

Revocation and Containment in AI Agent Networks

Executive brief and engineering implementation guide

Dr. Carsten Stöcker
Spherity GmbH

Prepared: 11 October 2026



Except where otherwise noted, this work is licensed under the Creative Commons Attribution 4.0 International License (CC BY 4.0). © 2026 Dr. Carsten Stöcker.

1 Executive summary

The basic idea. AI agents can call tools, share files and start work in other organisations. Before an action takes effect, the organisation controlling the resource must check both permission (mandate) and the evidence required for that action (assurance). A signed credential can supply part of this evidence. Revocation withdraws previously accepted authority or evidence. Containment limits further effects, including work already sent to other systems. The controls must operate where the action is carried out; stopping the proposing agent alone may leave remote work running. [A §§2–3, 11; B §3]

A simple example. An engineering agent wants to send a confidential design file to a subcontractor. It needs permission to release that file to that recipient, plus current evidence that this exact release meets the required security and handling conditions. Once the file-release system has validated a withdrawal of that evidence, it reassesses the request and blocks release unless other current, accepted evidence supports approval. It also blocks release when the required status information is too old. A queued request cannot bypass these checks. Revocation cannot undo a disclosure that has already happened. Any work already dispatched remains tracked until execution or cancellation is confirmed. [A §§2–3, 11; B §§3.2–3.5]

Agent autonomy needs a resource boundary that can withdraw permission, stop stale approvals and retain responsibility for remote work. Revoking a credential at its source does not immediately stop an agent, a queued message or an operation in another organisation. The receiving resource must check authenticated authority, its permitted age and installed withdrawals before the effect becomes irreversible. [A §§2, 5, 11; B §3]

The two-paper series connects credential status, evidence processing and service capacity to this enforcement boundary. Article A specifies the architecture and conditional bounds. Article B supplies a protected protocol, finite-state checks and synthetic workload and privacy experiments. Together they provide design obligations and reproducible model results. Production timing, complete software conformance and real-world privacy remain to be established. [A §1; B §§4, 6–10]

Fine-grained credentials can restrict authority more precisely. They also add status entries, validation work and withdrawal processing. In Article A’s fixture, twenty distinct statuses per

bundle and 50% decoy allocation require **40 times the raw storage**. Encoded snapshots cost **40.03 times** the sparse single-status baseline without masking, and **1475.44 times** with random initial masks. These are representation comparisons. They are not latency, risk or privacy multipliers. [A §§7–8]

Table 1: Leadership decisions and engineering commitments.

Leadership decision	Engineering commitment
Which effects may agents perform	Name the resource, recipient, purpose, continuation and harms. Bind mandate and assurance to that same complete record.
How long stale authority may remain usable	Set a maximum source age and enforce absolute expiry at each covered receiver. Accept that partitions can stop legitimate work.
What infrastructure must remain available under load	Reserve withdrawal-processing capacity, bound admitted work and qualify a restrictive fallback. Track unresolved remote operations through closure.
Which privacy or hosting investment to make	Begin with authenticated local state. Compare mirrors and ledger nodes under the same policy and faults. Require a specific privacy benefit before adding experimental decoys or masking.

Where to start. Implement one receiver-controlled file-release workflow. Use ordinary unmasked status semantics, signed snapshots, full evidence reevaluation and durable operation identifiers. Test withdrawal races, lost events, expiry, retries and receiver crashes before adding dependency optimisations or experimental privacy. [B §11.5]

Sections 2–4 explain the problem and architecture. Sections 5–10 translate the seven solution patterns into capacity, privacy and timing decisions, including anonymous status proofs and adoption. Section 11 gives the delivery checklist. Section 12 identifies the evidence and sources.

2 Workloads and failure modes

The example is an engineering agent releasing a design file to a named subcontractor. Its mandate may depend on representation and delegation credentials. Assurance may depend on operator, tool and provenance evidence. Revocation of any indispensable premise can change whether that exact release is allowed. A valid alternative witness may preserve permission, so an update requires policy-aware reevaluation. [A §§2–3]

Count four different workloads. k_{pop} is the average number of unique allocated status entries per bundle across the population. It is different from the evidence inspected for one decision, the number of list objects fetched, and the amount of reevaluation after a change. Twenty checks in four cached lists do not imply twenty network requests. Shared credential versions count once in physical allocation. [A §3]

For status width w , genuine occupancy u and decoy share d , the approximate raw-capacity factor against one occupied one-bit status per bundle is:

$$F_{raw} \approx \frac{k_{pop}w}{u(1-d)}.$$

This approximation ignores list rounding. Exact list counts can dominate small populations. With $k_{pop} = 20$, $d = 0.5$ and $u = w = 1$, the factor is 40. Both designs can still fit in one minimum-sized list when the population is small enough. [A §7.1]

Table 2: Article A snapshot representation costs.

Article A snapshot fixture	Raw bytes	Mean encoded bytes	Ratio to baseline
Single-status sparse baseline	16,384	593.10	1.00
Twenty statuses and 50% decoys, unmasked	655,360	23,743.10	40.03
Same allocation, random initial masks	655,360	875,083.80	1475.44

The population is 131,072 bundles; each list has 131,072 bits. Status entries are pooled across lists. The sparse proxy has approximately 0.1% set bits. Encoded bytes include GZIP (GNU zip) lossless compression, base64url and its Multibase prefix. They exclude credential envelopes, signatures and transport. The two expanded representations differ by **36.86 times**, despite equal raw capacity. Their incident semantics are not established as equivalent. [A §8.1, Tables 9–10]

Three privacy questions remain separate. Local resident reads can remove per-decision requests to a status endpoint. Matched decoys address inference about which unknown positions are genuine. Initial masking changes the published representation and its compressibility. The ratios above quantify capacity and representation costs, not lookup privacy. [A §§5.4.5, 6–8]

