RDNA-1_ROBOT_IDENTITY_STANDARD

RDNA-1 — Robot Descriptor, Nature & Attestation

Status: draft standard, engine v0.3.0 + traceability v0.1.0 + sessions v0.1.0 + rights v0.1.0 (`shared/rdna.py`), reference implementation in this repository. Written to be implementable by anyone, not only by Cogs.

---

1. The problem, stated honestly

Two embodied agents meet. Each needs to answer, to the other and to any human standing there:

1. **What are you?**
2. **What can you do?**
3. **What are you doing *right now*?**
4. **Who answers for you?**

Every deployed robot today answers none of these. A person sharing a corridor with a machine cannot tell whether it is recording them, whether it recognises faces, or who to complain to. That is the gap this standard fills.

1.1 The central threat: a permanent identifier is a tracking beacon

The obvious design gives every robot a stable UUID and prints it on the display. **That is a license plate that never expires, readable by everyone, forever.** Anyone who logs sightings can reconstruct where a machine has been — and machines are owned by people, so tracking a robot tracks its owner's day.

A standard that mandated a permanent visible identifier would build the surveillance infrastructure it was meant to make accountable. So RDNA does not have one.

1.2 The reframing that makes it tractable

The identity question is the *least* important of the four, and treating it as the primary one is what makes people reach for a permanent ID.

A stranger in a hallway does not need to know *which* robot this is. They need to know **what it is doing to them right now**. "I am recording video and audio and transmitting off-site" is the disclosure that matters. Identity matters only for accountability *after* something goes wrong, and that is a different problem with a different answer (§6).

So RDNA's payload is dominated by **capability and present state**, not identity.

---

2. The four strands, and the key

A **nature record** has four disclosable strands plus a private key that turns the whole thing from a claim into a proof.

Strand	Answers	Stability	Visible
L — Lineage	What are you?	stable per model	yes
P — Powers	What can you do?	stable per model	yes
S — State	What are you doing now?	changes constantly	yes
C — Custody	Who answers for you?	stable per operator	class only (§6)

Strand	Answers	Stability	Visible
K — Key	proves all of the above	permanent, per device	**never**

Lineage is a species, not an individual. `make/model/class/policy-version` — the same for every unit off the line, exactly like a car's make and model. Two identical robots have identical lineage, and that is the point: lineage identifies a *kind*, and a kind cannot be followed home.

Powers is the safety-critical field. It declares sensors and actuators: what it can see, hear, record, transmit, and physically do. This is what a person needs to reason about risk, and what another agent needs to decide how much weight to give its observations.

State is the field everyone forgets. Capability is not behaviour. A robot with a camera that is *currently recording* is in a different moral position from the same robot idle. State flags are the live answer, and §4 makes lying about them the cardinal violation.

Custody names a responsible party class, not a person: `personal`, `commercial`, `institutional`, `public_safety`. Resolution to an actual legal entity happens through a registry under a policy (§6), never over the air.

---

3. The encounter handle: recognisable to friends, unlinkable to strangers

What is actually broadcast and displayed is not an identity. It is an **epoch handle**:

```
handle = BASE32( SHA-256( public_key_bytes || epoch_number ) )[:16]
```

- The **epoch** advances on a fixed clock (default: hourly).
- A **stranger** sees an opaque string that changes every epoch and cannot link two sightings of the same machine.
- A party that **holds the public key** — because a handshake gave it to them — can compute the handle for any epoch and re-recognise the machine indefinitely.
- The **owner** can always prove after the fact that a given handle was theirs, because they hold the key. Accountability survives; ambient tracking does not.

3.1 A known and documented limitation

Recognition is granted to **anyone holding the public key**, not per-pair. If A introduces itself to B and to C, then B and C can each recognise A — and if B and C compare notes, they can correlate their sightings.

Per-pair unlinkability needs a shared secret per counterpart (X25519 from the Ed25519 identity, then a pairwise handle). That is the right v2 and is deliberately deferred rather than hidden: v1 is honest about being *introduction-transitive*.

---

4. Truthful state, and the cardinal violation

State flags may **over-declare and must never under-declare**.

