Unified ID Card & Attribute Provenance: Verified vs Self-Asserted Identity Fields

People & In-Person / Last updated 2026-06-11 / https://www.scrambleid.com/learn/unified-id-card-attribute-provenance

> **Status (June 2026):** Early access. The Unified ID Card ships as part of the People verification family, which is live with early-access customers and isn't generally available yet. This article describes the shipped design as it stands today; talk to your ScrambleID account team about access and timelines.

In one sentence: The Unified ID Card is a structured identity view where **every attribute** carries explicit **provenance**, verified (✓) vs self-asserted (•) vs temporarily unavailable (☁), so humans and policies can safely interpret what a card does (and does not) prove.

TL;DR (canonical)

- Most "digital IDs" obscure provenance, so viewers can't tell which fields are verified and which are self-asserted.
- ScrambleID makes provenance visible at the **field level** (not just "user is verified").
- Provenance enables **data minimization**: share the minimum verified subset needed.
- Provenance prevents "verified theater": attackers can't upgrade self-asserted claims into verified ones.

What is attribute provenance?

Attribute provenance is the field-level metadata that tells you who verified a value, how, and whether you should trust it. It answers:

> **Who verified this field, how, and should I trust it?**

ScrambleID treats provenance as a first-class security property:

- An attribute value without provenance is **not safe** to use in policy.
- Provenance is what makes the Unified ID Card meaningful to both humans and machines.

How does provenance map to NIST 800-63?

NIST separates broad *identity proofing* from narrower *attribute collection/validation*. ScrambleID provenance is primarily about **attribute-level evidence**, what the field came from and how it was checked, not a blanket statement that the entire person has been proofed at a specific IAL.

High-level translation table:

ScrambleID concept	What it means in practice	Roughly corresponds to in NIST language
Verified ✓ (HRM/authoritative)	field value comes from an authoritative system of record	attribute collection from authoritative source
Verified ✓ (domain proof / OAuth)	field value is validated via control of a domain/account	attribute validation / verification method
Verified ✓ (liveness / biometric unlock)	a bound device user verified at release time	authenticator/user verification (contextual)
Self-asserted •	user typed it	asserted attribute with no validation

Important: this is a **conceptual mapping** to help reviewers and auditors. Formal assurance levels depend on your end-to-end proofing and governance program.

Example provenance object (illustrative)

```
{
  "fieldId": "F8",
  "value": "user@company.com",
  "provenance": {
    "mark": "VERIFIED",
    "source": "HRM",
    "method": "SCIM_SYNC",
    "verifiedAt": "2026-01-18T18:02:00Z",
    "expiresAt": "2027-01-18T00:00:00Z"
  }
}
```

What do the trust indicator marks mean?

These markers must remain consistent across contexts (People, Verify-Me (in development), previews):

- **Verified check ✓**, the field is backed by an authoritative or validated source.

- **Self-asserted dot •**, the field is user-entered; treat as unverified.
- **Temporarily unavailable ~~•~~**, the field is pointer-based and can't be resolved right now.

What attributes does the Unified ID Card include?

> This is the stable "what fields exist and how to interpret them" contract.

ID	Attribute (UI label)	Sensitivity	Default profiles	Verified source (examples)	Self-entry allowed?	Trust mark
F1	Avatar / Photo	Low	All	Face liveness check	Yes	✓ if liveness passed
F2	Handle	Low	All	System-generated checksum	No	✓
F3	Human-Verified ring + Freshness	Low	All	Device key + biometric/passcode unlock	N/A	✓ (color-coded)
F4	Legal Name	Low	Work, Personal	HRM (e.g., Workday) or gov-ID scan	Yes	✓ if HRM/KYC ı • if self
F5	Company / Employer	Low	Work	HRM	Yes (custom)	✓ or •
F6	Job Title	Low	Work	HRM	Yes	✓ or •
F7	Department	Low-Med	Work	HRM	Yes	✓ or •
F8	Work Email	Med	Work	HRM sync or domain-control proof (confirmation link)	Yes	✓ if domain control proven ı • (bare MX does not qualify)
F9	Work Phone	Med	Work	HRM or SMS OTP	Yes	✓ or •
F10	Personal Email	High	Personal	OTP link	Yes	✓ if OTP
F11	Mobile / Personal Phone	High	Personal	SMS OTP	Yes	✓ if OTP
F12	LinkedIn URL	Low	Work, Personal	OAuth	Yes	✓ or •
F13	Location (City, Country)	Med	Custom	IP geo or self-entry	Yes	• (IP-derived is inferred, never ✓; coarse and VPN-spoofable)
F14	Mutual trusted contacts	Low	All	Graph calc	N/A	✓