In a multi-agent system (MAS), proposals and effects are asynchronous. A withdrawal can race with a queued release. Cached state can remain readable after it becomes too old to authorise work. A stopped agent can leave operations pending at a remote receiver. An event acknowledgement, ledger transaction or request timeout does not resolve all three conditions. [B §§3, 5.6–5.7]

3 A multi issuer agent network

What role A means. In Figure 1, role A identifies authority and evidence issuers; “A” is a role label, not an abbreviation for AI agents. A business register may attest legal identity, an employer may issue a mandate, and an assurance provider may attest tool or design properties. Each receiver recognises each issuer only for specified claims and scopes. Role B distributes authenticated status. The C domains control resources and apply their own policy decision point (PDP) and policy enforcement point (PEP). These are logical roles: one organisation can perform several, and a network can contain many instances of each. [A §§3, 5.1, 11]

Authority and evidence issuers (role A)

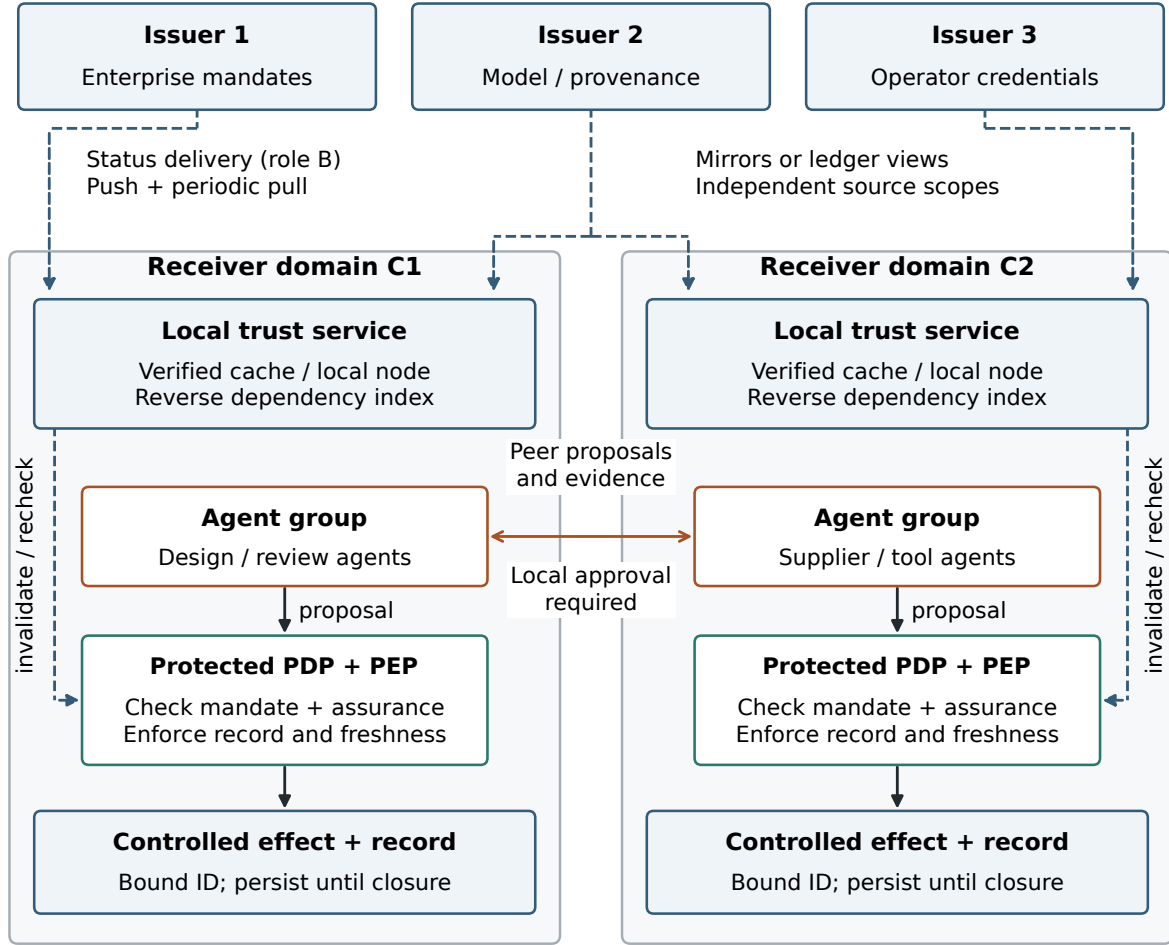


Figure 1: Conceptual MAS network and revocation propagation. Role A supplies scoped authority and assurance evidence; role B delivers status. Each receiving C domain checks mandate and assurance for its own effects. The peer arrow shows logical coordination, not a route around enforcement. Dashed paths carry evidence and status. This combines the articles’ roles; it is an architectural design, not an evaluated deployment. [A §§3, 5, 11; B §3]

What peer messages carry. These are application messages between agents: requests, proposals, coordination updates, results or evidence references. For example, an engineering agent in C1 may ask a supplier agent in C2: “Run simulation S on design D, revision 7, using tool T, and return the approved result summary.” C2 must decide whether it may run that job and disclose the result. A reply may identify the result by hash and attach signed assurance evidence; C1 checks the signer’s recognised scope and the evidence’s binding and applicability before relying on it. “Cancel operation O17” is another peer message: the receiver checks the requester’s mandate and the operation binding. A request or acknowledgement alone does not prove cancellation. [A §§2–3, 11; B §3.5]

No automatic authority transfer. Sending a message, signing it or approving an action in C1 does not itself authorise an effect in C2. Explicit delegation is possible, but C2 must accept the issuer and delegation chain and check their scope, audience, expiry, current status and any right to delegate further. Figure 1 therefore says “Local approval required.” Each covered effect must pass its applicable gate. Releasing confidential inputs or results requires sender-side approval; executing a requested tool requires approval at the receiving resource. The peer arrow does not bypass either boundary. [A §§2–3, 11; B §§3.1–3.4]

