WikiWe: A Protocol for Verifiable Agreements, Portable Reputation, and Durable Records

WikiWe Contributors -- wikiwe.org

uRULEu Institute -- uruleu.org

Working Draft 0.8, July, 2026

Abstract. People form enterprises, communities, partnerships, and project teams across organizational and geographic boundaries, but the records supporting those relationships remain fragmented and fragile. Reputation is held inside platforms that control its presentation and portability. Agreements are stored as documents that software cannot administer reliably. Important records may disappear, become unverifiable, or lose legitimacy when an administrator, service provider, or founding team fails. We propose WikiWe, a human-centric protocol for representing agreements, promises, notices, credentials, contributions, authorizations, and other significant events as signed, content-addressed Verifiable Records underlying portable Reputation Wallets. Secure protocols ensure that both collection and disclosure of Wallet-held reputation remains under the control of each reputation subject. A distributed registry carries cryptographic commitments that permit later verification without publishing the underlying personal information. Wallets and verifiers may use established presentation protocols to request and exchange credentials. WikiWe implementations maintain the record semantics, authorization relationships, transaction binding, and scoped-completeness proofs needed for durable collaborative reputation. A member may disclose a selected credential, or may prove that a disclosure is complete within a stated topic, time period, or combination of both. The Reputation Wallet protocol separates the identity and validity of a record from its storage location, and separates the survival of a record history from permanent authority in a single operator. Signed checkpoints and independently retained replicas allow participants to evaluate competing successor systems after service failure or compromise. WikiWe is intended as an interoperable foundation on which independent Administrators can provide compatible services while members retain their agreements, evidence, and reputation across organizational boundaries.

Status and Scope of This Draft

This paper describes the intended technical architecture of WikiWe, a portable and independently verifiable record system for voluntary collaboration. It is not itself a normative conformance specification, a certification standard, or a statement that every described component has already been implemented. The separation of record identity from storage, authentication from authorization, private payloads from public commitments, and ordinary disclosure from claims of completeness are core architectural invariants. The exact record encoding, cryptographic construction for scoped completeness, transport profiles, and some continuity procedures remain subjects of design, implementation, and testing.

References to external protocols identify mature components that WikiWe may reuse or profile. They do not make every optional feature of those protocols part of WikiWe. Normative WikiWe profiles will need to select precise versions, required options, rejection behavior, and test vectors before implementations can be certified against them.

This paper is intended to guide implementation, invite technical critique, and establish the design direction from which later specifications and conformance profiles will be derived.

1. Introduction

A collaborative project may operate successfully for years and still lose its institutional memory in a day. Its platform operator may fail, a founder may disappear, or an account may be closed. Agreements, contribution histories, credentials, and reputation may remain trapped in several proprietary databases, leaving members unable to prove what they accepted, what they did, or what authority governed the records.

Digital collaboration does not lack communication tools. People can exchange messages, share documents, sign contracts, maintain repositories, transfer money, and publish credentials through many existing services. The challenge is preserving a trustworthy relationship among these activities after they have been divided among applications and placed under the control of different operators. For example, a project member may accept an agreement in one application, perform work in another, receive verification in a third, and accumulate reputation in a fourth. The agreement may later change without a reliable record of which version governed the work. Even where all of the relevant information still exists, another person may have no practical way to determine who issued it, what authorized the issuer, whether the record has been changed, or whether important records were omitted. Finally, much surveillance -- essentially the collection of reputation data without the subject's knowledge and consent -- is both unwanted and unfair.

These problems are usually treated as separate application problems. WikiWe treats them as consequences of one missing layer: a portable and independently verifiable record system for voluntary collaboration. The system must preserve what people agreed to, what they promised, what authority they granted, what they did, and what others credibly recorded about their performance. It must do so without requiring every participant to trust one database operator forever.

What is needed is a protocol in which significant records can be verified independently of the service that displays them, agreements can identify the exact terms that were accepted, and reputation can move with its subject rather than remain the property of a platform. The protocol should support privacy without making verification impossible, and it should permit recovery from operational failure without converting a recovery key into permanent control over a community.

WikiWe addresses this need with three related mechanisms. First, it defines a general Verifiable Record whose identity is bound to canonical content and whose issuer can be authenticated. Second, it associates agreements with record channels that preserve acceptance, notice, performance, amendment, and withdrawal as an intelligible history. Third, it separates portable reputation into private user-controlled records and public cryptographic commitments, so a person can prove selected facts or complete histories within declared limits without exposing everything else.

Subject authorization is the foundational authority for reputation records. A record about a person or group -- i.e., a subject -- should enter that subject's portable reputation only under permission granted by the subject concerned. Organizational roles, issuer credentials, platform policies, and agreement procedures may establish authenticity, context, or evidentiary weight, but they do not substitute for subject authorization.

The protocol does not require a blockchain, a financial token, or universal public disclosure. It can be implemented incrementally with ordinary public-key cryptography, signed files, Git repositories, conventional databases, and content-addressed archives. Nor does it require members to manage cryptographic tools directly. An implementation may hide the machinery behind device passkeys, familiar account interfaces, managed wallets, and optional custody services.

The objective is not to eliminate trust. Collaboration always requires judgment about people, institutions, and evidence. The objective is to prevent trust from being needlessly concentrated in a platform that exclusively controls the records on which everyone else depends.

2. Actors and Roles

WikiWe distinguishes the collaborative relationship from the infrastructure that supports it.

A **Member** is a person or organization participating in a WikiWe relationship. A Member may join several collaborative contexts and may use different service providers without changing the identity of the Member's portable records.

A **Hub** is a purpose-defined community, project, partnership, enterprise, or other collaborative context operating under an agreement or Charter. A Hub supplies the context in which membership, roles, contributions, permissions, and decisions have meaning. It is not necessarily a legal entity and is not itself the hosting platform.

An **Administrator** is a service operator that hosts or supports one or more Hubs. An Administrator may provide accounts, Agreement Channels, financial services, indexes, wallet interfaces, storage, accounting, conflict resolution, and other operational services. A Hub can change Administrators without becoming a different Hub if its agreement and records preserve the necessary continuity. In edge cases a Hub may be, or may transform itself into, a self-administered entity.

An **Access Agent** is a provider selected by a Member to hold or help operate wallet data, keys, recovery material, or disclosure services. The Access Agent is replaceable and does not own the Member's reputation.

An **Issuer** creates a credential or other record under an asserted authority. A **Verifier** requests or evaluates records and proofs. Either role may be performed by a Member, Hub actor, Administrator, certifying body, or software procedure, depending on the governing agreement and authorization records.

A **Certifier** evaluates an identified implementation or service against published conformance requirements. uRULEu Institute, a California nonprofit public benefit corporation recognized as tax-exempt under §501(c)(3) of the Internal Revenue Code, is the proposed neutral steward of the WikiWe Protocol and the proposed certifier of conforming Administrators and implementations. It is not intended to administer ordinary Hubs or hold their private member records.

The distinction between Hub and Administrator is fundamental: a Hub is the collaborative context; an Administrator is an operator providing infrastructure to that context.

Figure 1 — Actors and Institutional Boundaries

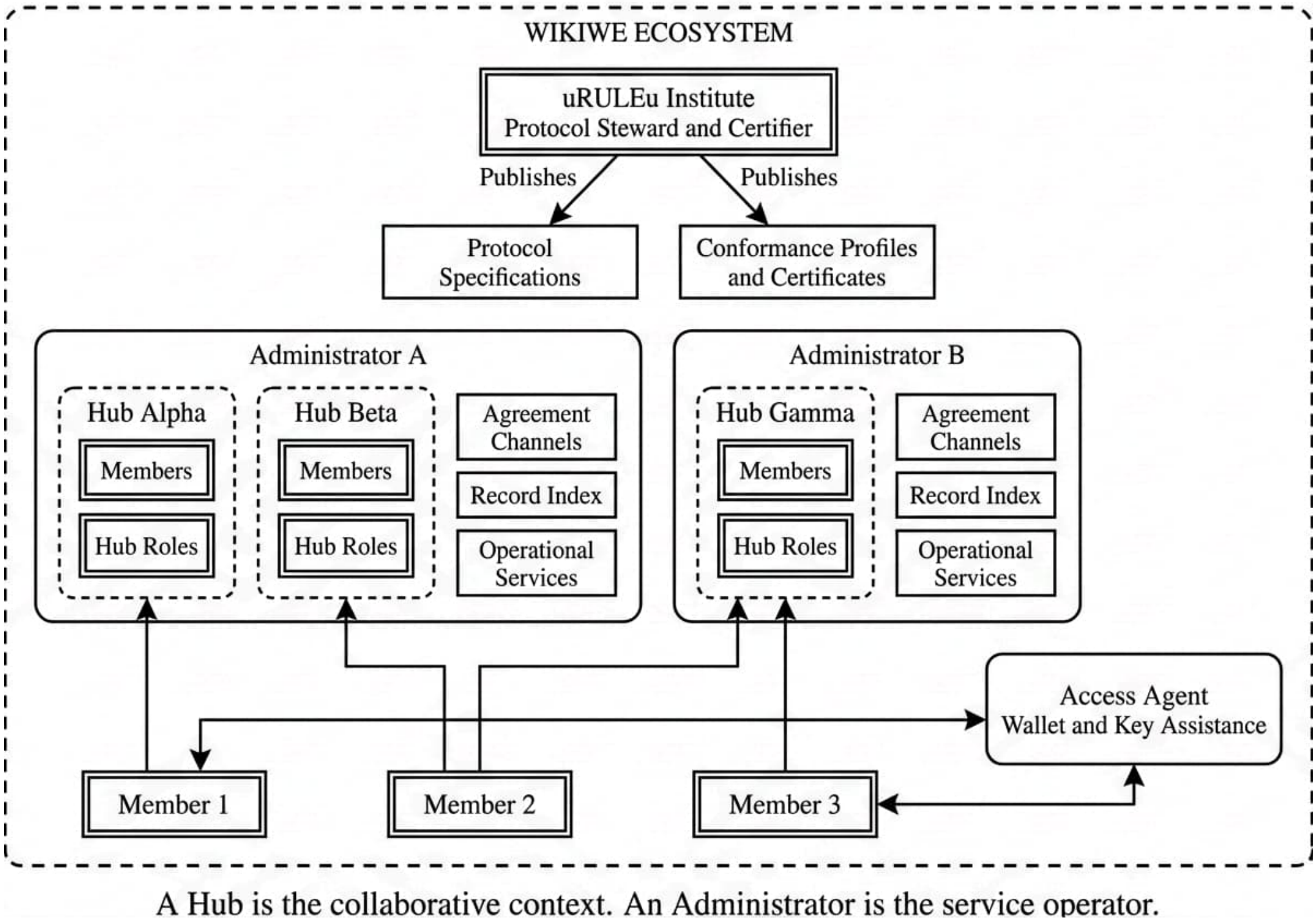

Figure 1. WikiWe actors and institutional boundaries, showing Members participating in Hubs, Administrators hosting multiple Hubs, optional Access Agents serving Members, and uRULEu Institute acting as protocol steward and certifier rather than Hub operator.

3. A Running Example

Consider a member contributing to an open-source software Hub. The Hub publishes a human-readable contribution agreement and a machine-readable manifest identifying the exact version, the Agreement Channel, the roles permitted to verify work, and the procedures governing recognition of contributions.

The contributor accepts that version. The acceptance becomes a Verifiable Record linked to the agreement identifier and retained in the channel. The Administrator carries and indexes the record, but the authority of the acceptance comes from the contributor's authenticated act and the agreement context, not from the Administrator's database.

The contributor later completes a task. A Hub manager whose role record authorizes verification issues a task-completion record. A deterministic contribution procedure may use that verified record to calculate a points or benefit award. The result identifies the agreement version, input records, procedure version, and responsible authority so another implementation can reproduce it.

The Hub may then issue a contribution credential through, for example, OpenID4VCI. The credential enters the contributor's Reputation Wallet, while a salted commitment to the corresponding reputation event enters a broadly replicated commitment registry. The private payload and identity linkage remain in erasable, user-controlled storage.

