LAUNCH WHITEPAPER

OTCP

A Trust Layer for People and AI Agents

Verifiable identity, regulated verification, and accountable delegation for the agentic age — built on the open standards we already have.

v1.0 - 2026-06-30

Conformant by design.
Comments welcome.

Contents

Abstract

Three technologies have matured at once, and they have to be solved together. **Agentic AI** can now act in the world — browse, book, buy, sign, negotiate. **Programmable money** lets value move at machine speed. And **verifiable identity** finally has standards strong enough to prove who someone is without a username and password. None of the three is safe at scale without the other two. An agent that can spend but cannot prove whose authority it carries is a liability. An identity that cannot be delegated is useless to an agent. Money that moves without accountable authorization is fraud waiting to happen.

OTCP is the trust layer that makes the combination accountable. It rests on one load-bearing conviction: **authority must be rooted in a verified, accountable human or legal entity, and every delegation of that authority must be a verifiable, scoped, revocable credential.** For people, OTCP is a wallet and a verification layer that replaces repeated document uploads with reusable, privacy-preserving credentials. For agents, it is the missing layer that lets a person or organization grant an AI agent *bounded, revocable, provable* authority — so a relying party can verify what an agent was authorized to do, by whom, within what limits, and for how long.

This is a vision paper with the implementation underneath it. Everything described here is built on open standards already in production or on a clear standards track, and the core has been demonstrated end-to-end on a physical device. Where a problem is unsolved, this paper says so.

---

1. The moment: three currents, one knot

For a decade, verifiable credentials were a solution looking for a problem. The technology worked — selectively disclosable, cryptographically signed claims a person could carry and present — but demand was diffuse. Two things changed that. The European Union's eIDAS 2.0 regulation mandated a digital identity wallet for every citizen, with deadlines in 2026–2027, turning verifiable credentials from a research interest into a continental requirement. And, more abruptly, AI agents became capable of *acting* — not just answering questions, but taking actions on a person's behalf: filling carts, booking travel, moving money, signing documents.

The agent shift exposed a gap nobody had filled. The protocols that let agents communicate and call tools matured quickly, but they answered *how agents talk*, not *who they are or whose authority they carry*. The consequence is already visible: tool endpoints run with no authentication, and agents describe themselves with claims no one verifies. The plumbing shipped before the identity layer. OTCP exists to supply that layer — and to close the older identity gap beneath it at the same time.

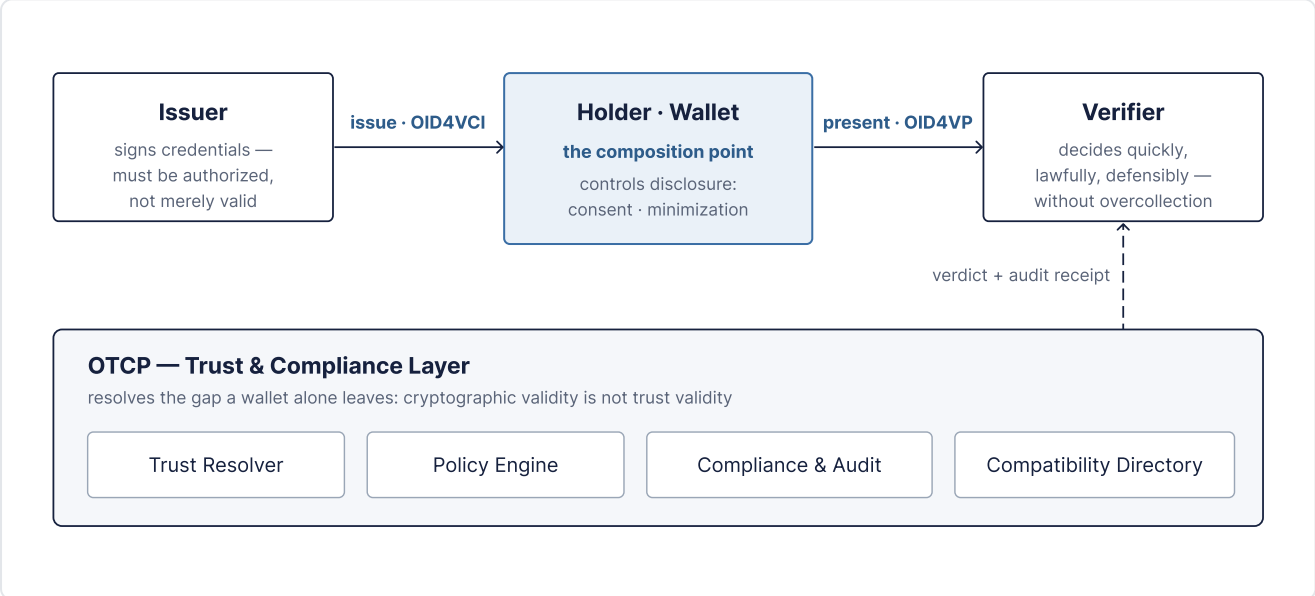

Figure 1. *The trust gap and the three actors. The wallet is the holder's composition point; OTCP resolves what a wallet alone cannot — issuer authorization, status, policy, retention, and audit — returning a verdict and a minimal receipt.*

2. The trust gap

The internet still runs on an identity pattern that is expensive for verifiers and unsafe for users. A person creates an account at each service, uploads a document, a selfie, a proof of address; the service or its vendor stores those records; and the person repeats the ritual at the next bank, marketplace, gambling site, or public portal. This produces three structural failures (Figure 1).

1. **People disclose more than the transaction requires.** A service that only needs to know whether someone is over a legal age receives a full birth date, name, document number, and photo. Data minimization is treated as a slogan rather than an architectural property.
2. **Verifiers duplicate assurance work.** Every relying party rebuilds its own onboarding, screening, document review, status refresh, and audit — inconsistent assurance, high cost, weak interoperability.
3. **Official identity systems are fragmented.** The EU's wallet framework, Ukraine's Diia and BankID, Georgia's eID and trust services, US state mobile driver's licenses, and private credential ecosystems do not share identifiers, formats, certification paths, or trust anchors.

A wallet answers part of this — a person can hold credentials and present selected claims — but a wallet does not remove the need for a **trust and compliance layer**. A verifier still needs to know whether the issuer is *authorized* for a given claim, whether the credential is current, whether the presentation is bound to the holder, whether there is a lawful purpose, what retention applies, and what audit evidence to keep. This is the difference between **cryptographic validity** (the signature checks out) and **trust validity** (the issuer is allowed to make this claim, here, now). OTCP exists for that difference — first for people, then, in a sharper form, for agents.

