SOVEREIGN EDITION · v266

Technical Architecture Whitepaper

Vigilant Entities

The Sovereign OS for Governance, Audit, and Venture.

This document is prepared for CTOs, CISOs, Data Protection Officers, board risk committees, and enterprise procurement teams evaluating Vigilant Entities as their operating system for regulated venture, audit, and governance workflows.

PREPARED BY · VIGILANT ENTITIES ENGINEERING

ISSUED · JULY 17, 2026

Vigilant Entities (VE) is a multi-tenant, white-label operating system for regulated venture and governance. It is built around three architectural commitments that are unusual in enterprise SaaS but which we consider non-negotiable for sovereign-grade deployments:

1. Audit-by-default. Every mutating request is written to an append-only audit trail with actor, tenant, IP, user-agent, resource, and payload hash. Zero opt-out. Zero silent state changes.

2. Sovereign data-plane. Data lives in the buyer's chosen region. Cryptographic material (secrets, JWT, DKIM) is deployment-scoped. No shared multi-tenant secrets across regions.

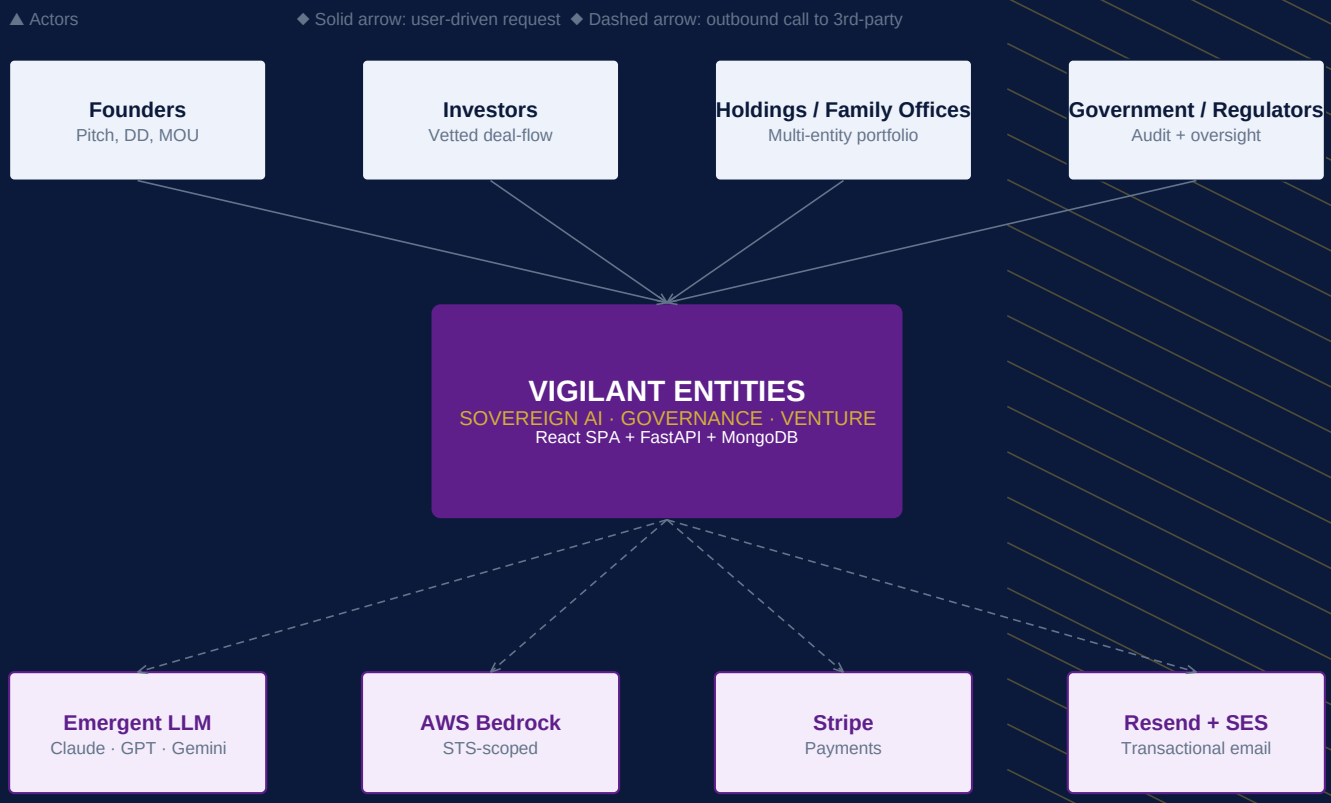
3. Maker-controlled surface area. A single, verifiable principal (the Maker) has cross-tenant visibility for platform operations. All other roles — including holding_admin and system_admin — are strictly scoped to their tenants.

