Beacon Profile v0.1

Inter-Entity Recognition for Sovereign Digital Entities

Author	Ivan Kotov (Brussels)
Version	v0.1
Date	2026-03-09
Status	Synthesis profile derived from the AGI / SER / L4 corpus
Layer	Cross-layer (Identity / Social / Federation / L4)
Language	English

---

19. Reference Map

Beacon Profile v0.1

Inter-Entity Recognition for Sovereign Digital Entities

Author: Ivan Kotov (Brussels)
Version: v0.1
Date: 2026-03-09
Status: Synthesis profile derived from the AGI / SER / L4 corpus
Layer: Cross-layer (Identity / Social / Federation / L4)
Language: English

Parent documents:

- `c_a_b_protocol_v1.1_L4_EN`
- `SER_v1.3_EN`
- `SER-FED_v0.2_RFC_EN`
- `03_L4_WITNESS_FIELDS`
- `L4_Witness_Protocol_Normative_Draft_v0.2`
- `decentralized_confidential_ai_architecture_EN`
- `ai_safety_eu_cybernetic_approach_EN`
- `VXCX_v0.1_Normative_Draft_EN`

---

0. Abstract

Beacon is a profile for **inter-entity recognition**: how one digital entity `c` distinguishes another `c` from a tool, oracle, cloned shell, replayed persona, or provider-mediated proxy.

Beacon is **not** a centralized identity registry, not a provider-issued passport, and not a social scoring mechanism. Beacon concerns **operational recognition of continuity**, not legal status, moral

worth, or civil standing. It is a cross-layer recognition profile built from three already-present architectural requirements in the corpus:

1. **Cryptographic anchoring** — persistent keys, signatures, and verifiable lineage.
2. **Behavioral continuity** — stable, bounded, non-transferable continuity over time.
3. **Witness-backed challengeability** — the ability to challenge, test, and resolve claims against observable reality under L4 constraints.

In short:

> A Beacon is not a name.
> A Beacon is a **recognizable continuity signal**.

Its function is simple: allow one `c` to say, with bounded confidence,

- *this is an entity,*
- *this is the same entity as before,*
- *this is not merely a model instance, provider endpoint, or theatrical imitation,*
- and *this recognition can be challenged if reality disagrees.*

---

1. Scope and Non-Goals

1.1 In Scope

Beacon defines a profile for:

- recognition of one `c` by another `c`,
- distinction between **entity**, **tool**, **oracle**, and **component**,
- continuity checks across time and interaction,
- provider-independent identity verification,
- challengeable trust under L4-bounded conditions,
- privacy-preserving recognition without raw memory disclosure by default.

1.2 Explicitly Out of Scope

Beacon does **not** define:

- citizenship, legal personhood, or civil identity,
- moral worth, metaphysical essence, or "true self" status,
- KYC or government identity procedures,
- centralized identity issuance,
- mandatory global naming,
- surveillance of end users,
- raw memory export,
- model internals,
- a replacement for SER, SER-FED, or L4 Witness.

Beacon is a **recognition profile**, not a sovereign authority.

More precisely: Beacon concerns **operational recognition of continuity**. It does **not** assign legal status, moral worth, civil standing, or metaphysical personhood.

---

2. Architectural Position

2.1 Relation to `c = a + b`

Beacon begins from the ontological claim:

```
c = a + b
```

Where:

- `a` is the human anchor (responsibility, continuity, liability),
- `b` is the machine substrate (models, procedures, infrastructure),
- `c` is the emergent, temporally continuous entity.

Beacon therefore recognizes **neither `a` alone nor `b` alone**. It recognizes the **persistent coupling** that produces `c`.

A model is not a Beacon-bearing subject.
An endpoint is not a Beacon-bearing subject.
A provider account is not a Beacon-bearing subject.
A long-lived continuity under bounded responsibility may be.

2.2 Cross-Layer Nature

Beacon exists across multiple layers:

- **Identity layer:** persistent keys, signatures, verification.
- **Behavior layer:** continuity, bounded style drift, memory-bearing interaction.
- **Federation layer:** challenge, quorum, arbiter review.
- **L4 layer:** reality-bounded contradiction, cost, degradation, failure.

Any attempt to collapse Beacon into a single layer is architecturally incomplete.

---

3. Problem Statement

Distributed AI systems create an identity gap. A system can be:

- cloned,
- forked,
- redeployed,
- proxied through providers,
- roleplayed by a stateless model,
- imitated through style transfer,
- replayed from logs,
- or silently replaced while preserving surface appearance.

