

Bridging the Transparency Gap: Distributed Remote Attestation

Making selected attestation artefact reusable across verifiers and trust domains

source: [draft-wang-rats-distributed-remote-attestation-02](#)

DongHui Wang

Faye Liu

Yuning Jiang

Jun Zhang

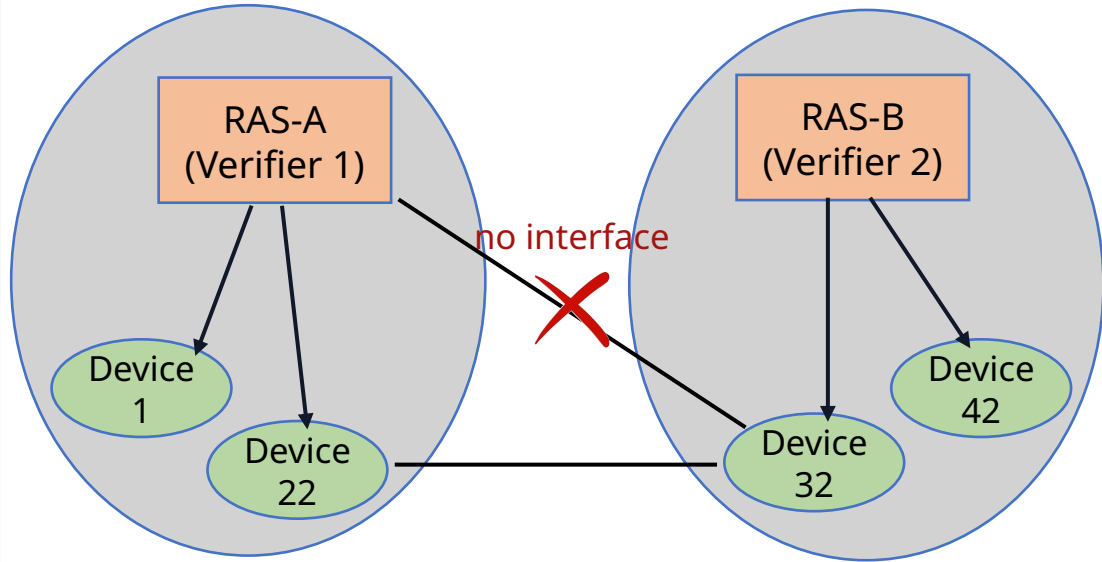
— Huawei

Why This Matters ?

Problem

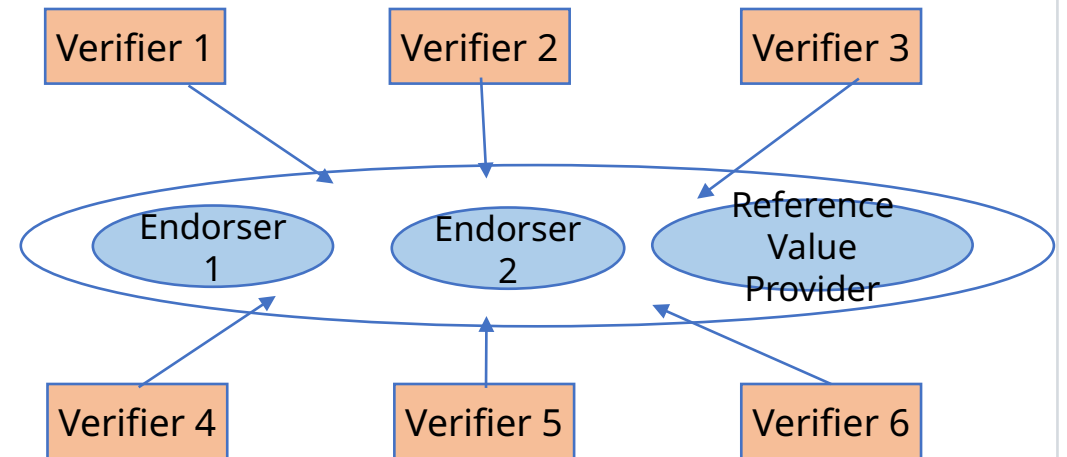
Cross-domain attestation transparency

A verifier or relying party in domain-A may need endorsement material, reference values, or attestation results that originate in domain-B. Direct point-to-point integration between operational sites does not scale and is often infeasible due to operational and policy constraints.



