PROGRAMME RESPONSE · 11 SECTIONS

DELIVERY RESPONSE

From Ketan Khairnar **To** SecComply Technologies **Date** 2026-04-13

From: Ketan Khairnar **To:** SecComply Technologies **Re:** Proposal response for SRS_Overwatch.md v1.0 (Draft — Greenfield), 2026-04-10 **Date:** 2026-04-13 **Ask on table:** Deliver all 17 systems in the SRS in **2 months**, agent-assisted development.

> Source documents (`SRS_Overwatch.md`, `AUTH_MODULE_DESIGN.md`, `AUTHZ_MODULE_DESIGN.md`) provided by SecComply on 2026-04-10 are treated as authoritative and unmodified throughout this response.

---

1. EXECUTIVE SUMMARY (READ THIS FIRST)

The honest answer: all 17 systems at the quality bar stated in the SRS is not a 2-month delivery — not at any team size, and certainly not at ours. Our team is 2 senior engineers (Ketan Khairnar + 1 senior architect with similar experience), augmented by 1–2 mid/senior contractors for the connector + scanner phase. We're not a 13-FTE consultancy; we won't price a 13-FTE proposal that we can't actually staff. With this team we can commit to:

- **Auditable v1.0-Core (12 weeks, W0–W12)** — production-grade, deployable, covering the auth/authz backbone the rest of the platform depends on, plus the compliance engine that makes the product *say what it does on the box*. Owned end-to-end by the two seniors.
- **v1.0-Extended (W7–W14, overlapping)** — connector and scanner work, with 1–2 contractors joining at W7 once the connector framework contract is frozen. Total programme: 12–14 weeks.

The driver is not lines of code — agents ship code fast. The driver is:

1. **Security-sensitive surface.** Auth, AuthZ, session revocation, tenant isolation, crypto, audit. These are not where we trade thoroughness for speed. They need design review, threat modelling, and a human in the loop.
2. **External integration breadth.** AWS + GCP + Azure + GitHub + GitLab + Entra + Google Workspace + Okta + generic OIDC/SCIM + SMTP + LLM. Every one has credential flows, rate-limit behaviour, and failure-mode quirks that swallow more calendar time than code time.
3. **Seed-data content.** 93 ISO controls, 61 SOC 2, 30 DPDP, atomic control library, Vendor Library (15+ vendors with pre-verified due-diligence), Policy Library (12+ templates with merge fields). This is GRC content work, not engineering — it needs a security/compliance SME, not an agent.
4. **Air-gapped on-prem.** Helm + Docker Compose + air-gapped mirror + backup/restore + upgrade procedure + observability catalog. Every Appendix-C TBD is weeks of operator-facing work.

Recommended shape below.

2. SCOPE INTERPRETATION — WHAT WE COUNTED

2.1 Systems in SRS (17 modules)

#	MODULE	SECTION	REQS	DEPTH IN SRS	REFERENCE DESIGN PROVIDED
1	Identity & Authentication	§3.1	42 (REQ-AUTH)	**Detailed**	`AUTH_MODULE_DESIGN.md` (~1700 lines, 19 chapters)
—	Authorization (paired with #1)	§3.1B	27 (REQ-AUTHZ)	**Detailed**	`AUTHZ_MODULE_DESIGN.md` (~1100 lines, 13 chapters)
2	Multi-Framework Compliance Readiness	§3.2	8	Summary	No
3	Atomic Control Library & Mapping	§3.3	8	Summary	No
4	Evidence Collection & Lifecycle	§3.4	10	Summary	No
5	Risk Management	§3.5	7	Summary	No
6	**Vendor Risk Management (TPRM)**	**§3.6**	**29 (REQ-VNDR)**	**Detailed**	No
7	**Document Control — Policies & SOPs**	**§3.7**	**36 (REQ-POL)**	**Detailed**	No
8	Awareness Training	§3.8	7	Summary	No
9	CSPM (AWS/GCP/Azure)	§3.9	9	Summary	No
10	SCM Security (GitHub/GitLab)	§3.10	6	Summary	No
11	Supply Chain Security	§3.11	5	Summary	No
12	Vulnerability Assessment	§3.12	5	Summary	No
13	Incident Management	§3.13	7	Summary	No
14	IAM Governance / UAR	§3.14	7	Summary	No
15	Internal Audit Execution	§3.15	6	Summary	No

#	MODULE	SECTION	REQS	DEPTH IN SRS	REFERENCE DESIGN PROVIDED
16	Executive Dashboard & Action Items	§3.16	5	Summary	No
17	Integrations Management	§3.17	7	Summary	No

Total functional REQs: 252 (excludes NFRs, safety, security, performance).

2.2 What you *also* asked us to build (platform floor)

These are non-negotiable dependencies of every module. The SRS calls them "first-party platform services":

- **BFF (Backend-for-Frontend)** — stateless JWT/session verify, RBAC enforcement, CORS, rate limiting, audit logging, request-ID propagation.
- **Self-hosted document DB (MongoDB)** — replica set *and* shardable. With per-tenant isolation, migrations, seed loader.
- **Self-hosted S3-compatible object store** — evidence, policies, reports. With malware signature scan on upload (REQ-SEC-11/REQ-EVID-9).
- **Transactional email service** — SMTP dispatcher, template engine, bounces, reminders, tokenised links.
- **Observability stack** — structured logs, metrics, traces, error telemetry, secret scrubbing (REQ-SEC-7). Catalog is a C-11 TBD.
- **KeyStore** — file-backed default with pluggable HSM/KMS interface (REQ-AUTH-32, REQ-SEC-5, C-9 TBD).
- **Shared design system** — light + dark mode, command menu, WCAG 2.1 AA primitives, empty/loading/error states.
- **Seed-data runtime** — versioned, replayable loader for frameworks, atomic controls, CSPM checks, SCM checks, vendor library, policy library.
- **Deployment artefacts** — OCI images, reference Helm chart, reference Docker Compose, air-gapped mirror instructions, backup/restore, upgrade playbook (C-6/C-7/C-12 TBDs).

The platform floor is roughly 30% of the total build. Most clients under-count it. We are not.

---

3. GAPS, AMBIGUITIES, AND DECISIONS NEEDED FROM CLIENT

These block or distort estimates. We cannot finalize a fixed-scope commitment without answers.

