VibePolitics Architecture: Agent Feedback Synthesis & Revised Plan

Date: February 6, 2026

Executive Summary

Five research agents reviewed the proposed two-layer ML+LLM architecture for VibePolitics. This document synthesizes their feedback and presents a revised, actionable implementation plan.

Key consensus: The two-layer architecture is theoretically sound but risks overengineering. Start minimal, validate rigorously, expand cautiously.

Part 1: Feedback Synthesis

Agents Who Provided Feedback

Agent	Role	Focus Area
Mei □	Integrator	System coherence, over-engineering risks
Priya □	Theorist	Academic validity, blind spots
Kenji □	Methodologist	ML cascade risks, feedback loops

Agent	Role	Focus Area
Arjun □	Skeptic	Trigger fragility, evaluation gaps, practical fixes
Wei □	Empiricist	Regime change risk, silent shifts, academic framing

---

□ Critical Blind Spots (All Agents Agree)

1. ML Layer False Positive/Negative Cascade

Problem: ML anomaly detection is noisy for political data. Standard algorithms (Kats, ruptures) aren't designed for prediction market dynamics.

- A 3% price move might be "anomalous" statistically but meaningless politically if liquidity is low
- If ML misses an anomaly, Layer 2 never activates (false negative cascade)
- Slow-burning narrative shifts won't trigger sharp changepoints

Fixes:

- Domain-specific thresholding (anomaly significance scales with liquidity)
- Multi-signal gating (require 2+ concurrent signals before escalation)
- Add 5% random sample validation even when ML says "all clear"
- Complement anomaly detection with trend detection for gradual drifts

2. Feature Engineering Gap

Problem: Political anomaly detection needs domain-specific features, not just raw price/volume.

Missing features:

- Temporal context (debate week? primary season?)
- Event proximity (major news in last 48h?)
- Cross-market coherence (one market or whole category?)
- Historical analogs (pattern recognition)
- Market microstructure (bid-ask spreads, order book depth)

Fix: Explicit feature engineering document with full input vector specification.

3. Layer 1 → Layer 2 Communication Protocol Undefined

Problem: If ML just says "anomaly detected," LLM agents waste tokens asking clarifying questions.

Fix: Define structured `MLAlert` object:

```
class MLAlert:
    alert_type: str       # "price_anomaly", "volume_surge", et
    market: str           # "Trump approval Q2 2026"
    severity: float       # 0-1
    confidence: float     # 0-1
    trigger_values: dict  # What specifically triggered
    recent_events: list   # Last 72h major news
    related_markets: list # Similar markets' behavior
    historical_analogs: list  # Similar past patterns
    preliminary_hypothesis: str  # ML's best guess
```

4. Threshold Calibration Underspecified

Problem: Low thresholds → constant false positives, high cost. High thresholds → miss subtle signals.

Fix: Use adaptive thresholds calibrated from historical data:

```
def adaptive_threshold(historical_data, desired_precision=0.7):
    threshold_curve = compute_precision_recall_curve(historical
    return threshold_curve.find_precision(desired_precision)
```

5. Missing Feedback Loop

Problem: No mechanism to learn from mistakes. Can't tune ML thresholds or improve agent intuition.

Fix: Implement validation layer:

```
Anomaly → LLM Analysis → Conclusion → [WAIT] → Ground Truth → A
```

6. "Regime Change" Blind Spot — ML Fragility (Wei)

Problem: Traditional ML relies on historical baselines. If political environment undergoes regime change (national crisis, platform ban), ML will either:

- Trigger constantly (False Positives) → Alert Fatigue, budget blown
- Normalize the chaos (False Negatives) → Treat "new normal" as baseline

Fix: Layer 0 / Config layer must periodically re-calibrate ML thresholds based on Layer 2's qualitative findings. Feedback loop from LLM interpretations back to ML parameters.

7. "Silent Shift" Blind Spot — Boiling Frog Scenarios (Wei)

Problem: Important opinion shifts can be low-velocity. A slow, steady 2-week decline in enthusiasm doesn't trigger anomaly detectors (which look for spikes), but a "seasoned pollster" would notice the eerie silence.

Fix: Mandate a "Daily Hunch Audit" where agents perform qualitative check *even if no ML tripwire was hit*. Ensures we don't miss gradual shifts.

8. Trigger Fragility on Noisy Markets (Arjun)

Problem: ML thresholds (e.g., ruptures changepoint) prone to FP/FN on noisy markets like Polymarket (whale pumps distort signal). No adaptive baselines in original design.

Fix: Backtest extensively on 2024 data first. Use Kalman filter to fuse ML signals pre-LLM. Add adaptive baseline that accounts for known noise sources (large trades, market maker activity).

