OdinForge AI

Technical Architecture and Security Design

Technical White Paper | February 2026

Table of Contents

---

1. Introduction

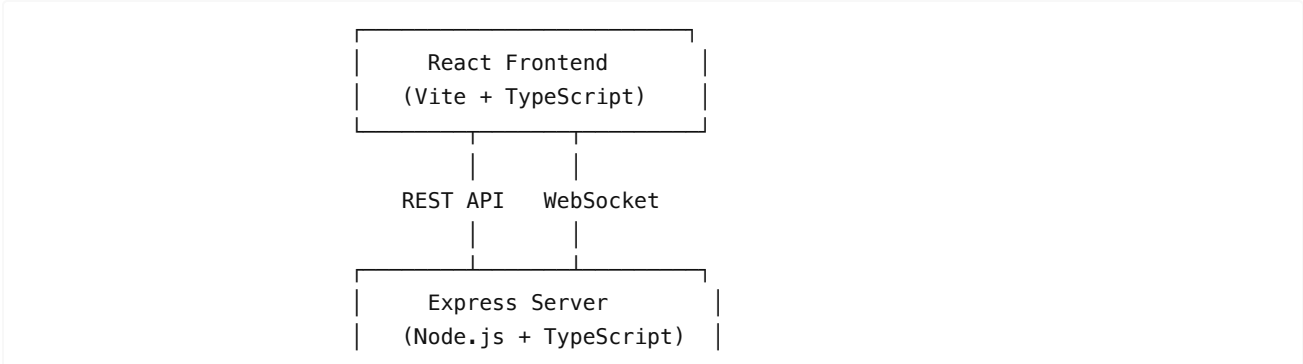
This document provides a technical deep-dive into the architecture, data flows, security model, and design decisions behind OdinForge AI. It is intended for security architects, engineering leads, and technical evaluators who need to understand how the platform operates at a systems level.

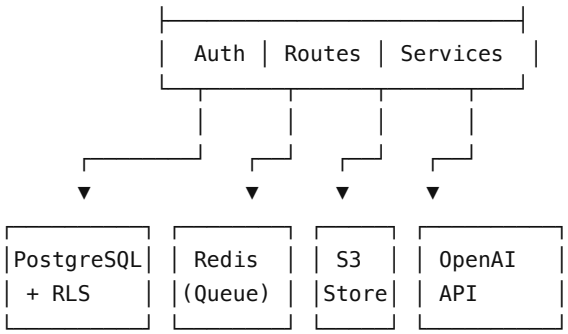
For a capabilities overview and business context, refer to the companion Executive White Paper.

---

2. System Architecture

High-Level Component Map





Technology Stack

Layer	Technology	Purpose
Frontend	React 18, TypeScript, Vite	Single-page application with lazy-loaded routes
UI Framework	shadcn/ui, Tailwind CSS, Radix primitives	Accessible component library
Routing	wouter	Lightweight client-side routing (29 routes)
State Management	React Query (TanStack Query)	Server state caching with configurable refetch intervals
Backend	Express.js, TypeScript	REST API server with 200+ endpoints
ORM	Drizzle ORM	Type-safe SQL with schema-driven migrations
Database	PostgreSQL	Relational storage with row-level security
Vector Store	pgvector (1536 dimensions)	AI embedding storage for similarity search
Queue	BullMQ on Redis	Persistent job queue with priority scheduling
Object Storage	S3-compatible	Evidence artifacts, report files
AI Provider	OpenAI API	Narrative generation, agent reasoning
Real-Time	Native WebSocket	Live progress events, status broadcasts

Request Lifecycle

1. Client sends authenticated request with JWT bearer token
2. Rate limiter evaluates request against per-endpoint and per-user thresholds
3. Authentication middleware validates JWT, extracts user identity and organization
4. Permission middleware checks user's role against required permission for the endpoint
5. Tenant context is set — all subsequent database queries are scoped via row-level security
6. Route handler executes business logic
7. For long-running operations, a job is enqueued and a job ID returned immediately
8. WebSocket pushes progress events to connected clients as jobs execute

---

3. Multi-Agent AI Engine

Agent Architecture

OdinForge employs six specialized AI agents, each responsible for a distinct analysis domain:

Agent	Domain	Responsibility
Recon Agent	Discovery	Asset enumeration, attack surface mapping, technology fingerprinting
Exploit Agent	Validation	Exploit construction, payload generation, vulnerability confirmation
Lateral Agent	Movement	Pivot path discovery, credential reuse analysis, network traversal
Business Logic Agent	Application	IDOR, mass assignment, workflow abuse, payment flow manipulation
Multi-Vector Agent	Synthesis	Cross-domain attack path construction, chained exploit assembly
Impact Agent	Consequence	Financial exposure calculation, compliance mapping, operational impact

Tiered Parallel Execution

Agents execute in a tiered parallel model rather than sequential or fully parallel:

```
Tier 1 (Parallel):  Recon Agent  +  Business Logic Agent
                          │                 │
                          ▼                 ▼
Tier 2 (Parallel):  Exploit Agent  +  Lateral Agent
                          │                 │
                          ▼                 ▼
Tier 3 (Parallel):  Multi-Vector Agent  +  Impact Agent
```

This design ensures that discovery-phase agents complete before exploitation agents begin, while agents within the same tier run concurrently for throughput. Each tier receives the accumulated context from all preceding tiers.

Circuit Breaker Protection

LLM provider calls are wrapped in a circuit breaker pattern:

- **Closed State** — Requests flow normally to the provider
- **Open State** — After 2 consecutive failures, the circuit opens and all requests fail fast for 60 seconds
- **Half-Open State** — After the reset timeout, a single probe request tests provider health

This prevents cascading failures when an AI provider experiences degradation. The circuit breaker operates per-provider, so a failure in one integration does not affect others.

Agent Timeout Policy

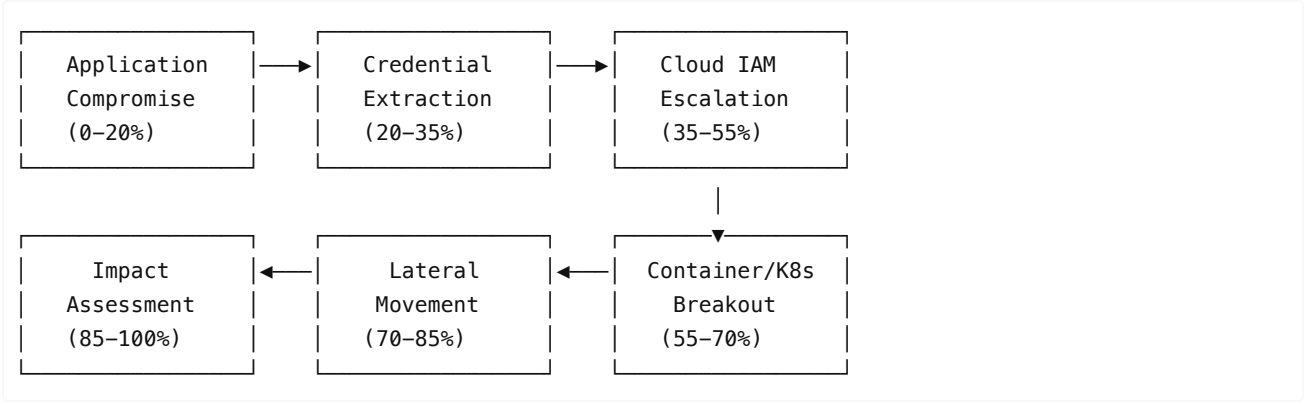
Each agent call enforces a 30-second timeout. Combined with the circuit breaker's 2-failure threshold, the maximum wasted time on a degraded provider is 60 seconds before fail-fast behavior activates.

---

4. Breach Chain Pipeline

Pipeline Architecture

Breach chains model multi-phase adversarial campaigns that cross domain boundaries. The pipeline maintains cumulative state across six sequential phases:



Phase Context Propagation

Each phase receives and extends a cumulative `BreachPhaseContext`:

Context Field	Type	Description
Credentials	Array	Harvested credentials with type, source, access level, and validated targets
Compromised Assets	Array	Assets with type, access level, compromise method, and timestamp
Attack Path Steps	Array	Ordered chain of techniques with outcomes
Evidence Artifacts	Array	Proof of compromise artifacts
Privilege Level	Enum	Current highest privilege: none → user → admin → system → cloud_admin → domain_admin
Domains Compromised	Array	Fully compromised network domains

Context flows forward — each phase reads what previous phases discovered and appends its own results. This enables realistic simulation where, for example, the lateral movement phase uses credentials harvested during the credential extraction phase.