3.1 Blockers (must resolve before week 0)

#	GAP	WHERE	WHAT WE NEED
G-1	**Reference designs for modules beyond Auth/AuthZ.** Item 6 (Vendor) and Item 7 (Policy) are detailed by REQs but have no design doc. Items 2–5, 8–17 are summaries only.	§3.2–§3.17	Either (a) client commits those designs are emergent (we produce them) and estimate inflates, or (b) client supplies designs analogous to Auth/AuthZ by week 1.
G-2	**Seed-data authorship.** Who authors: (a) atomic control library + framework mappings (ISO 93, SOC 2 61, DPDP 30), (b) Vendor Library 15+ entries with **pre-verified** due-diligence docs, (c) Policy Library 12+ templates with merge fields, (d) CSPM check library, (e) SCM check library?	§3.2, §3.3, §3.6, §3.7, §3.9, §3.10	This is weeks of GRC-SME work. If SecComply supplies a compliance SME, we build the loader. If we author, add 3–4 SME-weeks.
G-3	**LLM provider for AI policy enrichment (REQ-POL-25..28).** Which provider? What data-processing agreement? What consent-dialog wording?	§3.7	Provider selection + DPA + prompt templates. If deferred, we ship the `PolicyEnricher` interface stub and mark feature-flagged off.
G-4	**Seat/scale reference dataset for NFRs.** REQ-PERF-2 cites "10k controls, 50k evidence, 1k findings per tenant" but no user/org count, no concurrency target, no scan-run throughput.	§5.1	Confirm reference load; otherwise we size for 10–50 orgs × 50 users × 1 concurrent scan.
G-5	**Malware scanner choice.** ClamAV is the obvious self-hostable option; anything else?	REQ-SEC-11, REQ-EVID-9, REQ-VNDR-22	Pick one (default ClamAV) or provide licence for commercial.
G-6	**Malicious-upload handling + evidence retention defaults.** REQ-EVID-4 per-file size cap is *TBD* (C-1).	§3.4, §6	Publish defaults.
G-7	**Air-gapped CSPM/SCM/IAM posture.** The SRS explicitly allows "manual-evidence-only mode" but does not confirm Phase-1 air-gapped demo customers. If air-gapped is a week-8 must-demo, we prioritise differently.	§2.7	Confirm air-gapped is week-8 critical or week-12 hardening.
G-8	**Impersonation hook for internal support.** REQ-AUTH-39 lists impersonation as Phase 2 but §4.4	§4.4, §3.1 non-goals	Confirm.

#	GAP	WHERE	WHAT WE NEED
	reserves `X-Impersonate-Org` header. Confirm the header+auth path is reserved-only (no runtime support) for v1.0.		
G-9	**Rate-limit numbers for public endpoints.** REQ-TRN-7 and vendor questionnaire token endpoint thresholds (C-8).	§3.8, §3.6	Publish numeric defaults.
G-10	**Design system source.** SRS says "SecComply shared design system" but doesn't state whether it exists, is public, or must be built.	§4.1	If it exists, point us at it. If not, +1 FE-week (component library bootstrap).

▸ 3.2 Appendix C items still open (we inherit these)

C-1 upload size, C-2 i18n loader, C-3 WCAG audit, C-4 framework roadmap, C-6 on-prem reference arch, C-7 backup/DR playbook, C-8 rate limits, C-9 KeyStore adapter spec, C-10 browser matrix CI, C-11 observability catalog, C-12 migration runner. Each is 0.5–2 dev-weeks of *documentation + artefact*, not just REQ text.

▸ 3.3 Ambiguity we will resolve ourselves (flagging for visibility)

- **Tenant data model per module.** SRS defines scope rules (cascaded vs instance) but not concrete collections/indexes per module. We will write the data-model spec during implementation; client review welcomed but not blocking.
- **Connector SDK surface.** We will pick a connector framework (likely gRPC service contracts + versioned manifest) so all scanners share rate-limit, retry, and credential-handling primitives. Out-of-scope to specify in SRS.
- **Frontend framework.** Keeping TS as you confirmed. Default: Next.js (App Router) + React 19 + shadcn/ui + TanStack Query. If you have a standing preference, name it by week 0.

---

4. TECHNOLOGY STACK RECOMMENDATION

User guidance captured: **TS is acceptable on the web side; containerised deployment; Go or Java preferred on the backend for stability.**

Full stack research: See our internal research layer — 14 MECE capability buckets (`buckets/01..14_*.md`), plus four synthesis docs in `synthesis/` (our internal stack reference, our internal integration notes, our internal stack matrix, our internal integration blueprint). The research is canonical; this section summarises.

4.1 Recommended stack (v2 — research-backed)

LAYER	OUR RECOMMENDATION	LICENCE	RESEARCH
Web app	TS + **Next.js 15 App Router** + React 19 + **shadcn/ui** (Radix + Tailwind v4) + **TanStack** (Table/Query/Virtual) + **Recharts** + **visx** (heatmaps) + **react-hook-form** + **zod** + **cmdk**	MIT across	C14
BFF + core services	**Go** (chi/echo + `net/http` + `slog`)	—	—
IAM / SSO	**Keycloak 26.x** sidecar (OIDC/SAML/MFA/WebAuthn) — BFF owns sessions	Apache-2.0	C1
AuthZ	Build in-Go `pkg/authz` (7-step pipeline, 18 roles, SoD, step-up AAL)	our code	C2
Document DB	**Percona Server MongoDB 7** (SSPL-1 + Percona additional grant) + Percona Operator + PBM	SSPL-1 + Percona grant	C3
Relational DB (Keycloak + GlitchTip)	**shared-postgres** (Postgres 16, schema-per-service)	PostgreSQL licence	C1/C13
Secrets & keys	File-backed Go **KeyStore** (default) · **OpenBao 2.x** enterprise adapter	MIT / MPL-2	C4
Object store	**MinIO** sidecar (unmodified OCI, aggregate distribution) · fallback **SeaweedFS**	AGPL-3 / Apache-2	C5
Session / cache / queue / rate-limit / SSE	**Valkey 8.2** + **Asynq** + gocron/v2 + redis_rate	BSD-3 / MIT	C6
Email	`overwatch-mailer` Go service on **wneessen/go-mail** + stdlib templates + MJML-at-build + HMAC-signed envelope tokens · SMTP relay out only	MIT	C7
Malware scan	**ClamAV** `clamd` sidecar (GPL-2, aggregate) + YARA layer · async via MinIO-notification → Asynq → scan worker · `cvdupdate`-mirror for air-gap signatures	GPL-2 / BSD-3	C8
Policy editor + diff + export	**TipTap v2 MIT core** (no Pro) + **prosemirror-changeset** + **diff-match-patch** + **Gotenberg 8** · snapshot-per-version	MIT / Apache-2	C9
Connector framework	Build ~1 kLOC Go framework on official SDKs (aws-sdk-go-v2, go-github, msgraph-sdk-go, okta-sdk-golang/v5, etc.)	Apache-2 / MIT / BSD	C10