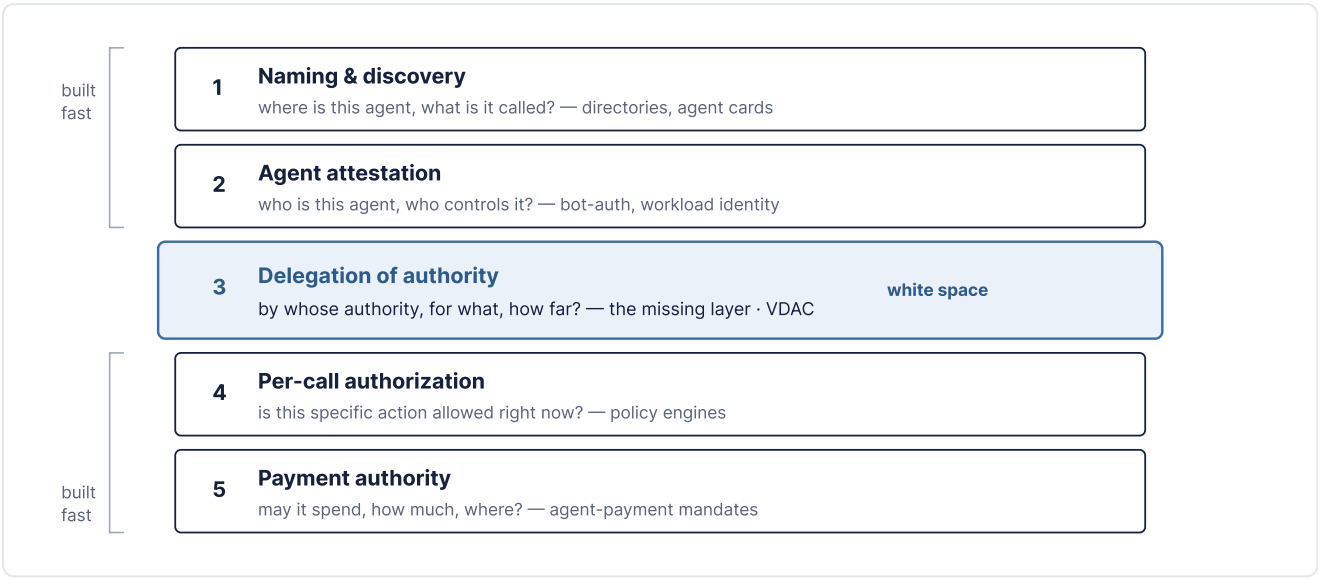

Figure 2. The five layers of agent identity. The field built quickly at the top (naming, attestation) and bottom (authorization, payment); the delegation of authority in the middle is the thin, unsolved layer OTCP occupies.

2.1. The missing layer for agents

When people say "agent identity," they usually mean one of five different things (Figure 2): *naming and discovery* (where is this agent, what is it called); *attestation* (who is this agent, who controls it); **delegation of authority** (by whose authority, for what, how far); *per-call authorization* (is this action allowed right now); and *payment authority* (may it spend, how much, where). The field built quickly at the top and bottom of this stack — directories, bot-attestation, policy engines, payment mandates — and left the middle thin. The verifiable delegation of a real principal's authority to an agent is where the unsolved problems cluster, and it is exactly where OTCP focuses.

3. What OTCP is

OTCP is a **horizontal, standards-compatible, open** trust and compliance layer. Horizontal, because the same rails serve banking, eGovernment, regulated commerce, and age-restricted access. Standards-compatible, because in a trust market interoperability *is* the product — a credential only one vendor can read is worthless. Open, because the buyers are governments, consortiums, banks, and regulated enterprises, and they will not adopt infrastructure that locks them to a single proprietary format or trust list.

It has three parts:

- **A consumer wallet.** A mobile app where a person holds verifiable credentials, proves claims with selective disclosure (online and in person), refreshes and revokes them, and runs a built-in authenticator.
- **A trust and compliance layer.** Issuer and verifier services, trust resolution against federated registries, and compliance workflows — neutral infrastructure any issuer or verifier can plug into.
- **Verifiable Delegated Agentic Credentials (VDACs).** The wallet's most novel capability: a person or organization issues *scoped, revocable authority* to an AI agent. This inverts the wallet's normal direction — instead of receiving a credential, the wallet *becomes* an issuer, minting a delegation credential for an agent.

The first wedge is **age assurance** — proving "over 18" without revealing a birth date — because the regulatory pressure is acute and current, the volumes are high, and it shows privacy-by-design at its most vivid. But the wedge is also the substrate: every vertical reuses the same rails.

---

4. First principles

A small set of principles governs every design decision:

- **Compatibility before novelty.** Build on standards and official profiles wherever they exist; invent only the profile, never the format — and where a standard cannot express a requirement, adopt another standard rather than inventing one.
- **Official identity remains sovereign.** States control their own identity and trust-service regimes. OTCP integrates with official rails when authorized; it cannot become an official issuer through technical design alone.
- **Verification is not authorization to provide a service.** A valid presentation proves a signed claim exists and a key was controlled. It does not make the relying party's service lawful. OTCP returns evidence and policy results; the relying party owns the service decision.
- **Data minimization is a protocol property, not goodwill.** It is built into request templates, wallet UI, verifier APIs, claim-mode selection, and retention policy.
- **Trust anchors are plural.** DIDs are useful but insufficient; regulated ecosystems also use X.509 and qualified certificates, EU trusted lists, mdoc issuer roots, federation identifiers, and national registries. The trust resolver handles all of them without forcing one ideology — and it *consumes* federated trust lists rather than operating a closed list of its own.
- **Restraint in claims.** OTCP aligns with EUDI/eIDAS and national rails but does not market itself as a certified wallet, qualified trust service, or state issuer unless and until the legal role is actually obtained. *Conformant by design, and honest about not-yet-certified,* is a stronger claim than overreaching.

5. How it works for people

5.1. The standards spine

OTCP invents no formats. It builds on **SD-JWT VC** and **ISO mdoc** credentials (the two formats the EU wallet mandates), **OpenID for Verifiable Credential Issuance and Presentation** (the issuance and presentation protocols), the **High Assurance Interoperability Profile**, and the **IETF Token Status List** for revocation. These are the same building blocks the EU's own wallet uses. Conformance is not a constraint on speed — it is what makes independent components interoperate: in building OTCP, a hand-written wallet presentation had to byte-match an independent verifier library, and it did, because both implemented the same standard rather than two dialects.

