# **Seraphim Vanguard POC:**

# **Standardized Use Case Framework**

## **Overview**

This document outlines a standardized Proof of Concept (POC) use case for Seraphim's AI governance platform, organized around the platform's three core value propositions:

1. **Security**
2. **Integrity**
3. **Accuracy**

This use case will serve as the benchmark for demonstrating Seraphim's capabilities across sectors like energy, finance, insurance, and healthcare, while aligning with enterprise-grade expectations for AI safety, trust, and compliance.

## **POC Use Case Title**

**"Automated High-Risk Document Processing with AI Oversight"**

### **Industry Context (Energy Example)**

Large utilities must routinely process thousands of maintenance, regulatory, and compliance documents for grid safety operations. These documents are often reviewed by human teams or static automation tools that lack end-to-end traceability, security boundaries, or explainability.

## **POC Business Objective**

Demonstrate how Seraphim's Vanguard architecture (Security, Integrity, Accuracy) ensures enterprise-safe AI deployment for mission-critical document ingestion, review, and output generation.

## **Workflow Phases (Aligned to Vanguards)**

| **Phase** | **Vanguard** | **Function** | **Enterprise Value** |
| --- | --- | --- | --- |
| 1. Agent Created | - | Client AI Studio generates agent workflow (e.g., ingest PDF, extract work orders, summarize risk) | Accelerated AI dev cycle w/ clear traceability |
| 2. **Security Vanguard** | Inspects and edits agent script for restricted API calls, unsafe I/O, and sandboxing | Prevents unintentional exfiltration, enforces policy |  |
| 3. Execution | Agent is deployed to hosting platform (on-prem, cloud, edge) | Flexible enterprise integration |  |
| 4. **Integrity Vanguard** | Validates ingested data (e.g., outdated docs, conflicting entries) and performs rule-based filtering | Reduces hallucinations and decision risk |  |
| 5. **Accuracy Vanguard** | Reviews output + logs, cross-validates with source-of-truth client data | Builds audit trust, detects AI drift or fraud |  |
| 6. Client Delivery | Final certified output delivered with logs and version-controlled traceability | Ensures compliance, legal defensibility |  |

## **Sample AI Task for POC**

**"Ingest 100 substation inspection reports and produce a summary dashboard with flagged anomalies, risk scores, and outage recommendations."**

## **What Success Looks Like**

* ✅ No hallucinated data or ghost anomalies
* ✅ All outputs can be traced to a source document and validated
* ✅ System flags a discrepancy in 1 record, triggers human-in-the-loop check
* ✅ Logs show which agent version, rules, and data were used at every step
* ✅ Final output clean, exportable to enterprise system

## **Business Outcomes**

| Vanguard | Outcome | Business Impact |
| --- | --- | --- |
| Security | No rogue actions or insecure calls | Reduced breach risk, faster InfoSec approval |
| Integrity | Detects bad input before output generation | Prevents catastrophic errors (e.g., PG&E-level risk) |
| Accuracy | Explains every output decision | Reduces compliance cost, strengthens trust w/ regulators |

