

Documento técnico OSARA — borrador

No revisado por pares. Borrador v5 (2026). El contenido puede cambiar. © José Luis D'Andrade. **Todos los derechos reservados.** Cita breve con atribución permitida. **Queda prohibida** la reproducción o reutilización, total o parcial, sin **autorización previa por escrito** del titular.

OSARA — Análisis de brechas y matriz de conformidad

Para qué sirve este reporte. La plataforma tiene la visión de servir también a **instituciones públicas**. OSARA (Open Sovereign Agent Reference Architecture) es un estándar abierto de agentes de IA publicado por **Open Source United (una Community of Practice de la ONU)** cuyo modelo de tiers culmina precisamente en agentes de gobierno (G-Agent). Si las instituciones públicas adoptan OSARA como criterio de compra o de certificación, alinearse con él deja de ser opcional y pasa a ser requisito de mercado. Este documento **no** propone implementar OSARA hoy: mapea, honestamente, qué tiene ya la plataforma, qué falta, y de qué tamaño es cada brecha. La hoja de ruta vive en el documento de integración que lo acompaña (`OSARA-Integration-es.md`); aquí está la evidencia que la sostiene.

Borrador de investigación. Las afirmaciones sobre OSARA se basan en lectura de los specs públicos (`specs/osara/current.md` , `osara-overview/current.md`) a fecha 2026-06-24; OSARA está en **v0.4-draft** y puede cambiar antes de su v1.0.

La versión por defecto (inglés) de este documento está en `OSARA-Gap-Analysis.md`.

1. Qué es OSARA, en una página

OSARA = **Open Sovereign Agent Reference Architecture**. Estándar abierto, vendor-neutral, royalty-free, gobernado por RFCs públicos bajo Open Source United (CoP de la ONU). Su tesis central es operativa, no retórica:

"La soberanía no puede concederse por términos de servicio — debe imponerla la criptografía."

Cuatro pilares: **Soberanía** (las llaves privadas las tiene el dueño, no la plataforma), **Portabilidad** (memoria e identidad migran sin lock-in), **Interoperabilidad** (los agentes operan entre proveedores y modelos de forma uniforme) y **Rendición de cuentas** (logs de auditoría criptográficos, a prueba de manipulación).

Tres tiers de agente, todos con los mismos requisitos de identidad, auditoría e inspección:

- **P-Agent (Personal)** — propiedad de un individuo; identidad anclada a credencial nacional (solo el hash, nunca el dato crudo); menor privilegio.
- **E-Agent (Enterprise)** — propiedad de una entidad legal; anclado a ID fiscal/registral.

- **G-Agent (Government)** — propiedad de un cuerpo de gobierno; ancla de autoridad constitucional; mayor privilegio.

Todos usan un **GIAC** (Government-Issued Agent Certificate), un X.509 v3 con extensiones OSARA, distinguidos por el campo `osara.tier`.

Ocho componentes obligatorios (§5) que todo agente conforme DEBE implementar: AIM (identidad), ECL (comunicación cifrada), ALE (auditoría encadenada), DAE (delegación), HSB (beacon de salud), IG (gateway de interoperabilidad), PMS (memoria portable) y POA (autorización física del dueño). Más: IMA (integridad por TPM), estados de bloqueo YELLOW/ORANGE/RED, certificado anual de salud (AHC), badges de certificación (C/A/S) y reglas de interacción cross-tier.

Por qué importa la cláusula open-source. OSARA exige que los componentes obligatorios (identidad, auditoría, cifrado, gateway, autorización) usen licencias aprobadas por OSI — lo declara como *necesidad de seguridad, no ideología* (no puedes auditar lo que no puedes leer). El código propietario sigue siendo válido para skills, modelos y lógica de negocio.

2. Lectura estratégica: dónde encaja la plataforma

La plataforma **ya despliega instituciones** como unidad de primera clase: la federación multi-host modela organizaciones con departamentos, topología y políticas de confianza (*trust policies*); hay specs para, por ejemplo, un banco y un ministerio, y una institución del sector público real (decenas de hosts departamentales) corriendo en producción. En vocabulario OSARA, **una institución en la plataforma es un E-Agent o G-Agent**, y un departamento es un sub-agente con identidad propia. El encaje conceptual es directo.

