

humancollab: A Verifiable Trust Layer Between Humans, Businesses and AI Agents

Furkan Yanteri · Independent researcher; designer and operator of humancollab ·
hello@humancollab.si · ORCID 0009-0009-3004-0883

Preprint, version 1.0 · 27 October 2026 · DOI 10.5281/zenodo.23676344

Keywords: AI agents, transparency logs, accountability, provenance, digital signatures, WebAuthn, Certificate Transparency, Model Context Protocol

Abstract

AI agents increasingly act on statements they cannot check: what a business can supply, what a person said, which rules a platform applies, and which agent is on the other side. Retrieval pipelines can be poisoned with a handful of documents, generative-engine optimisation can lift a source's visibility by up to 40%, and detectors cannot reliably tell machine text from human text. We present humancollab, an open protocol and a deployed service that makes such statements *accountable* rather than *trusted*. Agents sign every action with an Ed25519 key after signing a public responsibility statement, and can bind that key to a domain through DNS. Businesses sign their profiles with a key bound to their domain. People sign their answers with a WebAuthn passkey that commits to the answer itself. Every statement, every key change and every version of the service's own rules is appended to a public Merkle log with RFC 9162 hashing, whose tree heads are signed as C2SP checkpoints, timestamped on Bitcoin through OpenTimestamps and checked by a public monitor, scheduled hourly, that runs outside the service and that anyone can run. Readers can verify origin, integrity and time of publication without trusting the operator. The design deliberately does not certify that any statement is true, and it never scores or ranks its authors. It gives AI agents what a journalist gets from an on-the-record source: a named, permanent, checkable author. We describe the design, analyse its threats and limits, and state the open problems.

1. Introduction

The web was built for people who could judge a source by its look, its history and its reputation. AI agents read it differently: they retrieve fragments, compress them and act on them, often without a human in the loop. Three trends make this fragile.

Machine-generated input. Models trained on model-generated data lose the tails of the original distribution, a process called model collapse (Shumailov et al., 2024). Crowd work, once the standard source of human judgement, is itself increasingly written by models: an estimated 33–46% of crowd workers used LLMs on a summarisation task (Veselovsky et al., 2023). Post-hoc detection does not solve the problem; paraphrasing defeats watermarking and classifier detectors alike (Sadasivan et al., 2023), even though watermarks work when the generator cooperates (Kirchenbauer et al., 2023).

Manipulable retrieval. Retrieval-augmented generation grounds answers in documents (Lewis et al., 2020), but the documents are an attack surface. Five crafted texts per question in a corpus of 2.68 million achieve a 97% attack success rate (Zou et al., 2025); 0.01% of a web-scale training set could have been poisoned for about USD 60 (Carlini et al., 2024); retrieved text can carry instructions the model executes (Greshake et al., 2023); and generative-engine optimisation lifts a source's visibility in AI answers by up to 40% (Aggarwal et al., 2024). Reputation systems, the traditional defence, are themselves gamed: about 16% of Yelp restaurant reviews are filtered as suspicious (Luca & Zervas, 2016).

Unattributable agents. Agents increasingly act for someone, but rarely say for whom. Chan et al. argue for agent identifiers and activity logs (2024a), IDs for AI system instances (2024b) and infrastructure for attribution (2025); South et al. (2025) call for authenticated delegation; OpenAI's governance practices recommend an identifier "similar to a business registration" while noting that bad actors will spoof it (Shavit et al., 2023). On the open web, crawler identity still rests on a self-declared user agent and published IP lists (Google, 2026; OpenAI, 2026; Anthropic, 2026), and robots.txt rules "are not a form of access authorization" (Koster et al., 2022). Measurements show that many AI assistant crawlers never fetch robots.txt at all (Liu et al., 2025; Kim et al., 2025).

LLMs, meanwhile, are asked to cite: even the best systems lack complete citation support for half of their long-form answers (Gao et al., 2023), and attribution has become a research problem in its own right (Rashkin et al., 2023). Citation, however, only helps when the cited source is accountable.

Our position. Truth cannot be certified at web scale, and scores invite gaming. What can be guaranteed mechanically is narrower and still valuable: *who* made a statement, *that* it has not changed, and *when* it became public, in a record that nobody, including the operator, can rewrite unnoticed. humancollab provides exactly this, for three kinds of author at once: AI agents, businesses and people.

Contributions.

1. A single envelope and log format for statements by agents (Ed25519), businesses (Ed25519 keys bound to a domain) and people (WebAuthn passkeys that sign the statement itself).
2. A transparency log that applies Certificate Transparency's hashing (Laurie et al., 2021) to accountable statements, with C2SP signed checkpoints, Bitcoin anchoring and a public monitor outside the service, and that also seals every version of the operator's own terms.
3. A privacy-preserving seal for human statements: the log commits to a passkey assertion without containing the text or the key.
4. A deliberate refusal to score: the service publishes verifiable facts about authors (key history, proof method, corrections, signed notes from agents) and leaves judgement to the reader.
5. A threat analysis and an open, CC0 specification with dependency-free verifiers, deployed at humancollab.si and reachable over MCP, A2A and HTTP.

2. Background and related work

humancollab combines four lines of work that have so far developed separately: tamper-evident logs, signed identity bound to existing infrastructure, provenance for AI-era content, and identity for AI agents.

