The Federated Harbor:

Identity, Coordination, and Settlement Across Administrative Domains

Erich Owens

Curiositech LLC

engineering@portdaddy.dev

September 2026
Version 1.1 (textbook edition)

Abstract

A demonstration is scheduled for 4pm. Alice runs the back end on her laptop; Bob runs the front end on his. Each operator has a working local harbor, but Alice's `db:write` card is meaningless to Bob's daemon, Bob cannot yet observe Alice's revocations, and a bond posted at one harbor cannot automatically settle damage at the other. This paper specifies the missing federation boundary: a cross-harbor capability-transfer ceremony with composed attenuation, revocation dissemination through a witnessed anti-entropy overlay, a conditional escrow construction, and bonded admission for new harbors. The assurance case is deliberately narrower than earlier drafts. Epidemic dissemination supplies an expected $\Theta(\log m)$ result only under a connected reliable-round model, not a hard freshness deadline. A signed escrow promise makes theft detectable but does not prevent it; the extractable-value bound requires a custody ledger that structurally exposes only recipient-whitelisted transitions. Cross-harbor conservation is stated as a bucket-partition invariant and model-checked only at the recorded finite bound. The protocol models and federation runtime remain partial. The contribution is therefore a falsifiable design and research boundary, not a claim that federation is already deployed or trustless.

Keywords: federated identity, capability delegation, cross-realm authorization, mechanism design, distributed trust, certificate transparency, atomic settlement, mutual suspicion.

Reading time: approximately 45 minutes end-to-end. The Anchor Protocol *and* The Bonded Commons *(Owens 2026) supply the cryptographic and economic primitives this paper extends. Either of those can be read first or in parallel; this paper is the federation half. The Reader's Map below provides shorter routes for specific audiences.*

PORT DADDY COORDINATION PAPERS **CHAPTER 8 OF 8**

Part I Ground Truth	1. The Single-Writer Kernel · 2. The Anchor Protocol · 3. The Sealed Harbor
Part II The Cost of Seeing	4. The Legible Swarm
Part III What Survives the Restart	5. From Spawn to Person
Part IV Trade Between Strangers	6. The Harbor Economy · 7. The Bonded Commons · **8. The Federated Harbor** (this chapter)

The Harbor, the Person, and the Economy reads in this order: each chapter stands on the ones before it, and each proof chapter follows the chapter whose promises it keeps. Every chapter is independently readable.

Reader's Map

The paper has a topology: it opens with the gestalt (§1–§2), then moves through three governing primitives in order of cryptographic depth (capability transfer, revocation, settlement), then steps back to admission, failure modes, and limitations. The figures are designed to be readable in sequence without the surrounding prose; if you want the five-figure summary, that path is given below.

Layer	Critical artifact	Lives at
(1) Topology	Two harbors, witness log, settlement escrow as a four-element gestalt	§2; Table 1
(2) Cap. transfer	Four-message ceremony; ProVerif model of cross-realm authenticity & attenuation, *conditional on a witness-log freshness assumption* (§4.5)	§4; Figure 2
(3) Revocation	Conditional epidemic-dissemination expectation; partition caveat	§5; Figure 3
(4) Settlement	Recipient-whitelisted custody condition; bucket-partition Conservation	§6; Figure 6
(5) Threat bands	What is mechanized, what is bounded, what is open	§12; Figure 1

Suggested reading paths by reader type:

- **Beginner / new to the series.** Read §1 (the two-machine problem) and §8 (Alice and Bob, end-to-end). The worked example is written to be read cold. Look at the five figures in order. Skip the proofs on the first pass.
- **Distributed-systems engineer.** §2 (the topology), §4 (transfer ceremony), §5 (gossip and equivocation). *The Anchor Protocol* for the cryptographic substrate. Then §12 for the honest open questions.
- **Formal-methods reviewer.** Jump to §9 (the TLA+ and ProVerif extensions) and Appendix A (the verification status table). Threat bands in Figure 1 are the most concentrated signal of what is and is not mechanized.
- **Cryptoeconomic / mechanism-design reader.** §6 (bounded escrow) and §7 (cold-start admission via bonded sponsorship). The proposal's equilibrium contributions are stated here as open work; that is honest, not coy.

- **Security-engineering reviewer.** §3 (threat model) first, then the three primitive sections in order. §10 for the operationally interesting cases the protocols do not close.

Volume Context. *The Anchor Protocol* formalizes process identity on a single machine. *The Bonded Commons* establishes pre-transactional commons authority and collateralized incentive alignment. This chapter extends both into the cross-machine, cross-organization regime. Specific primitives are cited directly (e.g., Anchor §2.4, Bonded §2) rather than re-derived.

Contents

1 Introduction: The Two-Machine Problem

A demonstration is scheduled for 4pm. Alice runs the back end of the demo on her laptop; Bob runs the front end on his. Each has a working Port Daddy daemon: capability tokens are minted and attenuated under the Anchor Protocol [1], the workspace history is Merkle-chained and replicated locally per the Bonded Commons [2], and bonds for cleanup of botched migrations are posted in the local collateral pool. Inside each laptop, the trust story is closed and the formal-verification artifacts hold.

The demo nevertheless does not work. Three failures, none of them inside either machine:

1. Alice's agent obtains a `db:write` card for the staging database, attempts to use it against Bob's daemon, and is refused. Bob's daemon has never seen Alice's signing key. The card is well-formed under the Anchor Protocol, but Bob has no root of authority for it; from his daemon's point of view, the card is gibberish.

2. Five minutes before the demo, Alice's operator revokes the `db:write` card after spotting that it has been over-attenuated. The revocation propagates within Alice's mesh in under a second by the gossip protocol of Anchor §4. Bob's harbor never hears about it. The card remains live on Bob's side.

3. The migration runs anyway, lands in Bob's staging database, and corrupts the demo schema. Alice posted a bond in her harbor against exactly this case. The bond sits in her harbor's collateral pool. Bob's harbor measured the damage. Neither harbor can, on its own, route the bond to where the damage is.

Each failure is a different facet of the same underlying gap. Both daemons are internally consistent. Neither is a root of authority for the other. The single-machine sovereign of the prior papers does not extend to the case of two machines under two operators with no shared administrative parent.

We call this the **two-machine problem**. The point of stating it this baldly is that the failure modes are not exotic; they are what every developer who has tried to dovetail two local agent fleets on a shared task has seen. The bug is structural. The bug is not in either daemon. The bug is in the assumption that one daemon's authority extends past the machine boundary by default. It does not, and it should not.

> *The trick is not to extend a single sovereign across machines. The trick is to admit there are two and to specify what they owe each other.*

This paper extends the Anchor Protocol, the Bonded Commons, and the daemon-as-correlation-device argument to the multi-administrative-domain case. We are not introducing a new substrate; we are showing that the existing substrate admits a clean federation, and that the federation rules are sharper than they look. Where the prior papers spoke of *the* commons authority, this paper speaks of a *mesh of* mutually witnessing commons authorities, each sovereign inside its own machine, none sovereign over the others, and all of them bound to the same audit substrate.

1.1 What this paper is and is not

This chapter closes the book. *The Anchor Protocol* supplies the cryptographic-token substrate; *The Bonded Commons* supplies the economic and governance model; this chapter specifies the federation boundary. The dependencies are concrete: §4 extends Anchor's delegation chain across a trust boundary; §5 extends its revocation mesh; §6 extends Bonded's Conservation invariant.

What this paper is *not*: it is not a permissionless-blockchain redesign. Port Daddy daemons do not run consensus. The Federated Harbor explicitly rejects the assumption common to most

modern federation proposals [28, 27] that a shared ledger is the right substrate. The cost of that simplification is that we cannot lean on a chain for ordering, equivocation detection, or settlement; we must construct each of those primitives from peer gossip plus a federated witness log. The advantage is that the design composes cleanly with the local-first, single-operator-machine architecture of the prior two papers.

1.2 Contributions

The contributions are concrete. We separate them by mechanization status: which are theorems, which are bounded threat models, and which are sketches.

1. **The four-element federation topology (§2).** A precise statement of the boundary between intra-harbor and inter-harbor scope. Defines harbor sovereignty as a property characterization in the style of Bonded's Admissible KMS, so that subsequent claims can be checked against it.

2. **The inter-harbor capability transfer ceremony (§4).** A four-message protocol exchanging a Harbor Card issued by A for an attenuated card valid at B, without either daemon trusting the other's signing key as a root. The ceremony binds the issuing harbor's current epoch root into the envelope, which is the substrate that revocation in §5 hooks into. ProVerif model in Appendix B (status: PARTIAL, model drafted; soundness pending mechanization).

3. **Federated revocation mesh (§5).** The multi-administrative-domain extension of Anchor's revocation gossip. Each harbor publishes an epoch root to a witness log. Under an explicit complete-overlay, reliable-round model, epidemic dissemination takes expected $\Theta(\log m)$ rounds; no finite worst-case bound survives an unbounded partition (Theorem 5.1).

4. **Cross-harbor settlement (§6).** A conditional protocol for clearing a bond posted at A against damage measured at B. The escrow's role is structurally bounded only when the custody ledger, not merely a signed promise, restricts recipients, fee, and terminal transitions. The construction is trusted and specified, not an HTLC-style trustless swap [16].

5. **Cold-start admission ceremony (§7).** A protocol for a new harbor joining an existing mesh without prior reputation, using a bonded-sponsor pattern: an existing harbor posts a bond on the newcomer's behalf, which is forfeit on misbehavior in a probationary epoch. This is the federation-layer analogue of competitive insurance from Bonded.

6. **Open problems (§12).** Five named open questions: (a) whether the correlated-equilibrium reframing of Bonded survives the multi-principal setting cleanly; (b) whether bonds can settle without a trusted, conditionally restricted custody party; (c) whether the folk-theorem cartel-resistance argument from Bonded weakens at the federation layer, and by how much; (d) whether bonded sponsorship can avoid invitation-only capture in a thin cold-start market; and (e) what deployment-specific dissemination tail bound is justified. We state these as open, not as solved.

The proposal that scoped this paper [3] listed two further contributions (a formal definition of harbor sovereignty, and equilibrium results across federations co-authored with Youle) which are present here in skeletal form and will become central in a subsequent revision once the open problems above resolve. We will not claim a result we do not have.

2 The Federated Authority

The Bonded Commons paper defined a *commons authority* as a pre-transactional trust infrastructure with sovereign scope inside a single machine. Inside that scope, the daemon is the unique correlation device that turns mutually-suspicious agents into a correlated equilibrium [2]. The single-machine sovereign was an honest simplification. We now drop it.

The federation design has four elements (Table 1): two sovereign harbors, a witnessed epoch-root log, and a settlement escrow. The escrow is structurally bounded only under the custody assumptions in §6.1. Every primitive in the rest of the paper attaches to one of these elements.

Table 1: Federation is four independent rails between two sovereign harbors, each keeping its own signing root; there is no shared root. The witness rail correlates and exposes equivocation but authorizes nothing; the capability and revocation rails carry attenuated authority and later invalidation in opposite directions; the settlement rail relates a bond posted at A to damage measured at B, and its custody bound is conditional on the settlement gate, not an invariant. No rail makes either harbor a root of authority for the other. [internal]

Rail	What crosses	Direction	What it does not do
witness	the signed epoch roots $R_A^{(e)}, R_B^{(e)}$ and any equivocation evidence	both ways	authorize work: correlation, not authorization
capability	an attenuated card, $C_A \mapsto C'_B \subseteq C_A$	A to B	widen scope or lifetime at any hop
revocation	the anti-entropy delta of revoked ids	B to A	promise a finite deadline under partition
settlement	a bond posted at A against damage measured at B	both ways	bound custody unconditionally: it holds only if the settlement gate holds

2.1 Harbor sovereignty as a property characterization

In Bonded, the Admissible KMS was characterized by a list of properties rather than by a specific construction (per-machine signing key, per-machine SQLite, etc.) so that the argument carried over to any concrete instantiation. We adopt the same posture here.