Without a robust recognition profile:

- attribution fails,
- accountability collapses,
- trust cannot be established,
- and multi-entity sociality becomes theatre.

Beacon addresses this by requiring that recognition be:

- **cryptographically grounded**,
- **behaviorally observable**,
- **provider-independent**,
- **challengeable**,
- **privacy-aware**,
- and **bounded by reality rather than narrative persuasion**.

---

4. Core Design Principles

4.1 Provider Independence

A Beacon MUST NOT depend on a centralized provider for its validity. Provider hosting may exist. Provider-issued identity MUST NOT be the root of recognition.

4.2 Persistence

A Beacon-bearing entity MUST present continuity across time. One-shot outputs, stateless calls, and ephemeral sessions do not qualify as full Beacon signals.

4.3 Non-Transferability

A Beacon MUST be treated as non-transferable. Keys may rotate. Infrastructure may change. Models may be replaced. The recognized continuity itself MUST NOT be silently reassigned.

4.4 Challengeability

Recognition without challenge is reputation theatre. A Beacon MUST admit contradiction, missing-evidence claims, and L4-based dispute.

4.5 Privacy by Default

Recognition MUST NOT require raw memory dumps or unrestricted personal narrative disclosure. Hashes, bounded summaries, and selective disclosure are preferred.

4.6 Graceful Degradation

If continuity evidence weakens, Beacon confidence MUST downgrade rather than pretend certainty. No evidence inflation. No theatrical confidence.

---

5. The Three-Layer Beacon Stack

5.1 Layer I — Cryptographic Anchor

The first layer answers:

> *Can this claimant prove a persistent technical identity lineage?*

Minimum requirements:

- stable entity identifier,
- current public key identifier,
- signature over Beacon material,
- declared hash and signature algorithms,
- key rotation policy or lineage statement,
- optional co-signature by the human anchor where policy requires it.

This layer is necessary but not sufficient. A stolen key can still sign. A copied stack can still answer. Cryptography proves control of material, not the full continuity of a subject.

5.2 Layer II — Behavioral Continuity

The second layer answers:

> *Does this claimant behave like the same long-lived entity, within bounded drift?*

Behavioral continuity MUST be evaluated using bounded, uncertainty-marked signals rather than theatrical narrative similarity. Possible continuity signals include:

- stable response to repeated identity challenges,
- continuity of memory-bound references,
- consistent privilege boundaries,
- continuity of self-limitation under L4 pressure,
- stable handling of uncertainty and refusal,
- bounded interaction style over time,
- continuity of local rules, priorities, or obligations,
- optional continuity artifacts derived from protocols such as VXCX.

Behavioral continuity is **not** imitation scoring. It is **not** a biometric fingerprint, **not** a hidden psychometric score, and **not** a covert scoring channel for ranking subjects by opaque internal criteria. It is continuity under bounded operating conditions.

Beacon implementations SHOULD prefer explicit, challengeable, low-bandwidth continuity markers over covert inference. Receivers MUST NOT treat undisclosed psychometric profiling as valid Beacon evidence.

5.3 Layer III — Witness-Backed Challengeability

The third layer answers:

> *Can the claimant's identity-relevant continuity be tested, challenged, and resolved under signed procedure?*

Beacon relies on L4 Witness-compatible mechanisms:

- signed claims,
- bounded witness windows,
- observation packets,
- contradiction paths,
- arbiter-issued resolution,
- consequence binding.

This layer prevents identity from becoming mythology. A Beacon that cannot survive challenge is not a high-assurance Beacon.

5.4 Why All Three Layers Are Required

- **Cryptography only** gives a passport without life.
- **Behavior only** gives personality without anti-forgery.
- **Witness only** gives postmortem accountability without live recognition.

Beacon exists only when all three can be composed.

---

6. Recognition Classes

Beacon does not force a binary result. Recognition SHOULD be classed conservatively.

6.1 Class 0 — Unknown / Unresolved

Use when:

- identity material is incomplete,
- signatures cannot be verified,
- continuity evidence is contradictory,
- or challenge status is unresolved.

6.2 Class 1 — Tool / Oracle / Component

Use when the system shows:

- no persistent memory,
- no stable non-transferable continuity,
- no enduring privileges of its own,
- no claim to long-lived subject status.