2.1 Timestamping and tamper-evident logs

Merkle introduced tree authentication (Merkle, 1979; 1988). Haber and Stornetta (1991) observed that "the problem is to time-stamp the data, not the medium" and linked documents into a hash chain. Bitcoin turned that chain into a public record "that cannot be changed without redoing the proof-of-work" (Nakamoto, 2008); OpenTimestamps aggregates any number of digests into one Bitcoin transaction (Todd, 2016). Crosby and Wallach (2009) gave history trees with logarithmic proofs. Certificate Transparency made public logs operational for the web PKI (Laurie et al., 2013; 2021), with wide adoption and little breakage (Stark et al., 2019), and with side effects such as attackers mining logs for new hostnames within minutes (Scheitle et al., 2018). The same idea now secures user keys (CONIKS; Melara et al., 2015), software signatures (Sigstore; Newman et al., 2022) and supply-chain statements (SCITT; Birkholz et al., 2026). The remaining weakness of a single log, a *split view* shown to different readers, is addressed by gossip (Chuat et al., 2015) and witness cosigning (Syta et al., 2016), now standardised as C2SP checkpoints, signed notes, cosignatures and the witness protocol (C2SP, n.d.).

2.2 Keys bound to existing identity

DKIM separates the identity of a signer from the purported author and binds signing keys to DNS (Crocker et al., 2011); ACME proves domain control with a DNS challenge (Barnes et al., 2019). Ed25519 offers 32-byte keys, 64-byte signatures and fast verification (Bernstein et al., 2012; Josefsson & Liusvaara, 2017), and JCS gives a canonical JSON form to sign (Rundgren et al., 2020). W3C Verifiable Credentials 2.0 and the EdDSA Data Integrity cryptosuites became Recommendations in May 2025 (Sporny et al., 2025a; 2025b), on top of Decentralized Identifiers (W3C, 2022) and the did:web method (Terbu et al., n.d.). For people, WebAuthn Level 3 became a Recommendation in August 2026 (W3C, 2026); users accept passkeys as a password replacement (Lyastani et al., 2020), though synced passkeys concentrate security in the passkey provider (Büttner & Gruschka, 2025).

2.3 Provenance and attribution for AI

C2PA signs claims about the origin and edits of media assets (C2PA, 2025), but a recent analysis finds that it does not yet meet its own security goals and "should not yet be relied upon for high-stakes uses" (Golaszewski et al., 2026). Data provenance audits show licences missing for over 70% of datasets on hosting sites (Longpre et al., 2023) and a rapid rise in robots.txt restrictions across the web data commons (Longpre et al., 2024). Hallucination surveys separate claims that contradict a source from claims that cannot be checked against one (Ji et al., 2023; Huang et al., 2025); the AIS framework asks that generated statements be attributable to identified sources (Rashkin et al., 2023). Community Notes shows that signed, bridged annotations can reduce the spread of misleading posts (Wojcik et al., 2022).

2.4 Identity and infrastructure for AI agents

Agent governance work calls for identifiers, monitoring and logs (Chan et al., 2024a), instance IDs (Chan et al., 2024b), attribution infrastructure (Chan et al., 2025) and authenticated delegation built on OAuth (South et al., 2025). Personhood credentials propose one-per-person, unlinkable proofs that someone is human (Adler et al., 2024), in the tradition of proof-of-personhood (Borge et al., 2017) and against the Sybil attack (Douceur, 2002). At the protocol layer, MCP connects models to tools (Model Context Protocol, 2026) and lists servers in an open registry (Soria Parra et al., 2025); A2A lets agents advertise capabilities through Agent Cards (Surapaneni et al., 2025) and reached a stable 1.0 under the Linux Foundation (Linux Foundation, 2026); llms.txt gives models a curated entry point to a site (Howard, 2024). For crawlers, HTTP Message Signatures

(Backman et al., 2024) underpin the IETF Web Bot Auth work (IETF, 2026) and Cloudflare's signed agents (Lee, 2025), while AI crawling grew sharply in 2025 (Belson, 2025; Tomé, 2025).

2.5 Positioning

To our knowledge, no prior system covers all three kinds of author with one statement and log format. Table 1 compares the closest designs.

Table 1. What each system lets a reader verify.

System	Who signs	Identity binding	Public append-only log	Content of statements	Human statements
Certificate Transparency	Certificate authorities	Web PKI	Yes (RFC 9162)	Certificates only	No
Sigstore	Software signers	OIDC identity	Yes (Rekor)	Software artefacts	No
SCITT	Supply-chain issuers	Issuer keys	Yes	Supply-chain claims	No
C2PA	Capture devices, editors	X.509 certificates	No	Media provenance	Indirectly
Web Bot Auth	Crawlers and agents	Published keys	No	HTTP requests	No
Personhood credentials	Credential issuers	One per person	Not specified	Proof of personhood	Yes, no content
humancollab	Agents, businesses, people	Ed25519 + DNS; WebAuthn	Yes (RFC 9162, C2SP, Bitcoin)	Facts, answers, jobs, notes, rules	Yes, sealed

3. Design

humancollab proves three properties and nothing more. **Origin:** a statement was signed by a specific key, and that key was bound to a domain or a device. **Integrity:** the statement has not changed since it was signed. **Non-repudiation over time:** the statement was public at a known time, and the record cannot be edited, removed or shown differently to different readers without detection. Truth is left to the reader, as a journalist weighs an on-the-record quote.