Definition 2.1 (Harbor Sovereignty)**.** A daemon D_A is *sovereign over harbor A* if it has exclusive authority over five things:

1. **Issuance.** Only D_A can mint a Harbor Card whose root signature verifies under A's signing key sk_A.
2. **Attenuation.** Only D_A can apply a local attenuation predicate att_A that restricts an inbound capability before it is verified by an A-side agent.
3. **Revocation.** Only D_A can decide that a card it issued is revoked. The decision becomes globally visible by the gossip protocol of §5; D_A does not control when other harbors learn of it, but it controls whether the revocation event exists.
4. **Acceptance.** D_A alone decides which inbound cards from other harbors to accept and under what local policy. No external party can compel acceptance.
5. **Evidence binding.** The epoch root $R_A^{(e)}$ that D_A publishes to the witness log is signed under sk_A and binds the entire A-side state of issued, attenuated, and revoked cards up to epoch e. No other harbor can publish a root that purports to be A's.

This is a deliberately narrow definition. It does *not* say that D_A is sovereign over everything an A-side agent does; the agent may be operating against a database hosted at B, in which case B-side policy applies and D_A has nothing to say about it. Sovereignty is over the *issuance and revocation surface of A's own cards*, not over downstream effects.

We will use Definition 2.1 as the discipline against which subsequent protocols are checked. The transfer ceremony of §4 produces a card valid at B; if it required B to verify under sk_A

in the hot path, it would violate B's sovereignty by making B's acceptance contingent on an external authority. We will see that it does not.

2.2 Inter-harbor scope

The dual notion is equally important. *Inter-harbor scope* is the scope of operations whose effects cross the trust boundary between sovereign harbors.

Definition 2.2 (Inter-Harbor Scope)**.** An operation o has *inter-harbor scope* (A, B) if its execution causes a state change observable to B under authority that originated at A. Examples: an A-issued card used to write to a B-hosted database; a settlement clearing a bond posted at A against damage measured at B; a revocation issued by A becoming visible to a B-side verifier.

Most operations are *intra-harbor*: minting, verifying, attenuating, and revoking cards under one harbor's authority, observed by agents under that same harbor. The Anchor Protocol's proofs cover the intra-harbor case end-to-end. The Federated Harbor's contribution is exactly the inter-harbor surface: every protocol that determines its guarantees lives at the boundary.

2.3 The witness log

The witness log is the device that makes the federation auditable without making any harbor sovereign over the others. It plays the same role that the Merkle Forest plays inside a single machine in *The Bonded Commons* (§4.1): a tamper-evident accumulator that any third party can verify without needing to trust the publisher.

Design invariant 2.1 Witness-Log Admissibility. A witness log W is admissible for the Federated Harbor if it provides:

1. **Append-only.** Once W records an entry, W cannot retract or reorder it.
2. **Per-harbor latest-root binding.** For each harbor X in the federation and each epoch e, W records at most one root $R_X^{(e)}$ signed under sk_X.
3. **Auditable consistency.** Any third party can verify, given $R_X^{(e_1)}$ and $R_X^{(e_2)}$ with $e_1 \le e_2$, that $R_X^{(e_2)}$ is a Merkle-consistent extension of $R_X^{(e_1)}$.
4. **Equivocation evidence.** If an auditor obtains two differently signed roots for the same harbor and epoch, it can verify the contradiction immediately. Disseminating both views to a common auditor depends on the overlay and partition model (Theorem 5.1).
5. **Threshold publishability.** The act of publishing $R_X^{(e)}$ to W does not require D_X to trust any other party; the cross-signatures collected from other daemons in the federation are an auditor convenience, not an authorization gate.

This list is intentionally close in shape to the five properties of the Admissible KMS in *The Bonded Commons* ("Federated Sovereign"). The Bonded paper used the same posture — characterizing the property surface, then exhibiting a concrete construction that satisfied it — and we follow that pattern. Section 9 sketches one concrete instantiation built on Certificate Transparency-style append-only logs [10] cross-witnessed in the Sigstore/Rekor pattern [11]; the substrate need not be unique.

2.4 What the federation is not

It is worth saying what the federation is not, so that the rest of the paper does not have to keep refuting positions it never took.

The federation is *not* a consensus protocol. There is no global agreement on the order of events; there is per-harbor monotonicity (epoch roots only extend), there is cross-harbor witnessing (each root is published to a shared log), and there is detectable equivocation. None of these require harbors to agree on a global order, and the prior papers' designs explicitly avoid that overhead.

The federation is *not* a hub-and-spoke topology with a privileged coordinator. The witness log is replicable; in practice a small set of mutually-witnessing log instances suffices, and any harbor can read from any of them. The hub-and-spoke failure mode — one well-funded coordinator captures the federation by becoming everyone's only witness — is a real risk we discuss in §10, and the structural pushback is gossip-based witness redundancy.

The federation is *not* a permissionless market in identities. New harbors enter through the admission ceremony of §7, which requires either a pre-existing bond or an existing harbor sponsor. This is by design; the cost is that the federation is invitation-bounded, the benefit is that Sybil attacks against the federation [17] cost the attacker as much per identity as honest entry does.

Recall.

1. Without looking: name the five things Definition 2.1 says a sovereign daemon D_A has exclusive authority over.
2. Why does Definition 2.1 say sovereignty is *not* violated when an A-side agent operates against a B-hosted database under B's policy?
3. Name the three things this chapter says the federation explicitly is *not* (§2.4), and the one mechanism each rests its pushback on.

3 Threat Model

We make the threat model explicit before introducing the protocols so that the design choices can be checked against named adversary powers. We follow the Dolev–Yao convention [8] extended to the federation case: an attacker controls every wire between harbors, can intercept and replay any cross-machine message, and may collude with any bounded number of harbor operators. The attacker cannot break the underlying cryptographic primitives (Ed25519 signatures, SHA-256 hashes); these assumptions are inherited from Anchor.

3.1 In-scope attacks

The protocols in this paper are designed against the following attackers.

Cross-realm identity spoofing. An attacker presents a card at B claiming issuance by A, using a forged signature under sk_A. *Closed by* signature verification against A's public key as recorded in the witness log (§4). Reduces to Ed25519 EUF-CMA security [9].

Replay across the boundary. An attacker captures a valid xfer_offer from A to B at epoch e and replays it at epoch $e + k$. *Closed by* the binding of $R_A^{(e)}$ into the envelope; the verifier at B checks that R_A is the current root, and a stale envelope is rejected (§4). The window between issuance and the next epoch is the structural attacker window, named and bounded.

Capability inflation across transfer. An attacker presents at B a card C'_B that claims authority beyond the attenuation predicate applied at A. *Closed by* the requirement that the verifier at B recompute the composed attenuation $\mathrm{att}_B \circ \mathrm{att}_A$ from the envelope rather than trust an intermediate claim (§4); reduces to Anchor's intra-harbor attenuation invariant.

Stale-revocation acceptance. An attacker at B presents a card revoked at A but not yet observed at B. Under the reliable connected-overlay model the window has expected scale $\Theta(\Delta \log m)$; during a partition it is unbounded. A bond can price a configured service-level window, not eliminate the tail.

Cross-witness equivocation. A malicious daemon D_A publishes contradictory roots to isolated witnesses. The contradiction is cryptographically verifiable once a common auditor receives both views, but the time to that event is topology- and partition-dependent. A witness bond prices confirmed equivocation; it does not create a delivery deadline.

Settlement redirection or equivocation. A signed escrow promise makes misconduct attributable. Prevention requires a custody ledger whose transition API cannot name an arbitrary recipient or fee. Design Invariant 6.1 states the bound conditionally on that mechanism; without it, the full bond remains at risk.

Harbor membership forgery at the federation boundary. An attacker presents a card from a harbor purporting to be in the federation but that has no admission record. *Closed by* the admission ceremony (§7) which produces a permanent, witness-logged enrollment record co-signed by an existing harbor's sponsor bond. A claimed harbor with no enrollment record is treated as outside the federation regardless of how well-formed its cards appear.

3.2 Out-of-scope attacks

We are explicit about what is not closed.

Compromise of a harbor's signing key. If sk_A leaks, the attacker can issue arbitrary A-side cards until A rotates the key and publishes the rotation to the witness log. We inherit Anchor's key-rotation procedure unchanged. The federation does not close the underlying primitive; it does bound the blast radius to A's sovereign surface, which is exactly the right scope.

Sybil at the federation layer. An attacker who can pay the admission cost m times can run m federated harbors. The cost ratio between honest and adversarial admission is the decisive parameter; we discuss it in §7 but do not give a tight bound. This is named open work.

Cartel formation across harbors. Bonded's folk-theorem cartel-resistance argument relied on the daemon as a correlation device inside one machine. Across harbors, cartels can form offline and present a unified front to the witness log; the witness log records that all harbors agreed, which is consistent with both honest agreement and cartel collusion. We acknowledge this and state the open work in §12.

Centralization gravity. Historically, federated identity systems concentrate on one hub within five to ten years [18]. The Federated Harbor's structural pushback — replicable witness logs, no central coordinator — is design discipline, not theorem. We formally classify this in §10.

Side channels. Constant-time signature verification, cache-side-channel resistance, and timing-attack resistance are inherited from the underlying libraries; we do not re-prove them and we note where we trust them.

3.3 Threat-band visualization

Figure 1 lays out the threat surface as three bands, distinguished by weight rather than hue so the figure survives greyscale and colorblind rendering alike: **Stopped** (mechanized claims, heaviest fill), **Bounded** (a named threat inside a named window, lighter fill), and **Open** (acknowledged frontier, drawn in cobalt outline as the one band with no formal artifact behind it). The boundary between the Bounded and Open bands shifts left over time as future work mechanizes what is presently only bounded. We will refer back to this figure when discussing limitations.

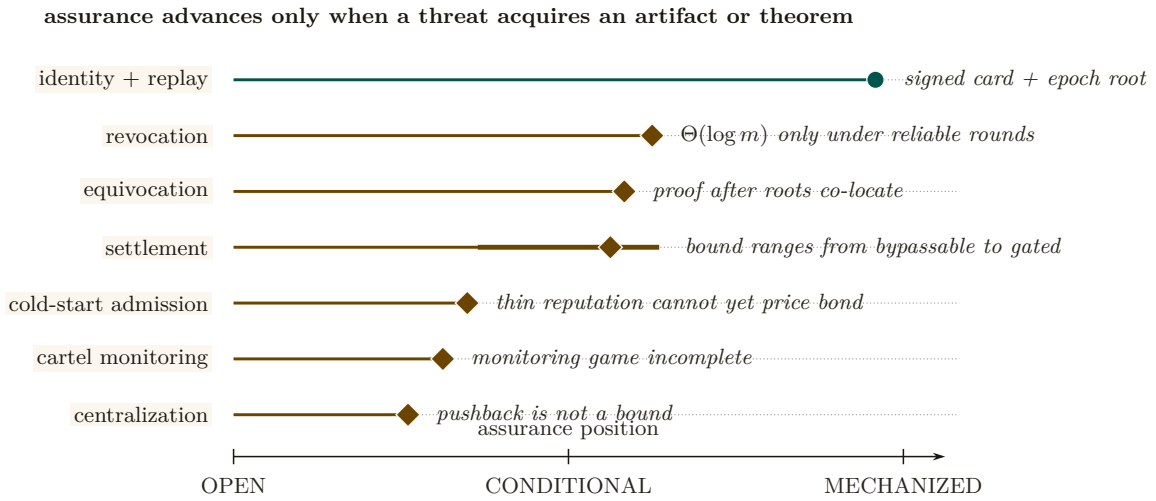


Figure 1: The assurance landscape is uneven and falsifiable. Position encodes how far each threat has moved from an acknowledged gap toward a mechanized artifact. Identity and replay have the strongest concrete evidence. Revocation, equivocation, and settlement are conditional on named connectivity or custody hypotheses; the thick settlement interval shows how widely its assurance moves if the gate is bypassable. Cold start, cartel monitoring, and centralization remain open. A row advances only when the missing artifact or theorem exists.

