BROADCOM

WHITE PAPER

AI and Layer7™ API Gateway

Navigating Security Challenges in LLM-Driven Policy Automation

WHITE PAPER

AI and Layer7 API Gateway

Navigating Security Challenges in LLM-Driven Policy Automation

BROADCOM

TABLE OF CONTENTS

Introduction

Customers adopting a large language model (LLM) to facilitate their tasks are enthusiastic about the prospect of using LLM for configuring API security rules in their Layer7™ API Gateway. Productivity gains are starting to emerge in other areas of their practice, and the same productivity gains should be applicable to Layer7. The appeal of agentic AI systems, AI that can autonomously generate, configure, and potentially deploy security policies, promises significant efficiency improvements. However, our conversations reveal that critical security considerations have not been fully addressed. This document examines four core challenges that organizations must navigate before deploying LLM-backed policy automation in security-critical infrastructure.

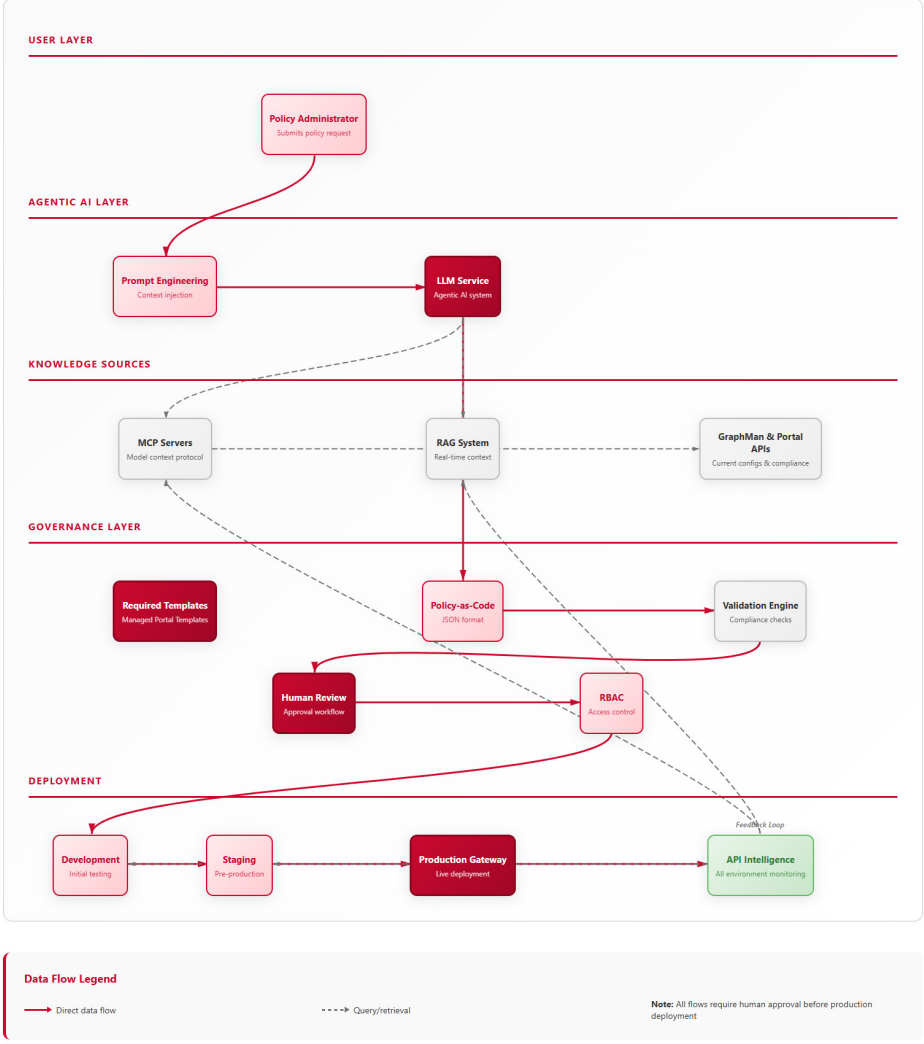
The intended audience for this document are individuals that are concerned about integrating LLMs while configuring their API security rules, platform leaders, governance teams, and technical decision-makers responsible for managing Layer7 API Gateways. The following architecture diagram shows how agentic AI capabilities integrate with governance controls and the human oversight mechanisms to enable the safe generation of policies.