Many-to-many distribution and reuse of security artefact

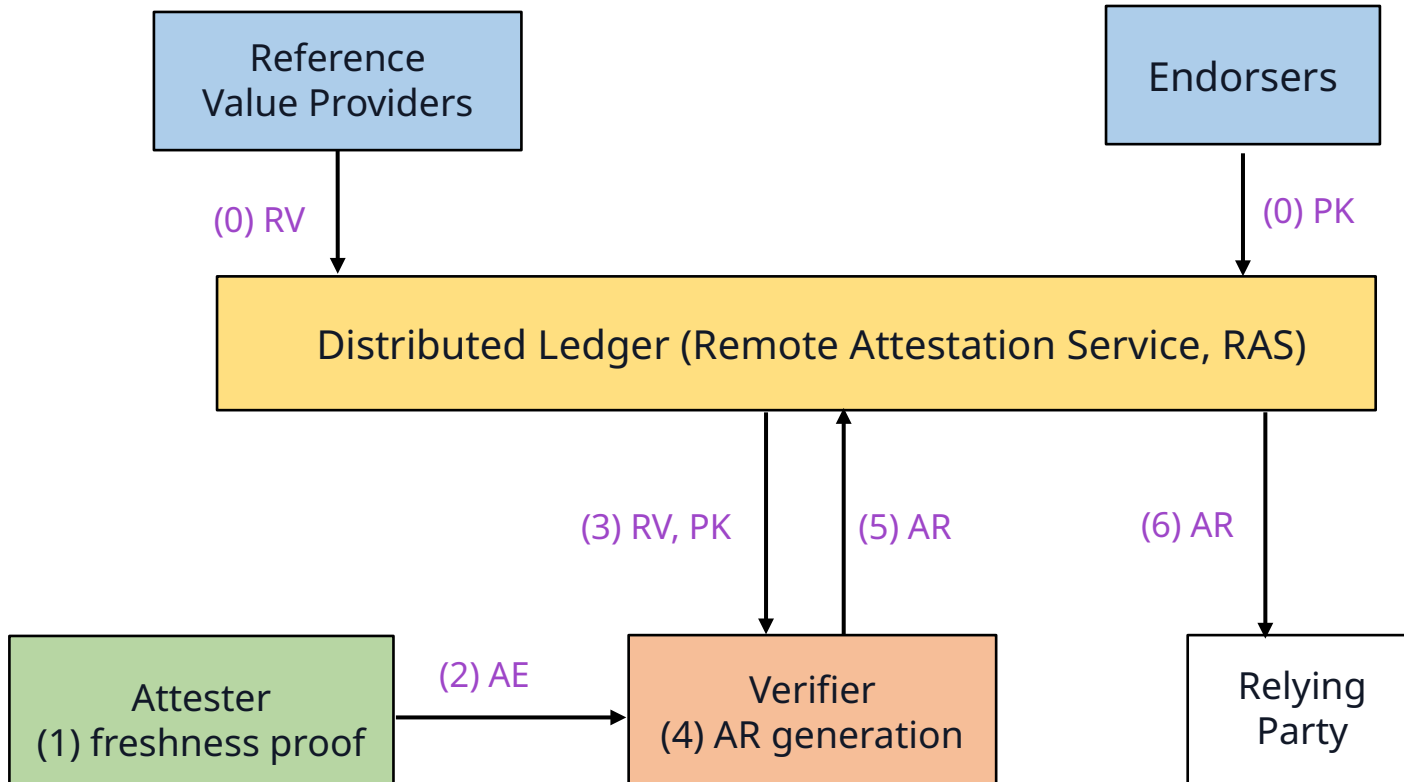
Verifiers need endorser public keys, endorsements, and reference values from multiple providers. Providers need to distribute artefact to multiple verifiers across domains. Similar many-to-many scaling issues apply when attestation results are shared for reuse.



Without a shared publication channel, each integration becomes a bespoke, brittle dependency.

Pattern 1: DL Publication with Off-Chain Attestation and Verification

Pattern A



Attestation evidence exchange and appraisal follow existing RATS flows, while the DL is used to publish and retrieve verifier inputs and verifier outputs.

(0) Publish inputs – RV providers and Endorsers publish RVs, PKs, and endorsements to the DL/RAS.

(1) Issue challenge – Verifier or authorised entity issues a freshness challenge.

(2) Send evidence – Attester generates AE with freshness proof and sends it to the Verifier (off-chain).