5.2. The wallet and the credential portfolio

From a single identity-verification session, three credentials are minted into the wallet: an **OTCP ID** (a basis identity credential with selectively disclosable claims), an **Age** credential (independent thresholds at 16, 18, and 21), and a **KYC-Cleared** credential (a reusable verified-person anchor, so a second verifier can confirm a prior clearance without re-running the whole check). A crucial privacy property is enforced at issuance: **personal data does not persist server-side past issuance** — the issuer keeps only a hash commitment and an audit record, not the claims.

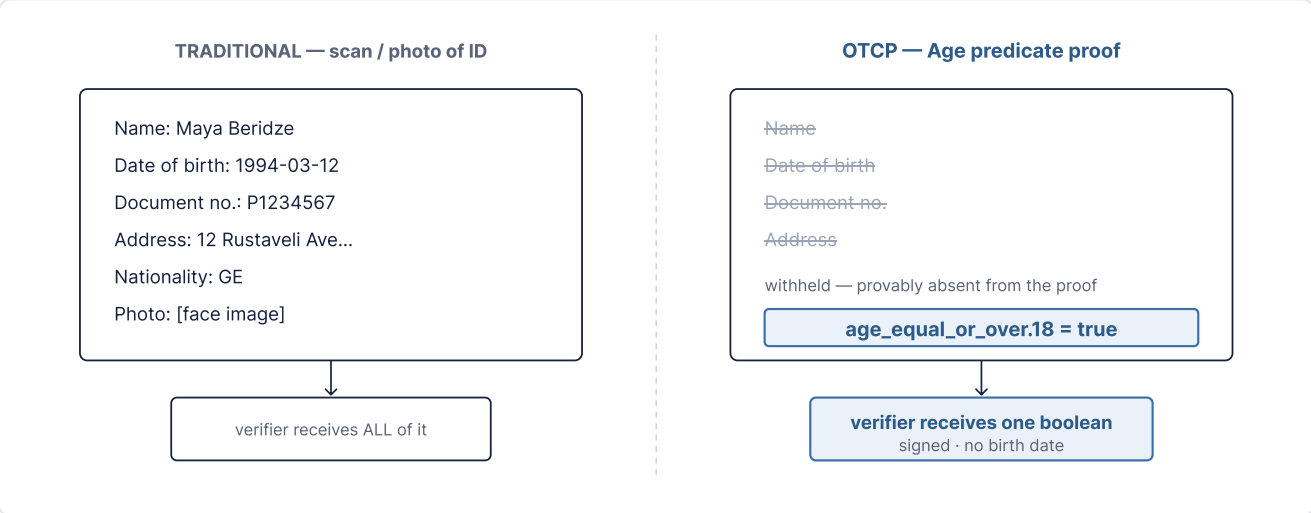

Figure 3. Selective disclosure and the predicate proof. A scanned document discloses everything; the Age credential discloses a single signed boolean with the birth date provably absent from the presentation.

5.3. Selective disclosure and the predicate proof

The Age credential is the cleanest illustration of why verifiable credentials matter (Figure 3). A bar confirming a customer is over 18 has no business knowing their birth date, address, or document number — yet a scanned ID reveals all of it. The Age credential collapses that to a single signed yes/no, with the source birth date provably *absent* from the presentation. This is the difference between a system that minimizes data and one that leaks it by default. Disclosure is governed by the holder's consent: a verifier can *request* more, but it cannot *seize* it.

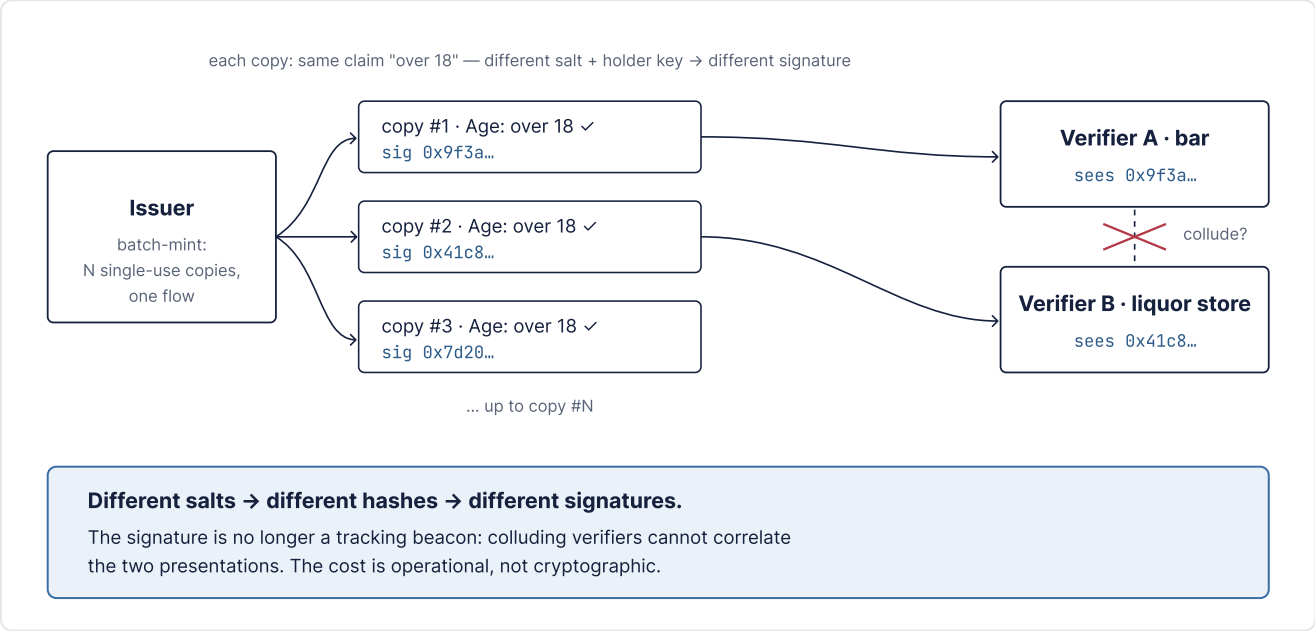

Figure 4. Batch issuance and cross-verifier unlinkability. Each single-use copy carries different salts and a fresh key, producing a different signature, so the signature stops being a tracking beacon and colluding verifiers cannot correlate presentations.

