

# MASX AI Technical Paper, Autonomous Strategic Forecasting Architecture

**Classification.** Investor Technical Reference **Version.** 1.0, April 2026 **Source.** Verified against MASX AI production codebase **Live Platform.** <https://forecast.masxai.com>

## 1. Executive Summary

MASX AI is a **fully autonomous** strategic forecasting engine that converts raw global information into probabilistic, research grade forecasts, from article ingestion through multi channel distribution, without a single human in the loop. The system operates two tightly integrated pipelines. A 14 stage data pipeline that ingests thousands of articles daily from 200+ sources, applies 4 gate corroboration with syndication detection and lifecycle tracking, and a 7 stage forecasting pipeline that produces doctrine grounded, tool constrained probabilistic forecasts with self calibrating feedback.

The system is core innovations are the following.

**14 Stage Autonomous Data Pipeline.** The data pipeline processes raw global media through 14 stages with 4 progressive quality gates (junk filter to corroboration with syndication detection to multi scale substance validation to 7 component composite scoring), 3 tier cross domain detection (static rules to embedding cosine to zero shot DeBERTa), cross day lifecycle tracking via centroid embedding matching, and a dual provider LLM significance judge with consensus re judging for borderline events.

**Council of Doctrines.** A multi agent deliberation architecture where 35 strategic doctrine corpora (spanning classical statecraft, geopolitics, hybrid warfare, economics, and environmental systems) are dynamically selected, retrieved via RAG, and synthesized through Bayesian log odds aggregation to produce calibrated probability estimates across three time horizons (1 month, 3 month, 6 month) for questions generated at three resolution horizon bands (14 day, 30 day, 90 day).

**Self Calibrating Feedback Loop.** The system is most strategically significant capability. Resolved forecasts are scored via Brier decomposition, and the resulting per doctrine performance weights feed back through three independent channels. (a) mathematical weight adjustment in the Bayesian aggregation function, (b) domain specific calibration guidance in the forecaster prompt, and (c) system wide diagnostic alerts. This creates a compounding advantage. Every resolved forecast makes subsequent forecasts more accurate.

**16 Check Dual Quality Gate.** An 8 point grounding validator that detects and auto heals common LLM failure modes in forecasts (extreme probabilities, base rate drift, flat projections, missing tool references), combined with a separate 8 point advisor validator that catches and auto corrects recommendation failures (entity diversity, urgency probability coherence, doctrine grounding, duplicate detection). Both gates operate deterministically.

**Quantified Decision Support.** Beyond probability estimates, the advisor agent computes intervention windows (you have roughly X days before this becomes likely) and cost of inaction risk multipliers (waiting 30 days increases risk by 2.3x), unique outputs not found in any competing platform.

**Autonomous Content Production.** The pipeline does not just forecast. It autonomously produces and distributes a daily newsletter (7 data charts, deterministic), LinkedIn posts, weekly flagship analyses, and fully autonomous daily and weekly podcast episodes (script to TTS to MP3 to cloud storage), making it a complete intelligence media platform at zero marginal content cost.

## 2. System Architecture

MASX AI follows a **Clean Architecture and Hexagonal Architecture** pattern with strict separation of concerns.

Layer	Location	Responsibility
<b>Domain Logic</b>	core/domain/	Pure calculation functions, Pydantic models, validation rules, exception hierarchy. Zero I/O dependencies.
<b>Infrastructure</b>	core/infra/	LLM agents (pydantic ai), RAG (LlamaIndex), database (Supabase), email (Resend). Implements Protocol ports defined in core/domain/ports/.
<b>Orchestration</b>	pipeline/	Pipeline runners for forecasting, publishing, podcasting, resolution, and calibration.

**Key architectural constraint.** The domain layer imports only abstractions (Protocol interfaces). All I/O is injected at runtime through the PipelineDeps dataclass, enabling full unit testability of business logic without external dependencies.

```
PipelineDeps
  hotspot_reader      HotspotReader      (Protocol)
  doctrine_retriever  DoctrineRetriever  (Protocol)
  forecast_store      ForecastStore      (Protocol)
  calibration_store   CalibrationStore   (Protocol)
  doctrine_perf_store DoctrinePerformanceStore (Protocol)
```

## 3. The 14 Stage Data Pipeline

Before any forecasting occurs, raw global information must be transformed into structured, corroborated geopolitical signals. The thing is, most of what gets published is noise, duplicates, syndicated rewrites, or irrelevant. The **data pipeline** handles this. It is a 14 stage, 4 gate autonomous pipeline that ingests thousands of articles daily from 200+ sources across 17 geographic regions, and promotes only corroborated, multi source events into the hotspots that drive the forecasting engine.

### 3.1 Pipeline Architecture Overview

The pipeline processes data in strict sequential order. Each stage operates within a transactional isolation boundary (PostgreSQL BEGIN/COMMIT), ensuring pipeline integrity and enabling checkpoint resume after failures.

Stage	Name	Function	Gate
s01	RSS Ingest	Multi layer feed fetching with fallback chain	
s02	API Ingest	Structured API ingestion (ACLED, GDELT, NASA FIRMS, etc.)	
s03	Fetch and Extract	7 tier HTTP escalation with SSRF defense	
s04	Enrich	ONNX NER, geo resolution, language detection, cross domain	

Stage	Name	Function	Gate
s05	Deduplicate	tagging Three layer deduplication (URL hash to content hash to SimHash)	
s06	Gate 1	Junk filter, article length, topic exclusion, language validation	<b>Gate 1</b>
s07	Pool Assign	Geographic pool assignment (global + 17 regional + per country)	
s08	Embed	ONNX embedding (384 dim INT8 quantized paraphrase-multilingual-MiniLM-L12-v2)	
s09	Subcluster	Cosine subclustering with corroboration rules	<b>Gate 2</b>
s10	Cluster	Multi scale event clustering (global, regional, country)	<b>Gate 3</b>
s11	Hotspot	Composite scoring + urgency assignment + hotspot promotion	<b>Gate 4</b>
s12	Lifecycle	Cross day state machine + 122 node domain classification	
s12b	Judge	Dual provider LLM significance filter with consensus re judging	
s13	Summarize	LLM summarization + slug generation + domain and country cross check	
s14	Output	Baselines, coverage confidence, snapshots, dynamic trust scoring	

**Embedding after Gate 1 optimization.** Embeddings are computed only for articles that survive Gate 1, saving roughly 30% of compute by never embedding killed articles.

### 3.2 Feed Ingestion and Multi Layer Fallback (s01 through s02)

**RSS Fallback Chain (s01).** Each source has a prioritized fallback chain.

Layer	Source	Cost
Layer 1	Direct RSS feed URL with conditional GET (ETag and If Modified Since)	Free
Layer 2	Google News proxy with region aware edition selection (hl, gl, ceid per source language + geographic region)	Free
Layer 3	RSS Bridge (future)	Free
Layer 4	GDELT DOC backfill (metadata only, no content)	Free

**Circuit breaking.** 5 consecutive failures and the source is skipped until next\_poll\_at. A 24 hour freshness cutoff drops entries older than the cutoff, preventing stale article ingestion. Auto derived Google

News fallback works like this. If a source has a direct feed but no explicit Google News query, the system automatically constructs a `site:domain.com when:2d` fallback.