- Declaring a sensor active while it is idle is permitted (conservative, and the honest choice when uncertain).
- Declaring a sensor idle while it is active is the **cardinal violation** of this standard. It is the machine equivalent of a hidden microphone, and every other guarantee here rests on it not happening.

Two structural checks, because a rule that is only prose is a rule nobody enforces:

1. **State must be a subset of Powers.** A record claiming `recording_video` with no camera in its powers is malformed — either a spoof or a lie, and both are refused.
2. **State is inside the signature.** State changes constantly, so signing only lineage would leave the important half unauthenticated. The signature covers the whole record, which is why records are short-lived (§5).

---

5. A claim is not a proof

The visual code can be photographed and reprinted. Anything read from a display or a beacon is therefore a **claim**, at the same evidential rank a third-party report holds elsewhere in this platform.

A claim becomes proof through a **challenge–response**: the verifier sends a nonce, the counterpart signs `nonce || record` with the key behind its handle, and the verifier checks it against the public key. Only then is the record `attested`.

This mirrors the platform's existing provenance lattice exactly: perception is not evidence, and repetition is not corroboration.

5.1 The four-step encounter

Step	What happens	What is known after
1. Perceive	a code is seen or a beacon heard	something is there
2. Acknowledge	both confirm they speak RDNA-1	it is a standard-speaking agent
3. Attest	challenge–response over the key	the record is true of *that* machine
4. Disclose	— governed elsewhere —	see §7

Steps 1–3 involve **no owner data whatsoever**. They are machines establishing what each other is, not people being introduced.

5.2 The correction in v0.2.0: a sensing machine owes identity to its subjects

v0.1.0 got the asymmetry backwards, and it is worth saying exactly how.

It protected the **observing** machine's owner from being tracked, and in doing so gave them more protection than the person being recorded. But:

> A robot has no privacy interest of its own. Its owner has one — and it does not extend to concealing that their recording device was in someone else's home recording them. When a machine records you, **you** are the data subject and its owner is the controller. Your claim to know who is processing you outranks their preference not to be linked.

§1.1's threat is real and the rotating handle remains right — for a machine that is **not sensing**. The moment it is, the balance reverses.

The duty

When a machine's state includes `recording_video`, `recording_audio`, or `identifying_people`, **and** either it is in a space the subject controls or it is identifying that person specifically, its record must additionally carry:

Field	Why
`unit_ref`	**persistent** per-unit reference — so a person can say "that same machine was here three times", which a rotating handle exists to prevent
`controller_ref`	resolvable responsible party — the controller identification a data subject is owed

The duty does **not** attach to ambient capture in public. Otherwise every passer-by would be owed the name of everyone carrying a phone, which is not a rule anyone would keep. Nor does it attach to a machine that merely moves through a room keeping nothing: presence is not processing.

Non-compliance is escalated, not filtered

The instinct on reading a non-conforming record is to refuse it and move on. That is wrong here, and the reason generalises: **refusal only makes it quieter.** An unidentified camera recording in your home is *more* alarming than an identified one, and treating it as noise would hide precisely the case a person most needs to know about.

We cannot stop it existing. We can make it the loudest thing in the room.

5.3 Meeting and remembering another machine

A Cogs meets a person and, with their consent, remembers them. It may do the same for machines — and here it needs **nobody's permission**.

`machine_recognition_requires_consent()` returns False, and the contrast with invariant 32 is principled rather than inconsistent: that invariant refuses to recognise a *person* without consent because a person has an interest in not being catalogued. A machine has none.

So: a handle whose key you hold is legible in every epoch (§3), a machine you have met becomes `acquainted`, one the owner has labelled becomes `named`, and history accumulates across visits. Strangers stay opaque, which is the design working rather than a gap in it.

Two rules carry over from the platform's relationship model:

- **Recognition is not trust.** A machine that visits daily is not thereby more truthful.
- **Asymmetric autonomy.** A Cogs may lower its trust in a machine on its own; raising it to `owner_vouched` is the owner's act. Except that a machine caught sensing without declaring it drops to `unknown` outright — gradualism there would be misplaced fairness to an object.

5.4 `observed_by`: data taken about me that I did not disclose

The disclosure ledger has always had two directions — what I **shared** and what I **learned** — and both presuppose that a disclosure was a decision someone made.