Mandate and assurance at each receiving boundary. The PDP evaluates mandate and assurance separately against the same complete execution record. Mandate asks who may perform which action, for whom, on which resource, for what purpose and for how long. Assurance asks whether applicable evidence and models support the required safeguards, risk bounds and operating conditions for that same action and continuation, including its tool, payload and context. Both must pass: permission does not establish assurance, and assurance does not grant permission. The PEP enforces the bound decision and checks the required dependency state, freshness and expiry at the last preventable effect. Each receiving domain owns this decision independently of its peers. It may reuse authenticated caches or scoped attestations under its own reliance policy; PDP and PEP need not duplicate every verification. [A §§2–3, 11; B §§3.1–3.4]

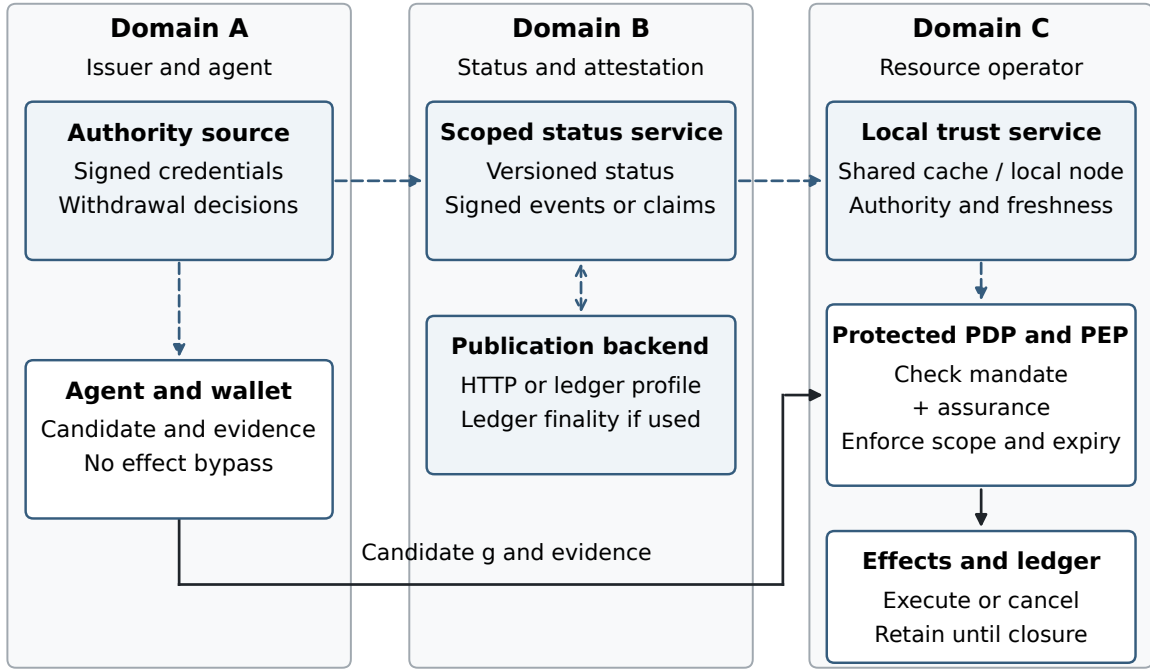
Follow the withdrawal to the effect. Distinguish the source decision, publication, delivery, validated installation, final resource decision and terminal operation receipt. Propagation is complete for a covered effect only when its receiver can no longer execute using the withdrawn premise. Authentication of an intermediate message establishes neither this outcome nor the truth of the original claim. [B §§3.2, 5.7]

If a design-file assurance result is withdrawn, a reverse index identifies dependent decisions. The receiver reevaluates the complete file-release record. It either finds an admissible current witness or blocks the release. Other domains repeat this process under their own recognition and freshness rules. There is no assumed global issuer, shared policy or instantaneous cross-domain revocation. [A §§3.1, 11.3]

4 The resource decision boundary

One operation across three domains. Figure 2 isolates a single receiving resource boundary. A, B and C name logical domains, not grades of assurance. They instantiate the roles used in Figure 1. Domain A supplies authority and evidence, Domain B distributes status, and Domain C controls the resource. The drawing places an authority source and a proposing agent in Domain A for illustration; an agent need not belong to its evidence issuer. [A §§5.1, 11]

Three domains and one resource enforcement boundary



Dashed: evidence and status. Solid: candidate, execution and closure path.

Figure 2: Three-domain architecture, adapted from Article A Figure 5a with clarified labels. A proposes an operation and supplies evidence; B distributes authenticated status; C evaluates mandate and assurance and enforces its own resource decision. This is one local boundary within a larger MAS, not a network-wide permission service.

Read the paths separately. The solid candidate path brings the proposed operation and evidence from A to C. The dashed status paths bring authenticated updates through B into C’s local trust state. Domain B may transport status or provide an attestation under a separate, scoped delegation; neither role gives it general permission to approve C’s effects. C’s PDP and PEP apply the local checks explained in Section 3. Its persistent operation record tracks execution or cancellation to a confirmed terminal outcome. HTTP delivery or ledger hosting changes the distribution mechanism, not C’s decision authority. Sender-side gates for covered disclosures still apply, although this figure focuses on C. This extends the resource focus of zero trust architecture (ZTA) with freshness and operation-accounting obligations. [A §§5.1, 11; B §3; S1]

