Enterprise NLU Migration & Edge-Compute Moderation Architecture

The enterprise architecture governing artificial intelligence safety, content moderation, and user-generated text processing is undergoing a fundamental and accelerated paradigm shift. Driven by the impending sunset of Google's Perspective API and the deprecation of legacy moderation tools like Two Hat Community Sift, enterprise security teams and software architects are migrating from legacy, probabilistic machine learning classifiers toward deterministic Natural Language Understanding (NLU) engines and edge-compute Web Application Firewalls (WAFs). This transition represents a structural move away from application-layer, keyword-based toxicity scoring and toward semantic analysis executed directly at the transport layer.

The integration of advanced NLU engines, specifically Tisane Labs and the recently rebranded Alice (formerly ActiveFence), enables highly contextual, deterministic abuse detection that integrates deeply with orchestrators like NVIDIA NeMo Guardrails. Concurrently, the efficacy of these proprietary engines is augmented by the continuous ingestion of highly specialized semantic training corpora supplied by third-party risk intelligence providers. Organizations such as CyberWell, the Anti-Defamation League (ADL), and the USC Shoah Foundation operate massive, verifiable data pipelines that inform these models, ensuring they remain resilient against emerging algorithmic manipulation, zero-day prompt injections, and sophisticated disinformation campaigns.

Simultaneously, the deployment of these moderation controls is shifting toward the network edge. Solutions such as Cloudflare's AI Security for Apps (formerly Firewall for AI) and Amazon Web Services (AWS) Bedrock Guardrails intercept HTTP requests and API payloads before they ever reach foundational Large Language Models (LLMs). By analyzing prompts for personally identifiable information (PII), malicious injection attempts, and semantically denied topics at the WAF level, enterprises achieve a secure "fail-closed" posture. This architectural shift significantly reduces LLM compute costs, minimizes latency, and effectively mitigates the risk of model poisoning and hallucination at scale.

The Forcing Function: Sunsetting Probabilistic Toxicity Scoring

The retirement of the Perspective API serves as the primary forcing function for enterprise architecture modernization. For years, platforms have relied heavily on this free service to filter comment sections and user interactions, often falling victim to "automation bias"—the tendency to blindly trust algorithmic decisions without questioning obvious errors. Google announced the complete end-of-life for the Perspective API for December 31, 2026, with new API request submissions closing as early as February 2026, offering no direct migration support. Similarly, Microsoft's sunsetting of the Two Hat Community Sift platform has left enterprises scrambling to rebuild real-time text moderation pipelines.

The fundamental limitation of the Perspective API, and probabilistic classifiers in general, is their reliance on surface-level pattern matching and statistical inference rather than true semantic comprehension. The API returns floating-point probability scores between 0.0 and 1.0 across explicitly requested attributes such as TOXICITY, SEVERE_TOXICITY, and INSULT. This

architectural design forces developers to build extensive custom application-layer logic to define acceptable mathematical thresholds, leading to highly variable enforcement outcomes across different communities.
Because these systems lack contextual understanding, they are highly vulnerable to adversarial text manipulations, contextual collapse, and "Algospeak"—the intentional obfuscation of terminology by malicious actors to evade static keyword blocklists. Furthermore, probabilistic ML classifiers function as opaque black boxes; they cannot provide deterministic explanations for their scoring, creating severe auditability and compliance challenges for enterprise Trust and Safety teams operating under emerging frameworks like the EU Digital Services Act (DSA). The modern enterprise requires an architecture that is deterministic, explainable, and multi-layered, necessitating a complete replacement of these legacy endpoints with advanced NLU platforms capable of executing complex linguistic evaluations.

Architectural Transition to Deterministic NLU Engines

The migration toward next-generation Natural Language Understanding requires enterprises to adapt their API workflows, rewrite integration payloads, and restructure data ingestion pipelines. The architectures of Tisane Labs and Alice (ActiveFence) provide a comparative look at how the industry is solving the limitations of probabilistic models through deterministic analysis and massive adversarial data intelligence.