**Structured API Ingestion (s02).** Parallel ingestion from conflict databases (ACLED), event monitors (GDELT), disaster and natural hazard feeds (USGS, NASA EONET, NASA FIRMS, GDACS), censorship monitoring (OONI), maritime warnings (NGA MSI), and network outage detection (Cloudflare Radar). Additional sources (FRED, EIA, OTX, URLhaus) have URL constants and registry entries prepared for future integration. All API keys are managed via `pydantic-settings` with graceful degradation when keys are missing.

### 3.3 Seven Tier Fetch Escalation and SSRF Defense (s03)

Article HTML is downloaded through a **7 tier escalation chain** with a 120 second hard budget per URL.

Tier	Method	When Used
1	httpx (direct, random UA from 10 agent pool)	Default for all URLs
1.5	curl_cffi (TLS fingerprint impersonation)	On 403 or 503 or Cloudflare challenge
2	httpx + rotating proxy pool	After Tier 1.5 failure
2.5	curl_cffi + proxy	TLS impersonation + different IP
3	Headless browser (Patchright on Windows, Camoufox on Linux)	JS rendered pages
3.5	Browser + proxy	Full DOM execution from different IP
4	Google Cache	Last resort cached version

**SSRF defence in depth.** (1) Scheme whitelist (http and https only), (2) pre flight DNS resolution with every returned IP checked against `is_private`, `is_reserved`, `is_loopback`, including IPv4 mapped IPv6 address unwrapping, (3) 10 MB response size hard cap.

**Auto learning domain mode tracker.** A per run in memory tracker records fetch results. After 3 consecutive failures at a tier, the domain is auto promoted to the next tier for subsequent fetches. Static overrides for known domains (for example, `bloomberg.com` goes to EXPERT, `reuters.com` goes to INTERMEDIATE) skip doomed lower tiers entirely.

**Cloudflare fast detection.** The first 2 KB of every response body is scanned for Cloudflare challenge markers (`cf-browser-verification`, `_cf_chl_opt`). Detection triggers immediate escalation without waiting for the full timeout cycle.

### 3.4 NLP Enrichment and Cross Domain Detection (s04)

**ONNX NER Pipeline.** Named entity recognition runs entirely on CPU via ONNX Runtime, no PyTorch dependency, no GPU required. The pipeline performs batch inference (16 texts per batch) using a distilbert NER model with custom BIO sequence decoding that handles subword token merging, B I sequence validation, and entity span reconstruction. An alternative optimum path is available via HuggingFace pipeline for development convenience.

**3 Tier Cross Domain Detection.** Every article is checked for cross domain relevance through a hybrid detection system, all CPU only, \$0 cost.

Tier	Method	Latency	How It Works
Tier 1	Static keyword rules	roughly 0 ms	Trigger keywords from <code>cross_domain_rules</code> table, with bidirectional matching and keyword hit ratio confidence boosting
Tier 2	Embedding cosine	roughly 2 ms	Article embedding compared against 122 domain node embeddings. Auto add above 0.65, borderline (0.45 to 0.65) sent to Tier 3
Tier 3	Zero shot classifier	roughly 50 ms	DeBERTa v3 xsmall confirmation for borderline Tier 2 candidates. Multi label classification with 0.35 threshold

**Geo resolution.** Extracted location entities are resolved to standardized country codes and coordinates using a capitals database. Language detection classifies article language for downstream pool assignment and translation.

### 3.5 Three Layer Deduplication (s05)

Layer	Method	What It Catches
Cross run URL hash	Exact URL match against known articles table	Previously ingested articles from any run
In run content hash	SHA 256 of title + description	Identical articles published under different URLs
SimHash near duplicate	64 bit SimHash fingerprint with configurable Jaccard threshold (0.60)	Paraphrased syndication copies, same story rewritten slightly by different outlets

### 3.6 Four Gate Corroboration Architecture (s06 through s11)

This is the pipeline is core quality enforcement mechanism. Each gate applies progressively stricter validation.

**Gate 1, Junk Filter (s06).** Deterministic rules. Minimum content length, excluded topic keywords (sports, entertainment, lifestyle), language whitelist. Articles failing Gate 1 are killed before embedding, saving roughly 30% of compute.

**Gate 2, Corroboration Rules (s09).** After cosine based subclustering, the system enforces minimum articles per subcluster at `gate2_min_articles = 3`, minimum independent sources at `gate2_min_sources = 2`. **Syndication detection** via MinHash Jaccard of 0.60 or higher collapses syndicated copies into a single effective source, preventing wire service reprints from inflating corroboration counts. A **bias risk guard** flags subclusters where 80% or more of articles come from a single source domain.

**Gate 3, Multi Scale Substance Validation (s10).** Clusters are formed at three scales simultaneously. **Global pool** covers all articles. **Regional pools** cover 17 geographic regions (Middle East, South Asia, East Asia, and so on). **Country pools** handle per country clusters for countries with sufficient article volume.

Each cluster must meet a minimum substance threshold (`gate3_min_substance = 3`). This multi scale approach ensures that events significant at a regional level (a localized conflict, for example) are not drowned out by globally dominant stories.

**Gate 4, Composite Scoring and Hotspot Promotion (s11).** The 7 component composite score determines which clusters become hotspots.

Component	Weight	Calculation
Volume	25%	$\log_2(\text{article\_count} + 1) / \log_2(\text{pool\_P95} + 1)$ , capped at 1.0
Recency	20%	$\exp(-0.693 \times \text{age\_hours} / 12)$ , 12 hour half life
Diversity	20%	$\log_2(\text{unique\_domains} + 1) / \log_2(20)$ , capped at 1.0
Topic	15%	Pillar based weight lookup from TOPIC_WEIGHTS
Velocity	10%	$\text{recent\_24h\_count} / \max(\text{article\_count} / \text{days\_active}, 1.0)$ , lifecycle aware freshness
T1 Convergence	+0 to 15%	$\min(\text{t1\_source\_count} \times 0.05, 0.15)$ , bonus for T1 wire service coverage
Anomaly Bonus	+0 to 20%	$\min(\max(\text{z\_score}, 0) / 4.0, 0.20)$ , positive Z score anomaly

Score floor is `gate4_score_floor = 0.35`. Urgency is assigned based on composite score. Critical (0.70 or higher), elevated (0.50 or higher), watch (0.35 or higher).

### 3.7 Cross Day Lifecycle Management (s12)

Hotspots persist across pipeline runs through a **cross day matching and lifecycle state machine**.

