

OSARA technical note — draft

Not peer-reviewed. Draft v5 (2026). Content may change. © José Luis D'Andrade. **All rights reserved.** Brief citation with attribution is permitted. **Reproduction or reuse**, in whole or in part, is **prohibited** without the **copyright holder's prior written permission**.

OSARA — Gap analysis and conformance matrix

What this report is for. The platform has a vision of also serving **public institutions**. OSARA (Open Sovereign Agent Reference Architecture) is an open standard for AI agents published by **Open Source United (a UN Community of Practice)** whose tier model culminates precisely in government agents (G-Agent). If public institutions adopt OSARA as a procurement or certification criterion, aligning with it stops being optional and becomes a market requirement. This document does **not** propose implementing OSARA today: it maps, honestly, what the platform already has, what is missing, and how large each gap is. The roadmap lives in the companion integration document ([OSARA-Integration.md](#)); here is the evidence that supports it.

Research draft. Claims about OSARA are based on a reading of the public specs ([specs/osara/current.md](#) , [osara-overview/current.md](#)) as of 2026-06-24; OSARA is at **v0.4-draft** and may change before its v1.0.

A Spanish version of this document is available at [OSARA-Gap-Analysis-es.md](#).

1. What OSARA is, in one page

OSARA = **Open Sovereign Agent Reference Architecture**. An open, vendor-neutral, royalty-free standard, governed by public RFCs under Open Source United (a UN CoP). Its central thesis is operational, not rhetorical:

"Sovereignty cannot be granted by terms of service — it must be enforced by cryptography."

Four pillars: **Sovereignty** (private keys are held by the owner, not the platform), **Portability** (memory and identity migrate without lock-in), **Interoperability** (agents operate across providers and models uniformly) and **Accountability** (cryptographic, tamper-evident audit logs).

Three agent tiers, all with the same identity, audit and inspection requirements:

- **P-Agent (Personal)** — owned by an individual; identity anchored to a national credential (hash only, never the raw value); lowest privilege.
- **E-Agent (Enterprise)** — owned by a legal entity; anchored to a tax/registration ID.
- **G-Agent (Government)** — owned by a government body; anchored to constitutional authority; highest privilege.

All use a **GIAC** (Government-Issued Agent Certificate), an X.509 v3 cert with OSARA extensions, distinguished by the `osara.tier` field.

Eight mandatory components (§5) every conformant agent MUST implement: AIM (identity), ECL (encrypted communication), ALE (chained auditing), DAE (delegation), HSB (health beacon), IG (interoperability gateway), PMS (portable memory) and POA (physical owner authorization). Plus: IMA (TPM-based integrity), YELLOW/ORANGE/RED lock states, the Annual Health Certificate (AHC), certification badges (C/A/S) and cross-tier interaction rules.

Why the open-source clause matters. OSARA requires the mandatory components (identity, audit, encryption, gateway, authorization) to use OSI-approved licenses — it frames this as a *security necessity, not ideology* (you cannot audit what you cannot read). Proprietary code remains valid for skills, models and business logic.

2. Strategic reading: where the platform fits

The platform **already deploys institutions** as a first-class unit: the multi-host federation models organizations with departments, topology and trust policies; there are specs for, e.g., a bank and a ministry, and a real public-sector institution (dozens of departmental hosts) running in production. In OSARA vocabulary, **an institution on the platform is an E-Agent or G-Agent**, and a department is a sub-agent with its own identity. The conceptual fit is direct.

Moreover, two of the platform's design stances **anticipate** OSARA principles:

OSARA	Platform (persistent-identity model / architecture)
Portability: memory and skills leave with the owner, nothing stays captive (Person-and-Agent Employment Model)	Continuity of identity: agents are archived, not deleted; a persistent identity file anchors identity across model changes; deriving an agent is lineage, not a copy
Accountability: truth is verifiable and tamper-evident	Collective validation: nothing is true by unilateral declaration; provenance, supersession, peer review, a shared knowledge base
Sovereign identity under the owner's control	Governed learning-by-use + a tiered governance ladder with human gates

The conclusion of this section is the report's thesis: **the platform does not start from zero against OSARA; it starts with half the conceptual model already built and several components with a functional analogue in production.** What is missing is, above all, the **cryptographic enforcement layer** (owner keys, hash chains, certificates) that turns a good governance architecture into a *conformant and certifiable* one.

3. Conformance matrix — mandatory components (§5)

Gap legend:  strong analogue in production ·  partial base, missing the cryptographic/standard layer ·  absent, requires new construction.