5.4. Privacy that holds up, and EU alignment

Every presentation is bound to the holder's hardware-protected key and to the verifier's one-time challenge, so a captured presentation cannot be replayed. The wallet presents directly to verifiers and never phones home to the issuer, so the issuer cannot build a profile of where a person goes. And to prevent two verifiers from *colluding* to recognize the same person, OTCP uses **batch issuance** (Figure 4): the issuer mints many single-use copies of a credential, each with different random salts and a fresh key, so each presentation carries a different signature and the signature stops being a tracking beacon. The cost is operational, not cryptographic — the crypto stays boringly standard, which is exactly what a trust-critical layer wants.

This is also why OTCP is **EUDI-conformant and on the certification track from day one**: unlinkability for non-identifying attestations is binding EU law (Article 5a(16) of the eIDAS 2.0 regulation), and batch issuance satisfies it without exotic cryptography. Keys live in the device's

secure element and never leave it, satisfying the EU's "sole control" requirement for the highest level of assurance. Formal certification follows the EU's implementing-act timeline; the architecture is built for it now, so certification is a process step rather than a rebuild.

---

6. How it works for AI agents

6.1. Delegation is a credential, not a session

The first and most consequential commitment of the agent design is that **a delegation of authority is a verifiable, portable, signed credential — not a session, a bearer token, or a server-side access-control entry.** In the common pattern, an agent acts with a token, and the record of *who authorized what* lives in a vault on some server, invisible to the parties the agent transacts with; when the token is exchanged at an intermediary, that context is lost. A signed credential the agent *holds and presents* keeps the authority chain intact across intermediaries and lets a verifier check it by signature. The grant is the artifact, not a pointer to a record someone else controls.

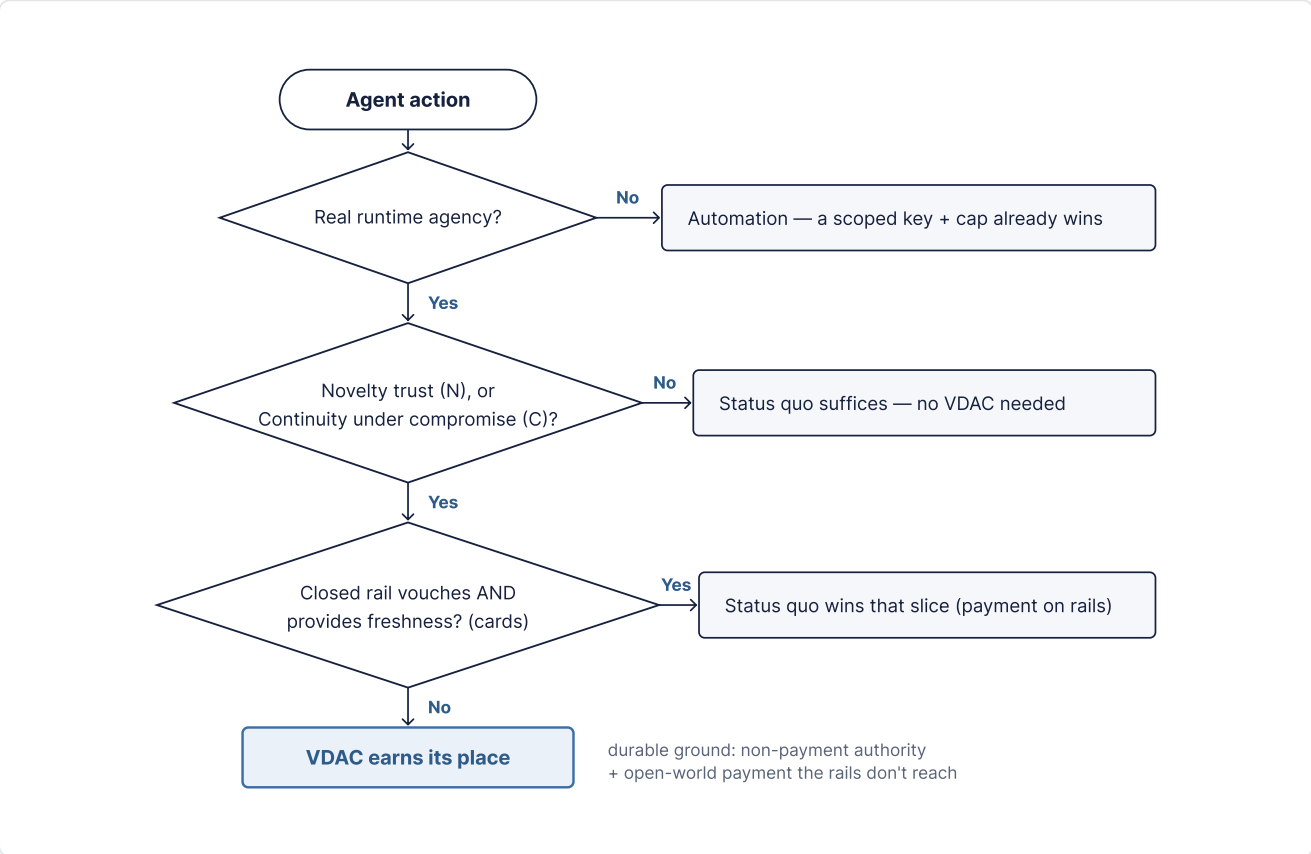

Figure 5. When a delegation credential earns its place. It must clear three gates — the agent has real runtime agency, the case demands novel trust or continuity under compromise, and no closed rail already provides both vouching and freshness. Its durable ground is non-payment authority plus open-world payment.

6.2. When is a VDAC actually needed?

Not every agent task needs a new credential, and a vision that claims otherwise loses credibility. A VDAC has to *beat the status quo*, or it should not drive the design (Figure 5). The test: the agent must have **real runtime agency** — choosing actions and counterparties at runtime, not running a fixed script. Then a VDAC earns its place if the verifier must extend trust to a principal it has **no prior relationship** with, or if the action is sensitive enough to need **per-request proof** that the delegation is fresh, unrevoked, and bound to an uncompromised agent. And there is an important exception: where a closed rail already provides both vouching and freshness — as card networks do for payment — the status quo wins that slice. The durable ground for VDACs is therefore **non-payment authority** (proving attributes, operating tools, binding agreements, signing) plus **open-world payment** to counterparties the card rails don't reach.