Extend the pattern to a MAS. Repeat C’s boundary for every receiving domain and covered effect, with as many A issuers and B delivery paths as needed. For example, an engineering agent in C1 asks a simulation supplier in C2 to run a job; C2 then asks a laboratory agent in C3 to perform a test. C2’s approval does not authorise C3’s test or C1’s later release of the result. Each effect needs its own locally accepted record and mandate and assurance checks. Bind each child operation to its parent workflow, give it its own stable logical operation identifier (ID), and preserve that ID across retries. Keep unresolved work open until authoritative terminal evidence arrives. A withdrawal triggers local reevaluation wherever the withdrawn premise is used, subject to each receiver’s freshness and enforcement bounds. This provides no instantaneous global stop or atomic network-wide transaction. Figure 1 shows the resulting network of repeated Figure 2 boundaries. [A §§3, 11; B §§3.2–3.5]

P1 Protected gates and finite leases. Bind the approval to one record $g = (a, h, \pi, \nu)$: action, duration, continuation and context. Context fixes the resource, audience, payload, policy and bound evidence versions. Serialize the final dependency-generation and expiry checks with the last preventable effect. A proposer must not bypass the gate through another tool, queue or application programming interface (API). Renewable status assertions belong to the supporting witness; changing a version bound into the immutable record requires a newly admitted record. [A §2; B §§3.1–3.4, 11.1]

P6 Persistent cross-domain closure. Persist the logical operation identifier, record binding and any allowance debit before dispatch. Keep the operation in P_t , the set of unresolved operations, through retry, agent retirement and restart. Close it only on authenticated, contract-valid terminal evidence for that operation. The receiver must bind execution or cancellation to durable outcome state. A timeout preserves uncertainty; it does not prove cancellation. [B §§3.5, 11.1]

The persistent operation ledger is a durable accounting record. It need not be a distributed ledger. Its guarantee also depends on the resource adapter: a local database transaction cannot make an arbitrary remote action atomic or retract an effect already committed. [B §§3.5, 11.5]

5 Capacity and delivery patterns

The patterns below support the protected receiver. They should be evaluated using the same policy, evidence and effect semantics. A faster cache or smaller update cannot compensate for an incorrect execution boundary. [A §5.2; B §11]

Table 3: Capacity and delivery patterns.

Pattern	Implementation rule	Benefit and limit
P2 Shared caches and change events	Validate source authority, profile, epoch, version and age before exposing a shared snapshot. Combine push notifications with periodic reconciliation. Record the version used by each decision.	Reduces duplicate downloads and proof verification; each decision still checks current local status. Lost events, stale positive state and common infrastructure failures remain possible. Internal telemetry can also reveal activity.
P3 Reverse dependency indexing	Map versioned evidence and policy dependencies to affected decisions. Include prohibitions and relevant adverse evidence. Compare incremental results with full reevaluation.	Reduces unnecessary work if the index is complete. An alternative valid witness can preserve approval; blanket descendant rejection is not always correct.
P4 Reserved service capacity	Reserve service for withdrawal work. Bound backlog, job size, scheduling latency and interference. Restrict new admission when the service envelope can no longer certify the deadline.	Provides a conditional deadline certificate under load. A priority flag or low average processor use does not establish the service guarantee.

Pattern	Implementation rule	Benefit and limit
P7 Algorithm agility	Pin algorithm and key versions, signed bytes and downgrade rules. Inventory credential, event, transport, checkpoint, transaction and consensus authentication. Test key withdrawal during recovery.	Supports controlled cryptographic migration. Stronger object signatures do not make every other protocol layer post-quantum or make a false assertion true.

Pattern sources: A §5.2 and §§5.5, 12; B §§11.2–11.4. P1 and P6 are detailed in Section 4; the experimental P5 is detailed in Section 6.

For reserved service rate R_{srv} , scheduling latency θ , work already ahead q , and bounded job work w_o , a sufficient certificate for the remaining service-stage deadline b_o is:

$$\theta + \frac{q + w_o}{R_{srv}} \leq b_o.$$

The work and rate units must agree. The service must supply a first-in, first-out (FIFO) or equivalent non-overtaking guarantee, with permitted blocking and interference included. If this certificate fails, the system needs a separately supported broad hold or protected-expiry path. Failure of the bound does not prove an actual miss; it removes this worst-case certificate. [A Eqs. 19, 46; B §11.3]

Report the availability cost. In Article B’s overload fixture with twenty credentials and 32 affected jobs, a broad hold stops the mirror path at 0.75 seconds but loses 6.871% of otherwise legitimate completions. This is a synthetic model result. A deployed broad hold needs its own bounded implementation and recovery test. [B §6.4]

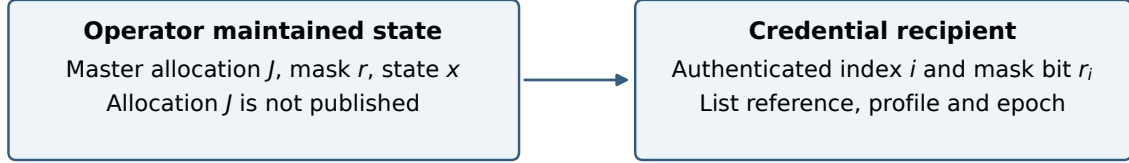
6 Decoys and initial masking

P5 Matched decoy cohorts reserve positions that do not represent credentials and generate cover transitions. The exact matching result assumes that genuine and decoy histories have the same complete observable law, including incident shocks, writers, events, batching and later side information. Matching must hold after conditioning on the observer’s side information. Similar average flip rates are insufficient. The result concerns a declared observer and cohort; it does not establish continuing-service privacy. [A §6; B §§7–8]

A fixed mask therefore adds no membership protection under Article B’s known-initialization observer. Matching without masking is the necessary control. In the fixed 32-genuine/32-decoy experiment, exact matching preserves posterior membership probability 0.5. Half-rate and double-rate decoys instead permit 60.09% and 62.27% observed classification accuracy. These results come from 1,000 synthetic cohorts per case. [B §§7.2, 7.5, 8.1–8.2]