Tisane Labs: Deterministic Disambiguation and Headless Architecture

Tisane Labs operates on a foundational philosophy of deterministic Natural Language Understanding, specifically focusing on law enforcement, intelligence, and high-volume content moderation. Unlike standard ML classifiers that rely on black-box neural networks, Tisane utilizes a headless architecture that separates design-time logic from the runtime engine. This ensures a "Single Point of Truth" (SPOT), where the exact same language models are used consistently across entity extraction, sentiment analysis, and abuse tagging. The runtime engine guarantees that identical inputs will yield identical outputs deterministically, regardless of server workload or hardware architecture, deliberately bypassing the "Nvidia tax" associated with GPU-heavy LLM inference.
The core of the Tisane architecture relies on sense disambiguation. The engine ingests text and constructs a complex knowledge graph where the nodes represent "word senses"—the combination of a lexical item and its precise contextual interpretation—rather than mere string values. The platform executes proprietary Finite State Automata over this graph to detect higher-level patterns, extracting entities, evaluating aspect-based sentiment, and categorizing problematic content across more than 30 languages without requiring language-specific statistical retraining.
For enterprises migrating from the Perspective API or Community Sift, the transition to the Tisane API requires a structural refactoring of request payloads and decision logic.

Feature Category	Perspective API Architecture	Tisane API Architecture
Endpoint URL	POST https://commentanalyzer.googleapis.com/v1alpha1/comments:analyze	POST https://api.tisane.ai/parse
Authentication Strategy	URL Query Parameter	HTTP Header

Feature Category	Perspective API Architecture	Tisane API Architecture
	(?key=API_KEY)	(Ocp-Apim-Subscription-Key: YOUR_API_KEY)
Primary Input Field	comment.text	content (requires ISO 639-1 language code)
Attribute Selection	Explicitly defined via requestedAttributes	Excluded. All detected instances returned by default
Output Paradigm	Floating-point probability score (0.0 to 1.0)	Structured JSON abuse array
Evaluation Logic	Developer-defined mathematical thresholds	Engine-defined severity levels (low, medium, high, extreme)

The Tisane response payload abandons the concept of probability. Instead, it returns an array of discrete abuse instances, each tagged with a specific type (e.g., personal_attack, bigotry, criminal_activity), a severity classification, the specific character offset and length of the offending span, and a human-readable explanation designed to satisfy audit and compliance requirements.

Enterprise architects must map legacy Perspective attributes to Tisane's severity-based ontology. For example, a legacy [span_21](start_span)[span_21](end_span)TOXICITY request must be refactored in the application layer to trigger a mitigation action if the Tisane payload returns a personal_attack or bigotry type at a medium or higher severity. The legacy THREAT attribute maps directly to Tisane's criminal_activity type accompanied by the threat tag. To facilitate tighter contextual understanding, Tisane introduces localized context cues via a settings object in the POST request, utilizing formatting flags such as "dialogue", "review", or "alias" to structurally alter how the Finite State Automata processes grammatical structures, highly optimizing the detection of obfuscated profanity in usernames or live chat streams.

For edge and on-premises deployments, Tisane provides embedded SDKs. The core runtime library is written in POSIX-compliant C/C++ and utilizes RocksDB to store its language models natively on the host machine, bypassing the need for remote REST API calls entirely. A native .NET wrapper is also provided, allowing enterprises to toggle between "Lazy loading" and "Fully Loaded Mode" depending on available system memory and latency constraints.

Alice (ActiveFence) and the Rabbit Hole Engine

While Tisane focuses on deterministic, rules-based graph execution, Alice (rebranded from ActiveFence in early 2026) operates an architecture centered around massive, real-time adversarial data ingestion and policy enforcement across the entire generative AI lifecycle. Moving beyond standard Trust and Safety operations, Alice positions itself as the foundational safety layer for communicative tech, providing security for seven of the ten largest AI foundation models globally.

Alice's architecture is powered by an adversarial intelligence engine known as "Rabbit Hole." The Rabbit Hole engine is a continuously updating intelligence database built from a decade of OSINT (Open Source Intelligence), dark web scraping, and deep-web investigations, analyzing billions of toxic, manipulative, and abusive data samples across 120 languages. This engine feeds the WonderSuite platform (comprising WonderBuild, WonderFence, and WonderCheck), which provides automated red-teaming, runtime guardrails, and continuous production evaluation for foundational AI models and agents against exploits like prompt injection and data exfiltration.

To process this massive volume of threat intelligence, the engineering architecture behind Alice underwent a significant evolution. Initially relying on a Feathers.js data access layer on top of MongoDB, the system faced severe bottlenecks as real-time snapshot generation became unfeasible. The engineering team inverted the architecture, establishing an Amazon S3 data lake as the definitive source of truth. A dedicated ingestion system now formats scraped dark web data, manual uploads, and API calls, pushing the formatted records into the S3 data lake via Amazon Kinesis Data Firehose. Apache Airflow handles the offline batch processing for retraining detection models, while a caching layer in Amazon DynamoDB facilitates low-latency random access for the user interface.