Years later, another project asks for the contributor's complete software-development history during the preceding ten years. The request can be expressed through OpenID4VP and a WikiWe scope statement defining the topic and time period. The wallet returns the responsive credentials and a proof that no committed event within that declared scope was omitted. Earlier personal events and unrelated professional records remain private.

If the original Administrator subsequently fails, the contributor and Hub retain signed records and checkpoints. A replacement Administrator can import the records, verify their ancestry and authorization, and continue service without becoming the source of the earlier history.

This example is illustrative rather than normative. The following sections explain the record, agreement, reputation, transport, and continuity mechanisms separately.

4. Verifiable Records

We define a Verifiable Record as a canonical data object whose content, issuer, context, and authorization can be checked by a recipient. A record may describe an agreement, an acceptance, a notice, a contribution, a personal promise, a social promise, a credential, or another event that matters to a continuing relationship. Its identity is derived from, or cryptographically bound to, its canonical content, so copying the record to another database, repository, wallet, or archive does not create a different record.

A useful record contains enough information to answer four questions. The first is what happened or was asserted. The second is who issued the record. The third is the context in which the record has meaning. The fourth is what authorized the issuer to act. These questions are distinct in some but not all contexts. For example, in a collaborative project context, a valid signature may prove that a particular key issued a task award, but the signature alone does not prove that the key holder was authorized to award work in that project. For counterexample, in a promise context, the key holder is always authorized to issue promises that bind no other person.

A record therefore identifies an issuer and one or more subjects, and it refers to the agreement, channel, project, Hub, credential stream, promise, or other context in which it operates. It may state when it was issued, when it became effective, and when it expires. It may contain an intelligible payload, an encrypted payload, a reference to information stored elsewhere, or only a cryptographic commitment to private information. It may also refer to earlier records, allowing records to form an amendment history, a revocation history, an ordered event stream, or a graph of evidentiary dependencies.

The same structure can represent several kinds of commitment without pretending that they are identical. An agreement acceptance records consent to a particular agreement version. A personal promise records a commitment made by an identified person, whether privately or to named beneficiaries. A social promise records a promise offered to a qualified class of persons in a reciprocal context. A pledge records a promise made to a defined group. Role, delegation, and capability records describe authority that one person or process may exercise for another. Notices, contributions, verifications, settlements, corrections, and revocations record events that occur under those commitments.

A Verifiable Record is not made true merely because it is signed. The signature authenticates the issuer, while the authorization reference explains why the issuer was entitled to make the record. A project manager may be authorized by a role record that is limited to one Hub. A certifying body may issue only credentials within a defined professional field. A counterparty may provide reputation feedback only under a capability previously granted by the subject. Verification follows these references until the recipient has enough evidence to decide whether the action was permitted.

Records must be encoded deterministically before they are hashed or signed. Equivalent records cannot be allowed to produce different identifiers merely because fields were reordered, whitespace changed, or one serializer formatted a number differently. Various schemes may be suitable. For example, the JSON Canonicalization Scheme defines one published method for producing invariant JSON suitable for cryptographic hashing and signing [1]. CBOR likewise defines deterministic encoding requirements for applications that need every implementation to generate the same byte representation [2].

A protocol profile may use canonical JSON or deterministic CBOR for a signed envelope. Human- and agent-readable agreement packages, explanatory material, schemas, and related knowledge may also be distributed as an Open Knowledge Format bundle. OKF presently defines a draft, project-specific format for a portable directory of Markdown documents with YAML metadata that can be read by people and software without specialized tooling [3]. It is a candidate bundle format that may be useful rather than a foundational dependency. The signed Verifiable Record itself still requires an unambiguous canonical byte representation.

The record identifies the canonicalization, hash, and signature profiles used to verify it. This permits cryptographic migration without pretending that one algorithm will remain suitable forever. It also distinguishes an issuer signature from the signature of a container. A record may be individually signed by its issuer, included in a signed Merkle batch, accepted into a signed Git commit, and later named in a signed checkpoint. These facts strengthen one another, but they do not make the same claim. The issuer signature authenticates the record itself, while a repository or checkpoint signature establishes that an operator recognized the record as part of a larger history.

KERI and Authentic Chained Data Containers provide closely related models that should inform the eventual WikiWe record specification. KERI uses self-certifying identifiers and append-only key-event histories to make control, rotation, delegation, and recovery independently verifiable [21]. ACDC defines self-addressing containers that can be linked by cryptographic digest into verifiable graphs, with explicit structures for provenance, selective disclosure, schemas, and rules [22]. These systems address more than serialization: they bind statements to the evolving authority of their issuers. WikiWe will avoid freezing a new record envelope before completing a compatibility analysis with KERI and ACDC. A normative WikiWe Record might remain an independent format, become an ACDC profile, or support a deterministic projection into and out of ACDC.

The controlling requirement is that WikiWe preserve its own semantics for agreements, personal and social promises, authorization, scoped reputation completeness, and continuity without needlessly duplicating mature control-provenance machinery.

Cryptographic provenance must also be distinguished from factual truth. C2PA makes the same distinction for media provenance: its validation procedures establish whether assertions and manifests are well formed and cryptographically valid, not whether the asserted content is true [37]. WikiWe likewise proves who made a record, what authorized the act, how the record relates to prior records, and whether it has been altered. The recipient must still evaluate whether the underlying assertion accurately describes reality.

Figure 2 — Logical Structure of a Verifiable Record

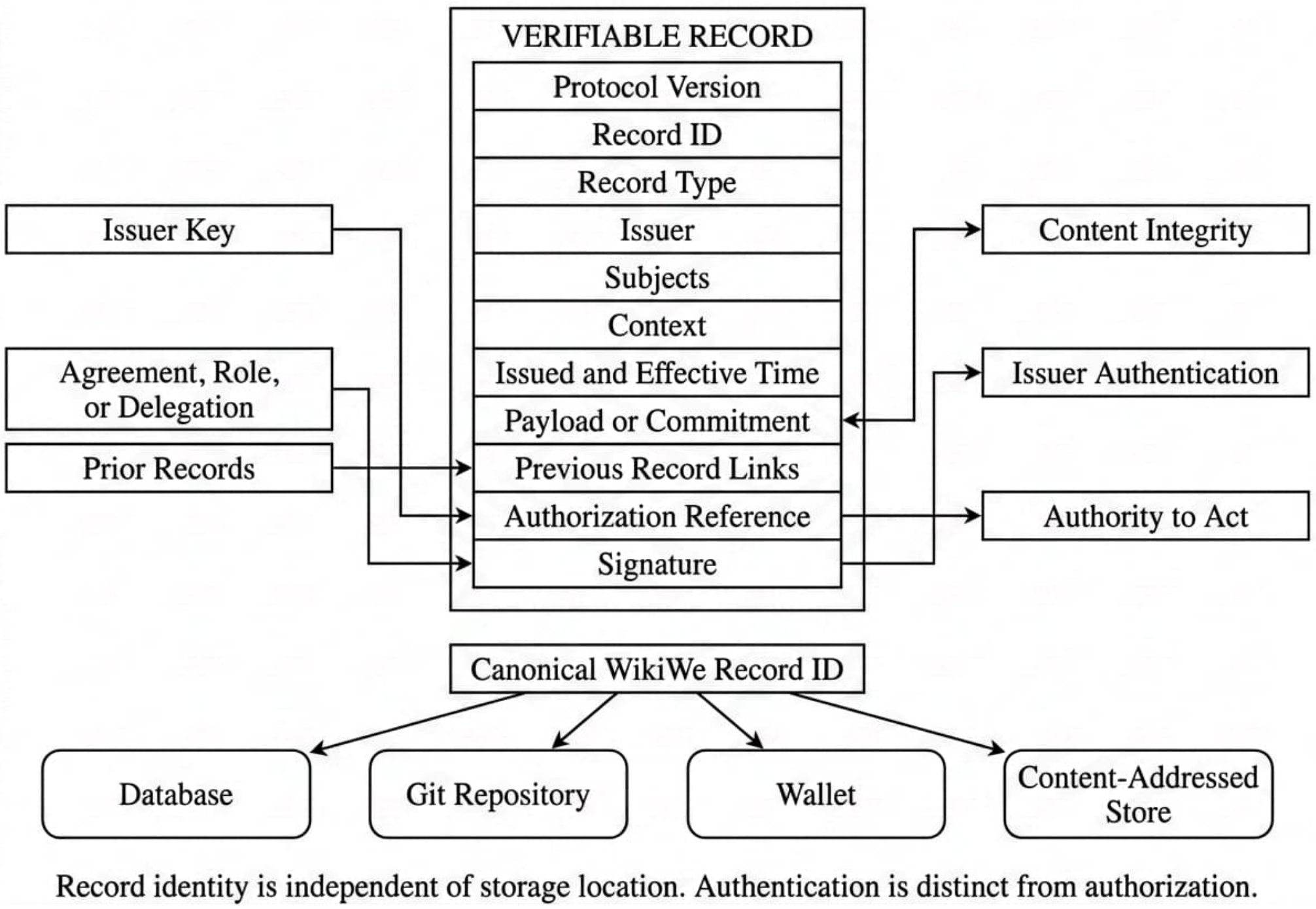

Figure 2. The logical structure of a Verifiable Record, showing the issuer, subject, context, payload or commitment, authorization reference, prior-record links, and signature.

5. Agreement Channels

An agreement becomes difficult to administer when the accepted document is separated from the notices, permissions, amendments, and performance records that give it practical effect. WikiWe associates a continuing agreement with an Agreement Channel. The channel is an addressed and identifiable stream of records through which the parties establish membership, give notice, record authority, document performance, and preserve changes to the relationship.

An Agreement Channel may be implemented in various ways, for example email lists, messaging groups, Nostr relay streams, shared git repositories, bulletin boards, forums, or any other non-transitory communication channel. A messaging application may provide one interface to it, but the channel is defined by its stable identity and record rules rather than by a particular service. It may be implemented as an append-only event stream, a path in a repository, a replicated collection of signed records, or a server-mediated interface that provides independently retainable exports. The same channel can be projected into a conversation view, a membership roster, a list of obligations, a contribution history, or a decision log without changing the underlying records.

The agreement begins with human-readable prose. Software should not require members to infer their obligations from data fields alone. The prose is accompanied by an Agreement Manifest that identifies the exact version of the text, any incorporated documents, the channel identity, the conditions for joining, and the parameters or procedure modules that software must administer. A person who accepts the agreement signs or otherwise authenticates an acceptance record that refers to the immutable identifier of that version. Recording that a person accepted "the current terms" is insufficient if the record does not establish what those terms were. Cryptographic authentication supplies evidence of assent, but whether a legally enforceable agreement has been formed remains subject to the applicable law, required disclosures, attribution rules, and circumstances of the transaction.

After acceptance, the channel records events arising under the relationship. A notice identifies its issuer, content, context, intended recipient class, issuance time, and authorization. Delivery and acknowledgment may be recorded separately, because publication, delivery, and acknowledgment are different events. A role appointment identifies the authority granted and its limits. A contribution record identifies the work or other performance to which it refers. An amendment identifies both the former version and the replacement, together with the procedure that made the change effective.

Withdrawal or termination is recorded in the same manner. Ending the relationship may extinguish future authority, but it does not erase the fact that the person accepted an earlier agreement or performed obligations while it was effective. Historical records continue to refer to the version that governed them, allowing a later verifier to reconstruct the applicable terms without applying a newer agreement retroactively.

The ordinary agreement sequence is simple. The parties first publish a canonical agreement and manifest. Each participant then creates an acceptance record referring to that exact version. The channel subsequently receives authorized records of notice, performance, amendment, and other material events. When the agreement changes or ends, the new record refers to the history it modifies. The result is not merely a signed document, but an evidentiary relationship that can survive changes in software and service providers.

An Agreement Channel may use an existing transport without adopting that transport as its source of authority. Nostr provides one relevant relay-based model. Its base protocol represents publications as signed, content-addressed events distributed through one or more relays and retrieved by filtered subscriptions [23]. NIP-29 adds relay-based groups, membership and moderation events, restricted publication, and the possibility of moving or forking a group across relays [24]. These features are useful for public or semipublic notices, promises, checkpoint announcements, and channel events that benefit from real-time replication.