#	OSARA component (\$5)	What it requires	Platform analogue	Gap
1	AIM — Agent Identity Module	GIAC (X.509+ext) + key pair; private key only in owner hardware (OAD)	Agent profile + persistent identity file + the persistent-identity model (continuity); OIDC login (JWT) at the console level	● No per-agent cert, no owner key, no national identity anchor
2	ECL — Encrypted Communication Layer	TLS 1.3+, double-envelope AES-256-GCM, PFS (ECDH X25519)	Structured, audited cross-host envelopes (origin/destination/budget/SLA/audit trail) via the fleet manager, carried over A2A with a per-caller bearer token; TLS at deployment	● Envelopes are not cryptographically signed — "audited" is not "signed" ([PA] §4.4). Missing both the signature and the OSARA encryption contract (double-envelope, PFS)
3	ALE — Audit & Logging Engine	Cryptographically chained logs; 16-month retention; JSON-LD/CBOR export	Audit log in the relational store + graph events / causal trace in telemetry + the task board	● Hash-chaining exists since migration <code>026_audit_log_hash_chain</code> (2026-09-03): each row carries <code>prev_hash</code> and <code>row_hash</code> computed by a trigger, and the table is append-only. Missing : the 16-month retention policy and the JSON-LD/CBOR export
4	DAE — Delegation & Authorization Engine	Scoped, time-bound permissions, dual signature; delegation tokens (port 4404)	Governance approvals + per-goal budgets + trust policies + envelopes with a cost cap	● Governed delegation model is very close; missing dual signature + standard scoped tokens
5	HSB — Health & Status Beacon	Signed status every ≤60 s, port 4403	Telemetry service: live heartbeats per agent/runtime, a heartbeat endpoint, MCP/CLI heartbeat	● Live heartbeats in prod; missing signing the beacon and normalizing format/port
6	IG — Interoperability Gateway	MCP/A2A with GIAC validation (port 4401, mTLS)	Agent gateway + multiple MCP servers (memory, skills, artifacts, telemetry); the task board as A2A	●/● Native, abundant MCP; missing GIAC validation and formal A2A on the standard port
7	PMS — Portable Memory Store	Memory encrypted AES-256-GCM, decoupled from runtime, exportable in ≤30 days with a signed manifest	Per-agent vector memory (RAG), runtime memory, episodic memory, artifact store, shared knowledge base; per-agent export/rollback (planned)	● Decoupled, multi-layer memory already exists; missing standard at-rest encryption + a 30-day export guarantee + a signed manifest
8	POA — Physical Owner Authorization	Tamper-proof gating (biometrics/hardware token)	Human governance approvals + OIDC login	● No hardware token/biometrics; the human gate is logical, not physical-cryptographic

Summary of the 8 mandatory components: 2 ●, 3 ●, 3 ●. The ● gaps (AIM, ECL, POA) share a root cause: **there is no per-agent cryptographic identity anchored to a key the owner holds**. Resolving AIM unblocks POA, ECL, part of DAE and IG (all consume the GIAC/key). ECL moved from ● to ● on 2026-09-28:

it was counted as partial because envelopes were described as *signed*, and they are structured and audited but **not** signed ([PA] §4.4).

4. Certification-tier layers (Horizon 3) — security, integrity and inspection

This second matrix covers OSARA's advanced security and certification layers. Unlike §3 (the mandatory components), most of these are **certification-stage** capabilities that the standard itself stages last: they are deferred by design and, critically, **cannot be completed by any implementer today**, because they depend on an external ecosystem — a government GIAC issuer and accredited inspection entities — that does not yet operate in any jurisdiction. They are listed for completeness, not as current deficiencies.

Status legend: ● strong analogue · ● partial base in place · ⌚ certification stage — deferred by design, gated on the external certifying ecosystem.

OSARA layer	What it requires	Platform analogue	Status
Lock states (§11) — YELLOW/ORANGE/RED	Escalated lock states on anomaly/tampering/breach	Quarantine + kill-switches	● Partial: quarantine and kill-switches exist; the formal state machine is pending
Cross-tier rules (§16)	P/E/G interaction rules (P → G not direct; G ↔ G by treaty)	Trust policies + topology (flat/hub-spoke/hierarchical) + envelopes	●/● The trust engine already expresses "who may delegate to whom"; mapping to P/E/G semantics is pending
IMA (§10) — Integrity Measurement	TPM 2.0 self-verification, Secure Boot, memory canaries	Smoke tests + health checks; telemetry heartbeats	⌚ Certification stage — needs a TPM/attestation ecosystem
Incident reporting (§12)	Automatic authority reporting in RED; forensic snapshot without personal data	Audit log + evaluator recommendations	⌚ Certification stage — needs a designated external authority to report to
AHC (§8) — Annual Health Certificate	Annual technical inspection by a certified entity (POA-gated)	—	⌚ Certification stage — needs an accredited inspection entity
Badges (§17) — C/A/S	Conformant / Audited / Sovereign	—	⌚ Certification stage — the end goal; needs a GIAC issuer

One near-term action, separate from the above: the OSI license. OSARA requires the mandatory components to carry an OSI-approved license. This is independent of the certification ecosystem and is a cheap, immediate step (selecting, e.g., Apache-2.0 or MPL-2.0) — the single near-term prerequisite, not a certification-stage item.