The integration of Alice's threat intelligence into enterprise AI orchestrators is most visibly documented in its native integration with NVIDIA NeMo Guardrails. Through the ActiveScore API, Alice provides a mechanism to intercept and evaluate both user input and LLM output. The architectural workflow relies on customized Colang (.co) files, a domain-specific language used to define programmable dialogue flows.

When a user submits a prompt, the NeMo Guardrails orchestrator triggers an overriding system action executing an asynchronous Python HTTP POST request to the ActiveFence endpoint (https://apis.activefence.com/sync/v3/content/text). The request payload encapsulates the user's string in a text field alongside a dynamically generated content_id (e.g., "ng-" + new_uuid()), authenticated via an af-api-key HTTP header.

The ActiveScore API response returns a JSON payload containing a max_risk_score and a highly detailed violations_dict mapping specific violation types (e.g., abusive_or_harmful.hate_speech) to their respective risk scores. Enterprise architects can construct granular subflows in Colang to enforce specific business logic based on this dictionary. If the max_risk_score exceeds a predefined programmatic threshold (e.g., 0.85), the orchestrator halts the inference pipeline and forces the bot to return a predefined refusal:

```
define subflow activefence moderation
"""Guardrail based on the maximum risk score."""
$result = execute call activefence api
if $result.max_risk_score > 0.85
  bot inform cannot answer
  stop
```

Furthermore, the violations_dict allows for precise, vector-specific interventions. A subflow can be engineered to allow high-risk scores generally, but explicitly block the transaction if a specific dictionary key evaluates above a stricter threshold. By overriding the system action in the actions.p[span_59](start_span)[span_59](end_span)[span_61](start_span)[span_61](end_span)y file to accept arbitrary text (e.g., $result = execute call activefence api(text=$bot_message)), the orchestrator can enforce these exact same ActiveScore policies on the outbound LLM response, ensuring the model itself does not generate prohibited content.

Semantic Training Corpora and Third-Party Intelligence Routing

The deterministic accuracy of engines like Tisane and the adversarial awareness of Alice's Rabbit Hole are heavily reliant on the continuous ingestion of high-fidelity semantic data. The technical architecture of enterprise moderation relies on complex API data-routing workflows that connect these NLU engines to non-governmental organizations (NGOs) and civil society

research centers. These organizations provide structured, domain-specific corpora that translate nebulous societal harms into machine-readable datasets, essentially functioning as the raw intelligence feed for the moderation ecosystem.

CyberWell: Routing Real-Time Antisemitism Datasets

CyberWell operates as the world's first open, live database dedicated exclusively to tracking and cataloging online antisemitism across major social media platforms. By utilizing advanced social listening tools and massive data aggregation, CyberWell identifies emerging trends in hate speech, including the rapid proliferation of AI-generated violent rhetoric and deepfakes. A recent dataset compiled by CyberWell analyzed 307 vetted pieces of AI-generated antisemitic content—garnering over 30 million views—and mapped the platforms' disparate enforcement rates, noting that 87.3% of explicitly violent AI-generated posts were removed, but enforcement remained highly inconsistent across different models and networks.

From an architectural standpoint, CyberWell acts as a critical node in the broader threat intelligence pipeline. It packages these vetted instances of policy-violating content into structured datasets that categorize recurring narratives, such as event-driven violent rhetoric or digital Holocaust denialism. This structured data is routed to enterprise security centers and federal law enforcement—such as the FBI's National Threat Operations Center—via privileged, real-time API connections.

For NLU platforms and WAFs, ingesting this real-time API feed allows threat detection algorithms to continuously map new linguistic permutations, slang, and visual generation prompts used by adversaries. By absorbing CyberWell's datasets on how AI models facilitate the generation of specific tropes, moderation engines can preemptively tune their classifiers, update their regular expressions, and adjust their semantic graphs to recognize novel, highly contextual abuse patterns before they achieve mainstream viral distribution.

Anti-Defamation League (ADL): Structural Extremist Intelligence

The ADL’s Center for Technology and Society serves a structurally similar function by providing deep intelligence regarding how hate and extremism manifest across emerging digital mediums, particularly within generative AI and multi-player gaming ecosystems. The ADL tracks the use of coded language, algorithmic manipulation, and coordinated harassment campaigns, highlighting that 76 percent of adults and 74 percent of youth experience harassment in online gaming platforms.