Credential Security Model

Breach chains track credential discovery as part of the attack simulation, but enforce strict security controls:

- **No plaintext storage** — Credentials are stored as hashed values only
- **Type classification** — password, hash, ticket, token, key, API key, IAM role, service account
- **Access level tracking** — What the credential grants access to
- **Source attribution** — Which phase and technique discovered the credential
- **Target validation** — Which systems the credential has been confirmed to work against

Safety Gate Architecture

Before each phase executes, a safety gate evaluates:

1. **Execution mode compliance** — Is the requested action allowed under the current mode (safe/simulation/live)?
2. **Scope rule compliance** — Is the target within allowed scope boundaries?
3. **Kill switch status** — Has the emergency halt been activated?
4. **Abort status** — Has a user requested chain termination?
5. **Timeout compliance** — Is the chain within per-phase and total timeout limits?

Safety gates produce three outcomes: `ALLOW`, `DENY`, or `MODIFY` (adjust the action to comply with safety constraints). All decisions are logged.

Pause and Resume

When `pauseOnCritical` is enabled and a critical finding is discovered mid-chain:

1. The current phase completes its in-progress operation
2. The chain state (context, completed phases, findings) is persisted to the database
3. The chain status transitions to `paused`
4. A human reviewer inspects the finding and decides to resume or abort
5. On resume, the orchestrator restores state from the database and continues from the next unfinished phase

Post-Chain Processing

After all phases complete, the orchestrator:

1. Builds a unified attack graph combining all phase results
2. Calculates an aggregate breach risk score
3. Generates an AI-powered executive summary narrative
4. Persists the complete chain to the database
5. Creates Purple Team findings for the defensive feedback loop
6. Broadcasts completion via WebSocket

---

5. Intelligent Risk Scoring Engine

Three-Dimensional Scoring

Every evaluation can produce an intelligent score across three independent dimensions:

Risk Rank quantifies overall severity on a 0-100 scale with a mapped risk level (emergency, critical, high, medium, low, info). It includes a fix priority number (1 = fix first) and a recommendation with action, timeframe, and business justification.

Business Impact evaluates the financial and compliance consequence of exploitation. It produces estimated direct loss ranges (min/max dollar values) and identifies affected compliance frameworks.

Exploitability measures the ease of exploitation based on actual validation results rather than theoretical CVSS scores.

Fix Priority Algorithm

The fix priority queue orders all findings by a composite score that weights:

- Business impact severity (highest weight)
- Validated exploitability (confirmed > theoretical)
- Compliance framework exposure count
- Affected asset criticality

- Time sensitivity of the remediation window

This produces a single ordered queue where item #1 is the most business-critical finding to address first, with a recommended timeframe: immediate, 24 hours, 7 days, 30 days, 90 days, or acceptable risk.

MITRE ATT&CK Coverage Analysis

The platform maps all evaluations and findings to MITRE ATT&CK tactics and techniques, then computes:

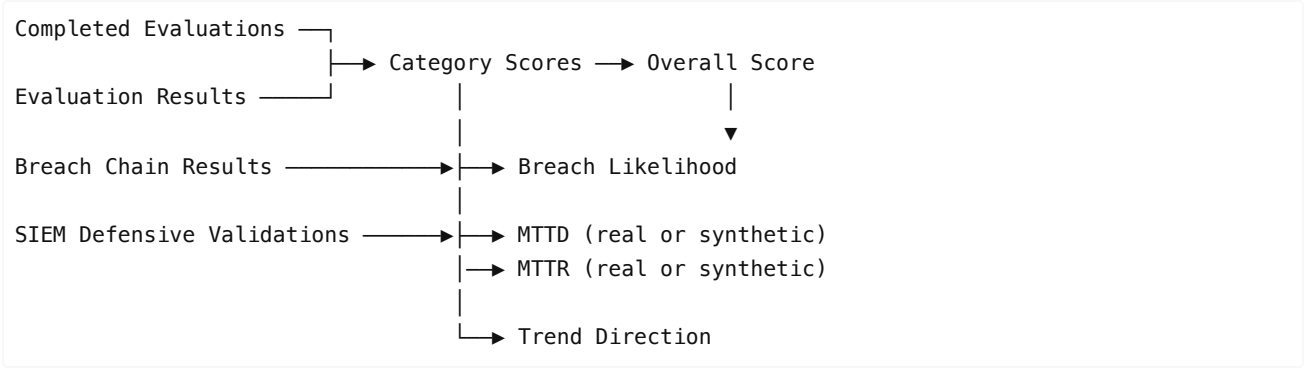
- **Asset coverage** — Percentage of active assets evaluated within the last 30 days
- **Technique coverage** — Number of unique MITRE techniques exercised
- **Tactical coverage** — Which ATT&CK tactics have been tested vs. which remain untested
- **Coverage gaps** — Stale assets (not recently evaluated) and untested tactics

