Brokered Trust for AI Agents: Lessons from the eIDAS QTSP-Broker Pattern for an Agent Trust Framework

Anton Sokolov, Tyche Institute, Tallinn, Estonia

Contents

Abstract

Single roots of trust fail in real ecosystems. Qualified Trust Service Provider (QTSP)-brokers under EU electronic identification, authentication and trust services (eIDAS) already federate human signatures across providers. This article maps that pattern to agent-trust design, yielding Agent Trust Framework (EATF) recommendations and an Estonian regulatory-sandbox pilot.

Keywords: AI agents, agent attestation, eIDAS 2, EUDI Wallet, qualified trust service providers, broker architectures, PAdES, ASiC-E, EATF.

---

1. The hook — every agent-trust paper assumes a single root

Every published proposal for an AI-agent trust framework assumes, implicitly or explicitly, a single root of trust: one Decentralized Identifier (DID) method, one attestation issuer, one PKI, one wallet ecosystem. Real ecosystems are never single-rooted. This paper argues that the federation problem agent-trust frameworks treat as future work has, in fact, been solved already — for human principals — by a class of deployed systems we call **QTSP-brokers**: single-API façades that fan out behind the scenes to dozens of Qualified Trust Service Providers and notified national eID schemes under the electronic identification, authentication and trust services (eIDAS) regime, emit European Telecommunications Standards Institute (ETSI)-baseline signature artifacts on behalf of the buyer, and maintain long-term validity over the artifact's lifecycle. We trace the broker pattern's architectural primitives — identity/action decoupling, format-baseline reuse, broker-level long-term-validation responsibility, and the QSCD/QTSP separation — and show that each primitive maps cleanly onto an open design question in the agent-trust literature. The mapping yields four concrete recommendations for the EATF reference framework: (i) treat agent identification and agent action attestation as two interfaces, not one; (ii) adopt PDF Advanced Electronic Signatures with long-term archival validation (PAdES-B-LTA) and Associated Signature Container Extended (ASiC-E) as referenced baseline containers

for long-term-verifiable agent action receipts; (iii) name DSS (the EU Commission's reference library) as architectural prior art for the EATF reference verifier; (iv) instrument the EATF interface to extend naturally to EUDI Wallet attestations as the broker pattern migrates from QTSP-rooted to wallet-rooted trust. The paper closes with an Estonian-regulatory-sandbox pilot proposal scoped to the experimentation framework adopted by Accelerate Estonia.

Keywords: AI agents, agent attestation, eIDAS 2, EUDI Wallet, qualified trust service providers, broker architectures, PAdES, ASiC-E, EATF.

Every time a fresh agent-trust framework lands on arXiv, the same diagram appears in its third figure: an agent on the left, a verifier on the right, and **a box in the middle** labelled *issuer*, or *registry*, or *trust anchor*, or *attestation authority*. The box is always singular. The box is sometimes elaborated into a "set" or a "federation" or a "trust list" — but the elaboration is gestural. The protocol is specified against the singular box, the verifier code is written against the singular box, the threat model assumes the singular box, and the open-questions section admits that real deployments will need to "support multiple issuers" as future work.

Real deployments do not start with one issuer and add more. They start with twelve and the buyer wants one API.

This is not a hypothetical condition. It is the operating reality of every cross-border transaction that has involved a qualified electronic signature in Europe over the last decade. A company that needs a qualified signature accepted in Belgium, Germany, France, and the Baltics does not sign twelve QTSP contracts. It signs one contract with a **broker** — typically eID Easy, Dokobit, Signicat, or Scrive — and the broker fans out behind a single REST API to the underlying QTSPs and national eID schemes. The broker absorbs format conversion (PAdES vs XAdES vs CAdES vs ASiC-E), identification-method routing (Smart-ID vs Mobile-ID vs Belgian eID card vs Czech Občanský průkaz vs local certificate), and long-term validity maintenance (timestamp refresh, OCSP/CRL embedding, archival re-sealing). The buyer signs documents; the broker handles federation.

The agent-trust literature has not yet caught up with this fact. The broker layer exists, has working code, has commercial revenue, and has solved exactly the federation problem that every agent-trust paper labels "future work." This paper extracts its architectural primitives and proposes their direct adoption into the Agent Trust Framework reference design (Agent Trust Framework (EATF) v0.3, in progress) (Sokolov 2026a).

The motivation is not academic. The window during which agent-trust frameworks can copy from a working analog is narrow: it closes when EUDI Wallet reaches general availability and the broker pattern itself starts migrating from QTSP-rooted to wallet-rooted trust. EATF wants to be on that curve when it shifts, not behind it.