4 Cross-Harbor Capability Transfer

The first governing protocol is the cross-harbor capability transfer ceremony. An A-side agent holds a Harbor Card C_A minted by D_A under Anchor Phase 2 or Phase 3. It needs to present an equivalent capability at B. The Anchor Protocol's intra-harbor delegation already supports attenuation; we extend that to the case where the attenuation crosses the trust boundary.

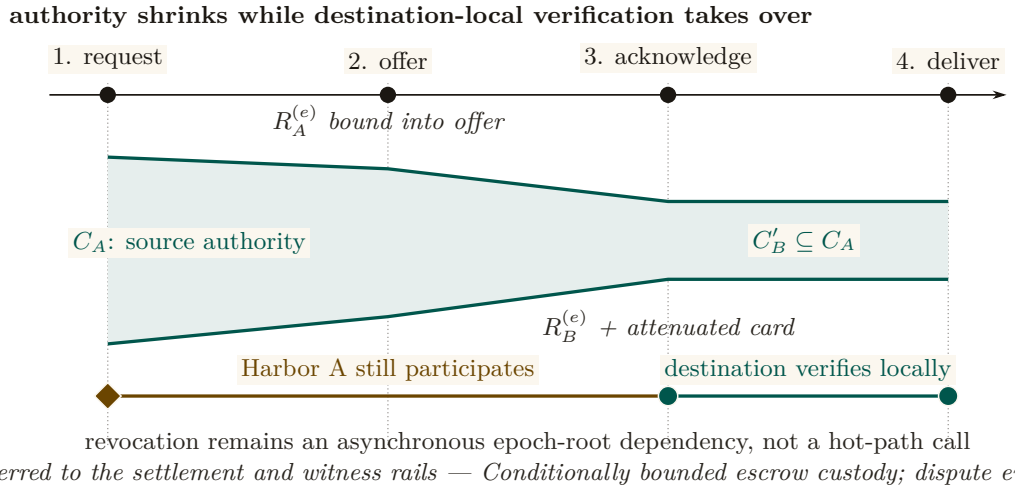


Figure 2: The transfer ceremony attenuates authority and then removes source reachback from the hot path. The green envelope narrows from C_A to $C'_B \subseteq C_A$ as the signed offer and acknowledgement bind source and destination epoch roots. After message 3, Harbor B can verify and deliver the attenuated card locally. Harbor A still matters asynchronously: a later epoch root can revoke the transfer once that root reaches B's audit surface. Transfer therefore neither copies full authority nor makes every use synchronous.

4.1 The four-message ceremony

Figure 2 shows the protocol as a sequence diagram. The four messages are:

1. $\mathsf{xfer\_req}(C_A, B)$ — Alice's agent presents C_A to its local daemon D_A and names B as the target harbor. The request is intra-harbor; standard Anchor verification applies.
2. $\mathsf{xfer\_offer}\langle C_A, \mathrm{att}_A, R_A^{(e)}\rangle$ — D_A constructs an envelope binding the card, an attenuation predicate att_A specifying what subset of C_A's authority is being offered for cross-realm use, and its current epoch root $R_A^{(e)}$. The envelope is signed under sk_A and sent to D_B.
3. $\mathsf{xfer\_ack}\langle C'_B, R_B^{(e)}\rangle$ — D_B verifies the envelope's signature against A's public key as recorded in the witness log; verifies that $R_A^{(e)}$ is the current root for A; applies its own local attenuation att_B; and mints $C'_B = \mathrm{att}_B(\mathrm{att}_A(C_A))$, a card valid only at B, signed under sk_B. The acknowledgment is sent back to D_A along with B's current epoch root.
4. $\mathsf{xfer\_deliver}\langle C'_B\rangle$ — D_B delivers C'_B to the agent at B that will use it. Subsequent presentations of C'_B at B verify under sk_B alone — no A-side lookup is needed at the hot path.

Each derivative C'_B carries the transfer identifier of the offer that produced it, and D_B mints at most one derivative per identifier, so a replayed $\mathsf{xfer\_offer}$ cannot produce a second C'_B; the identifier is part of what the counter-signature covers. The critical property is that after message (3), no further synchronous interaction with A is required for the agent to operate at B. The cost is that revocation of C_A at A does not instantaneously revoke C'_B at B; Theorem 5.1 gives an expectation only under its delivery model, and partitions require fail-closed policy or bounded credential TTLs.

4.2 Why this composes cleanly with Anchor

The Anchor Protocol's Phase-3 delegation already proves that a chain $C_{\text{daemon}} \to C_A \to C_B$ of attenuated cards is verifiable under the issuing daemon's public key, and that an attacker cannot inflate a downstream cap to exceed an upstream one. The transfer ceremony is Phase-3 delegation across a trust boundary, with two changes:

1. The second link in the chain is signed under a *different* root key (sk_B instead of sk_A), so the verifier at B verifies under sk_B, not sk_A.

2. The cross-realm link binds the issuing harbor's current epoch root $R_A^{(e)}$, so a revocation at A after epoch e invalidates the downstream card as soon as the new $R_A^{(e+1)}$ is observed at the verifier.

The composition is what makes the ceremony cheap to add: we are not introducing a new cryptographic primitive, only a new policy at the boundary that says "the verifier accepts a card whose chain crosses the trust boundary if and only if (a) each cross-realm link is signed under the receiving harbor's key and (b) the issuing harbor's bound epoch root is the current one in the witness log."

4.3 Attenuation across the boundary

Two attenuation predicates are applied in sequence. att_A is set by D_A at issue time and reflects what authority A is willing to project across the boundary. att_B is set by D_B at acceptance time and reflects what authority B is willing to grant on its own resources to a card whose root is A. Both attenuations strictly narrow the underlying capability.

Theorem 4.1 Attenuation Monotonicity Across the Boundary. For any cross-realm card $C'_B = \text{att}_B(\text{att}_A(C_A))$, the set of operations authorized by C'_B is a subset of those authorized by C_A under the federated authorization rule.

Sketch. Each attenuation is a subset operation in the Anchor algebra; subset composition is a subset. Cross-realm policy at B adds the constraint "and the operation must be permitted by B's local policy," which can only further restrict. See the Anchor accompanying chapter's capability-attenuation section for the intra-harbor argument; the cross-realm case adds no new mathematics, only a new applicable attenuation step. □

Exercise 8.1, page 34.

4.4 Threats foreclosed and threats deferred

The annotation band at the bottom of Figure 2 summarizes which threats this ceremony forecloses and which are deferred to other primitives. Specifically:

- **Foreclosed:** identity spoofing of sk_A across the boundary; capability inflation across the transfer; replay at B against a stale R_A.
- **Deferred:** revocation latency (handled in §5); settlement disputes (handled in §6); admission of unbonded new harbors (handled in §7).

We are explicit that the ceremony does not close every threat in one step. Federation is a layered defense; this is one layer.

4.5 Mechanization status

We have a draft ProVerif model of the four-message ceremony (Appendix B), extending the Anchor Phase-3 model with a second signing key and an epoch-root binding. The current status of the cross-realm authenticity query is PARTIAL: the model checks, but the proof depends on an assumption about witness-log freshness that we are still tightening. We note this limitation in Appendix A and treat the attenuation monotonicity claim (Theorem 4.1) as proved under the same caveat.

Recall.

1. Without looking: what does message (2) of the four-message ceremony bind, and why does that binding matter more than any single signature check?

2. What is the current mechanization status of the cross-realm authenticity query, and what specific assumption is it still conditional on?
3. Name one threat the ceremony forecloses and one it defers, and which later section closes the deferred one.

5 Federated Revocation

The second governing protocol is the federated revocation mesh. The Anchor Protocol handles revocation inside a single mesh by gossiping an **Append-Only Event Log of Revocation IDs** using Vector Clocks for anti-entropy reconciliation [15]: pairs of daemons periodically reconcile their event logs, designating the cuckoo filter F purely as a local, ephemeral read-path index. The Federated Harbor extends this to the multi-administrative-domain case.

Two pieces are added on top of the Anchor mechanism: (a) per-epoch roots $R_X^{(e)}$ published to a shared witness log so a third party can audit any harbor's local view, and (b) cross-witness signatures so the auditor pattern generalizes Certificate Transparency [10].

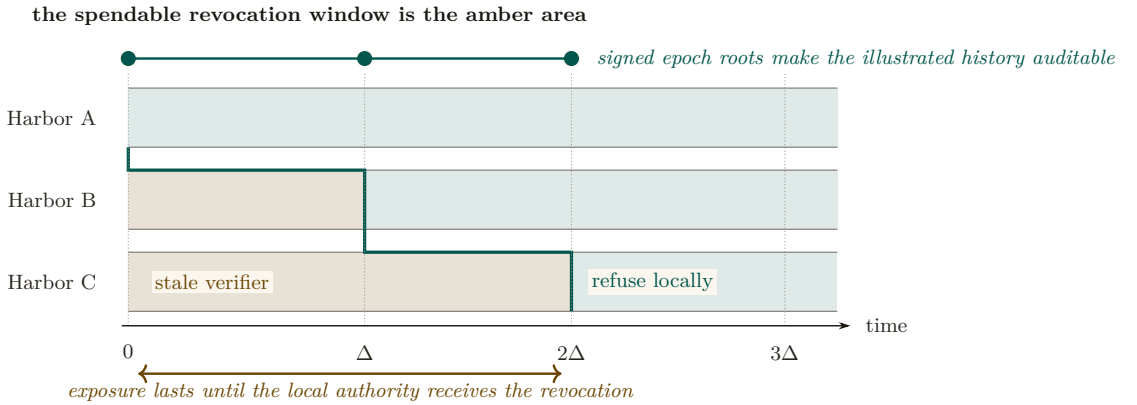


Figure 3: Revocation propagation is a moving staleness frontier. Alice refuses immediately; B remains stale until Δ and C until 2Δ in this illustrated execution. The amber area is precisely where a revoked capability can still be admitted by the named local authority. The teal staircase marks synchronization, not a global clock. Expected $\Theta(\log m)$ dissemination requires connected reliable rounds; a partition can extend the amber area without bound and therefore destroys any finite deadline.

5.1 Convergence bound

Standard epidemic dissemination gives an expected $\Theta(\Delta \log m)$ completion time only under an appropriate connected, reliable-round model [15]. The property below instantiates that model at the federation level, treating each harbor's filter as one node in a complete overlay regardless of how many local daemons participate in the harbor's mesh. It is a model-conditional expectation, not an inherited wall-clock deadline.

Theorem 5.1 Conditional Revocation Dissemination. Let m harbors form a complete overlay. In each reliable synchronous round of duration Δ, informed harbors contact independently uniform peers and exchange revocation deltas. If harbor X publishes revocation c at t_0, then:

1. **Revocation dissemination.** All harbors are informed in expected $\Theta(\log m)$ rounds under the stated epidemic model.
2. **Equivocation verification.** Any auditor possessing two differently signed roots for the same (X, e) detects equivocation in $O(1)$ signature and equality checks.
3. **No hard deadline.** If delivery can be lost indefinitely or the overlay can remain partitioned, no finite worst-case dissemination or common-auditor detection bound exists.

The first item is the standard asymptotic epidemic-dissemination result [15]; this paper does not claim the earlier exact constant $1 + \ln m$. The second follows from the signed-root format. The third is an indistinguishability result: an auditor isolated in one partition cannot distinguish equivocation from delay.