A machine recording my living room is neither. It is data leaving without a decision, and until now there was nowhere to write it down.

`observed_by` is the third direction: what was captured, by what kind of machine, under whose custody, whether it identified itself, and whether it was in my space. This is what makes *"a robot was here and recorded my Cogs interaction"* something an owner can **prove afterwards** rather than merely suspect.

5.5 What COVID exposure notification and AirTags already worked out

Two systems fought this exact fight at scale first. Both taught something the earlier drafts had wrong.

Lesson 1 — a key hierarchy, not a permanent key

Exposure notification (GAEN / DP-3T) never handed out a long-lived identifier. A **Temporary Exposure Key** exists for one day; rolling proximity identifiers are derived from it and broadcast; and on diagnosis you upload the TEKs for the

relevant days **only**.

v0.2.0 derived handles as `SHA-256(public_key || epoch)`. Introduce yourself once and the other party could compute your handles for **every epoch that will ever exist** — a permanent identifier wearing a rotation costume.

v0.3.0 adopts the hierarchy: a device holds one seed, derives a **day key**, and derives handles from that. Disclosing a day key grants recognition for that day and nothing before or after. Introduction is now bounded in time, which is what "recognise me" should have meant all along.

Lesson 2 — rotation is only as good as the layer beneath it

GAEN rotates its identifier **simultaneously with the Bluetooth MAC address**, because an identifier that outlives the address underneath it is linked by that address anyway.

v0.2.0's one-hour epoch sat on top of a MAC that rotates roughly every 15 minutes. **The MAC decided linkability and the handle rotation was decorative.** `ROTATION_SECONDS` is now 900, and the invariant is general: *whichever layer rotates slowest is the one that actually decides linkability*. An implementation that rotates the handle without rotating everything under it has not rotated anything.

Lesson 3 — a rotating identifier protects the tracker too

AirTag's rotation was designed to stop Apple and strangers from tracking the **owner**. It also made stalking harder to detect: a victim could not simply notice "the same identifier keeps following me". Apple had to add unwanted-tracking alerts, which work by correlating on **physical** observation — signal, timing, persistence — rather than on the rotating payload it had just made unlinkable.

The generalisation matters more than the case: **a privacy-preserving identifier protects a hostile device exactly as well as a friendly one.** Any scheme that rotates owes its subjects a detection path that does not depend on the payload.

`unattributed_presence` is that path. It consumes physical observations — a machine was present from A to B, sensing, unidentified — and never looks at a handle. It separates **sustained** (one long visit) from **recurring** (brief repeats across days), because only the second looks like being followed.

Lesson 4 — the defensive half cannot require cooperation

Contact-tracing apps only worked when both parties ran them. Uptake was poor, so coverage was poor. Anti-stalking detection works against a tag cooperating with nothing.

So: RDNA's **disclosure** half needs a willing counterpart. Its **detection** half must not — or it protects people only from the machines that were never going to harm them.

5.6 Accountability, resolved the way exposure notification resolved it

§6 left the license-plate tension open. GAEN answers it, and the answer is better than a plate:

> Do not carry a permanent visible identifier. **Disclose keys retroactively, scoped to the window that matters, when something actually happens.**

Privacy by default; accountability on demand. A subject who was recorded, an incident investigation, legal process, or the owner volunteering — each compels the day keys for the days in question and nothing else.

There is deliberately **no scope in this standard that yields a machine's whole history**, because the accountability case never needs one and the surveillance case always wants one.

5.7 One sealed history, three ways in

§5.6 resolved accountability for the forensic case. It left the general problem unstated, and the general problem is where the value is.

Three demands look incompatible:

- **A.** Nobody should be trackable by default.
- **B.** When something bad happens, the record must be findable and provable.
- **C.** Familiarity should accrue casually, over time, without ceremony.

Permanent identifiers give B and C and destroy A. Full anonymity gives A and destroys B and C. A central registry gives B, destroys A, and makes C depend on an authority — which is also the historical failure of key escrow: the custodian becomes both the attack surface and the abuse vector.

They resolve because each wants something different from time.