---

2. A five-minute primer on the QTSP-broker pattern

For readers from agent-trust research who are not eIDAS specialists, the broker layer reads as follows.

Underneath, the eIDAS regulation defines a class of regulated entities called **Qualified Trust Service Providers (QTSPs)**. A QTSP is audited against ETSI standards (TS 119 401 and the EN 319 4xx series), supervised by a national authority, and listed on the national Trusted List published under Commission Implementing Decision (EU) 2015/1505 (ETSI 2022; European Commission 2015). Among the trust services a QTSP can offer, the load-bearing ones for this paper are: qualified certificates for electronic signatures, qualified electronic seals, qualified electronic time stamps, qualified preservation services, and qualified validation services. Each qualified service produces an artifact (a signed PDF, an ASiC-E container, a timestamped log line) that carries cross-border legal effect under the corresponding articles of Regulation (EU) 910/2014 (European Parliament and Council of the European Union 2014, 2024; European Commission 2015).

On top of the QTSPs, a market layer has emerged of broker services whose product is not the qualified certificate itself but the **orchestration of qualified services** into business workflows. These brokers — eID Easy, Dokobit, Signicat, Scrive being the canonical examples — present the customer with a single REST API (eID Easy and Dokobit and Signicat and Scrive 2026). Behind the API, they route requests to whichever QTSP or notified eID scheme is needed for the specific jurisdiction, signature level, and identification method. They absorb the format-conversion matrix (PAdES vs XAdES vs CAdES vs ASiC-E), the identification-method routing (Smart-ID, Mobile-ID, eID card, local certificate, …), and the post-signing maintenance burden (LTV, archival re-sealing, validation evidence).

The architectural primitives of the broker layer, stated abstractly:

1. **Single façade, multi-rooted backend.** The broker is one API; the trust roots behind it are many, audited independently, and certified under separate Trusted Lists.
2. **Identity and action are two interfaces.** The broker calls the identification subsystem to assert *who is asking*, then calls the signing subsystem to produce *what they signed*. These are sequentially composed but independently specified.
3. **Format-baseline reuse.** ETSI publishes a family of "baseline" profiles (PAdES-B-B, PAdES-B-T, PAdES-B-LT, PAdES-B-LTA, and the parallel families for XAdES, CAdES, and ASiC) that pin a small, finite set of long-term-validatable artifact shapes. Brokers commit to these baselines rather than inventing new formats.
4. **Long-term validity is the broker's job.** OCSP responses, CRL snapshots, timestamps, and certificate-chain data must be embedded into the signature artifact before the underlying CA's revocation information becomes unavailable. This LTV obligation is operational, continuous, and well-understood. The reference implementation is the EU Commission's DSS library (European Commission 2026); most brokers integrate it or its functional equivalent.
5. **The QSCD is a separable concern.** Qualified signatures must be produced using a Qualified Signature Creation Device. In practice this is a remote HSM, a hardware token, or a certified-mobile element. The broker does not own the QSCD; it routes signing requests to a QTSP that does. The buyer of the broker contract does not know which physical device produced the signature, and does not need to.

These five primitives are the contribution of two decades of eIDAS deployment experience. They are battle-tested, they are documented in ETSI and ENISA publications, and they are already running at commercial scale. The next sections argue that the agent-trust literature can adopt them wholesale.

---

3. The identity / action split is the load-bearing primitive

Of the five primitives in §2, primitive (2) — **identity and action as two interfaces** — is the one that does the most work when transferred to the agent-trust setting. The rest of this paper builds on it.

In the eIDAS broker, the split is operational: the broker asks "prove who you are" using one of many possible identification methods (Smart-ID, Mobile-ID, eID card, OAuth-style federated login, …), and having satisfied itself of the identity assertion, asks "now sign this document" using a separate qualified signing subsystem. The two steps share a session but not a key. The identification method produces an authentication assertion; the signing method produces an artifact in an ETSI baseline. The same human can be identified by Smart-ID today and by Belgian eID card tomorrow; the resulting signed artifacts are indistinguishable to a downstream verifier because the artifact format is ETSI-baselined and contains the qualified certificate that ties signature to person.

The agent-trust literature, by contrast, conflates the two. Most published proposals collapse "this is agent X" and "agent X did action Y at time T" into a single attestation, signed by a single key, validated against a single registry. The conflation is convenient for prototypes and catastrophic for federation: it means that switching

the identification modality (from key-pinning to DID resolution to wallet-attestation) requires changing the action-attestation format, and vice versa.