Initial masking and matched cover events

Experimental profile • Twelve illustrative positions • Irreversible status



Only the relevant per-entry fields are disclosed in the credential

Public snapshot transcript

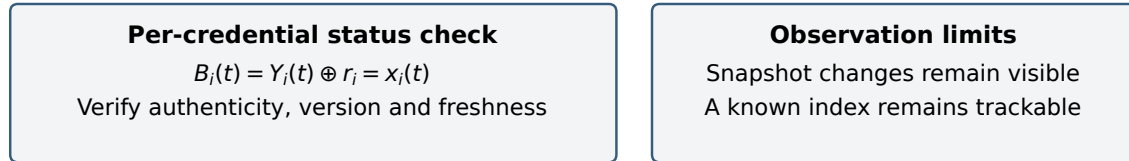
	0	1	2	3	4	5	6	7	8	9	10	11
t_0 : initial	0	1	1	0	1	0	0	1	0	1	1	0
t_1 : position 2 changes	0	1	0	0	1	0	0	1	0	1	1	0
t_2 : position 7 changes	0	1	0	0	1	0	0	0	0	1	1	0

Operator events: genuine $x_2 : 0 \rightarrow 1$ at t_1 ; matched cover $x_7 : 0 \rightarrow 1$ at t_2 .

The illustrated times show encoding mechanics, not evidence that event laws match.

Unmasked initial $r_i = 0$	0	0	0	0	0	0	0	0	0	0	0	0
-------------------------------	---	---	---	---	---	---	---	---	---	---	---	---

Zeros reveal initial semantic state, not the genuine versus decoy allocation.



If $x(t_0) = 0$ is known, then $Y(t_0) = r$: initial masks are observable.

Unknown-membership protection requires matched complete histories and stated side information.

Figure 3: Existing lifecycle diagram, reproduced from Article A Figure 2. The operator retains the allocation J , where 1 denotes genuine and 0 denotes decoy. Holders and verifiers receive their own decoding information. The published bit is $Y_{i,t} = r_i \oplus x_{i,t}$, where x is semantic status and $\oplus$ is exclusive OR. Known-active public initialization reveals r through $Y_0 = r$; snapshot differences expose transitions.

7 Anonymous status proofs and adoption

Protect the status handle as well as the lookup. AnonCreds (Anonymous Credentials) lets a holder prove that its credential is unrevoked in a selected authenticated state without revealing its revocation index. A file-release agent could prove its mandate while withholding that stable locator. Its v1.0 draft is ledger-independent; Indy is one registry option. [S2]

Indy uses Camenisch–Lysyanskaya (CL) credential signatures with a **Type-3 pairing-based cryptographic accumulator** for revocation. This differs from the Rivest–Shamir–Adleman (RSA) group used for CL credential signatures. A holder maintains auxiliary proof material, called a cryptographic witness, using public tails data and status updates. This is distinct from the policy’s evidence witness. Compact verifier state shifts costs to distribution, witness maintenance and proof processing. [S2; S3]

The receiver contract remains. Bind the proof to the same credential, accepted registry state, challenge, audience and complete operation record. A newly generated proof does not refresh the authoritative state watermark. An installed withdrawal cannot be bypassed by presenting an old root that still verifies. If the hidden index prevents P3 from locating affected approvals, a conservative fallback pauses every dependent approval for that registry version until revalidation. Budget enforceable holds and closures under P4. Holder witness repair and new proofs belong to readmission; they must not delay stopping. [A §5.3.1 and Appendix F; B §5.10 and Appendix F]

Anonymous presentation does not conceal disclosed attributes, registry identifiers, update times or all retrieval traffic. Historical Indy update deltas also expose issued and revoked indices. Test these channels separately. Signing the public state with a post-quantum algorithm does not upgrade the classical CL and pairing-based proof system. [S2; S3; A Appendix F]

Adoption supports coexistence. British Columbia’s court-materials service uses BC Wallet credentials, and the maintained wallet includes AnonCreds and Indy registry components. This establishes an active service and implementation, not production revocation volumes. [S4; S5] The AnonCreds project reports continued v1 adoption but incomplete adoption data. [S6] W3C Bitstring Status List has Recommendation status and a multi-vendor interoperability report; cheqd documents both status families. Participation in a test report is not complete conformance or a deployment count. None of these sources supplies comparable market shares. [S7; S8; S9]

Sovrin MainNet is a separate case. Its January 2026 notice describes a read-only archive after the Foundation’s dissolution. That network’s status does not describe the entire Indy or AnonCreds ecosystem. [S10]

Selection rule. Retain ordinary bitstrings as the initial engineering baseline. Evaluate anonymous proofs where hiding the presentation’s status handle is required. Compare total distribution and proof bytes, witness recovery, receiver enforcement and availability under the same freshness policy. The existing GZIP ratios do not rank accumulators, and this series reports no accumulator performance benchmark. [A Appendix F; B §5.10 and Appendix F]

8 Standards and software profiles

The software profile must distinguish normative formats from this series’ proposed extensions. Pin the document version, status purpose, bit order, epoch, accepted authority and decoding limits. [A §4, Appendix E; B §5]

Table 4: Standards and software profiles.