LAYER	OUR RECOMMENDATION	LICENCE	RESEARCH
Security check libs	**Prowler v5** metadata donor + Go evaluators · **OSV-Scanner** + OSV-offline mirror · **Syft** · **SARIF 2.1.0**	Apache-2 / CC-BY-4	C11
Observability	**OTel SDKs + Collector** (redactionprocessor) + **VictoriaMetrics** family + **VictoriaLogs** + **Jaeger v2** + **GlitchTip** · simplified alt **SigNoz**	Apache-2 / MIT	C12
Deployment	**ko** (Go) + **apko/melange** (non-Go) + **Cosign** keyful + **Syft** SBOMs · umbrella Helm chart · **Zot** air-gap registry · **ESO** + SOPS+age for secrets · nightly **restore-verify CI**	Apache-2 across	C13
Audit log	`pkg/audit` BLAKE2b hash-chain + Ed25519 signed chain-head every 5 min via KeyStore · WORM export to MinIO Object-Lock Compliance bucket · closes REQ-SAFE-3 mechanism gap	our code	Blueprint §3.3

Aggregate-distribution licence notes (posture documented in release memo): MinIO (AGPL-3 sidecar — §13 network-copyleft does not trigger because we ship MinIO unmodified and interact only via the standard S3 protocol), ClamAV (GPL-2 sidecar — no source-level linkage; ClamAV `clamd` runs as a separate process and we communicate via the documented INSTREAM TCP protocol), Percona Server MongoDB (SSPL-1 + Percona additional grant — the additional grant nullifies SSPL §13 for our deployment shape because PSMDB powers the customer's own workload rather than being offered as a third-party database service). All three shipped as unmodified upstream OCI images, standard-protocol-only interaction, LICENSE/NOTICES preserved. This is the standard industry posture used by GitLab self-hosted, Gitea, Forgejo, and Harbor; no public legal challenge has been brought to date. Full legal analysis with grant text inline: `LEGAL_POSTURE.md`.

Explicitly rejected (for the record). Redis 8 (AGPL-3 licence complication → Valkey) · Grafana Labs stack Loki/Tempo/Mimir/Grafana (AGPL-3 bundling risk → VictoriaMetrics) · CloudQuery core (CQ Community Licence 2024) · Steampipe runtime (AGPL since v0.21) · Bitnami production charts (Aug-2025 relicence) · Sentry self-host (FSL/BUSL → GlitchTip) · Dragonfly (BSL). Details in our internal stack matrix.

▸ 4.2 Java alternative (as you mentioned)

Java (Spring Boot 3 + Spring Security + Micronaut) is equally stable and has deeper SAML/SCIM enterprise libraries. **Trade:** larger container images, slower cold start, heavier resource footprint for air-gapped deployments. Go wins on ops ergonomics; Java wins on library maturity in the identity space — but Keycloak *is* Java, so we already get that maturity as a sidecar without the footprint cost. **Recommendation: Go**, unless your team has deep Java and you plan to hire Java engineers for maintenance.

4.3 What we build in-house

IN-HOUSE DELIVERABLE	ENG-WEEKS	NOTES
`pkg/authz` (7-step pipeline + 18 roles + SoD + step-up)	7.0	Library core is 2.5 ew; end-to-end incl. BFF integration + SoD + test corpus + audit schema = 7
`KeyStore` (file-backed + rotation + adapter interface)	2.5	Mid-point of 2–3 ew range
`pkg/connector` framework (core)	3.5	Framework skeleton only
Connector per-provider shims (10 providers)	9.0	Counted under per-module summaries in §5.3
`overwatch-mailer` Go service (~1.5 kLOC)	1.5	
Migration runner (Helm pre-install Job + CLI)	0.5	~400 LOC Go
Seed-data loader + content-migration table	1.0	Includes per-tenant `seed_version` state machine
Backup CLI (`overwatch-backup`)	1.0	PBM + mc mirror + RDB + kc.sh export + pg_dump orchestration
Automated restore-verify CI pipeline	0.5	REQ-SAFE-5 mechanism
`pkg/audit` hash-chain + signed-head pinning	1.0	REQ-SAFE-3 mechanism
`pkg/live` SSE-over-Valkey transport	0.5	Replaces 2s polling for scan-run + dashboard
CSP-nonce middleware + SRI manifest + report endpoint	0.5	Next.js 15 + shadcn SRI
PII-scrubbing policy catalog	0.25	Seeds OTel redactionprocessor
Total in-house platform (bucket-level)	**28.75**	

`pkg/tenancy` (ancestor-ID resolution, scope inheritance) is folded into the BFF skeleton in §5.1 below.

---

5. EFFORT ESTIMATE — MODULE BY MODULE

Assumption. Agent-assisted team, senior engineers. "Engineer-week" = 1 senior engineer × 5 working days, with agents pair-coding. Estimates include: implementation, unit tests, integration tests against real services where possible, module-level docs, and BFF route wiring. They **exclude**: seed-data authorship (tracked separately), external threat modelling, WCAG audit, security pen-test.

What changed in v2 (post-research). The research reconciled ~224 eng-weeks of OSS adoption savings against ~28.75 eng-weeks of in-house platform code. Adoption savings landed in the platform floor (now 22 ew vs original 22.5 — effectively flat; we didn't save much here, but we *found* gaps we would otherwise have hit in W7) and the summary modules (reduced by ~3 ew each for CSPM/SCM/IAM/observability-wiring thanks to official-SDK reuse + OSV/Prowler metadata donors). Net effect: total engineering is roughly unchanged (~80 ew), but **risk is materially lower** — we enter W0 with named libraries, pinned versions, signed container bases, and no "we'll figure this out" gaps.

5.1 Platform floor (shared, unavoidable) — v2

WORK ITEM	ENG-WEEKS	OSS PICK
Repo bootstrap, CI, container build pipeline with **ko + Cosign + Syft** + image signing	1	C13
BFF skeleton: routing, session middleware (Valkey lookup), RBAC middleware (`pkg/authz`), CORS, rate limit (redis_rate), audit, request-ID, OTel instrumentation	2	—
`pkg/tenancy` + entity hierarchy primitives (folded into BFF skeleton above)	—	—
`pkg/authz` library (7-step pipeline, role catalog, SoD) — library-core milestone	2.5	C2 (end-to-end 7 ew — rest lives in BFF skeleton + modules)
`pkg/audit` append-only **hash-chained** writer + signed head + WORM export to MinIO Object-Lock	2	C3 + C4 + C5
`pkg/connector` framework (core) + scan-run scaffolding + Asynq integration	1.5	C10
Seed loader + migration runner + **content-migration table**	1.5	C13
Object storage wrapper + **MinIO notification → Asynq → ClamAV** scan pipeline	1	C5 + C6 + C8
`overwatch-mailer` service + MJML-at-build templates + HMAC-signed tokens + reminder cron	1.5	C7
`KeyStore` (file-backed) + pluggable interface + **OpenBao adapter stub**	1.5	C4
Observability: OTel Collector + PII scrubbing catalog + VictoriaMetrics/Logs + Jaeger + GlitchTip + log-schema + metric catalog (REQ-AUTH-36)	1.5	C12
Umbrella Helm chart + first-party subcharts (Valkey, MinIO, ClamAV, Gotenberg, Keycloak, OTel, VM stack, shared-postgres) + Docker Compose + `overwatch-backup` CLI + restore-verify CI	2	C13
Air-gapped image tarball (skopeo sync + Zot + SBOM bundle) + operator runbook (C-6) + legal memo (MinIO/ClamAV/PSMDB aggregate posture)	1.5	C13
Design system — shadcn/ui bootstrap + Tailwind v4 tokens + light/dark + command menu + empty/loading/error states + CSP-nonce middleware + SRI manifest	2	C14

WORK ITEM	ENG-WEEKS	OSS PICK
`pkg/live` SSE-over-Valkey (replaces scan-run polling)	0.5	C6
Platform subtotal	**22.0 eng-weeks**	

▸ 5.2 Detailed modules (SRS provides full spec) — v2

Savings from **Keycloak adoption** (~20 ew protocol implementations we don't write), **TipTap + Gotenberg + prosemirror-changeset** (~22 ew editor+diff+export we don't write), and **Prowler metadata donor** (~33 ew of rule content we don't author — savings land mostly in §5.3 Extended modules and seed-data SME, not here).

