QORTARA CLOUD GOVERNANCE

Trust Federation

```
# Trust Federation

Trust federation lets one organization evaluate whether another organization's agent meets an agreed governance posture, without either side exposing its private policy internals.

> **Availability.** The portable trust attestation lifecycle is shipped today: mint a Qortara-signed attestation for an agent, look it up across organizations by agent identifier, and verify it with Ed25519. The broader bilateral cross-org federation operator (the Trust Mesh, branded Qortara Trust Federation) is planned. Treat the cross-org coordination described below as the target design, not a generally available service.

## What is a trust attestation

A trust attestation is a signed, portable statement about an agent's governance history. Rather than asking another organization to trust your internal policies, you hand them a verifiable summary they can check on their own.

Each agent registered in Qortara carries a trust score: a value between 0.0 and 1.0 that reflects the agent's behavioral track record. The score is not assigned by hand. It is derived from observable governance events over time, and recent events weigh more heavily than old ones. Consistently good governance behavior raises the score; policy violations and verification mismatches lower it.

The exact weighting is an internal detail and is intentionally not published. What matters for an integrator is the principle: the score summarizes history, and the attestation makes that summary portable and verifiable.

## Signals that move the score

| Signal | Direction | Example |
| --- | --- | --- |
| Policy evaluation allowed | Small positive | The agent requested access and was permitted. |
| Policy denied, then complied | Neutral to small positive | The agent was denied and did not try to work around it. |
| Compliance scan passed | Moderate positive | The agent's tenant passed a compliance scan. |
```

Saga verified	Positive	The agent completed an action-outcome cycle correctly.
Saga mismatch	Negative	The agent's action produced an unexpected outcome.
Rate limit exceeded	Negative	The agent hit rate limits, suggesting aggressive behavior.
Repeated denied pattern	Strongly negative	The agent repeatedly attempts actions it is not authorized for.

Where trust scores are used

1. **Policy conditions.** A policy can require a minimum trust score, for example allowing cross-tenant data sharing only when the requesting agent's score is above a threshold.
2. **Compliance evidence.** Trust history can be included in compliance reporting as evidence of behavioral governance over time.
3. **Cross-organization lookup.** When one organization's agent wants to interact with another's resources, the second organization can look up the agent's attestation to decide whether to allow the interaction.

Reading a trust attestation

The attestation lifecycle is the part that exists today. A cross-org lookup retrieves a signed attestation for an agent by its identifier.

```bash
curl https://api.qortara.com/v1/trust/federation/did:qor:agent:abc123 \
  -H "$AUTH_HEADER"
```

```json
{
  "attestation": {
    "agent_did": "did:qor:agent:abc123",
    "tenant_id": "qor:org_a",
    "trust_score": 0.87,
    "signal_count": 142,
    "issued_at": "2026-04-10T12:00:00Z",
    "valid_until": "2026-04-10T13:00:00Z",
    "issuer": "qortara.com"
  },
  "signature": "base64-encoded-ed25519-signature",
  "merkle_proof": {
    "root_hash": "abc123...",
    "audit_chain_length": 142
  }
}
```

The attestation is Ed25519-signed by Qortara. The receiving organization can verify the signature independently, so it does not have to trust the issuing organization directly; it only has to trust the signature and the verification math. The `merkle_proof` ties the attestation to a tamper-evident audit chain, so a modified history would fail verification.

The lifecycle has three operations:

- **Attest:** mint a Qortara-signed attestation for one of your own agents.
- **Look up:** retrieve another organization's agent attestation by identifier (this is the metered cross-org act).
- **Verify:** check an attestation's Ed25519 signature, either inline or against a stored copy.

The planned Trust Mesh

The cross-organization coordination below is the target design for Qortara Trust Federation. It is planned, not present-tense. The attestation lifecycle above is what exists today; the Mesh adds the bilateral operator and protocol that automate exchange between organizations.

The problem the Mesh is designed to solve

When one organization's agent needs to reach another organization's resources, the second organization has no behavioral history for that agent. Without portable trust it faces two poor options: reject all cross-org interactions (safe but kills collaboration) or allow them all (functional but insecure). Neither is acceptable at scale.

The intended flow

Qortara is designed to act as a neutral attestation authority that both organizations already trust, so neither has to expose internal policy to the other.

```text
Org A's Agent        Qortara                  Org B's System
     |                  |                          |
     |-- request to     |                          |
     |   access Org B -->|                         |
     |                  |<-- Org B looks up the     |
     |                  |    agent's attestation    |
     |                  |--- returns signed         |
     |                  |    attestation, score 0.87|
     |                  |                          |
     |                  |    Org B checks 0.87 against |
     |                  |    its own threshold (0.7)   |
     |                  |    --> allow                 |
```

```
     |<-----------------------------------------------|
     |   access granted |                             |
```

In this design Org B never sees Org A's policies. It sees a signed attestation, verifies it, and applies its own threshold. The decision stays local to Org B even though the evidence came from elsewhere.

Why an organization would not share policy internals

Policy internals often encode sensitive operational detail: which capabilities map to which roles, what time windows apply, which resources are considered high-risk. An attestation lets a partner make a sound trust decision without learning any of that. The partner gets a verifiable summary and a signature; it does not get your rulebook.

Limits and caveats

- The attestation lifecycle (attest, look up, verify) is shipped. The bilateral Trust Mesh operator is planned; do not build as if the full mesh exists today.
- A trust score is a summary of history, not a guarantee of future behavior. Use it as one input to an authorization decision, alongside your own policies.
- An attestation has a validity window (`valid_until`). Treat an expired attestation as stale and re-look-up rather than caching indefinitely.
- Verification proves the attestation was issued by Qortara and not altered. It does not, on its own, prove the receiving organization's policy was satisfied; that decision remains yours.

Related concepts

- [Policy enforcement](/docs/concepts/policy-enforcement) can require a minimum trust score as a condition.
- [Saga verification](/docs/concepts/saga-verification) produces signals that move the score.
- [Compliance evidence](/docs/concepts/compliance-evidence) can include trust history.

Product: Qortara Cloud Governance
Source owner: Qortara
Last reviewed: 2026-06-10
Verified against: Qortara pre-launch docs