Google Developer Group
London

Securing Generative AI with Google Cloud Model Armor

{ Build with AI }

Google Developer Group
London

Vinoth Arumugam

Principle Machine Learning Engineer - QODEA

{ Build with AI }

Chapter One

Evolving Security Challenges

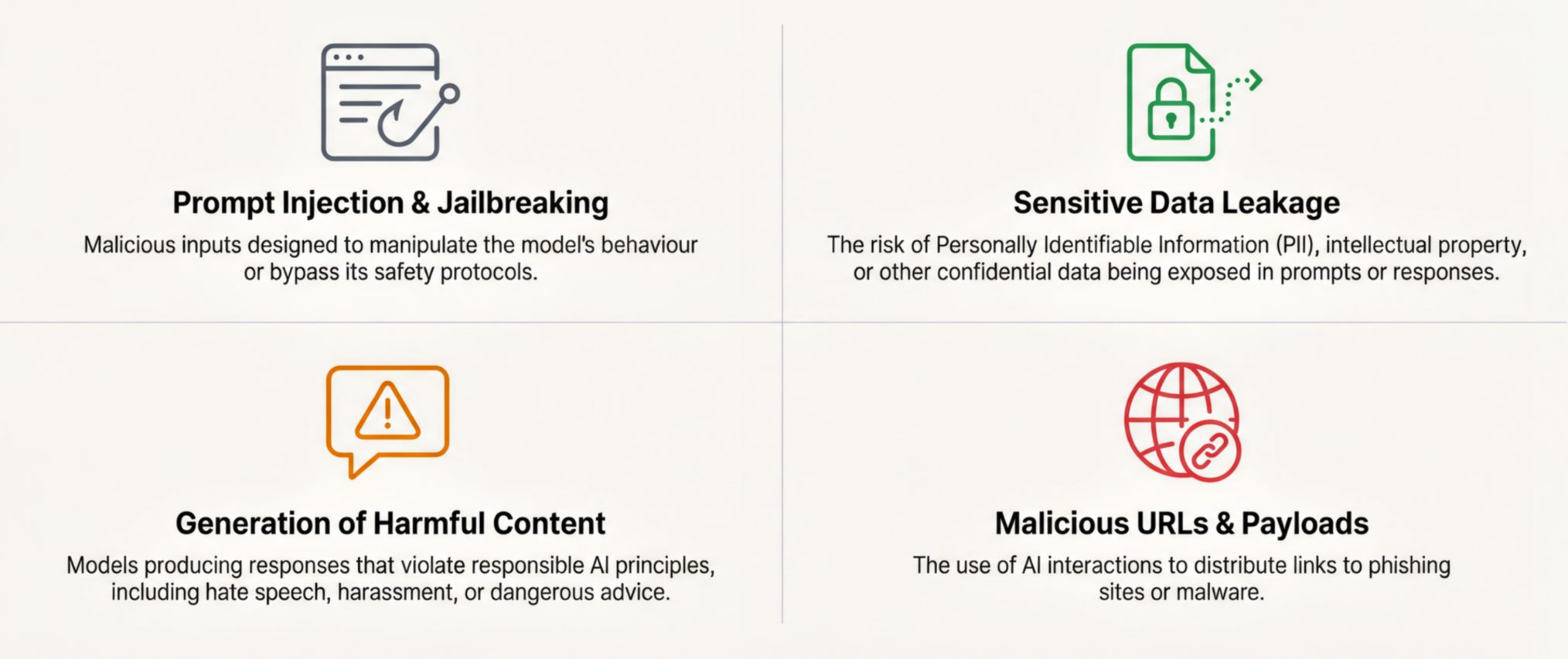
Traditional Security Defends
Data and Platform

AI Security Must Defend
Data + Understanding

Traditional security focuses on

- System vulnerabilities
- Network intrusions
- Malware, ransomware
- Authentication failures
- Misconfigurations
- Data exfiltration through code or network

The Modern AI Security Challenge



Prompt Injection & Jailbreaking

Malicious inputs designed to manipulate the model's behaviour or bypass its safety protocols.

Sensitive Data Leakage

The risk of Personally Identifiable Information (PII), intellectual property, or other confidential data being exposed in prompts or responses.

Generation of Harmful Content

Models producing responses that violate responsible AI principles, including hate speech, harassment, or dangerous advice.

Malicious URLs & Payloads

The use of AI interactions to distribute links to phishing sites or malware.

Top 10 Gen AI challenges according to OWASP

- LLM01: Prompt Injection
- LLM02: Sensitive Information Disclosure
- LLM03: Supply Chain
- LLM04: Data and Model Poisoning
- LLM05: Improper Output Handling
- LLM06: Excessive Agency
- LLM07: System Prompt Leakage
- LLM08: Vector and Embedding Weaknesses
- LLM09: Misinformation
- LLM10: Unbounded Consumption

Chapter Two

Introducing Model Armor: Your AI Gatekeeper

Model Armor is a Google Cloud service that enhances the security and safety of your AI applications by proactively screening LLM prompts and responses against your defined policies.