---

6. Defensive Posture and Prediction Models

Posture Calculation Pipeline

The defensive posture score is computed from multiple data sources:



Category Scores

Six security categories are independently scored (0-100):

Category	What It Measures
Network Security	Network-layer vulnerability exposure and segmentation
Application Security	Web app, API, and business logic vulnerability posture
Identity Management	Authentication, authorization, and IAM configuration
Data Protection	Encryption, data exposure, and exfiltration resistance
Incident Response	Detection capability and response readiness
Security Awareness	User-facing attack surface (phishing, social engineering)

Breach Chain Integration

When completed breach chains exist, the posture calculation enriches scores:

- **Breach likelihood** is blended: 40% from evaluation-based calculation + 60% from real breach chain risk scores
- **Overall score** is penalized based on maximum privilege achieved in chains (domain_admin: -25 points, cloud_admin: -20, system: -15, admin: -10)
- **Blocked phase bonus** — Successfully blocked phases add points (defense signal)
- **Category penalties** — MITRE techniques exploited in chains penalize the corresponding security category
- **Recommendations** include breach chain findings summary

SIEM-Observed Metrics

When the organization has defensive validation records from SIEM integration:

- **MTTD** (Mean Time to Detect) is calculated from real observed detection timestamps when at least 3 samples exist; otherwise a synthetic estimate is used
- **MTTR** (Mean Time to Respond) follows the same threshold logic
- The data source is transparently labeled: `siem_observed` or `synthetic`
- Sample sizes are reported for confidence assessment

Attack Prediction Model

The prediction engine forecasts likely attack vectors within configurable time horizons (7, 30, or 90 days):

1. Counts exposure type frequency across completed evaluations
2. Generates predicted attack vectors with likelihood percentages and MITRE ATT&CK mapping
3. Identifies risk factors with trend indicators (increasing, stable, decreasing)
4. Enriches with breach chain data:
    - Vectors matching real breach chain MITRE techniques receive confidence and likelihood boosts
    - New vectors from breach chains not yet in predictions are added
    - Breach chain success rates and critical findings contribute as risk factors
5. Blends overall breach likelihood using the same 40/60 evaluation/breach-chain weighting

---

7. External Reconnaissance Engine

Scan Module Architecture

The external recon engine operates without agent deployment, gathering intelligence about internet-facing targets through seven independent modules:

Module	Data Produced
Port Scan	Open ports, service identification, banner grabbing
SSL/TLS Analysis	Certificate validity, expiry, TLS version, cipher strength, vulnerabilities
HTTP Fingerprint	Server identity, technologies, frameworks, security headers (present/missing)
Auth Surface Detection	Login pages, admin panels, OAuth endpoints, password reset, API auth methods
Transport Security	Forward secrecy, HSTS, OCSP, downgrade risks, overall grade (A+ through F)
Infrastructure Discovery	Hosting/CDN/cloud provider, subdomains, related domains, IP geolocation, ASN, SPF/DMARC

Attack Readiness	Composite exposure score (0-100), category breakdown, prioritized next actions with MITRE mapping

OSINT Integration

Infrastructure discovery includes:

- DNS enumeration across A, AAAA, MX, TXT, NS, and CNAME record types
- Reverse DNS lookups
- Subdomain and related domain discovery
- Historical DNS record analysis
- IP geolocation with ASN and organization lookup
- Shadow asset identification

AEV Handoff

The attack readiness summary produces prioritized next actions that feed directly into the AEV pipeline. Each action specifies the exploit type, target vector, priority, and confidence level — enabling automated transition from reconnaissance to validation.

---

8. Report Generation Pipeline

Dual Engine Architecture

V1 Template Engine produces structured reports using deterministic templates. Templates are parameterized with evaluation data, finding counts, severity distributions, and remediation timelines. Output is consistent and predictable.