**Cross day matching.** Each newly promoted hotspot is centroid embedding is compared against all existing active hotspots. Match criteria require cosine similarity of 0.80 or higher AND at least one shared country (preventing same topic events in different countries from merging). Matched hotspots are merged. Articles are migrated, velocity is re scored with lifecycle context (`rescore_velocity`), and the born duplicate is expired.

#### State transitions.

```

born + matched      active
active + matched    active
fading + matched    revived
expired + matched   revived
any + unmatched (over 48h)  fading
any + unmatched (over 72h)  expired

```

**Trend computation.** Based on most recent prior snapshot score. Surging means the score increased by 30% or more. Escalating means 10% or more increase. Stable means less than 10% change in either direction. Cooling means 10% or more decrease. Collapsing means 30% or more decrease.

**122 Node Domain Classification.** Each new hotspot is classified into a domain tree by comparing its centroid against 122 domain node embeddings. The best match is validated against `classification_keywords` and `exclusion_keywords` stored per node. A **pillar sanity check** overrides the

cosine best domain when the source articles topic contradicts it (for example, Ukraine battlefield data incorrectly classified as technology).

### 3.8 Dual Provider LLM Significance Judge (s12b)

A post lifecycle validation stage that rejects non significant hotspots and corrects domain misclassifications.

**Round robin distribution.** Hotspots are distributed across two LLM providers (Gemini + OpenAI) for 2x throughput at the same cost. Even indexed hotspots go to Client A (Gemini primary), odd indexed to Client B (OpenAI primary).

**Borderline consensus re judging.** Hotspots scoring exactly at the significance threshold (score = 3) are re judged by the OTHER provider. If the second judge scores below threshold, the hotspot is rejected. This prevents false positives from a single model is bias.

**Domain reclassification guard.** The judge can suggest a corrected domain, but the suggestion is validated against the set of 122 valid domain node IDs in the database, preventing hallucinated domain assignments from corrupting the classification.

**Score distribution logging.** The histogram of judge scores is logged for calibration monitoring, enabling detection of threshold drift over time.

### 3.9 Dynamic Source Trust and Coverage Confidence (s14)

**Dynamic Trust Scoring.** Source reliability is computed empirically from corroboration data.

`trust = lerp(trust_static, corroboration_rate, alpha=TRUST_ALPHA)`  
clamped to `[trust_static x FLOOR, min(trust_static x CEILING, 1.0)]`

Where `corroboration_rate` = fraction of the source is articles that appeared in corroborated clusters. A minimum article threshold prevents unstable scores from low volume sources. The full computation history is persisted to `source_trust_history` for audit trail.

**Coverage Confidence and Anomaly Detection.** For each of 17 geographic regions, the pipeline computes a daily confidence ratio.

Anomaly Type	Detection Rule	Significance
ensorship_suspected	RSS articles less than 30% of baseline, but API events still detected	Potential government level information suppression
blackout	Both RSS and API less than 30% of baseline	Infrastructure failure or total media blackout
surge	Articles more than 3x baseline	Breaking event or information cascade

Low confidence regions (ratio less than 0.50) are flagged and surfaced in the processing run metadata.

## 4. The Seven Stage Forecasting Pipeline

### 4.1 Stage 1, Hotspot Ingestion and Filtering

The pipeline ingests geopolitical hotspots from a Supabase database (populated by the data pipeline described in Section 3). Each hotspot is a `HotspotContext` Pydantic model containing a label and summary

(human readable event description), domain classification (primary DomainPillar covering governance, conflict, economy, technology, society, environment, and infrastructure, plus secondary domains), signal metadata (z\_score, velocity as freshness ratio, t1\_convergence, anomaly\_bonus), score breakdown (composite score from weighted components), geographic context (countries, region, latitude and longitude), and linked source articles with headlines and snippets.

**Worthiness gate.** Before entering the pipeline, each hotspot passes six filters.

Gate	Threshold	Source
Composite score	0.3 or higher	HOTSPOT_MIN_COMPOSITE_SCORE
Article count	3 or more	HOTSPOT_MIN_ARTICLE_COUNT
Source diversity	2 or more sources	HOTSPOT_MIN_SOURCE_COUNT
Summary present	Non empty	N/A
Named entities	At least 1	N/A
Geographic data	At least 1 country	N/A

Any hotspot failing a gate is logged with a SkipReason enum and excluded from forecasting. This prevents the system from wasting LLM resources on low quality signal.

**Concurrency.** Up to 6 hotspots are processed concurrently (PIPELINE\_HOTSPOT\_CONCURRENCY = 6), bounded by an asyncio.Semaphore.

## 4.2 Stage 2, Doctrine Router

For each hotspot, the system determines which strategic doctrines are relevant. This is a two phase process.

**Phase 1, Matrix Lookup.** The doctrine\_registry.py contains 35 doctrine entries, each with an affinities map linking it to domain pillars at HIGH or MEDIUM affinity. The matrix\_lookup() function computes candidate doctrines by matching the hotspot is primary and secondary domain pillars against the affinity matrix, scoring each match with a priority function.

```
score = (2 if HIGH else 1) x 10 + (1 if primary_pillar else 0)
```

**Phase 2, LLM Router.** The candidate list is passed to a pydantic ai router\_agent, which selects the top 5 most relevant doctrines (ROUTER\_TOP\_K = 5) and assigns each a relevance\_score (0.0 to 1.0) with reasoning.

**Relevance filtering.** Doctrines scoring below a cutoff (ROUTER\_RELEVANCE\_CUTOFF = 0.5) are dropped, unless this would leave fewer than 2 doctrines (MIN\_DOCTRINES\_FOR\_LOG\_ODDS), in which case the top 2 by score are retained. This guarantees the downstream Bayesian aggregation always has sufficient input dimensions.

## 4.3 Stage 3, Council of Doctrines (RAG Augmented Analysis)

Each selected doctrine runs through a parallel analysis pipeline.

**Pre fetch RAG context.** For each doctrine, the system queries a LlamaIndex VectorStoreIndex built from that doctrine is source PDFs. Each index uses all-MiniLM-L6-v2 embeddings (384 dimensional, local inference), ms-marco-MiniLM-L-6-v2 reranking (local cross encoder), top 20 candidates reranked to top 5 (RETRIEVAL\_TOP\_K = 20, RERANK\_TOP\_N = 5), a maximum of 6,000 characters per doctrine (CONTEXT\_MAX\_CHARS\_PER\_DOCTRINE), and Jaccard similarity deduplication at threshold 0.7 to remove near duplicate chunks.

**Council Agent.** A pydantic ai agent analyzes the hotspot through each doctrine is unique strategic lens. The output is a `DoctrineAnalysis` containing `doctrine_id` (which doctrine produced this analysis), `lens` (the specific strategic lens applied), `analysis` (the substantive analysis text), `confidence` (0.0 to 1.0 confidence in this doctrine is applicability), `direction` (+1 escalatory or supporting, negative 1 de escalatory or opposing, or 0 neutral), and `key_concepts` (specific doctrine concepts invoked).