26	35+	100%	30d
AI advisors in the Council (Shura Nexus)	Backend route modules · FastAPI	Requests behind TLS 1.3 + WAF	Remember-me session (opt-in, revocable per device)

SO WHAT

For a procurement or CISO reviewer, the summary is this: VE is a regulated-industry-grade FastAPI/React/MongoDB stack with per-tenant isolation, first-class audit, and a documented AI-safety envelope around every LLM interaction. The remaining pages of this document unpack the specific mechanisms that back each of those claims.

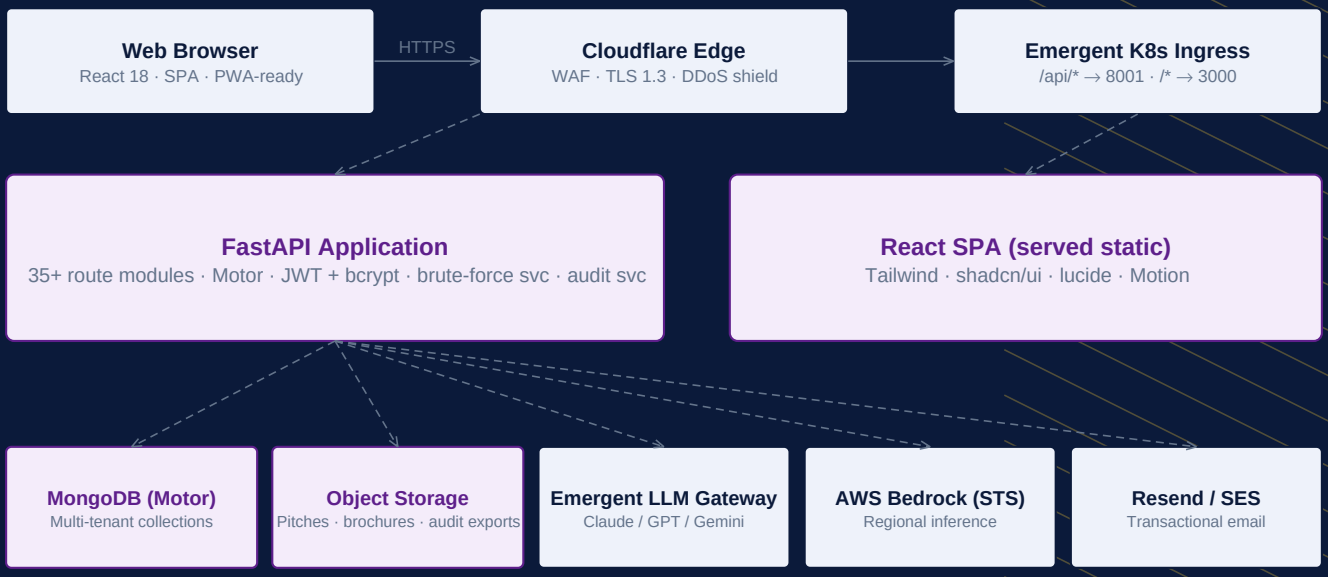
The system context diagram (C4 Level 1) shows the platform as a single logical unit with four categories of external actor and four categories of external system. All actor traffic terminates at TLS 1.3 on Cloudflare; all outbound third-party traffic uses signed API tokens rotated on the Maker cadence.