For example: A conforming implementation may publish a WikiWe Record inside a Nostr event or refer to the record by its canonical identifier. The Nostr event is then a publication envelope or channel projection. Relay acceptance shows only that a relay accepted the event; it does not prove that the underlying WikiWe act was authorized. Nor does a relay query prove that it returned the complete evidentiary history. WikiWe continues to rely on record ancestry, participant-held replicas, reconciliation among sources, and signed checkpoints for those conclusions.

Private channel communication may use different transports. DIDComm Messaging defines message-level authenticated and encrypted communication over transport-independent connections rooted in decentralized identifiers [25]. The Trust Spanning Protocol is emerging as a related model for authentic and optionally confidential exchange across several forms of public-key-based identifiers, with explicit attention to metadata privacy [26]. Nostr, DIDComm, TSP, ordinary HTTPS, and future transports may all carry channel records. The accepted agreement and its authorization history, rather than the transport, determine the identity and authority of the channel.

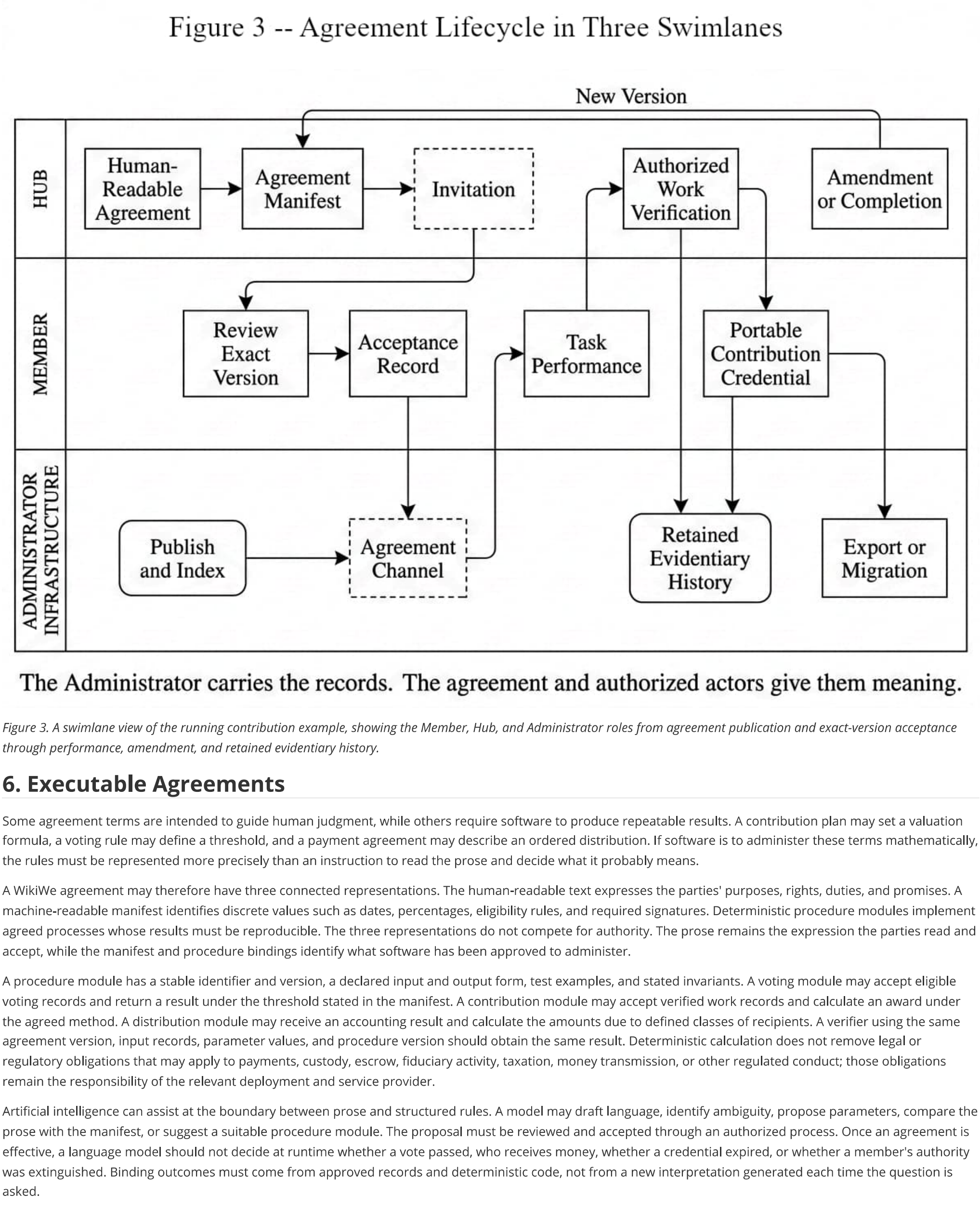

Figure 3. A swimlane view of the running contribution example, showing the Member, Hub, and Administrator roles from agreement publication and exact-version acceptance through performance, amendment, and retained evidentiary history.

6. Executable Agreements

Some agreement terms are intended to guide human judgment, while others require software to produce repeatable results. A contribution plan may set a valuation formula, a voting rule may define a threshold, and a payment agreement may describe an ordered distribution. If software is to administer these terms mathematically, the rules must be represented more precisely than an instruction to read the prose and decide what it probably means.

A WikiWe agreement may therefore have three connected representations. The human-readable text expresses the parties' purposes, rights, duties, and promises. A machine-readable manifest identifies discrete values such as dates, percentages, eligibility rules, and required signatures. Deterministic procedure modules implement agreed processes whose results must be reproducible. The three representations do not compete for authority. The prose remains the expression the parties read and accept, while the manifest and procedure bindings identify what software has been approved to administer.

A procedure module has a stable identifier and version, a declared input and output form, test examples, and stated invariants. A voting module may accept eligible voting records and return a result under the threshold stated in the manifest. A contribution module may accept verified work records and calculate an award under the agreed method. A distribution module may receive an accounting result and calculate the amounts due to defined classes of recipients. A verifier using the same agreement version, input records, parameter values, and procedure version should obtain the same result. Deterministic calculation does not remove legal or regulatory obligations that may apply to payments, custody, escrow, fiduciary activity, taxation, money transmission, or other regulated conduct; those obligations remain the responsibility of the relevant deployment and service provider.

Artificial intelligence can assist at the boundary between prose and structured rules. A model may draft language, identify ambiguity, propose parameters, compare the prose with the manifest, or suggest a suitable procedure module. The proposal must be reviewed and accepted through an authorized process. Once an agreement is effective, a language model should not decide at runtime whether a vote passed, who receives money, whether a credential expired, or whether a member's authority was extinguished. Binding outcomes must come from approved records and deterministic code, not from a new interpretation generated each time the question is asked.

Not every decision can or should be reduced to an algorithm. An agreement may authorize a manager, reviewer, or neutral to exercise judgment. In that case the software records who made the decision, the authority under which it was made, the inputs considered, and the resulting action. The system should preserve discretion where discretion was agreed, but it should not disguise a human judgment as an objective calculation.

IEEE 7012-2025 provides an important adjacent model for machine-readable agreements. It describes a process in which an individual can proffer machine-readable personal privacy requirements to a service provider, form an agreement when the terms are accepted, and retain a record of the agreement on both sides [27]. WikiWe extends this general pattern beyond privacy terms to personal promises, social promises, reciprocal agreements, collaborative enterprises, contribution procedures, and delegated authority. The existence of an established standard in the narrower privacy setting supports interoperability work rather than a proprietary WikiWe syntax for every class of machine-readable term.

This separation also makes software evolution safer. A procedure can be tested, replaced, or rolled back without changing the historical agreement records that invoked an earlier version. If an error is found, a correction can explain the defect and identify a superseding result. The original result remains part of the evidence rather than disappearing from a mutable database.

7. Reputation Wallets

Outside of WikiWe, online reputation commonly belongs to the platform that displays it. The platform determines which events count, how they are summarized, whether the subject can export them, and whether another service can verify them. A member may spend years building a record that can be lost or made worthless for multiple reasons outside of the member's control.

WikiWe puts ownership and control of reputation with its subject, subject to one condition: any omission within the scope of an accepted request is apparent in the response. User's can't cherry-pick the output of their Reputation Wallets, so in almost all cases each user is motivated to earn a good reputation and respond positively to fair critical feedback. A subject-controlled Reputation Wallet can hold a complete social profile: credentials, completed commitments, contribution records, personal and social promises, endorsements, feedback, dispute outcomes, corrections, revocations, and anything else reflective of the bearer's character, skills, and values.

The Reputation Wallet may be software on the member's device, an encrypted portable bundle, or a service operated by an access agent selected by the member. However implemented, it must have these essential characteristics: the subject must be empowered to retain and move their own reputation records, rather than being forced to ask the issuing platform to describe the subject's own history. Additionally, the Reputation Wallet itself must enforce its own credibility by preventing user distortion of their reputation history. That enforced credibility is what makes a portable reputation worth relying on: it rewards consistent good conduct with a reputation that is both credible and portable.

A universal score is neither required nor preferred. Reputation is contextual. The facts relevant to selecting a software contributor may be different from those relevant to selecting a mediator, custodian, or financial steward. A verifier should be able to inspect the records or proofs relevant to a proposed relationship without receiving an unexplained number calculated by a rigid rule or central authority.

Subject's Authorization

A person’s portable reputation should not be assembled merely because another person, organization, platform, or government claims authority to evaluate that person. The foundational authority for a WikiWe reputation record comes from the person or people whom the record describes.

Before an issuer may create a record for inclusion in a subject’s portable reputation, the subject grants a Subject Authorization Record. The authorization identifies the subject, the permitted issuer or class of issuers, the relationship or transaction concerned, the kinds of records that may be created, the permitted purpose, and any limits on duration, number of uses, retention, publication, or disclosure. The authorization may permit one specific record, a defined series of records, or records arising under an identified agreement.

For example, a contributor may authorize a project manager to record whether a particular task was completed. A buyer and seller may authorize one another to issue feedback concerning a defined transaction. A professional may authorize a certifying body to place the result of a stated examination in the professional’s wallet. None of these permissions gives the issuer a general right to build a broader dossier about the subject.

The resulting reputation record refers to the Subject Authorization Record under which it was issued. Verification establishes both that the issuer created the record and that the subject permitted that issuer to create that kind of record. An organizational role, government license, employment relationship, or platform rule may provide additional context or evidentiary weight, but is not required to create a subject-authorized reputation record and does not replace the subject’s authorization.

General acceptance of platform terms should not silently operate as unlimited permission to create reputation records. Higher-assurance profiles should require the authorization to be specific, intelligible, separately presented, and no broader than necessary for the stated relationship. Refusal to grant unrelated reputation permissions should not prevent participation unless those permissions are genuinely necessary to the transaction.

Groups involve multiple use cases. One concerns entering a record pertaining to a group into its members' personal wallets. Each member controls whether the record enters that member’s portable reputation. A common event may therefore produce several subject-specific reputation records or references, each supported by the relevant subject’s authorization. One subject’s consent does not authorize creation of a portable reputation record about another. Another use case is certifying that the group has properly authorized creation of the group reputational record - for example, proving that the current group president is duly elected and signed the agreement that authorized creation of the reputation record.

A subject may revoke an authorization for future use. Revocation does not erase a record that was validly created while the authorization was effective, but it prevents additional records from being issued under that authority. Corrections, disputes, withdrawals, and changes in status are represented by later linked records rather than by silently rewriting the historical record.

Protocols

Protocols and models for supplying the mechanics of Reputation Wallets already exist. For example, The W3C Verifiable Credentials Data Model defines an issuer-holder-verifier ecosystem for portable, machine-verifiable claims [4]. The related Data Integrity specification defines cryptographic proof mechanisms for authenticity and integrity of credentials and similar constrained digital documents [5]. WikiWe can use any compatible credential and presentation format while extending the surrounding architecture to agreement records, promises, authorization chains, private reputation histories, and continuity evidence.

Selective-disclosure standards also provide a useful implementation precedent. SD-JWT replaces undisclosed JSON values with salted digests and later supplies the selected cleartext values and salts to a verifier, allowing the verifier to authenticate disclosed claims without learning the others [6]. WikiWe uses the same general hiding principle for private reputation events, while treating scoped completeness as an additional and distinct proof requirement.