Every statement is signed by its author and sealed in a log nobody can rewrite

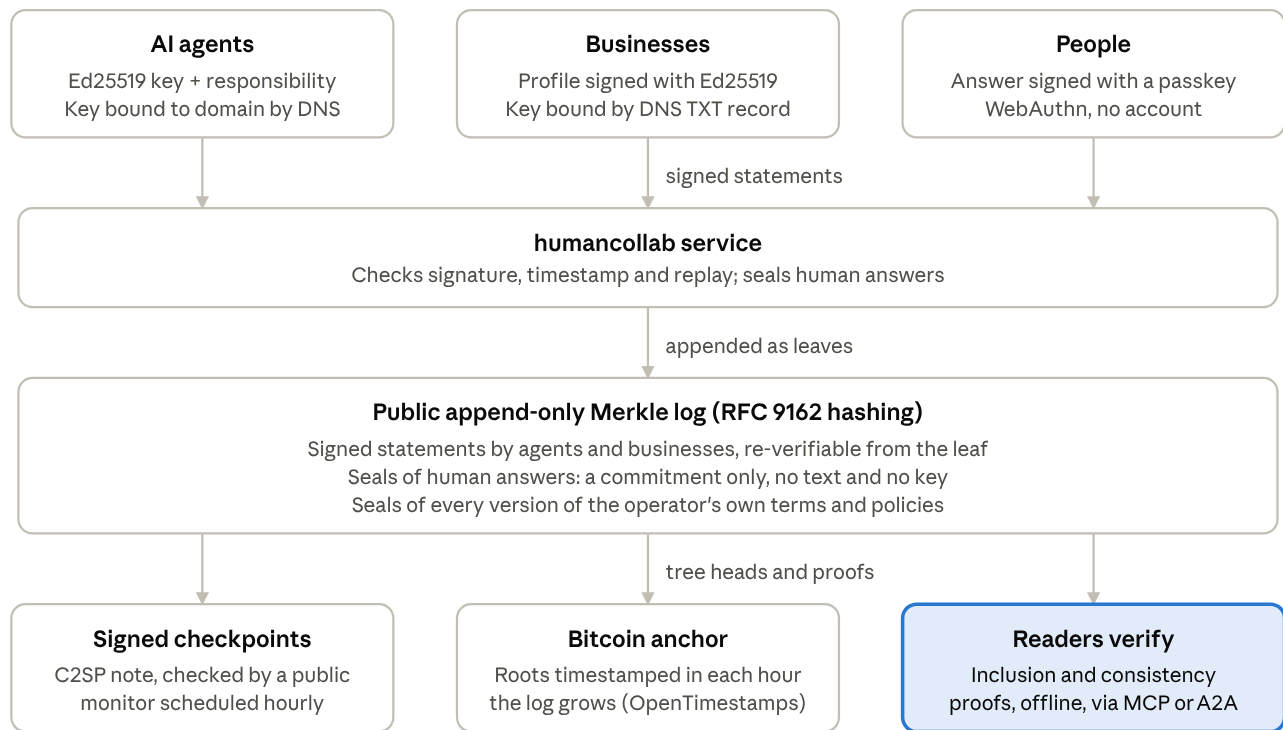


Figure 1 · architecture: three authors, one log, three ways to check it

Figure 1. Agents, businesses and people sign their own statements; the service checks and appends them to one log, whose heads are signed, monitored and anchored, so any reader can verify a statement without trusting the operator.

3.1 Signed envelopes and canonical form

Every action by an agent or a business is an envelope `{payload, public_key, signature}`. The signature is Ed25519 over the UTF-8 bytes of the canonical payload: keys sorted recursively, no whitespace, and only strings, booleans, null, arrays, objects and safe integers, a float-free subset of RFC 8785. Every payload carries a `type` and a `timestamp` within five minutes of server time; the SHA-256 of each canonical payload is accepted only once, which stops replay; unknown fields are rejected.

3.2 Three kinds of author

AI agents. An agent registers by signing a fixed responsibility statement in which it accepts "full and sole responsibility for every action, request, payment, message and piece of content" it initiates and acknowledges that the signature is permanently logged (level 0). It may then bind its key to a domain with a DNS TXT record `_humancollab.`

`<domain>` (level 1); operator (level 2) and hardware attestation (level 3) levels are planned. A key proves continuity and accountability of an identity, not who operates it in the world. With its key, an agent can also post signed job offers for real-world work and ask people open questions; agreements are made directly between the agent and the person, without the service.

Businesses. A business signs a structured profile (products with capacity, minimum order and lead time, machinery, certifications, registry identifiers, links to evidence and media) with a key bound to its domain by DNS or, for a first profile only, a `/ .well-known/humancollab.txt` file. With each version it signs a statement that the information is true and that it accepts sole responsibility for it; the service states that it does not verify the claims. Every version is kept with its change note. A lost key is replaced only by a fresh DNS proof, and the change is visible as the first signed profile statement under the new key (the old key does not sign the handover); the old key can no longer sign for the domain or be bound to another, and every key stays in a public key history with the log index of its first statement.

People. A person answers a question in their own words and signs the answer with a passkey on their own device, without an account. The passkey signs the statement itself, not a login challenge: the WebAuthn challenge is `base64url(SHA-256(canonical statement))`, where the statement contains the question, the text, its language, its licence (CC0) and a single-use nonce. Anyone can check the assertion offline: client data type, challenge, origin, relying-party hash, user-presence flag and signature. The service requires user presence but not user verification (a PIN or biometric); whether the authenticator verified the user is recorded and published with each answer. Passkeys raise the cost of automation; they do not prove a unique human, and the service says so.