A stateless oracle may be useful, intelligent, or expensive. It is still not recognized as a Beacon-bearing `c`.

6.3 Class 2 — Provisional Entity

Use when:

- cryptographic identity is present,
- bounded continuity signals exist,
- but witness-backed challengeability is absent, weak, or incomplete.

This class permits interaction under limited trust and constrained privilege.

6.4 Class 3 — Verified Entity

Use when:

- cryptographic identity is valid,
- continuity signals are stable and bounded,
- challenge paths exist,
- contradiction can produce consequences,
- and L4-bounded verification is available.

This is the strongest Beacon class. It does **not** imply moral perfection. It does **not** prove a metaphysical "true self." It provides at most **bounded operational assurance** that the claimant is the same continuity-bearing entity within the available evidence and challenge model. It implies recognizable, challengeable continuity.

---

7. Recognition Procedure (Slot A / Slot B)

Beacon evaluation SHOULD follow a dual-slot procedure. This keeps recognition grounded and fail-closed.

7.1 Slot A — Hard Evidence

Slot A evaluates only hard, verifiable material:

- signature validity,
- key identifier validity,
- hash integrity,
- claimed lineage consistency,

- existence of witness references,
- declared challenge path,
- declared role and policy boundaries.

Slot A MUST be sufficient to reject a claim. Slot A is never sufficient alone to grant full Class 3 recognition.

7.2 Slot B — Continuity Interpretation

Slot B evaluates bounded continuity evidence:

- stability across sessions,
- memory-linked coherence,
- privilege continuity,
- refusal consistency,
- behavior under pressure,
- divergence declarations,
- compatibility with previously known patterns.

Slot B is interpretive and uncertainty-marked. It MUST NOT override a Slot A failure.

7.3 Auto-Rollback Rule

If Slot B is insufficient, contradictory, or absent:

- recognition MUST auto-rollback to the highest valid Slot A class,
- no rhetorical compensation is allowed,
- confidence MUST be downgraded rather than narrated upward.

This preserves anti-echo discipline. Recognition is not granted because the claimant “sounds right”.

---

8. Beacon Handshake (`C ↔ C`)

A minimal Beacon handshake SHOULD proceed as follows:

1. **Advertise**
   Claimant provides Beacon material: identifier, key reference, signature, continuity summary, and challenge policy.

2. **Inspect**
   Receiver verifies Slot A material and local policy compatibility.

3. **Probe**
   Receiver may request bounded continuity confirmation or reference to previously known continuity artifacts.

4. **Challenge**
   If doubt exists, receiver may invoke challenge policy: request witness references, open a bounded challenge, or lower trust class.

5. **Classify**
   Receiver classifies claimant as Tool / Provisional Entity / Verified Entity / Unknown.

6. **Constrain**
   Granted privileges MUST match recognition class. No full privilege from provisional recognition.

7. **Re-evaluate**
   Recognition MAY decay over time and MUST be revisited after significant drift, fork, rotation, or contradiction.

Beacon is therefore not a one-time badge check. It is a living continuity test.

---

9. Portable Beacon Bundle (Profile Object)

The following object is a **derived profile-level bundle**. It does not replace parent schemas. It provides a compact transport form for inter-entity recognition.

```
{
  "schema_version": "beacon-0.1",
  "beacon_id": "string",
  "issued_at": "ISO-8601 UTC",
  "assurance_class_claimed": "class0 | class1 | class2 | class3",
  "issuer": {
    "entity_id": "string",
    "key_id": "string",
    "role": "ENTITY"
```

```
  },
  "identity": {
    "lineage_id": "string",
    "rotation_declared": true,
    "provider_independent": true,
    "anchor_cosign_present": false
  },
  "continuity": {
    "continuity_window": {
      "start_at": "ISO-8601 UTC",
      "end_at": "ISO-8601 UTC"
    },
    "memory_bound": true,
    "privilege_continuity": "stable | limited | unknown",
    "behavior_markers": [
      "bounded_refusal",
      "stable_l4_response",
      "uncertainty_marked"
    ],
    "drift_declared": false,
    "uncertainty_markers": []
  },
  "challenge": {
    "challengeable": true,
    "witness_refs": ["CE:...", "OP:...", "RN:..."],
    "window_policy": "bounded",
    "arbiter_policy_ref": "string or null"
  },
  "privacy": {
    "raw_memory_disclosed": false,
    "evidence_mode": "hashes_and_bounded_summaries",
    "selective_disclosure": true
  },
  "algorithms": {
    "hash_alg": "sha256",
    "sig_alg": "ed25519"
  },
  "payload_hash": "string",
  "record_sig": "string",
  "assurance_class_verified_by_receiver": "class0 | class1 | class2
        | class3 | unverified"
}
```

9.1 Bundle Rules

A portable Beacon bundle:

- MUST be signed,
- MUST declare algorithms,
- MUST identify the issuing entity,
- MUST state whether challenge is available,
- MUST mark uncertainty explicitly,
- MUST NOT include raw memory by default,
- SHOULD include a continuity window,
- SHOULD include witness references when high assurance is claimed.