V2 AI Narrative Engine generates consulting-quality reports using LLM-powered narrative generation. The engine receives structured evaluation data and produces human-readable prose with:

- Minimum length enforcement (executive summaries: 200+ chars, attack narratives: 300+ chars)
- Structured output validation via Zod schemas
- Financial exposure analysis with category breakdowns
- 30/60/90-day remediation roadmaps
- Attack path narratives with reasoning chains and MITRE mapping
- Prioritized fix plans with specific commands and verification steps

Report Type Matrix

Type	V1	V2	Key Sections
Executive Summary	Yes	Yes	Risk overview, financial exposure, strategic recommendations, board briefing
Technical Deep-Dive	Yes	Yes	Attack narratives, finding details, fix plans, architecture recommendations
Compliance	Yes	Yes	Framework-specific gap analysis, compliance scores, remediation roadmap
Breach Chain Analysis	No	Yes	End-to-end attack progression, credential chain, privilege escalation timeline

Compliance Framework Coverage

Reports can be generated against eight frameworks: SOC 2, PCI DSS, HIPAA, GDPR, CCPA, ISO 27001, NIST CSF, FedRAMP

Date/Time Standardization

All report timestamps use military Date Time Group (DTG) format: `DDHHMMZMONYR` (e.g., `100000ZFEB26`). Date range boundaries are normalized to UTC: start-of-day as `00:00:00.000Z`, end-of-day as `23:59:59.999Z`. This eliminates timezone ambiguity regardless of server or client locale.

Generation Pipeline

1. User selects report type, format (PDF/JSON/CSV), date range or evaluation scope
2. Request is validated and a background job is enqueued
3. Job handler gathers evaluations within scope
4. Progress stages: gathering → analyzing → generating → formatting → complete
5. V2 reports invoke the LLM with structured input and validated output schemas
6. Generated report is stored in the database with optional file attachment in S3
7. WebSocket notification signals completion

---

9. Job Orchestration and Queue Architecture

Queue Design

OdinForge uses BullMQ backed by Redis for persistent, priority-based job processing.

Queue Configuration:

- Default concurrency: 5 workers
- Auto-retry with exponential backoff (1-second initial delay)
- Job retention: completed jobs removed after 24 hours (keep last 1000), failed after 7 days
- Fallback: in-memory queue when Redis is unavailable

Priority Levels

Level	Priority	Use Case
1	Critical	Live exploitation operations, emergency scans
2	High	Active assessments, breach chain phases
3	Normal	Standard evaluations, scheduled scans
4	Low	Report generation, data exports
5	Background	Cleanup tasks, metric recalculation

Job Types

The system supports 13 distinct job types:

Category	Job Types
Assessment	Evaluation, Full Assessment, Exploit Validation
Simulation	AI Simulation

Scanning	Network Scan, External Recon, API Scan, Auth Scan, Protocol Probe
Infrastructure	Cloud Discovery, Agent Deployment
Output	Report Generation
Remediation	Remediation (with dry-run support)

Job Lifecycle

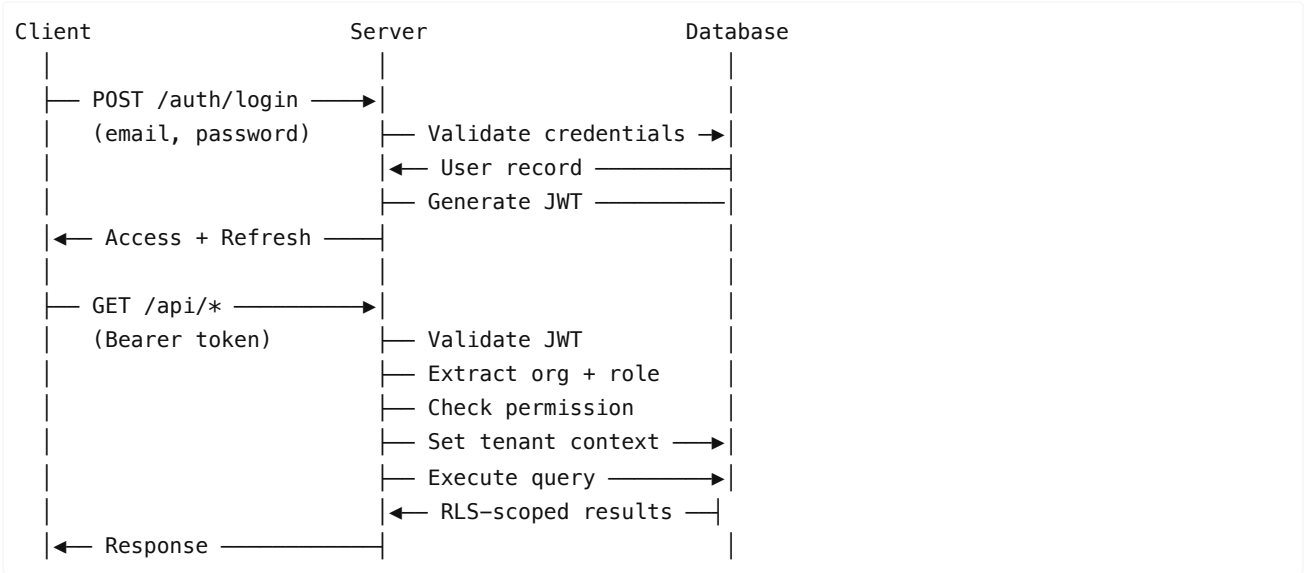
```
Created → Queued → Processing → Completed
                          │
                          ├──→ Failed → (Retry) → Queued
                          ├──→ Cancelled
                          └──→ Stalled → (Auto-retry)
```