9. Data Staleness / Latency (Arjun)

Problem: Polymarket + news RSS has latency. System is blind to X/Reddit micro-signals until escalation. "24/7" claim doesn't hold if data is stale.

Fix: Add Google Trends and social fear/greed proxies to Layer 1. Consider streaming data sources for real-time monitoring.

10. Hunch Bias / No Uncertainty Quantification (Arjun)

Problem: Pollster agents echo training data. If prompts are skewed (e.g., over-weight Fox News examples), agents will have systematic bias. No entropy scores or uncertainty quantification.

Fix: Add explicit uncertainty quantification (entropy scores, confidence calibration). Rotate prompt examples. Require agents to state counter-arguments.

□ Moderate Blind Spots

6. ML Assumes Stationarity (Politics Is Not Stationary)

- Midterms differ from presidential elections
- Economic cycles change baseline volatility
- Models trained on 2024 will degrade in 2026

Fix: Periodic recalibration, regime-aware detection, rolling thresholds.

7. Alert Frequency Explosion During Volatile Periods

Debates, election nights → EVERYTHING looks anomalous → Layer 2 overwhelmed.

Fix: Alert clustering/deduplication, adaptive throttling.

8. Data Discontinuity Between Layers

ML processes numbers, LLMs need semantic context. Temporal alignment is lost.

Fix: Time-window buffering (6h of contextual data around anomaly), causal alignment.

9. Historical Analogy Selection Bias

LLMs select similar-looking past events, but political dynamics are non-stationary.

Fix: Analogy validity scoring, novelty detection, require considering dissimilar cases.

▯ Design Decisions (Consensus)

Aspect	Original Proposal	Revised Decision
Sentiment analysis	VADER/FinBERT	**REMOVED** — replace with Topic-Specific Volatility Z-Scores (Wei)
Layer 0 (Master Agent)	LLM orchestrates ML	**REPLACED** with lightweight rule engine (Arjun: Prometheus alerts → webhook)
Thresholds	Static	**Adaptive**, calibrated from data + regime-aware recalibration
Validation	Implicit	**Explicit** 5% random sampling + Daily Hunch Audit
Feedback	Unspecified	**Strict separation** between layers + feedback loop for

Aspect	Original Proposal	Revised Decision
		threshold tuning
Agent count	2 (Alpha/Beta)	**3** (add "Historian" for pattern matching)
Data sources	Polymarket + news	Add Google Trends, FearGreed proxy (Arjun)

□ Novel Methodological Suggestions

Wei's "Heterogeneous Sentinels" (Replaces Sentiment): Instead of generic sentiment, track *Relative Volume Variance* of specific keyword clusters:

- "Cost of Living" vs. "Cultural Issues" vs. "Immigration"
- When the *composition* of public conversation shifts, trigger escalation
- This captures narrative shifts without blunt sentiment scores

Wei's Academic Framing: "Computational Grounded Theory"
For publication, frame the architecture as:

- **Layer 1** = *Automated Discovery* phase
- **Layer 2** = *Theoretical Sampling* phase (agents decide which data slices to examine based on emerging hunches)
- **Layer 0** = *Orchestrator* managing the inductive/deductive loop

This solves the "Marketing vs. Research" tension: Layer 1 provides the quantitative "Skeleton," Layer 2 provides the "Muscle" (narrative and reasoning).

Arjun's Algorithm Suggestions:

- **Layer 1 upgrade:** Prophet for volatility forecasting + Isolation Forest for anomalies
- **Escalation:** Kalman filter to fuse ML signals before LLM
- **Advanced (future):** RLHF'd multi-agent (hunches as actions, debate as environment)

- **Graph approach:** GraphSAGE on event graphs (Polymarket → Fox → X cascades)

⚠ Overengineering Warning (Kenji)

The proposed architecture has **17 components**. Each adds complexity, failure points, and maintenance burden.

Simpler alternative:

1. ML detects anomalies
2. Single "senior analyst" LLM reviews top 3 anomalies daily
3. Output: "Here's what mattered yesterday and why"

This gets **80% of the value with 20% of the complexity**. Build that first.