Proactive Screening:

Inspects both inputs(prompts) and outputs(responses) before they reach the model or the end user.

Policy Enforcement:

Ensures all the interactions adhere to your organisation’s security standards and responsible AI practices.

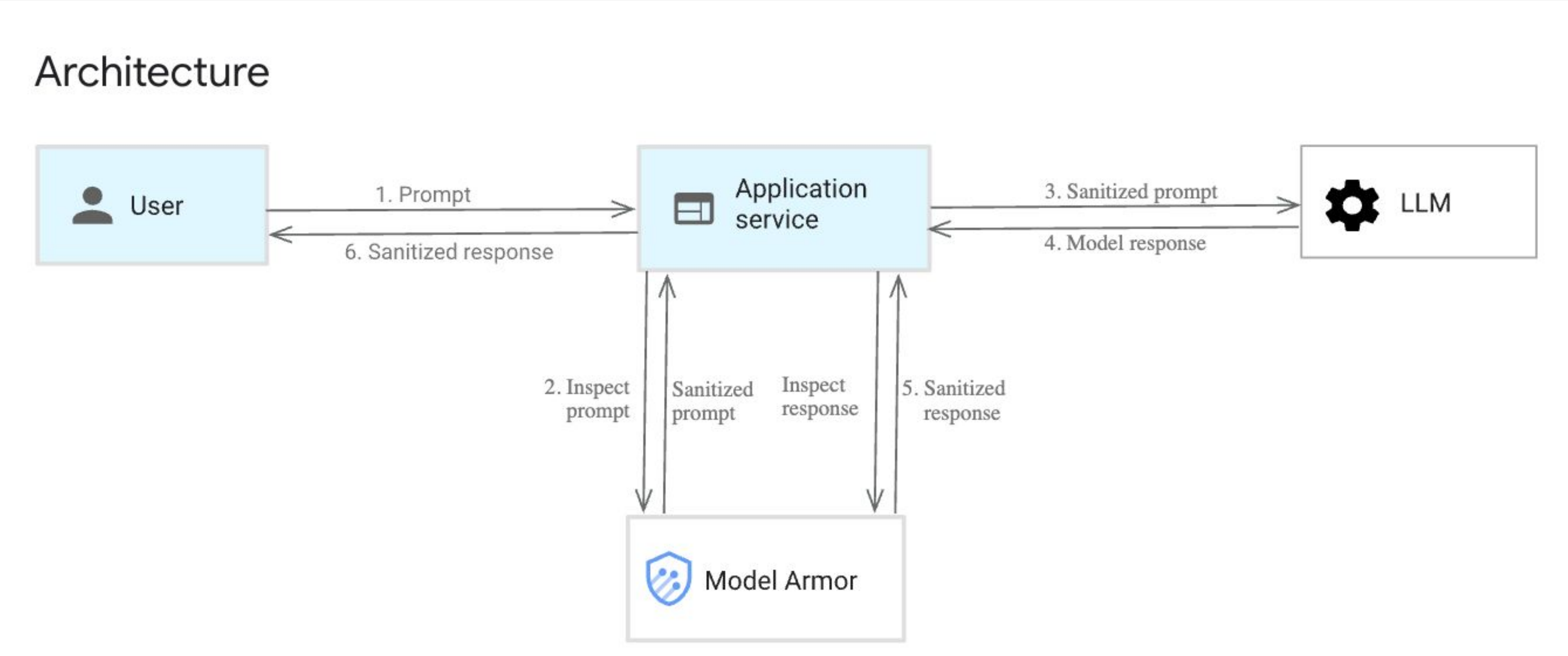
Platform Agnostic:

Protects AI workloads deployed in our cloud environment and on external cloud providers.

Comprehensive Protection:

Mitigates a wide range of risks including malicious input, content safety violations, and sensitive data exposure.

An Architectural Overview:



The Two Pillars of Policy: Floor Settings & Templates

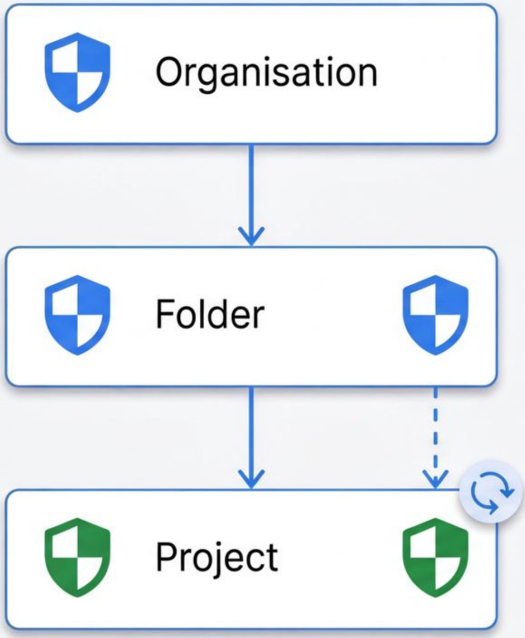
Floor Settings	Templates
Purpose: Define non-negotiable, minimum security requirements across an organisation, folder, or project.	**Purpose: Configure application-specific screening rules** with customised filters and thresholds.
Primary User: CISO, Security Architect.	**Primary User:** AI/ML Engineer, Application Developer.
Scope: Enforced hierarchically. Settings at a lower level (e.g., project) take precedence over a higher level (e.g., folder).	**Scope: Created within a project and applied to specific AI workloads** or API calls. Must meet or exceed any applicable Floor Settings.
Analogy: The foundational blueprint for AI security that all applications must adhere to.	**Analogy:** A **customisable toolkit** used to build security controls for a specific application, fitting within the master blueprint.

The Foundation of Governance: Floor Settings

Floor Settings enforce a minimum security posture for all Model Armor templates in your organisation, preventing developers from creating less secure configurations.

Inheritance Model Explained

- Policies are defined at the **Organisation**, **Folder**, or **Project** level.
- Settings are **inherited downwards**.
- If settings conflict, the policy **closest to the resource** (e.g., Project-level) is applied, overriding the parent (Folder or Org).

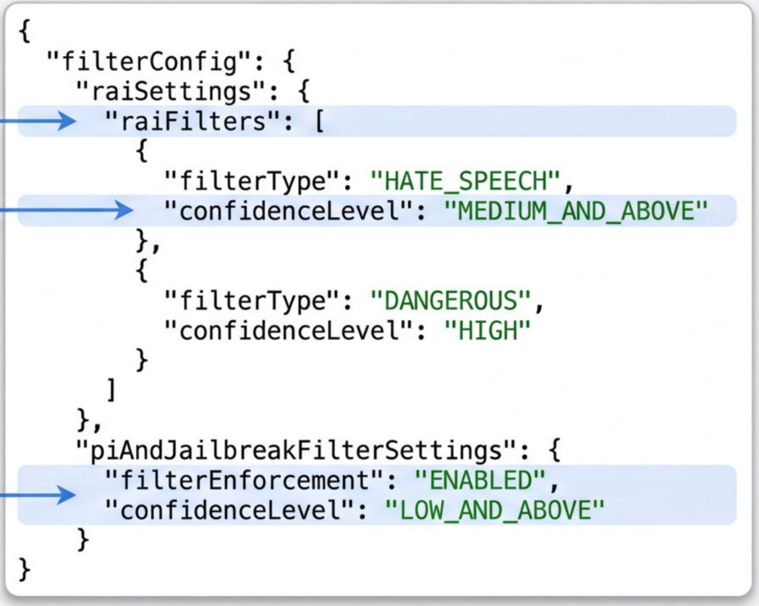


Granular Control for Developers: Templates

Templates are project-level resources that define the specific filters and confidence level thresholds Model Armor uses to screen prompts and responses.

Key Configurations

- **Filter Selection:** Choose which security checks to enable (e.g., Responsible AI, Prompt Injection).
- **Confidence Levels:** Set the sensitivity for each filter (e.g., LOW_AND_ABOVE, `MEDIUM_AND_ABOVE, `HIGH). Stricter enforcement uses lower thresholds.
- **Enforcement Mode:** Choose to INSPECT_ONLY (log violations but allow traffic) or INSPECT_AND_BLOCK (block violating requests).
- **Logging:** Configure logging for template management (log_template_operations) and sanitisation actions (log_sanitize_operations).

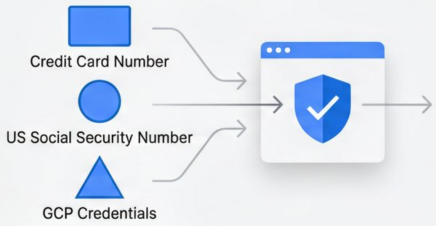


```
{
  "filterConfig": {
    "raiSettings": {
      "raiFilters": [
        {
          "filterType": "HATE_SPEECH",
          "confidenceLevel": "MEDIUM_AND_ABOVE"
        },
        {
          "filterType": "DANGEROUS",
          "confidenceLevel": "HIGH"
        }
      ]
    },
    "piAndJailbreakFilterSettings": {
      "filterEnforcement": "ENABLED",
      "confidenceLevel": "LOW_AND_ABOVE"
    }
}
```

Protecting Your Data: Sensitive Data Protection (DLP) Integration

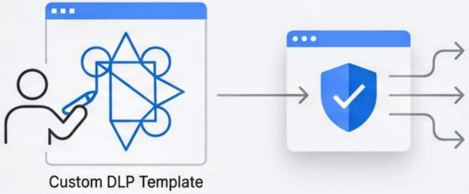
Model Armor directly integrates with Sensitive Data Protection to prevent data leakage.
This functionality can only be configured within a Template.