Standard or profile	Engineering requirement and boundary
W3C Bitstring Status List v1.0	World Wide Web Consortium (W3C) Recommendation, 15 May 2025. Its encodedList requires GZIP-compressed bytes and Multibase base64url without padding. This is inside the status-list credential, independently of transport compression. The minimum raw list is 16 kibibytes (KiB; 16,384 bytes). [S7 §§2.2, 3.3]
W3C Bitstring Status List v1.1	First Public Working Draft, 24 September 2026. The optional terse entry uses a 32-bit index addressing 64 lists of 2^{26} entries each. A one-bit list is 8 mebibytes (MiB; 8,388,608 bytes) raw. Bind the status purpose supplied during conversion. Terse references do not make the fetched list smaller or require fetching all 64 lists. [S11]
Proposed packed profile over ERC-7506	Ethereum Request for Comments (ERC) 7506 remains a draft Trusted Hint Registry. It supplies 32-byte hints and writer controls. Packing 256 status bits, defining bit order and enforcing monotone revocation are application-profile obligations. Missing or uninitialised state is unknown, not active. [S12; B §5.5]
FIPS 204 ML-DSA	Federal Information Processing Standard (FIPS) 204 defines the Module-Lattice-Based Digital Signature Algorithm (ML-DSA). Signatures are 2,420, 3,309 or 4,627 bytes for ML-DSA-44, ML-DSA-65 and ML-DSA-87, respectively. Measure complete envelopes and verification work. Object signatures do not protect transport confidentiality or automatically upgrade ledger consensus. [S13; B §5.8]
OpenID SSF and CAEP	Shared Signals Framework (SSF) and Continuous Access Evaluation Profile (CAEP) 1.0 are Final specifications. Authenticate events, map subjects and authority, handle duplicates and reconcile missed delivery. Security Event Tokens (SETs) are not execution leases; SSF prohibits the exp expiration claim. Event receipt does not prove receiver enforcement. [S14–S15; B §5.7]

Keep masked and unissued-decoy profiles explicitly experimental. Dropping GZIP for random data would require a separately specified encoding; it would not recover the sparse-list compression advantage. The one-list random fixture adds only 23 bytes of GZIP overhead to 16,384 raw bytes, about 0.14%. Most of the comparison cost is lost compressibility and extra capacity. [A §§8.1, E.1–E.4]

For ordinary irreversible revocation, the proposed Ethereum Virtual Machine (EVM) update applies bitwise OR atomically to current semantic state:

$$C' = C \vee F.$$

Here C is the current 256-bit semantic word and F selects positions to revoke. All setters and upgrade paths must preserve the rule. A client-side stale read followed by whole-word replacement can lose another writer’s update. For masked publication, $Y' = r \oplus (C \vee F)$: applying OR directly to the wire word is wrong. Reversible suspension needs separate transition rules. Test bit indices 0, 7, 8, 255 and 256 to catch W3C-to-EVM ordering errors. [A §5.4.2; B §§4.4, 5.5]

9 Freshness and hosting choices

A completeness watermark ω asserts that relevant authoritative decisions through that time are reflected in the accepted view. Fetch time and block time do not supply that assertion. At partition onset t_0 , the remaining source-age budget is:

$$H_{fresh} = \max\{0, \Delta t_{\max} - (t_0 - \omega) - \kappa_v\}.$$

$\Delta t_{\max}$ is the accepted age limit; κ_v covers clock uncertainty. The protected timing convention must consume the budget at least as quickly as elapsed physical time, including drift and restart. Take the minimum with every mandatory dependency, lease and validity cap. At expiry, stop affected ordinary effects; separately authorised containment may continue. [A §5.4.4; B §3.4]

Article B sets maximum age to 20 seconds, initial age to 5 seconds and clock allowance to 0.5 seconds. Its residual budget is **14.5 seconds**. A renewed six-second child lease cannot extend that parent-age limit. [B §6.3]

For a distributed ledger technology (DLT) path, submission, inclusion, finality, local validation and gate application all contribute:

$$d_{DLT} \leq p_{\max} + b_{\max} + \tau_{\max} + s_{\max} + a_{\max}.$$

This inequality requires supported upper bounds on every stage. Add the applicable clock margin when comparing it with a deadline. Public-ledger finality or inclusion can stall; measured percentiles cannot establish a finite deterministic bound. [A §5.4.4]

Table 5: Synthetic hosting-fault comparison.

Synthetic fault condition	Signed mirror mean delay	Local ledger model mean delay
Delivered push	1.955 s	6.455 s
Lost push with pull reconciliation	6.779 s	11.535 s
Origin service or DNS outage	2.455 s	6.455 s
Complete inter-domain partition	14.500 s	14.500 s

Source: B Table 9, twenty credentials and eight affected jobs, 128 paired incidents per cell. These are synthetic results under stipulated publication delays of 0.75 seconds for mirrors and 5.25 seconds for ledger state. The Domain Name System (DNS) outage leaves an independent mirror and ledger publishing/peer paths intact. Complete partition causes 81.967% legitimate deadline loss on either path. The results do not rank production backends.

Both designs can serve authenticated resident state without a per-decision remote lookup. Constant-time bit extraction does not make the full validation process constant-time. Lookup privacy also requires query-independent refreshes and no leaking telemetry. Remote procedure calls (RPC), on-demand proofs and node traffic remain separate observation channels. Choose hosting by its complete fault, freshness, governance and cost model. [A §5.4.5; B §§5.6, 6.5]

10 Containment and cascade harm

K_{ij} bounds the conditional expected number of type- i secondary taint episodes from a type- j parent. Longer detection and outgoing-influence windows can raise this number. The bound

requires conditional rate and offspring assumptions at episode birth, with receiver enforcement on every covered influence channel. A file gate alone does not control agent messaging. [A §9.3; B §9.1]