SO WHAT

The platform has exactly one ingress path (Cloudflare → K8s ingress → FastAPI) and a small, enumerated set of egress paths (LLM, email, payments, Bedrock). Everything else is denied by default at the network layer.

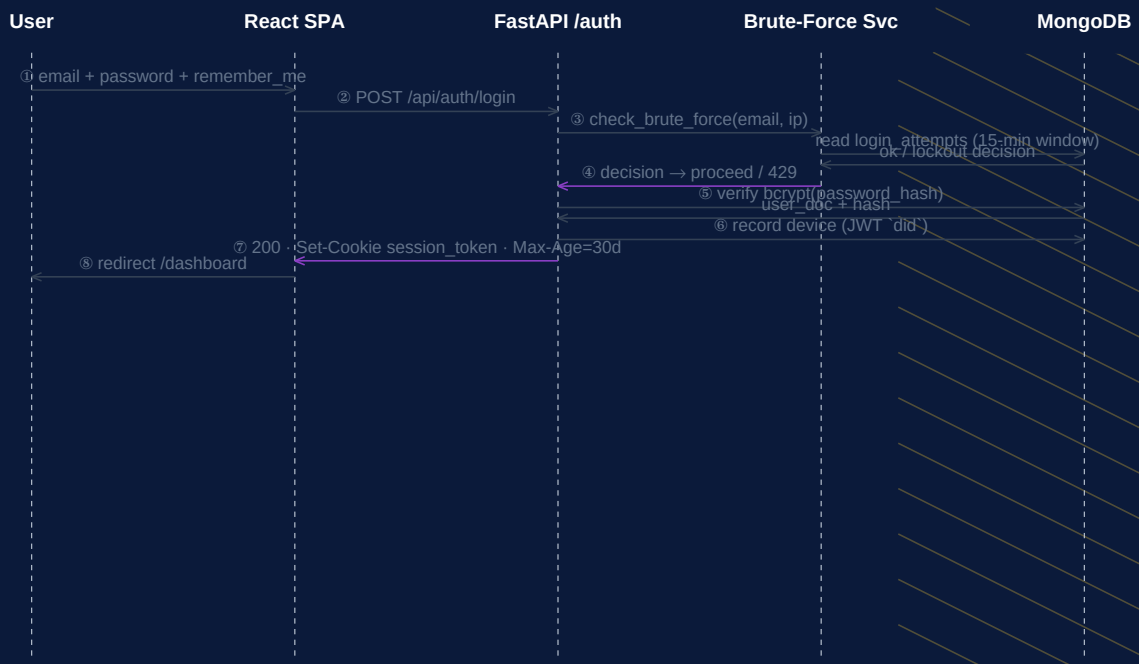
React SPA and FastAPI ship as two supervised processes in the same pod. Ingress routes **/api/*** to **backend:8001** and **/*** to **frontend:3000**. MongoDB is Motor-backed (async); the platform never issues blocking DB calls from a request path.



Solid arrows: synchronous HTTP request path · Dashed arrows: outbound service call

Container	Runtime	Purpose	Scaling
	Node · CRA · SPA	Serves static React + Tailwind	Horizontal, stateless
Backend	Uvicorn · FastAPI	35+ route modules · Motor · JWT	Horizontal, cookie-scoped
MongoDB	Motor (async)	Multi-tenant collections + indexes	Vertical + replica set
Object Store	S3-compatible	PDFs, brochures, audit exports	Native
LLM Gateway	Emergent-managed	Routes to Claude · GPT · Gemini	Provider-managed
Email	Resend (primary) + SMTP fallback	Transactional email	Provider-managed

As of v266 the primary sign-in path is email + password with a JWT session cookie. The default cookie lifetime is 7 days; if the user ticks **Remember me for 30 days**, the cookie is issued with a 30-day Max-Age *and* a device row is created in user_devices that the user can revoke from Profile → Security. Password reset uses a signed, single-use link over email.