Context and Prompt Engineering

Context and prompt engineering are crucial because LLMs operate on probabilities. When customers request an LLM to create a policy for the analytics API, the results will vary, but if specific requirements are provided, (for example, create an OAuth2 authentication policy with rate limiting for the analytics API requiring external partner support) the results are different. The specificity of the request, the framing of requirements, and the relevant context all influence the response. This influence mirrors the discipline developers have learned with code generation tools. Yet, security policy differs in one critical respect; errors in generated code are typically caught through testing and deployment processes, where policy misconfigurations can silently create security gaps.

From an architectural perspective, adopting prompt engineering also helps to reduce potential attack surfaces. Well-structured, validated inputs are foundational to any security boundary around LLM systems, reducing the risk of prompt injection and ensuring LLM-generated policies conform to organizational security patterns.

LLM-Backed API Gateway Architecture



Organizations that begin to explore LLM integration often underestimate the skill required to generate consistent, production-ready outputs. Teams would need to invest in understanding how to structure requests and what context to provide. This effort represents a capability gap that organizations have not yet fully acknowledged.

Beyond how the request is framed lies a deeper problem; even well-engineered prompts fail if the LLM lacks access to accurate information. This issue leads directly to the second challenge.

Knowledge Source Validation and Real-Time Policy Context

LLMs trained on public Internet data lack access to proprietary security patterns, organization-specific compliance requirements, and current deployment configurations. Their training data has a cutoff date, which means they are inherently unaware of emerging security threats, recently published best practices, or the latest capabilities and features within the Layer7 solution itself.

This LLM training requires architectural choices about what data sources LLMs can access. Research indicates that introducing malicious guidance into knowledge sources can cause LLMs to return attacker-chosen answers with high success rates. Broadcom is exploring how Model Context Protocol (MCP) servers and retrieval-augmented generation (RAG) approaches can provide LLMs with real-time information about deployed Layer7 API Gateway configurations through Graphman and Portal APIs. This approach enables LLMs to understand the current infrastructure state before making recommendations. By basing policy suggestions on live deployment data rather than generic public patterns, this approach minimizes the risk of generated policies conflicting with existing controls or missing critical environment-specific requirements. These approaches show an architectural pattern that customers could implement to ground LLM outputs in organizational reality rather than generic patterns. Broadcom is evaluating where to incorporate these capabilities into the Layer7 platform. Organizations beginning LLM integration should establish trusted policy sources using portal governance mechanisms while this evaluation is in progress.

API Intelligence will feed operational insights back to both RAG systems and MCP servers, creating a continuous learning cycle that enriches the contextual knowledge available for future policy generation. This feedback loop ensures that policy recommendations benefit from real-world deployment patterns and operational experience across all environments.

From an architectural perspective, designing boundaries for knowledge sources is critical. Architects should define which systems can access what trusted policy data, how that data is verified for integrity, and what actions are taken if a source is compromised. This threat modeling around knowledge sources becomes foundational when LLMs consume external data to inform policy recommendations.

Non-Deterministic Behavior and Deployment Governance

Yet even with access to accurate, current information, a third challenge emerges. LLMs fundamentally cannot guarantee consistent outputs, which creates operational risks in policy generation.

LLMs produce different outputs from identical inputs even with optimal temperature settings. This behavior creates configuration drift and complicates reproducibility requirements that regulatory frameworks mandate. For security policy, this means that identical requirements could generate different policy outputs across runs, making audit trails unreliable for compliance reviews that require reproducible security decisions.

The Layer7 Developer Portal provides governance mechanisms to address challenges related to non-deterministic behavior and deployment. Organizations can enhance their existing deployment governance by defining required deployment templates. These templates are managed through the Layer7 Developer Portal, specifically encapsulated assertions. These assertions enforce specific security controls such as authentication, routing, audit logging and compliance validation. Any API must satisfy these controls before deployment. These customer-designed templates would serve as guardrails. An agentic AI system could generate a policy draft, but deployment governance would ensure that all required templates are present and properly configured. This approach prevents fully autonomous deployment, as agentic AI systems generate candidates, governance constraints validate completeness, and human oversight is required before production.

From an architectural perspective, how templates are designed directly impacts the generation of safe policies. Templates should encapsulate security logic in a way that makes policy constraints explicit and enforceable. Complex assertion logic lives within the template, while LLM-generated policies reference and configure those templates rather than constructing security mechanisms from scratch. This architectural separation ensures that fundamental security decisions remain within expert-designed components while agentic AI operates at the composition and configuration layer.