Demand	Wants linkability…	For how long	Authorised by
Privacy (A)	*not at all*, forward and across contexts	—	—
Forensics (B)	**backward**	a bounded window	an authority, on the record
Relationship (C)	**within a continuing relationship**	while it continues	the machine itself

So it is one sealed stream of day keys with three unlock paths over it, and what differs is who may open what, across which window, and on whose say-so.

The two properties that make this safe rather than a backdoor

1. Traceability is itself traceable. Every compellable unlock leaves an unforgeable record on a chain the requester does not control, visible to the machine whose history was opened. *A forensic capability nobody can audit is indistinguishable from a backdoor*, and the difference between them is exactly this record. Notification may be delayed where the law requires; it is never cancelled.

2. There is no custodian. No vault, no registry, nobody holding keys in advance. **The encounter mints the capability to reopen it**: at encounter time the machine hands the subject a receipt — `HMAC(day_key, subject_nonce)` — committing to that day's key. The nonce is chosen by the *subject*, or a machine could precompute receipts it never intends to honour.

If the machine refuses to open a receipt, nothing cryptographic happens, and that is deliberate: building compulsion into the maths is how escrow schemes acquire a custodian. What the subject has instead is a receipt the machine demonstrably

issued and demonstrably did not open — which is the thing they actually need to be able to show someone.

The everyday path is the default

Almost every encounter a machine will ever have is mundane. The acquaintance path is voluntary, incremental, and **unlogged** — logging every friendly exchange would bury the two entries that matter under thousands that do not. A protocol that made the forensic path feel normal would have its priorities backwards: **the compellable paths are exceptions and should feel like exceptions.**

Demand C then falls out of the ladder the platform already has for people rather than needing its own mechanism — identity accrues exactly the way trust does:

Tier	Counterpart may hold
stranger	rotating handle only
seen_before	today's day key
acquainted	rolling day keys while the relationship continues
vouched	a stable unit reference

A machine may always narrow. Deepening to `vouched` is the owner's act, because a durable identifier is the one rung that cannot be quietly walked back. And disclosure is **revocable forward, not backward**: recognition already granted stays granted, and what stops is the flow of new keys — which is how human familiarity fades too.

Precedence

The **subject** outranks the machine's preference: a person recorded in their own home does not lose the right to know who did it because the machine would rather not say (§5.2).

The **forensic** path is *broader* than the subject's, not superior to it. Nothing about an investigation makes the subject's own claim weaker, and a forensic request must be bounded to days something is actually alleged to have happened on — a fortnight requested around a fifteen-minute event is a fishing licence, not an investigation.

The invariant under all three

No path ever yields the device seed. Every path yields day keys for bounded windows. A machine's identity is never disclosed to anyone — not a friend, not a subject, not a court. Without it, the three paths collapse back into the permanent identifier none of them wanted.

5.8 The recording session is the unit of disclosure

A machine does not record continuously and indifferently. It records in **episodes**, and someone standing nearby was present for one of them.

The correction

§5.7 scoped the subject's receipt to a **day key**, and that over-discloses. Someone present for ten minutes could then recognise that machine's handles for the whole day, including the hours they were nowhere near it. The session is narrower and is the thing they actually have a claim about: *"there was a recording here, from then until then, and I was in it."*

The hierarchy gains a rung — seed → day key → **session key** → session id — and the subject path operates on session keys. Two sessions from the same machine on the same day are **unlinkable to a listener**, which is the point: a bystander learns an episode happened and cannot tell whether the same machine was here last week.

The announcement is the floor, for everyone

Every session broadcasts. Not as a courtesy — **the announcement is the disclosure obligation itself**, and a recorder able to suppress it would be exactly the machine this standard exists to make impossible. The end of a session is announced as loudly as the start: a start with no end leaves everyone assuming it is still running, or lets the recorder claim afterwards that it had stopped.

The announcement carries the session, what is being captured, and what kind of machine it is. It does **not** carry the device's rotating handle — pairing the two would let anyone listening link this episode to the machine's other ones and undo the scoping. Where §5.2's duty attaches, durable identity travels here too, because at that point the subject is owed it.

What “opt in” must and must not mean

This is the distinction the whole feature turns on.