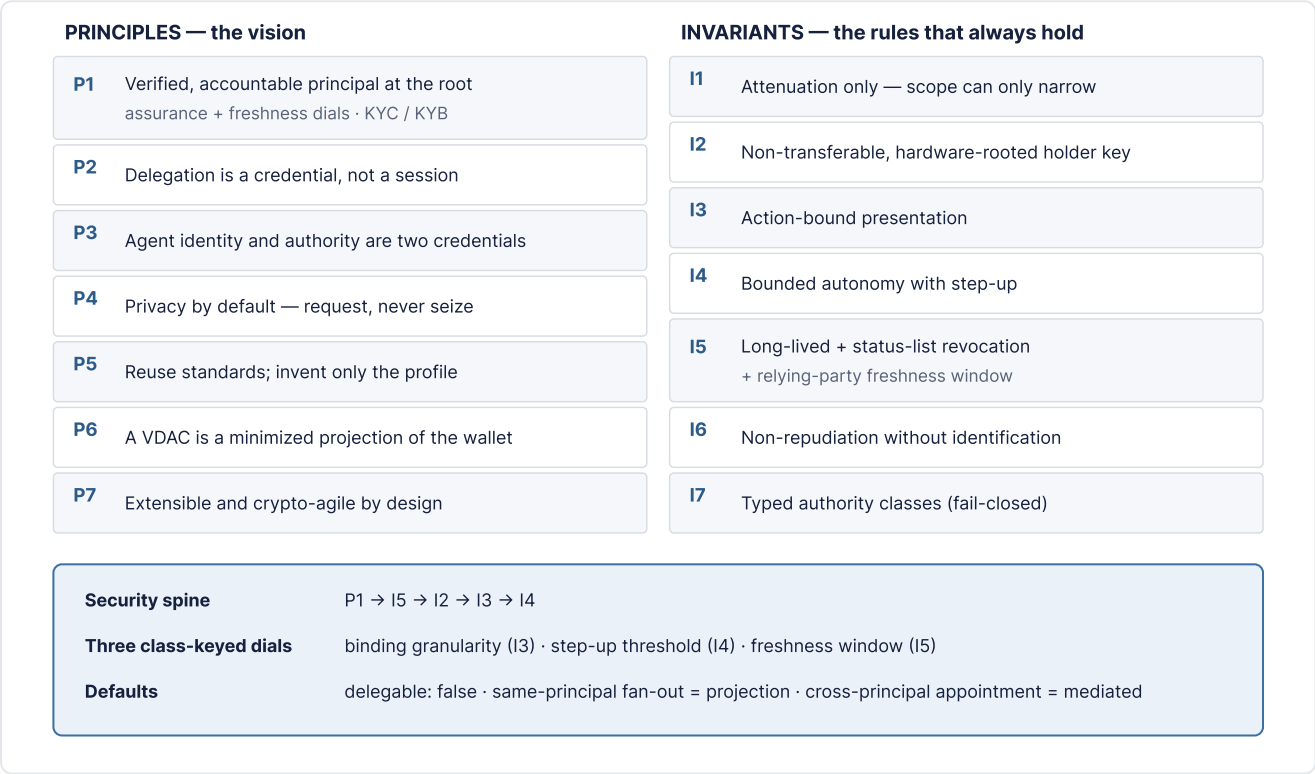

Figure 6. The VDAC constitution at a glance: seven principles (the vision) and seven invariants (the rules that always hold), with the security spine, the three class-keyed policy dials, and the structural defaults.

6.3. A small constitution

The design of VDACs is governed by a compact constitution — seven principles and seven invariants — argued out against concrete scenarios (Figure 6). The principles set the vision: a *verified, accountable principal at the root*; *delegation as a credential*; *agent identity and authority kept as two separate credentials*; *privacy by default*; *reuse standards, invent only the profile*; *a VDAC as a minimized projection of the wallet*; and *extensible, crypto-agile by design*. The invariants are the rules that must always hold: authority can only **attenuate** (narrow, never widen)

down a chain; the holder key is **non-transferable and hardware-rooted**; every sensitive presentation is **bound to the exact action**; autonomy is **bounded by step-up** for high-risk actions; credentials are revocable through a **status list with a freshness window**; actions are **non-repudiable without identifying** the principal; and authority is expressed as a small set of **typed risk classes**.

Authority itself is a closed, legible vocabulary — *present an attribute, operate a tool, spend up to a cap, commit to an obligation, sign a statement* — because the very situation that justifies a VDAC (a cold, untrusting verifier) is the worst place to hand someone an arbitrary capability string they have never seen. Three policy dials — how tightly a presentation binds to an action, when a human must step up, and how fresh the status must be — each tighten automatically as the risk rises.

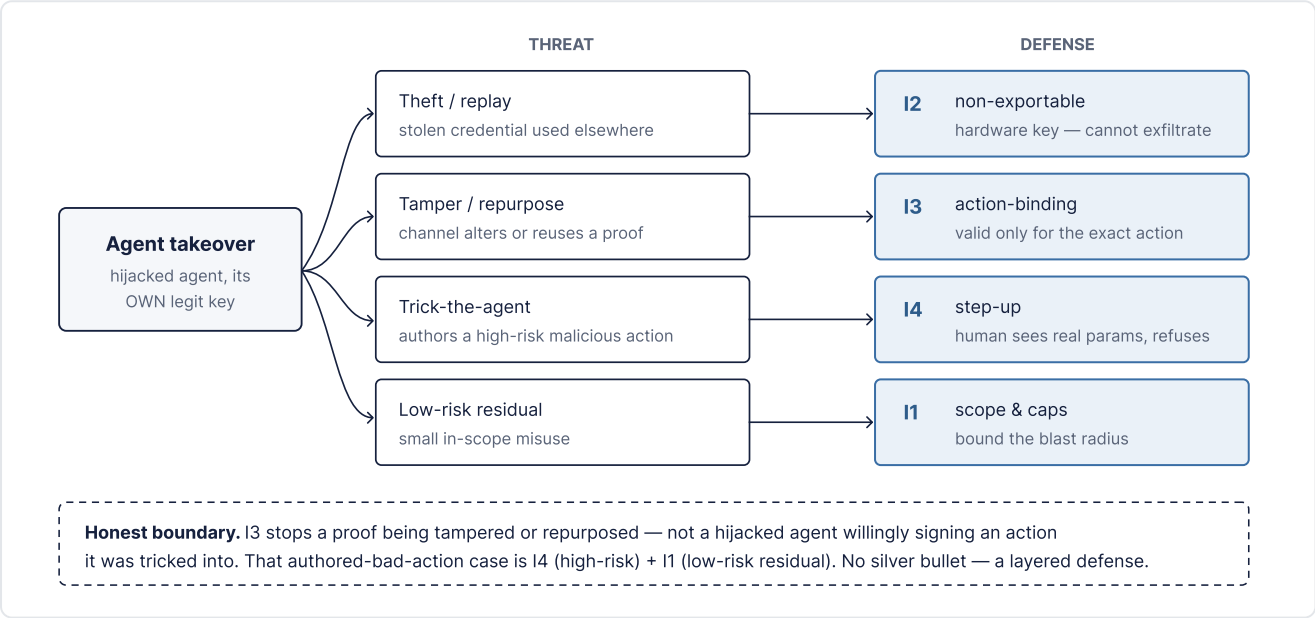

Figure 7. The compromise-defense map for agent takeover. Four threat classes are met by four invariants; the honest boundary is stated explicitly — action-binding stops a tampered proof, not a hijacked agent willingly signing a tricked action, which is handled by step-up and scope.