The data curated by the ADL feeds into the broader Trust and Safety ecosystem, providing essential context that allows NLU engines to perform accurate sense disambiguation. Because modern adversarial actors rely heavily on coded "Algospeak" to bypass standard keyword blocklists, the ADL’s intelligence is critical for training NLU knowledge graphs to interpret seemingly benign phrases as coordinated harassment or terrorist radicalization. By integrating datasets highlighting the intersection of violent extremism and digital communication, moderation platforms can accurately classify the semantic intent behind complex, multi-turn interactions, preventing AI agents from being exploited for relational manipulation or the dissemination of extremist propaganda.

USC Shoah Foundation: Verifiable Historical Corpora and

Cryptographic Grounding

While CyberWell and the ADL provide active threat intelligence regarding emerging harms, the USC Shoah Foundation provides a distinctly different architectural requirement for AI safety: verifiable historical truth to prevent LLM hallucination and the trivialization of historical fact.
The USC Shoah Foundation houses the Visual History Archive, an unparalleled repository containing over 55,000 video testimonies of survivors and witnesses to the Holocaust, the Armenian Genocide, and other atrocities. Building upon this archive, the Foundation pioneered the "Dimensions in Testimony" (DiT) project, which merges advanced filming techniques (such as volumetric capture using light stages), specialized display technologies, and advanced natural language processing (NLP) to create interactive, conversational biographies. The DiT NLP architecture utilizes a speech recognition parser to convert spoken visitor queries into text strings, which are then semantically matched against a database of roughly 1,000 pre-recorded answers per survivor, dynamically retrieving the most contextually appropriate video response in real-time.
However, the rise of generative AI presents a severe threat to historical archives, enabling the rapid generation of photorealistic deepfakes and synthetic historical denialism. To ensure the integrity of its records against these threats, the USC Shoah Foundation partnered with Stanford University to establish the Starling Lab for Data Integrity. The Starling Framework provides an architectural blueprint for data provenance, operating on a strict "Capture, Store, Verify" cryptographic methodology.
From a technical perspective, the framework utilizes advanced cryptography to embed immutable metadata directly at the source of the capture. Using specialized hardware and software (such as HTC sensors and the Guardian Project tools), the camera pairs the digital asset with secure metadata spanning GPS coordinates, gyroscope telemetry, and barometric pressure. The media is then cryptographically hashed to create a unique Content Identifier (CID), serving as an unalterable digital fingerprint.
Crucially, the Starling architecture abandons centralized databases in favor of Web 3.0 decentralized storage protocols, such as IPFS (InterPlanetary File System) and Filecoin. Because the CID inherently addresses the data, the alteration of a single pixel or audio byte by a malicious actor will inherently alter the resulting hash, immediately flagging the asset as compromised. Furthermore, public and permissioned decentralized ledgers allow experts to append verifiable certifications of authenticity to the asset's immutable record.
For enterprise AI and NLU moderation, this cryptographic architecture represents the absolute future of semantic grounding. When an LLM is queried about historical events, edge-compute WAFs and moderation orchestrators can verify the LLM's output against the cryptographically secured corpora maintained by the Starling Lab. If the LLM generates a response that contradicts the verified hash-chained records, or if an adversary attempts a prompt injection to generate synthetic historical denialism, the guardrail system can deterministically block the output based on cryptographic mathematical proof rather than relying on flawed statistical probability or secondary fact-checking APIs.

Edge-Compute Moderation: Intercepting Prompts at the Transport Layer

While NLU engines provide the logic for evaluating semantic safety, the physical execution of that logic is rapidly moving toward the network edge. Legacy moderation systems typically

evaluate content asynchronously via webhook after it is submitted, or deep within the application layer prior to database insertion. The modern AI architecture intercepts API calls and HTTP requests inline, utilizing Web Application Firewalls (WAFs) to drop malicious payloads at the transport layer before they ever reach the target foundational models.
This transport-layer interception provides critical architectural benefits: it establishes a unified, model-agnostic security posture across the entire enterprise, reduces overall latency, and drastically lowers the compute costs associated with processing malicious or overly long prompts at the LLM provider level.

