People Verification: An Implementation Guide for Trust Checks and Consent UX

People & In-Person / Last updated 2026-06-11 / https://www.scrambleid.com/learn/people-verification-implementation-guide

> **Status (June 2026):** Early access. People Trust Checks are live with early-access customers and aren't generally available yet. This guide describes the shipped design as it stands today so implementation teams can plan; talk to your ScrambleID account team about access and timelines before scheduling a rollout.

In one sentence: Implement People Trust Checks with three initiation modes (QR, typed code, messaging link), consent-first UX with data minimization, and enterprise policy controls for attribute provenance and verification expiry.

This guide answers: **"How do we deploy People Trust Checks as a reliable, policyable enterprise workflow?"**

TL;DR (canonical)

- People is verifier-initiated: the verifier starts the Trust Check session.
- The presenter always sees a **preview** of what will be shared and must **consent**.
- Sensitive data release is gated behind step-up (device user verification) when policy requires it.
- All request artifacts are **single-use** and **time-limited** (defaults: **60 seconds** real-time QR/code; **24 hours** messaging link).
- The Unified ID Card renderer must preserve **attribute provenance** (verified ✓ vs self-asserted •) and **freshness**.

Product invariants (do not break these)

1. **Verifier-initiated**, the verifier starts the request.
2. **Presenter consent**, the presenter sees exactly what will be shared.
3. **Step-up before share**, for policies that demand it, no attribute release before step-up.
4. **Data minimization defaults**, sensitive toggles default OFF.
5. **Replay resistance**, artifacts are time-limited + single-use.

If you break any of these, the system degenerates back into screenshot theater.

Initiation methods (and when to use each)

Method	Best for	Why it works	Watch-outs
QR	in-person / field ops	fastest, least friction	camera permissions + glare
Short code	voice/video or constrained cameras	low cognitive load; no scanning	ensure code entry UX is tolerant
Smart link	remote initiation	fits messaging and email	link forwarding risk mitigated by single-use + TTL
iMessage app drawer	messaging-native consumer flows	request bubble in-thread	requires iOS extension build

Session model (canonical)

> **Binding to the Trust Check session:** the presenter's app signs a confirmation that includes the verifier-issued session id; the renderer on the verifier's side rejects any payload whose session id does not match the Trust Check it initiated. Full specification of the signed payload, atomic verification, and TTL handling is in Session binding cryptography.

- A session is created by the verifier and returns:
  - `sessionId`
  - one or more user-facing artifacts (QR payload, short code, link)
- Presenter joins via any artifact.
- Presenter selects a compliant profile + optional field toggles.
- Step-up is performed (if required).
- Presenter shares; verifier receives a one-time ID Card view.

Deep link formats (recommended)

Keep request artifacts opaque; do not embed raw PII.

- App scheme (legacy-safe): `scrambleid://people/verify/<token>`
- Universal/App link: `https://scramble.id/people/verify?session=<sessionId>`

All payloads should be integrity-protected (signed) to prevent tampering.

Default lifetimes (policy-controlled)

Recommended defaults:

- **Real-time request (QR/code): 60 seconds** (show a countdown)
- **Messaging request link: 24 hours** (label clearly)
- **Single-use semantics:** once consumed successfully, reuse is rejected

Consent UX requirements (what the presenter must see)

Before the presenter can share, show a consent sheet that includes:

- the verifier identity cues available (name, org, handle, enterprise tier if applicable)
- the requirement pill (Need: Work/Minimal/Anonymous/None)
- the exact fields that will be shared (with provenance badges)
- the session TTL and single-use behavior
- a clear action: **Accept & Share** (gated by step-up if required)

Profile compliance and the ID-Card Picker

The picker adapts to the verifier requirement to reduce friction.

Requirement contract (concept)

```
{
  "requiredProfile": {
    "level": "WORK",
    "mandatoryFields": ["legalName", "companyName", "title"]
  }
}
```

Receiver UI states (canonical)

- **S0 Full Picker:** no requirement set → show all profile tiles
- **S1 Accept/Reject:** requirement set, last-used profile already complies → 1-tap accept
- **S2 Filtered Picker:** requirement set, last-used profile non-compliant → show only compliant tiles

The pattern is the inversion of typical UX: stricter requirements reduce choice rather than add complexity.