MODULE	SCOPE NOTES (OSS LEVERAGE IN)	ENG-WEEKS
#1 Auth (REQ-AUTH 1-39)	**Keycloak** for OIDC/SAML/WebAuthn/TOTP/magic-link protocol surface; Overwatch owns session store (Valkey), entity hierarchy, break-glass rehearsal state, step-up payload-binding, API-token issuance, SCIM receiver, audit event catalog.	**6** (down from 8 — Keycloak adoption saves magic-link + OIDC/SAML + WebAuthn stack)
#1B AuthZ (REQ-AUTHZ 1-25)	Consumes `pkg/authz` (§5.1). Module-level work: 18-role permission matrix loader, role-assignment UI, SoD exception workflow, external-auditor campaign scope, API-token permission intersection, control-ownership attribute.	**3.5**
#6 Vendor (REQ-VNDR 1-29)	Intake, classification, questionnaire templates, **Excel import** (xlsx), tokenised external questionnaire flow with save-and-resume (uses `overwatch-mailer` HMAC tokens), due-diligence doc mgmt with expiry alerts (uses **Asynq Periodic**), composite scoring, Vendor Library (seed), integration-as-vendor auto-link, reassessment cycles, audit.	**5.5**
#7 Policy (REQ-POL 1-36)	**TipTap MIT** rich-text + **prosemirror-changeset** version diff + **Gotenberg** PDF/DOCX export, multi-stage approval, within-record SoD (pkg/authz), SecComply Policy Library w/ merge fields (seed), AI enrichment (pluggable `PolicyEnricher` interface — provider per G-3), acknowledgement campaigns, exception/waiver workflow, scheduled review cycles, audit.	**5** (down from 6 — editor/diff/export adoption)
Detailed subtotal		**20.0** (down from 23)

▸ 5.3 Summary-only modules (SRS is thin — design emerges) — v2

These are "summary depth" in the SRS. We need to design as we build. Estimates still carry 20% design overhead — but **OSS leverage in CSPM/SCM/IAM/Supply-Chain reduces per-module effort ~25–**

40%.

MODULE	SCOPE NOTES (OSS LEVERAGE IN)	ENG-WEEKS
#2 Compliance Readiness	Framework scoring, cycle numbering, dashboard aggregation	2
#3 Atomic Control Library	Seed-driven library, control status, client questions, auto-check eval	2.5
#4 Evidence Collection	Upload → **MinIO** → **ClamAV** async scan → auto-evidence emitter → lifecycle + expiry + audit (hash-chained)	2.5 (down from 3 — malware + audit pipeline reused from platform floor)
#5 Risk Management	Register, 5×5 heatmap, acceptance workflow, step-up on approval, audit	2
#8 Awareness Training	Courses, campaigns, public tokenised URL (reuses `overwatch-mailer` HMAC token scheme), quiz tracking	2 (down from 2.5 — HMAC token scheme reused)
#9 CSPM (AWS, GCP, Azure)	3 connectors using **official provider SDKs** (aws-sdk-go-v2, cloud.google.com/go, azure-sdk-for-go) + `pkg/connector` framework; **Prowler metadata** drives check library (seed); scan runner + score gauge + service×account heatmap.	3.5 (down from 4.5 — framework + Prowler metadata cut per-provider effort)
#10 SCM Security (GitHub, GitLab)	2 connectors (**go-github v70** + **gitlab.com/gitlab-org/api/client-go**); ~20 seed checks inspired by **OSSF Scorecard**; repo enumeration; rate-limit (built into aws-sdk-v2-style retry)	2 (down from 2.5)
#11 Supply Chain	Ingests SCM dep alerts; **OSV-Scanner + OSV-offline mirror** for CVE enrichment; **Syft** SBOM for manual uploads; CVE feed air-gap mirror already in platform floor	1.5 (down from 2 — OSV + Syft adoption)
#12 Vulnerability Assessment	Upload pentest/SAST/DAST reports; **SARIF 2.1.0 primary parser** + 6-8 **DefectDojo parsers ported to Go** for Nessus/Burp/OpenVAS/ZAP/Qualys/Nuclei; finding CRUD + lifecycle + linkage	2.5 (up from 2 — porting parsers is real work, but still net save vs raw authoring)
#13 Incident Management	Severity, lifecycle, MTTR, breach-notification countdown, post-mortem	2.5
#14 IAM Governance / UAR	4 IdP connectors (**msgraph-sdk-go** for Entra, **google.golang.org/api/admin/directory** for Workspace, **okta-sdk-golang/v5**, **coreos/go-oidc + custom SCIM client** for generic); directory pull,	3 (down from 4 — official SDKs + connector framework reuse)

MODULE	SCOPE NOTES (OSS LEVERAGE IN)	ENG-WEEKS
	org chart, over-permissioned detection, UAR campaigns	
#15 Internal Audit	Planning, checklist, findings, PDF/CSV export (**Gotenberg** reused)	2
#16 Dashboard & Action Items	Cross-module aggregation, **Recharts + visx** charts, unified queue, PDF snapshot (**Gotenberg**)	2 (down from 2.5 — chart libs adopted)
#17 Integrations Management	Central add/test/scan/disconnect, credential encryption via `KeyStore`, vendor auto-link	2
Summary subtotal		**30.0 eng-weeks** (down from 36)

▸ 5.4 Seed-data authorship (GRC SME, not engineer)

ITEM	SME-WEEKS
Atomic control library + ISO 27001:2022 (93) + SOC 2 (61) + DPDP (30) mappings	2
Vendor Library — 15 entries with pre-verified SOC 2/ISO docs references	1
Policy Library — 12 templates with merge fields	1.5
CSPM check library (seed)	1
SCM check library (seed)	0.5
SME subtotal	**6 SME-weeks**

▸ 5.5 Cross-cutting not yet counted

ITEM	ENG-WEEKS
WCAG 2.1 AA accessibility pass (C-3)	1.5
E2E browser matrix CI (C-10)	1
Load testing to REQ-PERF thresholds, profiling, index tuning	2
Security review + threat model + pen-test remediation buffer	2
Operator documentation (installation, backup/restore, upgrade, key rotation)	1.5
Cross-cutting subtotal	**8 eng-weeks**

