Circle of Trust: Verified Coworkers, Verified Brands, and Trust Context at Decision Time

Trust & Risk / Last updated 2026-06-11 / https://www.scrambleid.com/learn/scrambleid-circle-of-trust

> **Status (June 2026):** In development. Circle of Trust is on the ScrambleID roadmap and isn't shipping yet. This article describes the design we're building toward, published now so architects evaluating the model can plan around it. If a near-term deployment depends on Circle of Trust behavior, talk to your ScrambleID account team about current timelines before committing.

In one sentence: Circle of Trust (CoT) is a **trust-graph service** that answers *"Is this person/channel/entity trusted?"* so ScrambleID products can display **verified cues** (coworker tiers, verified brands, personal trust) at the moment of interaction, **without** confusing context for authentication.

TL;DR (canonical)

- CoT does **not** grant access. It provides **context** that other policies can use.
- CoT returns **trust labels** with strict semantics: *unknown stays unknown; verified requires exact match.*
- CoT makes trust **queryable in <150ms (p95 cache-hit goal)** so it can be used in live flows (web, IVR, People Trust Checks).
- CoT is designed to reduce social engineering and misattribution, especially in channels where "who is this?" is ambiguous.

The problem CoT solves

Security teams often maintain trust lists (executive phone numbers, vendor domains, internal directories). But when the moment of decision arrives, a high-stakes call, a link, an in-person handoff, those lists are usually:

- **not available** in the UI,
- **not normalized** across products,
- **not enforceable** in policy,

- **not auditable**.

CoT solves this by turning "who/what do we trust?" into a **low-latency API** and an **event stream**.

What CoT is and is not

CoT is: a trust context plane.

CoT is not:

- an authentication method,
- an access-control system,
- a replacement for cryptographic proof.

If you need a mental model: CoT is a **signal provider**. Products still require proof via phishing-resistant rails (see Phishing-Resistant Web Authentication and Caller Authentication).

Trust signals CoT can return

CoT is intentionally simple. It returns a small set of **high-confidence labels**:

Signal	Meaning	Where it comes from	Safe UI cue	Policy usage
Enterprise tier: Trusted	Person is in your tenant's curated trusted cohort	Directory/HR sync + admin overrides	"Trusted coworker"	Reduce friction or enable targeted policies
Enterprise tier: Authoritative	High-importance cohort (e.g., manager, finance, exec)	Directory/HR sync + overrides	"Authoritative"	Require step-up / co-approval for sensitive actions
Verified brand	Channel belongs to a verified brand entity (exact match)	Brand registry (vetted domains/numbers)	"Verified brand"	Increase scrutiny on lookalikes; faster routing
Personal trust edge	Mutual personal trust established in-app	Invite + accept	"Trusted contact"	Improve intent cues; simplify future interactions
Possible spoof *(optional)*	Looks similar to a verified brand, but is **not** an exact match	Pattern/heuristic layer	"Possible spoof"	Drive warnings; never auto-trust

Key safety invariant

CoT must never produce "verified" labels from fuzzy matching.

- **Exact match** may produce "Verified brand". Exact means exact after normalization: E.164 for phone numbers, lowercased registrable domain for web origins, and no subdomain credit (mail.example.com does not inherit example.com verification).

- **Near match** may (optionally) produce "Possible spoof".
- Otherwise: **unknown**.

Where CoT fits in ScrambleID

CoT is valuable anywhere humans make trust decisions. These are the integration points the design targets:

- **Caller (voice):** show brand/coworker trust cues for inbound calls or during verification.
- **Online (web):** show "Trusted"/"Authoritative" context for sensitive requests.
- **ScrambleID People:** show enterprise-tier cues during a Trust Check initiation (pre-proof cue; never a bypass).
- **Desktop:** show trust cues for shared-workstation user switches.

For cross-channel monitoring, the design also exports CoT labels into Overwatch (also in development) as part of the investigative timeline.

End-to-end example: CoT + cryptographic proof (Caller → App)

CoT is most valuable when it **sets expectation**, and proof **finishes the job**.

```
sequenceDiagram
    participant U as User
    participant IVR as IVR / Agent Console
    participant CoT as Circle of Trust
    participant S as ScrambleID (Auth)
    participant OW as Overwatch

    IVR-->>CoT: evaluate(subject=+1415...)
    CoT-->>IVR: verified_brand=Acme Bank (or unknown)
    IVR-->>U: Read DID (one-time code)
    U-->>S: Confirm DID in app (bound device)
    S-->>IVR: Verified=TRUE (session-bound)
    S-->>OW: Emit events + trust labels
```

Rule: treat CoT labels as **pre-proof cues**. The actual state transition to Verified requires session-bound proof.

How CoT works

Inputs

1. **Enterprise tiers** (Trusted/Authoritative)

- Built from directory/HR ingest
- Admin overrides + audit

2. **Verified brands**

- Canonical brand entities
- Vetted channel set (E.164 numbers, domains)
- Periodic sync + cache

3. **Personal trust edges**

- User invites and acceptance
- Revocation

Consumption

A consuming product calls an evaluate endpoint with a **subject**, such as:

- a phone number (E.164)
- a domain
- a ScrambleID identity handle (SUID)

CoT returns labels and a TTL suitable for caching.

Evaluate API (example contract)

> The exact endpoint names may vary by deployment; this illustrates the stable contract.