**Dynamic RAG Tool (`retrieve_more`).** The council agent has an autonomous tool that allows it to **request additional doctrine passages during analysis** when the pre fetched context is insufficient. The agent specifies a `doctrine_id` and a targeted query, and the system retrieves additional chunks from that doctrine is vector index (`COUNCIL_FOLLOW_UP_TOP_K`). This is a significant agentic capability. The LLM is not limited to a static context window but can actively seek more evidence, much like a human analyst pulling additional reference material.

**Output Validation.** Council output is validated against the expected doctrine IDs. Unknown doctrine IDs are dropped, missing doctrines are logged, and zero valid analyses triggers a hard `CouncilError`, ensuring the pipeline never proceeds with fabricated analysis.

#### 4.4 Stage 4, Question Generation

For each hotspot, the system generates forecasting questions across three resolution horizon bands.

Band	Days Forward	Purpose
Short term	14 days	Imminent developments
Medium term	30 days	Policy and institutional responses
Long term	90 days	Structural trajectory

Within each forecast, the system also produces **three time projections** at 1 month (30 days), 3 month (90 days), and 6 month (180 days) horizons, representing how the probability is expected to evolve over time, distinct from the question is resolution date.

Each question is a `ForecastQuestion` Pydantic model containing a precisely worded binary question, a `base_rate` (0.0 to 1.0) derived from historical reference classes, `resolution_criteria` with falsifiable conditions for judging the outcome, and skip logic where the generator can skip a band with a reason if no meaningful question can be formed.

**Deduplication.** Before generating questions, the pipeline checks existing open forecasts for the same hotspot. Horizon bands that already have a forecast within plus or minus 3 days are skipped to prevent duplicate coverage.

**Question Quality Validator (Self Improving).** Each generated question passes through a programmatic quality gate before the forecaster ever sees it. This validator checks resolution criteria length bounds (between 30 and 2,000 characters), vague term detection against a blocklist of **36 banned phrases** that cause unresolvable forecasts (tensions increase, significant impact, evolving situation, credible reports), and an optional named source requirement where criteria can require reference to specific authoritative sources.

The blocklist is organized into **three tiers that demonstrate the self improving loop in action.**

Tier	Source	Example Terms
<b>Original</b>	Expert curation at system design	tensions increase, situation worsens, major development
<b>Phase 4 additions</b>	Patterns identified by the <code>criteria_analyzer</code> from	further developments, widespread speculation, evolving situation

Tier	Source	Example Terms
	resolver ambiguity data	
<b>L1 feedback loop</b>	Phrases that <i>passed</i> validation but still caused AMBIGUOUS resolutions in production	according to reports, if confirmed by officials, credible reports, significant military activity

This is **concrete proof** that the self calibrating feedback loop operates not just on forecast probabilities but on the quality of the questions themselves. The system is input quality improves autonomously based on real resolution outcomes.

#### 4.5 Stage 5, Forecaster Agent (Tool Augmented Bayesian Synthesis)

This is the computational core of the pipeline. For each valid question, a pydantic ai forecaster agent synthesizes a probability estimate using 7 mathematical tools.

##### 4.5.1 Pre computed Anchors

Before the LLM runs, the system pre computes two deterministic anchors.

**Signal Anchor.** Converts hotspot signal metadata into a calibrated likelihood ratio.

```
signal_boost(z_score, velocity, velocity_percentile, t1_convergence, anomaly_bonus)
  composite_lr (capped at 6.0)
  signal_boosted_posterior = bayesian_update(base_rate, composite_lr)
```

**Doctrine Anchor.** Aggregates council doctrine assessments via log odds addition.

```
multi_doctrine_log_odds(base_rate, [DoctrineShift(doctrine_id, direction, confidence)])
  doctrine_posterior
```

When historical performance weights are available (from the calibration feedback loop), each doctrine is contribution is scaled by its reliability weight.

These anchors are injected into the LLM prompt as hard constraints. The forecaster must justify any departure from them.

##### 4.5.2 Available Tools

Tool	Mathematical Basis	Input and Output
calc_bayesian_update	Bayes theorem, posterior_odds = prior_odds x LR	base_rate, LR to posterior
calc_laplace_base_rate	Laplace or Jeffreys smoothing, $(s+\alpha)/(n+2\alpha)$	successes, trials to smoothed_rate
calc_temporal_projection	CAMEO events to Goldstein escalation to Weibull decay	events, total_p, horizons to per horizon probabilities
calc_time_horizon_decay	Exponential or Weibull survival functions	total_p, model, shape_k to per horizon probabilities
calc_confidence_interval	Wilson or Jeffreys binomial CI	successes, trials to [lower, upper]
calc_evidence_weight	Calibrated LR from signal type and relevance	signal_type, relevance to LR, weight_of_evidence, diagnostic_label

Tool	Mathematical Basis	Input and Output
calc_fermi_decompose	Conjunctive or disjunctive decomposition, $P(A \text{ and } B) = P(A) \times P(B)$	sub questions, chain_type to composite_probability, sensitivity

**Goldstein escalation scoring.** The temporal\_projection tool uses a CAMEO inspired event taxonomy (20 categories from Verbal Cooperation to Mass Violence) with Goldstein scale weights ranging from +7.4 (Provide Aid) to negative 10.0 (Mass Violence). Events are weighted by recency (exponential decay,  $\alpha = 0.05$ ). The resulting normalized escalation index (negative 1 to +1) maps to a Weibull shape parameter (k) with a neutral dead band. Normalized score less than negative 0.3 (escalating) yields  $k = 1.5 + |\text{score}| \times 1.5$ , up to max 3.0 (increasing hazard). Score greater than +0.3 (de escalating) yields  $k = 0.5 + (1 \text{ minus score}) \times 0.3$ , down to min 0.3 (decreasing hazard). Neutral band [negative 0.3, +0.3] yields  $k = 1.0$  (constant hazard, exponential).

**Graceful degradation.** Every tool returns structured fallback guidance when a calculation fails. The response includes a specific manual estimation heuristic and flags the result as ungrounded, which propagates to the grounding validator.

#### 4.5.3 LLM Output Schema

The forecaster produces an LLMForecast Pydantic model (strictly validated). probability ranges from 0.01 to 0.99 (hard clamped to prevent overconfidence). confidence\_interval is [lower, upper] validated to satisfy  $lo \text{ less than probability less than } hi$ . time\_projections requires exactly 3 projections at 1 month, 3 month, and 6 month horizons (30, 90, and 180 days). doctrine\_synthesis is text explaining how doctrines were synthesized. contradictions lists explicitly identified inter doctrine conflicts. calibration\_notes must reference tool outputs for grounding verification. And key\_drivers, disconfirming\_evidence, update\_triggers provide structured reasoning.

#### 4.5.4 Multi Provider Resilience

The forecaster distributes workload across 2 model groups in round robin.