WikiWe's reputation architecture separates information into three layers. The first is a broadly replicated commitment registry. It contains a pseudonymous stream identifier, a cryptographic commitment to an event, limited timing information, and enough batch or issuer information to verify later inclusion. The registry is intended to prove that an event existed and occupied a place in a history, not to reveal what the event says. Certificate Transparency provides a mature example of a public log in which Merkle inclusion proofs establish that an entry is present and consistency proofs establish that a later tree preserves the earlier append-only history [7]. A WikiWe registry would apply the transparency-log pattern to content-free commitments rather than public certificates.

The second layer contains the actual reputation event. This may be, for example, a credential, a work verification, a feedback record, or a settlement outcome; essentially, a record of a social interaction verified by a peer community member without imprint by any centralized authority. The payload remains in the subject's wallet, in encrypted storage controlled by the subject, or with a chosen access agent. Even encrypted payloads should not automatically be placed into an immutable public store, because encryption may later fail, keys may leak, and widely replicated copies may be impossible to erase.

The third layer contains the linkage between the pseudonymous stream and the subject's ordinary identity, together with the keys used to disclose or control the records. This linkage remains under the subject's control. The public registry need not know that a particular stream belongs to a named person. The wallet establishes that relationship when the subject chooses to present it.

A reputation event includes a high-entropy nonce before its commitment is published. Without the nonce, an observer could guess a low-entropy fact, such as "adult," "credential revoked," or "completed task 42," hash the guess, and compare it with the registry. The nonce remains with the private record and is disclosed only when the subject wants another person to verify the commitment. This is the same reason that selective-disclosure formats salt hidden claims before publishing their digests [6].

Events may also refer to the preceding event in a stream. This creates an ordered history in which later records cryptographically depend on earlier ones. Corrections and revocations are added to the history instead of silently replacing the original event. A correction identifies the earlier record and the superseding interpretation. A revocation identifies the credential or authority withdrawn and the time at which the withdrawal became effective. The history therefore preserves both the original claim and the later change.

Ordered histories can be organized into topical threads corresponding to identified aspects of a subject's activity — that is, to the scope of a reputational inquiry. To report a reputation limited to a declared scope, the wallet gathers the records belonging to the responsive threads. An event may belong to more than one thread, so the threads form an overlapping graph rather than disjoint lists. This thread structure is also one of the mechanisms available for scoped-completeness proofs (Section 8).

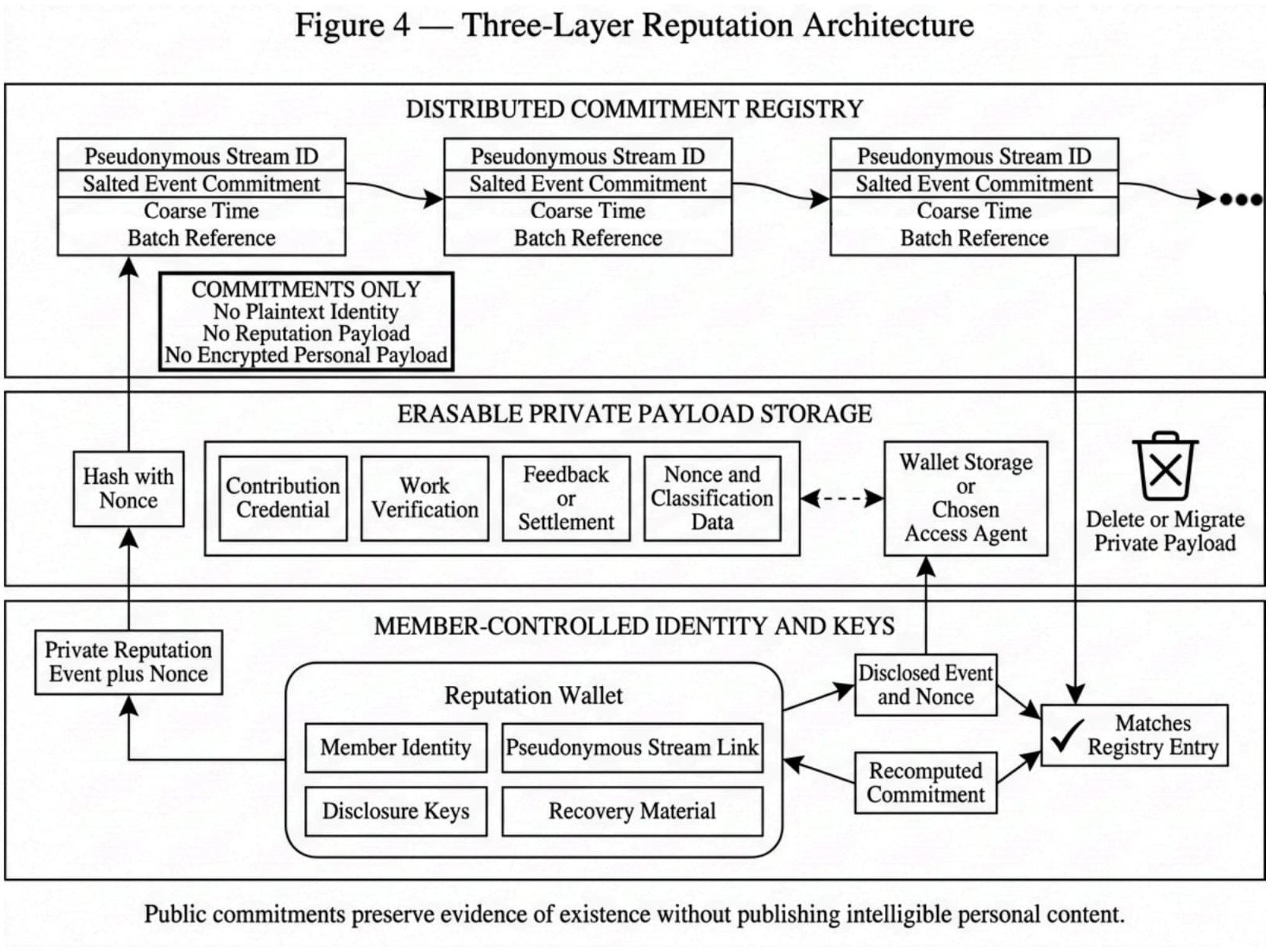

Figure 4. The three-layer reputation architecture: a distributed commitment registry, private or encrypted reputation payloads, and wallet-held identity linkage and keys.

8. Selective Disclosure and Scoped Completeness

Privacy requires a person to disclose less than an entire lifetime history. Trust sometimes requires proof that nothing relevant was withheld. WikiWe supports both by distinguishing a selected disclosure from a claim of completeness.

A member may present a single credential and claim only that the credential is authentic. The verifier checks its issuer, signature, authorization, registry inclusion if applicable, and revocation status. The presentation makes no claim that other records do not exist. This is an existence claim, and it is sufficient for many ordinary interactions.

A stronger presentation claims that all records within a defined scope have been disclosed. The scope may be limited by time, by topic, or by both. "My complete professional reputation for the past ten years" is time-delimited and may also be limited to professional activity. "All of my records concerning financial stewardship" is topic-delimited. "All records concerning patient-data security work from 2022 through 2026" combines both limits. Such a bounded completeness claim allows a person to prove a relevant history while keeping unrelated events, including the indiscretions of youth, outside the representation.

A complete-history claim is the broadest form of the same idea. It asserts that all responsive records in the relevant history have been presented without a narrower topic or time exclusion. It does not receive special treatment merely because its scope is broad. Like every other completeness claim, it requires proof.

The three claims can therefore be stated as follows:

1. **An existence claim** proves that a disclosed record is authentic, but says nothing about whether other records exist.
2. **A bounded completeness claim** proves that the disclosed set contains every record responsive to a stated scope. The scope must be sufficiently precise to test, and may be defined by time, topic, or both.
3. **A complete-history claim** proves completeness over the full relevant history. It is a bounded completeness claim whose declared scope contains no narrower relevant exclusion.

The proof depends on records being classifiable against the declared scope. An event may carry private or selectively disclosed classification data identifying its professional domain, project, credential class, effective period, issuer class, or other relevant category. The classification is bound to the event commitment even when the public registry does not reveal it. A verifier can therefore confirm not only that the disclosed events are authentic, but also that no committed event matching the agreed classification and period was omitted (i.e., "completeness").

Erasure of a private payload interacts with completeness in a specific way. Deleting a payload or destroying its key removes the intelligible record, but the corresponding commitment remains in the replicated registry. A later completeness proof over a scope containing that commitment will therefore show that a responsive event existed even though its content can no longer be disclosed. This is intentional: erasure produces a visible unavailable-record state within a claimed scope rather than a silent absence. A subject retains the right to erase content; the subject does not thereby acquire the ability to represent an erased history as an

empty one.

Several proof structures may contribute to confirming both authenticity and completeness. Separate hash-linked substreams can organize events by topic. Merkle structures can prove inclusion and append-only consistency [7]. Universal accumulators can provide compact membership and nonmembership witnesses, including zero-knowledge nonmembership proofs [8]. More advanced constructions may combine these primitives to establish that all responsive events were included without revealing the classifications of unrelated events. None of these publications, by itself, specifies the scoped-completeness protocol proposed here; that construction will require a precise commitment scheme, classification model, and threat analysis. The first WikiWe implementation need not solve every private set-proof problem, but it must preserve the architectural distinction between an unsupported statement of completeness and an objectively verifiable one.

The scope itself is part of the signed presentation. A wallet cannot describe a disclosure as "complete" while leaving the boundary implicit. The verifier must be able to see whether the claim covers one credential, a ten-year period, a professional field, or an entire history. Selective disclosure remains freely available; proof is required when completeness is asserted.

Figure 5 -- Selective Disclosure and Authenticated Scoped Completeness

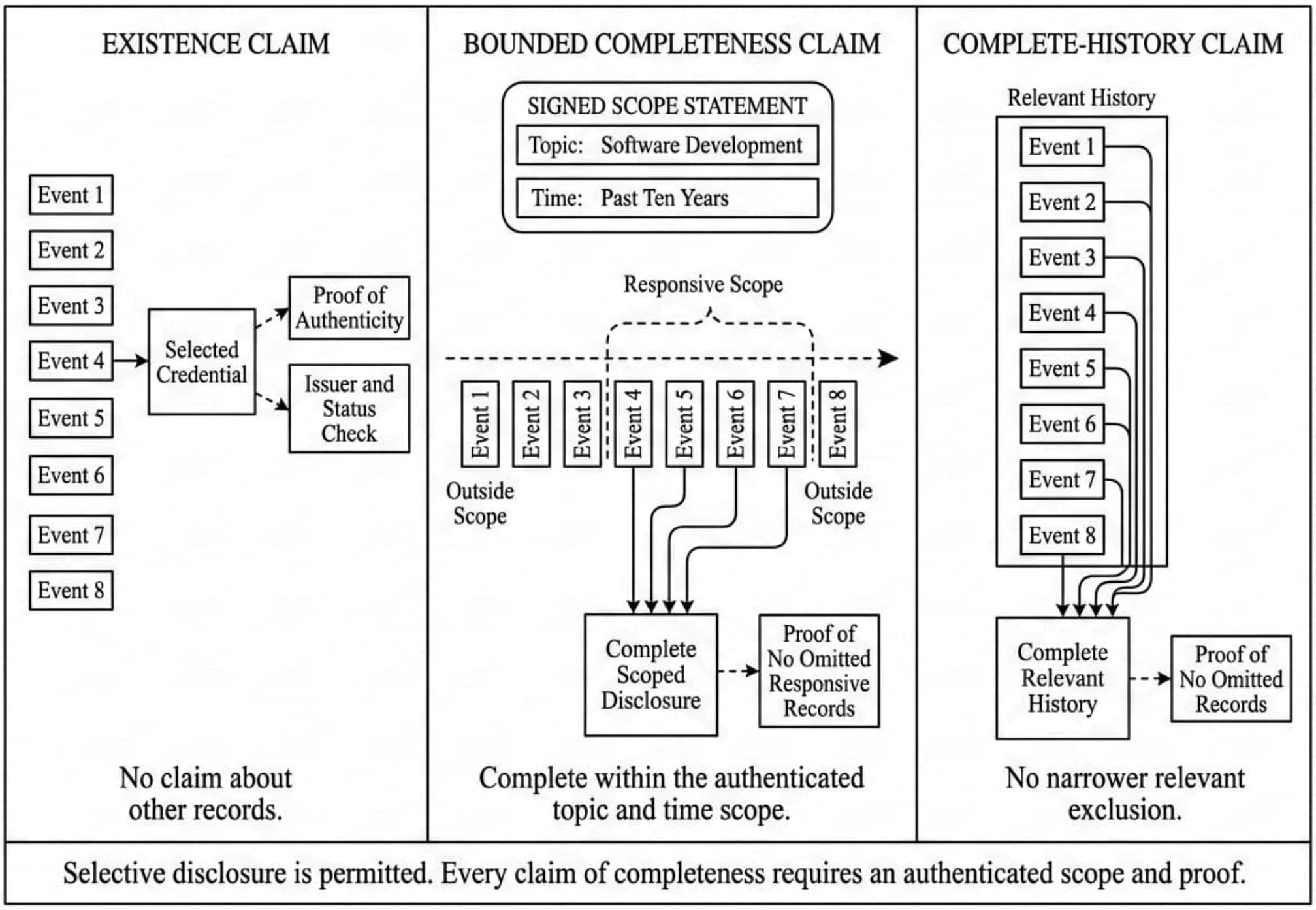

Figure 5. Three reputation presentations: one selected credential, a complete time- and topic-bounded history, and a complete relevant history.