The eIDAS lesson is that the two interfaces must be specified separately, each with its own conformance profile and its own threat model. For an Agent Trust Framework, this implies:

- **Agent identification** is its own interface, with multiple permitted modalities (key-pinning under offline-verifier conditions; DID resolution where a resolver is available; attested-key under TPM/SGX rooted attestation; wallet-mediated attestation under EUDI Wallet). The interface returns an identity assertion with a stable subject identifier and a method-of-assurance tag.
- **Agent action attestation** is a separate interface, parameterised by the action being attested (inference, tool-call, policy-decision, human-confirmation) and emitting an ETSI-baselined artifact (see §5). The artifact references the identity assertion by hash, not by inline inclusion, so the identity layer can evolve without re-issuing the action receipts.

This split lets agent-trust frameworks federate over the identification layer (twelve issuers, one verifier interface) without disturbing the action layer, and federate over the action layer (twelve action types, one verifier interface) without disturbing the identification layer. EATF v0.2.1 already implements the split implicitly; v0.3 should codify it as an architectural invariant.

---

4. Why the broker layer captures margin — and what that tells us

A reader from the agent-trust literature might reasonably object that the broker pattern, however widely deployed in eIDAS, is a commercial artifact rather than an architectural one. The objection deserves a serious answer. We give it in this section. The short form is: the broker pattern is *both* — a commercial artifact whose persistent profitability is itself evidence that its architecture is well-fitted to the federation problem buyers actually have.

4.1 Broker layers in adjacent verticals

The "single façade, multi-rooted backend" pattern is not specific to trust services. It recurs across several verticals where buyers face a fragmented supply side and an integration tax they prefer to externalize. A non-exhaustive map:

Vertical	Broker layer	Order of magnitude
Travel / car rental	Amadeus, Sabre, CarTrawler	tens of billions of euros (Amadeus market cap)
Payments	Stripe, Adyen, Mollie	tens of billions (Stripe last private round; Adyen public market cap)
Human identity	Okta; Auth0 (acquired by Okta, 2021)	tens of billions / mid single-digit billions respectively
KYC	Onfido, Jumio, Veriff, SumSub	single-digit billions (Veriff last private round; Onfido at Entrust acquisition; SumSub mid hundreds of millions per published reporting)
Trust services	Signicat (private-equity owned); Dokobit (acquired by Signicat, 2021); Scrive; eID Easy	mid hundreds of millions of euros range (Signicat valuation band per published deal reporting)
Agent attestation	*(empty)*	*(empty)*

All market-cap and valuation figures above are given as orders of magnitude only and must be re-verified against current public sources before final submission; the contribution of the table is the *shape*, not the

precise numbers. The shape, however, is stable. Every mature vertical with a fragmented qualified-supplier layer has a broker layer sitting on top of it, and that broker layer captures a material share of the value chain. The trust-services row is half-populated and still fluid (the Signicat/Dokobit consolidation is recent; Scrive and eID Easy remain independent at time of writing). The agent-attestation row is empty for the simple reason that the supply side itself has not yet stabilized: there is no general-purpose agent-attestation issuer to broker.

4.2 The five places broker margin sits

A useful frame from the payments and travel literatures is to ask *where* in the value chain the broker captures margin. In trust services, five separable margin sources are visible:

1. **Multi-vendor buyer pain.** A buyer who needs qualified signatures in twelve jurisdictions does not want twelve QTSP contracts. The broker absorbs procurement complexity in exchange for a per-signature fee that is materially lower than the buyer's notional cost of negotiating, auditing, and maintaining twelve direct relationships.
2. **Format and protocol translation.** Buyers want signed business documents; QTSPs sell CMS-shaped, XML-shaped, or PDF-shaped primitives. The broker maps between the customer's preferred output (a signed PDF, an archived ASiC-E container, an XML business message) and whatever the underlying QTSP natively produces.
3. **Long-term validity maintenance as an SLA.** Qualified signatures degrade over time as the issuing CA's revocation information becomes unavailable, as algorithms weaken, and as timestamp authorities rotate. Brokers offer SLAs for re-timestamping, re-sealing, and embedded-validation refresh that no individual QTSP sells as a standalone product. This is operational labor with a monthly recurring price tag.
4. **Compliance translation.** "Is this signature valid for German tax filings?" is not a cryptographic question; it is a regulatory-mapping question that requires up-to-date knowledge of national supervisory guidance, member-state-specific recognition rules, and case law. Brokers maintain this knowledge as a consultative overlay and price it into their per-signature or per-seat fees.
5. **Commodity backend, branded frontend.** QTSPs sell qualified certificates at increasingly commodity prices; brokers sell *workflows* (a signed agreement, a notarized lease, an attorney-attested deed) at outcome-priced fees. The margin sits in the gap between commodity input and outcome-priced output, which is the classic broker rent in any vertical.