Request

```
POST /circle/v1/evaluate
{
  "tenantId": "t_123",
  "subject": {
    "type": "phone_e164",
    "value": "+14155551234"
  },
  "context": {
    "channel": "caller",
    "appId": "cc_ivr",
    "locale": "en-US"
  }
}
```

Response

```
{
  "result": {
    "enterpriseTier": "authoritative",
    "verifiedBrand": {
      "brandId": "brand_acme",
      "displayName": "Acme Bank",
      "match": "exact"
    },
    "personalTrust": "none",
    "flags": ["finance_role"],
    "ttlSeconds": 86400
  }
}
```

Privacy and abuse prevention (non-negotiable)

CoT is only useful if it doesn't become an enumeration oracle.

Recommended guardrails:

- **Tenant scoping:** enterprise tier results must be tenant-bound.
- **Response minimization:** return "unknown" unless a strong match exists.
- **Rate limiting:** per tenant + per subject to prevent probing.
- **Audit:** every evaluate call is auditable (who queried what, when, and why).

- **UI safety:** cues must be informational; no "green checks" that imply proof.

How CoT complements call authentication standards

Caller ID spoofing is a known problem. Telecom frameworks such as **STIR/SHAKEN** help providers authenticate caller ID at the network layer, producing one of three attestation levels per call:

- **Attestation A (Full):** originating carrier authenticated the caller and verified their right to use the number.
- **Attestation B (Partial):** caller authenticated but right-to-use not verified.
- **Attestation C (Gateway):** the call entered the network through a gateway and could not be authenticated at origin.

STIR/SHAKEN attestation is useful as one input to a trust signal, but it does not answer "who is calling?" for the end-user experience. A call with attestation A still does not tell the recipient whether the human on the line is authorized to act on a specific account. Caller ID can be perfectly authenticated and the caller can still be an attacker who has the device.

CoT operates at a different layer. It is not a telecom protocol; it is an **identity-layer trust context** that products can display and apply in policy. CoT signals (enterprise tier, verified brand, personal trust edge) can be combined with STIR/SHAKEN attestation as inputs to a layered decision: STIR/SHAKEN tells you whether the network vouches for the number, CoT tells you whether the identity behind the number is recognized in your trust graph, and cryptographic proof from a ScrambleID flow tells you whether the person can complete a session-bound challenge with their enrolled device. Each layer answers a different question.

Implementation checklist

- [ ] Define enterprise tier rules (Trusted, Authoritative) and data sources (directory/HR).
- [ ] Configure admin override workflows and audit.
- [ ] Seed Verified Brands for your top support/sales/security numbers and domains.
- [ ] Decide whether to enable "possible spoof" hints (and keep the UI careful).
- [ ] Plan to integrate CoT evaluate into Caller/Online/People surfaces.
- [ ] Plan to emit CoT labels to Overwatch for cross-channel timelines.

Key Takeaway

Circle of Trust is a trust-graph service that answers "is this party/channel trusted?" using enterprise directory tiers, verified brand registrations, and personal trust edges. It enables context-aware

decisions like labeling incoming calls as verified, gating sensitive actions based on requester trust level, and preventing social engineering by surfacing trust signals at decision time.

FAQ

Does Circle of Trust grant access?

No. CoT is **context**. Access is still gated by cryptographic authentication (e.g., WebAuthn, QR(DID) confirmation).

Can CoT create false confidence?

It can if you design cues poorly. Use conservative language ("Verified brand" only on exact match, "Unknown" otherwise), and never imply authentication.

What's the difference between CoT and an address book?

An address book is personal and UI-only. CoT is **policyable**, tenant-scoped, and **auditable**, with strict semantics for verified tiers and brands.

What about lookalike domains or similar phone numbers?

Treat them as **not verified**. If you want, emit a separate "possible spoof" warning, but never elevate to "verified".

Does CoT store personal data?

It should minimize data and store only what it needs to return trust labels. Personal trust edges are opt-in and revocable.

Can CoT be used across enterprises?

The safe default is tenant-bound enterprise tiers; nothing crosses organizations unless you design it to. The cross-org model the design supports is explicit federation: each organization vouches only for its own people through its own enterprise tier, a brand entity is vouched for by the organization that verified it, and a relying tenant chooses which external tiers it accepts. The failure mode to design against is transitive trust: if org A trusts org B and B vouching is weak, A inherits B weakest link, so cross-org edges should carry the issuing tier assurance level rather than a flat trusted bit.

How fast does CoT need to be?

Fast enough to be used in live experiences (web page state, IVR routing). A practical target is **<150ms p95 on cache-hit evaluations**.

How do we prove it works?

Export metrics: coverage (labeled vs unknown), correctness (spot-check), and outcomes (reduced escalations; improved completion of People Trust Checks).

References (public)

- FCC overview of caller ID authentication and STIR/SHAKEN: https://www.fcc.gov/call-authentication
- NIST SP 800-207 Zero Trust Architecture (policy decision point / context signals): https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf
- CISA guidance on social engineering and phishing: https://www.cisa.gov/news-events/news/avoiding-social-engineering-and-phishing-attacks

---

Related reading

- Overwatch: Unified Identity Risk Monitoring
- Caller Authentication: Replace KBA and Stop Vishing
- People Trust Checks: People Verification With Consent
- ScrambleID Architecture: One Identity Rail Across Channels