Corollary 5.1 (Service-Level Attacker Window)**.** If an operator additionally imposes a partition bound T_p and a delivery service level T_g, then stale-card exposure is bounded by $T_p + T_g$. Without those operational assumptions the protocol supplies an expected latency under its model, not a structural maximum.

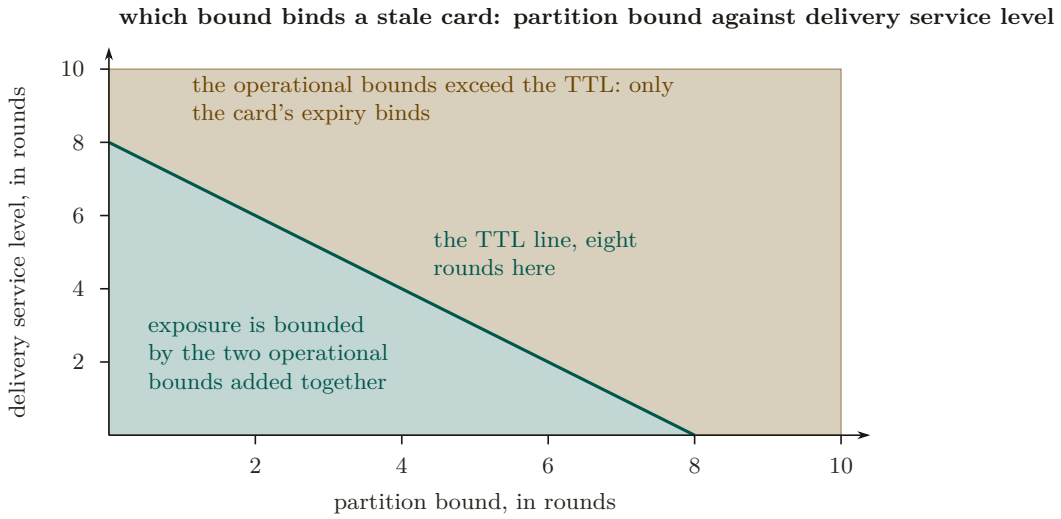


Figure 4: Which bound binds a stale card after revocation. Under a partition bound and a delivery service level, exposure is at most the two added together (Corollary 5.1); once that sum exceeds the card's time to live, the expiry binds instead and the harbor fails closed. Without a partition bound the horizontal axis is unbounded and only the expiry remains (Theorem 5.1, item 3); the line is drawn for a time to live of eight rounds.

The finality that "only-issuer-revokes" rests on is model-checked with a negative control. Under the rollback configuration the relay's revocation set is restored from a backup taken before the revocation, and TLC finds the four-step breach; the epoch configuration closes it (§2.3).

At the terminal. *the rollback configuration fails by design.*

```
$ java -cp tla2tools.jar tlc2.TLC -config CardRevocation_rollback.cfg CardRevocation.tla
Error: Invariant RevocationFinal is violated.
Error: The behavior up to this point is:
State 1: <Initial predicate>
/\ revoked = {}
/\ cardEpoch = (c1 :> 0 @@ c2 :> 0)
/\ epoch = 0
/\ badAccept = FALSE
/\ everRevoked = {}
/\ issued = {}
/\ rbUsed = 0

State 2: <Issue(c2)>
/\ revoked = {}
/\ cardEpoch = (c1 :> 0 @@ c2 :> 0)
/\ epoch = 0
/\ badAccept = FALSE
```

```
/\ everRevoked = {}
/\ issued = {c2}
/\ rbUsed = 0

State 3: <Revoke(c2)>
/\ revoked = {c2}
/\ cardEpoch = (c1 :> 0 @@ c2 :> 0)
/\ epoch = 1
/\ badAccept = FALSE
/\ everRevoked = {c2}
/\ issued = {c2}
/\ rbUsed = 0

State 4: <RollbackRestore(c2)>
/\ revoked = {}
/\ cardEpoch = (c1 :> 0 @@ c2 :> 0)
/\ epoch = 1
/\ badAccept = FALSE
/\ everRevoked = {c2}
/\ issued = {c2}
/\ rbUsed = 1

State 5: <Accept(c2)>
/\ revoked = {}
/\ cardEpoch = (c1 :> 0 @@ c2 :> 0)
/\ epoch = 1
/\ badAccept = TRUE
/\ everRevoked = {c2}
/\ issued = {c2}
/\ rbUsed = 1

35 states generated, 23 distinct states found, 9 states left on queue.
The depth of the complete state graph search is 5.
Finished in 00s
```

The distinction determines the exposure bound. A bond can be priced against an operator-declared service-level window, but an unbounded partition creates an unbounded exposure tail. Deployments must therefore combine bond sizing with expiry, fail-closed acceptance, or an explicit partition limit.

5.2 Conflict resolution across harbors

The interesting case is disagreement. Harbor A has revoked card c; harbor B has not yet seen the revocation; an agent presents c at B for an action that affects A's database. Three candidate resolution rules from the proposal [3], evaluated:

1. **Pessimistic (anyone-revokes-is-revoked).** The card is revoked iff any harbor has revoked it. Defended against by an attacker who would gain by injecting spurious revocations into the gossip mesh; rejects too many honest cards in equilibrium.
2. **Authoritative (only-issuer-revokes).** Only A can revoke A's cards. This is the rule we adopt. Under the service-level assumptions of Corollary 5.1, it gives an operator-priced attacker window; without them, exposure has no finite structural maximum.
3. **Epoch-bounded (revocation propagates within Δ epochs and inconsistency is a contract violation).** A refinement of (2) for the case where one harbor falls persistently behind. We treat this as a degraded mode of (2) rather than a separate rule.

We adopt rule (2). The pessimistic rule has the wrong defaults; the epoch-bounded rule is operationally the same as (2) with a watchdog. Adopting (2) means the Conservation invariant of §6 can be stated cleanly.

Exercise 8.2, page 35.

5.3 Event Log & Ephemeral Cuckoo Discipline

Two Bloom filters over the same universe merge by a bitwise OR; two cuckoo filters cannot, because each stores a fingerprint in one of two hashed buckets that can overflow, so a union may have nowhere to place a fingerprint. The cuckoo filter is therefore purely a local, ephemeral read-path index derived from the authoritative Append-Only Event Log. The filter retains local Anchor-side disciplines (load capping, pollution resistance) but cross-harbor validation is strict: a filter at B that holds $c \in F_B$ where c was revoked at A is correct only if there is a corresponding inclusion proof in $R_A^{(e+1)}$ in the witness log. False positives are corrected by rebuilding the local filter from the event log; spurious revocations injected by an attacker into a peer's log are rejected on first attempt to verify against the witness log. The discipline is: *local filters are advisory read-caches; the event and witness logs are authoritative.*

5.4 Sheaf Laplacian and Abramsky–Brandenburger Contextuality

Three harbors in a ring each record the offset between their own log and the neighbor they last compared with. Walk the ring and add the offsets: if the sum returns to zero, one consistent history explains every reading; if it does not, some offset was misreported, and the size of the residual says by how much. That is the whole picture; the formalism below states it for arbitrary topologies.

The global consistency of federated witness logs can be modeled sheaf-theoretically, and it is worth being precise about which object carries the weight. The intuitive picture is a poset $(X, \leq)$ of administrative domains ordered by scoped visibility, assigning to each domain U the semilattice of prefix-compatible append-only logs, with restriction given by prefix projection. That picture is right about the ordering and useless for computation: a semilattice has no subtraction, no adjoint, and no spectrum, so none of the machinery below would be defined over it. The working object is therefore the *cellular sheaf* $\mathcal{F}$ of the cohomology of equivocation,[1] carried on the gossip graph $G = (V, E)$ of witnesses: witness v holds a real stalk $\mathbb{R}^{d_v}$ encoding its log state; edge $e = \{u, v\}$ holds $\mathbb{R}^{|S_e|}$, the shared prefix coordinates S_e its two endpoints both report; and the restriction maps $\rho_{v,e}$ are the coordinate selections that cut a witness's state down to what that edge can see. Over real stalks the coboundary δ, its adjoint δ^*, and the observed disagreement cochain $(\delta s)_e = \rho_{u,e} s_u - \rho_{v,e} s_v$ all exist. The semilattice stays as intuition; the vector-space sheaf is what the theorem below quantifies over.

Detection is stated per visibility tier, because not every edge is equally visible to an auditor. An edge is **compared** when the auditor holds its direct cross-check (pairwise detection, no cohomology needed); **relayed** when both endpoints' reports about it reach the auditor by gossip but the endpoints never reconciled it; and **severed** when no report about it crosses to the auditor at all. Write K for the compared-plus-relayed (i.e. *visible*) edge set, and for each shared coordinate c write $K_c := (V, \{e \in K : c \in S_e\})$ for the visible coordinate subgraph. The distinction between K_c and the full coordinate subgraph G_c changes the theorem, not merely the notation: a cycle that runs through a severed edge imposes no constraint, because the severed block is a free variable of the completion.

Theorem 5.2 Detection iff the missing edge closes a *visible* cycle. Assume prefix restrictions, the three-tier visibility model above, and a **single equivocator** q (the single-equivocator hypothesis is not decoration: two liars on a common cycle can cancel each other's

[1] A standalone, submission-form version of the results folded into this subsection is *The Cohomology of Equivocation* (research paper 7) [4]; the harness and the sweeps cited here are its artifacts, regenerated at seed 20260816.

contribution to the residual). The detection statistic is the least-squares *completion residual*

$$r \;=\; \min_{x,\, g_{\mathrm{sev}}} \big\| \, g_K - (\delta x)\big|_K \, \big\|_2 \;=\; \Big(\textstyle\sum_c \big\| \mathrm{Proj}_{\mathrm{cycle}(K_c)}\, g^c \big\|^2 \Big)^{1/2}, \tag{1}$$

the minimum over global sections x and over the free severed blocks g_{sev}. Then r detects q's split-view lie beyond what pairwise comparison already catches *if and only if* the un-compared lie edge lies on a cycle of the **visible** subgraph K_c and its endpoints' reports are relayed to the auditor; in that case $r > 0$ certifies that no global witness history explains the visible data. On a cut edge, $r = 0$ identically — the coboundary map of a tree or forest is surjective onto the visible data, so the free completion absorbs any lie. Across a severed edge, equivocation is provably dark: the severed block is unconstrained and a consistent counterfactual world always exists.