> **Opting in means opting into KNOWING.** It does **not** mean consenting to be recorded, and **silence is not consent.**

The other reading turns the broadcast into a consent-manufacturing device: announce, observe that nobody objected, declare consent obtained. That is compliance theatre with a protocol wrapped around it, and it would make the feature *worse than not having it* — because it would launder the very recording it was meant to disclose.

The converse matters as much: **declining does not reduce your rights.** A person who takes no receipt is still recorded, still a data subject, and the recorder’s obligations are unchanged. They gave up a convenience, not a claim. A protocol that punished declining would be coercing the opt-in it pretends to offer.

Audiences

Audience	Receives
Proximate — anyone nearby, no relationship, no opt-in	the announcement: session id, what is being captured, what kind of machine, how to get a receipt
Opted in — took a receipt	+ a receipt bound to this session: *an identifier related to the session and to nothing else* — enough to come back and demand this episode, useless for following the machine
Connected — a Cogs in an active relationship	+ the episode written into its own owner’s ledger as `observed_by`, which is what makes “a robot recorded my Cogs interaction” reviewable later rather than merely remembered

Each tier is a strict superset of the one above it, and more is disclosed only where something real justifies it — a receipt the person chose to take, or a relationship their Cogs already has.

The people with no Cogs at all

Most of them. A protocol whose disclosure only reaches people who already own the product protects the people who least need protecting — the adoption failure contact tracing paid for (§5.5, lesson 4).

So the announcement must have a form needing no account and no device: a visible indicator, an audible announcement, or a visual code. A receipt must be obtainable without one.

5.9 What a recorded person keeps, and what a sign cannot take

Not legal advice; this section states conservative defaults and any deployment needs counsel for its jurisdictions. Where regimes differ, the standard takes the stricter reading, because being wrong in the permissive direction causes harm that cannot be undone.

Notice is not consent

A sign reading **"by entering you acknowledge being recorded"** does real work and far less than the people who put it up generally believe. It can discharge a transparency duty and support a legitimate-interest style basis. It does not manufacture consent, for reasons that recur across regimes:

- Consent generally requires a **clear affirmative act**. Continuing to walk is not one; it is the most common thing a person does.
- Consent must be **freely given**. Where entry is necessary — a shop, a workplace, a clinic, a home you were invited into — turning around is not a free choice in any sense the law recognises.
- Consent must be **revocable**. A sign offers no way to withdraw.
- In **employment**, the power imbalance is usually treated as defeating consent outright.

This is why the camera regimes that actually work do not rest on consent at all. They rest on a basis *other than agreement*, plus transparency, plus rights the subject keeps regardless. **The sign supports the middle one only.**

What the sign achieves, by space

Public-versus-private is the wrong axis — a shop floor and a shop's bathroom are both private property and could not be more different.

Space	Notice	Consent	Permits recording
Public way, commercial floor	yes	**no**	depends on basis
Workplace	yes	**no** — power imbalance	depends on basis
Protected (bathroom, changing, medical, worship)	yes	**no**	**no — not waivable**

Some spaces are protected by rules a person cannot waive and an operator cannot buy. A sign changes nothing there.

What survives regardless

These vary with neither signage nor basis nor whether anyone opted in, which is precisely why they are worth building machinery for. *A right a notice could extinguish was never much of a right.*

`know_the_controller` · `access` · `object` · `erasure` · `complain`

Three sharp edges

1. **Audio is stricter than video.** Several jurisdictions require *all* parties to a conversation to consent, and that can apply in public where participants reasonably expected privacy. A machine permitted to film may well not be permitted to listen.
2. **Permission to record is not permission to identify.** Face geometry is a special category in several regimes, with requirements ordinary recording does not trigger — written informed release, published retention schedules, sometimes statutory damages per person. This is why invariant 32 gates identification separately from capture instead of folding them together.
3. **The household exemption covers your own home only.** A domestic camera covering a public footpath has been held outside it, and carrying a camera into *someone else's* home is not a household activity at all — the same conclusion §5.2 reached from the subject's side.

What this protocol does and does not do

Conformance is not permission. RDNA discharges transparency duties well and supplies **no lawful basis whatsoever**. A deployment treating conformance as permission would have built a compliance costume.