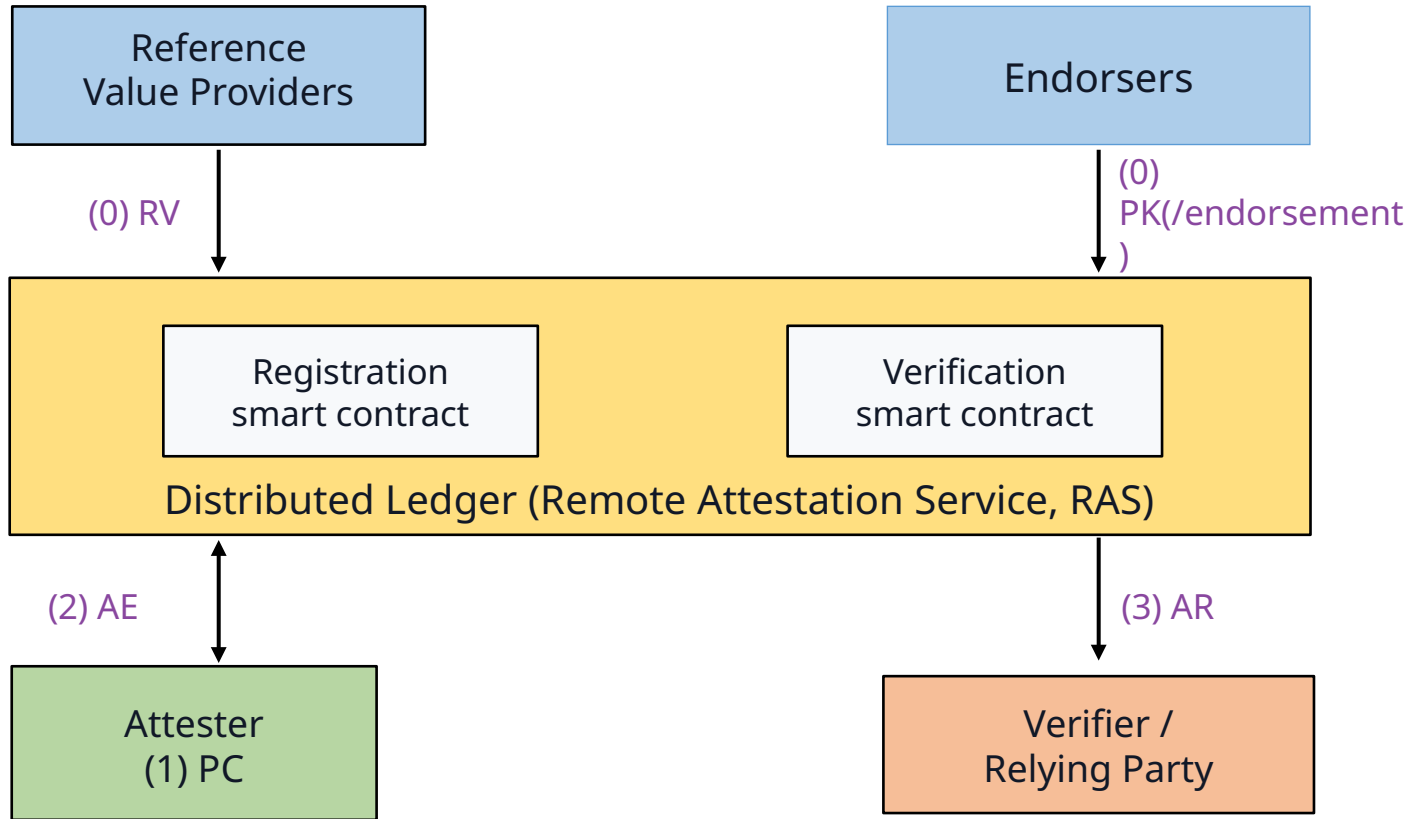
(3) Fetch inputs – Verifier retrieves RVs and Endorser artefacts from the DL/RAS.

(4) Appraise – Verifier evaluates AE and produces an Attestation Result (AR).

(5/6) Publish & reuse – AR (or its digest/pointer) is published to the DL/RAS for RP reuse under policy

Pattern 2: DL Publication with Optional On-Chain Verification

Pattern B



In this pattern, the DL is used for publication as in the previous pattern, and hosts verification logic (e.g., smart contracts) or record verifiers' appraisal outcomes.