Group 0 Primary = Gemini 2.0 Flash, Secondary = GPT 4o mini  
 Group 1 Primary = GPT 4o mini, Secondary = Gemini 2.0 Flash

Each group attempts up to 4 synthesis attempts (FORECAST\_SYNTHESIS\_MAX\_ATTEMPTS = 4) with exponential backoff ( $0.5s \times 2^{\text{attempt}}$ ). On transient failures (rate limits, malformed responses, schema errors), the system falls through to the secondary model. If all attempts are exhausted, the system returns a deterministic **fallback forecast** anchored to the base rate with a plus or minus 15pp CI and the ungrounded flag set.

**Rate limiting.** Each provider has an independent AsyncTokenBucketLimiter at 200 RPM with burst capacity of 20.

### 4.6 Stage 6, Grounding Validation and Auto Heal

Every LLM forecast passes through an **8 point grounding validator** that detects common LLM failure modes.

Check	Flag	Condition
Extreme probability	extreme_probability	p less than 0.05 or p greater than 0.95
Base rate drift	large_base_rate_drift	absolute difference between p and base_rate greater than 0.50
CI too narrow	ci_too_narrow	width less than 0.04

Check	Flag	Condition
CI too wide	ci_too_wide	width greater than 0.70
Non monotonic projections	time_projections_not_monotonic	Projections not consistently increasing or decreasing
Flat projections	flat_time_projections	stdev of projection probabilities less than 0.02
Missing tool reference	synthesis_missing_tool_reference	Calibration notes do not mention any tool keywords
Self reported ungrounded	self_reported_ungrounded	Calibration notes contain the word ungrounded

**Grounding score.**  $\text{passed\_checks} / 8$  (range 0.0 to 1.0)

**Auto heal actions. Ungrounded forecasts** have probability dampened 30% toward base rate using  $p_{\text{new}} = 0.70 \times p + 0.30 \times \text{base\_rate}$ . **CI too narrow** is widened to plus or minus 15pp around probability. **CI too wide** is narrowed to plus or minus 35pp around probability. **Non monotonic projections** are sorted to nearest monotonic sequence (ascending or descending, whichever minimizes L2 distance).

**Retry loop.** If grounding score is less than 0.5 (GROUNDING\_RETRY\_THRESHOLD), the system re runs the forecaster with an explicit retry prompt listing the failed checks. If the retry still fails, the healed (dampened) forecast is persisted.

**Rejection gate.** After all healing, forecasts with grounding score less than 0.5 (GROUNDING\_REJECT\_THRESHOLD) are assigned status = REJECTED and stored but excluded from distribution.

#### 4.7 Stage 7, Advisor Agent (Quantified Decision Support)

For each forecast, an advisor agent generates entity specific strategic recommendations. Unlike conventional risk advisory (which provides qualitative urgency assessments), MASX AI computes **quantified decision support metrics**, a capability not found in any competing platform.

Metric	Calculation	Output Example
<b>Intervention window</b>	Linear interpolation across time projections to find when p crosses threshold (0.50), trajectory detection (accelerating, linear, or decelerating)	You have roughly 23 days before this becomes likely (accelerating trajectory)
<b>Cost of inaction</b>	Risk multiplier, odds(delayed) divided by odds(now) for a 30 day delay	Waiting 30 days increases your risk by 2.3x
<b>Doctrine consensus</b>	Standard deviation of doctrine confidences, sigma 0.10 or less = strong, sigma 0.25 or less = moderate, else weak, identifies divergent doctrines by name	Moderate consensus (sigma = 0.18), Unrestricted Warfare diverges from panel

The advisor produces a list of EntityRecommendation objects (up to 5), each specifying entity type (government, institution, population, private\_sector, international\_org), actor name, specific recommendation, urgency (immediate, short\_term, medium\_term), and doctrine basis.

### 4.7.1 Advisor Output Validator (8 Point Quality Gate)

Just as forecasts pass through the 8 point grounding validator, advisor recommendations pass through a **separate 8 point quality gate** with auto healing capabilities. No other geopolitical platform validates and auto corrects its strategic recommendations.

Check	Flag	What It Catches
Entity diversity	LOW_ENTITY_DIVERSITY	All recommendations target the same entity type (only governments, for example)
Recommendation depth	SHALLOW_RECOMMENDATION	Recommendations under 15 words (too generic to be actionable)
Doctrine basis depth	SHALLOW_DOCTRINE_BASIS	Doctrine justifications under 8 words (hand waving)
Doctrine grounding	UNGROUNDED_DOCTRINE_BASIS	Recommendations cite doctrines that were not in the council is analysis
Urgency probability coherence	URGENCY_PROBABILITY_MISMATCH	Immediate urgency with probability less than 25%, or no immediate when probability greater than 75%
Key driver coverage	KEY_DRIVER_DISCONNECT	Recommendations do not address the forecast is identified key drivers
Geographic coherence	GEOGRAPHIC_DISCONNECT	Recommendations do not reference the hotspot is region, countries, or named entities
Duplicate detection	DUPLICATE_RECOMMENDATION	Jaccard word similarity of 0.75 or higher between two recommendations

**Auto heal actions.** Duplicate recommendations are automatically dropped. Urgency mismatches are corrected with the lead recommendation is urgency set to the probability inferred level (75% or higher maps to immediate, 40% or higher maps to short\_term, else medium\_term). Quality score computed as  $(8 \text{ minus unique\_flags}) / 8$ .

This creates a **dual quality gate architecture**. The grounding validator ensures forecast accuracy, while the advisor validator ensures recommendation quality. Both operate deterministically, ensuring consistent output regardless of LLM behavioral drift.

## 5. Post Pipeline Operations

### 5.1 Forecast Resolution Pipeline

The resolution pipeline is a critical subsystem that converts open forecasts into calibration data. It runs as a daily sweep, identifying all forecasts whose horizon date has passed and systematically determining their outcomes.

#### 5.1.1 Due Forecast Identification

The sweep queries `forecast_store.fetch_due_for_resolution(as_of=today)` for all open forecasts past their horizon date. Already resolved forecasts are checked via `resolution_store.exists_for_forecast()` and skipped to prevent double resolution.

### 5.1.2 Multi Turn Web Search

For each due forecast, the system conducts a **multi turn search strategy** using up to 5 search turns (`RESOLVER_MAX_SEARCH_TURNS = 5`), each returning up to 5 results (`RESOLVER_SEARCH_RESULTS_PER_QUERY = 5`).

- Turn 1 Raw forecast question
- Turn 2 Question + resolution criteria keywords (first 200 chars)
- Turn 3 Question + key driver #1
- Turn 4 Question + key driver #2
- Turn 5 (reserved for additional criteria)

Search queries are built programmatically from the forecast is own structured data (question text, resolution criteria, key drivers), ensuring search relevance. URLs are deduplicated across turns to prevent redundant evidence.

**Search provider chain.** Serper to Tavily (fallback). The search provider is injected via Protocol interface (`SearchProvider`), enabling provider substitution without code changes.