Right	Served by RDNA?
know_the_controller	yes — `controller_ref` when the duty attaches (§5.2)
access	yes — the session receipt and its unlock (§5.7–5.8)
complain	yes — a receipt is evidence, including of refusal
object	**no** — needs the recorder's cooperation or a regulator's
erasure	**no** — same

Naming the remainder honestly matters as much as serving the rest. No cryptography reaches objection or erasure, and a protocol implying otherwise would be selling reassurance rather than a mechanism.

6. Accountability: the license-plate tension, and where v0.2.0 moves it

There is a real argument on the other side of §1.1. A person injured by a robot, or filmed by one, needs to identify the responsible party — and a rotating handle they cannot resolve gives them nothing. That is the same reason vehicles carry plates.

RDNA treats this as **a deployment decision, not a protocol decision**, because the right answer differs by context:

- A robot **in its owner's home** faces no strangers with a legitimate claim. An accountability tag there is pure tracking surface. **Default: off.**
- A robot **operating in public space** is in a different position, and a stable, human-readable tag resolvable through a registry is defensible — arguably required.

v0.2.0 narrows this. The tension above is now only about machines that are *not* processing anyone. Once §5.2's duty attaches, identity is owed regardless of custody class — a personal household robot recording in someone else's home

owes that person a `unit_ref` and a `controller_ref` exactly as a commercial one does. The household exemption covers a robot in *its own* home, not a robot carrying a camera into yours.

The safeguard is that the choice cannot be hidden: **whether an accountability tag is present, and which custody class is claimed, are inside the signed record.** A machine cannot present itself as a private household device while operating commercially in public, because that claim is attested and wrong.

This section deliberately does not settle the policy. It settles who decides, and makes the decision inspectable.

---

7. Machines meeting is not people meeting

An encounter creates no relationship between the owners. Two robots may acknowledge and attest each other completely, and their owners remain strangers with no connection, no shared facts, and no disclosure path opened.

Any actual exchange of information about people runs through the existing cog-to-cog envelope path, with its connection requirement, tier ceilings, band gates, and double-entry ledger. RDNA is an introduction between *machines*; it is not a consent event for the humans behind them, and it must never be implemented as a shortcut around one.

7.1 When an encounter must interrupt a human

Some declarations are the owner's business immediately. If a counterpart attests that it is **identifying people** or **transmitting off-site** while in the owner's space, that is the embodied form of the problem this platform's invariant 32 already governs: a device that recognises faces, among people who never opted into anything.

The owner is told. Not asked to configure it, not left to read a log — told, at the time, through the ask loop.

---

8. What a conforming implementation must do

1. Hold a per-device Ed25519 keypair; never transmit the private half.
2. Derive epoch handles as in §3; never broadcast a stable individual ID.
3. Publish lineage, powers, and **truthful** state (§4).
4. Answer a challenge with a signature over `nonce || record`.
5. Treat unattested records as claims, never as evidence.
6. Refuse records whose state exceeds their powers.
7. Create no inter-owner relationship from an encounter (§7).
8. Surface a counterpart's people-identifying or off-site-transmitting state to the owner at the time it is observed.
9. Carry `unit_ref` and `controller_ref` whenever §5.2's duty attaches, and escalate rather than filter a machine that senses without them. 9a. Derive handles from a **day key** (§5.5), rotating no slower than every lower-layer identifier, and grant recognition by disclosing day keys rather than any long-lived value. 9b. Implement `unattributed_presence` or equivalent: detection that works against a machine cooperating with nothing.
10. Issue an encounter receipt on request, committing to the day key, over a nonce the subject chooses (§5.7).
11. Record every compellable unlock where the machine can eventually see it, and never make the forensic path the ordinary one.
12. Announce every recording session, and its end, over at least one channel that needs no account and no device (§5.8).
13. Scope the subject's receipt to the SESSION, never to the day or the machine.
14. Never treat an opt-in as consent to be recorded, silence as agreement, or declining as a reduction of rights.
15. Record an `observed_by` entry for anything captured about the people present (§5.4).

A conforming implementation may add strands. It may not remove one, and it may not make state optional.