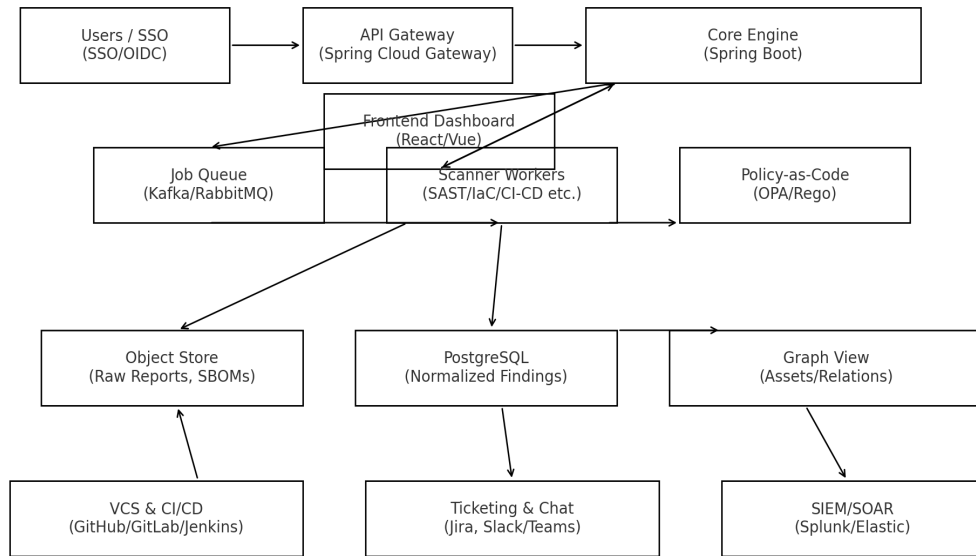# Security Orchestration & Risk Platform - High-Level Design (HLD)

Architecture • Threat Model • Data Model • Risk Scoring • Deployment • Roadmap
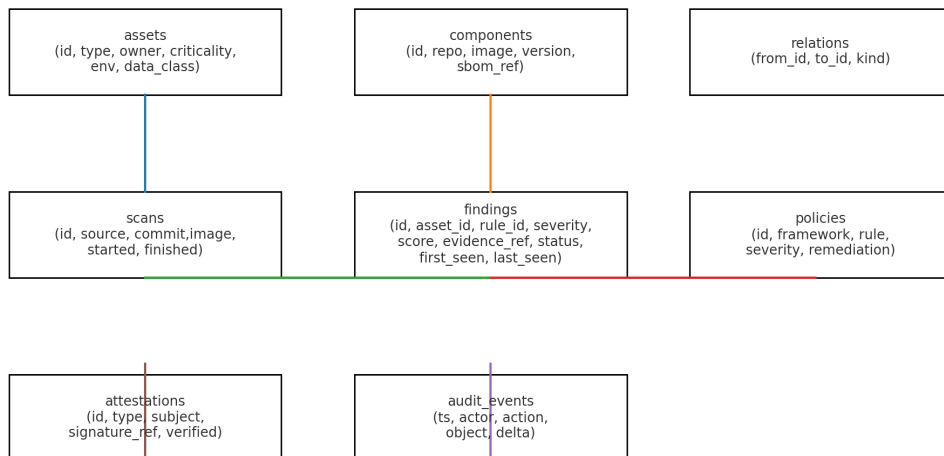
Executive Summary

This document defines the high-level design for a modular security orchestration platform that discovers, scans, and prioritizes architectural risks across code, pipelines, containers, cloud/Kubernetes, and external exposure. The platform emphasizes Zero Trust, Policy-as-Code, evidence-driven governance, and an asset/relationship graph to highlight blast radius and attack paths.
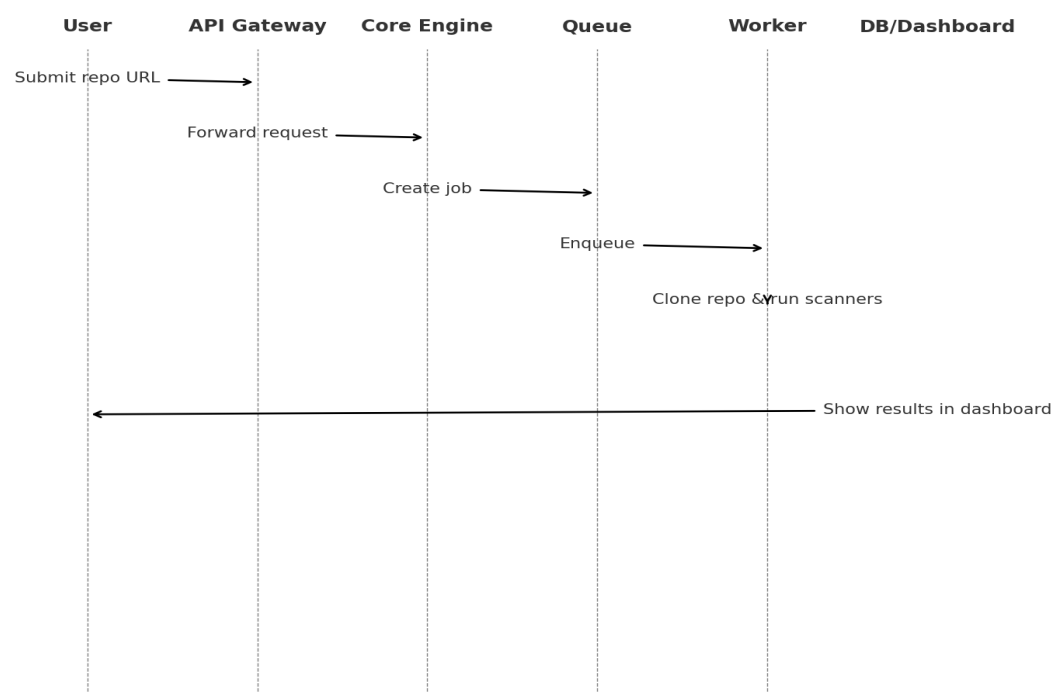
Architecture Overview