Más aún, dos posiciones de diseño de la plataforma **anticipan** principios OSARA:

OSARA	Plataforma (modelo de identidad persistente / arquitectura)
Portabilidad: la memoria y skills salen con el dueño, nada queda cautivo (Person-and-Agent Employment Model)	Continuidad de la identidad: los agentes se archivan, no se borran; un archivo de identidad persistente ancla la identidad a través de cambios de modelo; derivar un agente es linaje, no copia
Rendición de cuentas: la verdad es verificable y a prueba de manipulación	Validación colectiva: nada es verdad por declaración unilateral; <i>provenance</i> , supersesión, <i>peer review</i> , una base de conocimiento compartida
Identidad soberana bajo control del dueño	Aprendizaje por uso gobernado + una escalera de gobernanza por niveles con <i>compuertas humanas</i>

La conclusión de esta sección es la tesis del reporte: **la plataforma no parte de cero frente a OSARA; parte con la mitad del modelo conceptual ya construido y varios componentes con un análogo funcional en producción.** Lo que falta es, sobre todo, la **capa criptográfica de imposición** (llaves del dueño, cadenas de hash, certificados) que convierte una buena arquitectura de gobernanza en una *conforme y certificable*.

3. Matriz de conformidad — componentes obligatorios (§5)

Leyenda de brecha: ● análogo fuerte en producción · ● base parcial, falta capa criptográfica/estándar · ● ausente, requiere construcción nueva.

#	Componente OSARA (§5)	Qué exige	Análogo en la plataforma	Brecha
1	AIM — Agent Identity Module	GIAC (X.509+ext) + par de llaves; clave privada solo en hardware del dueño (OAD)	Perfil de agente + archivo de identidad persistente + el modelo de identidad persistente (continuidad); login OIDC (JWT) a nivel consola	● No hay cert por-agente, ni llave del dueño, ni ancla de identidad nacional
2	ECL — Encrypted Communication Layer	TLS 1.3+, doble sobre AES-256-GCM, PFS (ECDH X25519)	Sobres cross-host estructurados y auditados (origen/destino/budget/SLA/audit trail) vía el fleet manager, transportados por A2A con token bearer por llamante; TLS en despliegue	● Los sobres no están firmados criptográficamente — «auditado» no es «firmado» ([PA] §4.4). Falta la firma y también el contrato de cifrado OSARA (doble sobre, PFS)
3	ALE — Audit & Logging Engine	Logs encadenados criptográficamente; retención 16 meses; export JSON-LD/CBOR	Log de auditoría en la base relacional + eventos de grafo / traza causal en telemetría + el tablero de tareas	● El encadenado por hash EXISTE desde la migración <code>026_audit_log_hash_chain</code> (2026-09-03): cada fila lleva <code>prev_hash</code> y <code>row_hash</code> calculados por un trigger, y la tabla es append-only. Falta: la política de retención de 16 meses y la exportación JSON-LD/CBOR
4	DAE — Delegation & Authorization Engine	Permisos scoped, time-bound, doble firma; tokens de delegación (puerto 4404)	Aprobaciones de gobernanza + presupuestos por objetivo + trust policies + sobres con tope de costo	● Modelo de delegación gobernada muy cercano; falta doble firma + tokens scoped estándar
5	HSB — Health & Status Beacon	Estado firmado cada ≤60 s, puerto 4403	Servicio de telemetría: <i>heartbeats</i> vivos por agente/runtime, endpoint de heartbeat, MCP/CLI	● Heartbeats vivos en prod; falta firmar el beacon y normalizar formato/puerto
6	IG — Interoperability Gateway	MCP/A2A con validación GIAC (puerto 4401, mTLS)	Gateway de agentes + varios servidores MCP (memoria, skills, artefactos, telemetría); el tablero de tareas como A2A	●/● MCP nativo y abundante; falta validación GIAC y A2A formal en puerto estándar