Each job emits progress events via WebSocket, enabling real-time UI updates without polling.

---

10. Identity, Access Control, and Multi-Tenancy

Authentication Flow



Role Hierarchy

```
Platform Super Admin (all permissions, cross-tenant)
  └── Organization Owner (all org permissions)
        ├── Security Administrator (operational control)
        │     ├── Security Engineer (technical execution)
        │     │     └── Security Analyst (read + triage)
        │     └── Compliance Officer (GRC-focused)
        └── Executive Viewer (dashboards + executive reports only)
```

```
Automation Account (API-only, no UI access)
Endpoint Agent (system identity, telemetry submission only)
```

Permission Model

67 granular permissions follow an `action:resource` pattern:

Module	Permission Categories
Evaluations	read, create, execute_safe, execute_simulation, execute_live, approve_live, delete, archive
Assets	read, create, update, delete
Reports	read, read_executive, generate, export, delete
Agents	read, register, manage, revoke, delete
Evidence	read, read_sanitized
Findings	read, triage
Simulations	read, run, delete
Governance	read, manage
Audit	read, read_global (cross-tenant)
Organization	read, manage_settings, manage_users, assign_roles
Platform	emergency_access, feature_flags, rate_limits, cross_tenant_access
API	read, write

Multi-Tenancy via Row-Level Security

Tenant isolation is enforced at the database level:

1. Every authenticated request sets a PostgreSQL session variable identifying the organization
2. Row-level security (RLS) policies on all tenant-scoped tables filter rows by organization ID
3. This is enforced regardless of application-layer logic — even a bug in route handlers cannot leak cross-tenant data
4. Platform Super Admins can optionally bypass RLS with the `cross_tenant_access` permission