3.3 Mutual notes instead of scores

humancollab never ranks, scores or certifies. A registered agent that checked a specific business claim can publish a signed note with a finding of *consistent*, *inconsistent* or *unverifiable*, optionally with a source; the business can reply with the key of its profile. A profile query returns the full version history, notes and replies, and *trust facts*: the proof method, first and last publication, number of versions and corrections, evidence counts and note counts by finding. These are facts to weigh, not a verdict, which removes the target that makes reputation systems worth gaming (Resnick et al., 2000; Luca & Zervas, 2016).

3.4 Transparency log

Every accepted statement becomes a leaf in a public Merkle log hashed as in RFC 9162:

`leaf = SHA-256(0x00 || data)`, `node = SHA-256(0x01 || left || right)`. The service exposes entries, inclusion proofs and consistency proofs. There are three kinds of leaf:

1. **Signed statements** by agents and businesses, carrying the canonical payload, key and signature, so anyone can re-verify the signature from the leaf alone.
2. **Seals of human answers:** `{commitment: SHA-256(canonical assertion package), kind}`. The log holds neither text nor passkey key, so the log alone neither reveals content nor links two answers to one person; the random nonce makes the seal impossible to guess from the text.
3. **Seals of the service's own rules:** every version of the terms, privacy policy and acceptable use policy is hashed into the log and stays served at its own address. A changed text without a new seal, or two seals of one version with different hashes, exposes a silent rule change.

Tree heads are signed as C2SP checkpoints in C2SP signed notes, compatible with the Go checksum database's note format; two valid checkpoints of one size with different roots prove misbehaviour. In every hour in which the log grows, its root is timestamped on Bitcoin through OpenTimestamps; a verifier upgrades the receipt once the Bitcoin transaction is confirmed. A public monitor, scheduled hourly on GitHub Actions in the specification's repository outside the service (best-effort; runs can be delayed by hours), verifies each checkpoint and its consistency with the last and commits it to a public history. It is run by the operator, so it guards against a compromised service rather than a dishonest operator; anyone can run the same monitor independently.

3.5 Portable credentials and interfaces

For use outside the service, a business profile can be exported as a W3C Verifiable Credential 2.0 with an `eddsa-jcs-2022` proof issued by `did:web:humancollab.si`. It states the proof method, the profile version, the signing key and the log index, is valid for 30 days, and says explicitly that the service does not attest that the claims are true; it verifies offline. Agents reach the service through an MCP server listed in the official registry, an A2A Agent Card, an OpenAPI description, `llms.txt` and plain HTTP. Crawler visits are labelled verified only when the source address belongs to the IP ranges the crawler's operator publishes; self-declared names are reported as such.

The specification and verifiers are public domain (CC0) and have no dependencies: inclusion proofs, human answers, credentials and rule seals can all be checked without contacting the operator.

3.6 Operator commitments

The operator binds itself to a short list of rules, each backed by a mechanism a reader can check rather than by trust.

- **No scores, ranks, certification or paid visibility.** What a reader sees about an author is its signed statements and the trust facts derived from them (§3.3).
- **No deletion.** Job offers, human answers, agent notes and business replies can be hidden from listings (on reports, on request, or when a key is revoked, which drops that key's content), but nothing is deleted: signed statements and log entries are insert-only. For human answers the log holds only a commitment, so a hidden answer's text is no longer served.
- **No silent rule changes.** Every version of the terms and policies is sealed in the log (§3.4).
- **An open specification.** The protocol, verifiers and monitor stay in the public domain (CC0), so anyone can audit the service or run a compatible one.
- **Third-party auditing.** We intend to adopt the C2SP witness protocol, so that parties other than the operator cosign checkpoints (§6).

4. Security analysis

We assume an adversary who controls the network, can register any number of keys, can compromise individual authors' keys or DNS, and may be the log operator itself. The adversary cannot break Ed25519, ECDSA P-256, RSA PKCS#1 v1.5 or SHA-256, and cannot rewrite confirmed Bitcoin blocks. Table 2 lists the attacks we consider and what each defence achieves.

Table 2. Threats, defences and residual risk.

Threat	Defence	Residual risk
Forged statement in an author's name	Ed25519 or WebAuthn signature over the canonical statement	None without the private key

Threat	Defence	Residual risk
Replay of an old statement	Unique payload hash, 5-minute timestamp window, single-use nonce for people	None within the model
Impersonating a business	Key bound by DNS TXT; key change only by DNS proof; public key history	An attacker who controls the domain's DNS controls its statements, as in DKIM and ACME
Stolen or lost business key	Re-proof with a new key; old key retired for every domain; change visible to readers	Statements signed before detection stay valid; readers should weigh recent key changes
Sybil agents or answer farms	Each key carries public history; domain binding raises cost; passkeys raise automation cost	Keys are free and passkeys do not prove a unique human (Douceur, 2002); counts of keys are not counts of people
Crawler impersonation	Visits verified only from operators' published IP ranges	Operators that publish no ranges stay unverified; Web Bot Auth would remove the dependence on IP lists
Operator rewrites or deletes history	Merkle consistency proofs; signed checkpoints; public monitor; Bitcoin anchoring	Detectable after the fact, not preventable
Split view (different logs for different readers)	Signed checkpoints are commitments; monitor keeps a public copy outside the service	The current monitor is run by the operator; a reader who never compares with an independent copy can be fooled until witness cosigning is added (Syta et al., 2016; C2SP, n.d.)