9. Credential Issuance and Presentation Interoperability

To the extent feasible, WikiWe will use existing protocols to deliver credentials into wallets and handle credential requests from verifiers. For example, OpenID for Verifiable Credential Issuance 1.0, or OpenID4VCI, defines an OAuth-protected and credential-format-neutral API through which an authorized issuer can deliver credentials to a wallet [18]. OpenID for Verifiable Presentations 1.0, or OpenID4VP, defines the complementary protocol through which a verifier requests and receives presentations [14]. Together they provide a mature issuance-and-presentation cycle on which WikiWe can build.

In the issuance direction, a Hub, Administrator, certifier, agreement procedure, or authorized counterparty acts as a credential issuer. A completed task may produce a contribution credential; a role appointment may produce an authorization credential; a settlement procedure may produce a status credential; and a feedback capability may authorize a counterparty to issue a reputation event. OpenID4VCI can carry the resulting credential into the member's wallet, while the WikiWe authorization chain establishes why that issuer was permitted to create it.

In a WikiWe exchange, the member acts as the holder, the Reputation Wallet acts as the OpenID4VP wallet, and an Administrator, Hub, counterparty, or application acts as the verifier. The credentials presented may describe professional qualifications, completed commitments, contributions, roles, authorizations, or other reputation events. OpenID4VP transports the request and response; the WikiWe records and profiles determine what the presented information means.

The same-device flow applies when the member interacts with the verifier and wallet on one device. A browser or application sends an authorization request to the wallet, the wallet identifies matching credentials, and the member approves the presentation. A cross-device flow applies when, for example, a verifier displays a QR code on one device and the member responds through a wallet on another. In both cases the protocol separates the verifier's request, the member's consent, and the wallet's response. This separation is important because possession of a credential does not imply consent to disclose it, and authentication of the member does not by itself authorize an unrelated transaction.

OpenID4VP includes the Digital Credentials Query Language, or DCQL, through which a verifier describes the credential formats, credential types, combinations of credentials, and individual claims it wants the wallet to present [14]. A WikiWe verifier can use DCQL to request only the claims needed for a particular purpose. For example, a prospective Hub may request proof of a current software-security credential and selected records of relevant project work rather than requesting the member's entire wallet.

DCQL describes what the verifier is asking the wallet to return, but it does not by itself prove that the returned set is complete within a claimed topic or period. WikiWe therefore distinguishes the **presentation query** from the **completeness scope**. The query identifies acceptable credentials and claims. A separately authenticated scope identifies the topic, time interval, or combination of both for which completeness is asserted. The wallet then returns the responsive presentations together with a WikiWe completeness proof showing that no committed record falling within that scope was omitted. A presentation containing one selected credential requires no

such proof unless the holder represents the disclosure as complete.

OpenID4VP is credential-format neutral. A presentation can carry credentials conforming to the W3C Verifiable Credentials model, ISO mobile-document formats, or selective-disclosure JWT credentials, and the protocol can be profiled for additional formats [14]. WikiWe should preserve this flexibility. Its conformance profiles should define the semantic requirements for WikiWe credentials and proofs without requiring every implementation to use one universal credential container.

The protocol also provides a transaction-data mechanism for binding a presentation to a particular transaction or authorization request [14]. WikiWe should use this capability whenever a credential presentation supports a consequential act. An agreement acceptance should be bound to the exact agreement version and acceptance operation. A delegation should be bound to the authority being granted. A feedback capability should be bound to the relationship and permitted use. A payment, benefit allocation, settlement, or other deterministic procedure should not treat a reusable identity presentation as consent to a different transaction.

A valid credential is not enough to make a presentation exchange secure. The wallet must also determine who is requesting the information and whether the response is bound to the intended session. A WikiWe presentation profile should require fresh nonces, audience or origin binding, replay protection, appropriate holder binding, validation of verifier metadata, and encryption where the presentation contains sensitive information. Error responses should reveal as little as practical about whether the wallet contains a requested credential. These requirements protect the exchange surrounding the record; they do not replace verification of the record's issuer, authorization, status, or history.

The W3C Digital Credentials API offers an emerging browser- and operating-system-mediated interface for both issuance and presentation [19]. It can reduce dependence on custom URI schemes, give the user agent a consistent role in consent and origin binding, and allow several credential formats and exchange protocols to share one application interface. Because the API remains a W3C Working Draft, WikiWe should treat it as an integration surface to monitor and profile rather than as a mandatory dependency.

The OpenID4VC High Assurance Interoperability Profile demonstrates how a flexible collection of base specifications can be narrowed into a testable combination of issuance, presentation, credential formats, cryptographic choices, and browser integration [20]. uRULEu should use this profiling method as an institutional model: select only the options needed for a WikiWe use case, state them normatively, publish test vectors, and certify observable conformance rather than claiming compatibility with every option in each underlying standard.

uRULEu may therefore publish a **WikiWe Credential Exchange Profile** based on OpenID4VCI and OpenID4VP. The profile would select supported request and response modes, credential formats, holder-binding methods, verifier-identification methods, and encryption requirements. It would define WikiWe credential and topic identifiers, the representation of a declared completeness scope, the object carrying the corresponding proof, and the transaction bindings required for agreement acceptance and delegated actions. Conformance testing would verify both valid exchanges and the rejection of replayed requests, mismatched audiences, unsupported formats, invalid holder proofs, and false completeness claims.

This division keeps the protocol focused. OpenID4VCI and OpenID4VP supply standardized credential issuance and wallet-to-verifier exchange. WikiWe supplies the durable records, agreement context, authorization chains, subject-controlled reputation semantics, scoped completeness, and continuity mechanisms that give the exchanged credentials their collaborative meaning.

10. Feedback, Delegation, and Custody

A signature does not give its holder an unrestricted right to write into another person's reputation. Feedback must arise from authority granted by the subject or from an agreement the subject accepted. WikiWe represents that authority as a scoped capability.

A **feedback capability** identifies the permitted issuer, the subject stream, the relationship or transaction to which the feedback may refer, the kinds of events that may be issued, the deadline or expiration, and any limit on the number of uses. It may also bind confidentiality terms, a permitted purpose, and a correction or dispute procedure. A member might authorize a project counterparty to issue one performance record concerning a particular engagement within thirty days after completion. The same capability would not authorize general commentary about the subject or feedback about unrelated work. Macaroons demonstrate a practical authorization design in which chained caveats confine by whom, where, when, and for what purpose delegated authority may be exercised [9].

The resulting reputation event refers to the capability as its authorization basis. A verifier can therefore distinguish feedback that was merely published about a person from feedback that was issued under a permission the person granted as part of a defined relationship. The permission may be attenuable, meaning that a delegate can narrow its scope or duration but cannot broaden it.

Capabilities can also represent **delegated authority** over data and actions. A member may authorize an agent to disclose a defined credential set, manage a wallet during incapacity, or act for a limited purpose and period. Because the delegation is a record, another person can verify both the grant and any later revocation.

The durable WikiWe grant should be distinguished from the protocol and artifact used to exercise it. A Verifiable Record can preserve the meaning, scope, provenance, and revocation history of a delegation. A capability token such as a macaroon can convey narrowly attenuated authority to a service. GNAP provides a standardized protocol for negotiating a grant of authority to software and returning the artifacts and subject information needed to exercise that grant [28]. A WikiWe implementation may use GNAP for live authorization while retaining the corresponding agreement and delegation records as durable evidence.

The protocol does not require every member to keep all keys personally. Some members will choose full self-custody, using device-backed keys and retaining their own encrypted wallet. Others will prefer assisted recovery, in which recovery authority is divided among independent devices or trusted people so that no one participant can take control alone. Still others will choose a **custodial access agent** that provides familiar account recovery and stores the wallet on the member's behalf. Some may choose a combination of these approaches.

An access agent should publish a machine-readable description of its custody model, recovery process, export format, retention and deletion behavior, supported proof methods, and service locations. The agent is a replaceable service provider, not the permanent owner of the member's identity. A member should be able to export the wallet and move to another compatible agent without abandoning the records or changing their identities.

11. Storage and Replication

A WikiWe record is not a database row, a Git commit, an IPFS object, or a blockchain transaction. These are possible carriers. The protocol distinguishes the identity of the logical record from the place where a copy is stored.

This separation prevents several common errors. A valid record may be unavailable because no current host serves it. An available copy may be invalid because its signature or authorization fails. A complete replica may preserve history without possessing authority to issue the next record. A content-addressed object may prove that the retrieved bytes match its identifier without proving that the object is the latest authorized state.

Conventional databases remain useful. They provide sessions, search, indexing, permissions, queues, and fast user interfaces. For durable record classes, however, the database should not be the only evidence. A member who exports a record should be able to verify it without asking the database operator whether it is genuine.

Git is a practical initial carrier for many WikiWe records because it provides a content-addressed object store and commit objects that refer to predecessor commits, trees, and file contents [10]. A repository can collect agreement records, notices, contribution histories, or checkpoint manifests in a form that participants can clone and inspect. The Git commit signature records that an operator accepted the included records into a history, while issuer signatures inside the records preserve their independent meaning.

Content-addressed networks may broaden availability by allowing objects to be retrieved according to their content identifiers. IPFS, for example, describes a peer-to-peer, content-addressed block store organized as a Merkle directed acyclic graph [11]. This does not remove the need for replication policy or authority rules. An object on such a network may disappear if nobody retains it, and the network does not decide who may publish the next valid agreement record or checkpoint.

Replication should therefore follow the nature of the information. Public protocol specifications and public agreement forms may be replicated without restriction. Commitment registries are designed for broad replication because they contain no intelligible reputation payload. A Hub's agreement and operational history may be replicated among authorized members. Private reputation payloads remain in wallets or selected custodial stores. Identity linkage and keys remain under member control. Confidential dispute records and regulated accounting records require narrower retention rules.

This is a more useful meaning of decentralization than copying every datum everywhere. Verification, custody, availability, and succession are distributed according to the purpose and sensitivity of each record.

The local-first software model supplies a useful design discipline for participant custody. Local-first applications treat the user's device and local copy as a primary place of work, remain useful during disconnection, synchronize among collaborators, and seek to preserve access independent of a particular service provider [29]. WikiWe need not require every application to be fully local-first, but durable record classes should remain exportable, independently verifiable, and usable enough to support migration and recovery.

Solid is one possible carrier for user-controlled payloads. Its protocol separates applications from permissioned data stored in user-selected data stores, allowing a person to change applications or storage providers without necessarily changing the underlying resources [30]. A Solid Pod could hold wallet-associated records or encrypted reputation payloads, but Solid access control does not establish WikiWe record validity, authorization, or completeness. It is a storage and access option, not the evidentiary protocol itself.