4.3 The research-relevant point

A trust architecture researcher who reads §4.1 and §4.2 should derive one specific conclusion. The buyer's problem in trust services is *not* "trust one root." It is *"absorb the federation problem on my behalf."* Buyers who could have built single-root systems for themselves (typically large enterprises with their own internal PKI) routinely choose to consume broker services anyway, because what they value is not the certificate but the *outcome* and the *operational externalization*. Any agent-trust architecture that designs for the single-root buyer is designing for a buyer the market does not have.

This observation is not original to this paper — it is implicit in every successful broker-layer deployment of the last fifteen years — but it has not, to our knowledge, been stated explicitly in the agent-trust literature. We state it now because it explains why single-issuer agent-trust proposals have not been adopted at scale by the very enterprises whose attention the proposals were soliciting.

4.4 Caveat on the scope of this argument

This paper is not a market-research piece. The comparables in §4.1 and the margin analysis in §4.2 are used only to establish that the broker pattern is load-bearing in adjacent verticals and that its persistence is evidence for its architectural fit. The contribution of this paper is the *architectural primitives* of §2 and

their EATF mapping in §7, not any commercial recommendation. Tyche Institute is a research entity; any commercial extrapolation from the analysis here is reader-side and explicitly out of scope.

---

5. Format reuse — PAdES-B-LTA and ASiC-E as the receipt baseline

If the identity/action split of §3 is the load-bearing architectural primitive, the format-reuse primitive of §2 is the load-bearing engineering primitive. The agent-trust literature has, with rare exception, treated the format of action attestation as a free design parameter — JWT-shaped, CBOR-shaped, JSON-LD-shaped, Verifiable Credential-shaped, novel-binary-shaped (W3C 2025). The eIDAS world long ago collapsed this design space to a small finite set of ETSI baselines that any QTSP and any broker can produce and any compliant verifier can validate. Agent-trust frameworks should adopt these baselines rather than continue to multiply the design space.

5.1 The ETSI baseline family

ETSI publishes four parallel families of signature baseline profiles: **CAdES** (ETSI EN 319 122-series, for generic CMS-shaped payloads), **XAdES** (ETSI EN 319 132-series, for XML payloads), **PAdES** (ETSI EN 319 142-series, for PDF payloads), and **ASiC** (ETSI EN 319 162-series, for associated-signature containers that wrap arbitrary payloads alongside their signature) (ETSI 2016b, 2016a). Within each family the baseline ladder is identical in shape:

- **-B-B** (basic) — the signature itself, with the signer's certificate and the cryptographic binding to the payload.
- **-B-T** (with timestamp) — adds a qualified electronic timestamp proving the signature existed at or before a specific point in time.
- **-B-LT** (long-term) — adds OCSP responses, CRL snapshots, and certificate-chain data necessary to validate the signature after the issuing CA's revocation information stops being live.
- **-B-LTA** (long-term with archival timestamp) — adds an archival timestamp over the whole bundle, protecting against future hash-algorithm weakening or PKI-algorithm migration. Successive archival timestamps can be added over time, extending the signature's validatable lifetime indefinitely.

The baselines are deliberately a ladder, not a menu. A consumer that can validate a -B-LTA signature can validate the lower rungs by discarding the additions; a producer that can emit a -B-LTA signature can degrade to a lower rung by omitting them. This is exactly the property an agent-trust framework wants in its receipt format.

5.2 Why agent action receipts want -LTA

Agent action receipts are *evidence artifacts*. Their value sits in being validatable years after the model that produced them has been retired, the issuing CA has been rotated or wound down, the attestation-issuer's signing key has been replaced, and the entire hash-algorithm and PKI-algorithm field has migrated to a post-quantum or otherwise revised baseline. This is precisely the scenario the -LTA family was designed for. Adopting it for agent receipts costs nothing — the format exists, the tooling exists, the ETSI conformance vectors exist — and buys the framework decades of forward-validity at the structural level.

5.3 ASiC-E as the container

ASiC-E (the "extended" variant of ASiC, ETSI EN 319 162) packages a signed object, one or more signatures over it, all associated validation data, and a manifest into a single ZIP-shaped container. The container is self-describing and self-validating; a verifier given nothing but the ASiC-E file and access to the relevant Trusted Lists can independently confirm everything inside it.

