2️⃣ Security Architecture Review (SAR) – Draft Document

Below is a **Security Architecture Review document** you can submit to a security or platform review board.
It is intentionally concise, defensible, and enterprise-aligned.

Security Architecture Review

System: ADDIE-Based Prompt Management & Vertex AI Agent Retrieval
Phase: Pilot → Production Readiness
Reviewer Audience: Security, Platform, Architecture Review Board

---

1. System Overview

This system enables structured creation, governance, and retrieval of AI prompts aligned to the ADDIE instructional design framework. A Vertex AI Agent provides natural-language access to **approved prompts only**, enforcing quality and security controls before reuse.

The architecture follows a **defense-in-depth model** across:

- Agent Layer
- Management / Policy Layer
- Data Layer

---

2. Threat Model Summary

Primary Threats Considered

- Prompt injection attacks
- System prompt or policy exfiltration
- Credential or secret leakage
- Unauthorized tool or data access
- Agent hallucination due to ungoverned data sources

---

3. Security Controls by Layer

A. Agent Layer (Vertex AI Agent / Agentspace)

Controls

- Hardened system instructions (non-overridable)
- Explicit refusal of:
    - Instruction override requests
    - System prompt disclosure
- Agent restricted to **approved data sources only**

Key Principle

> The agent never enforces security — it benefits from upstream enforcement.

B. Management Layer (Cloud Run / Workflow)

Controls

- Regex-based input filtering (e.g., `BANNED_PATTERNS`)
- Pre-invocation validation:
    - Prompt injection attempts
    - Credential-related language
- Standardized refusal responses
- Audit logging of blocked requests (with redaction)

Why this layer matters

- Prevents unsafe input from ever reaching the LLM
- Enables deterministic, testable enforcement

C. Data Layer (Cloud Storage → BigQuery)

Controls

- Separation of concerns:
    - Draft prompts (staging)
    - Approved prompts (registry)
- Agent connected only to **approved registry**
- Read-only access for agent service account

- No secrets stored in prompt content

---

4. Human-in-the-Loop Security Governance (Pilot)

Security Review Rubric

- Injection Risk
- System Exposure Risk
- Credential Risk
- Tool/Data Boundary Risk

Decision Rules

- Any critical failure → Reject
- Borderline → Approve with edits
- Clean → Approve

This produces labeled data for future automation.

---

5. Phase 2 Automation Readiness

Pilot Control	Phase 2 Automation
Manual security scoring	Automated policy engine
Regex screening	Expanded policy rules
Human approval	Conditional workflow promotion
Sheet-based audit	BigQuery + Cloud Audit Logs

---

6. Residual Risk & Mitigations

Risk	Mitigation
Novel prompt injection	Human review + policy updates
Over-blocking	Reviewer override with notes
Model behavior drift	Agent instruction hardening

---

7. Compliance & Alignment

- Least privilege (IAM)
- No credential exposure at inference
- Explainable agent responses
- Audit-ready prompt lifecycle

---

8. Security Posture Statement

> The system is designed to prevent prompt-based attacks, credential exposure, and unauthorized instruction overrides through layered controls, human governance, and agent-safe data sourcing.

What I recommend next (strongly)

1. Add **conditional formatting** to security scores (red = auto-reject)
2. Define **Phase 2 policy YAML** (machine-readable)
3. Map security events to **Cloud Logging + Alerts**
4. Create a **one-slide Security Control Overlay** for your architecture diagram

If you want, I can do any of these next — or all of them in sequence.