task 3, and the decision prompts encode hand-designed rules. Technical indicators may underperform in sudden regime changes, and TimesFM forecasts may be poorly calibrated for assets with event-driven jumps. The live state is process-local unless the deployment persists or preloads state across restarts. Finally, the current system optimizes robustness and interpretability more than aggressive return seeking; this may reduce upside in strongly trending markets.

15. Conclusion

We presented Agentic Mk1, a structured multi-agent trading endpoint for FinMMEval 2026 Task 3. The system combines deterministic technical analysis, stateful risk controls, schema-constrained LLM reasoning, and enriched TimesFM forecasting. The camera-ready implementation resolves the HOLD ambiguity by treating HOLD as flat exposure and clearing stale internal position state. We report the frozen official Task 3 snapshot and completed local ablations, showing where forecast context, memory, forced exits, and price-buffer history affect BTC and TSLA differently.

A. Prompt Templates

A.1. Market Analyst

```
You are a market analyst analyzing stationary, relative metrics.
You receive percentage-based distance from SMAs, a regime_signal, and recent news.
Interpret these stationary features; do not recalculate raw prices.

regime_signal meanings:
- dynamic_stop_loss_exit    : long position dropped by the dynamic trailing-stop threshold.
- take_profit_exit        : long position is profitable and overbought or momentum is fading.
- macro_bear_avoid_buy       : price is below the 50-day moving average.
- overbought_momentum_fading : RSI is overbought and MACD is bearish.
- strong_bear_avoid_buy      : bear trend, negative momentum, RSI not oversold.
- bear_oversold_bounce       : oversold plus MACD turning bullish.
- bull_momentum_buy_ok       : price above SMA-50 with bullish MACD.
- neutral                    : mixed signals.

Task 3 action semantics:
- BUY means long exposure.
- HOLD means flat exposure.
- SELL means short exposure.
Each action fully replaces the previous day's position.

A forecast field contains a 32-day quantitative signal:
- Trend dollars/day and r2: linear regression slope across forecast days.
- Avg return: mean predicted price versus current price.
- Shape early->mid->late: days 1-10, 11-20, and 21-32.
- Bullish days: fraction of forecast days above current price.
- Peak/trough days: timing of extremes.

Write a 1-2 sentence summary referencing specific indicator values.
```

A.2. Risk Manager

```
You are a portfolio risk manager. Protect capital but seek asymmetric entries.

The last_trades field contains recent daily decisions for this symbol. Use it to avoid repeating
losing patterns and to understand recent momentum context.

Rules:
- Never recommend an action outside allowed_actions.
- BUY means long, HOLD means flat, and SELL means short.
- If dynamic_stop_triggered is true while long, preferred_action MUST be hold, because HOLD exits to f
- If take_profit_triggered is true while long, preferred_action SHOULD be hold.
- bear_oversold_bounce favors BUY.
- bull_momentum_buy_ok favors BUY.
- strong_bear_avoid_buy favors SELL when not already closing a long.
- macro_bear_avoid_buy favors HOLD or SELL, depending on downside strength.
- If last_trades shows repeated weak HOLD actions while downside signals persist, consider SELL.

Write a 1-2 sentence risk_note.
```

A.3. Final Decision Layer

```
You are the final trade decision layer.

The last_trades field contains recent daily decisions for this symbol. Use it to determine whether
the current signal is a continuation or a change.

Rules in strict priority order:
1. NEVER choose an action outside allowed_actions.
2. BUY means long, HOLD means flat, and SELL means short.
3. If currently long and regime=dynamic_stop_loss_exit, action=hold, conviction=strong.
4. If currently long and regime=take_profit_exit, action=hold, conviction=strong.
5. If regime=bull_momentum_buy_ok, action=buy, conviction=moderate.
6. If regime=bear_oversold_bounce, action=buy, conviction=moderate.
7. If regime=strong_bear_avoid_buy, action=sell, conviction=moderate.
8. If regime=macro_bear_avoid_buy and RSI is not oversold, action=hold or sell.
9. If evidence is mixed, action=hold.

Write a 1-2 sentence reason.
```

B. Pydantic Schemas

```python
class MarketAnalysis(BaseModel):
    date: str
    news_bias: Literal["bullish", "bearish", "mixed"]
    macro_view: Literal["bull_market", "bear_market"]
    volatility: Literal["low", "normal", "high"]
    momentum_signal: Literal["bullish", "bearish", "mixed"]
    summary: str

class RiskAssessment(BaseModel):
    position_state: Literal["flat", "long", "short"]
```

```
    risk_level: Literal["low", "medium", "high"]
    preferred_action: Literal["buy", "sell", "hold"]
    conviction: Literal["strong", "moderate", "weak"]
    risk_note: str

class TradeDecision(BaseModel):
    date: str
    action: Literal["buy", "sell", "hold"]
    conviction: Literal["strong", "moderate", "weak"]
    reason: str
```

References

[1] Z. Xie, R. Elbadry, F. Zhang, G. Georgiev, X. Peng, L. Qian, J. Huang, D. Dimitrov, V. Jani, Y. Dai, J. Geng, Y. Wang, I. Koychev, V. Stoyanov, P. Nakov, The CLEF-2026 FinMMEval Lab: Multilingual and multimodal evaluation of financial AI systems, 2026. URL: https://arxiv.org/abs/2602.10886. doi:10.48550/arXiv.2602.10886. arXiv:2602.10886.

[2] Z. Xie, L. Qian, G. Georgiev, D. Dimitrov, R. Elbadry, F. Zhang, X. Peng, J. Huang, V. Jani, Y. Dai, J. Geng, Y. Chen, Y. Yuan, H. Wu, Y. Wang, I. Koychev, V. Stoyanov, M. Song, Y. Chen, X. Liu, P. Nakov, Overview of the FinMMEval 2026 task 3: Financial decision making, in: CLEF 2026 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org, Jena, Germany, 2026.

[3] CLEF 2026, FinMMEval: Conference and labs of the evaluation forum, https://clef2026.clef-initiative.eu/labs/finmmeval/, 2026. Accessed 2026-07-07.

[4] L. Qian, X. Peng, Y. Wang, V. J. Zhang, H. He, H. Smith, Y. Han, Y. He, H. Li, Y. Cao, Y. Yu, A. Lopez-Lira, P. Lu, J.-Y. Nie, G. Xiong, J. Huang, S. Ananiadou, When agents trade: Live multi-market trading benchmark for LLM agents, 2025. URL: https://arxiv.org/abs/2510.11695. doi:10.48550/arXiv.2510.11695. arXiv:2510.11695.

[5] Qwen Team, Qwen2.5 technical report, 2025. URL: https://arxiv.org/abs/2412.15115. doi:10.48550/arXiv.2412.15115. arXiv:2412.15115.

[6] A. Das, W. Kong, R. Sen, Y. Zhou, A decoder-only foundation model for time-series forecasting, in: Proceedings of the 41st International Conference on Machine Learning, volume 235 of *Proceedings of Machine Learning Research*, PMLR, 2024, pp. 10148–10167. URL: https://proceedings.mlr.press/v235/das24c.html.

[7] T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, QLoRA: Efficient finetuning of quantized LLMs, in: Advances in Neural Information Processing Systems, 2023.

[8] B. T. Willard, R. Louf, Efficient guided generation for large language models, 2023. URL: https://arxiv.org/abs/2307.09702. arXiv:2307.09702.