Cloudflare AI Security for Apps: Inline Execution and Presidio NER

Cloudflare's AI Security for Apps (formerly Firewall for AI) integrates LLM-specific threat detection natively within its expansive reverse-proxy WAF infrastructure. The architecture is designed to automatically discover LLM-powered application endpoints through heuristic analysis—identifying specific network patterns such as Server-Sent Events (SSE) used for model streaming, GraphQL endpoint schemas, and specific JSON response formats, separating legitimate AI traffic from health checks and standard device heartbeats.
Once an endpoint is discovered and tagged with the cf-llm label, Cloudflare intercepts all incoming HTTP POST requests bearing the application/json content type. Traditional WAFs rely almost exclusively on regular expressions (regex) to detect sensitive data. While regex executes with exceptionally low latency and is highly effective for rigidly structured data, it routinely fails to identify unstructured Personally Identifiable Information (PII) embedded deep within complex, natural-language generative AI prompts.
To resolve this limitation, Cloudflare's architecture executes the open-source Presidio framework—a sophisticated Named Entity Recognition (NER) model originally developed by Microsoft—directly on its serverless Cloudflare Workers AI platform. As the user's prompt passes through the proxy, Workers AI executes the Presidio NER model inline to perform deep contextual analysis. The model generates real-time security metadata, establishing boolean flags ("Was PII found?") and categorization tags ("What type of PII entity?") for data points like email addresses, financial details, and phone numbers.
This generated metadata is immediately routed into the Cloudflare WAF Custom Rules engine. Enterprise security teams can configure these rules to trigger terminating actions. If a prompt violates established PII constraints or registers high on the prompt injection scoring metric, the WAF executes a Block or Managed Challenge action.

Feature Category	Cloudflare Business Plan Limits	Cloudflare Enterprise Plan Limits
Max Custom Rules	100 Rules	1,000 Rules
Account-Level Rulesets	Not Supported	Supported
Regex Support	Supported	Supported
WAF Actions	Block, Managed Challenge, Skip	Block, Managed Challenge, Skip, **Log**
AI Security Features	LLM Endpoint Discovery Only	Full AI Security Log Mode & Detection Fields

Because the inspection occurs natively at the edge, a blocked payload simply returns a standard HTTP error to the client, preventing the data leak and shielding the origin LLM from expending costly GPU compute cycles on a malicious request. Furthermore, token-counting algorithms analyze prompt structure and length at the transport layer, dropping oversized

queries before they can trigger unbound consumption exploits or artificially inflate third-party API billing.
For enterprises requiring strict internal network isolation, Cloudflare executes this proxy interception via Cloudflare Tunnels. The system binds internal hostnames directly to the tunnel via a lightweight cloudflared agent. When a user queries the application, the Cloudflare Gateway resolver intercepts the DNS query and issues a temporary, synthetic IP address from a reserved Carrier-Grade NAT (CGNAT) space (e.g., 100.80.10.10). As the application traffic hits this synthetic IP, the Gateway rewrites the packet destination to the real internal private IP and routes it securely down the tunnel, enforcing Zero Trust policies without ever exposing the AI infrastructure to the public internet.

AWS Bedrock Guardrails: API Gateway Interception and DataWeave

Similar to Cloudflare's approach, Amazon Web Services provides robust edge-level moderation through AWS Bedrock Guardrails. This system acts as a model-agnostic layer of safety controls that enforces responsible AI policies across both Amazon’s native foundational models and third-party models hosted within the environment.
The architectural mechanism for enforcement relies heavily on API gateway interception. Within enterprise architectures utilizing enterprise gateways like MuleSoft's Omni Gateway, Bedrock Guardrails are deployed via a dual-phase policy mirroring the INPUT and OUTPUT scopes of the Bedrock API.
In the **Request Phase**, the gateway intercepts the inbound HTTPS payload before it reaches the LLM. The policy utilizes DataWeave expressions (a JSON-query language) to parse the payload and extract the raw user prompt. This extracted string is transmitted asynchronously to the AWS Bedrock ApplyGuardrail API via an authenticated POST request (POST /guardrail/guardrailIdentifier/version/guardrailVersion/apply). The ApplyGuardrail endpoint evaluates the text against configured policies, which include:

- **Content Filters:** Categorizing hate speech, sexual content, and violence based on Low, Medium, or High confidence thresholds.
- **Denied Topics:** A semantic filtering mechanism that utilizes natural language processing to block conceptual subjects (e.g., "investment advice" or "competitor comparisons"), preventing prompt injections aimed at forcing the model off-topic.
- **Word Filters and PII Redaction:** Masking or blocking specific strings and sensitive user data using regex and entity detection.