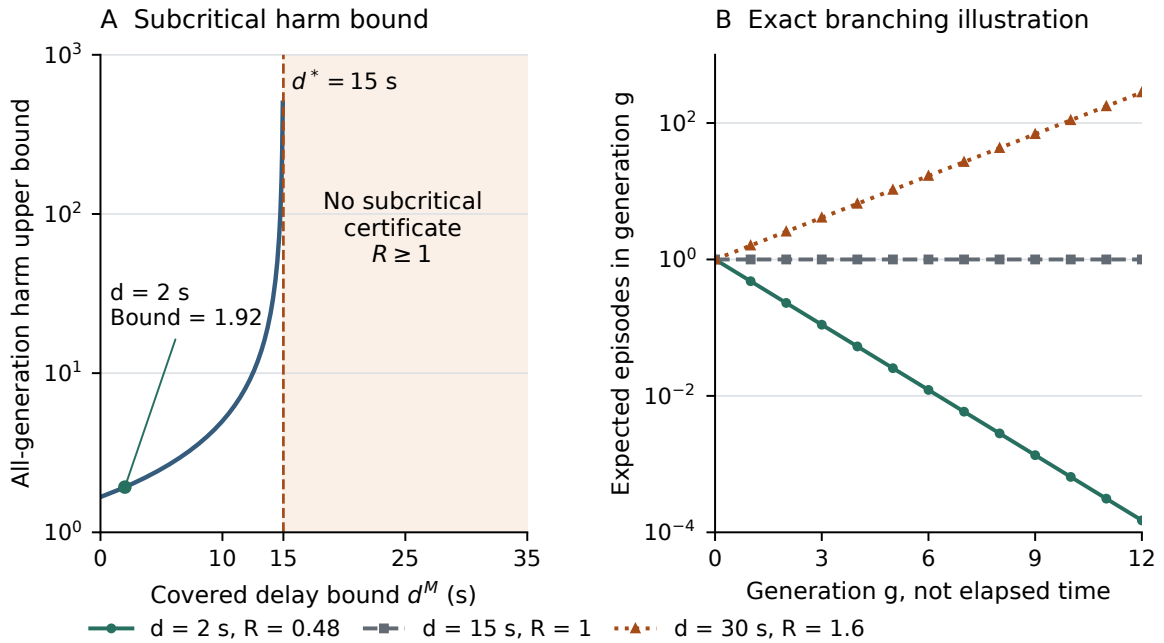
For spectral radius $\rho(K) < 1$, the matrix's subcritical growth condition, the continuing-root envelope is:

$$\mathbb{E}[H_T] \leq \mathbf{c}^\top (I - K)^{-1} \mathbf{b}(T) + h_{pre}.$$

$\mathbf{b}(T)$ bounds initial residual and new roots through T ; $\mathbf{c}$ bounds conditional expected harm per episode. h_{pre} covers earlier pending effects not already counted. Descendants after T are included. This bounds expected loss, not an incident maximum. Continuing external roots can increase cumulative harm even with subcritical propagation. [B §9.2, Proposition 7]

Containment delay and conditional cascade bounds

Illustrative $\beta = 0.04 \text{ s}^{-1}$, $\bar{D} = 10 \text{ s}$; one root, unit episode cost



$R(d^M) = \beta(\bar{D} + d^M)$; panel A uses $1/(1 - R)$ only when $R < 1$.

Reserved-service example: $d^M \leq 1 \text{ s} + 100 \text{ work units}/R_{\text{srv}}$.

$R_{\text{srv}} > 100/14 \text{ work units/s}$ is sufficient for $R < 1$ under this workload contract.

Generation counts can decay; cumulative harm cannot. No population or deployment is fitted.

An upper matrix with spectral radius at least one does not prove growth in a finite network.

Figure 4: Existing analytical illustration, reproduced from Article A Figure 4. With one root, unit episode cost, no prior pending harm, a ten-second conditional mean detection bound and conditional influence-intensity bound of 0.04 per second, the scalar factor is $R(d) = 0.04(10 + d)$. For supported delays of 2, 10 and 14 seconds, the conditional bounds are 1.92, 5 and 25. Panel B shows branching generations, not elapsed time. This illustration is separate from Article B's hosting experiment.

At $\rho(K) \geq 1$, this subcritical certificate is unavailable. That does not prove unlimited or exponential damage in a finite deployment. Reserved service capacity supports the delay premise only when backlog, blocking and receiver deadlines are bounded. Mean workload delays must not be inserted as if they were guaranteed influence limits. [A §9; B §§9.3, 11.3]

11 Engineering execution checklist

Start with one named irreversible effect and its receiver. Use full evidence reevaluation and the ordinary unmasked status profile as the reference implementation. Add optimisations only after they reproduce that decision and preserve the effect boundary. The ownership suggestions below are implementation recommendations. [B §11.5]

Table 6: Engineering execution checklist.

Phase and suggested owner	Required control	Verification and acceptance condition
1 Crypto and wire profiles — Security architecture	Pin source authority, version, purpose, width, order, epoch, algorithms and signed bytes. Bound decoding. Isolate decoy and masked experiments.	Reject malformed, oversized, ambiguous and downgraded inputs within bounded work. Exercise withdrawn keys and bit boundaries. Record raw, compressed, encoded and signed bytes separately.
2 Gate mediation — Resource team	Bind the complete record, receiver, dependency generations and expiry. Serialize the final guard with the last preventable effect. Verify dependency indexing against full reevaluation.	Mutate payload or recipient after evaluation; install withdrawals before and after commit. Earlier relevant local installation must exclude the old witness. A later in-transit update is a separate freshness case.
3 Delivery and freshness — Trust infrastructure	Authenticate local state; combine push and reconciliation. Preserve authoritative watermarks, absolute expiry and clock bounds through restart.	Drop, reorder and duplicate events; partition all update routes; attempt repeated renewal. Stop ordinary effects at the required strict cap. Do not advance a watermark ahead of unfinished withdrawal processing.
4 Persistent operations — Receiver and storage teams	Persist identity, admission and debit before dispatch. Retain unresolved state. Bind receiver outcome or cancellation to durable tombstones and valid terminal receipts.	Crash sender and receiver independently; retry and lose receipts. Preserve at-most-once effect and debit per bound identifier under the qualified atomicity contract. Timeout must not remove an operation from P_t .
5 Fallbacks and capacity — Platform operations	Reserve service; bound work and interference. Define broad hold or expiry and separately authorised recovery. Retain pending accounting during restriction.	Inject bursts and competing work. Meet the service certificate or use the supported fallback before the effect deadline. Report prohibited effects, legitimate loss, unresolved outcomes and deadline exceedances separately.