ID	Attribute (UI label)	Sensitivity	Default profiles	Verified source (examples)	Self-entry allowed?	Trust mark
F15	Invite / CTA	Low	All	System	N/A	N/A
F16	Custom (≤3)	Varies	Work/Personal/Custom	Self	Yes	•

The freshness contract

The Human-Verified ring carries a freshness component with a defined contract. It encodes when the presenter last completed a live device-key ceremony (biometric or passcode unlock signing a fresh challenge): every provenance object carries verifiedAt and expiresAt. When expiresAt passes, the ring degrades from current to stale in every renderer context, and a stale ring cannot satisfy a verifier policy that requires current freshness. Re-verification happens at the next Trust Check the presenter accepts: the ceremony re-signs, resets verifiedAt, and restores the ring.

Why the catalog matters

- It provides a **canonical vocabulary** for policies and UI.
- It lets security teams write rules like:
  - "Allow Work email only if verified ✓ and sourced from HRM or domain proof."
  - "Never treat personal phone as identity proof."

Why does the same card render differently across contexts?

The same underlying card renders differently depending on the moment:

Context	Shown to	Moment	Goal
C-People (Normal)	initiator + receiver	after successful Trust Check	show trusted identity + growth CTA
C-PREVIEW	receiver only	before Accept/Reject	show exactly what will be shared
C-BADGE-COMPACT	public viewer (likely non-user)	Verify-Me badge / smart-link unfurl	teaser + install CTA
C-BADGE-EXPANDED	Scramble user viewer	after tapping "View Details"	show full card with toggles
C-OFFLINE	either party	network down / pointer stale	show core trust + data gaps

The two badge contexts (C-BADGE-COMPACT and C-BADGE-EXPANDED) are forward-looking: they render once Verify-Me ships.

This context system is why a card can be safe across channels, the renderer never "forgets" provenance.

How do picker guardrails prevent oversharing?

Provenance makes minimization enforceable.

- Profiles (Work/Personal/Minimal/Anonymous) are presets, not uncontrolled freeform.
- Sensitive fields default OFF.
- Requirements can enforce minimum fields without forcing maximum fields.

See: ID Card Picker (Consent UX)

What should you log for provenance auditing?

If you are integrating People or Verify-Me, log these *without leaking raw PII*:

- which fields were shared (field IDs only)
- provenance per field (verified/self)
- method + context (People, badge, preview)
- session TTL and consumed status

These logs enable auditability and downstream risk detection.

Key Takeaway

The Unified ID Card presents identity attributes with provenance metadata (verified vs. self-asserted), freshness cues, and context-appropriate rendering. Verified attributes come from authoritative sources (employer directory, government ID scan); self-asserted attributes are user-provided. This transparency helps relying parties make appropriate trust decisions without over-collecting data.

FAQ

What is the Unified ID Card?

A structured identity view with stable field IDs and provenance markers, rendered per context.

What does "verified ✓" mean?

It means the value is backed by an authoritative or validated source (e.g., HRM, domain proof, liveness, OAuth). It does **not** automatically mean identity proofing at the highest assurance.

What does "self-asserted •" mean?

It means the user entered the value. It can be useful context but should not be treated as strong proof.

Does provenance mean you are doing identity proofing?

Not always. Provenance tells you the source and verification method for a field. Formal identity proofing is broader.

Can I add my own verified fields?

Yes, if you have a validation method and can preserve provenance. The minimum bar is auditable verification.

Why include "temporarily unavailable ☁"?

Because pointer-based attributes may not resolve offline; showing this explicitly prevents confusion and overtrust.

References (public)

- NIST Digital Identity Guidelines overview (assurance concepts): https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63-4.pdf
- NIST SP 800-63A (identity proofing concepts and evidence): https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63A-4.pdf

---

Related reading

- People Trust Checks: Overview
- People Verification Implementation Guide
- Verify-Me: Public Identity Badges (in development)