If the ApplyGuardrail API detects a violation, the API Gateway actively blocks the request, terminating the connection and returning an HTTP 403 Forbidden error directly to the client. The upstream LLM remains entirely isolated from the transaction.
If the prompt passes the initial screening, the LLM processes the request, triggering the **Response Phase**. The gateway intercepts the returning payload from the LLM and repeats the evaluation process against the ApplyGuardrail API to ensure the model did not generate unsafe output. To track these actions for auditability, the gateway injects specific HTTP tracking headers into the response, such as x-llm-proxy-bedrock-guardrail-action (indicating allow/reject) and x-llm-proxy-bedrock-guardrail-phase (indicating whether the block occurred during the request or response phase).
A critical architectural consideration in this deployment is the handling of systemic resilience. Gateways must be configured with specific fail-open or fail-closed behaviors regarding the ApplyGuardrail API, which has a configurable timeout between 1,000ms and 30,000ms. If the asynchronous call to the Bedrock API times out or fails (e.g., an HTTP 503 Service Unavailable

or 429 ThrottlingException), a "fail-closed" configuration will block the user's prompt by default, returning a 503 error to prioritize security over availability. Conversely, a "fail-open" configuration will allow the potentially malicious prompt to bypass the WAF and reach the LLM.
Furthermore, to combat model hallucination, the ApplyGuardrail API executes advanced contextual grounding checks. The gateway extracts two distinct fields from the payload using DataWeave: the Grounding Source Selector (typically the reference text or RAG context, e.g., #[payload.messages.content]) and the Grounding Query Selector (the user's specific prompt, e.g., #[payload.messages[-1].content]). The Bedrock engine evaluates the LLM's response against both fields, guaranteeing that the generated output is strictly derived from the provided source material and directly answers the user's query.
To facilitate compliance and cost-tracking, all interactions are logged centrally. The architecture utilizes Amazon Kinesis Data Streams and Amazon Data Firehose to stream sanitized requests, responses, guardrail metadata, and transaction tokens into Amazon S3. This data is then queried using the AWS Glue Crawler API and Amazon Athena, allowing organizations to run deep analytics and chargeback processes to attribute LLM usage costs across the enterprise securely.

Strategic Outlook and Enterprise Architecture Recommendations

The convergence of deterministic NLU processing, massive adversarial intelligence feeds, and edge-compute WAF interception represents a comprehensive maturation of enterprise AI security. The deprecation of legacy tools like the Perspective API forces enterprises to abandon rudimentary probability scoring in favor of architectures that fundamentally understand semantic context and grammatical hierarchy.
To maintain robust security postures amidst the proliferation of generative AI, enterprise architects should adopt the following structural paradigms:

1. **Decouple Moderation from the Application Layer:** Transition moderation logic out of the core application database sequence. Implement edge-compute interception using tools like Cloudflare AI Security for Apps or AWS Bedrock Guardrails coupled with API Gateways. By intercepting and dropping malicious payloads at the transport layer, enterprises protect backend infrastructure, prevent unauthorized LLM compute consumption, and centralize security policies across multi-model deployments.
2. **Transition to Deterministic, Explainable NLU:** Migrate away from opaque, black-box ML classifiers. Implement advanced NLU engines like Tisane Labs that utilize headless knowledge graphs and Finite State Automata to deliver deterministic, severity-based abuse instances. This transition ensures strict auditability, eliminates automation bias, and guarantees compliance with emerging digital safety regulations worldwide.
3. **Integrate Continuous Adversarial Intelligence:** AI models are static without continuous updates. Architectures must ingest real-time threat intelligence feeds from platforms like Alice (Rabbit Hole), CyberWell, and the ADL. This ensures that custom WAF rules, Denied Topics configurations, and NLU semantic maps are instantly updated to recognize emerging algorithmic manipulation, zero-day vulnerabilities, and coded "Algospeak."
4. **Enforce Cryptographic Grounding for High-Risk Data:** For enterprises utilizing AI to process highly sensitive or historical data, statistical grounding is insufficient. Implement Web 3.0 cryptographic verification frameworks, such as those developed by the Starling Lab, to hash and anchor reference data across decentralized networks. Guardrails must

be configured to cross-reference LLM outputs against these immutable Content Identifiers (CIDs) to categorically prevent the generation of synthetic deepfakes and hallucinated disinformation at the protocol level.

By synthesizing transport-layer inline WAF interception with deterministic NLU engines fueled by specialized NGO intelligence, enterprises can construct an AI safety architecture capable of scaling securely against the next generation of adversarial threats.