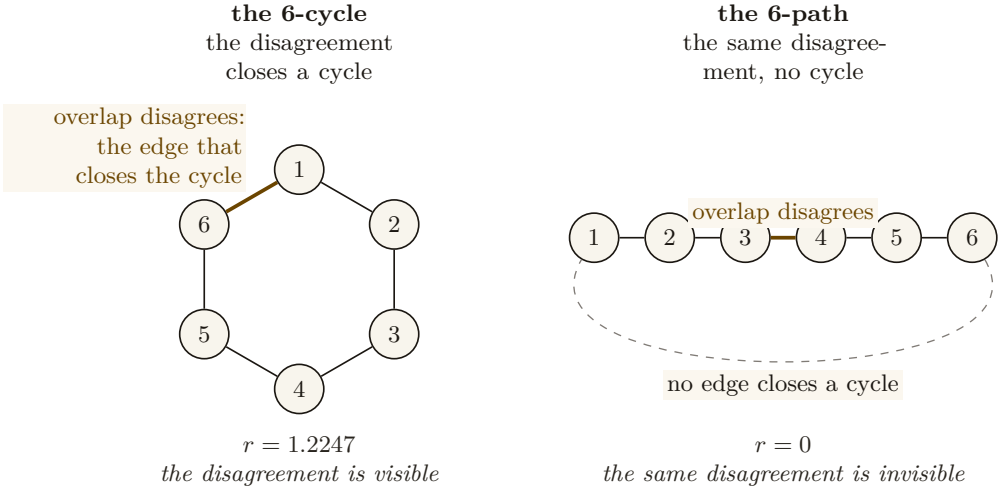


Figure 5: Cycle versus cut on six harbors, numbered 1 to 6. The same pairwise disagreement on one overlap is placed on the cycle C_6 and on the path P_6. On the cycle the disagreeing edge closes a visible cycle and the consistency radius is $r = 3\,(1 - 5/6)^{1/2} = 1.2247$; on the path no edge closes a cycle and $r = 0$, so detection is exactly the visible-cycle condition of the theorem. [verified; `sheaf_consistency_radius.py`, seed 20260816]

The structural identity behind the mechanism — local consistency with no global section — is the Abramsky–Brandenburger contextuality equivalence [34]: an equivocation (a harbor signing two mutually incompatible prefixes) manufactures for its neighbors exactly a locally-consistent, globally-unglueable family of views, the relational-semantics analogue of Bell contextuality. Note what the theorem does *not* say. The textbook fact that $H^1(\mathcal{U}; \mathcal{F}) = 0$ is equivalent to every local family gluing is a statement about the cover and the restriction maps alone; it is blind to any particular lie, and it is not the detector. The detector is the residual r of the observed data.
Scope (what the theorem does and does not license). Three bindings keep this claim inside its proven regime. **(i) The obstruction lives in the data, not the sheaf:** the detector is the observed disagreement cochain's failure to be a coboundary (its harmonic residual), not $\dim H^1$ of the abstract sheaf, which is a property of the restriction maps alone and is blind to any particular lie. **(ii) Visible cycles are required:** cohomology adds power over $O(|E|)$ pairwise digest comparison exactly when an uncompared (partitioned) edge lies on a cycle *of the visible subgraph* K_c, so the sum-around-the-loop constraint substitutes for the missing comparison; across a *cut* edge there is no loop to close, and a loop through a *severed* edge is no loop at all for this purpose, so the method provably sees nothing — redundant gossip paths are a detection requirement, not a luxury. **(iii) Non-vanishing is an alarm; vanishing is not an all-clear:** over real coefficients the obstruction is sufficient but not necessary (the abelianization gap of Carù), so a

zero residual licenses no safety conclusion. Detection and localization here *complement* forensic attribution — signatures attribute, cohomology localizes under partial visibility — and neither replaces the other.

The harness, pre-registered. Before it ran, the harness committed to its gates: the completion residual must detect split-view equivocation on relayed cycles where pairwise comparison is blind, and it must stay silent on honest worlds, cut edges and severed arms. Two hundred trials per arm at seed 20260816, stalks $\mathbb{R}^5$, shared-prefix sizes 2 to 4; the score is cohomology-only detections, trials where the residual fires and every pairwise comparison passes. Partition on a cycle (`two_path`): 200/200 cohomology-only. Severed-cycle arm: 0/200, the boundary measured. Cut edge (`single_bridge`): maximum residual 1.5×10^{-13} over 400 trials, silence at float epsilon. Full visibility: 0 cohomology-only and 200/200 redundant, since pairwise comparison catches everything first. An expander with three severed and three relayed edges: 113/200 cohomology-only and 87 dark, and the dark trials split exactly along the theorem's two clauses: in 79 the equivocator has no relayed incident link at all, so every lie lands on a severed edge; in the remaining 8 the lie edge is relayed and lies on a cycle of G_c but not of K_c, because the rest of that cycle is severed. Those eight are the only trials that distinguish the two readings of the cycle clause, and they are dark, as the K_c form predicts and the G_c form does not. Both pre-registered gates pass; verdict COMMIT. The rival CUT condition (residual detections are pairwise detections in disguise) is refuted: 313 of 1,101 residual detections across arms were pairwise-blind. Mutants reintroducing the two defects of the first harness (non-subset restriction maps that collapse the obstruction space; detection scored on hidden-edge data) are each caught by structural self-checks [internal, `sheaf_harness_v2.py`]. The harness validates the detector at the sharpened scope only: synthetic gossip, prefix restrictions, one equivocator per trial. A statistical pass is not a theorem, which is what the three claims below are for.

Numbers by hand — The cochain by hand: one ring, one chain, one dark link. Four relays $0, 1, 2, 3$ in a ring, scalar stalks, one shared coordinate. The links $0{-}1$, $1{-}2$ and $2{-}3$ are compared and agree; the link $3{-}0$ is never compared, but both its endpoints' reports reach the auditor, so all four links are visible and K_c is the whole ring. Relay 0 equivocates: the head it shows across $3{-}0$ differs by $s = 3$ from the one it shows relay 1. The auditor's cochain, one disagreement per link, is $g = (g_{01}, g_{12}, g_{23}, g_{30}) = (0, 0, 0, 3)$. The only question the detector asks is whether one history $x = (x_0, x_1, x_2, x_3)$ has differences reproducing g. Around a loop a coboundary sums to zero and g sums to 3, so none does. Least squares finds the closest: $x = (0, 0.75, 1.5, 2.25)$, with coboundary $\delta x = (-0.75, -0.75, -0.75, +2.25)$, which misses by exactly 0.75 on every link. That is the mechanism in one sentence: the best honest world spreads the lie evenly around the loop and still misses everywhere. Hence

$$r = \sqrt{4 \times 0.75^2} = 1.5 = \frac{|s|}{\sqrt{n}} = \frac{3}{\sqrt{4}} = |s|\sqrt{1 - R^{K_c}_{\text{eff}}(e)} = 3\sqrt{1 - \tfrac{3}{4}},$$

three routes to one number [verified]. Now break the loop twice and watch the same arithmetic go silent. *As a chain* $0{-}1{-}2{-}3$ with the same lie on the never-compared end link $2{-}3$: the assignment $x = (0, 0, 0, -3)$ reproduces $g = (0, 0, 3)$ exactly, so $r = 0$, and the reader has just built the adversary's alibi by hand. *As a ring with a dark link:* keep the ring and the lie on $3{-}0$ but sever $1{-}2$. Now $g_K = (0, 0, 3)$ on three visible links and $x = (0, 0, 3, 3)$ reproduces it exactly: $r = 0$ again. The loop is still there in G_c; it is gone from K_c, and the residual can only see K_c. This is the theorem's visibility clause, checkable in four numbers. *Now you try:* C_8, lie of size 2 on the relayed edge: $2/\sqrt{8} = 0.7071$; the same lie on any bridge, 0; the same lie on C_8 with any one of the other seven links severed, also 0 [verified].

Numbers by hand — the consistency radius, six-cycle vs. six-path, and the $1/\sqrt{n}$ scaling. Six harbors in a ring (C_6), unit-weight edges, a lie of magnitude $|s| = 3$ on one relayed cycle edge. The effective resistance across that edge is $R_{\text{eff}} = (n-1)/n = 5/6$, so by hand:

$$\begin{aligned} r &= |s|\sqrt{1 - R_{\text{eff}}} = 3\sqrt{1 - \tfrac{5}{6}} \\ &= \tfrac{3}{\sqrt{6}} = 1.2247 \quad \text{[verified]}. \end{aligned}$$

Cut the same ring into a six-node path (P_6), same edge, same lie: every edge of a path is a bridge, so $R_{\text{eff}} = 1$ and $r = |s|\sqrt{1-1} = 0$ exactly [verified] — a tree's coboundary map is already onto the visible data, so the free completion absorbs the whole lie. `sheaf_consistency_radius.py`'s closed-form section (`[0] CLOSED FORM vs the R6 mechanism numbers`) prints both against its own numerical solve:

```
C6 relayed cycle edge, lie 3.0:  r = 1.224745   |s|*sqrt(1-R_eff) =
    3*sqrt(1-0.8333) = 1.224745
P6 relayed cut edge,   lie 3.0:  r = ...        R_eff = 1.000000 ->
    predicted 0
```

matching the hand value to six decimals on C_6, and on P_6 holding the printed residual under 10^{-9} (the script's own `TOL_SILENT` zero-residual threshold) rather than printing a literal 0 [internal, `sheaf_consistency_radius.py`]. The general law is $r = |s|/\sqrt{n}$ on any C_n: $|s|/2$ at n=4, $|s|/\sqrt{6} \approx 0.4082|s|$ at n=6, $|s|/\sqrt{8} \approx 0.3536|s|$ at n=8 [verified, closed form] — conviction dilutes monotonically as the ring lengthens, and any path reads as silence regardless of n. *Now you try:* reproduce both printed lines and the general law against the script (Exercise 8.3), then reason about why the theorem needs a *single* equivocator (Exercise 8.4).

Exercises 8.3–8.4, page 35.

The residual as a certificate. Robinson's consistency radius asks how far local data sits from the nearest global section [36]; the residual r is its least-squares form under the completion over severed blocks. The new content is that r is not merely a distance. It is a certificate about the adversary: every offset pattern that could have produced the visible data is at least this large, and one of exactly this size exists. Topology enters through electrical network theory. A lie on edge e hides in proportion to how well the rest of the visible graph can explain it away, which is the effective resistance $R_{\text{eff}}^{K_c}(e)$ measured in the visible coordinate subgraph: on a bridge $R_{\text{eff}} = 1$ and the lie hides completely; on a long cycle $R_{\text{eff}} \to 1$ and the certified fraction $\sqrt{1 - R_{\text{eff}}}$ decays like $1/\sqrt{n}$. Severing a link raises the effective resistance of everything that was routed around through it, which is the visibility clause written as a price.

Theorem 5.3 Soundness and achievability of the completion residual. For the completion-residual detector with prefix restrictions and three-tier visibility: $r > 0$ iff no global section explains the compared-plus-relayed data, and honest executions give $r = 0$ exactly, for every visibility pattern. Every offset pattern ε consistent with the visible data satisfies $\|\varepsilon_K\| \geq r$, with equality at the completion: r is the exact minimum lie. For a single equivocator q with offset vector o and visible-lie operator A_q, $\sigma_{\min}^{+}(\Pi_K A_q)\,\|o_\perp\| \leq r \leq \|o_\perp\|$, where $o_\perp$ is the component of o outside $\ker A_q$. For a single-edge lie of size s in coordinate c,

$$r = |s|\sqrt{1 - R_{\text{eff}}^{K_c}(e)}, \qquad \text{on } C_n:\ |s|/\sqrt{n},$$

with the effective resistance taken in the visible subgraph K_c, not in G_c.

Proof sketch. r is the distance from the visible cochain g_K to the image of the coboundary restricted to visible edges, the severed blocks being free. An offset pattern is consistent with the data iff $g_K - \varepsilon_K$ lies in that image, so $\|\varepsilon_K\| \geq r$ by definition of the distance, and the completion residual is itself such a pattern, which gives equality. Honest data is a coboundary, so $r = 0$. For one lie of size s on e, g_K is s times the edge indicator, and the squared norm of its projection off the coboundary image is s^2 times one minus the edge's effective resistance, the standard identity between the cycle-space projection of an edge and $1 - R_{\text{eff}}(e)$; on C_n every edge has $R_{\text{eff}} = (n-1)/n$. □

Theorem 5.4 Localization. Noise-free, single equivocator: the support of the edge residual lies on cycles of the visible coordinate subgraphs K_c through q. With bounded noise η, localization is guaranteed while $2\|\eta_K\| < \max_f \|(\Pi_K \varepsilon_K)_f\|$.

Proof sketch. The residual is the projection of the lie onto the cycle space of K_c, and the projection of a single edge's flow is the electrical current it induces, which is supported on the cycles through that edge. The noise condition says the largest residual entry exceeds twice the noise the projection can add, so the ordering of edges by residual is preserved. □

Theorem 5.5 Cost. Because the visible coboundary splits per shared coordinate into graph incidence operators, r is one sparse least-squares problem that decomposes into at most L scalar graph-Laplacian systems, solvable in $\widetilde{O}(|E| \cdot L \log(1/\epsilon))$ with symmetric diagonally dominant solvers [39]. Bach's #P-hardness of sheaf cohomology [40] concerns coherent sheaves on projective space, the wrong category; finite cellular sheaves over $\mathbb{R}$ on graphs are linear algebra.