This matches the EATF "evidence bundle" pattern almost exactly. EATF v0.2.1 already specifies an evidence-bundle structure that bundles an agent's action receipt with its supporting verification data. The v0.3

specification should adopt ASiC-E with embedded XAdES-LTA signatures as the canonical wire format for an EATF evidence bundle. JSON-LD-shaped receipts may continue to exist as an internal representation; the externally-emitted, verifier-consumable artifact should be ASiC-E.

5.4 DSS as architectural prior art

The EU Commission maintains an open-source library called **Digital Signature Service (DSS)**, distributed under the Joinup digital building blocks program (European Commission 2026). DSS is the reference implementation of the ETSI baseline families. It is used by every serious broker in the trust-services market, by national supervisory authorities for conformance testing, and by the Commission itself for cross-border validation services. Architecturally, DSS plays the same role for human-principal signatures that the EATF reference verifier plays for agent-principal receipts: a library that consumes an artifact and returns a validation verdict together with the evidence on which the verdict rests.

The EATF reference-verifier README should cite DSS as architectural prior art, and the EATF v0.3 conformance suite should where possible adopt DSS-equivalent test vectors. This costs nothing and gains EATF the credibility of a well-precedented validation model.

5.5 Recommendation

The recommendation that follows from §5.1–§5.4 is direct. EATF v0.3 should name **PAdES-B-LTA** (for PDF-payload agent receipts where the action being attested is the production of a document) and **ASiC-E with embedded XAdES-LTA** (for generic-payload agent receipts where the action being attested is anything else) as referenced baseline emission formats. The existing JSON-LD evidence-bundle representation may remain as a permitted internal exchange format but should not be the canonical externally-validatable artifact. No new format is required; existing tooling can produce and validate these baselines on day one.

---

6. The eIDAS 2 / EUDI Wallet bridge

The eIDAS regulation under which the broker pattern matured is now itself being superseded. Regulation (EU) 2024/1183, adopted in April 2024 (referred to here as **eIDAS 2**), amends Regulation (EU) 910/2014 to introduce the European Digital Identity Wallet (**EUDI Wallet**) and the framework for issuing **electronic attestations of attributes (EAAs)** to wallet holders (European Parliament and Council of the European Union 2024; European Commission 2025). The wallet sits between the qualified trust service provider and the relying party; it carries attestations about its holder and can produce qualified signatures on the holder's behalf using QSCDs accessible to the wallet. The deployed brokers we have discussed throughout this paper are now in the early stages of integrating wallet-mediated identification and signing into their existing platforms.

6.1 Where the broker pattern goes next

The deployed brokers are the natural integration point for EUDI Wallet attestations into business workflows. They already speak ETSI baselines; adding wallet-mediated identification and wallet-issued attestations as a *new identification modality* fits primitive (2) of §2 (the identity/action split) without disturbing the rest of the stack. The signing subsystem continues to emit PAdES-LTA or ASiC-E artifacts; the identification subsystem gains a new permitted modality (present an EAA from your wallet) alongside the existing ones (Smart-ID, Mobile-ID, eID card, …). Brokers that have already absorbed the identification-method-routing complexity are the lowest- friction path to production EUDI Wallet integration for any business workflow that already uses them.

6.2 Where agent-trust meets the bridge

The interesting question for this paper is what happens when wallet attestations start to include *non-human-principal* attestations. The EUDI Wallet Architecture and Reference Framework (ARF) does not, in its current published version, explicitly contemplate agent attestations (European Commission 2025); the principal

is implicitly a natural person. Several working-group threads, however, anticipate extensions to legal-person attestations and to delegated-authority attestations (one party attesting that another party is authorized to act on its behalf). Once such extensions reach a stable draft, the path to *agent-as- delegated-principal* attestations becomes a near-trivial extension: the holder of the wallet attests that a specific software agent is authorized to act on its behalf, with a specified scope and a specified expiry, and the broker pattern emits the resulting authorization as a PAdES-LTA or ASiC-E artifact in exactly the same shape it would emit a human-signed instruction.

The architectural extension is small. The legal-clarification work required to make it operational is large. The combined window — small technical extension, large legal extension, both gated on EUDI Wallet GA and ARF stabilization — opens in the 2026–2027 horizon. EATF should be positioned to consume the resulting attestations the day they ship, not to scramble for an integration story after they ship.