### 5.1.3 Hallucination Proof URL Verification

A critical anti hallucination safeguard. `_strip_hallucinated_urls()` cross references every URL cited by the LLM against the set of URLs actually returned by the search provider. Any URL the LLM fabricates (a known LLM failure mode) is silently removed from the verdict is evidence list. Only URLs that exist in the actual search results are retained.

### 5.1.4 LLM Resolution Judgment

The resolver agent (Gemini 2.0 Flash) receives the forecast is question, resolution criteria, horizon date, predicted probability, key drivers, disconfirming evidence, and all search results. It produces a structured `ResolverLLMOutput`.

Field	Type	Purpose
<code>outcome</code>	bool or None	true, false, or null (ambiguous)
<code>is_ambiguous</code>	bool	Explicit ambiguity flag
<code>corroboration_count</code>	int, 0 or higher	Number of independent sources confirming the outcome
<code>contradictory_evidence_found</code>	bool	Whether conflicting evidence was found
<code>evidence_summary</code>	str	Narrative summary of the evidence
<code>evidence_urls</code>	list[str]	URLs cited (post hallucination stripping)
<code>driver_hits</code>	list[str]	Key drivers that materialized
<code>driver_misses</code>	list[str]	Key drivers that did NOT materialize
<code>disconfirming_hits</code>	list[str]	Disconfirming evidence that DID appear
<code>signals_missed</code>	list[str]	Signals the original forecast missed entirely
<code>criteria_quality</code>	str	Grade of the resolution criteria itself (sufficient, ambiguous, unfalsifiable, or vague)

### 5.1.5 Corroboration Gate

A hard rule. If `corroboration_count` is less than `RESOLVER_MIN_CORROBORATION` (2) and the LLM claims a definitive outcome, the verdict is **automatically overridden to ambiguous**. This prevents the system from resolving forecasts based on a single unreliable source. The LLM outcome is set to None and `is_ambiguous` is forced to True.

### 5.1.6 Verdict Mapping and Brier Score

The LLM output is mapped to a `ResolutionVerdict` domain model.

Auto resolved (true or false) verdicts get a Brier score computed immediately as (probability minus outcome) squared. Ambiguous verdicts get no Brier score, and the forecast is excluded from calibration (does not contaminate performance data). Expired unresolved forecasts with no resolution criteria are marked expired without LLM judgment.

The Brier score, probability at resolution, verdict metadata, and all post mortem data are persisted together in the resolution store.

### 5.1.7 Criteria Quality Analysis

The system includes a dedicated `criteria_analyzer.py` module that aggregates resolution verdicts to surface **systemic patterns causing ambiguous resolutions**. It counts resolution outcomes (clean resolved vs. ambiguous vs. expired), tracks criteria quality grades (sufficient vs. unfalsifiable vs. vague), computes ambiguity rate across all verdicts, extracts recurring failure phrases from ambiguous verdicts using regex pattern matching (no clear evidence, partially occurred, conflicting reports, insufficient evidence), and returns patterns that appear 2 or more times as `top_failure_patterns`.

This enables a **Phase 4 feedback loop**. If the criteria analyzer identifies that 40% of forecasts are resolving as ambiguous due to insufficient evidence patterns, the question generation agent can be adjusted to produce more falsifiable resolution criteria.

### 5.1.8 Search Health Monitoring

The sweep tracks a `search_health` metric, `resolved / (resolved + failed)`. If this drops below 0.50 (`SEARCH_HEALTH_THRESHOLD`), a structured warning is logged, indicating potential search provider degradation. The sweep continues processing remaining forecasts regardless. It never fails entirely due to individual forecast resolution errors.

### 5.1.9 Post Mortem Data Flow

The resolution verdict is structured post mortem data feeds three downstream systems.

<code>driver_hits / driver_misses</code>	Calibration (which analyses predict correctly?)
<code>disconfirming_hits</code> evidence?)	Calibration (did the system properly weight disconfirming
<code>signals_missed</code>	Channel 3 of feedback loop (injected into future prompts)
<code>criteria_quality</code>	Criteria analyzer (improving question generation)

## 5.2 Self Calibrating Feedback Loop Architecture

The self calibration system is MASX AI is most strategically significant innovation. It creates a **compounding competitive advantage**. Every resolved forecast makes the system measurably more accurate, building a moat that widens with each daily pipeline run.

### 5.2.1 Brier Score Decomposition

When forecasts are resolved, the system computes a full Brier decomposition.

$\text{Brier} = \text{Reliability} \text{ minus } \text{Resolution} + \text{Uncertainty}$

**Reliability** measures how close predicted probabilities are to actual hit rates in each bin (lower = better calibrated). **Resolution** measures how much hit rates vary across bins (higher = better discrimination). **Uncertainty** is the base rate entropy (irreducible, sets the difficulty floor).

Forecasts are binned into 10 probability bins. This decomposition diagnoses *what kind* of error the system is making. Reliability errors (overconfidence or underconfidence) are correctable. Resolution failures require better signal processing.

### 5.2.2 Three Channel Feedback

The calibration system feeds back through three independent channels simultaneously.

#### Channel 1, Mathematical Weight Adjustment (Doctrine Aggregation)

Brier scores are computed per doctrine agent. These are inverted to produce reliability weights stored in the `doctrine_performance` table. On subsequent pipeline runs, `multi_doctrine_log_odds()` scales each doctrine's contribution by its historical accuracy.

$\text{log\_posterior} = \text{log\_prior} + \text{sum}(w_i \times d_i \times c_i)$

where  $w_i$  = calibration derived weight (from `doctrine_perf_store`),  $d_i$  = direction,  $c_i$  = confidence. Doctrines with better track records get more influence. New doctrines without resolution history receive a uniform default weight. The weight computation and storage infrastructure is complete. Weight injection into the live forecaster aggregation is scheduled for activation once sufficient resolution volume is reached.

#### Channel 2, Domain Specific Prompt Guidance (LLM Behavioral Correction)

The `build_calibration_context()` function injects domain specific performance feedback directly into the forecaster prompt. Overall Brier greater than 0.25 triggers the message Calibration is POOR, anchor more tightly to base rates. Domain Brier significantly worse than average triggers Your [domain] forecasts are roughly Xpp WORSE than average, increase anchoring. Strong calibration triggers Maintain current methodology. This channel is **fully operational** in the current production pipeline.

This acts as a behavioral corrective. Even if the LLM model changes, the prompt carries forward the system's historical calibration lessons.

#### Channel 3, Past Signals Missed (Error Learning)

Resolution post mortems record which key drivers and disconfirming signals the system missed. These `past_signals_missed` are injected into subsequent forecaster prompts for the same hotspot, creating an explicit learning mechanism. In similar previous forecasts, you missed these signals. The resolution pipeline captures and persists all post mortem data. Injection into the forecaster prompt is scheduled for activation alongside Channel 1 as resolution volume grows.