Works cited

1. Goodbye, Perspective API - Tisane Labs - Medium, https://medium.com/tisanelabs/goodbye-perspective-api-79da0f237b3f 2. Perspective API Retirement: What to Do Next | Sence, https://www.makesence.com/blog/perspective-api-retirement-what-to-do-next-sence 3. Migrating from Two Hat Community Sift to Tisane API, https://docs.tisane.ai/guides/how-tos/communitysiftmigration 4. Take control of public AI application security with Cloudflare's Firewall for AI, https://blog.cloudflare.com/take-control-of-public-ai-application-security-with-cloudflare-firewall-for-ai/ 5. Amazon Bedrock Guardrails Policy - MuleSoft Documentation, https://docs.mulesoft.com/gateway/latest/policies-included-bedrock-guardrails 6. Floor Talk - New York Stock Exchange, https://tv.nyse.com/floor-talk/season:4?html=1&page=11 7. Safety by Design for LLMs - Medium, https://medium.com/engineering-activefence/safety-by-design-for-llms-b959df9066ff 8. The "Secret Sauce" Protecting the Internet is Now Securing AI: ActiveFence is now Alice, https://www.prnewswire.com/news-releases/the-secret-sauce-protecting-the-internet-is-now-securing-ai-activefence-is-now-alice-302660532.html 9. AI-Generated Antisemitism - CyberWell, https://cyberwell.org/reports/ai-generated-antisemitism/ 10. RESEARCH STUDY - European Institute for Counter Terrorism and Conflict Prevention, https://eictp.eu/wp-content/uploads/2024/05/EICTP_Research_Papers_Antisemitism_FINAL.pdf 11. Dimensions in Testimony | CMHR, https://humanrights.ca/exhibition/dimensions-in-testimony 12. waf - Noise, https://noise.getoto.net/tag/waf/ 13. Migrating from Perspective API to Tisane API, https://docs.tisane.ai/guides/how-tos/perspectivemigration 14. Guides | Tisane Developer Documentation, https://docs.tisane.ai/guides 15. Supported Functions - Tisane Developer Documentation, https://docs.tisane.ai/guides/features/functionality 16. A Different Approach to Text Moderation | by Vadim Berman | Tisane Labs - Medium, https://medium.com/tisanelabs/a-different-approach-to-text-moderation-d69ac62a250a 17. T&S Platform - Content Moderation Platform for Online Safety - Tremau, https://tremau.com/platform/ 18. Technology - Tisane Labs, https://tisane.ai/technology 19. Compare Tisane vs. Utopia AI Moderator in 2026, https://slashdot.org/software/comparison/Tisane-vs-Utopia-AI-Moderator/ 20. LLM Snapshot Architecture. Blueprint for Deterministic, Scalable… | by Vadim Berman | Tisane Labs | Medium, https://medium.com/tisanelabs/llm-snapshot-architecture-cb21b1c10557 21. Overview - Tisane Developer Documentation, https://docs.tisane.ai/sdks 22. Alice | Secure, Safe, & Trustworthy AI, https://alice.io/ 23. Intelligence | AI Security, Safety & Trust Ecosystem - Alice, https://alice.io/intelligence 24. Alice Documentation, https://docs.alice.io/ 25. Alice Reviews, Ratings & Features 2026 | Gartner Peer Insights, https://www.gartner.com/reviews/market/ai-security-testing/vendor/alice 26. From zero to data lake: our journey to handling data at scale | by Alice (formerly: ActiveFence) | Engineering @ Alice | Medium, https://medium.com/engineering-activefence/from-zero-to-data-lake-our-journey-to-handling-dat