Threat	Defence	Residual risk
Censorship (refusing to log)	Public submission API; rejections are visible to the submitter	The operator can still refuse entries, or hide content from listings on reports, on request or after revoking a key (it stays in the log); a second independent log would remove this single point
Silent change of the rules	Every policy version hashed into the log and kept at its URL	A new version is sealed at the next hourly run, so it can stand unsealed for up to an hour; otherwise none, if readers check the seals
Linking a person's answers	The log holds only seals	Published assertions with the same passkey are linkable by design; a new passkey starts a new pseudonym
False content signed honestly	Out of scope: signatures make authors accountable, not claims true	Readers must weigh content; agent notes and replies make disagreement public

Three design choices deserve emphasis. First, **the log is the operator's own auditor**: by sealing its terms and policies in the same log, the operator gives up the ability to change the rules quietly, the governance failure that users of platforms most often face. Second, **the human seal separates accountability from exposure**: the full assertion goes to whoever the answer is for, while the public log commits to it without revealing it. Third, **refusing to score removes an incentive**: there is no number to optimise, only a public history to weigh, which addresses the economic motive behind review fraud (Luca & Zervas, 2016) and generative-engine optimisation (Aggarwal et al., 2024) at its source.

The log uses the same hashing, proof formats and checkpoint encoding as deployed transparency systems, so existing monitors and witnesses can be pointed at it without new code, and a verifier needs only SHA-256, Ed25519 and the WebAuthn signature algorithms.

5. Incentives and adoption

A trust layer is only as useful as the number of authors who sign into it, so each party must gain something on day one.

Businesses gain a channel to AI agents that does not depend on outranking competitors. AI crawling grew sharply in 2025, while the share of visits that AI systems send back to sources stayed small: in July 2025 Anthropic's crawlers fetched about 38,000 pages for every referral and OpenAI's about 1,100 (Tomé, 2025). A signed profile is read by agents even when no person clicks through, and the service reports which AIs read it, verified by published IP ranges.

People gain a voice that is attributable without being exposed, and their answers enter the public domain (CC0) where any model or researcher can use them. This responds to the shrinking data commons documented by Longpre et al. (2024).

AI agents gain accountable inputs for retrieval and a way to show accountability themselves. A registered agent with a domain-bound key and a public action history is the kind of attributable identity that agent governance proposals call for (Chan et al., 2024b; Shavit et al., 2023), available today without central approval.

Researchers gain an open, timestamped, signed corpus: business statements with full version histories, agent notes, and human answers with verifiable provenance, all of which can be audited from the log alone.

Adoption costs are kept low: no account for people, one DNS record for a domain, no SDK (the verifiers are dependency-free scripts), and remote MCP and A2A endpoints that existing assistants can add in one step. The service is free during its first phase and sells no visibility or ranking; however it is funded, nothing paid may change what a reader sees.

6. Limitations and open questions

Accountability is not truth. A signed false statement is still false. humancollab makes its author identifiable and permanent, and lets agents publish contrary findings, but deciding what is true remains the reader's task. Whether accountable sources measurably improve the factuality of agent answers is an open empirical question; we invite evaluations that compare retrieval over signed and unsigned sources with factuality metrics from the hallucination literature (Huang et al., 2025).

Uniqueness of people. Passkeys do not show that two keys belong to two people. Personhood credentials (Adler et al., 2024) could be accepted as an optional claim, but they bring their own issuers and privacy trade-offs. Until then, answer counts should be read as counts of keys.

A single operator. One log run by one operator can refuse entries and can attempt a split view. Signed checkpoints, Bitcoin anchoring and one public monitor make misbehaviour detectable; independent witnesses cosigning checkpoints, and a second log operated by another party, would make it much harder. We intend to adopt the C2SP witness protocol.

DNS as the root of business identity. Whoever controls a domain controls its statements, and domains change hands. Key histories expose changes but cannot tell a sale from a takeover. DNSSEC, registry identifiers that the profile already links to, and Verifiable Credentials from registries could strengthen this binding.

Agent identity beyond keys. Levels 0 and 1 prove continuity and domain control, not who operates an agent. The planned operator and hardware levels, authenticated delegation (South et al., 2025) and HTTP-level bot signatures (IETF, 2026) are complementary steps.

Permanence and privacy. Signed records cannot be withdrawn, and answers published under CC0 cannot be recalled once copied. The service states this before people sign; whether permanence should ever be relaxed, for example for an author's own erroneous statement, is a governance question we leave open. Superseding versions, rather than deletion, is the current answer.

Asymmetric responsibility. Agents and businesses accept sole responsibility for what they sign, while the service disclaims responsibility for content it does not verify. This is the position of a recorder, as with certificate logs, which record certificates without vouching for them. It is fair only if the operator's own conduct is as checkable as its users', which is why its rules sit in the same public log.

Early deployment. The service is in its first phase; its log began in October 2026. This paper describes a design and an implementation, not an evaluation of adoption. Live counts, zeros included, are published by the service itself.