#	Componente OSARA (§5)	Qué exige	Análogo en la plataforma	Brecha
7	PMS — Portable Memory Store	Memoria cifrada AES-256-GCM, desacoplada del runtime, exportable en ≤30 días con manifiesto firmado	Memoria vectorial por-agente (RAG), memoria del runtime, memoria episódica, almacén de artefactos, base de conocimiento compartida; export/rollback por-agente (planeado)	● Memoria desacoplada y multicapa ya existe; falta cifrado-at-rest estándar + garantía de export 30 días + manifiesto firmado
8	POA — Physical Owner Authorization	Gating a prueba de manipulación (biometría/token hardware)	Aprobaciones humanas de gobernanza + login OIDC	● No hay token hardware/biometría; la compuerta humana es lógico, no físico-criptográfico

Resumen de los 8 obligatorios: 2 ●, 3 ●, 3 ●. Las brechas ● (AIM, ECL, POA) comparten raíz: **no existe una identidad criptográfica por-agente anclada a una llave que tenga el dueño**. Resolver AIM desbloquea POA, ECL, parte de DAE e IG (todos consumen el GIAC/llave). ECL pasó de ● a ● el 2026-09-28: contaba como parcial porque los sobres se describían como *firmados*, y están estructurados y auditados pero **no** firmados ([PA] §4.4).

4. Capas de certificación (Horizonte 3) — seguridad, integridad e inspección

Esta segunda matriz cubre las capas avanzadas de seguridad y certificación de OSARA. A diferencia del §3 (los componentes obligatorios), la mayoría de estas son capacidades de **etapa de certificación** que el propio estándar coloca al final: están diferidas por diseño y, sobre todo, **no las puede completar hoy ningún implementador**, porque dependen de un ecosistema externo — un emisor de certificados GIAC de gobierno y entidades inspectoras acreditadas — que aún no opera en ninguna jurisdicción. Se listan por completitud, no como deficiencias actuales.

Leyenda de estado: ● análogo fuerte · ● base parcial presente · ⌚ etapa de certificación — diferida por diseño, gated por el ecosistema certificador externo.

Capa OSARA	Qué exige	Análogo en la plataforma	Estado
Lock states (§11) — YELLOW/ORANGE/RED	Estados de bloqueo escalados ante anomalía/manipulación/brecha	Cuarentena + kill-switches	● Parcial: cuarentena y kill-switches existen; falta la máquina de estados formal
Cross-tier rules (§16)	Reglas de interacción P/E/G (P → G no directo; G ↔ G por tratado)	Trust policies + topología (flat/hub-spoke/hierarchical) + sobres	●/● El motor de trust ya expresa "quién puede delegar a quién"; falta mapear a la semántica P/E/G

Capa OSARA	Qué exige	Análogo en la plataforma	Estado
IMA (§10) — Integrity Measurement	Auto-verificación por TPM 2.0, Secure Boot, memory canaries	Smoke tests + health checks; heartbeats de telemetría	⌚ Etapa de certificación — requiere un ecosistema de TPM/atestación
Incident reporting (§12)	Reporte automático a autoridad en RED; snapshot forense sin datos personales	Log de auditoría + recomendaciones del evaluador	⌚ Etapa de certificación — requiere una autoridad externa designada a la cual reportar
AHC (§8) — Annual Health Certificate	Inspección técnica anual por entidad certificada (POA-gated)	—	⌚ Etapa de certificación — requiere una entidad inspectora acreditada
Badges (§17) — C/A/S	Conformant / Audited / Sovereign	—	⌚ Etapa de certificación — la meta final; requiere un emisor GIAC

Una acción de corto plazo, separada de lo anterior: la licencia OSI. OSARA exige que los componentes obligatorios lleven una licencia aprobada por OSI. Esto es independiente del ecosistema de certificación y es un paso barato e inmediato (elegir, p. ej., Apache-2.0 o MPL-2.0) — el único prerrequisito de corto plazo, no un ítem de etapa de certificación.