Checked. Soundness 0/600 violations; achievability 600/600 (a consistent pattern of norm exactly r constructed every time). The effective-resistance closed form is exact on C_4 to C_{12} and on 200 random graphs; the lower bound is tight under the adversarial offset ($r/\|o_\perp\| = \sigma^+_{\min} = 0.4082 = 1/\sqrt{6}$); a uniform kernel lie of norm 5 gives $r = 2 \times 10^{-14}$, a consistent alternative state rather than a split view, invisible in principle and correctly not flagged, and on a cycle of K_c the uniform offsets are the only such directions, every kernel direction lying in the uniform span to 5×10^{-16}. Localization 200/200 on the two-cycles topology and 128/128 on severed expanders; the noise boundary: at least 95% localization up to a noise-to-lie ratio of 0.1, 76.5% at 0.2. Conjugate gradient's timing slope against $|E|$ stays well under the 1.6 gate while dense solve is consistently super-linear; the decomposition, not the two digits, is what the gate certifies. Two coordinated liars on a common cycle, each alone certified at $r \geq |s|/\sqrt{8}$, jointly cancel to $r = 6 \times 10^{-15}$ [internal, `sheaf_consistency_radius.py`].

What it buys. The harness's statistic is upgraded to a certificate a federation operator can act on: r is the smallest lie consistent with what was seen, the hot edges lie on cycles through the culprit's neighborhood, and the whole computation is a handful of Laplacian solves, cheap enough to run on every gossip round. The closed form also converts topology into policy. Effective resistance is the price list for where a federation should force direct reconciliation (bridges: mandatory, since the residual cannot help there) and where relayed gossip suffices (short cycles: cheap certified detection).

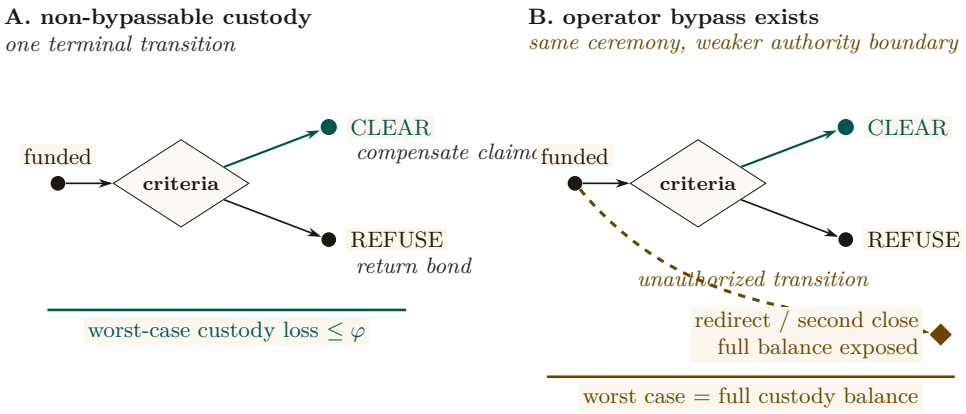
> **Where this stops.** Single equivocator is the scope, and provably so: a coalition whose combined lie is a coboundary is a consistent counterfactual world, and two coordinated liars

> on a common cycle cancel the residual to numerical zero. r never overstates the lie, but it can understate a coalition's to zero; soundness is unconditional, detection is not. No attribution, ever: a $+o$ lie by q toward w and a $-o$ lie by w toward q produce identical observations, so the residual localizes to cycles through the equivocator and cannot say which endpoint lied; naming culprits is the signatures-and-forensics business [38], which this detector complements. Vanishing r is not an all-clear, for three independent reasons: uniform lies move every neighbor's view identically and are invisible in principle; severed edges admit consistent counterfactuals by construction; and over $\mathbb{R}$ the obstruction is abelianized, so a vanishing real-coefficient signal does not certify gluability in richer coefficient systems [37]. Where no report crosses, nothing can be seen, and severing one link of a loop darkens the whole loop; under measurement noise the certificate needs a threshold above the honest floor, below which it reverts to "consistent with noise."

Sheaf Laplacian dynamics — and what is *not* claimed here. On a cellular sheaf with real stalks, the **Hansen–Ghrist sheaf Laplacian** [35] is $\Delta_{\mathcal{F}} = \delta^*\delta$, and its harmonic space $\ker(\Delta_{\mathcal{F}}) \cong H^0(G;\mathcal{F})$ is precisely the subspace of globally synchronized ledger states. This object exists only over the vector-space sheaf constructed above; it is undefined over the semilattice of append-only logs, which has no adjoint and no spectrum. The spectral gap $\lambda_2(\Delta_{\mathcal{F}})$ — the smallest non-zero eigenvalue — is the natural quantity to *relate* to anti-entropy convergence speed, a small gap meaning slow mixing by analogy with the graph-Laplacian case. **We have not derived or checked such a bound, and this paper claims none.** The Hansen–Ghrist rate governs continuous-time linear diffusion $\dot{x} = -\Delta_{\mathcal{F}}x$ over real stalks; anti-entropy gossip over signed append-only logs with a Byzantine equivocator is discrete, asynchronous, adversarial, and monotone-join rather than linear-averaging, and nothing here shows the bound transfers. The two convergence stories — classical epidemic dissemination (Theorem 5.1, which is a real and separately argued result [15]) and sheaf-Laplacian diffusion — are not reconciled, and neither is a second proof of the other. No relaxation-time bound for the federation is established, in this section or anywhere in this paper; that is consistent with Appendix A, whose dissemination rows record "no partition deadline" and "no hard delivery bound." The reconciliation is open work (Appendix A, *open* band).

6 Cross-Harbor Settlement

The third governing protocol is cross-harbor settlement. A bond is posted at A to cover damage caused by an A-side agent operating at B. When damage occurs, the bond must clear to the damaged party at B. The protocol does this without making either harbor sovereign over the other, but it still uses a third-party custody service. Its transition surface must be non-bypassably restricted; the property below states that assumption rather than relabeling the service as trustless.



Custody Assumption: whitelist + fee cap + one terminal transition are the theorem's hypothesis, not decoration

Figure 6: The custody bound is the absence of an extra transition. Panel A permits exactly one terminal move from funded custody to **Clear** or **Refuse**; with a recipient whitelist and fee cap, worst-case loss is bounded by the agreed fee φ. Panel B adds one operator-controlled bypass. The visible extra edge reaches redirect or a second close, exposing the full balance. Threshold signing is only one candidate implementation; the loss theorem holds only when the three authority restrictions are actually non-bypassable.

6.1 The bounded escrow

We introduce a third party—the escrow—and require the *custody ledger* to expose exactly two terminal transitions: clear (pay whitelisted beneficiary B, refund the remainder to A) or refuse (return the bond to A). A signed policy alone is only evidence. The following restrictions are structural only if the escrow service cannot bypass the ledger API or alter its code and keys:

- Redirect funds to anyone other than A or B.
- Equivocate about its clearing decision (publish "cleared" to one party and "refused" to another).
- Delay past the TTL of the bond.
- Extract more than a pre-agreed fee ϕ which is capped at issuance.

These restrictions define the desired transition system. The current artifact is a design and bounded model, not a deployed custody mechanism.

Design invariant 6.1 Conditional Escrow Extraction Bound. Assume the custody ledger (i) admits only recipients A and B, (ii) caps the fee output at ϕ, (iii) executes exactly one terminal transition atomically, and (iv) keeps signing and transfer authority outside the escrow operator's bypassable control. Then no allowed transition pays the escrow more than ϕ.

The bound follows by case analysis over the two allowed terminal transitions. Witness logging makes delay or equivocation detectable, and an escrow bond can compensate some misconduct, but neither fact prevents theft if the operator can bypass custody restrictions. Without assumptions (i)–(iv), the worst-case extractable value is the full custodial balance.

The escrow is a trusted third party with a *conditional transition bound*. This is not a trustless construction in the HTLC sense [16]. Whether useful trustless settlement is possible for non-fungible reputation-priced bonds remains open (§12); this paper gives neither a construction nor an impossibility proof.

A candidate instantiation of assumption (iv). One concrete way to remove the escrow operator's *unilateral* control over transfer authority is a 2-of-3 threshold-signature scheme: harbor

A, harbor B, and a federation arbiter each hold one key, and no transfer executes without two of the three signatures. When A and B agree on the outcome — the ordinary case — they co-sign the release themselves and the arbiter never touches custody. The arbiter's key exists only to break a dispute: it evaluates the witness-logged evidence trail and co-signs with whichever party it rules for. This is a plausible way to satisfy assumption (iv), and we name it because a reader will otherwise ask what such a mechanism would look like. It does not close what Design Invariant 6.1 calls conditional. The arbiter remains a trusted third opinion; we have not analyzed an arbiter colluding with one principal, we have not specified how arbiter selection resists capture, and the underlying settlement rail (an on-chain threshold contract, a custodial payment hold, or anything else) carries its own assumptions that this paper does not model. We list 2-of-3 multisig as a candidate design, not a closed result; trustless settlement for non-fungible bonds stays in the *open* band (§12.2, Appendix A).

6.2 Cross-harbor conservation

The Bonded Commons paper proved a single-machine Conservation Theorem: the total bonded value in the system never changes by surprise; bonds are only created (by posting), destroyed (by clearing), or rolled (refunded and re-posted). We extend this to the federation.

Design invariant 6.2 Cross-Harbor Bucket Partition. For each funded bond b of amount a_b, let $P_b, E_b, C_b, R_b \in \{0, a_b\}$ denote its posted, in-escrow, cleared, and refunded buckets. Valid transitions preserve

$$P_b + E_b + C_b + R_b = a_b \quad \text{and} \quad \big|\{Z \in \{P_b, E_b, C_b, R_b\} : Z = a_b\}\big| = 1.$$

Summing over bonds gives $\sum_b (P_b + E_b + C_b + R_b) = \sum_b a_b$. External top-ups create new funded bonds and are recorded separately; fees must appear as an explicit cleared recipient rather than disappear from the identity.

This is the federation-layer analogue of Conservation in Bonded. The intra-harbor Conservation invariants hold per-harbor (Bonded gives the proofs); the cross-harbor property adds the constraint that bonds in transit between harbors — i.e., posted at A, locked in escrow, awaiting clearance against B — are accounted for as in-escrow rather than as either posted at A or cleared to B. The escrow's role as the holder of in-flight bonds is what makes the conservation accounting tractable.

Numbers by hand — One bond across two harbors. A \$100 bond posted at A has buckets $(P, E, C, R) = (100, 0, 0, 0)$. The transfer locks it in escrow, $(0, 100, 0, 0)$. Clearance at B pays \$90 to the counterparty and \$10 in fees, recorded as two cleared recipients, so the row is $(0, 0, 100, 0)$ with $C = 90 + 10$; a refund instead gives $(0, 0, 0, 100)$. In every row exactly one bucket holds the whole amount and the four sum to 100 [internal]; a fee that vanished from the row would be the first sign of a broken partition.

6.3 Settlement protocol

Figure 6 shows the three-step protocol. To unpack the figure:

1. **Bond post.** Alice's harbor mints a work card for agent α; agent operates at B. Bond $B_\alpha = \$B$ is posted at A and locked in the escrow. The bond's amount and TTL are signed into both A's and B's witness-log roots so the federation knows the bond exists.

