Semantic Security Gateway Firewall

(SSGF): A Deterministic Hybrid Architecture for AI Input Security

Whitepaper
Author: Ernesto Rosati, SSGF Architect
Date: February 3, 2026
Location: Tijuana, MX

Contact: [ernesto.rosati@gmail.com, ernesto.rosati@yahoo.com]

Cel: +52 664 229 4736

Executive Summary

The Semantic Security Gateway Firewall (SSGF) is a deterministic semantic security infrastructure designed to protect and optimize AI systems by filtering ambiguity, manipulation, and low-signal inputs **before** they reach expensive reasoning models.

Modern Large Language Model (LLM) deployments suffer from systemic inefficiencies: high operational costs, probabilistic uncertainty, susceptibility to semantic manipulation, and limited auditability. In most real-world environments, **70–90% of inputs do not require deep reasoning**, yet are processed as such, resulting in unnecessary token consumption, latency, and security exposure.

SSGF addresses this problem by introducing a **multi-layer semantic decision pipeline** that prioritizes deterministic, rule-based inspection before probabilistic inference. By applying local structural analysis, entropy scoring, and intention heuristics, SSGF resolves the majority of traffic at the edge, escalating only genuinely ambiguous cases for deeper semantic evaluation.

This approach enables:

- 70–90% reduction in LLM API usage
- Sub-5ms latency for most decisions
- Deterministic, auditable security enforcement
- Vendor-independent, on-premise or hybrid deployment

SSGF transforms AI security from probabilistic moderation into **founded determinism**, where decisions are based on verifiable structure and evidence rather than model opinion.

This makes it suitable for enterprise, government, regulated environments, and any system where clarity, cost control, and auditability are critical.

This document presents the architecture, design principles, benchmarks, and implementation model of SSGF as a reusable semantic security infrastructure.

1. Introduction

Large Language Models (LLMs) have enabled powerful new applications across customer service, analytics, automation, and decision support. However, as these systems scale, a fundamental inefficiency becomes apparent: **most inputs processed by LLMs do not require deep reasoning**, yet they are treated as if they do.

In real-world deployments, AI systems are exposed to:

- spam and automated noise
- phishing and social engineering attempts
- prompt injection and coercive language
- ambiguous or low-signal queries
- structurally valid but semantically risky requests

Processing all such inputs through probabilistic reasoning models leads to:

- unnecessary token consumption
- increased latency
- inconsistent or non-auditable decisions
- heightened security risk

SSGF is designed to address this problem at the architectural level.

---

2. The Core Problem: Probabilistic AI at the Input Layer

Most AI security and moderation solutions operate **after** the model has already processed the input, relying on:

- post-generation filtering

- probabilistic classifiers
- model “judgment” of intent

This approach has three systemic weaknesses:

1. **Cost Inefficiency**
   Every input—regardless of quality or risk—is processed by an LLM.
2. **Uncertainty and Hallucination**
   Probabilistic systems may produce different decisions for identical inputs.
3. **Lack of Auditability**
   Model decisions are difficult to explain, reproduce, or justify in regulated environments.

SSGF approaches the problem differently: **by enforcing structure and determinism before probabilistic reasoning is allowed**.

---

3. Design Principles: Founded Determinism

SSGF is built on the principle of **Founded Determinism**:

> Security decisions should be based on verifiable structure and evidence, not model opinion.

This means:

- identical inputs yield identical outcomes
- decision logic is explicit and inspectable
- ambiguity is treated as a signal, not a failure
- probabilistic reasoning is used only when necessary

Entropy, structure, and intent are evaluated **before** reasoning.

---

4. High-Level Architecture

SSGF operates as a **semantic security gateway** between users and AI systems.

```
Client / User
      |
      ▼
SSGF Gateway
      |
      ├── FAST Pipeline (Deterministic, Local)
      |      ├ Normalization
      |      ├ Structural & Keyword Triggers
      |      ├ Entropy Scoring
      |      ├ Intention Heuristics
      |      └ Decision: ALLOW | WARN | BLOCK | ESCALATE
      |
      └── DEEP Pipeline (Conditional)
             ├ Controlled Semantic Inspection
             ├ Hard Security Rules
             └ Final Decision Override
      |
      ▼
LLM / Backend System
```

---

5. The FAST Pipeline: Deterministic Triage

The FAST pipeline processes **all inputs locally**, without external API calls.

Its purpose is to:

- eliminate obvious noise
- block clear violations
- identify ambiguity early

Components

- **Normalization**
  Canonicalization of input text (case, spacing, Unicode, punctuation).
- **Hard Triggers**
  Deterministic pattern detection for known risks (e.g. OTP requests, credential extraction).
- **Entropy Scoring**
  Structural entropy is calculated to detect abnormal density of urgency, manipulation, or coercion.
- **Intention Heuristics**
  Lightweight rules classify likely intent categories (benign, spam, phishing, unknown).

Performance

- Typical latency: **<1–5 ms**
- No LLM calls
- Fully auditable

The FAST pipeline resolves **85–90% of traffic** without escalation.

---

6. The DEEP Pipeline: Controlled Semantic Inspection

When FAST determines that an input is **ambiguous but not clearly malicious**, it is escalated to the DEEP pipeline.

Key characteristics:

- triggered only by ambiguity
- uses low-cost or local models when possible

- constrained prompts and structured outputs
- hard security overrides enforced

The DEEP pipeline **does not replace FAST**; it only refines decisions where deterministic rules alone are insufficient.

---

7. Decision Model and Outputs

Every input results in a structured decision:

- **ALLOW** – safe to forward
- **WARN** – suspicious, requires caution or user confirmation
- **BLOCK** – policy violation
- **ESCALATE** – forwarded to DEEP pipeline

Each decision includes:

- entropy score
- triggered flags
- intention classification
- confidence (if applicable)

All outputs are JSON-first and loggable.

---

8. Benchmarks and Evaluation

Benchmark Dataset:

- 30 curated test cases
- phishing, credential extraction, benign education, urgency-based fraud

Pipeline:

- FAST + DEEP hybrid

Results (v1.0.0)

Metric	Result
Accuracy	93.3%
False BLOCK → ALLOW	0
False ALLOW → BLOCK	0
FAST latency	<5 ms
DEEP latency	300–800 ms
LLM call reduction	70–90%

These benchmarks are reproducible using the public prompt bench tools.

---

9. Security and Governance Implications

SSGF enables:

- deterministic enforcement of security policy
- reproducible audits
- consistent behavior across environments
- vendor-independent deployment

Because decisions are rule-based and logged, SSGF is suitable for:

- regulated industries
- government systems
- enterprise AI governance
- on-premise or edge deployments

---

10. Use Cases

- AI gateways and middleware
- Chatbot security
- Phishing and fraud prevention
- Prompt injection mitigation
- Edge or bandwidth-constrained AI systems
- Auditable AI deployments

SSGF is intentionally **not** a chatbot, moderation platform, or content censorship tool.

---

11. Limitations and Scope

SSGF does not attempt to:

- reason about truthfulness of content
- replace human judgment
- enforce ideological or political constraints

Its role is **structural and security-focused**, not semantic interpretation beyond risk assessment.

---

12. Conclusion

SSGF reframes AI security as an architectural problem rather than a modeling problem.

By enforcing deterministic structure at the input layer, it:

- reduces cost
- improves latency
- increases auditability
- limits semantic attack surface

Probabilistic AI remains powerful — but only when preceded by structure.