Additionally, Broadcom is developing API Intelligence capabilities that will enable organizations to build configuration rules displaying deployed gateway configurations and policy usage patterns across all environments. This capability will create an auditable view of what policies actually do and detect policy deviations before they create incidents, providing comprehensive visibility from development through production environments.

Complementing this development, Broadcom is also seeing customers start to embrace policy-as-code capabilities that will enable gateway policies to be represented in JSON format. This structured, machine-readable representation will enable LLMs to generate policies more reliably, support efficient version control and change tracking within repositories, and maintain consistency with organizational governance standards. When combined with required templates and approval workflows, policy-as-code capabilities will provide the foundational infrastructure necessary for safe LLM-assisted policy generation.

Human Oversight and Validation Controls

Given these technical constraints, including inconsistency, knowledge gaps, and the need for governance, the fourth challenge involves establishing human oversight. This oversight is a critical component of governance, necessary to manage risk.

NIST guidance, EU regulatory requirements, and NSA recommendations converge on a single point; consequential security decisions must include human oversight. While agentic AI systems show impressive autonomous capabilities, regulatory frameworks explicitly require human judgment for security-critical decisions. Many organizations have approval processes in place, but should revisit or formalize these decisions as they consider LLM integration. Clarifying who approves policies, at what point in the workflow, and with what authority becomes essential when introducing AI-generated recommendations. Organizations cannot delegate threat modeling validation, compliance verification, or edge case handling to agentic AI systems.

The Layer7 Developer Portal supports this work through role-based API deployment governance. Policies generated by LLMs flow through staging environments, where they undergo validation against organizational templates and compliance requirements. Only users with appropriate authority approve production deployment. Version control integration tracks changes, enabling rollback when issues arise. This governance separates policy generation from deployment permissions, because LLM service accounts never have direct access to production.

From an architectural perspective, approval workflows should be designed as security controls themselves, not merely administrative steps. Architects should enforce clear separation between policy generation systems (LLM service accounts with limited privileges) and policy deployment systems (human users with explicit authority). Role-based access controls, audit logging, and exception handling for policy anomalies all become architectural boundaries that ensure consequential decisions remain under human control.

Conclusion

Configuring Layer7 API security rules with LLMs can be viable when appropriate safeguards are in place. Organizations should begin by establishing portal-based governance with required templates, creating role-based approval workflows for human review, and preparing to utilize API intelligence for compliance monitoring as these capabilities mature. Successful implementation requires architects to design these controls as security boundaries. Organizations establishing these foundations while starting in development and staging environments with extensive human validation will realize productivity gains safely. Those deploying agentic AI capabilities without these guardrails could encounter expensive operational failures and expose organizations to significant reputational and financial risks.

The architecture described in this document demonstrates how agentic AI capabilities can be safely integrated within a governance framework that maintains human oversight, validates against organizational standards, and provides audit trails for compliance. This balanced approach enables organizations to capture the productivity benefits of LLM-assisted policy generation while mitigating the inherent risks associated with autonomous systems in security-critical infrastructure.

References

NIST AI Guidance and Standards

- NIST AI Risk Management Framework (AI RMF) 1.0
- NIST AI 600-1: Artificial Intelligence Risk Management Framework - Generative AI Profile
- NIST SP 800-218A: Secure Software Development Practices for Generative AI and Dual-Use Foundation Models
- NIST Trustworthy and Responsible AI Resource

EU AI Act

- EU Artificial Intelligence Act (Regulation (EU) 2024/1689)
- Article 14: Human Oversight Requirements
- Article 26: Obligations of Deployers of High-Risk AI Systems

NSA AI Security Guidance

- Deploying AI Systems Securely - Best Practices for Deploying Secure and Resilient AI Systems
- AI Data Security: Best Practices for Securing Data Used to Train & Operate AI Systems

Research on LLM Security Vulnerabilities

- Enhancing Prompt Injection Attacks to LLMs via Poisoning Alignment
- PoisonedRAG: Knowledge Corruption Attacks to Retrieval-Augmented Generation of Large Language Models
- Human-Imperceptible Retrieval Poisoning Attacks in LLM-Powered Applications
- System Prompt Poisoning: Persistent Attacks on Large Language Models Beyond User Injection
- From Prompt Injections to Protocol Exploits: Threats in LLM-Powered AI Agents Workflows

BROADCOM
connecting everything®

For more information, visit our website at: www.broadcom.com

AI-DPA-WP100 November 11, 2025