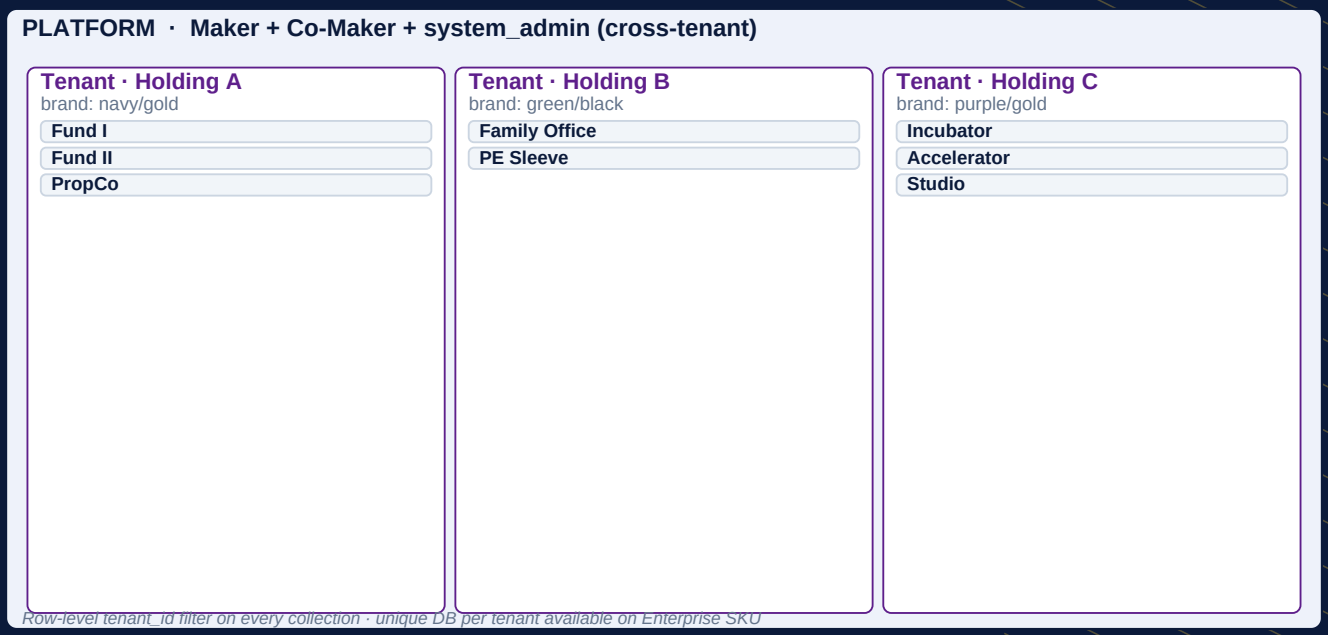


Control	Implementation
	bcrypt (cost 12) · never logged · never returned
Password policy	NIST 800-63B · 12+ chars · class-diversity · blocklist · HIBP k-anonymity check
Brute-force protection	5 failures / 15 min (per email) · 10 failures / 15 min (per IP) · 429 Retry-After
Session token	HS256 JWT · HttpOnly · Secure · SameSite=None · did claim per device
Session revocation	Bump password_changed_at → invalidates all outstanding JWTs
Device management	GET /api/auth/devices · DELETE /api/auth/devices/{id}
Password reset	Signed single-use link via Resend · 15-minute TTL
Audit trail	Every login (success + failure) written to audit_events + account_events

SO WHAT

Nothing about the auth stack is novel — that's the point. It is intentionally boring, aligned with NIST 800-63B, and independently auditable. The Remember-me cookie is device-scoped, meaning a stolen laptop can be revoked without invalidating the user's other sessions.