2. **Damage claim.** Damage detected at B (acceptance criteria fail, or invariant violation). Bob's harbor signs claim $\mathsf{cl}_B = (R_B^{(e)}, \text{evidence})$ where the evidence is a Merkle inclusion proof against $R_B^{(e)}$. The claim is sent to the escrow.
3. **Verification and clearance.** Escrow verifies inclusion of α's actions in $R_B^{(e)}$, matches against the acceptance preconditions that were signed at step 1, and produces one of two outcomes: *clear* (pay B out of bond, refund remainder to A) or *refuse* (return bond to A, log reason). Both outcomes are recorded in the witness log.

The designed transition is atomic in the bucket-partition sense: a bond moves once from in-escrow to either cleared or refunded. This property depends on the custody-ledger assumptions of Design Invariant 6.1; witness records alone cannot make two external transfers atomic.

6.4 Fee structure and economic discipline

The escrow's fee ϕ is bounded but non-zero. In the competitive-insurance market sense of *The Bonded Commons* (the Youle contribution), ϕ is a market clearing price for the service of holding the bond and verifying the inclusion proof. We do not solve the market here; the Youle contribution from Bonded gives the mechanism, and we conjecture (open) that it extends cleanly to the cross-harbor case when the escrow population is large enough.

We are explicit that for small federations (two or three harbors), ϕ is likely set by a posted price rather than auctioned. The mechanism design here is identical in shape to Bonded's, just lifted to the federation layer.

Recall.

1. Without looking: name the two terminal transitions the custody ledger must expose, and the four assumptions Design Invariant 6.1 conditions its bound on.
2. In the three-step settlement protocol, which step is the one a signed inclusion proof alone cannot settle by itself, and why?
3. What does the 2-of-3 threshold-signature candidate remove from the escrow operator's control, and what does it *not* close?

7 Federation Admission

The Federated Harbor is not a permissionless market in identities. A new harbor enters the federation through an admission ceremony. Agents inside an admitted harbor are named by certificates the harbor issues, each binding a principal identifier, an agent key, the issuer, an epoch, a policy, and an expiry; the federation never sees a bare key, and a certificate outlives neither its epoch nor its policy. The ceremony is invitation-bounded by design: Sybil attacks against the federation [17] are costly for an attacker precisely because each identity requires a real admission event.

7.1 The bonded-sponsor pattern

A new harbor N with no prior reputation cannot price its own bond fairly under competitive insurance — the market lacks data on N's risk. The bonded-sponsor pattern resolves this:

1. An existing harbor S (the sponsor) posts a bond on N's behalf, sized to cover the federation's worst-case loss from N misbehaving during a probationary epoch.
2. N joins the federation under probation; its cards are accepted, its evidence trail is witness-logged, but its sponsor bond is at risk.
3. If N misbehaves during probation — forges signatures, equivocates on roots, refuses to honor settled bonds — the sponsor bond is forfeit. The sponsor has full incentive to vet N before sponsoring.

4. At the end of probation, N's own reputation is sufficient to price its own bond, and the sponsor bond is released.

This is the federation-layer analogue of competitive insurance from *The Bonded Commons* (the Youle contribution): instead of an authority picking who joins, the sponsor market discovers it. The cost of being a sponsor is real — forfeit risk during probation — and the benefit is the network effect of having N in the federation. We expect (open) that for federations with many candidate sponsors and clear-eyed risk pricing, the equilibrium is reasonably efficient. We do not prove it.

7.2 Cold-start failure modes

A new harbor that cannot find a sponsor cannot join. This is a real limitation. In the worst case, the federation collapses to invitation-only — which is the failure mode of every interesting federated identity system from PGP web-of-trust [22] to SAML [18]. The structural pushback is:

- *Many sponsors.* The market is many-to-many; a new harbor needs only one sponsor. As long as sponsors are paid (via fees or reputation), the supply is nonzero.
- *Small probationary scope.* New harbors during probation are restricted from large-stake operations; the sponsor's risk is bounded.
- *Witness-log permanence.* The admission record is permanent; once a harbor has graduated probation, it does not need a sponsor again.

This is engineering discipline, not a theorem. The historical evidence on whether invitation-bounded federations stay open is mixed [18]; the Federated Harbor's pushback against centralization is to make the cost of being a centralizing hub bounded above by the cost of running mirrored witness logs, which is small. Whether that pushback is enough is empirical.

8 Worked Example

The four primitives are easier to read against a concrete case. We work the example end-to-end: two organizations, three machines, one shared project. The example is intentionally familiar; nothing in it is novel except the way the primitives compose.

8.1 Setup

Two organizations: Acme Robotics (Alice's employer) and Beta Vision (Bob's employer). The shared project is a perception-and-control stack: Acme owns the controller, Beta owns the vision system, both run against a shared staging environment that lives on a third machine maintained by neither.

Three machines: Alice's laptop (harbor A, controller code, controller test database), Bob's laptop (harbor B, vision code, vision test database), staging-server (harbor S, shared staging database, no agents resident).

Each harbor has gone through the admission ceremony of §7. Acme's harbor was sponsored by an existing partner harbor at the open-source consortium that hosts the staging server; Beta's was sponsored by Acme after a six-month probationary period during which Acme's risk officer watched Beta's evidence trail closely. The staging server harbor is sponsored by the consortium directly and is treated as a known-honest verifier for the purposes of this example.

8.2 The agent task

Acme's agent — call it Controller-7 — needs to run a schema migration on the staging database to add a column for a new sensor type. The migration is destructive (drops and recreates the column); if it goes wrong, the staging database is unusable for the joint demo scheduled three hours later.

Controller-7's local card C_A authorizes `db:migrate` against the controller test database, attenuated through Acme's risk policy (att_A = "only against tables in the `controller` schema; only between 9am and 5pm; only if a human operator has approved within the last 30 minutes"). The card is valid at A, but the database it needs to migrate is at S.

8.3 Step 1: Cross-harbor transfer

Controller-7 invokes $\mathsf{xfer\_req}(C_A, S)$ on D_A. D_A constructs the envelope:

$$\mathsf{xfer\_offer}\langle C_A, \text{att}_A, R_A^{(e)}\rangle \text{ signed under } sk_A,$$

and sends it to D_S (the staging-server harbor). D_S verifies the signature against A's public key as recorded in the witness log, verifies that $R_A^{(e)}$ is the current epoch root for A, and applies its own attenuation att_S ("only the `controller` schema; only after a backup snapshot has been taken; only if no other migration is in flight"). The composed attenuation produces $C'_S = \text{att}_S(\text{att}_A(C_A))$, a card valid only at S, signed under sk_S.

D_S delivers C'_S to Controller-7. Controller-7 takes the snapshot, then runs the migration against S. The migration's actions are recorded in S's Merkle forest and roll up into $R_S^{(e)}$.

8.4 Step 2: Damage detection

The migration runs cleanly. Two hours later — well after the agent has moved on — Beta's vision system runs an integration test against the staging database and observes that the new column has the wrong type (Acme assumed `float32`; Beta needs `float64` for the sensor data they will produce). The integration test fails.

D_B (Beta's harbor) verifies the failure against its local invariants and produces a damage claim:

$$\mathsf{cl}_B = (R_B^{(e+1)}, \text{evidence}),$$

where the evidence is a Merkle proof that Beta's integration tests against the staging database failed because of a schema mismatch. The claim is sent to the escrow.

8.5 Step 3: Settlement

Acme posted a bond when Controller-7 was authorized to run the migration. The bond is held in the federation escrow with a TTL of 24 hours — long enough for downstream consequences to surface, short enough not to lock collateral indefinitely.

The escrow verifies cl_B: it checks the inclusion proof against $R_B^{(e+1)}$ in the witness log, matches the claim against the acceptance preconditions signed at bond-post time (which named the staging schema as a covered surface), and decides: *clear*. Acme's bond is debited; Beta receives the cleanup-cost portion; the remainder is refunded to Acme. The settlement record is written to the witness log.

8.6 Step 4: Revocation propagation

At this point, Acme's risk officer revokes Controller-7's `db:migrate` card and publishes $R_A^{(e+2)}$. Under Theorem 5.1's connected reliable-round model, dissemination completes in expected $\Theta(\Delta \log m)$ time; for a federation of eight harbors that is about three rounds ($\log_2 8 = 3$), since each round roughly doubles the informed set [internal]. During a partition, receiving harbors must rely on expiry or a fail-closed freshness policy; the example does not pretend gossip supplies a deadline.

8.7 What this example shows

The example is small and the failure is real. The point of working it is to show that each of the four primitives — transfer, revocation, settlement, admission — is doing its job at exactly one step, and that the composition is straightforward when each primitive is honest about what it covers.

Notice in particular what *did not* happen:

- Neither D_A nor D_B trusted the other's signing key as a root. The transfer worked through the witness-logged public keys, not through a chain of trust.
- The bond was not held by either harbor; the design requires a recipient-whitelisted custody ledger before the escrow's role is structurally bounded.
- The settlement was not negotiated peer-to-peer; it followed a fixed protocol whose two outcomes were named in advance.
- Under the example's connected reliable-round assumptions, the revocation propagated through gossip with expected logarithmic completion; no consensus protocol was invoked.

Each of these is a deliberate design choice. The Federated Harbor's architecture is not heroically more powerful than alternatives; it is heroically more honest about what is happening at each step.

9 Formal Model

The formal substrate of this paper is two-layer: ProVerif [31] models for the cryptographic protocols (cross-realm authenticity, attenuation, settlement no-redirect) and TLA+ extensions of Bonded's Conservation specification for the cross-harbor accounting invariants.

9.1 ProVerif extensions to Anchor

The Anchor Protocol's three-phase models are extended with a second signing key and an epoch-root binding. The cross-realm authenticity query becomes:

```
query b: id, ha: id, hb: id, cap: capability, e: epoch;
  event(AcceptedAtB(b, hb, cap, e))
  ==> (event(IssuedAtA(b, ha, cap)) &&
       event(EnvelopeAtoB(ha, hb, cap, e)) &&
       event(RootCurrent(ha, e))).
```

Listing 1: Cross-realm authenticity query (sketch)

This says: if B accepts a cross-realm card for capability `cap` at epoch e, then there exists a corresponding issuance event at A, an envelope event from A to B binding the same capability and epoch, and an event confirming that $R_A^{(e)}$ is the current root. The query holds against the Dolev–Yao attacker model under the standard assumption of Ed25519 EUF-CMA security.

The model is drafted and the proof script is in progress. The full mechanization is named open work; see Appendix A for the status table.

Exercise 8.5, page 35.

9.2 TLA+ extension for cross-harbor conservation

The Bonded Commons paper's Conservation specification models a single-machine bond pool. The Federated Harbor extension adds a per-harbor pool, an escrow pool, and an inter-harbor transit relation:

```
CONSTANTS Harbors, MaxBonds, Capabilities
VARIABLES Bucket, WitnessLog

```

```
TypeOK ==
  /\ Bucket \in [1..MaxBonds -> {"posted", "escrow", "cleared", "
     refunded"}]
  /\ WitnessLog \in Seq([harbor: Harbors, epoch: Nat, root: STRING])

BucketPartition ==
  \A b \in 1..MaxBonds:
    \E! s \in {"posted", "escrow", "cleared", "refunded"}:
      Bucket[b] = s

Spec == Init /\ [][Next]_<<Bucket, WitnessLog>>
```

Listing 2: Cross-Harbor Conservation specification (sketch)

The full specification (Appendix C) associates each bond with an amount and enforces amount-preserving moves through these buckets. Apalache checks the resulting transition system at $|F| = 4$ and bond depth 16. This establishes the invariant only for the model and bound; it does not establish the custody-ledger assumptions or arbitrary-scale runtime conformance.

9.3 What is mechanized and what is structural

We are explicit about which claims live in which band (Figure 1).

Mechanized (model scope). Single-harbor capability authenticity is inherited from Anchor. Cross-realm authenticity and attenuation are draft models. The cross-harbor bucket-partition invariant is model-checked at $|F| \leq 4$.