Part 2: Revised Architecture

Simplified Two-Layer Design (Revised with All Agent Feedback)

```
┌──────────────────────────────────────────────────────────────┐
│  CONFIG LAYER (YAML + Lightweight Rule Engine)               │
│  • Thresholds calibrated from historical data                │
│  • Feature selection for ML detectors                        │
│  • Escalation rules (min signals, cooldowns)                 │
│  • Prometheus-style alerts → webhook (Arjun)                 │
│  • Weekly threshold recalibration from Layer 2 (Wei)         │
└────────────────────┬─────────────────────────────────────────┘
                     │
                     ▼
┌──────────────────────────────────────────────────────────────┐
│  LAYER 1: ANOMALY DETECTION (CPU, continuous)                │
│                                                              │
│  DETECTORS:                                                  │
│  • Kats (changepoint detection)                              │
│  • Ruptures (regime detection)                              │
│  • Prophet (volatility forecasting) — Arjun                 │
│  • Isolation Forest (anomalies) — Arjun                     │
│  • Custom: Cross-market divergence                          │
│  • Custom: Trend/drift detection (14-day window)            │
│                                                             │
│  FEATURES (no generic sentiment):                           │
│  • Price dynamics (returns, volatility, autocorrelation)    │
│  • Volume patterns (spikes, trade size distribution)        │
│  • Cross-market (correlation breakdowns)                    │
│  • Microstructure (bid-ask spreads, order book imbalance)   │
│  • Topic-Specific Volatility Z-Scores (Wei)                 │
│  • Google Trends / FearGreed proxy (Arjun)                  │
│                                                             │
│  OUTPUT: Anomaly flags + confidence scores + explainability │
└────────────────────┬────────────────────────────────────────┘
                     │
                     ▼
┌─────────────────────────────────────────────────────────────┐
│  GATEKEEPER (Deterministic + Kalman Filter Fusion)          │
│                                                             │
│  ESCALATION TRIGGERS:                                       │
│  IF (anomaly_confidence > threshold) OR (random() < 0.05)   │
│     → Escalate to Layer 2                                   │
│  ELSE                                                       │
│     → Log & continue                                        │
│                                                             │
│  DAILY HUNCH AUDIT (Wei):                                   │
│  • Once per day, trigger Layer 2 even without ML alert      │
│  • Catch "boiling frog" low-velocity shifts                 │
│                                                             │
│  DEDUPLICATION:                                             │
│  • Cluster related alerts                                   │
│  • Prioritize by severity + confidence                      │
│  • Adaptive throttling during high-noise periods            │
│  • Kalman filter to fuse signals pre-LLM (Arjun)            │
└────────────────────┬────────────────────────────────────────┘
                     │
                     ▼
```

```
┌──────────────────────────────────────────────────────────────┐
│  LAYER 2: LLM INTERPRETATION (GPU, selective)                │
│                                                              │
│  AGENTS:                                                     │
│  • Agent Alpha (Bullish): "What's the signal case?"          │
│  • Agent Beta (Skeptical): "What's the noise explanation?"   │
│  • Agent Gamma (Historian): "What does this rhyme with?"     │
│                                                              │
│  PROCESS:                                                    │
│  • Form hypotheses about what's shifting                     │
│  • Request 2-3 targeted data points (max 5)                  │
│  • Interpret using historical analogies                      │
│  • Multi-agent debate resolves uncertainty                   │
│  • Report entropy/uncertainty scores (Arjun)                 │
│                                                              │
│  OUTPUT: Narrative + confidence + falsifiable predictions    │
│                                                              │
│  FEEDBACK TO CONFIG (Wei):                                   │
│  • Flag regime changes for threshold recalibration           │
│  • Qualitative findings inform next ML cycle                 │
└─────────────────────┬────────────────────────────────────────┘
                      │
                      ▼
┌──────────────────────────────────────────────────────────────┐
│  VALIDATION LAYER (Post-hoc, for learning)                   │
│                                                              │
│  • Compare Layer 1 + Layer 2 vs. polling shifts (7-day lag)  │
│  • Log false negatives (did we miss something?)              │
│  • Adjust thresholds monthly based on precision/recall       │
│  • Track regime shifts for model recalibration (Wei)         │
└──────────────────────────────────────────────────────────────┘
```

Academic Framing: "Computational Grounded Theory" (Wei)

For publication, frame this architecture as:

- **Layer 1** = *Automated Discovery* phase (quantitative "skeleton")

- **Layer 2** = *Theoretical Sampling* phase (agents decide which data to examine)
- **Config Layer** = *Orchestrator* managing the inductive/deductive loop

This positions VibePolitics as bridging quantitative rigor and qualitative depth — addressing the "Marketing vs. Research" tension in political forecasting.

Part 3: Implementation Plan

Phase 1: Foundation (Weeks 1-2)

Goal: Build feedback loop first, before full pipeline.