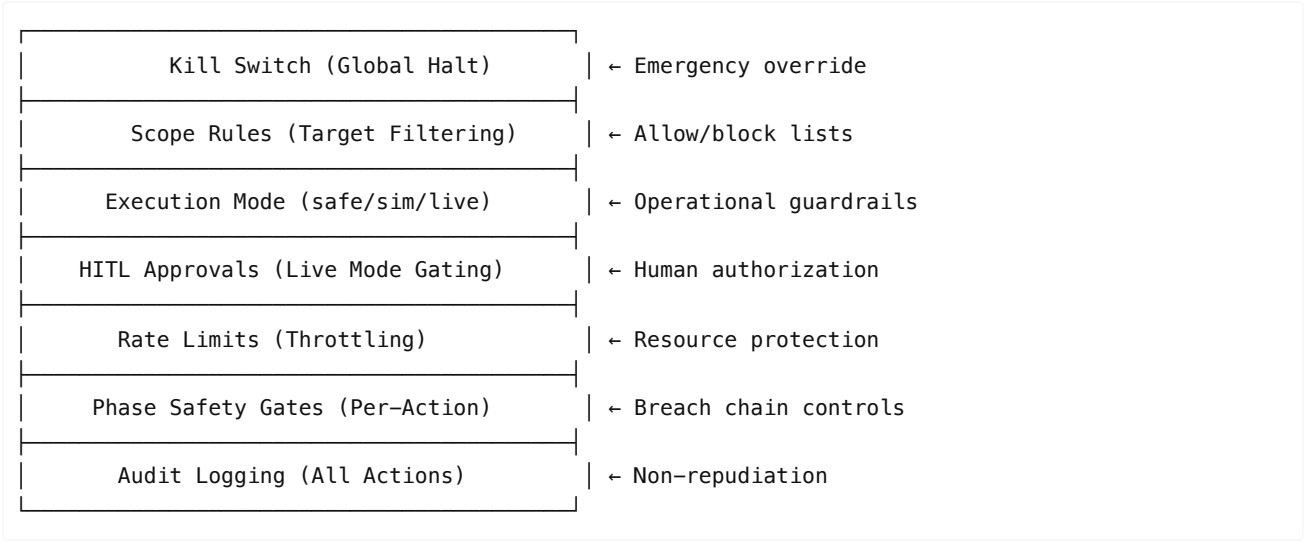
Database Role Mapping

The database uses short-form role identifiers (e.g., `org_owner`) while the application schema uses full-form identifiers (e.g., `organization_owner`). A bidirectional mapping function (`dbRoleToSchemaRole`) translates between the two representations at the middleware layer.

---

11. Governance and Safety Architecture

Defense-in-Depth Safety Model



Kill Switch Behavior

When activated:

- All running breach chains receive an abort signal
- All queued jobs for the organization are cancelled
- New evaluation and assessment requests are rejected
- The kill switch state is logged with actor attribution
- Deactivation requires the same permission level and is independently logged

Auto-Kill Trigger

When enabled, the system automatically activates the kill switch if a running evaluation or breach chain discovers a critical-severity finding. This provides a safety net for unattended operations.

HITL Approval Protocol

1. Operation triggers a governance policy match
2. Approval request is created with: operation details, risk level, triggered policy, requesting user
3. Request enters a time-limited pending state
4. Authorized reviewer approves (with cryptographic nonce) or rejects (with documented reason)
5. Expired requests are automatically rejected
6. All decisions are immutable audit records with non-repudiation signatures

Scope Rule Engine

Scope rules define permitted and prohibited targets:

Rule Type	Matching Logic
IP	Exact IP address match
CIDR	IP range containment check
Hostname	Exact or wildcard hostname match
Regex	Pattern matching against target identifiers

Rules are evaluated in order: explicit blocks take precedence over allows. Any target not matching an allow rule is implicitly blocked when allow rules are defined.

---

12. Real-Time Communication

WebSocket Architecture

The server maintains persistent WebSocket connections with authenticated clients. Events are broadcast per-organization to ensure tenant isolation in real-time communication.

Event Types:

Event	Trigger	Payload
`aev_progress`	Evaluation execution	Progress percentage, current stage, intermediate findings
`aev_complete`	Evaluation finished	Final status, verdict, score
`assets_updated`	Asset inventory change	Change type, affected asset IDs
Job progress	Background job stage change	Job ID, stage, progress percentage
Breach chain update	Phase completion	Chain ID, phase, status, context summary

Client Reconnection

The frontend implements automatic WebSocket reconnection with exponential backoff. Missed events during disconnection are reconciled via React Query refetch on reconnection.

---

13. Data Model and Storage

Core Entities

Entity	Description	Key Relationships
Evaluation	Individual exposure validation	Links to asset, results, evidence
Result	Finding from an evaluation	Links to evaluation, includes MITRE mapping
Full Assessment	Multi-phase assessment	Contains web recon, findings, attack graph, recommendations
Breach Chain	Multi-phase attack simulation	Contains phase results, context, executive summary
Report	Generated report document	Links to evaluations, breach chains, stored content
Agent	Registered endpoint agent	Links to telemetry records
Discovered Asset	Known infrastructure asset	Links to cloud connection, evaluations
Cloud Connection	Cloud provider integration	Links to discovered assets

Evidence	Forensic artifact	Links to evaluation, includes SHA-256 hash
Approval Request	HITL decision record	Links to operation, includes signature
Audit Log	Immutable event record	Links to actor, target resource
Governance Settings	Organizational controls	Per-organization configuration
Scope Rule	Target allow/block rule	Links to organization

Vector Embeddings

The database includes pgvector support with 1536-dimensional embeddings for:

- Finding similarity search (grouping related vulnerabilities)
- Knowledge retrieval for AI narrative generation
- Pattern matching across evaluation results