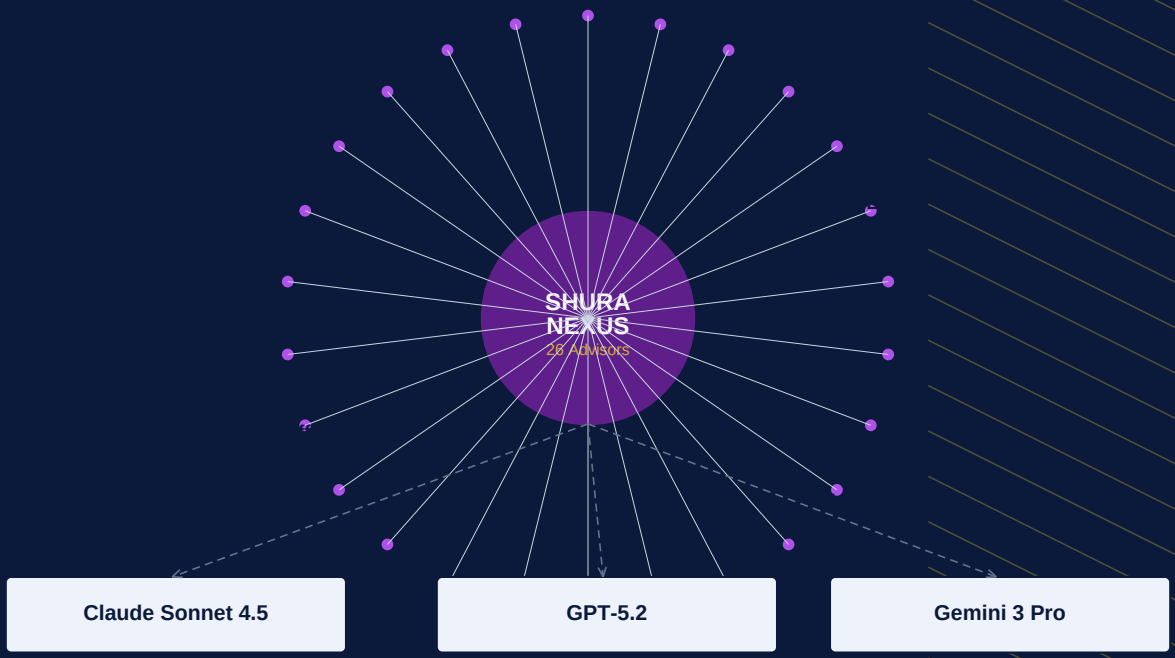
Every mutable collection carries a **tenant_id** and is filtered on every read. Cross-tenant access is only possible via the Maker principal, which is enforced at the middleware layer (not at the query layer). On the Enterprise SKU, each tenant receives a dedicated MongoDB database — same code path, different connection string.



Isolation vector	Standard SKU	Enterprise SKU
	Row-level tenant_id filter	Dedicated DB per tenant
Compute	Shared pod	Dedicated pod, optional VPC peering
	Per-tenant brand_name + palette	Full white-label + custom domain
Encryption keys	Platform-managed	BYO KMS / HSM
Audit export	Weekly PDF via Maker	Live SIEM stream (Splunk / Datadog)
SLA	99.9%	99.95% + credit

The Shura Nexus is a fan-out orchestration layer that presents a single question to 26 role-shaped advisors — Legal, Finance, Product, Security, Public Affairs, and so on — each configured with its own system prompt, model preference, and temperature. Outputs are streamed back to the client via Server-Sent Events. A final composer condenses the responses into a consensus opinion *and* a minority-dissent capture so the requester can see who disagreed and why.