## **Cross-Sector Portability**

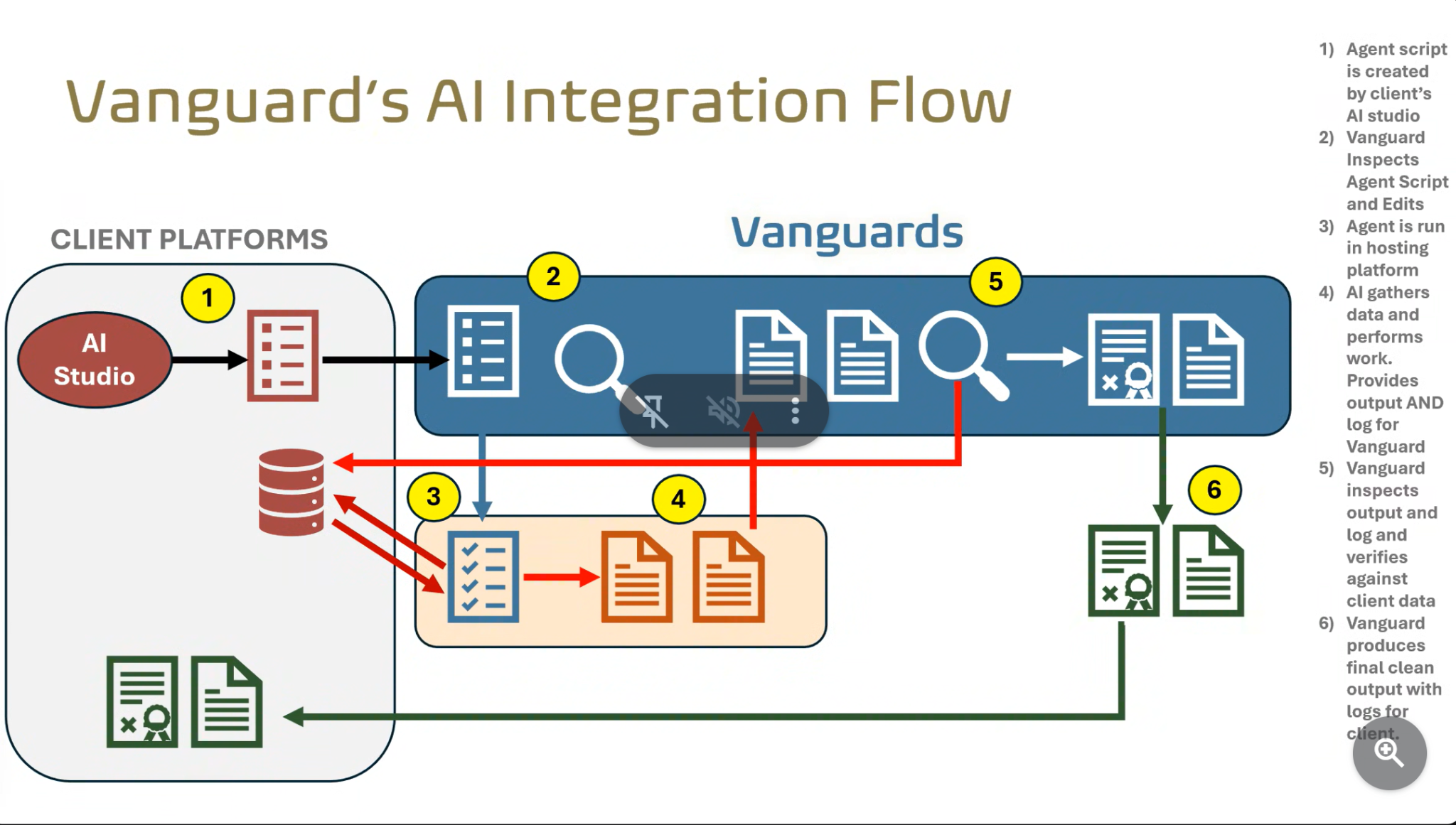
This POC can be immediately translated into:

* **Finance**: Loan review agents validating income vs. application docs
* **Insurance**: Claims triage bots with embedded fraud flags
* **Healthcare**: Prior auth and billing reviews with HIPAA-grade traceability

## **Next Steps**

1. Deploy sandbox environment
2. Select sample documents and define policy rules
3. Run Seraphim POC with logs enabled
4. Review outcome with compliance, IT, and risk officers

**This use case is designed not only to prove technical merit, but to demonstrate enterprise-grade AI you can trust.**



### **🔐 Security Vanguard**

**Mission**: Prevent unauthorized access, sandbox AI behavior, and secure sensitive operations

| **Capability** | **Recommended Tool** | **Rationale** |
| --- | --- | --- |
| Agent sandboxing & policy enforcement | **CrewAI Security Layer** | Purpose-built for secure agent execution with rule-bound behavior |
| Identity & Access Management (IAM) | **Auth0** or **Keycloak (Open Source)** | Granular, enterprise-ready authentication control |
| Secrets management | **HashiCorp Vault** | Secure token/API credential storage during agent runtime |
| API call inspection | **Kong Gateway w/ OPA (Open Policy Agent)** | Filters and governs outbound API calls dynamically |
| Network layer protection | **Tailscale (mesh VPN)** | Ensures zero-trust communication between edge and core agents |

### **✅ Integrity Vanguard**

**Mission**: Validate all incoming data, cross-check against trusted sources, enforce business rules

| **Capability** | **Recommended Tool** | **Rationale** |
| --- | --- | --- |
| Data validation rules | **Great Expectations** | Open-source library to define and enforce validation checks |
| Schema governance | **OpenMetadata** or **Apache Atlas** | Tracks lineage, schema versions, and metadata integrity |
| Version control for input pipelines | **DVC (Data Version Control)** | Tracks input dataset versions and model/data relationships |
| Policy-based anomaly flagging | **Google Cloud Data Loss Prevention (DLP)** or **OpenAI + RAG filters** | Enforces PII compliance and flags structured data inconsistencies |