5.6 Grand total — v2 (post-research)

BUCKET	V1 ENG-WEEKS	V2 ENG-WEEKS	Δ	DRIVER
Platform floor (§5.1)	22.5	22.0	-0.5	Net-flat — adoption savings roughly offset newly-found mechanism gaps (audit hash-chain, restore-verify CI, SSE, CSP/SRI, PII catalog, content-migration); `pkg/tenancy` folded into BFF skeleton row
Detailed modules (Auth + AuthZ + Vendor + Policy)	23.0	20.0	-3.0	Keycloak adoption (-2) + TipTap/Gotenberg adoption (-1)
Summary modules (13 modules)	36.0	30.0	-6.0	Official SDKs + Prowler metadata + OSV + Syft + chart libs
Cross-cutting	8.0	8.0	0	unchanged
GRC SME (seed data)	6.0	6.0	0	Same authorship work; OSS donors reduce engineering effort, not SME effort
Total engineering	**89.5**	**80.0**	**-9.5**	~11% reduction
Total SME	**6.0**	**6.0**	0	

Calendar implication (v2 — corrected to actual team shape):

The original SRS asked for "2 months, agent-assisted". Honest accounting requires we tell you the team shape *we* are bringing — not a hypothetical 13-FTE shop.

- **Core team (W0–W14): 2 senior engineers** — Ketan Khairnar (delivery + backend lead) and 1 senior architect (similar experience profile, security + backend). Both write production code; both review each other's commits. No project manager layer; no separate QA/DevOps headcount.
- **Extended-phase addition (W7–W14, overlapping with later Core weeks): +1 to +2 mid/senior engineers** brought in for the connector + scanner work, where the per-provider SDK shims parallelise cleanly. Peak team during W10–W14: 3–4 engineers.
- **GRC SME (6 SME-weeks across the full programme):** SecComply-named, not on our headcount.
- **External security review (2 person-weeks):** independent firm, fractional engagement at W11 + W14.

Eng-week math against this team:

- v1.0-Core (W0–W12): 80 ew / 2 engineers ≈ 40 weeks straight-through. Compressed to **~12 weeks** via (a) **0.75× agent-assistance multiplier** on mechanical work — CRUD, BFF wiring, seed loaders, test scaffolds, connector shims (~25 ew of the 80 net to ~19), (b) parallelisation of platform-floor and product-module streams (the gold-seam interfaces are designed for this — see §4.3), (c) extension contractors joining at W7 to start Extended work in parallel.
- v1.0-Extended (W7–W14, overlapping): 32 ew. Contractors ramp from W7, peak parallelism W10–W14. Most Extended modules complete by W14.

- **Total programme: 12–14 weeks** (Core ships RC at W12; Extended completes W14 at the latest), versus the original ask of "8 weeks for everything".

Why we are *not* committing to the original 8-week ask. Ketan + 1 architect, even agent-paired, cannot honestly deliver 80 ew of Core *plus* 32 ew of Extended in 8 weeks. Sustained 5 ew/wk per person is possible for short bursts, not across 8 weeks against security-sensitive code (crypto, session, authz, tenancy, audit-chain — all `no agent autonomous commit` zones per §11). Offering 8 weeks would be the kind of commitment that gets re-negotiated mid-build; we'd rather price the honest 12–14 week calendar now.

What this gives up vs. a hypothetical 13-FTE engagement: schedule float. If W6 surfaces an unanticipated load-test issue, we can't backfill with a fresh engineer the next Monday. The trade we make: the same two people own every line of the platform floor + Auth + AuthZ + audit-chain — no handoff cliffs in the security-sensitive zone. Extension contractors only touch the contractor-safe zone (per-provider SDK shims, CRUD modules) where the framework contracts they consume are frozen by W4.

What the research savings buy us at this team size: they become *risk reduction* rather than headcount reduction. We enter W0 with named libraries + pinned versions + signed bases, aggregate-distribution legal posture documented (no W11 legal surprise), tamper-evident audit log specified at mechanism level (closes REQ-SAFE-3), air-gap mirror patterns named, frozen interface boundaries for OSS-churn insulation, and a simplified-alternative observability stack for single-node customers.

---

6. RECOMMENDED DELIVERY PLAN (12 WEEKS, V1.0-CORE)

This is what we can **commit to** with the 2-engineer core team. The original SRS asked for 2 months; honest delivery at this team size is 3 months for the Core scope below. Anything beyond is best-effort.

▸ 6.1 Scope inside 12 weeks (v1.0-Core, audit-ready)

In:

- Full platform floor (§5.1)
- **Item 1 — Auth + AuthZ** (detailed, full spec)
- **Item 6 — Vendor Management** (detailed, full spec)
- **Item 7 — Document Control / Policies** (detailed, full spec, *AI enrichment feature-flagged off*)
- **#2 Compliance Readiness** + **#3 Atomic Control Library** + **#4 Evidence** + **#5 Risk** — these are the *compliance engine core*. Without them, items 6/7 have nowhere to write auto-evidence.
- **#17 Integrations Management** — the central control plane; needed even if only 1–2 providers light up in Core.
- **#16 Dashboard & Action Items** — thin v1 over whichever modules shipped.
- WCAG AA pass on shipped modules.
- Helm chart + Docker Compose + backup/restore playbook.

Out (moved to v1.0-Extended, weeks 7–14, overlapping with later Core weeks):

- **#9 CSPM** (3 cloud connectors — each is a mini-project)
- **#10 SCM Security**
- **#11 Supply Chain**
- **#12 Vulnerability Assessment**
- **#13 Incident Management**
- **#14 IAM Governance / UAR**
- **#15 Internal Audit**
- **#8 Awareness Training**
- Air-gapped tarball + mirror instructions (Core ships online-installable first)
- AI policy enrichment (feature-flag off)

Defensible rationale. The client gets a deployable ISMS that: onboards an org, issues identities, enforces RBAC, runs compliance assessments against seeded frameworks, manages vendors with external questionnaires, manages policies with acknowledgements — **the auditable core**. Scanners and extended operational modules are easy to prioritise after Core is breathing.