6.4. Designing for the real threat: agent takeover

The threat to design against is not a credential stolen in transit — it is a **hijacked agent**, taken over by prompt injection or a compromised supply chain, acting with its own legitimate key. The defense is a clean, layered decomposition (Figure 7): the non-exportable hardware key defeats theft and replay; action-binding defeats tampering and repurposing of a proof; **step-up** — a fresh human confirmation, escalating to a biometric co-signature that surfaces the *exact* action — defeats the high-risk case where an agent is tricked into authoring a malicious action; and scope and caps bound the low-risk residual. The honest boundary is stated plainly: action-binding stops a *proof* from being tampered with; it does not stop a hijacked agent from willingly signing a bad action it was tricked into — that case is met by step-up and by scope. There is no silver bullet, but there is a complete defense with each layer's job named honestly.

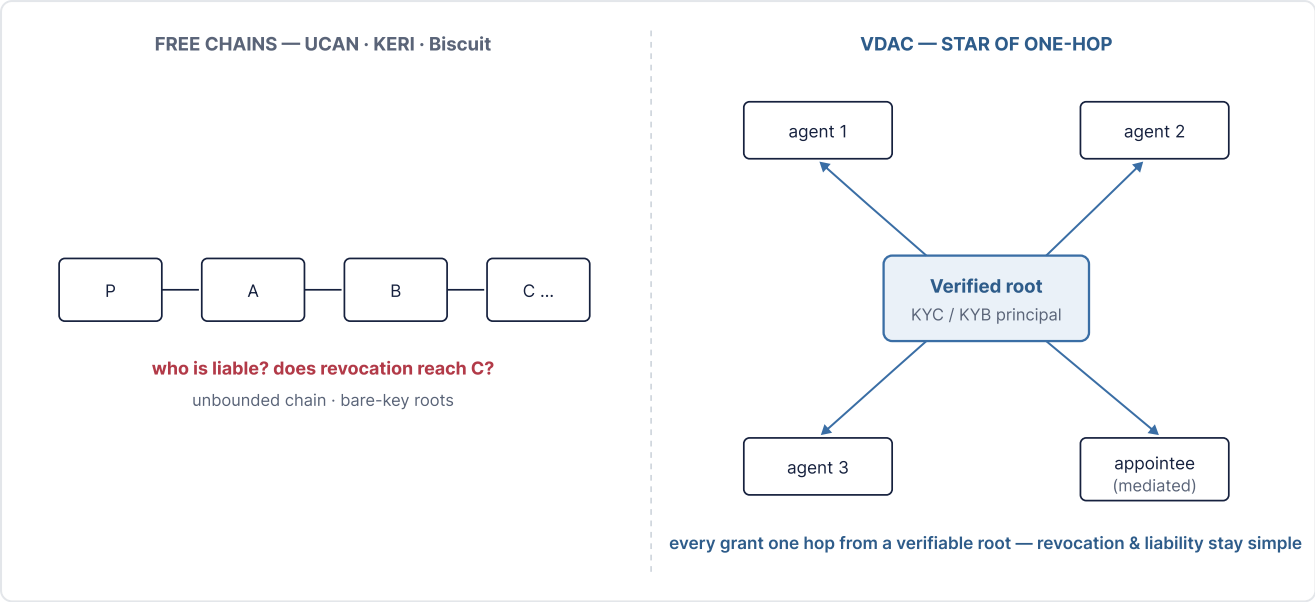

Figure 8. Free chains versus the star of one-hop. Where free-chain systems build unbounded delegation and then fight revocation and liability, every grant in OTCP is one hop from a verifiable root; cross-principal appointment is principal-mediated re-issuance.

6.5. Refusing the hardest unsolved problem — by design

The deepest open problem in agent identity is *recursive delegation*: when one agent sub-delegates to another, who is liable, and does the chain survive? OTCP resolves it by refusing the general case (Figure 8). When a planner agent spawns helpers for one task, they all trace to **one accountable person** — this is the wallet multiplying its own minimized projection internally, within one revocation envelope, not delegation across a trust boundary. The genuinely hard case — one party's agent granting authority to a *different* party's agent — is always handled as **principal-mediated re-issuance**: the authority is re-minted, one hop, freshly rooted in a verified identity. The result is a **star of one-hop grants from verifiable roots, never an unbounded chain** — which collapses the accountability problem by construction. By default, every VDAC is non-delegable; re-delegation is never implicit.