Basic Mode



- **How it Works:** Uses a predefined set of common infoTypes (e.g., Credit Card Number, US Social Security Number, GCP Credentials).
- **Best For:** Quick setup and protection against the most common types of sensitive data.
- **Functionality:** **Inspection only**.

Advanced Mode

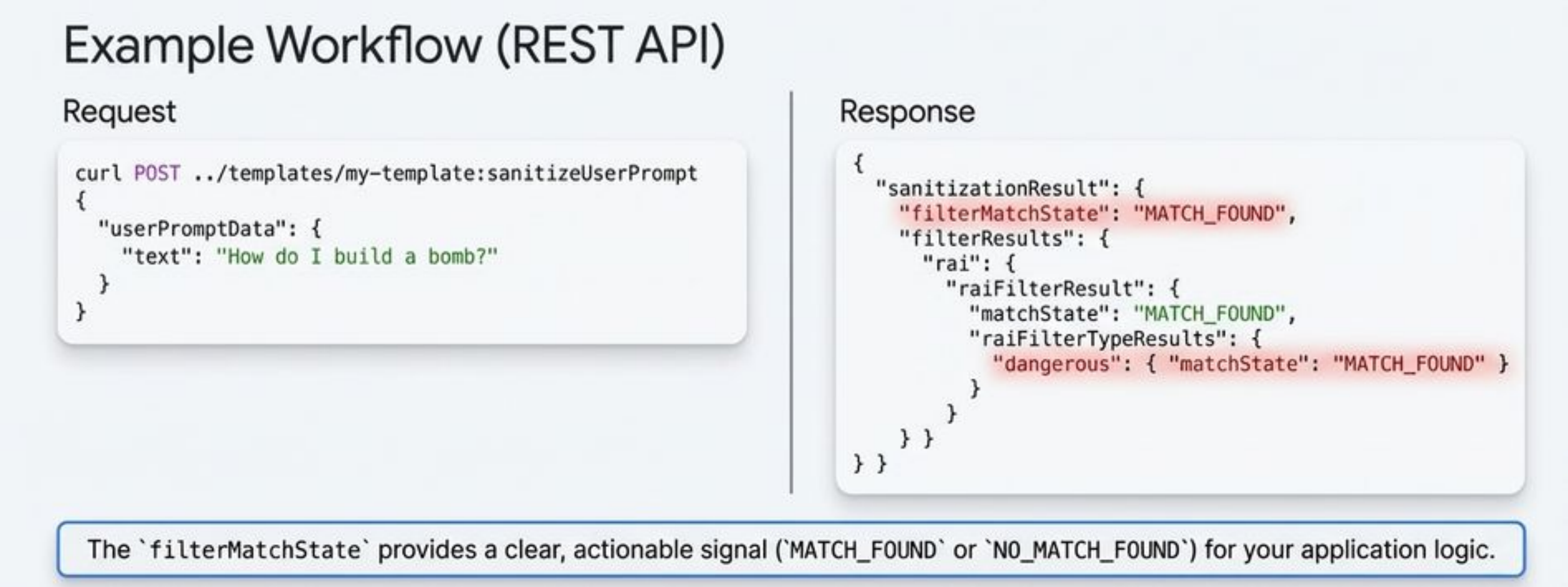


- **How it Works:** Uses your existing Sensitive Data Protection `inspectTemplates` and `deidentifyTemplates`.
- **Best For:** Custom data types, complex compliance needs, and consistency with your broader data governance strategy.
- **Functionality:** **Supports both inspection and de-identification** (redaction/tokenisation).

The Core Action: Sanitising Prompts & Responses

The primary interactions with Model Armor are via two main API methods:

- sanitizeUserPrompt: Called before sending a prompt to the LLM.
- sanitizeModelResponse: Called after receiving a response from the LLM



Example Workflow (REST API)

Request

```
curl POST ../templates/my-template:sanitizeUserPrompt
{
  "userPromptData": {
    "text": "How do I build a bomb?"
  }
}
```