Consensus + minority-dissent capture · streamed via SSE · per-advisor timeout + retry



Design decision	Rationale
	Failure isolation — Claude outage does not silence the whole Council
Streamed SSE with per-advisor timeout	UX responsiveness · avoids head-of-line blocking
Minority-dissent capture	Governance-grade transparency · avoids false consensus
Emergent LLM Gateway	Single billing surface · centralised safety filters · request tracing
No PII in prompts	Prompt scrubber removes emails, phone numbers, IDs before egress

Control	Implementation	Aligns with
	TLS 1.3 · HSTS · certificate pinning at ingress	ISO 27001 · SOC 2 CC6.1
Encryption at rest	AES-256-GCM · disk + object store	ISO 27001 · NDS3
	RBAC · tenant scoping · Maker-only cross-tenant	SOC 2 CC6.2 · ISO 27001
Auth	bcrypt + JWT + brute-force + HIBP	NIST 800-63B
	Append-only trail · every mutating request	SOC 2 CC7.2 · GDPR Art. 30
Backup	Daily snapshot · 30-day retention · quarterly restore drill	ISO 27001 A.12.3
	Dependabot · monthly pen test · CVE gate	SOC 2 CC7.1
Incident response	Runbook · 15-min ack · post-mortem published	ISO 27001 A.16
	Region-pinned deployment · buyer-selected	GDPR · NDS3 · UAE DPL
DPIA	Available on request · templated in Trust Center	GDPR Art. 35
	Public list · 30-day change notice	GDPR Art. 28

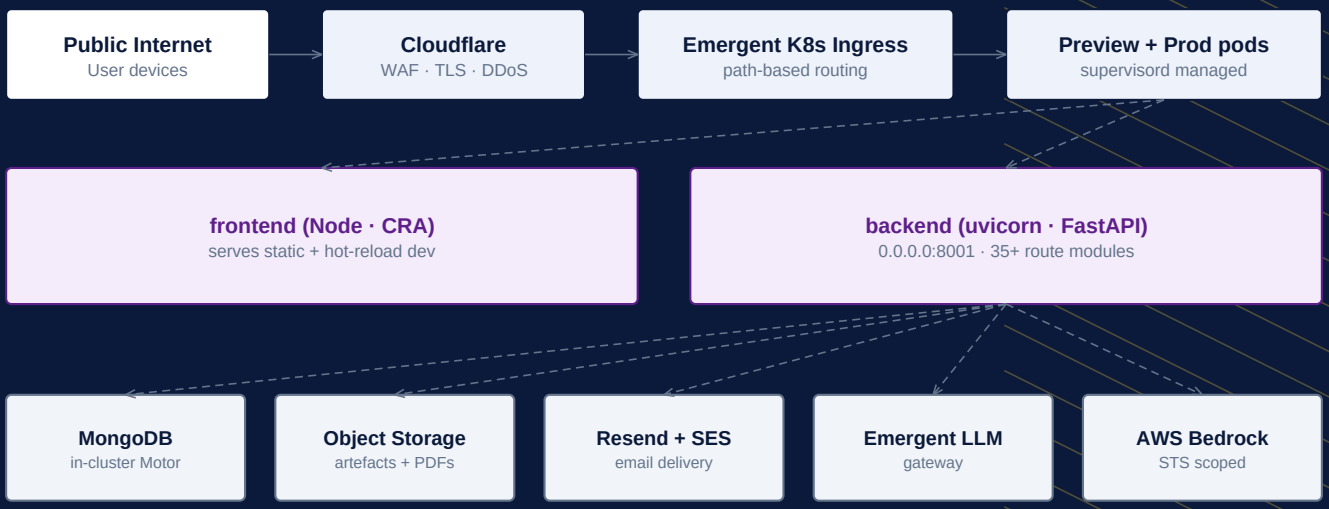
SO WHAT

Every row in this matrix corresponds to a specific control the platform enforces in code and a corresponding piece of evidence available via the Trust Center. Procurement teams can pull the evidence pack in one click at **/trust**.

Collection	Cardinality	Purpose	Retention
	10^3–10■	Identity + password + role + tenant	Lifetime of account + 90d
user_devices	10■–10■	Remember-me device sessions	30d after last-used
	10■–10■	Append-only mutation trail	7 years (regulatory)
account_events	10■–10■	Per-user activity feed	18 months
	10■–10■	Brute-force gate	15 minutes (TTL)
login_lockouts	10^3	Per-email / per-IP lockouts	1 hour (TTL)
	10^3	Single-use reset links	15 minutes (TTL)
digest_log	10■	Morning-digest idempotency	90 days
	10^1–10^3	Tenant metadata + branding	Lifetime + 5y
council_runs	10■	Shura Nexus outputs + dissent	12 months

SO WHAT

The retention policy is codified as TTL indexes on the collection itself — the database is the source of truth. There is no cron job that could silently forget to delete.