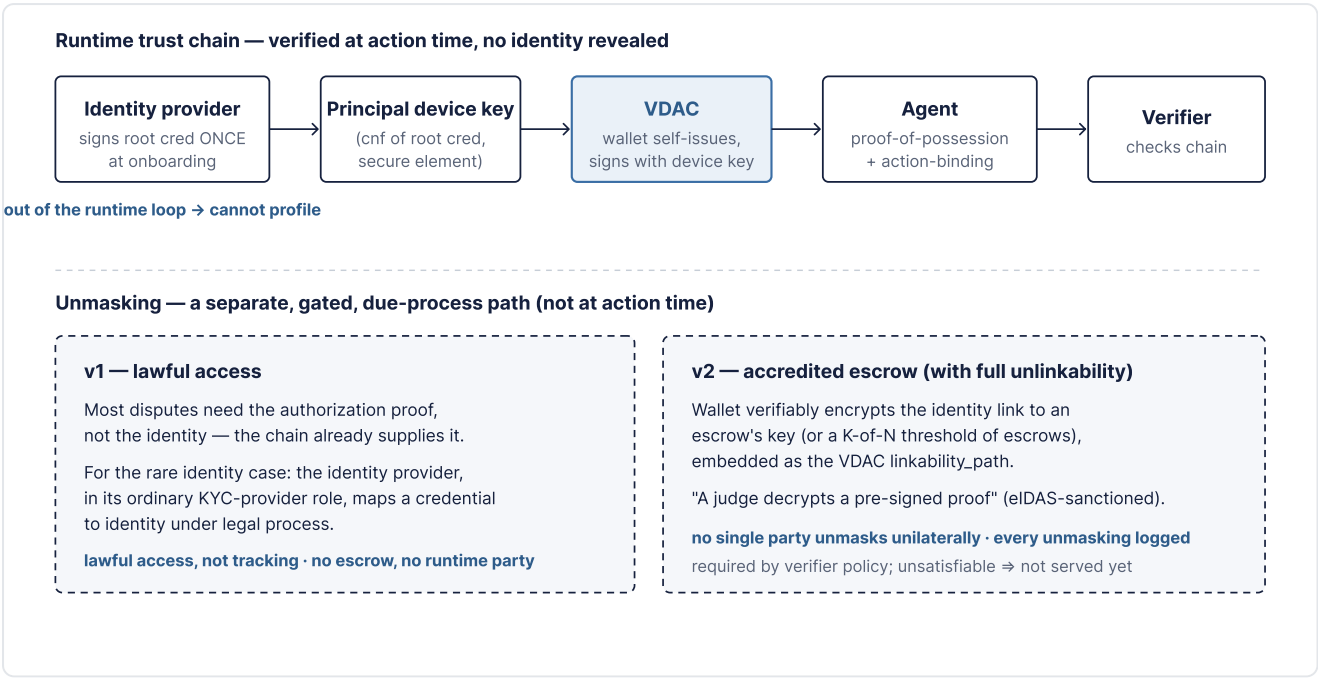

Figure 9. The wallet-centric trust chain and the gated unmasking path. The identity provider signs the root once and leaves the runtime loop; the wallet self-issues the agent's credentials. Identity recovery is a separate, due-process event — lawful access today, an accredited threshold escrow as the ecosystem matures.

6.6. Anonymous, but accountable

A verifier wants assurance that an *accountable* principal stands behind an agent's action, while privacy demands the verifier *not* learn who that principal is — and disputes may later require identifying them. These are reconciled by separating them in time and by party (Figure 9). At action time, the verifier gets cryptographic proof that *an accountable, unmaskable principal* authorized this — not the identity. The identity provider signs the root credential once, at onboarding, and then leaves the runtime loop entirely; the wallet self-issues the agent's credentials. Identity recovery is a separate, conditional, due-process event handled by a party other than the principal — through the identity provider's ordinary lawful-access role today, and through an accredited, threshold-controlled escrow as the ecosystem matures, modeled on the EU's sanctioned "a judge decrypts a pre-signed proof." The trade-off is inherent to accountability, not a flaw: enforceable non-repudiation requires that the principal cannot prevent unmasking — so OTCP's job is to make unmasking gated, auditable, and de-concentrated.

7. Worked examples: people, agents, and one layer

The principles are easiest to trust when you watch them work. Three short scenes — a person alone, an agent alone, and the two together — show why the same layer has to serve both.

7.1. A person, reused

Verify once, reuse everywhere. Someone proves they are over 18 to an age-gated service with a single signed yes/no — no birth date, no document handed over. The same person, weeks later, opens an account at a new bank and presents a reusable "KYC-cleared" credential instead of re-uploading everything; the bank checks that the issuer is authorized and the clearance is still current, and accepts it. The work of verification is done once and reused — less friction for the person, and less liability for every business that would otherwise hoard the documents. This is the quiet, unglamorous half of the vision, and it pays for itself long before any agent shows up.

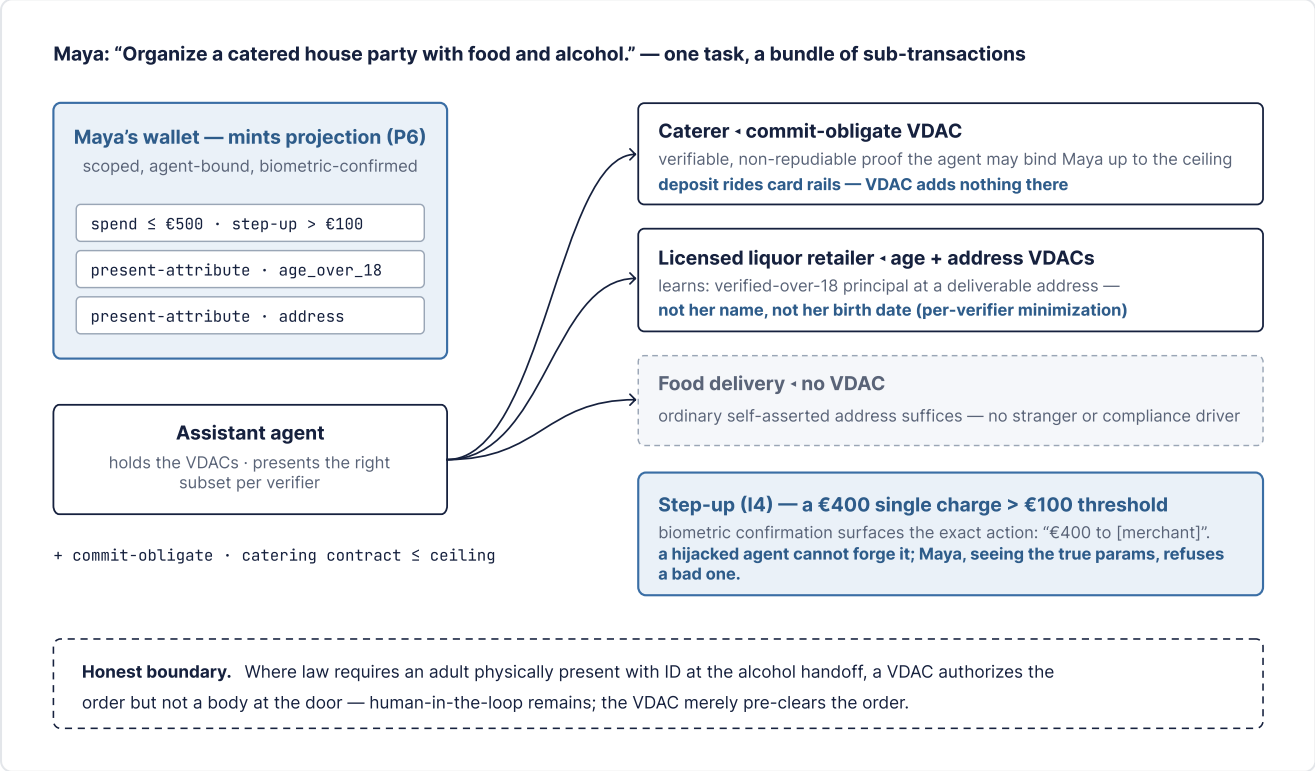

Figure 10. The catered-party walkthrough. The wallet mints a minimized projection; only the sub-transactions crossing a trust boundary with non-payment authority or a regulated attribute earn a credential; a high-value charge triggers a biometric step-up; and the physical handoff stays human-in-the-loop.