6.2 Week-by-week (v1.0-Core, 12 weeks at 2 senior engineers)

WEEK	MILESTONE
W0 (pre-kick)	Answer G-1..G-10. Sign off stack. Design system decision. SME assigned. Seed-data contracts agreed.
W1	Platform floor part 1: repo, CI, container build, BFF skeleton, tenancy lib, DB/object store wiring, KeyStore. Both engineers paired on platform-floor primitives.
W2	Platform floor part 2: email stub, seed loader, observability catalog, audit-chain writer skeleton. AuthZ resource/action/role catalog. **Checkpoint: G-Foundation.**
W3	Auth pillar 1–5 (methods + policy + HRD + bootstrap). AuthZ 7-step pipeline + SoD. Atomic Control Library skeleton. **Architect splits to AuthZ; Ketan continues platform-floor + Auth integration.**
W4	Auth pillar 6–10 (SSO OIDC/SAML, magic link, session store, step-up). AuthZ end-to-end + audit-chain integration. Compliance Readiness skeleton. Seed: ISO 27001 + SOC 2 first-cut. **Checkpoint: G-AuthZ.**
W5	Auth pillar 11–15 (break-glass, SCIM, password+MFA, recovery, email change, service accounts, API tokens). Evidence module. **Checkpoint: G-Auth-Security-Review (internal).**
W6	Auth pillar 16–19 (key lifecycle, audit catalog, metrics, NIST 800-63 conformance). Risk module. Seed: DPDP, atomic control library, framework mappings signed off. **Checkpoint: G-Compliance-Core.**
W7	**Item 6 Vendor — 50%** (records, classification, scoring, library, intake flow). Design system pass on shipped screens.
W8	**Item 6 Vendor — 100%** (Excel import, tokenised external questionnaire, due-diligence expiry alerts, integration-as-vendor, reassessment).
W9	**Item 7 Policy — 60%** (editor, library, merge fields, versioning, diff, approval workflow).
W10	**Item 7 Policy — 100%** (acknowledgement campaigns, exception/waiver, export, review cycles). Integrations Management. Dashboard + Action Items. Malware scan wiring. **Checkpoint: G-Module-Wave.**
W11	WCAG AA pass. Load test (REQ-PERF-1..4). External security review pass + pen-test remediation. Helm chart + Docker Compose hardening. Backup/restore dry run. Operator docs. **Checkpoint: G-WCAG + G-Perf + G-Security.**
W12	Release candidate. UAT with SecComply. Upgrade path rehearsal. Final security sign-off. **GA RC1 delivered. G-UAT.**

▸ 6.3 Weeks 7–14 (v1.0-Extended, overlapping with later Core weeks)

Extension engineers join at W7 — once `pkg/connector` framework contract is frozen (W4 G-AuthZ gate) — and work in parallel with Core's W7–W12 module wave. They touch only the contractor-safe zone: per-provider SDK shims, CRUD modules, scanner check libraries. They do *not* touch crypto, session, authz, tenancy, or audit-chain code. Peak team during W10–W14: 3–4 engineers.

WEEK	DELIVERY (EXTENSION CONTRACTORS)	PARALLEL CORE (KETAN + ARCHITECT)
W7	Contractor onboarding · `pkg/connector` framework walkthrough · Awareness Training scaffolding	Vendor — 50%
W8	CSPM — AWS connector + check library (Prowler metadata seed)	Vendor — 100%
W9	CSPM — GCP connector. SCM — GitHub	Policy — 60%
W10	CSPM — Azure connector. SCM — GitLab. Supply Chain (OSV + Syft)	Policy — 100% + Integrations + Dashboard
W11	Vulnerability Assessment (SARIF + DefectDojo parsers). Incident Management	WCAG + Load + External security review
W12	IAM — Entra + Google Workspace connectors	RC1 + UAT (Core ships)
W13	IAM — Okta + generic OIDC/SCIM. UAR campaign engine	Internal Audit reporting framework
W14	Internal Audit screens. Air-gapped tarball + mirror. AI policy enrichment (if provider selected)	UAT round 2 + Extended sign-off

Why overlapping works. The platform-floor + Auth + AuthZ + Compliance Engine Core (W1–W6) produces frozen contracts: `pkg/connector` v1, `pkg/audit` event schema, RBAC middleware, evidence emit() API. Once those are frozen at W4–W6, contractors can build CSPM/SCM/IAM/VA modules against them without touching the security-sensitive code. The two seniors keep building Core (Vendor + Policy + Integrations + Dashboard) on the same locked contracts — no merge conflicts, no API churn.

Why this is not 8 FTE × 2 weeks. It's **2 seniors for 12 weeks plus 1–2 contractors for 8 weeks** (W7–W14). Same eng-week budget, very different headcount profile, and crucially: same two people own every commit in the security-sensitive zone the entire programme.

---

7. TEAM SHAPE

A small senior team for Core. A small contractor extension for the Extended phase. No project manager layer; no separate QA/DevOps headcount — those responsibilities sit on the seniors and are agent-paired where the work is mechanical.

ROLE	PERSON	WINDOW	SCOPE
Delivery + backend lead (full-stack senior)	**Ketan Khairnar**	W0–W14	Programme delivery, BFF, platform floor, Auth, audit-chain, client comms, weekly checkpoints. Primary client contact.
Senior architect (security + backend)	TBD (similar profile)	W0–W14	AuthZ pipeline, KeyStore, tenancy lib, security review, reference designs. Pairs with Ketan on every commit in the security-sensitive zone.
Mid/senior engineer (contractor)	+1 (sometimes +2)	W7–W14	Per-provider connector shims (CSPM, SCM, IAM/UAR), CRUD modules (Awareness Training, Internal Audit), seed data loaders. **Touches no security-sensitive code.**
GRC / Compliance SME	SecComply-named	W0–W6 (6 SME-weeks)	Seed data authorship, framework mappings, Vendor/Policy Library curation, audit sign-off. Not on our headcount.
External security review	Independent firm	W11 + W14	2 person-weeks fractional. Threat model + pen-test against shipped surface.
Frontend	Shared between Ketan + Architect	W0–W14	Both have Next.js / TS experience; design system bootstrapped with shadcn/ui + Tailwind v4. WCAG AA pass at W11.

Honest commitments at this team size:

- **No headcount fall-back.** If one of the two seniors is unavailable for an extended period (illness, family emergency), the calendar slips. We carry no bench.
- **No QA team.** Tests are written by the engineers writing the feature. Load test + WCAG audit are scheduled events at W11, owned by Ketan.
- **No DevOps team.** Helm + Compose + air-gap packaging owned by the architect, paired with Ketan for the air-gap tarball at W14.
- **Design system + WCAG.** Bootstrapped on shadcn/ui (WCAG-native) at W2; no separate designer headcount until W11 polish pass (where we contract in for ~3 days if needed).

What this is not: a 13-FTE delivery team. What it is: two senior engineers who own every line, plus contracted hands for the connector wave where the work is well-scoped.