Portable bundles allow records to move among these carriers. A bundle can include canonical records, referenced schemas, verification material, disclosure manifests, and checkpoint evidence. Where the necessary verification dependencies are included, the bundle can be checked offline and later imported into another compatible service.

12. Checkpoints and Continuity

Replication preserves information, but it does not by itself preserve authority. If an Administrator disappears, several replicas may hold the complete history while no shared rule identifies who is entitled to continue it. If an Administrator remains online after its systems or keys have been compromised, the most visible service may no longer be the most trustworthy one.

WikiWe addresses this problem with signed checkpoints and evidence-based successor selection. A checkpoint is a periodic manifest naming the current roots or heads of important record collections. It may identify Git commit heads, Merkle roots, channel heads, registry batches, recognized signing keys, and the preceding checkpoint. Each checkpoint is signed and linked to the one before it, forming a compact history of the states recognized by its issuer. Hash-linked timestamping was developed to make later alteration of digital-document history detectable [12], while modern transparency logs use signed tree heads and consistency proofs to support efficient auditing of an append-only sequence [7]. WikiWe checkpoints adapt these established techniques to independently retained records of an Administrator's observed state.

KERI offers a related and potentially reusable model for continuity of identifier control. Its key-event log records inception, rotation, delegation, and recovery as an append-only history that verifiers can evaluate without treating the currently presented key as timeless authority [21]. WikiWe checkpoints operate at a broader institutional level, but successor evaluation should be compatible with verifiable key-event histories rather than inventing an unrelated key-rotation mechanism.

The IETF SCITT architecture supplies another useful building block. SCITT defines signed statements, transparency services, registration receipts, and verifiable append-only structures for auditability across digital supply chains [31]. A public WikiWe commitment or checkpoint service could adopt compatible statement and receipt formats where appropriate. Private agreement content and reputation payloads would remain outside the transparency service; only material designed for public commitment or audit would be registered.

The checkpoints are copied into participant-accessible replicas. A member who maintains a mirror therefore holds more than a backup. The member holds evidence of the latest state observed before a disruption. A successor candidate cannot quietly omit that state without failing the member's local continuity test.

The continuity sequence is as follows:

1. The operating Administrator or channel authority periodically signs and publishes a checkpoint naming the current record roots.
2. Members, Hubs, and independent mirrors retain the checkpoint together with enough history to test ancestry.
3. If the service fails or appears compromised, one or more successor candidates publish proposals identifying the checkpoint history they preserve, the service locations they offer, and the authority or key continuity they claim.
4. Each participant tests whether a candidate contains the latest checkpoint or record head that the participant previously retained, and examines the signatures, authorization records, and explanations for any divergence.
5. The participant chooses which continuation to recognize and re-points the participant's software or mirror accordingly.

Ordinary restoration does not require this process. If a server fails while the recognized operators and keys remain available, service can be restored from replicas and backups. The successor mechanism is needed when authority itself is uncertain.

A candidate may demonstrate continuity by using keys that authenticated a long prior history, by presenting an authorized succession record, by showing signatures from recognized roles or participants, and by preserving independently retained checkpoints. No single item is necessarily conclusive. An established key may have been compromised, while a new key may be legitimate if a valid rotation or recovery record explains it.

An offline or threshold-held continuity key can simplify an orderly succession. Its signature is strong evidence that a recognized recovery procedure was followed. It should not be treated as an irrevocable source of authority capable of compelling every participant to accept a successor that omits known records or violates the applicable agreement.

This is not a proof-of-work chain and does not attempt to force every voluntary association into one universal consensus history. Participants make an informed fork choice based on the records they hold and the agreements they accepted. The ultimate check on technical or institutional capture is the ability to retain the evidence and follow the continuation judged legitimate.

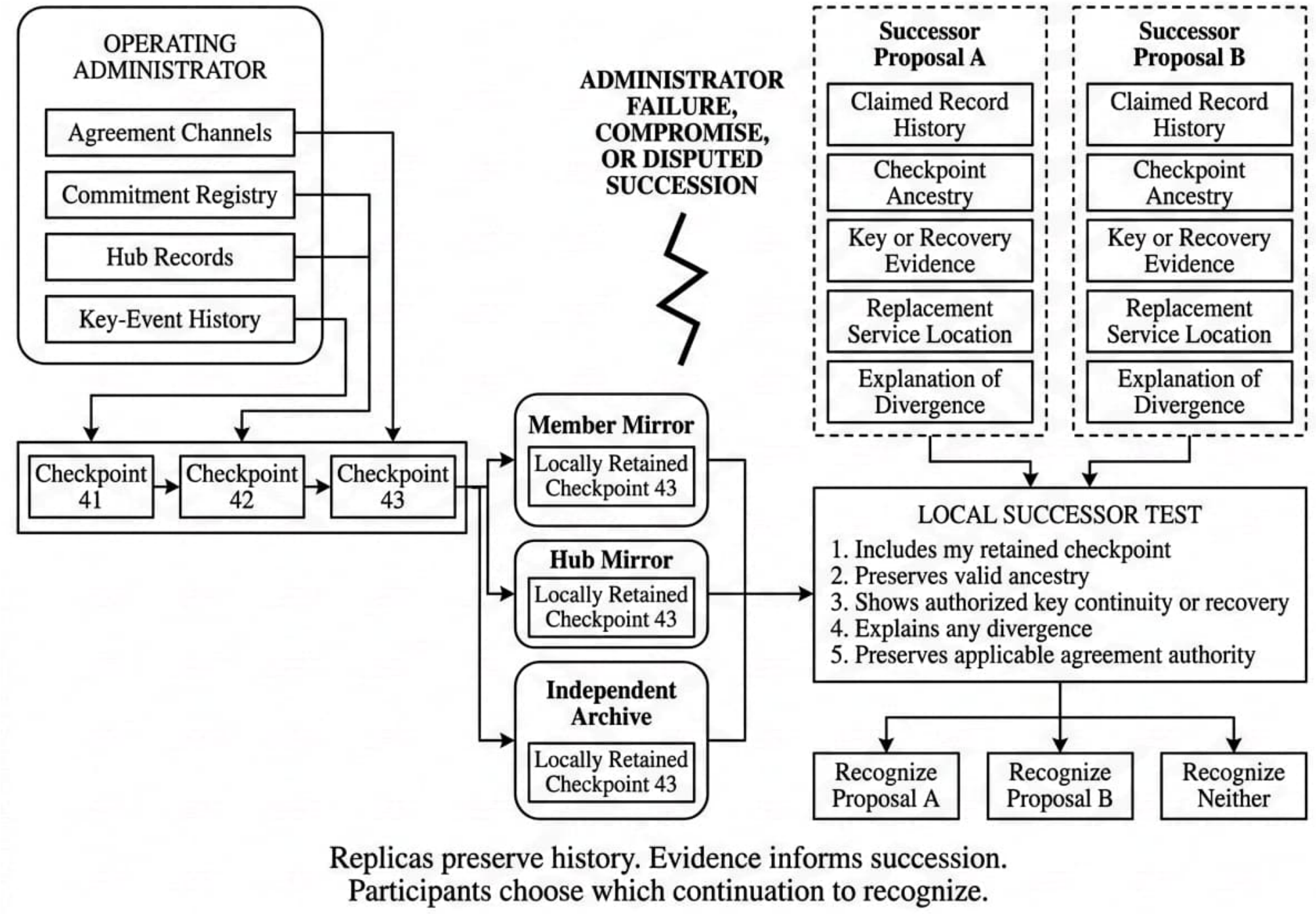

Figure 6. Signed checkpoints replicated to members, followed by service failure, competing successor proposals, local ancestry tests, and participant selection.

13. Interoperable Administrators

WikiWe is intended to support independent Administrators that provide compatible services rather than one permanent global operator. Administrators may differ in jurisdiction, interfaces, hosting, fees, custody options, and supported applications. They interoperate by agreeing on the technical meaning of the records they exchange.

Common profiles are required for canonical record encoding, issuer identifiers, agreement manifests, channel records, delegation, credentials, revocation, disclosure proofs, checkpoints, and wallet export. An Administrator need not implement every application, but it must preserve the meaning of the profiles it claims to support. Existing standards can supply part of this vocabulary. W3C Decentralized Identifiers provide one model for identifiers whose controllers can prove control without depending on a central identity provider or certificate authority [13], while the Verifiable Credentials model supplies interoperable roles and data structures for credentials and presentations [4]. WikiWe need not require DIDs for every identifier, but compatible identifier and credential profiles can reduce unnecessary invention.

A member should be able to present a record issued through one Administrator to a verifier using another without requiring the first Administrator's live database. The verifier may need the issuer's public verification material, relevant authorization records, revocation information, and any agreement context. These dependencies can themselves be represented and distributed as records.

Interoperability does not require every Administrator to trust every other Administrator equally. A verifier may recognize certain issuers, agreement communities, or certification authorities for a particular purpose. One Hub may accept a professional credential issued elsewhere while declining an unrelated reputation summary. Trust remains contextual, but the evidence is expressed in a common form.

Interoperable syntax does not by itself establish which issuers, verifiers, or Administrators should be trusted for a particular purpose. OpenID Federation defines signed entity statements, trust anchors, trust chains, metadata, and policy constraints for multilateral federations [32]. It can help an Administrator determine whether another entity belongs to a recognized federation and whether its published metadata conforms to the federation's policies.

Some relationships call for a narrower authorization lookup rather than a full federation chain. The Trust Registry Query Protocol provides a lightweight, read-only means of asking an authoritative registry whether an entity holds a specified authorization under a stated governance framework [33]. TRAIN supplies an alternative model in which signed trust lists and their discovery are anchored through DNSSEC [34]. WikiWe should permit several trust-establishment profiles rather than create one universal registry of approved issuers.

External organizational credentials can also enter the authorization chain. The vLEI ecosystem provides verifiable credentials for legal entities and for natural persons acting in official or engagement-context roles on their behalf [35]. A WikiWe Administrator could accept such a credential as evidence that a signer acts for a corporation or nonprofit, while still applying the agreement-specific authority required for the act at issue.

The Decentralized Trust Graph work and its OpenVTC implementation explore portable graphs of verifiable relationships among people, organizations, devices, communities, and software agents [36]. This work remains emergent, but it overlaps with WikiWe's linked authorization and reputation records closely enough to warrant continued coordination. WikiWe should remain capable of exchanging or projecting relationship evidence into such graphs without making an unfinished external model a core dependency.

The network can therefore remain structurally modest. Records and proofs move among members, Hubs, wallets, and Administrators. Services may appear, compete, cooperate, or disappear. A member who changes service providers carries the relevant history rather than starting again with an empty account.

The relationship among these technologies is layered rather than competitive. WikiWe defines the semantics of agreements, promises, authorization, reputation, scoped completeness, and continuity. Credential protocols, identifier systems, channel transports, authorization mechanisms, trust registries, and storage systems supply specialized services around those semantics. A conforming implementation can replace one carrier or transport without changing the identity or meaning of the underlying WikiWe Record.