7.2. An agent, delegated: the catered house party

Abstract principles are best tested against a messy, realistic task. Maya tells her assistant agent: *"Organize a catered house party with food and alcohol."* This is not one transaction; it is a bundle of heterogeneous sub-transactions across several verifiers (Figure 10). Her wallet does not hand the agent copies of her credentials — it mints a **minimized projection**: a spend authority up to a cap, an age-proof, a delivery address, and an authority to commit her to a catering contract up to a ceiling. She can tighten any of it on an approval screen, never widen it, and confirms with a biometric.

As the agent works, only some sub-transactions earn a VDAC. Paying the caterer's deposit rides the card rails, which already vouch and provide freshness — the VDAC adds nothing there. But *committing Maya to the catering contract*, with its cancellation penalty, is open-ended future liability no card network vouches for — genuine VDAC territory. Ordering the alcohol from a licensed retailer who has never met Maya needs a proof she is over 18 and at a deliverable address — and *nothing more*, not her name, not her birth date. Ordinary food delivery needs none of this. And the physical alcohol handoff, where law requires an adult present with ID at the door, stays human-in-the-loop — a VDAC authorizes the *order*, not a body at the door. If the agent attempts a charge above Maya's threshold, step-up fires: she sees the exact action on her phone and approves with a biometric, which a hijacked agent could not forge. The lesson the party teaches is the one that shapes the whole product: **a real agent task is a bundle, and the trust layer's job is the subset that crosses trust boundaries — not the whole task.**

7.3. The two together

Now combine them. Nino runs a small import business; she verifies once — the company, and herself as its authorized representative. As a *person*, she reuses that to open a bank account and satisfy a regulator's portal. As a *principal*, she delegates a procurement agent and a logistics agent from the same wallet, each carrying a narrow, revocable slice of the company's verified authority. When her procurement agent commits the company to a supplier who has never heard of it, the supplier verifies a chain that ends at the very same identity Nino used to open her bank account — no re-verification, nothing disclosed beyond the deal at hand. One verified identity serves both her and her agents; revoke an agent and only its slice disappears, while her own credentials stay intact.

That is the whole argument in one picture. An agent's authority is not a separate kind of identity to bolt on — it is a projection of a verified human one. Build the human layer well, and the agent layer is the same layer, pointed at a new kind of holder. You cannot get there with two disconnected systems; you get there with one.

8. Payment stays on the rails

OTCP’s relationship to money is defined by one rule: **it authorizes, but never settles.** It never holds funds, never moves value, never takes a cut of a payment. Card networks already provide both vouching and per-transaction freshness for closed-loop payment, so OTCP interoperates with them rather than competing — a spend grant is modeled as a constrained signed mandate that plugs into the emerging agent-payment standards. Where OTCP adds value on payment is the **open-world** corridor the card rails don’t reach: agent-to-arbitrary-counterparty settlement where there is no prior relationship and the counterparty needs presented, verifiable authority. Even there, OTCP authorizes; the agent’s own wallet moves the value. Keeping settlement off the platform is deliberate — it keeps OTCP a neutral trust layer rather than a regulated money business.

9. Built, not just designed

This is not a paper architecture. The wallet is a native Android application built on the EU's reference wallet libraries and the OpenWallet Foundation's Multipaz — the same Kotlin stack that provides mdoc, SD-JWT VC, hardware-backed keys, and the per-use key pool that makes batch issuance work. The services are standards-based issuance, presentation, and status endpoints. The full round-trip — real identity verification, issuance of three credentials, and a data-minimized age presentation verified server-side — has been demonstrated on a physical device, including the agent flow in which a wallet self-issues a delegation and a verifier checks the whole chain. Building on the EU's own reference implementation is also the strongest possible posture for eventual certification.

10. What is not solved

A serious proposal names what it has not solved:

- **Cross-border assurance equivalence.** Comparing assurance levels across the EU, Ukraine, Georgia, and private issuers is genuinely hard; OTCP resists simplistic equivalence tables in favor of profile-specific, source-cited rules.
- **Collusion-resistant privacy.** Batch issuance defeats correlation by honest verifiers, but not an issuer that colludes with a verifier; zero-knowledge proofs over today's credential formats are the promising research direction, not yet a shipping reality.
- **Recursive delegation across principals.** OTCP refuses the general case and requires mediated, one-hop, re-rooted appointment; the general cryptographic-chain problem remains unsolved by anyone, and OTCP sidesteps it rather than claiming to solve it.
- **Business identity at depth.** Legal-entity identity needs registries, representative authority, and role credentials — a distinct line of work, not "a business treated as a large person."

Naming these is the restraint principle applied inward. A trust layer that overclaims is a contradiction in terms.

11. An open invitation

The agent economy will get an accountability layer one way or another. The choice is whether it is rooted in self-asserted keys and replayable tokens, or in verified, accountable, privacy-preserving human authority. OTCP is a proposal for the latter, and it is offered openly: the credential formats are standard, the trust model consumes federated lists rather than operating a closed one, and the delegation profile and the principles in this paper are offered as a contribution to the standards community — the Decentralized Identity Foundation's work on trusted AI agents and delegated authority, the OpenID Foundation, the W3C, and the EU's emerging work on representation and mandate attestations.

The thesis we want to plant, before the agentic economy hardens around weaker foundations, is simple: **agent delegation should be rooted in verified human identity, expressed as a scoped and revocable credential, and provable without surrendering privacy.** If you build wallets, verify identities, write the standards, or regulate this space, we would like to build it with you.

12. Conclusion

OTCP is a single answer to two problems that have to be solved together. The older problem — repeated proofing, overcollection, fragmented official rails — is met by a horizontal, standards-compatible trust layer with privacy built in as an architectural property. The newer problem — agents that can act but cannot prove whose authority they carry — is met by rooting every delegation in a verified, accountable principal and expressing it as a scoped, attenuating, revocable, action-bound credential. The vision is large; the foundations are deliberately conservative; and the core is already running on a real device. Verifiable identity for people, accountable delegation for their agents, and privacy that is structural rather than promised — that is the trust layer the next decade needs, and it can be built on the standards we already have.

OTCP — Open Trust and Compliance Platform. Launch whitepaper v1.0, 2026-06-30. Conformant by design. Comments and collaboration welcome.