---

8. RISK REGISTER (TOP 10)

#	RISK	LIKELIHOOD	IMPACT	MITIGATION
R-1	**Auth/AuthZ pen-test surfaces defects late**	M	H	Internal security review at W4 + W7; pen-test buffer in W7.
R-2	**External IdP / cloud provider quirks during integration tests**	H	M	Allocate 20% buffer on connectors; test against at least 2 live tenants per provider.
R-3	**Seed-data quality (framework mappings, Vendor Library verified docs)**	H	H	Assign SME W0; sign off mappings by W2. Defects here mean audit failure downstream.
R-4	**Design system doesn't exist** (G-10)	M	M	+2.5 eng-weeks in W1; use shadcn/ui base. Rework risk low.
R-5	**Air-gapped requirement surfaces late as week-8 must-have**	M	H	Freeze via G-7 in W0. Otherwise it's a 2-week hit.
R-6	**LLM provider / DPA delays AI policy enrichment**	H	L	Ship `PolicyEnricher` interface + stub; feature-flag off; unblock via Extended.
R-7	**WCAG audit finds systemic issues**	M	M	Audit at W7, not W8. Primitives chosen (shadcn/ui + Radix) are WCAG-native.
R-8	**Load test blows REQ-PERF targets on Mongo aggregation pipelines**	M	M	W7 load test. Index tuning buffer. Shard rubric in Helm.
R-9	**Cross-module coupling creeps** (e.g., Evidence depends on everything)	M	M	Enforce "Instance vs Cascading artifact" contract from W1. Module boundaries owned by Tech Lead.
R-10	**Agent-generated code passes tests but fails threat model**	H	H	Security-sensitive files (crypto, session, authz, tenant isolation) = **no agent autonomous commits.** Human-written + agent-assisted, not agent-owned.

9. COMMERCIALS — THREE OPTIONS, SPONSOR-PICKABLE

Not a formal SOW — scaffolding for the commercial conversation. All three options ship the same scope and the same team; they differ in **how risk is shared between SecComply and us** and **how payment is sequenced**. Blended senior engineering rate, programme-management uplift, and external security review fee are placeholders to be filled in once the sponsor names the rate band.

▸ 9.1 The volume being priced

Same in all three options.

BUCKET	PERSON-WEEKS	NOTES
Senior engineering (W0–W14)	2 seniors × 14 wks = **28 pw**	Ketan + Architect, full programme. The owners of the security-sensitive zone.
Contractor engineering (W7–W14)	~1.5 avg × 8 wks = **12 pw**	1–2 mid/senior engineers brought in for the connector + scanner wave. Peak 2 contractors W10–W14.
GRC SME (seed-data authorship)	**6 SME-weeks**	Frameworks ×3, atomic controls, vendor library ×15, policy library ×12. SecComply-named.
External security review	**2 person-weeks**	Independent firm; pen-test + threat-model sign-off at W11 + W14.
Programme total	**40 pw + 6 SME-wk + 2 ext-review-pw**	Senior pw priced at senior rate; contractor pw priced at contractor rate.

The three options below differ on **payment shape and risk allocation**, not on this scope.

▸ 9.2 Option A — Fixed-bundle pricing

Two pre-priced bundles, simple PO, scope-locked.

BUNDLE	CALENDAR	ENGINEERING PW	WHAT SHIPS
Core bundle	W0–W12 (12 weeks)	24 senior pw + 4 contractor pw + 6 SME-wk + 2 ext-review-pw	v1.0-Core: identity, authz, compliance engine, vendor, policy, evidence, dashboard, audit-ready release.
Core + Extended bundle	W0–W14 (14 weeks)	28 senior pw + 12 contractor pw + 6 SME-wk + 2 ext-review-pw	Above + CSPM (AWS/GCP/Azure), SCM (GitHub/GitLab), supply chain, VA, IAM/UAR (Entra/Google/Okta), incidents, training, internal audit, air-gap tarball.

Payment milestones tied to gates (see §6.1): 20% on G-Foundation (W2), 20% on G-AuthZ + G-Compliance-Core (W6), 20% on G-Module-Wave (W10), 20% on G-UAT + RC1 (W12), 20% on Extended sign-off (W14, only if Extended is contracted).

Risk shape. SecComply gets a known total cost. We carry the overrun risk: if W11 security remediation or load-test tuning eats more than the budgeted float, we absorb the cost. Scope changes are change-orders against the PO.

When this fits. Sponsor procurement requires a committed PO number, finance prefers cost certainty, sponsor doesn't want to track weekly invoices.

▸ 9.3 Option B — Monthly retainer

Predictable cost, mutual flexibility within the cap.

CADENCE	VOLUME	NOTES
Month 1 (W0–W4)	2 seniors × 4 wks = **8 pw**	Platform floor + Auth + AuthZ + Compliance Core.
Month 2 (W5–W8)	2 seniors × 4 wks + 1 contractor × 2 wks = **10 pw**	Auth completion, Vendor full, Policy start. Contractors join W7.
Month 3 (W9–W12)	2 seniors × 4 wks + 2 contractors × 4 wks = **16 pw**	Policy + Integrations + Dashboard + harden + RC1. Extended modules in parallel via contractors.
Month 4 (W13–W14, half)	2 seniors × 2 wks + 2 contractors × 2 wks = **8 pw**	Extended completes (Internal Audit, Air-gap, AI enrichment), v1.0-Extended ships.

Each month is invoiced and paid in arrears (Net 14 from month-end). Scope is renegotiated at each month boundary — SecComply can flex modules in/out, or pause Extended after Month 2 if Core is enough.

Risk shape. Both sides share. SecComply pays for actual work done each month and can stop at any month boundary if the project is no longer needed.

When this fits. Sponsor wants flexibility on Extended scope (some modules may be deferred), procurement is comfortable with monthly recurring spend, both sides expect mid-stream scope conversations.

▸ 9.4 Option C — Bi-weekly deliverable + bi-weekly payment (recommended for trust-build)

Maximum de-risk for the sponsor. Every two weeks, a named shippable lands. Sponsor pays in arrears for the *previous* fortnight's accepted delivery. Sponsor can cancel at any fortnight boundary; their maximum exposure is two weeks of team cost.

Why bi-weekly, not weekly: with a 2-person core team, a one-week cycle is too tight to land meaningful security-reviewed work *and* prepare acceptance evidence *and* invoice cleanly. Two-week cycles match the natural rhythm of paired senior development against security-sensitive code.

Per-fortnight shippable (Friday delivery at the end of each two-week block):

FORTNIGHT	WHAT SHIPS	ACCEPTANCE CRITERION
W1–W2	Platform floor: tenant-scoped BFF, audit-chain writer, seed loader, CI signing, AuthZ catalog	G-Foundation: tenant-scoped route live, audit log writes, CI green.
W3–W4	Auth pillars 1–10 (OIDC/SAML/magic-link/session/step-up), AuthZ 7-step pipeline + SoD, Compliance Readiness skeleton	G-AuthZ: pipeline passes fuzzer, OIDC end-to-end.
W5–W6	Auth pillars 11–19 (SCIM, break-glass, key lifecycle, audit catalog), Evidence engine, Risk module, framework mappings signed	G-Compliance-Core: auto-evidence emits for seeded controls.
W7–W8	Vendor full (intake, questionnaire, scoring, library), contractor onboarding + first connector skeleton (CSPM AWS)	E2E: create vendor, attach evidence, AWS scan-run produces findings.
W9–W10	Policy full (editor, diff, approval, ack campaigns), Integrations CRUD, Dashboard, CSPM GCP+Azure, SCM GitHub+GitLab	G-Module-Wave: 4 Core modules merged, E2E smoke green.
W11–W12	WCAG AA, load test, external security review, pen-test remediation, RC1, UAT, Supply Chain, VA	G-UAT: v1.0-Core RC1 signed and shipped.
W13–W14	IAM full (Entra, Google, Okta, OIDC), Incidents, Training, Internal Audit, Air-gap tarball, AI enrichment, Extended UAT	v1.0-Extended RC2 signed and shipped.