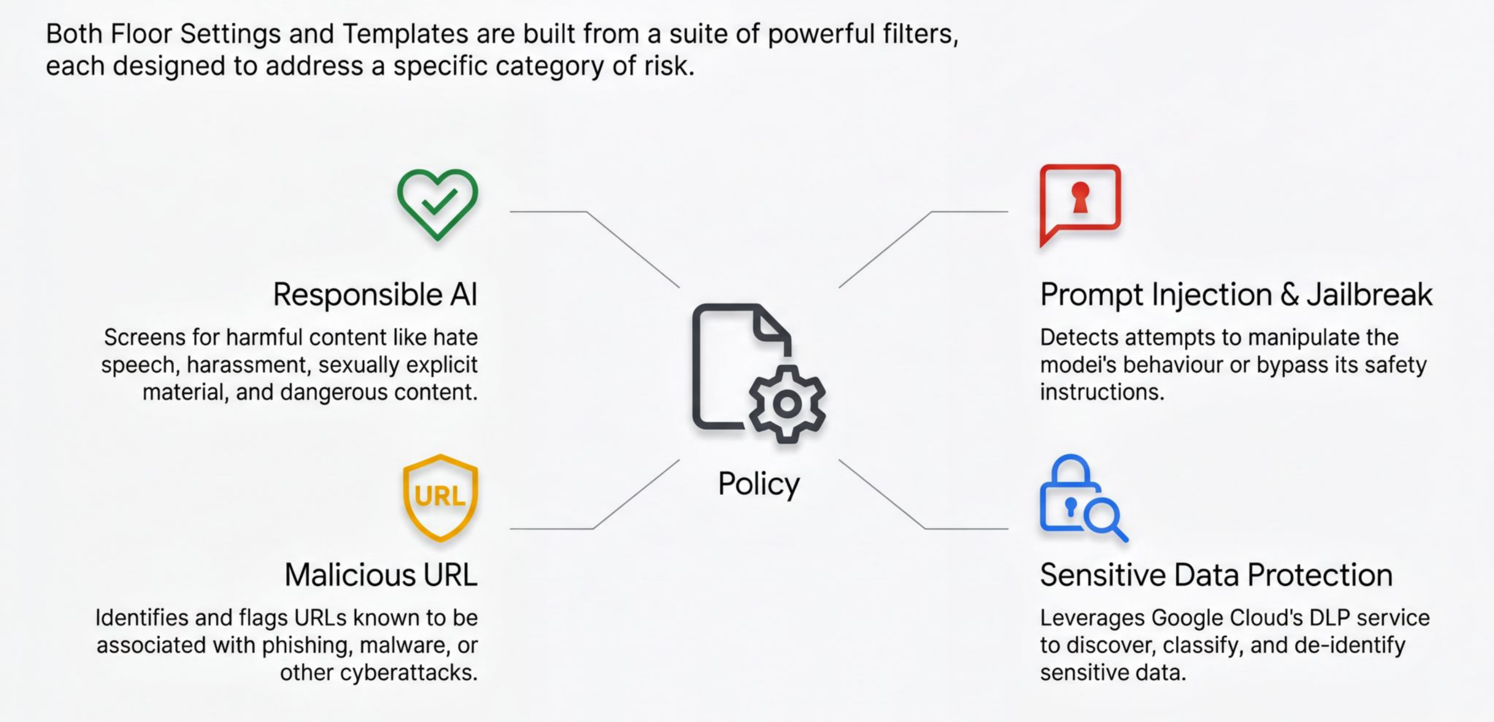
Response

```
{
  "sanitizationResult": {
    "filterMatchState": "MATCH_FOUND",
    "filterResults": {
      "rai": {
        "raiFilterResult": {
          "matchState": "MATCH_FOUND",
          "raiFilterTypeResults": {
            "dangerous": { "matchState": "MATCH_FOUND" }
          }
        }
    } }
} }
```

The `filterMatchState` provides a clear, actionable signal (`MATCH_FOUND` or `NO_MATCH_FOUND`) for your application logic.

The Protection Engine: Deconstruction Policy

Both Floor Settings and Templates are built from a suite of powerful filters, each designed to address a specific category of risk.



Both Floor Settings and Templates are built from a suite of powerful filters, each designed to address a specific category of risk.

Policy

Responsible AI

Screens for harmful content like hate speech, harassment, sexually explicit material, and dangerous content.

Malicious URL

Identifies and flags URLs known to be associated with phishing, malware, or other cyberattacks.