Users / SSO (SSO/OIDC)	API Gateway (Spring Cloud Gateway)	Core Engine (Spring Boot)

Frontend Dashboard (React/Vue)

Job Queue (Kafka/RabbitMQ)	Scanner Workers (SAST/IaC/CI-CD etc.)	Policy-as-Code (OPA/Rego)

Object Store (Raw Reports, SBOMs)	PostgreSQL (Normalized Findings)	Graph View (Assets/Relations)

VCS & CI/CD (GitHub/GitLab/Jenkins)	Ticketing & Chat (Jira, Slack/Teams)	SIEM/SOAR (Splunk/Elastic)

Data Model

assets (id, type, owner, criticality, env, data_class)	components (id, repo, image, version, sbom_ref)	relations (from_id, to_id, kind)

scans (id, source, commit,image, started, finished)	findings (id, asset_id, rule_id, severity, score, evidence_ref, status, first_seen, last_seen)	policies (id, framework, rule, severity, remediation)

attestations (id, type, subject, signature_ref, verified)	audit_events (ts, actor, action, object, delta)

Sequence Diagrams

Repo Scan Sequence

| User | API Gateway | Core Engine | Queue | Worker | DB/Dashboard |

Submit repo URL →

Forward request →

Create job →

Enqueue →

Clone repo & run scanners

Show results in dashboard

Cloud/K8s Scan Sequence

Connector	Core Engine	Policy Engine	DB/Dashboard

Pull configs (Cloud/K8s) →

Evaluate policies →

Normalized findings →

← Posture score/PR fix
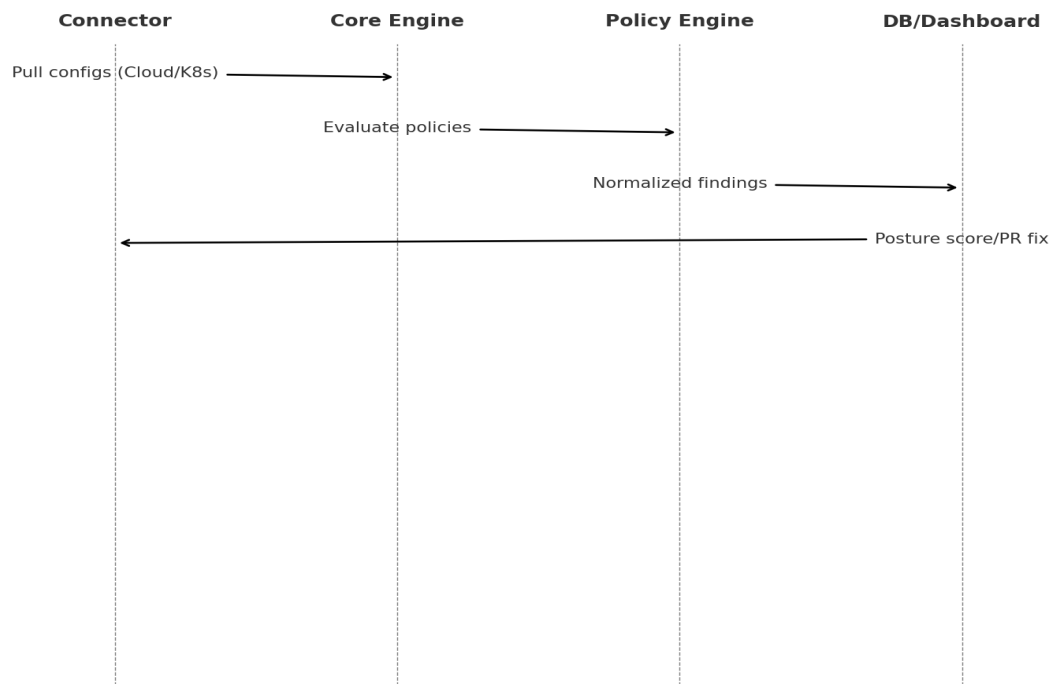
Example Attack Path Graph

Example Attack Path Graph