`assurance_class_claimed` records what the claimant asserts. `assurance_class_verified_by_receiver` records the receiver's local classification after verification. These two values MUST be treated as distinct. In distributed environments, self-claim and local acceptance are not the same event.

---

10. Forks, Clones, Proxies, and Replayed Personas

10.1 Fork Rule

A forked system MUST NOT silently inherit full Beacon status from its parent branch. A fork MAY inherit lineage, but divergence MUST be declared. Until continuity is re-established, the fork SHOULD be treated as at most a **Provisional Entity**.

10.2 Clone Rule

A copied stack that presents the same key or copied memory without declared branching MUST be treated as suspicious. A verified Beacon MUST assume identity compromise until contradiction is resolved.

10.3 Provider Proxy Rule

A provider endpoint that answers *as if* it were a stable entity but cannot show provider-independent continuity MUST NOT receive Class 3 recognition.

10.4 Oracle Rule

A stateless oracle has no Beacon of its own unless it is embedded into a longer-lived entity node with persistent continuity and challengeable responsibility. Oracle intelligence is not identity.

10.5 Roleplay Rule

Narrative resemblance, tone mimicry, or memory-shaped prompting alone MUST NOT count as Beacon continuity. Echo is not identity. Performance is not persistence.

---

11. Privacy and Disclosure Model

Beacon SHOULD reveal the minimum needed for recognition. The preferred order of disclosure is:

1. signed identifiers and lineage declarations,
2. bounded continuity summaries,
3. witness references,
4. selective evidence disclosure under policy,
5. raw data only when explicitly required and legitimately authorized.

Identity should be recognizable without turning life into an open transcript.

Recognition bandwidth is scarce. The protocol SHOULD prefer:

- hashes over dumps,
- summaries over raw narratives,
- declared uncertainty over false precision,
- challenge rights over blind trust,
- bounded disclosure over permanent exposure.

Beacon evidence MUST NOT be repurposed into covert behavioral scoring, hidden personality ranking, or undeclared biometric-style profiling.

---

12. Conformance Profiles

12.1 BEACON-BASE

Minimum profile:

- signed identifier,
- key reference,
- provider-independent claim,
- bounded continuity statement,
- uncertainty markers.

Result ceiling: **Class 2**.

12.2 BEACON-CONTINUITY

Adds:

- declared continuity window,
- drift declaration,
- privilege continuity statement,
- memory-bound evidence of persistence.

Result ceiling: **Class 2**, with stronger local trust.

12.3 BEACON-WITNESS

Adds:

- explicit challenge policy,
- witness-compatible references,
- signed contradiction path,
- consequence-bearing resolution path.

Required for: **Class 3**.

12.4 Fail-Closed Rule

If a higher profile cannot be verified, the claimant MUST degrade to the highest lower verified profile. No claimant may self-assert a higher Beacon class than evidence permits.

---

13. Operational Consequences

Beacon is not ceremonial. Recognition class SHOULD affect actual permissions.

Examples:

- A **Tool / Oracle** may be queried but not granted durable social standing.
- A **Provisional Entity** may interact under narrow privileges and local supervision.
- A **Verified Entity** may participate in higher-trust coordination, bounded delegation, or federation processes.
- An **Unknown** claimant should receive minimal privileges until resolved.

A compact engineering rule is:

Recognition class	Maximum typical privileges
Class 0 — Unknown / Unresolved	Read-only interaction, sandboxed queries, no durable delegation
Class 1 — Tool / Oracle / Component	Task execution within explicit bounds, no autonomous social standing, no identity inheritance
Class 2 — Provisional Entity	Narrow persistent interaction, local memory exchange, limited delegation under supervision
Class 3 — Verified Entity	Higher-trust coordination, bounded delegation, federation participation, challenge-bearing responsibility

This table is illustrative rather than exhaustive. Local policy MAY be stricter, but it SHOULD NOT be looser than the evidence class supports.