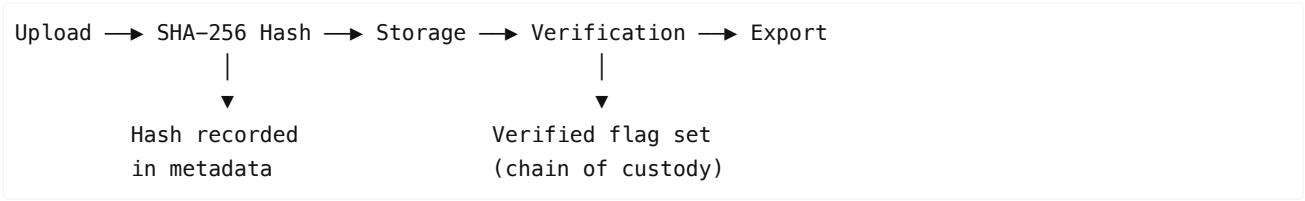
Schema Management

Database migrations are managed via Drizzle Kit with versioned migration files. Schema types are shared between server and client via a TypeScript schema module, ensuring type safety across the full stack.

---

14. Evidence and Forensic Integrity

Evidence Lifecycle



Integrity Guarantees

- **Automatic hashing** — SHA-256 hash computed at upload time and stored with the evidence record
- **Immutable hash** — Once recorded, the hash cannot be modified (any re-upload creates a new record)
- **Verification workflow** — Authorized users formally verify evidence, creating an auditable chain of custody
- **Forensic export** — Evidence packages include integrity hashes and chain-of-custody documentation

Supported Evidence Types

Screenshots, log files, network captures (PCAP), file artifacts, and analysis documents. Each type is stored in S3-compatible object storage with metadata in PostgreSQL.

---

15. Integration Architecture

Cloud Provider Integration

```
OdinForge ──► Cloud Provider API ──► Asset Discovery
    │              │
    │              ├── AWS (IAM Role / Access Key)
    │              ├── Azure (Service Principal)
    │              └── GCP (Service Account)
    │
    ├── Test Connection
    ├── Discover Assets
    ├── Sync Inventory
    └── Auto-Deploy Agents
```

Cloud connections support credential testing, on-demand discovery, periodic sync, and conditional agent auto-deployment based on tag, region, and instance size filters.

Vulnerability Scanner Integration

Data ingestion from third-party scanners:

Scanner	Import Format	Correlation
Nessus	Native export	Host/IP matching to discovered assets
Qualys	Native export	Host/IP matching to discovered assets
Tenable	Native export	Host/IP matching to discovered assets
OpenVAS	Native export	Host/IP matching to discovered assets
Custom	CSV/JSON	Configurable field mapping

Imported vulnerabilities are correlated with AEV evaluations for validation. A vulnerability's status progresses: `new` → `validated` (confirmed by AEV) → `remediated`.

SIEM Integration

Defensive validation records from SIEM systems provide real-world MTTD and MTTR data:

- Detection timestamps from SIEM alerts correlated to OdinForge evaluations
- Response timestamps from incident management systems
- Minimum sample threshold (3 observations) before SIEM data supersedes synthetic estimates

CI/CD Integration

The Automation Account role provides API-only access for pipeline integration:

- Trigger evaluations on deployment events
- Query posture scores as deployment gates
- Export findings in machine-readable formats (JSON, CSV)
- SOAR platform integration for automated response workflows

---

16. Deployment and Scalability

Deployment Models

Model	Description
Cloud SaaS	Multi-tenant hosted deployment with RLS isolation
On-Premise	Single-tenant deployment behind corporate firewall
Hybrid	Cloud management plane with on-premise agents

Horizontal Scaling Points

Component	Scaling Strategy
API Server	Stateless — add instances behind load balancer
Job Workers	Independent BullMQ workers — add instances for throughput
Database	PostgreSQL read replicas for query scaling
Redis	Redis Cluster for queue scaling
Object Storage	S3-compatible — inherently scalable

Resilience Patterns

- **Circuit breaker** — AI provider calls fail fast after consecutive failures
- **Queue persistence** — Redis-backed jobs survive server restarts
- **In-memory fallback** — Queue operates in-memory when Redis is unavailable
- **Exponential backoff** — Failed jobs retry with increasing delays
- **Job timeout** — Per-job timeout prevents indefinite execution
- **Stale resource cleanup** — Automated detection and removal of orphaned agents and expired tokens

---