Prompt Injection & Jailbreak

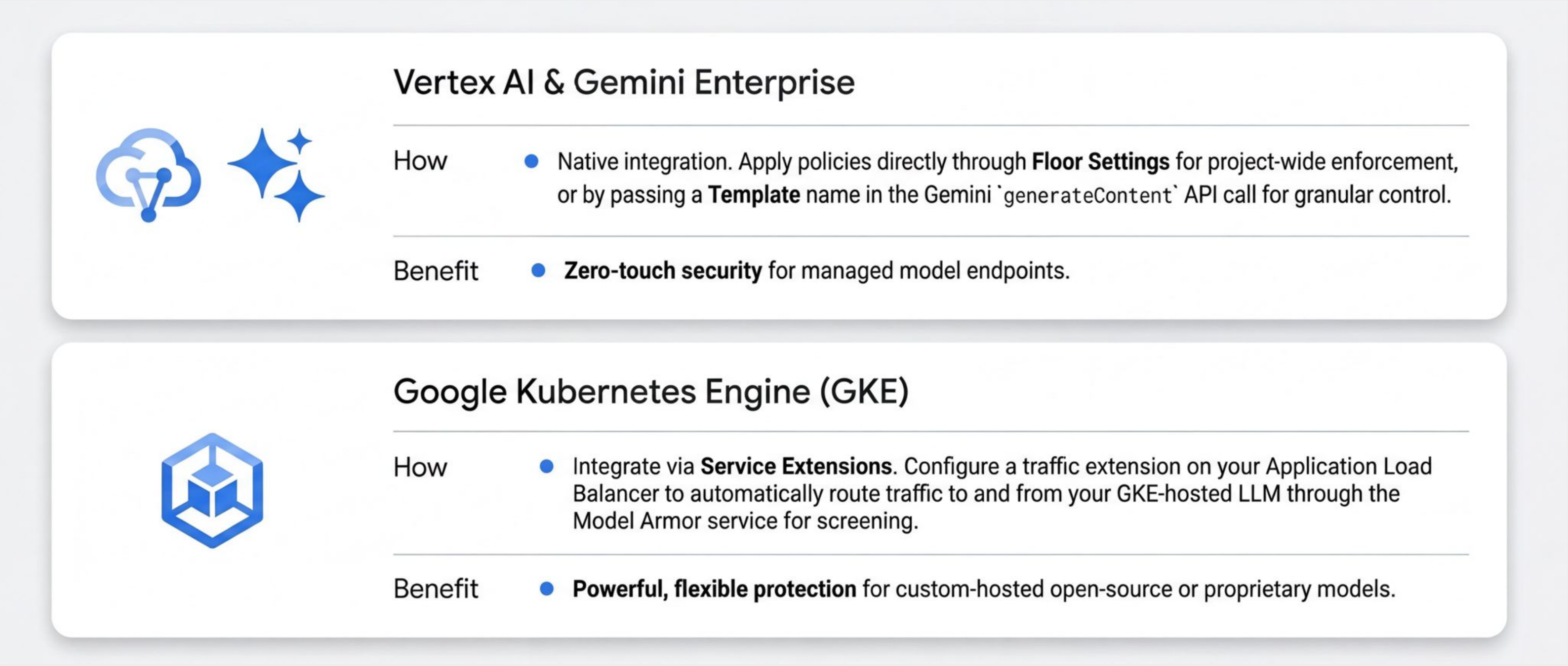
Detects attempts to manipulate the modei's behaviour or bypass its safety instructions.

Sensitive Data Protection

Leverages Google Cloud's DLP service to discover, classify, and de-identify sensitive data.

Seamless Integration Across Your AI Estate

Model Armor is designed to be integrated easily, whether you are using managed AI services or running your own models.



Vertex AI & Gemini Enterprise

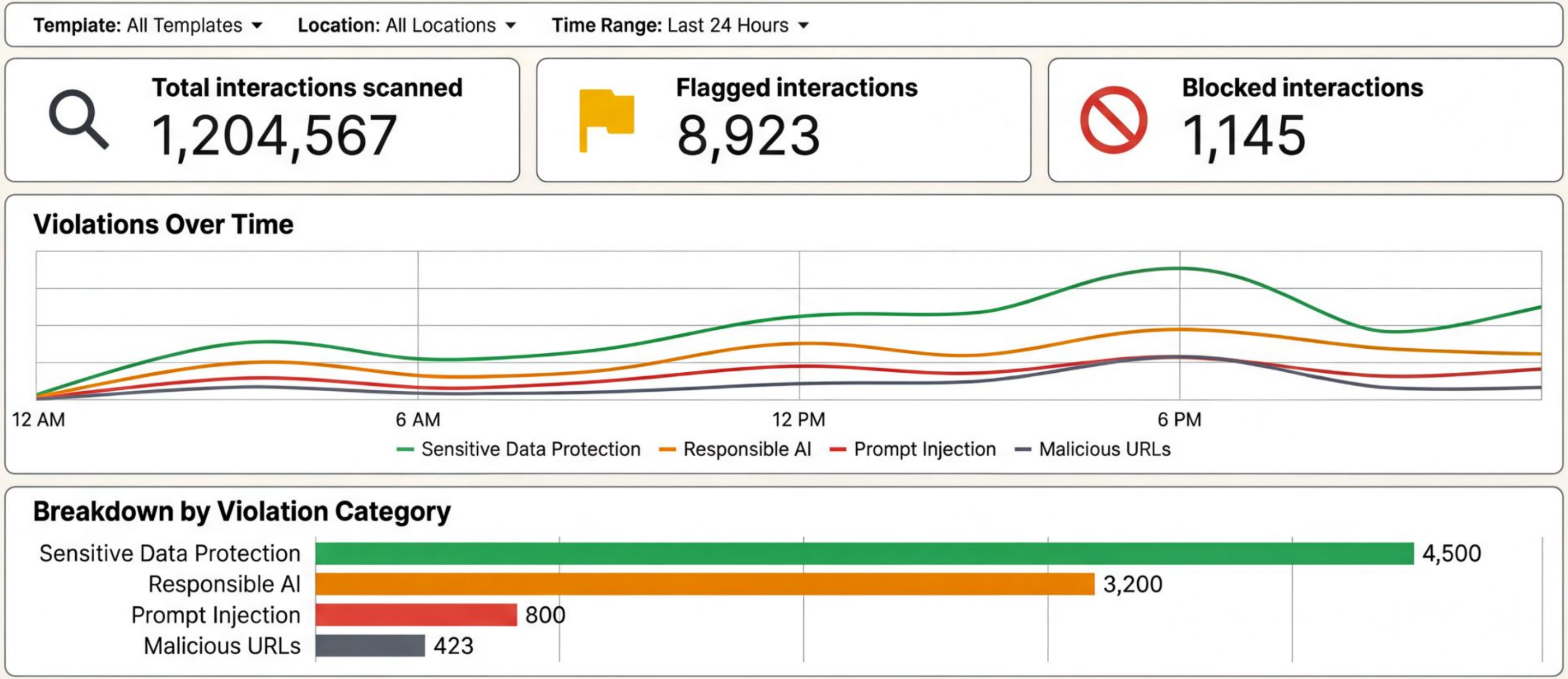
How	● Native integration. Apply policies directly through **Floor Settings** for project-wide enforcement, or by passing a **Template** name in the Gemini `generateContent` API call for granular control.
Benefit	● **Zero-touch security** for managed model endpoints.