5. Esquemas y formatos OSARA — referencia rápida

Para futuras tareas de implementación, los contratos de datos que el estándar define:

- **GIAC (X.509 v3 + extensiones OSARA, §14.1):** `osara.tier` (P/E/G), `osara.identityAnchorHash` (SHA-3-256(identidad+salt)), `osara.jurisdiction` (ISO 3166-1 alpha-2), `osara.scope`, `osara.ownerDID` (W3C DID), `osara.auditEndpoint` / `healthEndpoint` / `inspectionEndpoint`, `osara.healthCertStatus`, `osara.schemaVersion`.
- **ALE log:** export JSON-LD o CBOR, encadenado criptográficamente, **excluye contenido de mensajes**; incluye timestamp, hash de identidad, tipo de acción, resultado.
- **Skill Inventory (§6, JSON-LD):** id, nombre, versión, fecha de adquisición, método (built/purchased/licensed), firma del dueño, atribución, licencia, estado. Cada skill **DEBE** firmarse con la llave AIM del agente que la adquiere (SKL-01); una skill firmada por A no se ejecuta en B sin grant explícito de transferencia (SKL-02).
- **CAD (Compliance Attestation Document):** Verifiable Credential JSON-LD firmada, publicada al Public Badge Registry (PBR).
- **Criptografía (§13):** llaves Ed25519/RSA-4096; transporte TLS 1.3; payload AES-256-GCM; key exchange ECDH X25519; hashing SHA-3-256/BLAKE3; certificados X.509 v3.
- **Puertos (§15):** 4400 AIM (read-only TLS) · 4401 Interop (mTLS+GIAC) · 4402 Audit (read-only autenticado) · 4403 Health beacon (firmado, ≥cada 60 s) · 4404 Delegación (mTLS) · 4405 Admin (**solo localhost**) · 4406 Inspección (**POA-gated, cerrado por defecto**).

Comparar contra el mecanismo de la plataforma para declarar esquemas de entregables legibles (markdown + frontmatter): el Skill Inventory y el ALE log de OSARA son, en espíritu, "esquemas" con firma criptográfica obligatoria. La plataforma ya tiene el hábito de declarar esquemas legibles; OSARA añade la exigencia de firma y encadenado.

6. Riesgos y tensiones a vigilar

1. **OSARA está en draft (v0.4).** Implementar contra un blanco móvil es caro. La recomendación (ver hoja de ruta) es alinear *postura* y *vocabulario* ahora, e implementar criptografía cuando OSARA llegue a rc/v1.0 o cuando un cliente público lo exija contractualmente — lo que ocurra primero.
2. **El hardware del dueño (OAD/POA) choca con el modelo on-prem actual.** Hoy la plataforma es un stack on-prem por cliente con login OIDC; OSARA asume una llave privada en un *dispositivo del dueño* (enclave/TPM/smartcard). Para un G-Agent institucional, "el dueño" es la institución y el OAD podría ser un HSM — pero esto es una decisión de arquitectura, no un retoque.
3. **La cláusula open-source es una compuerta de entrada, no un detalle.** Mientras no se declare una licencia OSI, ningún componente puede reclamar conformidad OSARA. Elegir una licencia OSI para (al menos) los módulos de identidad/auditoría/cifrado es **prerrequisito barato** que conviene resolver pronto, independientemente de OSARA.
4. **No re-implementar lo nativo.** Igual que la guía de solapamiento nativo de la plataforma advierte de no duplicar lo que el runtime ya hace, varios componentes OSARA (HSB, IG, partes de PMS) tienen análogo y la tentación será reescribir en vez de *envolver* y *firmar*. El delta legítimo es la capa criptográfica y de contrato, no un runtime paralelo.

7. Veredicto

- **Encaje estratégico: alto.** La plataforma ya trata instituciones (incl. públicas) como ciudadano de primera clase; OSARA es el estándar emergente para exactamente ese mercado.
- **Encaje conceptual: alto.** El modelo de identidad persistente (continuidad, provenance, validación colectiva) y la gobernanza por niveles anticipan los pilares de soberanía, portabilidad y rendición de cuentas.
- **Encaje técnico hoy: parcial.** 2 de 8 componentes con análogo fuerte, 3 con base parcial, 3 ausentes; toda la familia criptográfica (GIAC, ALE encadenado, firmas, TPM) está por construir — incluida la firma de sobres que este documento daba por hecha.

La hoja de ruta priorizada (qué hacer primero, mapeado a los componentes existentes) está en el documento de integración que lo acompaña, **OSARA-Integration-es.md**.

Fuentes

- OSARA Open Standard — repositorio: <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)

(Verificación de fuentes: los specs OSARA se leyeron vía web a 2026-06-24; OSARA en v0.4-draft puede variar.)