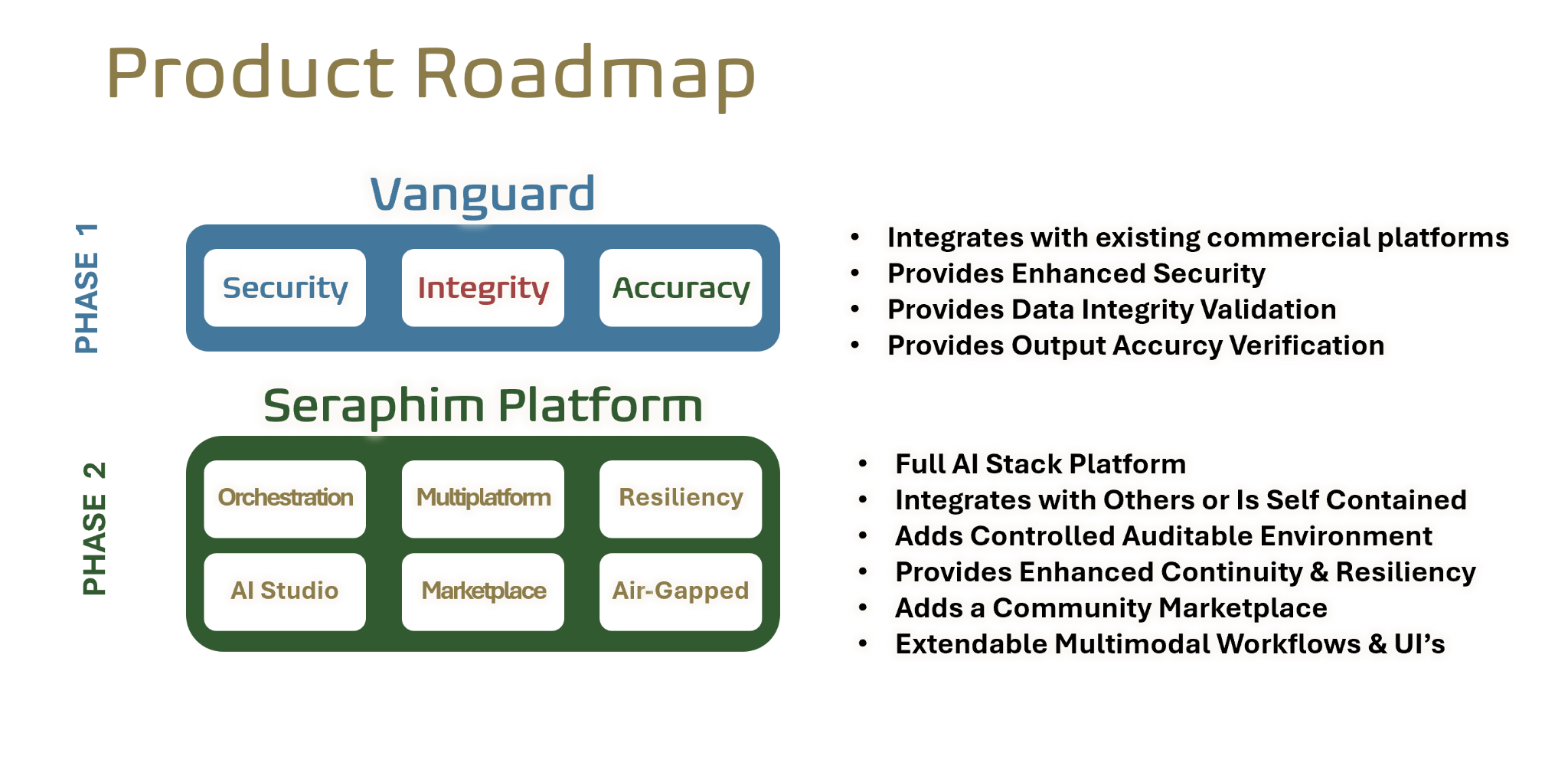
### **🧠 Accuracy Vanguard**

**Mission**: Ensure outputs are verifiable, traceable, and grounded in truth

| **Capability** | **Recommended Tool** | **Rationale** |
| --- | --- | --- |
| Retrieval-Augmented Generation (RAG) | **LlamaIndex or LangChain w/ Pinecone/Weaviate** | Ensures all answers are grounded in verifiable enterprise data |
| Agent logging and explainability | **Weights & Biases (WandB)** or **Arize AI** | Provides model behavior traceability, even across agent chains |
| Output comparison against source-of-truth | **Deepchecks or Evidently AI** | Model monitoring & performance validation across outputs |
| Workflow trace logs | **OpenTelemetry** or **Jaeger** | Distributed tracing across agent workflows and subprocesses |