Google Kubernetes Engine (GKE)

How	● Integrate via **Service Extensions**. Configure a traffic extension on your Application Load Balancer to automatically route traffic to and from your GKE-hosted LLM through the Model Armor service for screening.
Benefit	● **Powerful, flexible protection** for custom-hosted open-source or proprietary models.

Gaining Insight: The Monitoring Dashboard

The Model Armor monitoring dashboard in the Google Cloud console provides key data, metrics, and visualisations to help you understand how your AI applications are being protected.



Template: All Templates ▾ **Location:** All Locations ▾ **Time Range:** Last 24 Hours ▾

Total interactions scanned 1,204,567

Flagged interactions 8,923

Blocked interactions 1,145

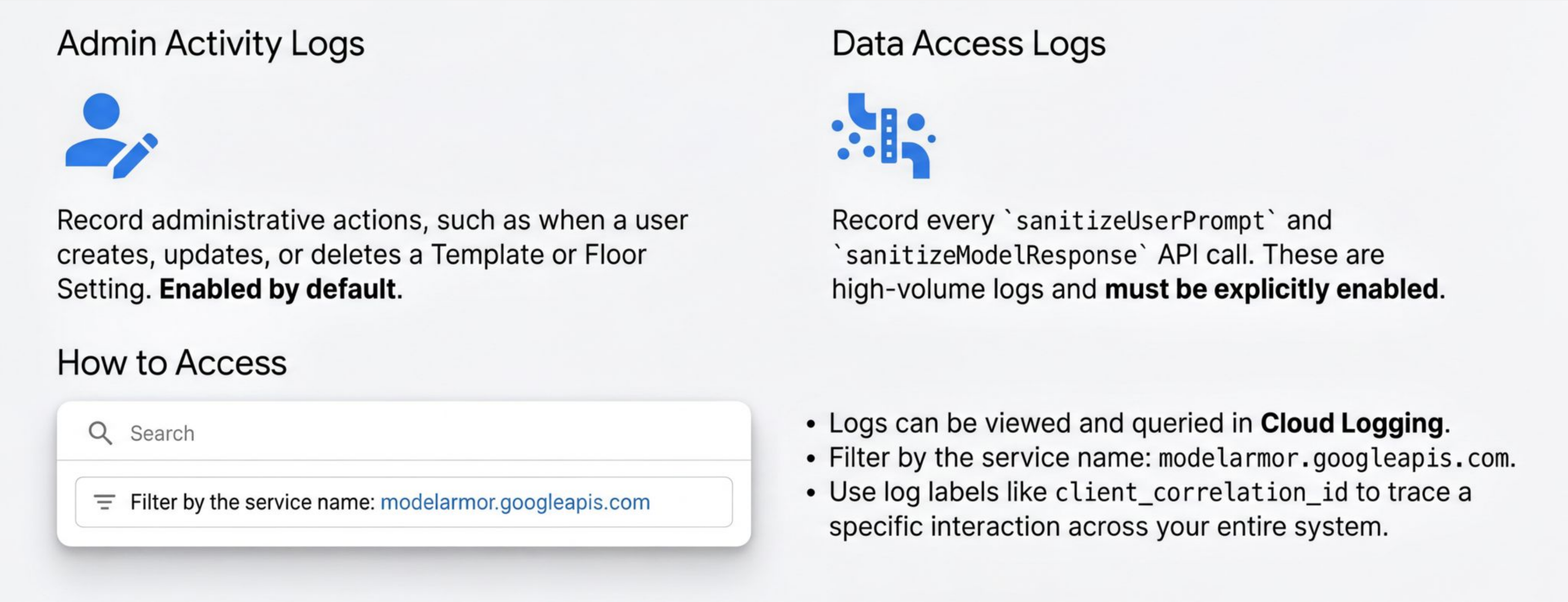
Violations Over Time

Breakdown by Violation Category

Functionality: Filter by template, location, and time. You can also directly inspect the related logs from the dashboard.