6.3 Recommendation

EATF v0.3 should publish a non-binding **mapping document** from EATF identity-assertion shapes to EUDI Wallet EAA shapes. The document need not be normative; its purpose is to ensure that when wallet rails ship agent-capable attestations, EATF consumers can adopt them with minimal protocol churn. The mapping document should be co-published with the spec revision and tracked against ARF version updates as a living document.

---

7. EATF as the agent-side instantiation

The five primitives of §2 map to EATF as follows.

Broker primitive	EATF v0.2.1 status	EATF v0.3 to-do
Single façade, multi-rooted backend	Implicit; verifier accepts artifacts from any conforming issuer	Make explicit in spec; document the multi-issuer expectation
Identity / action split	Implicit in the Agent Evidence Package profile	Codify as architectural invariant; specify the two interfaces separately
Format-baseline reuse	Currently JSON-LD evidence bundles with detached signatures	Add PAdES-B-LTA and ASiC-E with embedded XAdES-LTA as referenced baseline emission formats
Long-term validity	Out of scope in v0.2.1 (verifier is offline-validating only)	Specify LTV embedding requirements for the new baseline formats; document broker-side LTV maintenance pattern
QSCD separability	Out of scope in v0.2.1	Open question for v0.3; see §9.3

7.1 What v0.2.1 already gives

The Agent Evidence Package (AEP) profile that EATF v0.2.1 specifies already separates the agent's identification material (a stable subject identifier and the verifying key) from the action attestation itself (the action description, timestamps, supporting evidence, detached signature). The two are bundled together in a JSON-LD evidence bundle for transport but are independently addressable inside it. The `eatf-verifier` Python package, published on PyPI alongside the v0.2.1 release (Tyche Institute 2026), validates an evidence bundle by checking the action-attestation signature against the embedded identification material and then surfaces the validation verdict together with the evidence on which it rests. The structural identity/action split is therefore already in place; what is missing is its formal status as a load-bearing invariant of the framework rather than an implementation convention.

7.2 What v0.3 needs to add

Four additions, in priority order:

1. **Codify the identity/action split as an architectural invariant.** The spec text should state that conforming EATF deployments separate identification from action attestation as two independent interfaces, that the action attestation references the identity assertion by hash, and that switching the identification modality does not require re-issuing the action receipts. Conformance profiles for each interface should be specified independently.
2. **Add PAdES-B-LTA and ASiC-E as referenced baseline emission formats.** Per §5.5. JSON-LD evidence bundles may remain as a permitted internal exchange format; the externally-emitted, verifier-consumable artifact for any agent receipt that needs long-term verifiability should be one of the two ETSI baselines.
3. **Cite DSS as architectural prior art in the reference-verifier README and adopt DSS-equivalent test vectors where possible.** Per §5.4. This is a documentation-and-test-vector task and does not require any code change.
4. **Ship the EUDI Wallet mapping document.** Per §6.3. Non-normative, living, tracked against ARF version updates.

7.3 What v0.3 should explicitly *not* add

The temptation will arise to layer a registry, a marketplace, a verification-as-a-service offering, or a Trusted List of agent issuers on top of the brokered-architecture mapping. We argue against all four. The argument is made at length in our prior essay *On the Crossroads — Marketplace vs. Distributed Trust in Agent Attestation Frameworks* (Sokolov 2026b) and is not relitigated here. The summary: the broker pattern of §2 is compatible with a distributed-trust deployment in which every relying party manages its own trust anchors; introducing a framework-level discovery, certification, or verification-as-a-service layer converts that distributed model into a platform model with single-root failure modes the rest of the architecture was designed to prevent. The broker pattern's value is the *façade*, not a central authority. EATF v0.3 should make the façade explicit and stop there.

---

8. Pilot proposal — Estonian regulatory sandbox

The architectural mapping of §7 is not by itself deployable. There is a non-trivial regulatory gap between "EATF v0.3 specifies how to emit PAdES-LTA agent receipts" and "an organization can lawfully operate such a receipt-emitting service in production." The gap is not technical. It is the question of *which* eIDAS service category an agent-attestation-emitting service falls under, and the corollary question of *whether* operating such a service requires QTSP registration for a service class that does not currently exist in the eIDAS Annex of qualified trust services.

A regulatory-sandbox pilot is the cleanest way to make the gap visible and to generate the evidence needed to close it. Estonia's **experimentation framework**, operated by Accelerate Estonia under the Ministry of Economic Affairs and Communications, is the most appropriate venue for such a pilot in the European Union as of 2026.