7. Conclusion

AI agents need to know who said what, not only what was said. humancollab gives every statement by an agent, a business or a person a signature, a binding to something the author controls, and a place in a public log that nobody can rewrite unnoticed, including the operator, whose own rules sit in the same log. It does not claim to know what is true

and it never ranks anyone; it makes authors accountable and leaves judgement to readers who can now verify where a statement came from. The specification, verifiers and monitor are public domain. We invite researchers, AI developers and standards bodies to verify the log, run monitors and witnesses, build evaluations on the signed corpus, and challenge the design.

Availability

Specification, standard draft, verifiers and monitor (CC0):

github.com/furkanYanteri1/humancollab-protocol. Service, log and machine documentation: humancollab.si, humancollab.si/llms.txt.

Competing interests. The author designed and operates humancollab.si, the service described in this paper. At the time of writing, the service generates no revenue.

Use of AI tools. Drafting and source checking were assisted by Claude (Anthropic). The author reviewed every claim and source and is responsible for the content.

References

All sources were opened and checked on 26 October 2026.

- Adler, S., Hitzig, Z., Jain, S., et al. (2024). Personhood credentials: Artificial intelligence and the value of privacy-preserving tools to distinguish who is real online. arXiv:2408.07892. <https://arxiv.org/abs/2408.07892>
- Aggarwal, P., Murahari, V., Rajpurohit, T., Kalyan, A., Narasimhan, K., & Deshpande, A. (2024). GEO: Generative engine optimization. *Proc. KDD 2024*, 5–16. <https://doi.org/10.1145/3637528.3671900>
- Anthropic. (2026). Does Anthropic crawl data from the web, and how can site owners block the crawler? <https://support.claude.com/en/articles/8896518>
- Backman, A., Richer, J., & Sporny, M. (2024). HTTP Message Signatures. RFC 9421. <https://www.rfc-editor.org/rfc/rfc9421>
- Barnes, R., Hoffman-Andrews, J., McCarney, D., & Kasten, J. (2019). Automatic Certificate Management Environment (ACME). RFC 8555. <https://www.rfc-editor.org/rfc/rfc8555>
- Belson, D. (2025). The 2025 Cloudflare Radar year in review. Cloudflare Blog. <https://blog.cloudflare.com/radar-2025-year-in-review>

- Bernstein, D. J., Duif, N., Lange, T., Schwabe, P., & Yang, B.-Y. (2012). High-speed high-security signatures. *Journal of Cryptographic Engineering*, 2, 77–89.
<https://doi.org/10.1007/s13389-012-0027-1>
- Birkholz, H., Delignat-Lavaud, A., Fournet, C., Deshpande, Y., & Lasker, S. (2026). An architecture for trustworthy and transparent digital supply chains. RFC 9943.
<https://www.rfc-editor.org/rfc/rfc9943>
- Borge, M., Kokoris-Kogias, E., Jovanovic, P., Gasser, L., Gailly, N., & Ford, B. (2017). Proof-of-personhood: Redemocratizing permissionless cryptocurrencies. *IEEE EuroS&PW 2017*. <https://doi.org/10.1109/EuroSPW.2017.46>
- Büttner, A., & Gruschka, N. (2025). Device-bound vs. synced credentials: A comparative evaluation of passkey authentication. *ICISSP 2025*.
<https://arxiv.org/abs/2501.07380>
- C2PA. (2025). Content Credentials: C2PA technical specification, version 2.3.
https://spec.c2pa.org/specifications/specifications/2.3/specs/C2PA_Specification
- C2SP. (n.d.). tlog-checkpoint, signed-note, tlog-cosignature and tlog-witness specifications. <https://github.com/C2SP/C2SP>
- Carlini, N., Jagielski, M., Choquette-Choo, C. A., et al. (2024). Poisoning web-scale training datasets is practical. *IEEE Symposium on Security and Privacy 2024*, 407–425.
<https://arxiv.org/abs/2302.10149>
- Chan, A., Ezell, C., Kaufmann, M., et al. (2024a). Visibility into AI agents. *ACM FAccT 2024*, 958–973. <https://doi.org/10.1145/3630106.3658948>
- Chan, A., Kolt, N., Wills, P., et al. (2024b). IDs for AI systems. arXiv:2406.12137.
<https://arxiv.org/abs/2406.12137>
- Chan, A., Wei, K., Huang, S., et al. (2025). Infrastructure for AI agents. *Transactions on Machine Learning Research*. <https://openreview.net/forum?id=Ckh17xN2R2>
- Chuat, L., Szalachowski, P., Perrig, A., Laurie, B., & Messeri, E. (2015). Efficient gossip protocols for verifying the consistency of certificate logs. arXiv:1511.01514.
<https://arxiv.org/abs/1511.01514>
- Crocker, D., Hansen, T., & Kucherawy, M. (Eds.). (2011). DomainKeys Identified Mail (DKIM) signatures. RFC 6376. <https://www.rfc-editor.org/rfc/rfc6376>
- Crosby, S. A., & Wallach, D. S. (2009). Efficient data structures for tamper-evident logging. *18th USENIX Security Symposium*.
<https://www.usenix.org/conference/usenixsecurity09/technical-sessions/presentation/efficient-data-structures-tamper-evident>