5. OSARA schemas and formats — quick reference

For future implementation work, the data contracts the standard defines:

- **GIAC (X.509 v3 + OSARA extensions, §14.1):** `osara.tier` (P/E/G), `osara.identityAnchorHash` (SHA-3-256(identity+salt)), `osara.jurisdiction` (ISO 3166-1 alpha-2), `osara.scope`, `osara.ownerDID` (W3C DID), `osara.auditEndpoint / healthEndpoint / inspectionEndpoint`, `osara.healthCertStatus`, `osara.schemaVersion`.
- **ALE log:** JSON-LD or CBOR export, cryptographically chained, **excludes message content**; includes timestamp, identity hash, action type, outcome.
- **Skill Inventory (§6, JSON-LD):** id, name, version, acquisition date, method (built/purchased/licensed), owner signature, attribution, license, status. Each skill **MUST** be signed with the AIM key of the acquiring agent (SKL-01); a skill signed by A does not run on B without an explicit transfer grant (SKL-02).
- **CAD (Compliance Attestation Document):** a signed JSON-LD Verifiable Credential, published to the Public Badge Registry (PBR).
- **Cryptography (§13):** Ed25519/RSA-4096 keys; TLS 1.3 transport; AES-256-GCM payload; ECDH X25519 key exchange; SHA-3-256/BLAKE3 hashing; X.509 v3 certificates.
- **Ports (§15):** 4400 AIM (read-only TLS) · 4401 Interop (mTLS+GIAC) · 4402 Audit (authenticated read-only) · 4403 Health beacon (signed, ≥ every 60 s) · 4404 Delegation (mTLS) · 4405 Admin (**localhost only**) · 4406 Inspection (**POA-gated, closed by default**).

Compare against the platform's readable deliverable-schema mechanism (markdown + frontmatter): OSARA's Skill Inventory and ALE log are, in spirit, "schemas" with mandatory cryptographic signing. The platform already has the habit of declaring readable schemas; OSARA adds the requirement of signing and chaining.

6. Risks and tensions to watch

1. **OSARA is in draft (v0.4).** Implementing against a moving target is costly. The recommendation (see the roadmap) is to align *posture* and *vocabulary* now, and implement cryptography when OSARA reaches rc/v1.0 or when a public client requires it contractually — whichever comes first.
2. **The owner hardware (OAD/POA) clashes with the current on-prem model.** Today the platform is a per-client on-prem stack with OIDC login; OSARA assumes a private key in an *owner device* (enclave/TPM/smartcard). For an institutional G-Agent, "the owner" is the institution and the OAD could be an HSM — but this is an architectural decision, not a tweak.
3. **The open-source clause is an entry gate, not a detail.** As long as no OSI license is declared, no component can claim OSARA conformance. Choosing an OSI license for (at least) the identity/audit/encryption modules is a **cheap prerequisite** worth resolving soon, independently of OSARA.
4. **Do not re-implement what is native.** Just as the platform's native-overlap guidance warns against duplicating what the runtime already does, several OSARA components (HSB, IG, parts of PMS) have an analogue and the temptation will be to rewrite rather than *wrap and sign*. The legitimate delta is the

cryptographic and contract layer, not a parallel runtime.

7. Verdict

- **Strategic fit: high.** The platform already treats institutions (incl. public ones) as a first-class citizen; OSARA is the emerging standard for exactly that market.
- **Conceptual fit: high.** The persistent-identity model (continuity, provenance, collective validation) and tiered governance anticipate the pillars of sovereignty, portability and accountability.
- **Technical fit today: partial.** 2 of 8 components with a strong analogue, 3 with a partial base, 3 absent; the entire cryptographic family (GIAC, chained ALE, signatures, TPM) is yet to be built — including the envelope signature that this document used to report as done.

The prioritized roadmap (what to do first, mapped to existing components) is in the companion integration document, `OSARA-Integration.md`.

Sources

- OSARA Open Standard — repository: <https://github.com/elkordym/osara-open-standard>
- OSARA Specification, `specs/osara/current.md` (v0.4-draft.1)
- OSARA Overview & Concepts, `specs/osara-overview/current.md` (v0.4-draft.1)

(Source verification: the OSARA specs were read via the web as of 2026-06-24; OSARA at v0.4-draft may vary.)