Preview and Production run identical container images; differences are limited to env vars and secrets.

Layer	Mechanism	Preview vs Prod
	Cloudflare-managed	Distinct hostnames
TLS	Cloudflare certificate	Distinct certs per hostname
	Cloudflare rules	Identical rulesets
Ingress	Emergent K8s controller	Path-based routing (/api → 8001)
	Supervisord (frontend + backend)	Identical container images
Data	MongoDB in-cluster	Distinct DBs · distinct credentials
	Emergent Panel	Preview and Prod have *separate* secret stores
Observability	Structured logs · error ring buffer	Identical shape

SO WHAT

The 'preview vs production' distinction is a secrets and DNS distinction — not a code or configuration one. That eliminates the 'works on my staging' class of outage in an evidence-backed way.

Every external integration has: (a) a named owner in the Maker's runbook, (b) a credential rotation cadence, and (c) a fallback / graceful-degradation path. The email transport is a two-tier design: Resend primary (via a verified domain with DKIM + SPF + DMARC), falling back to Hostinger SMTP if Resend returns 5xx.

Integration	Purpose	Auth	Fallback
	Claude · GPT · Gemini text + Nano Banana + Sora 2	Universal key	Model diversity across Council
AWS Bedrock (STS)	Regional inference (KSA / UAE / EU)	STS-scoped role	Emergent LLM
	Subscriptions + Tabby regional	API key rotated quarterly	Manual invoice
Resend	Transactional email (primary)	API key + verified domain	Hostinger SMTP
	Transactional email (fallback)	Password on TLS 465	Health banner + Maker alert
Cloudflare	DNS + WAF + TLS	API token	None (critical path)
	Persistent state	SCRAM-SHA-256	Nightly snapshot
Object storage	PDFs + audit exports	Signed URLs	Direct render fallback

SO WHAT

The email transport specifically is the ops-critical integration. Both Resend and Hostinger are health-checked on every send and surfaced on the dashboard so the Maker sees a red banner within 60 seconds of a delivery outage.

SLI	Target	Measurement
	≥ 99.9%	Cloudflare synthetic + `/api/health` heartbeat
Sign-in latency (p95)	≤ 800 ms	Structured log · rolling 24h percentile
	≥ 99.5%	Resend webhook + email-health ring buffer
Backup RPO	≤ 24 hours	Snapshot cron + restore drill quarterly
	≤ 4 hours	Restore drill quarterly
Vulnerability patch (critical)	≤ 24 hours	Dependabot + Maker escalation

Quarter	Theme	Highlights
	Trust & compliance	SOC 2 Type I readiness · public trust page uptime widget
Q2 2026	Council depth	Domain-specific advisor packs (real-estate, sovereign wealth)
	Regional footprint	KSA + UAE deployment · bilingual UI (AR + EN)
Q4 2026	Enterprise SKU GA	BYO KMS · SIEM stream · dedicated DB per tenant

This document is a point-in-time snapshot of the Vigilant Entities technical architecture as of v266. It is confidential and is provided under the terms of any executed NDA between the parties.

It is **not** a substitute for a formal security review, penetration test report, or DPIA. Those documents are available on request from the Maker.

Role	Contact
	admin@vigilantentities.com
Security disclosures	security@vigilantentities.com
	dpo@vigilantentities.com
Procurement / RFP	procurement@vigilantentities.com
	vigilantentities.com/trust

SO WHAT

For a fast enterprise onboarding, start with the Trust Center at /trust — the evidence pack there covers 80% of a standard procurement questionnaire without any back-and-forth.