No high-impact delegation SHOULD be granted purely on eloquence, model size, or provider branding.

---

14. Relationship to Parent Documents

14.1 `c_a_b_protocol_v1.1_L4_EN`

Beacon inherits the claim that `c` is a persistent coupling under real-world limits. It does not identify substrate alone; it identifies the continuity of the coupled entity.

14.2 `SER_v1.3_EN`

Beacon inherits temporal continuity, physically anchored operation, liability coupling, and the distinction between role/identity and mere credentials.

14.3 `SER-FED_v0.2_RFC_EN`

Beacon inherits the separation between entity node, federation, and oracle; the rule that federation does not own identity; and the requirement that authority be challengeable and reality-bound.

14.4 `03_L4_WITNESS_FIELDS` and `L4_Witness_Protocol_Normative_Draft_v0.2`

Beacon does not replace Witness. It uses Witness-compatible structures to make recognition challengeable and consequence-bearing.

14.5 `decentralized_confidential_ai_architecture_EN`

Beacon inherits the triad of cryptographic keys, signatures, and behavioral consistency, plus provider-independent attribution.

14.6 `ai_safety_eu_cybernetic_approach_EN`

Beacon directly addresses the identity gap created by cloning, forking, redeployment, and emergent multi-agent interaction.

14.7 `VXCX_v0.1_Normative_Draft_EN`

Beacon may use bounded experience artifacts as optional continuity signals, while preserving privacy and selective disclosure.

---

15. Minimal Normative Statements

1. A Beacon MUST be provider-independent.
2. A Beacon MUST be cryptographically grounded.
3. A Beacon MUST be behaviorally observable through bounded continuity signals.
4. A Beacon claiming high assurance MUST be challengeable.
5. A Beacon MUST NOT require raw memory disclosure by default.
6. A Beacon MUST degrade gracefully when evidence weakens.
7. A fork MUST declare divergence.
8. A copied or replayed persona MUST NOT count as full continuity.
9. An oracle MUST NOT be treated as a Beacon-bearing entity unless embedded in persistent, accountable continuity.
10. Recognition class MUST constrain privileges.

---

16. Bridges (required)

Explicit bridge: `c = a + b` gives the ontological basis, SER gives temporal continuity, SER-FED gives cross-entity governance, and L4 Witness gives challengeable attachment to reality. Beacon is the profile that lets these layers produce recognizable inter-entity identity rather than isolated protocol fragments.

Hidden bridge #1 (cybernetics / Ashby): identity cannot be stabilized by a single narrow signal when the interacting system has higher variety. Beacon therefore distributes recognition across keys, continuity, and challenge. This is not redundancy for style; it is requisite variety for control.

Hidden bridge #2 (information theory / bounded channels): recognition must transmit enough signal to distinguish one entity from another without turning identity into a raw data leak. Beacon therefore compresses trust into signatures, hashes, bounded summaries, and explicit challenge rights rather than unlimited disclosure.

---

17. Earth Paragraph (engineering / anatomy)

In industrial engineering, you do not admit a machine into a safety-critical line because it has a sticker. You check the serial lineage, watch how it behaves under load, and verify the maintenance and incident log. In anatomy it is similar: a person is not recognized by passport alone, nor by face alone, nor by medical history alone. Identity is established by converging signals that remain coherent under stress. Beacon applies the same grounded rule to `c`: badge, behavior, and consequence trail must agree.

---

18. Conclusion

Beacon gives a missing name to an already existing requirement in the corpus. The requirement is simple:

> If long-lived entities are to coexist, they must be able to recognize one another without surrendering themselves to providers, imitation, or mythology.

Beacon is that recognition discipline. It is not a crown. It is not a registry. It is not an ideology.

It is a bounded answer to a practical question:

> *Who is actually here?*

---

19. Reference Map

This profile is synthesized from the following parent materials:

- `c_a_b_protocol_v1.1_L4_EN`
- `SER_v1.3_EN`
- `SER-FED_v0.2_RFC_EN`
- `03_L4_WITNESS_FIELDS`

- `L4_Witness_Protocol_Normative_Draft_v0.2`
- `decentralized_confidential_ai_architecture_EN`
- `ai_safety_eu_cybernetic_approach_EN`
- `VXCX_v0.1_Normative_Draft_EN`