Attribute provenance (must be visible)

Every field needs a provenance marker:

- ✓ verified (HR, KYC, liveness, OAuth, domain proof)
- • self-asserted (user-entered)
- ☁ ~~t~~emporarily unavailable (offline/pointer stale)

This makes the system citeable: it defines the meaning of "trust" at a field-by-field level.

See: Unified ID Card: Attribute Provenance

Security controls (release)

- **Step-up before share:** require biometric/PIN for sensitive releases.
- **Anti-replay:** server enforces single-use and marks artifacts as consumed.
- **Origin binding:** deep links + QR payloads are signed; apps validate environment.
- **Rate limiting:** wrong-code attempts limited; abusive IP/device throttled.
- **Audit trail:** emit a structured event per stage. Minimum schema:

```
events:       trustcheck.started | trustcheck.joined | trustcheck.shared |
              trustcheck.completed | trustcheck.expired | trustcheck.denied
fields:       eventId, tenantId, sessionId, verifierSuid, presenterSuid,
              deviceIds, method (qr | typecode | link), attributeSetHash,
              ttlExpiry, outcome, timestamp
never logged: attribute values, biometric data, raw QR payloads
retention:    your tenant audit policy; events reference attributes by hash,
              so log retention never extends attribute exposure
```

APIs (illustrative)

> Endpoints are illustrative; keep the contract stable even if the paths change.

- `POST /people/session/start` (verifier) → returns `sessionId` + `code` + `qr` + `link`
- `POST /people/session/join` (presenter) → validates token, binds presenter, returns verifier display info
- `POST /people/session/share` (presenter) → submits selected attributes + signed step-up proof
- `GET /people/history` (either) → recent People events
- `POST /people/contacts/save` (verifier) → saves a consented snapshot

- `POST /people/session/report` (either) → abuse/fraud report

Telemetry and KPIs

Publish these metrics. They give security, CX, and procurement a basis for trade-off conversations:

- completion rate by method (QR / code / link)
- median and p95 time-to-verify for real-time sessions
- timeout and cancel rates
- step-up shown rate and step-up fail/cancel rate
- contact-save conversion after successful verification

(See: Metrics + ROI Playbook)

Enterprise admin controls (recommended)

- tenant defaults:
  - enterprise → Work requirement ON by default
  - consumer → no requirement by default
- disable Anonymous tile (policy toggle)
- lock mandatory fields (compliance checker)

Example policy config (YAML)

Use policy-as-code to make People deployments reviewable and repeatable:

```
people:
  defaults:
    required_profile_level: "WORK"
    mandatory_fields: ["legalName", "companyName", "title"]
    ttl_seconds_realtime: 60
    ttl_seconds_link: 86400
    require_stepup_for_sensitive_fields: true
  overrides:
    - match:
        verifier_group: "field_ops"
      policy:
        required_profile_level: "MINIMAL"
        mandatory_fields: ["legalName", "companyName"]
    - match:
        workflow: "school_pickup"
      policy:
        required_profile_level: "MINIMAL"
        mandatory_fields: ["legalName", "photo"]
```

Rollout playbook

1. pick one workflow (e.g., technician arrival)
2. define a default requirement (Work) and mandatory fields
3. publish a 1-page script for verifiers
4. monitor completion rate + time-to-verify and iterate on prompts

Key Takeaway

People Trust Checks have three initiation modes (QR scan, typed code, messaging link) and three profile tiers (Work, Minimal, Anonymous). Non-negotiables include explicit consent before sharing, single-use verification tokens, short TTLs, and audit trails. Messaging links are session join artifacts, completion requires the ScrambleID app context.

FAQ

What are the non-negotiables for People Trust Checks?

Verifier-initiated sessions, explicit consent, step-up before release (when required), data minimization defaults, and single-use/TTL replay resistance.

When should I use short codes vs QR?

Use QR in-person; use short codes for voice/video or constrained cameras; use links for remote initiation.

Can the verifier force the presenter to overshare?

No. Requirements ensure a minimum, but the presenter retains control over optional fields, and sensitive fields should default OFF.

Are messaging links just OTP links?

No. Links are just session join artifacts; completion requires the ScrambleID app context and (optionally) step-up.

Can someone forward the link?

They can, but a forwarded link is mitigated by single-use + TTL + binding. A forwarded link should not produce a fresh verified share.

Do I need iMessage support to ship People?

No. In-app Trust Checks (QR/code/link) ship independently.

References (public)

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

Related reading

- People Trust Checks: Overview
- Unified ID Card Specification
- ID Card Picker (Consent UX)