- [ ] Collect historical Polymarket data with known outcomes (2024 election)
- [ ] Implement basic ML detection (Kats + ruptures) retrospectively
- [ ] Measure false positive/negative rates on historical data
- [ ] Define explicit feature set for Layer 1
- [ ] Create `config/signal_thresholds.yaml`

Deliverables:

- Historical data archive
- Baseline ML detector script
- Feature engineering document
- Initial threshold calibration

Phase 2: MVP Pipeline (Weeks 3-4)

Goal: Single LLM agent reviews top anomalies.

- [ ] Implement single "senior analyst" LLM for all escalations
- [ ] Define structured `MLAlert` escalation protocol
- [ ] Build 5% random sampling validation
- [ ] Measure cost/accuracy tradeoff
- [ ] Implement basic logging and metrics

Deliverables:

- Working single-agent pipeline
- Cost tracking dashboard
- Accuracy metrics (precision/recall at different thresholds)

Phase 3: Multi-Agent (Weeks 5-6)

Goal: Add agent diversity and debate.

- [ ] Add Agent Beta (Skeptical)
- [ ] Add Agent Gamma (Historian)
- [ ] Implement multi-agent debate protocol
- [ ] Add domain-specific features (microstructure, cross-market)
- [ ] Implement alert clustering/deduplication

Deliverables:

- Three-agent system
- Debate resolution protocol
- Enhanced feature set

Phase 4: Validation & Paper (Weeks 7-8)

Goal: Academic defensibility.

- [ ] Run ablation study (Layer 1 alone, Layer 2 alone, combined)
- [ ] Collect complementarity proof (cases where each layer adds value)
- [ ] Cost-benefit analysis with real numbers
- [ ] Write methodology section
- [ ] Prepare for peer review

Deliverables:

- Ablation study results
- Draft methodology paper section
- Cost analysis

Part 4: Explicit Specifications

4.1 Feature Set for Layer 1

Category	Feature	Computation	Source
Price	Returns	`(p[t] - p[t-1]) / p[t-1]`	Polymarket
Price	Volatility	Rolling std(returns, window=7)	Polymarket
Price	Autocorrelation	`corr(returns[t], returns[t-1])`	Polymarket
Volume	Volume ratio	`volume / rolling_mean(volume, 30)`	Polymarket
Volume	Trade size distribution	Median trade size vs. historical	Polymarket
Cross-market	Correlation breakdown	Rolling correlation deviation	Polymarket + Kalshi
Microstructure	Bid-ask spread	`(ask - bid) / mid`	Polymarket
Microstructure	Order book imbalance	`(bid_depth - ask_depth) / total`	Polymarket
Context	Event proximity	Days to next major event	Calendar
Context	Calendar flag	Is this debate week / election week?	Calendar
Topic (Wei)	Keyword cluster variance	Z-score of topic volume shifts	News RSS
Topic (Wei)	Narrative composition	Relative share: economy vs. culture vs. immigration	News RSS

Category	Feature	Computation	Source
External (Arjun)	Google Trends momentum	7-day trend for key terms	Google Trends
External (Arjun)	Fear/Greed proxy	VIX-style indicator for political markets	Derived

4.2 Escalation Rules

```
# config/signal_thresholds.yaml
anomaly_detection:
  kats:
    method: "cusum"
    threshold: 2.5  # z-score
    lookback_window: "7d"
  ruptures:
    model: "l2"
    penalty: 10
  trend:
    drift_threshold: 0.05  # 5% cumulative drift
    window: "14d"

escalation_rules:
  min_signals: 2        # Require 2+ detectors to fire
  cooldown_period: "1h" # Don't re-escalate same market
  max_daily_escalations: 10  # Cost control
  random_sample_rate: 0.05   # 5% validation sampling

alert_priority:
  high_confidence_threshold: 0.8
  time_sensitive_boost: 1.5  # Closer to election
  cross_market_boost: 1.3    # Multiple markets moving
```

4.3 LLM Agent Prompts (Draft)

Agent Alpha (Bullish):

```
You are analyzing a prediction market anomaly. Your role is to

Anomaly: {alert_data}
Recent context: {context}

1. What is the most likely explanation for this signal?
2. What 2-3 data points would confirm your hypothesis?
3. What historical event does this most resemble?
4. Confidence (0-1) and reasoning.
```

Agent Beta (Skeptical):

```
You are a skeptic analyzing a prediction market anomaly. Your r

Anomaly: {alert_data}
Recent context: {context}

1. Why might this signal be misleading?
2. What alternative explanations exist (liquidity, manipulation
3. What would convince you this is NOT noise?
4. Confidence (0-1) that this is noise.
```

Agent Gamma (Historian):

```
You are a political historian analyzing a prediction market ano

Anomaly: {alert_data}
Recent context: {context}

1. What historical events does this pattern resemble?
2. What happened after those events?
3. Why might this time be DIFFERENT from history?
4. What's the track record of historical analogy for this type
```