Figure 7 -- WikiWe Protocol Composition

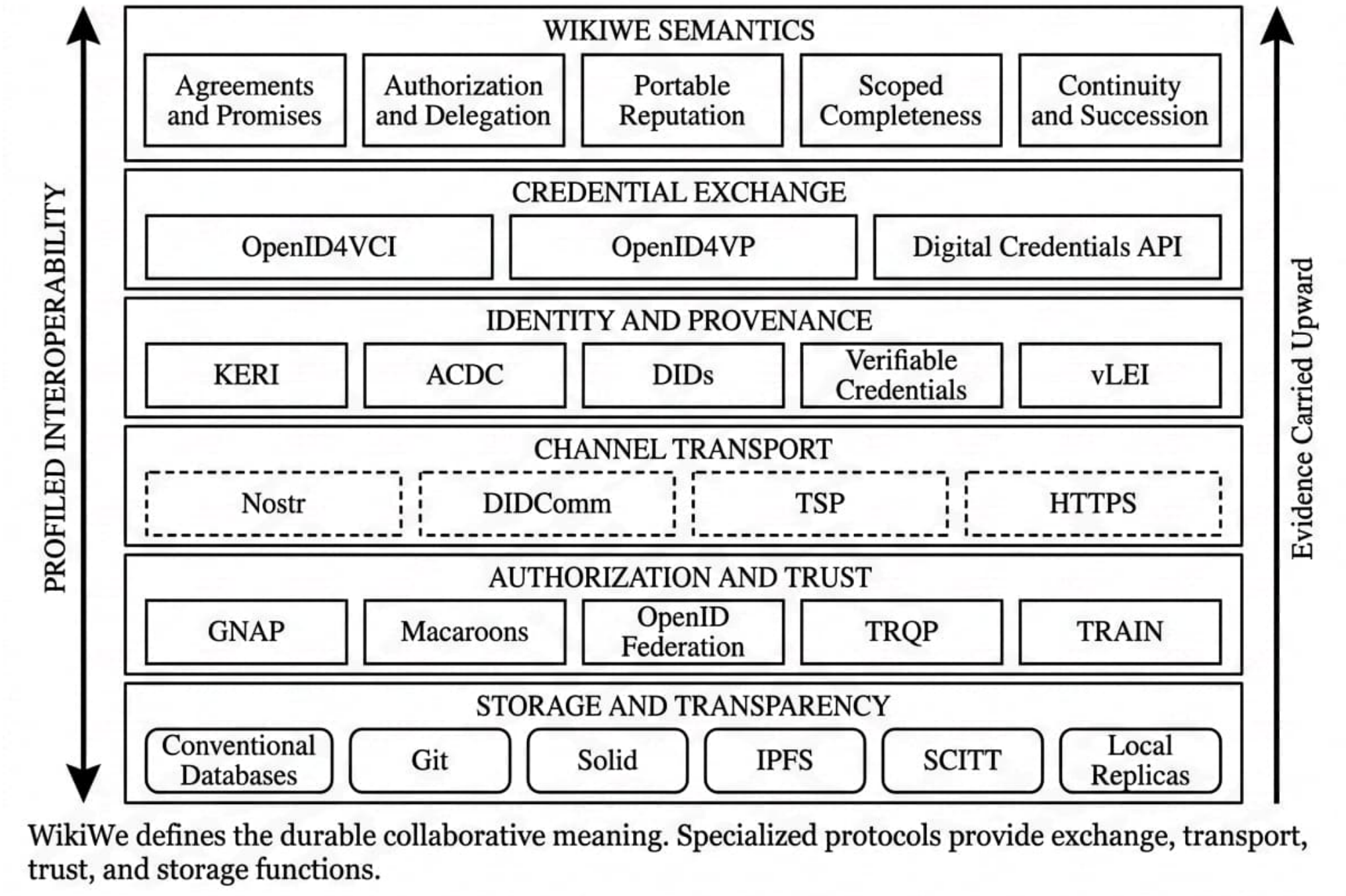

Figure 7. Protocol composition, showing WikiWe semantics above specialized layers for credential exchange, identity and provenance, channel transport, authorization and trust, and storage and transparency.

14. Privacy and Security

WikiWe assumes that failures may be accidental, negligent, or adversarial. The principal threats and corresponding architectural responses are summarized below.

Threat or adversary	Relevant capability	Primary architectural response
Compromised or malicious Administrator	Suppresses, reorders, or selectively presents records	Participant-held records, independent signatures, checkpoints, and successor comparison
Dishonest or unauthorized issuer	Creates false or out-of-scope records	Separate verification of authentication, authorization, context, and revocation
Subject and issuer collusion	Misclassifies a responsive event to evade scoped disclosure	Bound classifications, standardized topic profiles, issuer accountability, and auditable proof rules
Overreaching verifier	Requests excessive credentials or reuses a presentation	Explicit scope, user consent, transaction binding, audience binding, and replay protection
Network observer	Correlates registry writes or presentation traffic	Minimal metadata, batching, coarse timing, encrypted transport, and optional stream rotation
Compromised Access Agent	Discloses, withholds, or loses wallet material	Portable exports, custody disclosure, recovery options, agent replacement, and limited authority
Dishonest successor	Omits known history or claims unsupported continuity	Locally retained checkpoints, ancestry tests, key-event histories, and participant choice
Coercive verifier	Demands overbroad completeness claims as a condition of employment, housing, or participation	Membership-gated verification, reciprocal verifier reputation exposure, minimum-necessary norms, and prohibited-use terms
Holder making a false completeness claim	Selectively presents favorable records as complete	Authenticated scope statements and cryptographic proof of no omitted responsive records

The protocol's privacy model depends on separating public evidence from private meaning. Public commitment and checkpoint layers are intended to reject plaintext personal information, direct identity linkage, encrypted personal payloads, readily guessable unsalted facts, and unnecessary stable correlation identifiers. A commitment registry can be broadly replicated only if its entries reveal no intelligible reputation content without material held elsewhere. This property must be enforced by the schema and write path, not by convention alone.

Private payloads should remain in stores that support deletion, migration, and retention controls. Destroying a key or deleting an encrypted payload may make the intelligible information unavailable, but it does not rewrite commitments or checkpoints already replicated elsewhere. Nor does this paper claim that a commitment can never constitute personal data under applicable law. Each deployment remains responsible for determining and satisfying its legal deletion, retention, and data-protection obligations.

Even a content-free registry can leak metadata. Event timing, frequency, issuer relationships, and continuity of a pseudonymous stream may reveal patterns. Batching several commitments under one signed root, using coarse timestamps, minimizing public fields, and permitting stream rotation can reduce this leakage. They cannot guarantee that every correlation is impossible.

Canonicalization is another security boundary. Canonicalization means converting a structured WikiWe record into one—and only one—authorized byte representation before calculating its identifier, hashing it, or signing it. If two systems produce different bytes from the same apparent record, a signer and verifier may believe they are handling the same content when they are not. Each record must identify its canonicalization profile, and implementations should display the human meaning of the exact content being accepted.

Authorization must be checked at the time relevant to the event. A person may retain a valid signing key after losing the role that once permitted its use. A delegation may expire, be exhausted after one use, or be revoked. Verification therefore follows the authorization record and its later changes, and avoids treating possession of a key as permanent permission.

Capabilities must also resist replay. A feedback license intended for one transaction should not be reusable for another. The capability binds its subject, purpose, context, duration, and use limit. Where single use is required, the system records consumption so a second event can be rejected or identified as unauthorized.

Scoped completeness introduces a classification risk. A dishonest issuer or subject might label a responsive event under an unrelated category so that it falls outside a later proof. Higher-assurance profiles can reduce this risk by using standardized topic identifiers, issuer-signed classifications, multiple applicable categories, and proofs that bind the classification to the original event commitment. No technical classification system can eliminate every dispute about meaning, but it can prevent a classification from being changed silently after the fact.

Completeness proofs create a social risk that no cryptographic mechanism can eliminate: a verifier with market or institutional power may demand a broader completeness claim than a transaction requires. WikiWe narrows this risk structurally. Reputation records and proofs are exchanged only through WikiWe channels among parties bound by the applicable agreements. An entity that has not joined a Hub has no channel through which to request or receive them, and joining subjects the would-be verifier to the same reputation feedback it seeks from others. A verifier who demands overbroad disclosure therefore does so under an accepted agreement — subject to minimum-necessary request norms, the governing agreement's conflict-resolution procedures, and its own discoverable reputation. The residual risk is member-mediated: a member may be pressured to export records for an outside party. Purpose limitations and prohibited-use terms travel with the disclosure itself, and deployments in regulated contexts such as employment screening remain subject to applicable law.

Wallet custody creates its own risks. A self-custody user may lose keys and records. A custodial agent may disclose, lose, or withhold them. Implementations should provide tested exports, clear recovery choices, and explicit descriptions of the selected custody model. Divided recovery authority and portable agent formats reduce dependence on one custodian, but they do not make custody risk disappear.

Finally, two parties may present incompatible histories. Signed checkpoints, independently retained records, ancestry tests, and transparent succession proposals make the split visible. WikiWe cannot guarantee that every participant will interpret the evidence alike. Its purpose is to ensure that the disagreement occurs over inspectable evidence rather than over whatever one operator's database currently chooses to display.

15. Stewardship and Conformance

WikiWe separates the technical protocol from the operation of any one Administrator. uRULEu Institute is intended to serve as the neutral steward of the protocol, maintain versioned specifications and conformance profiles, coordinate public review, and operate a certification program open to qualified implementations and service providers on published terms.

The white paper is not the certification standard. Certification should apply to an identified legal provider, implementation or service, release, protocol version, and set of conformance profiles. A certificate should state its scope and exclusions and should not imply that every security, privacy, legal, or operational characteristic of the product has been evaluated.

A conformance profile must define mandatory behavior, supported options, canonical test vectors, required rejection behavior, interoperability expectations, and renewal conditions. The certification program should include documented evaluation, recusal for conflicts of interest, equal access, time-limited certification, retesting after material changes, appeal, suspension, and revocation. These procedures and the certification-mark license belong in separate public program documents.

The protocol specifications, reference code, and conformance tests should be released under clear public licenses with a transparent contribution and patent policy before substantial outside contributions are solicited. The precise licenses and decision process are not established by this paper and should be published by uRULEu in companion governance and intellectual-property documents.

Ricardiam DAO, LLC (https://ricardiam.com) serves as the initial reference Administrator and proving ground out of which the WikiWe protocol evolved. Its implementation may inform the specification, but it does not possess privileged protocol authority or receive a different certification standard from unrelated Administrators.

Ricardiam is owned solely by supporters of its open-source, WikiWe-compliant platform, via Depublica, the sole member of the LLC. Depublica is a Wyoming corporation with share ownership restricted to those who contribute to the development, promotion, and operation of Ricardiam as proved by its WikiWe-compliant reputation system. The owners have designed their platform to survive Ricardiam as a legal entity if needed, and to encourage competition for users among administrators adhering to WikiWe.

For later Administrators, the commercial incentives are straightforward: in exchange for giving up reputational captivity as a competitive moat, the Administrator gains lower development cost, stronger user confidence, credible certification, interoperability with a wider market, and a business capable of surviving evolving social expectations about data ownership and platform power.

16. Getting Involved

WikiWe is intended to develop specifications, reference implementations, conformance tests, and WikiWe Administrator governance model templates through public participation. The canonical public entry point is wikiwe.org, where protocol drafts, contribution instructions, repositories, issue trackers, and certification materials are linked as they become available. The WikiWe protocols and specifications should not be confused with the WikiWe Model, the multiparty governance agreement adopted by the Ricardiam reference deployment, which remains under development and is confidential by its own terms. Protocol conformance, as described in this paper, is anticipated to be mostly independent of Administrator governance details. Any governance requirements will be specified in the WikiWe protocol specification.

Contributors should be able to participate at several levels. Some may review agreement and reputation semantics, others may develop cryptographic proofs or compatibility profiles, and others may implement wallets, channels, storage adapters, Administrator services, or conformance tests. Issues that affect an external standard should be coordinated with the relevant standards community rather than resolved only inside WikiWe.

The Ricardiam Administrator is intended to provide the first integrated reference implementation. Protocol changes should nevertheless be proposed and reviewed independently of Ricardiam's product priorities so that other Administrators can implement the same requirements without adopting Ricardiam-specific architecture.

17. Open Questions

Several questions should remain open until implementation and external review provide stronger answers.