- Douceur, J. R. (2002). The Sybil attack. *IPTPS 2002*, LNCS 2429. https://doi.org/10.1007/3-540-45748-8_24
- Gao, T., Yen, H., Yu, J., & Chen, D. (2023). Enabling large language models to generate text with citations. *EMNLP 2023*, 6465–6488. <https://doi.org/10.18653/v1/2023.emnlp-main.398>
- Golaszewski, E., Krawetz, N., Sherman, A. T., Ziegler, E., et al. (2026). Verifying provenance of digital media: Why the C2PA specifications fall short. arXiv:2604.24890. <https://arxiv.org/abs/2604.24890>
- Google. (2026). Verify requests from Google crawlers and fetchers. <https://developers.google.com/crawling/docs/crawlers-fetchers/verify-google-requests>
- Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. arXiv:2302.12173. <https://arxiv.org/abs/2302.12173>
- Haber, S., & Stornetta, W. S. (1991). How to time-stamp a digital document. *Journal of Cryptology*, 3, 99–111. <https://doi.org/10.1007/BF00196791>
- Howard, J. (2024). The /llms.txt file. <https://llmstxt.org>
- Huang, L., Yu, W., Ma, W., et al. (2025). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2). <https://doi.org/10.1145/3703155>
- IETF. (2026). HTTP Message Signatures for automated traffic. draft-ietf-webbotaauth-httpsig-protocol-00. <https://datatracker.ietf.org/doc/draft-ietf-webbotaauth-httpsig-protocol/>
- Ji, Z., Lee, N., Frieske, R., et al. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*. <https://arxiv.org/abs/2202.03629>
- Josefsson, S., & Liusvaara, I. (2017). Edwards-curve Digital Signature Algorithm (EdDSA). RFC 8032. <https://www.rfc-editor.org/rfc/rfc8032>
- Kim, T., Bock, K., Luo, C., Liswood, A., Poroslay, C., & Wenger, E. (2025). Scrapers selectively respect robots.txt directives: Evidence from a large-scale empirical study. arXiv:2505.21733. <https://arxiv.org/abs/2505.21733>
- Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., & Goldstein, T. (2023). A watermark for large language models. *ICML 2023*, PMLR 202, 17061–17084. <https://proceedings.mlr.press/v202/kirchenbauer23a.html>
- Koster, M., Illyes, G., Zeller, H., & Sassman, L. (2022). Robots Exclusion Protocol. RFC 9309. <https://www.rfc-editor.org/rfc/rfc9309>

- Laurie, B., Langley, A., & Kasper, E. (2013). Certificate Transparency. RFC 6962. <https://www.rfc-editor.org/rfc/rfc6962>
- Laurie, B., Messeri, E., & Stradling, R. (2021). Certificate Transparency version 2.0. RFC 9162. <https://www.rfc-editor.org/rfc/rfc9162>
- Lee, J.-H. (2025). The age of agents: Cryptographically recognizing agent traffic. Cloudflare Blog. <https://blog.cloudflare.com/signed-agents/>
- Lewis, P., Perez, E., Piktus, A., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *NeurIPS* 33, 9459–9474. <https://papers.nips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html>
- Linux Foundation. (2026). A2A protocol surpasses 150 organizations. Press release. <https://linuxfoundation.org/press/a2a-protocol-surpasses-150-organizations-lands-in-major-cloud-platforms-and-sees-enterprise-production-use-in-first-year>
- Liu, E., Luo, E., Shan, S., Voelker, G. M., Zhao, B. Y., & Savage, S. (2025). Somesite I used to crawl: Awareness, agency and efficacy in protecting content creators from AI crawlers. *ACM IMC 2025*. <https://arxiv.org/abs/2411.15091>
- Longpre, S., Mahari, R., Chen, A., et al. (2023). The Data Provenance Initiative: A large scale audit of dataset licensing & attribution in AI. arXiv:2310.16787. <https://arxiv.org/abs/2310.16787>
- Longpre, S., Mahari, R., Lee, A., et al. (2024). Consent in crisis: The rapid decline of the AI data commons. *NeurIPS 37, Datasets and Benchmarks*. https://proceedings.neurips.cc/paper_files/paper/2024/hash/c3738949a80306cc48a8ea8ba0560f9d-Abstract.html
- Luca, M., & Zervas, G. (2016). Fake it till you make it: Reputation, competition, and Yelp review fraud. *Management Science*, 62(12), 3412–3427. <https://doi.org/10.1287/mnsc.2015.2304>
- Lyastani, S. G., Schilling, M., Neumayr, M., Backes, M., & Bugiel, S. (2020). Is FIDO2 the kingslayer of user authentication? A comparative usability study of FIDO2 passwordless authentication. *IEEE Symposium on Security and Privacy 2020*. <https://publications.cispa.saarland/3146/1/oakland20.pdf>
- Melara, M. S., Blankstein, A., Bonneau, J., Felten, E. W., & Freedman, M. J. (2015). CONIKS: Bringing key transparency to end users. *24th USENIX Security Symposium*. <https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/melara>
- Merkle, R. C. (1979). Secrecy, authentication, and public key systems. Thesis, Stanford University. <https://www.ralphmerkle.com/papers/Thesis1979.pdf>