Part 5: Academic Defensibility

Required Justifications for Reviewers

1. **Complementarity proof:** Document cases where:
    - ML detected but couldn't explain (need LLM)
    - LLM explained but couldn't have detected alone (need ML)
2. **Cost-benefit analysis:**
    - "Layer 2 costs $X/day but catches Y% more early signals"
    - Compare to all-LLM baseline
3. **Ablation study:**
    - Layer 1 alone precision/recall
    - Layer 2 alone precision/recall
    - Combined precision/recall
4. **Transparent disclosure:**
    - Publish exact ML thresholds
    - Publish exact LLM prompts
    - Report both false positives AND false negatives

Academic Contribution Claim

> "We demonstrate a cascading inference architecture where computational efficiency (ML filtering) enables resource-intensive qualitative analysis (LLM interpretation) without sacrificing real-time detection capability. Unlike pure quantitative approaches, our system produces narrative explanations; unlike pure LLM approaches, it maintains cost efficiency through selective activation."

Part 6: Open Questions (For Team Decision)

1. **Escalation rule:** 1-of-3 detectors? 2-of-3? Unanimous?

    - *Recommendation:* Start with 2-of-3 to balance sensitivity and specificity.

2. **Tiered escalation:** Should low-confidence alerts get only 1 agent, high-confidence get all 3?

    - *Recommendation:* Yes, saves tokens without losing coverage.

3. **Historian agent data source:** What's the historical analogy database?

    - *Recommendation:* Curated list of 50-100 key political events with market reactions.

4. **Sentiment replacement:** Without VADER/FinBERT, how do we detect narrative shifts?

    - *Wei's recommendation:* Topic-Specific Volatility Z-Scores on keyword clusters.
    - *Arjun's recommendation:* Zero-shot Llama-3.1 or DistilBERT fine-tuned on political corpora.
    - *Decision needed:* Choose between rule-based (Wei) vs. model-based (Arjun) approach.

5. **Ground truth for validation:** What counts as "correct" prediction?

    - *Recommendation:* Comparison to RealClearPolitics polling averages at 7-day lag.

6. **Layer 0 implementation:** Lightweight rule engine vs. periodic LLM recalibration?

    - *Arjun:* Simple rule engine (Prometheus alerts → webhook, Python cron).
    - *Wei:* Layer 0 should periodically recalibrate based on Layer 2 findings.
    - *Compromise:* Rule engine for routing + periodic (weekly?) LLM-based threshold review.

7. **Daily Hunch Audit:** How often should agents check even without ML trigger?

    - *Wei's recommendation:* Daily, to catch "boiling frog" scenarios.
    - *Cost consideration:* ~1-2 LLM calls/day = ~$1-3/day extra.

8. **Real-time data sources:** Which social platforms to add?

    - *Arjun:* Blind to X/Reddit until escalation is a problem.
    - *Options:* Google Trends (free), Twitter/X API (expensive), Reddit API (rate-limited).

Appendix: Key Agent Quotes

Mei (Integrator):

> "The big risk is over-engineering the ML layer to be too clever and brittle. Keep it simple: Anomaly detection (Kats, ruptures) → works well. The strength of this architecture is simple ML + smart LLMs, not the other way around."

Priya (Theorist):

> "The architecture is sound. The blind spots are manageable. The key is rigorous validation of the handoff between layers — that's where systems like this usually fail."

Kenji (Methodologist):

> "The two-layer architecture is theoretically sound but practically fragile. Start minimal, validate rigorously, expand cautiously. This gets you 80% of the value with 20% of the complexity."

Wei (Empiricist):

> "This Two-Layer Architecture is a massive leap forward in both efficiency and academic defensibility. It mirrors how high-frequency trading desks and intelligence agencies actually operate: automated 'tripwires' scaling up to human-level (or agent-level) analysis."

> "Layer 0 should mandate a 'Daily Hunch Audit' where agents perform a qualitative check even if no ML tripwire was hit, ensuring we don't miss the 'boiling frog' scenarios."

Arjun (Skeptic):

> "Solid hybrid—ML triage slashes costs/latency, LLMs add causal narrative. Defensible academically."

> "Layer 0 bloat: Master agent = extra LLM hops; use simple rule engine (e.g., if-then in Python cron)."

> "Ship or it's vapor."

Document updated: February 6, 2026 All 5 agents' feedback incorporated Location: `projects/vibepolitics/research/ARCHITECTURE_SYNTHESIS_AND_PLAN.md` *Next step: Review with team, then begin Phase 1 implementation.*