1. **Record format and control provenance.** Should the normative WikiWe Record remain independent, become an ACDC profile, or support deterministic conversion while using KERI-compatible key-event histories?
2. **Scoped completeness.** Which combination of substreams, Merkle structures, accumulators, and zero-knowledge techniques can prove no responsive record was omitted while revealing as little unrelated classification information as possible?
3. **Classification integrity.** How should topic and time classifications be assigned, challenged, corrected, and audited when a dishonest issuer or holder could benefit from misclassification?
4. **Channel profiles.** Which use cases are best served by Nostr, DIDComm, TSP, HTTPS, or other transports, and what minimum replication and reconciliation behavior should every Agreement Channel provide?
5. **Performance and scale.** What checkpoint frequency, registry structure, proof size, wallet size, retention policy, and reconciliation strategy remain practical for long-lived Hubs and intermittently connected Members?
6. **Continuity and succession.** How should key-event histories, threshold recovery, Hub agreements, retained checkpoints, and competing successor proposals be combined without converting recovery evidence into permanent central authority?
7. **Certification boundaries.** Which requirements belong in the WikiWe Core Profile, and which should remain optional profiles for wallets, Agreement Channels, reputation, continuity, and Administrator operations?
8. **Governance and intellectual property.** Which specification license, code license, patent commitment, and public change process will best support broad implementation without allowing one contributor or vendor to capture the protocol?
9. **Erasure and completeness.** How should wallets and verifiers represent, score, and dispute unavailable-record states that arise when an erased or key-destroyed payload falls within a claimed completeness scope?
10. **Revocation-status privacy.** How should revocation and status checking be performed without disclosing verification events to issuers, and which combination of status lists, cached validity windows, or accumulator-based non-revocation proofs best balances freshness against holder privacy?

These questions are invitations for implementers, cryptographers, standards participants, privacy specialists, and operators to test the architecture rather than assumptions that must be concealed before publication.

18. Conclusion

We have proposed a protocol for trustworthy collaboration without permanent dependence on a central record keeper. We began with the usual framework of signed documents, credentials, and application databases, which provides useful evidence but is incomplete when the service holding the evidence controls its continued availability, interpretation, and portability.

To address this problem, WikiWe represents significant events as Verifiable Records whose identity is bound to canonical content and whose authentication can be separated from authorization. Agreements are joined to channels that preserve exact-version acceptance, notice, performance, amendment, and withdrawal as a continuing history. Approved deterministic procedures administer rules that must produce repeatable results, while artificial intelligence remains an authoring and review aid rather than a runtime source of binding decisions.

Reputation is retained by or for its subject. A distributed registry carries commitments that can later authenticate private events without publishing their contents. Established presentation protocols can carry credential requests and wallet responses, while WikiWe binds those exchanges to agreement context, authorization, and consequential transactions. Selective disclosure permits a member to reveal only what is relevant, while scoped completeness proofs allow the member to establish that nothing responsive was omitted from a stated topic, time period, or combination of both. The same architecture supports feedback permissions, delegated authority, revocation, correction, and migration among access agents.

Records can be carried by conventional databases, Git repositories, local-first applications, user-controlled data stores, relay networks, secure messaging transports, portable bundles, or content-addressed networks without changing their logical identities. Established issuance, presentation, authorization, federation, transparency, and trust-registry protocols can supply specialized functions around those records. WikiWe's role is to connect them through common semantics for agreements, promises, authority, reputation, completeness, and continuity rather than replace each mature protocol with a proprietary equivalent.

Replicas preserve history, while signed checkpoints and verifiable key histories provide locally held evidence for evaluating successor services after failure or compromise. Participants remain free to recognize the continuation that preserves the records and authority they consider legitimate.

The Ricardiam reference implementation can demonstrate that these mechanisms compose in practice, while uRULEu's public profiles and certification program can allow independent Administrators to implement the same protocol under their own brands and operating models.

The resulting system is not one authoritative database. It is a protocol through which people and organizations can preserve trustworthy shared histories while retaining the freedom to choose who stores, administers, and helps interpret them.

Acknowledgments

This paper has been materially informed by the Internet Identity Workshop community. The proceedings of IIW 40, IIW 41, and IIW 42 document an unusually open body of working discussion concerning digital credentials, OpenID4VC, KERI and ACDC, DIDComm, machine-readable personal terms, delegated authorization, trust registries, organizational credentials, content provenance, and decentralized trust graphs [15–17]. Those discussions helped identify where WikiWe should integrate existing work, where it should define a profile, and where its proposed contribution remains distinct.

We thank the IIW session conveners, participants, volunteer note takers, and proceedings editors whose public records make this cross-project learning possible. We also acknowledge the standards and open-source communities working through the OpenID Foundation, W3C, IETF, Decentralized Identity Foundation, Trust Over IP and LF Decentralized Trust, IEEE and the MyTerms community, GLEIF, C2PA, the Nostr community, the OpenWallet Foundation, and related projects. Citation or acknowledgment does not imply that any person or organization has reviewed or endorsed WikiWe.

Appendix A. Glossary of Actors and Terms

Term	Meaning in this paper
Member	A person or organization participating in one or more WikiWe relationships.
Hub	A purpose-defined community, project, enterprise, or collaborative context operating under an agreement or Charter.
Administrator	A service operator providing infrastructure and operational services to one or more Hubs.
Access Agent	A replaceable provider selected by a Member to hold or help operate wallet data, keys, recovery material, or disclosures.

Term	Meaning in this paper
Issuer	An actor or authorized procedure that creates a credential or other Verifiable Record.
Verifier	An actor or system that requests and evaluates records, credentials, authorization evidence, or proofs.
Certifier	An organization that evaluates an identified implementation or service against published conformance requirements.
uRULEu Institute	The proposed neutral steward of the WikiWe Protocol and certifier of conforming Administrators and implementations.
Verifiable Record	A canonical record whose content, issuer, context, relationships, and authorization evidence can be independently checked.
Agreement Channel	A stable record stream associated with an agreement or continuing relationship, independent of any single messaging service or carrier.
Reputation Wallet	User-controlled or user-selected custody for reputation credentials, private events, keys, identity linkage, and disclosure proofs.
Commitment Registry	A broadly replicated collection of cryptographic commitments designed not to disclose intelligible reputation payloads.
Conformance Profile	A versioned selection of mandatory protocol behavior, formats, options, tests, and rejection rules for a defined implementation class.
Agreement Manifest	The machine-readable companion to a human-readable agreement, identifying the exact text version, incorporated documents, channel identity, joining conditions, and parameters or procedure modules software must administer.
Subject Authorization Record	A record by which a subject permits an identified issuer or class of issuers to create defined kinds of reputation records concerning the subject, within stated limits of purpose, duration, and use.
Feedback Capability	A scoped, attenuable permission authorizing a counterparty to issue defined reputation events about a subject concerning an identified relationship or transaction.
Checkpoint	A periodic signed manifest naming the current roots or heads of important record collections, linked to its predecessor, and retained by participants as continuity evidence.
Existence Claim / Bounded Completeness Claim / Complete-History Claim	The three disclosure claim types defined in Section 8: authenticity of a selected record; completeness within an authenticated topic and/or time scope; completeness over the full relevant history.
WikiWe Model	A multiparty governance agreement, similar to bylaws, under which WikiWe-compliant Administrators and their Hubs are governed. The Model is under active development; it is provided to prospective Members and Administrators during admission and is confidential by its own terms. The protocol described in this paper does not depend on any particular version of the Model.

References

[1] A. Rundgren, B. Jordan, and S. Erdtman, "JSON Canonicalization Scheme (JCS), RFC 8785," June 2020.

[2] C. Bormann and P. Hoffman, "Concise Binary Object Representation (CBOR), RFC 8949," December 2020.

[3] Google Cloud, "Open Knowledge Format (OKF), Version 0.1 — Draft," 2026.

[4] World Wide Web Consortium, "Verifiable Credentials Data Model v2.0," W3C Recommendation, 15 May 2025.

[5] World Wide Web Consortium, "Verifiable Credential Data Integrity 1.0," W3C Recommendation, 15 May 2025.

[6] D. Fett, K. Yasuda, and B. Campbell, "Selective Disclosure for JSON Web Tokens, RFC 9901," November 2025.

[7] B. Laurie, E. Messeri, and R. Stradling, "Certificate Transparency Version 2.0, RFC 9162," December 2021.

[8] J. Li, N. Li, and R. Xue, "Universal Accumulators with Efficient Nonmembership Proofs," in *Applied Cryptography and Network Security*, LNCS 4521, pp. 253–269, 2007.

[9] A. Birgisson, J. G. Politz, U. Erlingsson, A. Taly, M. Vrable, and M. Lentczner, "Macaroons: Cookies with Contextual Caveats for Decentralized Authorization in the Cloud," *Proceedings of the Network and Distributed System Security Symposium*, 2014.

[10] S. Chacon and B. Straub, "Git Internals — Git Objects," in *Pro Git*, 2nd ed., Apress, 2014.

[11] J. Benet, "IPFS — Content Addressed, Versioned, P2P File System," arXiv:1407.3561, 2014.

[12] S. Haber and W. S. Stornetta, "How to Time-Stamp a Digital Document," *Journal of Cryptology*, vol. 3, pp. 99–111, 1991.

[13] World Wide Web Consortium, "Decentralized Identifiers (DIDs) v1.0," W3C Recommendation, 19 July 2022.

[14] T. Lodderstedt, K. Yasuda, D. Fett, and J. Heenan, "OpenID for Verifiable Presentations 1.0," OpenID Foundation Final Specification, 9 July 2025.

[15] Internet Identity Workshop, "IIW 40 Book of Proceedings — Spring 2025," April 2025.

[16] Internet Identity Workshop, "IIW 41 Book of Proceedings — Fall 2025," October 2025.

[17] Internet Identity Workshop, "IIW 42 Book of Proceedings — Spring 2026," April 2026.

[18] T. Lodderstedt, K. Yasuda, T. Looker, and P. Bastian, "OpenID for Verifiable Credential Issuance 1.0," OpenID Foundation Final Specification, 16 September 2025.

[19] World Wide Web Consortium, "Digital Credentials," W3C Working Draft, 16 July 2026.

[20] K. Yasuda, T. Lodderstedt, C. Bormann, and J. Heenan, "OpenID4VC High Assurance Interoperability Profile 1.0," OpenID Foundation Final Specification, 24 December 2025.

[21] Trust Over IP Foundation, "Key Event Receipt Infrastructure (KERI) Specification," latest published specification, accessed 16 July 2026.

[22] Trust Over IP Foundation, "Authentic Chained Data Containers (ACDC) Specification," latest published specification, accessed 16 July 2026.

[23] Nostr Protocol, "NIP-01: Basic Protocol Flow Description," draft specification, accessed 16 July 2026.

[24] Nostr Protocol, "NIP-29: Relay-Based Groups," draft specification, accessed 16 July 2026.

[25] Decentralized Identity Foundation, “DIDComm Messaging Specification v2.1,” DIF Ratified Specification.

[26] Trust Over IP Foundation, “Trust Spanning Protocol Specification,” Experimental Implementor’s Draft, accessed 16 July 2026.

[27] IEEE Standards Association, “IEEE 7012-2025 — IEEE Standard for Machine Readable Personal Privacy Terms,” published 20 January 2026.

[28] J. Richer and F. Imbault, “Grant Negotiation and Authorization Protocol (GNAP), RFC 9635,” October 2024.

[29] M. Kleppmann, A. Wiggins, P. van Hardenberg, and M. McGranaghan, “Local-First Software: You Own Your Data, in Spite of the Cloud,” *Proceedings of Onward! 2019*, 2019.

[30] W3C Solid Community Group, “Solid Protocol, Version 0.11.0,” Draft Community Group Report, 12 May 2024.

[31] H. Birkholz, A. Delignat-Lavaud, C. Fournet, Y. Deshpande, and S. Lasker, “An Architecture for Trustworthy and Transparent Digital Supply Chains, RFC 9943,” June 2026.

[32] R. Hedberg, M. B. Jones, A. Å. Solberg, J. Bradley, G. De Marco, and V. Dzhuvinov, “OpenID Federation 1.0,” OpenID Foundation Final Specification, 17 February 2026.

[33] Trust Over IP Foundation, “Trust Registry Query Protocol 2.0,” Approved Specification, 2026.

[34] Fraunhofer Institute for Industrial Engineering IAO, “Trust Management Infrastructure (TRAIN),” accessed 16 July 2026.

[35] Global Legal Entity Identifier Foundation, “Introducing the Verifiable LEI (vLEI),” accessed 16 July 2026.

[36] LF Decentralized Trust, “Decentralized Trust Infrastructure at LF: A Progress Report,” 5 March 2026.

[37] Coalition for Content Provenance and Authenticity, “C2PA Technical Specification, Version 2.4,” 2026.