- (0) Publish inputs - RVs and Endorser artefacts (PKs, endorsements) are registered on the DL/RAS via the Registration smart contract.
- (1) Publish evidence - Attester submits AE (with freshness proof and optional public challenge) to the DL/RAS.
- (2) Verify on-chain - Verification smart contract validates AE using RVs/endorsements.
- (3) Retrieve result - Verifier or RP queries the DL/RAS to obtain AR for decision or reuse.

Implementation: TLS + Hyper-ledger Fabric Remote Attestation

Demo

The entire remote attestation process consists of 4 steps: Attester builds evidence → RP forwards → Verifier verifies and writes to chain → RP queries aggregated result for decision.

tpm-tls13-fabric-ra-demo
Landscape flow console

One-click verify | End & shutdown components | Flow board | Live logs | Refresh logs

ATTESTER · RP · VERIFIER · FABRIC

TLS-bound attestation, one-screen view.

Built for demos: the full verification path stays visible in one landscape frame.

BINDING POINT
TLS 1.3 transcript + CertificateVerify

Artifacts + run locally

TLS CA | AK chain

ClientHelloRandom

Evidence / descriptor

Signed AR

Aggregated verdict

```
cd demo-web && python3 server.py
```

Open `http://localhost:8000` and trigger One-click verify. Use `LOG_DIR=./demo-web/logs` for log parsing.

End-to-end attestation flow

12 animated checkpoints compressed into a landscape matrix.

Run state: idle | Step 5 failed

RP LANE | ATTESTER LANE | VERIFIER LANE | FABRIC LANE

- 01 RP HANDSHAKE ClientHello**
RP opens TLS 1.3 and advertises the attestation-aware
- 02 ATTESTER TLS ServerHello + certs**
Attester replies with its TLS 1.3 handshake
- 03 ATTESTER OXFFA5 Attach evidence**
Evidence is carried in the private-use CertificateEntry
- 04 RP NONCE BIND Parse + bind nonce**
RP reads PeerAttestationData
- 05 RP VERIFYREQUEST Send to Verifier**
RP forwards evidence, descriptor data, and the
- 06 VERIFIER POLICY Validate evidence**
Verifier checks AK chain, PCR/eventlog, TCB, revocation, and
- 07 FABRIC OPS Record nonce**
Replay protection is enforced with CheckAndRecordNonceV2.
- 08 FABRIC COMMIT Commit AR + verdict**
Evidence commitment, AR
- 09 VERIFIER RETURN Return Signed AR**
Verifier sends back the signed on-path attestation result
- 10 RP CROSS-CHECK Verify on ledger**
RP re-checks signatures, commitment hash, nonce hash, verifier id, and evidence id
- 11 RP DECISION Send decision**
Allowed / denied and evidence id are returned over TLS channel trust anchors, participant policies
- 12 ATTESTER GATE Gate app traffic**
Post-handshake data is released only after the attestation decision

channel-private-rv: reference values and rollout profiles

channel-ops: nonce records, commitments, AR set, aggregated verdict

RecordEvidenceCommitmentV3(rp_id, attester_id, nonce_hash, evidence_digest, uri, format)

SubmitARV3(evidence_id, verifier_id, signed_ar, policy_digest, trusted)

TryAggregateWithPolicy[evidence_id] -> AggregatedVerdictRecord

LIVE RUN | LAST DECISION

No logs loaded

evidence_id: -
time: -

SOURCE: ./demo-web/logs | MODE: On-chain aggregation

Live stream preview

Canvas capture feeds a compact video monitor.

Start stream | Stop

Streaming replay... step 5/12

VIDEO PREVIEW

Attestation Live Stream

- 01 ClientHello
- 02 ServerHello + certs
- 03 Attach evidence
- 04 Nonce-bind nonce

Live execution log

Local replay and backend SSE both land here.

```
[system] API connected
[system] event stream reconnecting...
[system] start requested (run_id=4)
```

logs/rp.log | logs/attester.log | logs/verifier.log

Continue the Conversation

Next

Feedback is more than welcome!

Whether the draft should stay DL-focused or be framed more generally as a shared tamper-evident publication channel; how to align DRA with existing RATS information models; and what trust, access-control, provenance, and freshness assumptions should be explicit.

Where to read more

[draft-wang-rats-distributed-remote-attestation](#)

The current draft already covers two DRA patterns and a discussion of freshness, access control, and provenance.

Thank you