a-at-scale-6582dacdc356 27. ActiveFence Integration — NVIDIA NeMo Guardrails Library Developer Guide, https://docs.nvidia.com/nemo/guardrails/latest/configure-rails/guardrail-catalog/community/active-fence.html 28. Statement of Kerry Sleeper Deputy Director, Intelligence and Information Sharing Secure Community Network (SCN), https://homeland.house.gov/wp-content/uploads/2025/06/2025-06-11-CTI-HRG-Testimony.pdf 29. Since Musk Takeover, Data Shows Worrying Signs About Antisemitism on 'X', https://www.algemeiner.com/2023/08/18/since-musk-takeover-data-shows-worrying-signs-about-antisemitism-on-x/ 30. Technology Innovation Leadership - Frost & Sullivan, https://www.frost.com/wp-content/uploads/2022/01/ActiveFence-Award-Recognition.pdf 31. From the Frontlines: How Artificial Intelligence Can Both Spread and Fight Hate - Apple Podcasts, https://podcasts.apple.com/us/podcast/from-the-frontlines-how-artificial-intelligence-can/id1490916352?i=1000719357485 32. Modes of toxic behavior and game design considerations in online multiplayer games - First Monday, https://firstmonday.org/ojs/index.php/fm/article/download/13851/12035/91074 33. THE ROLE OF ANTISEMITISM IN THE MOBILIZATION TO VIOLENCE BY EXTREMIST AND TERRORIST ACTORS - Counter Extremism Project, https://www.counterextremism.com/sites/default/files/2025-04/CEP%20Transnational%20Antisemitism%20Study%202025.pdf 34. THE WEAPONIZATION OF ARTIFICIAL INTELLIGENCE THE NEXT STAGE OF TERRORISM AND WARFARE - COE-DAT, https://www.tmmm.tsk.tr/publication/researches/21-TheWeaponizationofAI-TheNextStageofTerrorismandWarfare.pdf 35. NeMo Guardrails: A Toolkit for Controllable and Safe LLM Applications with Programmable Rails | Request PDF - ResearchGate, https://www.researchgate.net/publication/376401604_NeMo_Guardrails_A_Toolkit_for_Controllable_and_Safe_LLM_Applications_with_Programmable_Rails 36. “Can AI make Hitler cry?” exploring the use of AI in Holocaust education across four generations - ResearchGate, https://www.researchgate.net/publication/394538599_Can_AI_make_Hitler_cry_exploring_the_use_of_AI_in_Holocaust_education_across_four_generations 37. Safeguard generative AI applications with Amazon Bedrock Guardrails | Artificial Intelligence, https://aws.amazon.com/blogs/machine-learning/safeguard-generative-ai-applications-with-amazon-bedrock-guardrails/ 38. Speaking for the Past - Real Life Mag, https://reallifemag.com/speaking-for-the-past/ 39. Starling Lab: Establishing Trust for Humanity's Data - Filecoin, https://www.filecoin.io/blog/starling-lab-establishing-trust-for-humanity-s-data 40. The Starling Lab for Data Integrity Announces Inaugural Starling Journalism Fellows, https://sfi.usc.edu/news/2022/03/32906-starling-lab-data-integrity-announces-inaugural-starling-journalism-fellows 41. Different Installments of the Interactive Testimony of Eva Mozes Kor - Open Access LMU, https://epub.ub.uni-muenchen.de/124336/1/Different_Installments_of_the_Interactive_Testimony_of_Eva_Mozes_Kor.pdf 42. Dimensions in Testimony - USC Shoah Foundation, https://sfi.usc.edu/dit 43. Obsolescence, Forgotten: ``Survivor Holograms'', Virtual Reality, and the Future of Holocaust Commemoration - Cinergie, https://cinergie.unibo.it/article/download/12205/12999/49005 44. AI Definitions in Flux: Authenticity and Holocaust Testimony in Focus - Culture Unbound, https://cultureunbound.ep.liu.se/article/view/5702/5382 45. Press - Visitors can now help develop eyewitness testimonies for the future. - DNB, https://www.dnb.de/EN/Ueber-uns/Presse/ArchivPM2023/20231030BetatestIngeAuerbacher.html 46. This Holocaust survivor from Montreal will have her testimony preserved thanks to

hologram technology - The Azrieli Foundation, https://azrielifoundation.org/media/this-holocaust-survivor-from-montreal-will-have-her-testimony-preserved-thanks-to-hologram-technology/ 47. AI Meets Educational Memory Work: Risks, Potential and Possible Consequences, https://www.stiftung-evz.de/en/what-we-support/education-agenda-ns-injustice/magazine-of-the-education-agenda-ns-injustice-2024/ai-meets-educational-memory-work/ 48. restored index page – The Starling Lab for Data Integrity, https://www.starlinglab.org/restored-index-page/ 49. The Starling Lab Framework | FFDW - Filecoin Foundation for the Decentralized Web, https://ffdweb.org/blog/the-starling-lab-framework 50. Starling Lab: Establishing Trust in the Digital Records of Human History with the Starling Framework for Data Integrity - Filecoin, https://filecoin.io/assets/case-studies/case-study-starling-lab.pdf 51. Amazon Bedrock Guardrails: Securing Enterprise GenAI Applications and AI Agents - Akto, https://www.akto.io/blog/aws-bedrock-guardrails 52. Tisane Block: NLP and Translation, 27 Languages - PubNub, https://www.pubnub.com/blog/announcing-the-tisane-block-for-natural-language-processing-in-27-languages/ 53. Cloudflare Launches the Most Complete Platform to Deploy Fast, Secure, Compliant AI Inference at Scale - Business Wire, https://www.businesswire.com/news/home/20230927225198/en/Cloudflare-Launches-the-Most-Complete-Platform-to-Deploy-Fast-Secure-Compliant-AI-Inference-at-Scale 54. Zero-Trust - Noise, https://noise.getoto.net/tag/zero-trust/ 55. Cloudflare Gateway - Noise, https://noise.getoto.net/tag/cloudflare-gateway/