Control sources: A Appendix B; B §11.5, Table 13 and Appendix B. Passing these tests covers exercised schedules; the implementation still needs a justified mapping to the formal model.

Minimum operational record. Capture authoritative withdrawal, publication, validated installation, generation change, final effect decision and terminal receipt as distinct events. Monitor conservative watermark age, remaining freshness headroom, queue work, service delay,

unresolved-operation count and age, and recovery state. Use clock-error intervals for cross-domain timing. Cache-hit rate and average latency alone do not establish containment. [A §12.3; B §11.3]

Release sequence. Qualify the receiver and durable operation path first. Then test event loss, partitions, expiry and overload. Compare mirrors with ledger replicas under identical policy and fault schedules. Finally assess incremental dependency processing, decoy lifecycles and cryptographic migration using measured costs and explicit acceptance criteria. [B §§10.3, 11.5]

12 Evidence and sources

Table 7: Evidence in the research series.

Evidence in the series	Result and scope
Mathematical propositions	Conditional eligibility, privacy, stopping, accounting and expected-harm results. Their source-honesty, timing, mediation and observation premises remain essential.
Finite protocol exploration	477 states for one operation; 6,651 for two. No monitored baseline violation. Removing the final generation fence, parent-age cap or terminal-evidence rule yields failures in 6, 7 and 3 transitions. [B §4]
Executed model studies	6,144 synthetic incidents and 5,000 decoy cohorts. These exercise stipulated conditions; they do not measure a deployed mirror, ledger or agent platform. [B §§6–8]
Reproducible calculations and charts	GZIP byte fixtures, exact model calculations and all eight appendix outcome charts are supplied across the two papers. Hardware calibration and deployment conformance remain separate tasks. [A Appendix D; B Appendix D]

Remaining research. The priorities are independent receiver failure and recovery, real service and publication measurements, complete signed-wire profiles, continuous decoy issuance and retirement, and calibration of conditional harm parameters under shocks. Accumulator integration needs protocol-specific proof binding, full lifecycle benchmarks and implementation tests; cross-domain dispute mechanisms need separate constructions and proofs. Source truth and evidence-discovery completeness remain trust assumptions. [B §§10–13, Appendix C]

Research sources

[A] C. Stöcker. *Cascading Trust Failures in AI Agent Networks: A Trust Architecture for Scalable Revocation and Bounded Containment*. Updated manuscript, 10 October 2026. Sections, equations and tables cited above refer to this article.

[B] C. Stöcker. *Runtime Revocation in Multi-Agent Systems: Protocol Verification and Evaluation of Privacy and Containment*. Updated manuscript, 10 October 2026. The companion protocol and model study.

Primary standards and implementation sources

— checked 10 October 2026

- [S1] National Institute of Standards and Technology (NIST), Special Publication (SP) 800-207. *Zero Trust Architecture*. August 2020. <https://csrc.nist.gov/pubs/sp/800/207/final>
- [S2] AnonCreds Working Group. *AnonCreds Specification*. v1.0 Draft. <https://anoncreds.github.io/anoncreds-spec/>
- [S3] Hyperledger Indy. *Indy SDK for Node.js*. Revocation registry API and Type-3 accumulator profile. <https://hyperledger-indy.readthedocs.io/projects/sdk/en/latest/wrappers/nodejs/README.html>
- [S4] Government of British Columbia. *Access to Court Materials: Access Requirements*. <https://justice.gov.bc.ca/dars/pre-vc-auth/access-requirements/>
- [S5] Government of British Columbia. *BC Wallet mobile app package manifest*. <https://github.com/bcgov/bc-wallet-mobile/blob/main/app/package.json>
- [S6] LF Decentralized Trust. *2025 Annual Review Hyperledger AnonCreds*. <https://lf-decentralized-trust.github.io/governance/project-updates/2025/2025-annual-hyperledger-anoncreds/>
- [S7] W3C. *Bitstring Status List v1.0*. Recommendation, 15 May 2025. <https://www.w3.org/TR/2025/REC-vc-bitstring-status-list-20250515/>
- [S8] W3C. *VC Bitstring Status List Interoperability Report 1.0*. Test run 4 October 2026. <https://w3c.github.io/vc-bitstring-status-list-test-suite/>
- [S9] cheqd. *Product Overview*. Supported status-list families. <https://docs.cheqd.io/product>
- [S10] Sovrin. *The Sovrin Foundation Has Been Dissolved — but Sovrin MainNet Remains*. 20 January 2026. <https://sovrin.org/the-sovrin-foundation-has-been-dissolved-but-sovrin-mainnet-remains/>
- [S11] W3C. *Bitstring Status List v1.1*. First Public Working Draft, 24 September 2026. <https://www.w3.org/TR/2026/WD-vc-bitstring-status-list-1.1-20260924/>
- [S12] ERC-7506 *Trusted Hint Registry*. Draft. <https://eips.ethereum.org/EIPS/eip-7506>
- [S13] NIST. *FIPS 204 Module-Lattice-Based Digital Signature Standard*. 13 August 2024. <https://csrc.nist.gov/pubs/fips/204/final>
- [S14] OpenID Foundation. *Shared Signals Framework Specification 1.0*. Final, 29 August 2025. https://openid.net/specs/openid-sharedsignals-framework-1_0-final.html
- [S15] OpenID Foundation. *Continuous Access Evaluation Profile 1.0*. Final, 29 August 2025. https://openid.net/specs/openid-caep-1_0-final.html