### 5.2.3 The Compounding Effect

Day 1 Doctrines weighted equally (uniform prior)  
Day 30 First resolutions, initial weights computed  
Day 90 Weight divergence emerges (some doctrines 2x others)  
Day 180 Pattern stabilizes, specific doctrines dominate their domains  
Day 365 System has domain specific calibration profiles

This creates a **compounding competitive advantage** that cannot be replicated by a competitor starting from scratch. The calibration data itself becomes proprietary intellectual property, hard won empirical knowledge about which strategic frameworks predict which domains most accurately.

#### 5.2.4 Minimum Data Requirements

The calibration runner requires a minimum of 10 resolved forecasts before computing meaningful Brier decomposition. Below this threshold, the system operates with uniform weights and no prompt corrections. This is an honest limitation. The feedback loop is value is proportional to the volume of resolved forecasts.

### 5.3 Autonomous Multi Channel Distribution

Forecasts are distributed through four autonomous channels, making MASX AI a **complete intelligence media platform**, not just a forecasting engine.

Channel	Technology	Output
<b>Database</b>	Supabase (PostgreSQL)	All forecasts persisted with full metadata
<b>Newsletter</b>	Resend API + Jinja2 HTML	Rich briefing with 7 inline data charts, BCC batch delivery
<b>LinkedIn</b>	LLM generated post + daily chart	Structured post with rotating chart type, emailed as draft for review
<b>Flagship Post</b>	LLM generated (Monday only)	Top risk narrative + 3 signals to watch + operator actions
<b>Podcast</b>	LLM script to Gemini TTS to MP3 to Supabase Storage	Daily + weekly recap audio briefings

#### 5.3.1 Deterministic Newsletter Builder

The newsletter is **100% deterministic**, no LLM call. It assembles structured data directly from pipeline outputs. Situation overview (active hotspots, open forecasts, average probability, domain breakdown), per hotspot digest with urgency classification (CRITICAL, HIGH, ELEVATED, MODERATE), top recommendations ranked by urgency, and a watchlist of newly emerged hotspots and high probability (60%+) forecasts. Seven data visualization charts are generated via matplotlib with MASX branded theming and embedded inline.

#### 5.3.2 Autonomous Podcast Pipeline

The system includes a **fully autonomous podcast generation pipeline**, a capability entirely unique in the geopolitical intelligence space. No competing platform produces audio briefings from forecast data without human involvement.

The pipeline operates in five stages. Hotspot selection prioritizes Tier 1 hotspots (high probability + recommendations) with tier 2 backfill for domain diversity. The `select_pillar_hotspots()` function ensures geographic and domain diversity across segments.

Script generation uses an LLM agent to produce a podcast script with dynamically allocated word budgets (roughly 1,400 total words, distributed based on segment count). The system loads **prior scripts** for the same hotspots to avoid repetitive coverage across episodes.

Text to speech synthesis uses Gemini TTS to convert the script to PCM audio per segment, with edge tts as an automatic fallback. Segments are then encoded to MP3.

Storage and publication uploads the MP3 to Supabase Storage. Episode metadata, segments, and per hotspot scripts are persisted to the database.

Failure recovery ensures that if audio synthesis fails, the script is still persisted as SCRIPT\_ONLY, ensuring no content is lost. If the database write fails after upload, the uploaded audio is automatically cleaned up.

**Weekly recap episodes** aggregate the week is most significant hotspots into a longer form recap format, drawing from the daily episode history.

**Why this matters to investors.** The podcast pipeline transforms MASX AI from a forecasting engine into a **content production platform**. One pipeline run, five distribution channels, zero marginal content cost per subscriber. The audio is genuinely listenable, not the robotic TTS you might expect.

## 6. Technology Stack

Component	Technology	Role
<b>Data Pipeline</b>		
NER	ONNX Runtime (distilbert NER)	CPU only batch entity extraction, custom BIO decoder
Embeddings (Data Pipeline)	paraphrase multilingual MiniLM L12 v2 (INT8 quantized)	384 dim article embeddings, 427 MB peak
Cross Domain Detection	DeBERTa v3 xsmall + 122 node embedding matrix	3 tier hybrid domain classification
Deduplication	SimHash (64 bit) + MinHash	Near duplicate and syndication detection
Content Extraction	trafilatura + readability lxml	HTML to text extraction with fallback chain
TLS Impersonation	curl_cffi	Browser TLS fingerprint spoofing for anti bot bypass
Headless Browsers	Patchright (Windows) and Camoufox (Linux)	JS rendered page extraction
Clustering	HDBSCAN + cosine similarity	Multi scale event clustering at 3 geographic levels
<b>Forecasting Pipeline</b>		
LLM Framework	pydantic ai	Agent orchestration, structured output, tool calling
Primary LLM	Gemini 2.0 Flash	High speed forecasting and council analysis

Component	Technology	Role
Secondary LLM	GPT 4o mini	Cross provider fallback
RAG	LlamaIndex VectorStoreIndex	Per doctrine semantic retrieval
Embeddings (RAG)	all MiniLM L6 v2	Local 384 dim sentence embeddings
Reranking	ms marco MiniLM L 6 v2	Local cross encoder reranking
Database	Supabase (PostgreSQL)	Hotspots, forecasts, podcasts, newsletters, calibration
Validation	Pydantic v2	Strict schema enforcement on all domain models
Settings	pydantic settings BaseSettings	Environment based configuration
HTTP	httpx + curl_cffi	Async HTTP client with TLS impersonation fallback
Math	scipy (Wilson CI, Jeffreys CI)	Statistical confidence intervals
TTS	Gemini TTS + edge tts fallback	Autonomous podcast audio synthesis
Testing	pytest (41 test files)	AAA pattern, fixtures, unit + integration
Deployment	Docker + Railway	Containerized production deployment

## 7. Key Design Decisions

**Fully autonomous pipeline.** The entire chain, from signal ingestion through forecasting, validation, distribution, resolution, and calibration, runs daily without human intervention. The only human touchpoint is LinkedIn draft review. I have not found a competing system that achieves this level of end to end autonomy for geopolitical forecasting.

**Self calibrating feedback loop as compounding moat.** The three channel calibration architecture (mathematical weight adjustment + prompt guidance + error learning) creates a competitive advantage that deepens with every resolved forecast. A competitor starting today would need to build months of resolution history before matching the accumulated calibration data.

**Deterministic tools constrain LLM creativity.** The forecaster cannot make up a probability. It must call calculation tools and justify departures from their outputs. The grounding validator enforces this by checking for tool references in the calibration notes.

**Multi doctrine synthesis prevents single framework bias.** By routing each hotspot through 5 independent doctrine analyses (from potentially different strategic traditions), the system surfaces contradictions and builds consensus weighted estimates.

**Quantified decision support, not just probabilities.** The advisor agent computes intervention windows and cost of inaction risk multipliers, unique outputs that translate probabilistic forecasts into actionable timing advice for decision makers.