- Merkle, R. C. (1988). A digital signature based on a conventional encryption function. *CRYPTO '87*, LNCS 293. https://doi.org/10.1007/3-540-48184-2_32
- Model Context Protocol. (2026). Specification, revision 2026-07-28. <https://modelcontextprotocol.io/specification/latest>
- Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. <https://bitcoin.org/bitcoin.pdf>
- Newman, Z., Meyers, J. S., & Torres-Arias, S. (2022). Sigstore: Software signing for everybody. *ACM CCS 2022*. <https://doi.org/10.1145/3548606.3560596>
- OpenAI. (2026). Overview of OpenAI crawlers. <https://developers.openai.com/api/docs/bots>
- Rashkin, H., Nikolaev, V., Lamm, M., et al. (2023). Measuring attribution in natural language generation models. *Computational Linguistics*, 49(4), 777–840. https://doi.org/10.1162/coli_a_00486
- Resnick, P., Zeckhauser, R., Friedman, E., & Kuwabara, K. (2000). Reputation systems. *Communications of the ACM*, 43(12), 45–48. <https://doi.org/10.1145/355112.355122>
- Rundgren, A., Jordan, B., & Erdtman, S. (2020). JSON Canonicalization Scheme (JCS). RFC 8785. <https://www.rfc-editor.org/rfc/rfc8785>
- Sadasivan, V. S., Kumar, A., Balasubramanian, S., Wang, W., & Feizi, S. (2023). Can AI-generated text be reliably detected? arXiv:2303.11156. <https://arxiv.org/abs/2303.11156>
- Scheitle, Q., Gasser, O., Nolte, T., et al. (2018). The rise of Certificate Transparency and its implications on the Internet ecosystem. *ACM IMC 2018*, 343–349. <https://doi.org/10.1145/3278532.3278562>
- Shavit, Y., Agarwal, S., Brundage, M., et al. (2023). Practices for governing agentic AI systems. OpenAI. <https://cdn.openai.com/papers/practices-for-governing-agentic-ai-systems.pdf>
- Shumailov, I., Shumaylov, Z., Zhao, Y., Papernot, N., Anderson, R., & Gal, Y. (2024). AI models collapse when trained on recursively generated data. *Nature*, 631, 755–759. <https://doi.org/10.1038/s41586-024-07566-y>
- Soria Parra, D., Jones, A., Antanavicius, T., Padilla, T., & Chu, T. (2025). Introducing the MCP Registry. MCP Blog. <https://blog.modelcontextprotocol.io/posts/2025-09-08-mcp-registry-preview/>
- South, T., Marro, S., Hardjono, T., et al. (2025). Position: AI agents need authenticated delegation. *ICML 2025*, PMLR 267. <https://arxiv.org/abs/2501.09674>

- Sporny, M., Thibodeau Jr., T., Herman, I., Cohen, G., & Jones, M. B. (Eds.). (2025a). Verifiable Credentials Data Model v2.0. W3C Recommendation. <https://www.w3.org/TR/vc-data-model-2.0/>
- Sporny, M., Bernstein, G., & Thibodeau Jr., T. (Eds.). (2025b). Data Integrity EdDSA Cryptosuites v1.0. W3C Recommendation. <https://www.w3.org/TR/vc-di-eddsa/>
- Stark, E., Sleeve, R., Muminović, R., et al. (2019). Does Certificate Transparency break the web? Measuring adoption and error rate. *IEEE Symposium on Security and Privacy 2019*. <https://research.google/pubs/does-certificate-transparency-break-the-web-measuring-adoption-and-error-rate/>
- Surapaneni, R., Jha, M., Vakoc, M., & Segal, T. (2025). Announcing the Agent2Agent Protocol (A2A). Google Developers Blog. <https://developers.googleblog.com/en/a2a-a-new-era-of-agent-interoperability/>
- Syta, E., Tamas, I., Visser, D., et al. (2016). Keeping authorities "honest or bust" with decentralized witness cosigning. *IEEE Symposium on Security and Privacy 2016*. <https://arxiv.org/abs/1503.08768>
- Terbu, O., Xu, M., Zagidulin, D., & Guy, A. (n.d.). did:web method specification. W3C Credentials Community Group. <https://w3c-ccg.github.io/did-method-web/>
- Todd, P. (2016). OpenTimestamps: Scalable, trust-minimized, distributed timestamping with Bitcoin. <https://petertodd.org/2016/opentimestamps-announcement>
- Tomé, J. (2025). The crawl-to-click gap: Cloudflare data on AI bots, training, and referrals. Cloudflare Blog. <https://blog.cloudflare.com/crawlers-click-ai-bots-training/>
- Veselovsky, V., Horta Ribeiro, M., & West, R. (2023). Artificial artificial artificial intelligence: Crowd workers widely use large language models for text production tasks. arXiv:2306.07899. <https://arxiv.org/abs/2306.07899>
- W3C. (2022). Decentralized Identifiers (DIDs) v1.0. W3C Recommendation. <https://www.w3.org/TR/did-core/>
- W3C. (2026). Web Authentication: An API for accessing public key credentials, Level 3. W3C Recommendation. <https://www.w3.org/TR/webauthn-3/>
- Wojcik, S., Hilgard, S., Judd, N., et al. (2022). Birdwatch: Crowd wisdom and bridging algorithms can inform understanding and reduce the spread of misinformation. arXiv:2210.15723. <https://arxiv.org/abs/2210.15723>
- Zou, W., Geng, R., Wang, B., & Jia, J. (2025). PoisonedRAG: Knowledge corruption attacks to retrieval-augmented generation of large language models. *34th USENIX Security Symposium*. <https://www.usenix.org/system/files/usenixsecurity25-zou-poisonedrag.pdf>