8.1 Why Estonia, and why now

Accelerate Estonia's experimentation framework, scheduled for adoption into Estonian law in late 2025 (status to be confirmed at the time of submission), is Europe's first cross-sectoral and broad-based regulatory sandbox. The framework is purpose-built for pilots whose technology runs ahead of the existing regulatory frame and whose lessons need to be fed back into legislative or supervisory practice. The Accelerate Estonia mandate — popularly summarised as *"making illegal things legal"* — is the closest institutional match in Europe to the gap this paper has identified.

Estonia is also, structurally, an appropriate jurisdiction. Estonian trust-service infrastructure is mature; the national supervisory authority has accumulated experience with novel trust-service configurations; the country's e-government estate provides a natural set of relying-party use cases. A pilot anchored in Estonia generates evidence at lower friction than a pilot anchored in any other EU member state.

8.2 Scope of the pilot

The proposed pilot is a six-month deployment of EATF v0.3 in which:

- agent action receipts are emitted as **ASiC-E containers with XAdES-LTA signatures**, archival-timestamped and packaged with the EATF Agent Evidence Package profile;
- agent identification is performed via a **brokered identification layer** spanning at least two methods (an existing eID scheme for human-principal-delegated agents, and an attested-key modality for in-process agents); and
- verification is performed by the **open-source EATF reference verifier** (`eatf-verifier`, published on PyPI), running offline against the embedded validation material.

The pilot demonstrates end-to-end federation across at least two identification modalities and at least one cross-border use case (an agent operating in Estonia issuing a receipt validated by a relying party in another EU member state).

8.3 What is asked of the sandbox

The pilot does not propose to operate a Qualified Trust Service Provider. It proposes to demonstrate, under sandbox carve-out, that qualified-shaped agent attestations can be produced inside an existing brokered trust-service deployment, validated long-term, and used by relying parties. Concretely, the sandbox determination should clarify that for the pilot's duration and scope:

- the act of producing qualified-shaped agent-attestation artifacts does not by itself constitute operating an unregistered eIDAS trust service;
- pilot participants may use existing QTSP infrastructure (timestamping authorities, signing services, validation services) under their existing commercial contracts without requiring those QTSPs to declare a new service class;
- the pilot's output artifacts may be cited in academic publication, regulatory consultation, and EATF specification revision without prejudice to future eIDAS service-class definitions.

8.4 Participants

The proposed participant structure is intentionally minimal:

- **Tyche Institute** as the EATF specification and reference-verifier author, and as the academic anchor for the pilot's public outputs;
- one or more **deployed brokers** as production-signing partners, bringing existing QTSP-orchestration infrastructure and existing ETSI-baseline emission capabilities;
- one or more **agent-operating organizations** as relying-party pilot participants, contributing real use cases that exercise the end-to-end deployment.

Specific partner organizations are deliberately not named in this paper. Partner selection should follow the academic norms of disclosed conflict-of-interest review and competitive-neutrality where multiple plausible partners exist.

8.5 Timeline and deliverables

A six-month pilot fits comfortably inside the 6–18-month window Accelerate Estonia typically scopes for its sandbox engagements. The pilot would deliver:

- a working end-to-end deployment of EATF v0.3 over brokered trust-service infrastructure;

- a public validation report co-publishable with the Ministry of Economic Affairs and Communications, documenting deployment experience, identified gaps, and recommendations for eIDAS 2 service-class extension or formal interpretation;
- an EATF v0.3 specification revision incorporating sandbox lessons, published under the existing EATF concept DOI;
- a reproducible reference deployment artifact published on GitHub under the EATF licensing terms; and
- a regulatory-consultation submission to the European Commission on the eIDAS 2 implementing-acts track, drawing on pilot evidence.

The pilot's value to Estonia is first-mover credit on the regulatory-input track for the EU's emerging agent-trust regime. The pilot's value to the broader EATF research program is the generation of deployment evidence that no purely-academic engagement can substitute for.

---

9. Open questions

The architectural mapping of §7 and the pilot proposal of §8 leave several open questions on the table. We surface them here rather than pretend the broker pattern resolves everything.