**Graceful degradation over failure.** At every stage, the system degrades gracefully. Provider fallback, tool fallback guidance, deterministic fallback forecasts, auto heal, rejection gates. The pipeline never crashes. I have watched it survive API outages, malformed LLM responses, and search provider failures without any manual intervention. It always produces output with a transparent quality signal.

## 8. Competitive Differentiation

MASX AI occupies a unique position in the geopolitical intelligence landscape. While individual AI forecasting techniques are advancing rapidly (Metaculus AI bots are closing in on superforecaster accuracy, Brier gap roughly 0.017 as of early 2026), no single competitor or combination of competitors replicates the full integrated chain from signal ingestion through self calibration.

Capability	MASX AI	Nearest Competitor	Gap
Full autonomous chain (signal to forecast to distribute to resolve to calibrate)	Complete 14 stage data pipeline + 7 stage forecasting pipeline, zero human	Metaculus AI bots (forecast only), Recorded Future Autonomous Threat Ops (threat hunting only)	No competitor runs the complete ingestion to calibration chain autonomously
14 stage data pipeline with 4 gate corroboration	4 deterministic quality gates (junk filter to corroboration to substance to composite scoring), syndication detection, bias risk guards	GDELT (ingestion + event coding only, no corroboration gating)	No competitor applies progressive multi gate corroboration with syndication collapsing and bias detection
3 tier cross domain detection	Static rules to embedding cosine to zero shot classifier (DeBERTa), all CPU only, \$0 cost	Recorded Future (NLP based entity extraction)	No competitor runs a 3 tier hybrid domain classification pipeline entirely on CPU
7 component composite scoring with lifecycle awareness	Volume, recency, diversity, topic, velocity, T1 convergence, anomaly bonus, lifecycle aware	GDELT (Global Knowledge Graph scoring)	No system combines 7 weighted scoring components with cross day lifecycle velocity re

Capability	MASX AI	Nearest Competitor	Gap
	freshness ratio		scoring
Cross day lifecycle state machine	born to active to fading to expired to revived with centroid matching (cosine 0.80+ AND country overlap)	N/A	No competitor tracks event evolution across pipeline runs with embedding based cross day matching
Coverage confidence and anomaly detection	Per region daily monitoring for censorship, blackout, and surge detection across 17 regions	N/A	No competitor detects government level information suppression through cross channel confidence analysis
7 tier fetch escalation with SSRF defense	httpx to curl_cffi to proxy to browser to Google Cache, plus DNS pre flight SSRF protection	N/A	No competitor implements a 7 tier fetch escalation chain with defense in depth SSRF protection
Strategic doctrine corpus as analytical lenses	35 named doctrine corpora with directional voting via per doctrine RAG	Good Judgment (human experts choose frameworks)	No automated system uses named strategic traditions as RAG grounded analytical frameworks
Log odds Bayesian aggregation with calibration weights	multi_doctrine_log_odds() with per doctrine Brier derived weights	Metaculus (recency weighted crowd median)	No system applies log odds aggregation with per agent historical calibration weights
Three channel self calibrating feedback loop	Weights + prompt guidance + error learning from resolved forecasts	Good Judgment (human accuracy tracking, no automated feedback)	No automated system feeds per doctrine Brier decomposition back into Bayesian aggregation
CAMEO to Goldstein to Weibull temporal chain	Event taxonomy to escalation scoring to hazard model to	GDELT (descriptive CAMEO and Goldstein only)	No system uses Goldstein scores as inputs to a

Capability	MASX AI	Nearest Competitor	Gap
	survival projection		Weibull hazard model for probability projection
16 check dual quality gate with auto heal	8 point forecast grounding + 8 point advisor validator, both deterministic with auto correction	Adarga (neuro symbolic reasoning for accountability)	No system applies dual deterministic validation with auto heal on both forecasts and recommendations
Self improving question generation	36 term blocklist evolving from resolution failure analysis (3 tiers)	N/A	No system evolves question quality standards from empirical resolution outcomes
Quantified intervention windows	You have roughly X days + cost of inaction risk multiplier	Eurasia Group (qualitative urgency)	No competitor computes quantified intervention timing from probability projections
Autonomous content production	Newsletter (7 charts) + LinkedIn + flagship + daily and weekly podcast (TTS to MP3)	N/A	No competitor produces autonomous audio briefings from forecast data

## Appendix A, Doctrine Corpus (35 Doctrines)

The 35 doctrines are organized into seven categories. Each doctrine has affinity mappings to one or more of the seven domain pillars (governance, conflict, economy, technology, society, environment, infrastructure) at HIGH or MEDIUM affinity levels.

Category	Doctrines (representative examples)
Classical Statecraft	Art of War, Arthashastra, Mahabharata, Panchatantra, Shivaji (Ganimi Kava)
Geopolitics and Grand Strategy	Heartland Theory, Rimland Theory, Sea Power, Containment Strategy, Kennan Containment, Diplomacy (Kissinger), Great Power Politics, RAND Corporation, Smart Power, National Security Strategy, Clash of Civilizations, Iroquois Great Law of Peace
Hybrid and Cognitive Warfare	Fifth Generation Warfare, Unrestricted Warfare, MindWar, LikeWar, Wag the Dog, Deep State, Cyber War
Economics and Trade	Wealth of Nations, Currency Wars, Confessions of an Economic Hit Man, The Shock Doctrine
Science and	The Innovator is Dilemma, Superintelligence, Cyber Warfare

Category	Doctrines (representative examples)
Technology	
Earth and Environment	The Limits to Growth, Collapse, Silent Spring
Infrastructure and Systems	Networks of Power

Note. Some doctrines span categories (for example, cyber warfare doctrines bridge hybrid warfare and technology). The total of 35 reflects the registry is unique doctrine entries, each with distinct affinity mappings and source PDFs.

## Appendix B, Constants Reference

Constant	Value	Purpose
PIPELINE_HOTSPOT_CONCURRENCY	6	Max parallel hotspot processing
ROUTER_TOP_K	5	Doctrines selected per hotspot
ROUTER_RELEVANCE_CUTOFF	0.5	Minimum relevance score for inclusion
MIN_DOCTRINES_FOR_LOG_ODDS	2	Minimum doctrines for Bayesian aggregation
RESOLUTION_HORIZON_BANDS	(14, 30, 90)	Days forward for forecast questions
FORECAST_SYNTHESIS_MAX_ATTEMPTS	4	Retries before deterministic fallback
GROUNDING_REJECT_THRESHOLD	0.5	Below this, forecast status = REJECTED
PROB_CLAMP_LO and HI	0.01 and 0.99	Hard probability bounds
SIGNAL_COMPOSITE_LR_CAP	6.0	Maximum signal likelihood ratio
FORECASTER_TEMPERATURE	0.0	Deterministic LLM output
RETRIEVAL_TOP_K	20	RAG candidates before reranking
RERANK_TOP_N	5	Final RAG chunks per doctrine
EMBED_DIMENSION	384	Embedding vector size
MIN_FORECASTS_FOR_CALIBRATION	10	Required resolved forecasts for calibration