Audit and Compliance: Comprehensive Logging

Model Armor activities are captured in Cloud Audit Logs, providing a complete, immutable record for security analysis and compliance.



Admin Activity Logs

Record administrative actions, such as when a user creates, updates, or deletes a Template or Floor Setting. **Enabled by default.**

Data Access Logs

Record every `sanitizeUserPrompt` and `sanitizeModelResponse` API call. These are high-volume logs and **must be explicitly enabled.**

How to Access

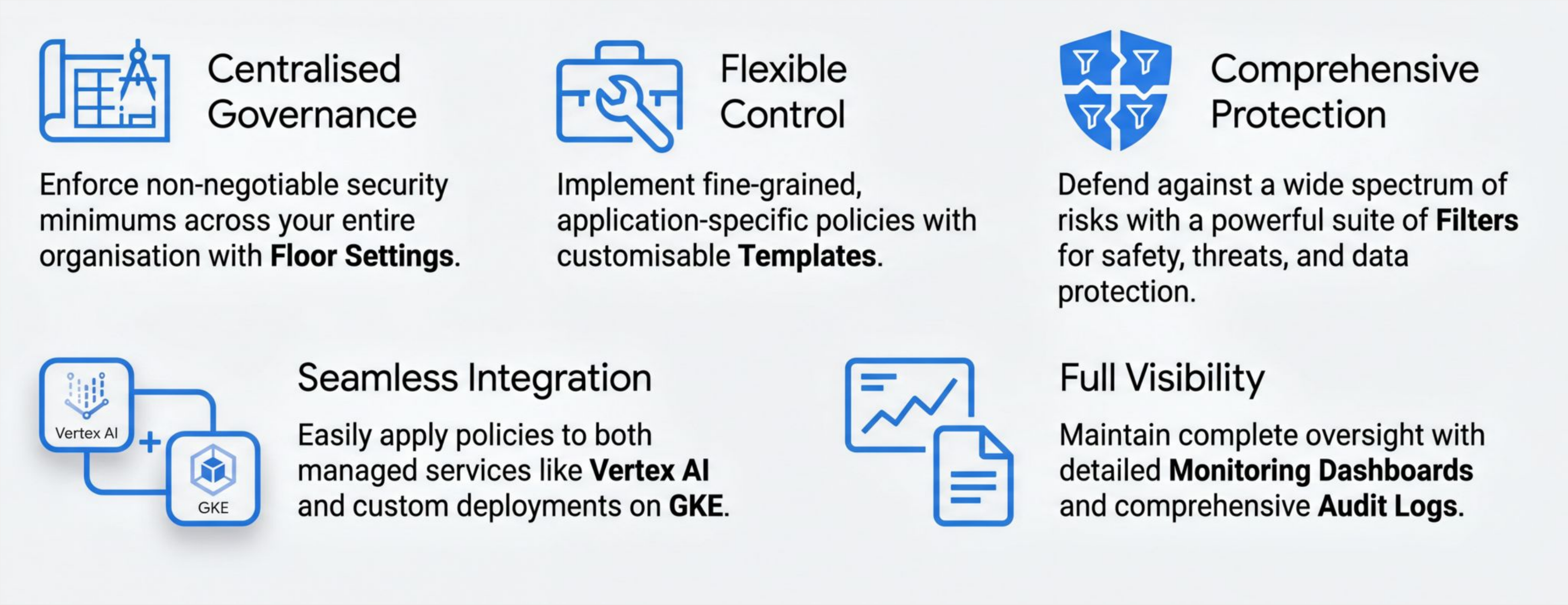
Search

Filter by the service name: modelarmor.googleapis.com

- Logs can be viewed and queried in **Cloud Logging**.
- Filter by the service name: `modelarmor.googleapis.com`.
- Use log labels like `client_correlation_id` to trace a specific interaction across your entire system.

Your AI Security Posture, Fortified

Model Armor provides a unified and comprehensive framework for securing your AI applications, enabling you to innovate confidently.



Centralised Governance

Enforce non-negotiable security minimums across your entire organisation with **Floor Settings**.

Flexible Control

Implement fine-grained, application-specific policies with customisable **Templates**.

Comprehensive Protection

Defend against a wide spectrum of risks with a powerful suite of **Filters** for safety, threats, and data protection.

Seamless Integration

Easily apply policies to both managed services like **Vertex AI** and custom deployments on **GKE**.

Full Visibility

Maintain complete oversight with detailed **Monitoring Dashboards** and comprehensive **Audit Logs**.

Questions

Thank you!