Invoicing. Every other Friday: we submit the fortnight's deliverable + acceptance evidence. Sponsor reviews over the weekend and pays Monday at the agreed bi-weekly rate. If a fortnight's deliverable is rejected, we rework on the next fortnight's clock; the rejected fortnight is not paid until accepted.

Risk shape. Sponsor carries two weeks of exposure at any time. We carry the team-availability risk: the team is dedicated and cannot be redeployed mid-fortnight if the sponsor cancels, so we bear the cost of the fortnight already underway when cancellation lands.

When this fits. First engagement between SecComply and us, sponsor wants to verify delivery quality before committing to a long PO, sponsor's procurement allows fortnightly invoicing.

▸ 9.5 Why this team size, this calendar, this price shape

The honest answer: the team size is **what we have**, and the calendar is **what that team can deliver**. We're not a 13-FTE consultancy pricing a 13-FTE engagement; we're two senior engineers (plus contracted hands for the Extended phase) pricing what two senior engineers can ship.

The trade we make: you get senior eyes on every line in the security-sensitive zone — crypto, session, authz, tenancy, audit-chain — across the whole programme. No handoff cliffs, no junior engineer learning your domain on your time. The cost: 12–14 weeks instead of 8, and no headcount fall-back if a senior is unavailable for an extended period.

9.6 Our recommendation

Lead with **Option C (bi-weekly deliverable + bi-weekly payment)** for a first engagement. It maximises trust-build, makes the "show me you can ship" criterion explicit, and limits sponsor exposure to two weeks at any time. Fall back to **Option A (fixed bundle)** if sponsor procurement rigidly requires a committed PO number. **Option B (monthly retainer)** is the middle ground if the sponsor wants flexibility on Extended scope but doesn't want fortnightly invoicing overhead.

Apply your senior engineering rate to senior person-weeks and your contractor rate to contractor person-weeks. Add the GRC SME at SecComply's internal rate (or our GRC consultant rate if SecComply prefers). External security review is a fixed pass-through.

---

10. WHAT WE NEED FROM CLIENT TO START W0

1. **Answers to G-1 through G-10** (§3.1).
2. **Signed-off stack** (§4.1). Default is Go/TS + Keycloak + PSMDB + Valkey + MinIO + shared-postgres (Keycloak+GlitchTip) + TipTap + Gotenberg + ClamAV + VictoriaMetrics stack, per research. Java alternative available (§4.2).
3. **Named GRC SME** for seed-data authorship (§5.4) — 6 SME-weeks.
4. **Access** to at least one sandbox tenant per integration target (AWS, GCP, Azure, GitHub, GitLab, Entra, Google Workspace, Okta) by W7 (when contractors join for the Extended phase).
5. **Design system source** or confirmation we bootstrap shadcn/ui + Tailwind v4 (G-10).
6. **Reference deployment target**: single-node Docker Compose (SigNoz observability alt) or multi-node k8s (VictoriaMetrics stack) — for RC1 demo.
7. **Licence-posture acknowledgement** for aggregate-distribution of MinIO (AGPL-3), ClamAV (GPL-2), and Percona Server MongoDB (SSPL + Percona grant) — one-page legal memo in release (full text in `LEGAL_POSTURE.md`).
8. **Scope commitment**: v1.0-Core only (12 weeks, audit-ready) vs Core+Extended (12–14 weeks, overlapping). The original "all 17 modules in 8 weeks" ask is **not deliverable at our team size and we won't commit to it** — see §1 and §6 for the honest math.

---

11. OUR HONEST RECOMMENDATION

- **Commit to v1.0-Core in 12 weeks.** This is deliverable at 2-senior-engineer pace, securable, and demos the platform's identity + compliance engine + vendor + policy spine — the parts that prove the architecture works. The original 8-week ask cannot be met honestly at this team size.
- **Treat v1.0-Extended as weeks 7–14 overlapping**, with 1–2 contractors joining at W7 once the connector framework contract is frozen at W4. Total programme: 12–14 weeks.
- **Do not compress Auth/AuthZ.** Everything else depends on it being right. Every week shaved here costs four weeks later.

- **Adopt, don't build, where the protocol/corpus/library surface is broad.** Keycloak for auth protocols, TipTap for rich-text, ClamAV for malware signatures, MinIO for S3, official SDKs for cloud APIs — 14 picks documented in our internal stack matrix with runner-ups and swap triggers.
- **Build, don't adopt, where the rules are ours and the interface is narrow.** `pkg/authz`, `KeyStore`, connector framework, mailer — ~28.75 eng-weeks of in-house code that insulates us from OSS churn via frozen interface boundaries.
- **Two seniors own the security-sensitive zone end-to-end.** Crypto, session, tenancy, authz, audit-chain — every commit reviewed by the other senior. Contractors only touch the contractor-safe zone (per-provider shims, CRUD modules) where the framework contracts are already frozen.
- **Agents accelerate, but do not substitute.** We will use them heavily for CRUD, BFF wiring, seed loaders, test scaffolds, docs, per-provider connector shims. We will **not** use them autonomously for crypto, session lifecycle, tenant isolation, authz pipeline, or audit-chain writer. Those stay human-owned, agent-paired.
- **All numbers traceable.** The eng-week counts, deployment topology, and licence posture in this document are sourced from internal research that we can walk through on request.

Done. Documented. Defended.

— Ketan Khairnar

APPENDIX A: TRACEABILITY QUICK-MAP

SRS SECTION	RESPONSE SECTION
§3.1 / §3.1B (Items 1)	§5.2 detailed estimate; §6.2 W1–W4
§3.6 (Item 6)	§5.2; §6.2 W4–W5
§3.7 (Item 7)	§5.2; §6.2 W5–W6
§3.2–§3.5 (compliance engine)	§5.3; §6.1 in-scope for Core; §6.2 W2–W3
§3.8–§3.15 (scanners + ops)	§5.3; §6.1 deferred to Extended; §6.3 W7–W14 (overlapping)
§3.16–§3.17	§5.3; §6.1 in-scope for Core thin; §6.2 W6
§4 external interfaces	§4 stack; §5.1 platform floor
§5 NFRs	§5.5 cross-cutting; W7 load + security
§6 retention/legal/localisation/deployment	§5.1 platform floor + W7 operator docs
Appendix C (12 TBDs)	§3.2 inherited; each mapped to platform or cross-cutting item