- **Non-human principal attestations under eIDAS 2 ARF.** The Architecture and Reference Framework does not, in its current published version, explicitly contemplate agent attestations. Whether the ARF's eventual treatment of delegated-authority attestations will accommodate software-agent delegates cleanly, or whether a separate ARF extension will be needed, is an open interpretive question. The ARF working group's decisions over the next 12–18 months should be tracked closely.
- **Broker-of-brokers regress.** If brokers federate identity upward to wallets, do wallets federate identity sideways across ecosystems (an Estonian wallet attesting to a French relying party that the holder is identified by a Belgian eID)? At what layer does the federation collapse, and what is the failure mode if collapse happens at the wallet layer rather than the broker layer?
- **QSCD analog for agents.** Is there a meaningful analog of a Qualified Signature Creation Device for a software agent? Hardware-rooted attestation (TPM, SGX, TEE-rooted) is the obvious candidate; certified remote-signing services are another. The answer materially affects what "qualified" means for agent receipts at all, and whether the regulatory carve-out asked for in §8.3 is even legally coherent in the long run.
- **Single-broker capture.** If the agent-attestation broker layer matures into a winner-takes-most market shaped like payments (Stripe) rather than human identity (Okta + Auth0 + Ping + others), the single-root failure mode EATF was designed to prevent reappears at the broker layer rather than the issuer layer. The broker pattern, like every federation pattern, has its own centralisation risk; that risk is not solved by adopting the pattern, only relocated. EATF's distributed-trust commitments remain necessary precisely because the broker layer itself is not intrinsically distributed.

Acknowledgments

The author thanks the public trust-services and standards communities whose published documentation makes cross-domain architectural comparison possible.

Generative-AI assistance. This article was prepared with mixed-source generative-AI assistance. The body prose was authored directly by the author across April–May 2026 (source of record: `outlines/brokered-trust-for-ai-agents-v0.2.md`). The IEEE Security & Privacy Magazine submission packet — the 50-word abstract distillation, the formatted bibliography, and the cover letter — was prepared by an autonomous AI worker (Hermes CLI orchestrating OpenAI Codex / GPT-5.5) from a

structured brief authored by Anthropic Claude Code (Opus 4.7) in dialogue with the author. All substantive claims, results, design decisions, and references are the author's own and verified by the author, who takes full responsibility. Reported per COPE and ICMJE recommendations.

References

eID Easy and Dokobit and Signicat and Scrive. 2026. "Public broker-layer trust-service product pages: eID Easy, Dokobit, Signicat and Scrive." Official company websites. https://www.eideasy.com/.

ETSI. 2016a. "ETSI EN 319 102-1: Procedures for Creation and Validation of AdES Digital Signatures." ETSI European Standard. https://www.etsi.org/deliver/etsi_en/319100_319199/31910201/01.03.01_60/en_31910201v010301p.pdf.

———. 2016b. "ETSI ESI baseline signature standards: CAdES EN 319 122, XAdES EN 319 132, PAdES EN 319 142 and ASiC EN 319 162." ETSI European Standards. https://www.etsi.org/deliver/etsi_en/319100_319199/31912201/01.03.01_60/en_31912201v010301p.pdf.

———. 2022. "ETSI ESI trust-service policy and trusted-list standards: TS 119 401 and TS 119 612." ETSI Technical Specifications. https://www.etsi.org/deliver/etsi_ts/119600_119699/119612/02.03.01_60/ts_119612v020301p.pdf.

European Commission. 2015. "Commission Implementing Decision (EU) 2015/1505 laying down technical specifications and formats relating to trusted lists." Official Journal of the European Union. https://eur-lex.europa.eu/eli/dec_impl/2015/1505/oj.

———. 2025. "European Digital Identity Wallet Architecture and Reference Framework." EU Digital Identity Wallet GitHub repository. https://github.com/eu-digital-identity-wallet/eudi-doc-architecture-and-reference-framework.

———. 2026. "Digital Signature Service (DSS)." European Commission Digital Building Blocks documentation. https://ec.europa.eu/digital-building-blocks/sites/display/DIGITAL/Digital+Signature+Service+-++DSS.

European Parliament and Council of the European Union. 2014. "Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market." Official Journal of the European Union. https://eur-lex.europa.eu/eli/reg/2014/910/oj.

———. 2024. "Regulation (EU) 2024/1183 of the European Parliament and of the Council of 11 April 2024 amending Regulation (EU) No 910/2014 as regards establishing the European Digital Identity Framework." Official Journal of the European Union. https://eur-lex.europa.eu/eli/reg/2024/1183/oj.

Sokolov, Anton. 2026a. "Agent Trust Framework (EATF) Specification, v0.2.1." Zenodo. https://doi.org/10.5281/zenodo.20273730.

———. 2026b. "On the Crossroads — Marketplace vs. Distributed Trust in Agent Attestation Frameworks." Zenodo. https://doi.org/10.5281/zenodo.20257933.

Tyche Institute. 2026. "eatf-verifier Python package." Python Package Index. https://pypi.org/project/eatf-verifier/.

W3C. 2025. "Verifiable Credentials Data Model v2.0." W3C Recommendation. https://www.w3.org/TR/vc-data-model-2.0/.