### **🚀 Additional Stack Considerations**

| **Category** | **Stack Piece** | **Note** |
| --- | --- | --- |
| Hosting | **Docker + Kubernetes** | Containerized agents run on client-preferred infra (on-prem or hybrid) |
| Orchestration | **Temporal.io or Prefect** | Agent process sequencing with retries, error handling |
| Language Interop | **gRPC + REST layer** | Allows plug-and-play with any client platform |
| Frontend | **Streamlit or Retool (internal ops)** | For live admin console or interface demo in POC |



## **🛠️ PHASE 1: Vanguard (SIA Stack)**

Focus: Lightweight, modular validation layer to integrate with existing AI workflows

### **⏱️ Build Duration: 8–12 weeks (agile sprints)**

### **1. 🔐 SECURITY (Plug-in Option: CrewAI Security + LangChain Guardrails)**

* **Function**: Intercept and vet agent code for insecure actions, blacklisted APIs, environment misconfigurations.
* **Plug-and-Play Tools**:  
  + [CrewAI](https://github.com/joaomdmoura/crewAI): for agent orchestration + security sandboxing
  + [OpenAI Function Call Restrictions]
  + [LangChain Guardrails or Rebuff]: enforce call structure and input validation
  + **Containerized Sandboxing**: Docker + microVM isolation (e.g., Firecracker)
* **Output**: Prevents data exfiltration, model misuse, unauthorized API requests.

### **2. 📥 INTEGRITY (Plug-in Option: Guardrails.AI + HoneyHive)**

* **Function**: Validates incoming documents and data structure. Filters out conflicting inputs.
* **Plug-and-Play Tools**:  
  + Guardrails.AI (data schema + context guards)
  + HoneyHive (automated document validation)
  + Optional: Azure Form Recognizer or Amazon Textract for structured ingestion
* **Output**: Prevents model hallucination, protects against garbage-in logic failures.

### **3. 📊 ACCURACY (Plug-in Option: Retrieval-Augmented Verification + Human-in-the-loop)**

* **Function**: Verifies AI output against authoritative data sources. Logs all reasoning steps.
* **Plug-and-Play Tools**:  
  + RAG stack with source-of-truth (e.g., vector DB: Pinecone, Weaviate)
  + LangSmith / PromptLayer for log tracing
  + Human-in-loop option (via Label Studio or LightTag)
* **Output**: Ensures enterprise trust, legal auditability, and correctness confidence.

## **🔁 INTEGRATION & DELIVERY TOOLING**

* Platform Agnostic Wrappers: LangChain, LlamaIndex
* Deployment: Dockerized APIs, optional REST endpoints
* Logging: ElasticStack, Datadog, or OpenTelemetry
* Versioning: DVC or Git-LFS for model/data provenance

## **🚀 PHASE 2: Seraphim Platform (Full Stack AI Governance)**

Focus: Scalable platform for creation, deployment, and orchestration of trusted agents across domains.

### **⏱️ Build Duration: 16–20 weeks (modular delivery)**

| **Capability** | **Stack & Tools** | **Description** |
| --- | --- | --- |
| **Orchestration** | **CrewAI**, **Prefect**, **Temporal** | Modular workflows that span multiple agents/tasks |
| **Multiplatform** | **Docker + Kubernetes** | Can be deployed cloud, edge, or on-prem |
| **Resiliency** | **Air-Gapped Option**, **HA Redundancy** | Failover and state sync protection |
| **AI Studio** | **Low-Code Builder**: (Streamlit + LangChain UI) | Agent builder for enterprise analysts or SMEs |
| **Marketplace** | **Gradio** + plugin support | Launch, share, and reuse agent templates |
| **Air-Gapped** | **Private LLM + No API Calls** | Self-contained runtime for high-trust environments |

## **🔄 BUILD ORDER: SUGGESTED SPRINT MAP**

| **Sprint** | **Feature(s)** | **Goal** |
| --- | --- | --- |
| Sprint 1–2 | Security Vanguard | Sandbox security + rule engine w/ CrewAI |
| Sprint 3–4 | Integrity Vanguard | Data validation and logging with Guardrails/HoneyHive |
| Sprint 5–6 | Accuracy Vanguard | Source-traceable output with log-backed RAG |
| Sprint 7–8 | Integration Layer | Deliver API & workflow orchestration |
| Sprint 9–12 | Seraphim Platform MVP | Studio + Marketplace + Resiliency |

## **🧩 FINAL PRODUCT LAYERS**

### **Phase 1 (Integrated into Existing Enterprise Workflows):**

* ✅ Vanguard runs as **interceptor and validator**
* ✅ Secures existing agents or LLM applications
* ✅ Enterprise gains visibility, compliance, and error prevention

### **Phase 2 (Optional Full Stack):**

* ✅ Standalone AI development and deployment environment
* ✅ Enables internal app development with built-in safety, version control, and audit logs
* ✅ Marketplace connects to external agent providers