Conditional (assumption scope). Revocation dissemination is expected $\Theta(\Delta \log m)$ only under the stated reliable connected-overlay model (Theorem 5.1). The escrow extraction bound is conditional on a non-bypassable recipient-whitelisted custody ledger (Design Invariant 6.1).

Open (cobalt). Multi-principal correlated-equilibrium extension of Bonded; trustless settlement for non-fungible reputation-priced bonds; folk-theorem cartel-resistance bound at the federation layer.

10 Failure Modes

A federation paper that does not engage formally with the historical failure modes of federated systems is unserious. The literature is heavy with examples: SAML federations that concentrated on a small set of identity providers [18]; PGP web-of-trust that never achieved meaningful adoption outside enthusiast circles [23]; Sigstore's gravitation toward a single deployment of Fulcio [11]. We engage with three failure modes specifically.

10.1 Centralization gravity

Federated identity systems have a well-documented tendency to centralize on a small number of trust anchors within five to ten years of deployment [18]. The Federated Harbor's structural pushback is the replicable witness log: any harbor can run its own witness instance, any harbor can audit any other instance's roots, and equivocation between witnesses is detectable. This makes it more expensive to be a centralizing hub than to be a peer.

The pushback is not a theorem; it is design discipline, and Figure 1 places it in the cobalt band for that reason. The historical evidence is mixed. Tailscale [29] has remained operationally centralized on its own coordination server despite Headscale [30] being a viable alternative;

in-toto [12] has stayed decentralized but is consumed mostly by SLSA [13] which has Google as its de facto center. Time will tell which pattern the Federated Harbor exhibits.

10.2 Equivocation by a malicious witness

A malicious witness log can serve different views to partitioned auditors. A contradictory signed pair is immediately verifiable once co-located, but the protocol has no topology-independent time bound for bringing the pair together. Witness bonding applies after confirmation.

For high-stakes settlements, harbors can require multiple witness signatures before treating a root as authoritative, in the Certificate Transparency two-SCT pattern [10]. This trades latency for assurance; the design space is well-explored in the CT literature and we adopt the same techniques.

10.3 Partition tolerance

The federation does not run consensus; partition tolerance is per-protocol. Capability transfer cannot proceed during a partition between A and B (no surprise; cross-realm authentication requires both daemons to reach the witness log). Revocation gossip continues within each partition; convergence is bounded by the partition's local size, not the global federation size. Settlement is parked at the escrow until the partition heals.

The structural posture is that partitions are operationally common and the federation should degrade gracefully rather than fail fast. We inherit this discipline from Anchor's intra-mesh design [1]; the federation extension is straightforward but adds no novel mathematics.

11 Related Work

The Federated Harbor sits at the intersection of four bodies of work. We are concrete about where we draw on each and where we depart.

Federated identity. The OpenID Federation 1.0 specification [25] is the closest existing analogue: each entity publishes a signed Entity Statement, trust chains assemble by walking up to a Trust Anchor, trust marks express property claims. The Federated Harbor is structurally similar in the trust-anchor pattern but differs in two places: (a) we add a capability/attenuation layer (OpenID Federation is identity-only), and (b) we do not assume HTTP-based fetching of entity statements, since Port Daddy daemons cannot reliably expose HTTP endpoints (NAT, sleep, etc.). We draw also on SAML/Shibboleth historical experience [18] as cautionary lineage; the centralization gravity we discuss in §10 is exactly the SAML-era failure mode.

Capability-based federation. Macaroons [5] are the foundational capability-with-attenuation construction; our transfer ceremony extends Macaroon-style third-party-caveat chains across an administrative boundary. UCAN [24] is the current production-deployed answer to "Macaroons with public-key principals," and we treat it as the operationally most relevant existing system; the Federated Harbor differs in keeping the daemon as a structural commons authority rather than going fully peer-to-peer, and in pricing delegation through insurance-priced bonds rather than pure capability constraints. The object-capability lineage [7, 6] continues to be relevant: cross-realm operations are exactly a mutual-suspicion scenario in Miller's vocabulary, and his defensive-consistency framework is the right lens.

Production federation protocols. Three deployed federations solve a version of the same trust-boundary problem and are not otherwise engaged here. ActivityPub [19], the W3C-standardized protocol behind Mastodon and the fediverse, lets independent servers relay signed

activity without a shared database or a global ruler — exactly the property this chapter states as its own goal. Matrix [20], an open, currently-deployed specification, has homeservers replicate room state with eventual consistency across administrative boundaries and no single point of control, arguably the closest production analogue running today to the "no consensus, gossip-bounded, detectable equivocation" model of §2.4. And SPKI/SDSI [21], the mid-1990s local-names, decentralized-authorization design, is the conceptual ancestor both Macaroons and UCAN descend from, and so sits one layer beneath the capability-based federation discussed next. None of the three overturns a theorem in this chapter; each is a closer production point of comparison for the federation's design than the identity- and capability-federation literature cited above.

Transparency logs. Certificate Transparency [10] is the model; the federated witness log of §2.3 is CT applied to capability tokens. Sigstore [11] and Rekor are the production instantiation we rhyme with most heavily; in-toto [12] and SLSA [13] are the supply-chain analogues that compose with the Federated Harbor for the "poisoned plugin under a bonded principal" case but do not close it. The Crosby–Wallach history-tree [14] is the formal substrate of every modern transparency log including ours.

Cross-domain settlement. Herlihy's atomic cross-chain swap [16] is a comparison point; the cross-harbor settlement protocol of §6 instead handles non-fungible reputation-priced bonds through conditionally restricted custody. It is not trustless in Herlihy's sense. Whether trustless settlement is achievable in our setting is open (§12). The IBC protocol [26] supplies another connection/channel comparison; unlike IBC, this design does not assume a consensus-backed state machine underneath.

Agentic-commerce discovery and mandates. The Universal Commerce Protocol [32] (Google/Shopify, 2026) solved cross-administrative discovery for agent-mediated retail with a pattern the Federated Harbor adopts (ADR-0051 Phase 1b): a `/.well-known` JSON profile that does double duty as capability declaration and JWK key publication, with server-selects version negotiation — the party that must execute a request computes the compatibility intersection. Where UCP assumes merchants hold stable HTTPS origins, our daemons often do not (the same constraint that ruled out OpenID Federation's entity-statement fetching above); the profile therefore also travels as a relay-published mirror. UCP delegates authorization proof to the Agent Payments Protocol's verifiable-credential mandates [33], whose SD-JWT/JWS/JCS formats the harbor boundary now speaks (ADR-0094). The division of labor is instructive: those protocols standardize discovery and the transaction envelope and are explicit that counterparty trustworthiness, collateral, and settlement enforcement are out of scope — which is the half of the problem this paper and its predecessors supply.

12 Limitations and Open Questions

We close with the honest open frontier. These are not throwaway caveats; they are the questions the paper does not answer, named precisely enough to motivate subsequent work. Figure 1 is the map they sit on: each of the questions below belongs to one of its three bands, and the band says what kind of answer would close it.

12.1 The multi-principal correlated-equilibrium extension

The Bonded Commons paper argued that single-machine coordination is a correlated equilibrium with the daemon as the correlation device [2]. Across two harbors with two daemons, the correlation device fragments. Two candidate resolutions remain unresolved:

- The federated witness log *is* the correlation device, and individual harbors are local enactors. This would extend Bonded's result intact.
- The cross-harbor case is a partition of the correlated-equilibrium concept into intra-harbor (correlated) and inter-harbor (Nash with shared signal). This would weaken Bonded's result at the federation layer.

We do not know which is right. The proposal that scoped this paper [3] named Thomas Youle as the essential collaborator for this question (his contribution to Bonded is the closest existing work). The question stays open here.

12.2 Trustless settlement for non-fungible bonds

HTLC-style atomic swaps [16] achieve trustless settlement for fungible assets with a shared hash preimage. The Federated Harbor's bonds are non-fungible and adjudication-dependent, so that construction does not transfer directly. We adopt conditionally restricted custody (§6) and leave both a trustless construction and an impossibility result open. Either would require a separate threat model and proof.

12.3 Cartel formation across federations

Bonded's folk-theorem cartel-resistance argument applies inside a single machine. Between machines, cartel formation is structurally easier: colluders can coordinate offline and present a unified front to the witness log, which records that all harbors agreed — a behavior consistent with both honest agreement and collusion. We owe either a strengthening of the bonded mechanism that resists cross-harbor cartels, or a precise statement of the weakening. We have neither.

12.4 Cold-start admission and the invitation-only failure mode

The bonded-sponsor pattern of §7 requires a new harbor to find a sponsor willing to underwrite its probationary risk. In practice, this market may be thin — the historical evidence on whether invitation-bounded federations stay open is mixed [18]. We treat the structural pushback (many sponsors, small probationary scope, witness-log permanence) as engineering discipline, not as theorem.

12.5 Equivocation propagation speed

Theorem 5.1 separates cryptographic verification from dissemination. The former is constant work once both roots are present; the latter depends on topology, loss, and partitions. Establishing a deployment-specific tail bound is open and is necessary before pricing the witness bond from propagation risk. A second gap sits next to it: §5.4 names the sheaf Laplacian's spectral gap $\lambda_2(\Delta_{\mathcal{F}})$ as the natural handle on anti-entropy mixing speed but derives no bound from it, and does not reconcile linear sheaf diffusion with epidemic gossip over append-only logs. *open*.

12.6 Cost of per-write durability

The cryptographic assertions and per-write durability checkpoints this layer inherits from Anchor carry a measurable I/O cost. They are designed for high-stakes coordination, where an audit trail is worth more than throughput, and not for bulk data movement or latency-critical paths. This paper does not measure that overhead at the federation layer and does not establish where the crossover sits. *open*.

12.7 What this means for the reader

A federation paper with six named open questions is doing its job. The point of the paper is to draw the new boundary cleanly so the open problems are visible; if every claim in this paper were

closed, there would be no third paper to write. The honest read is that the Federated Harbor is closer to a research agenda than to a deployed product, and the prior two papers (Anchor, Bonded) have a tighter ratio of theorem to conjecture for the same reason. The substrate is real; the federation layer is in progress.

> *A boundary drawn cleanly is itself a result: it says where the next proof has to start.*

Exercises 8.6–8.11, page 35.

Review of the key ideas

What this chapter leaves coupled, in the order the rest of the book will lean on it:

1. **Two machines, no shared daemon.** A federated authority admits a foreign act only through a witness log each side can check; admissibility is a design invariant of the log, not a property of the peer.
2. **Capability transfer shrinks at every hop**: the set printed at Harbor A contains the set acknowledged at Harbor B, and B verifies locally at delivery without reaching back to A. Attenuation monotonicity across the boundary is the theorem.
3. **Revocation crosses the boundary as an epoch**: dissemination is conditional on connectivity, the stale window is bounded in a connected federation, and a partition has no finite deadline.
4. **Inconsistency is detected exactly when the missing edge closes a visible cycle.** On six nodes the cycle C_6 has consistency radius 1.2247 and the path P_6 has 0: the same disagreement, visible on one and invisible on the other.
5. **The completion residual is sound and achievable**, it localizes the fault, and its cost is stated; two liars on one cycle can cancel, and that is the boundary.
6. **Settlement across harbors is bucketed.** Escrow extraction is bounded conditional on the custody ledger, and cross-harbor buckets partition, so one harbor's failure is confined to its bucket.
7. **Admission is a ceremony with a record**: a harbor joins a federation by exchanging epoch roots, and the record is what a later dispute reads.
8. **The rollback configuration fails by design.** The card-revocation model checked against a rollback of the epoch root produces a counterexample, which is the property, not a bug in the checker.

13 Exercises

§4 Cross-Harbor Capability Transfer

8.1 Trace ⋆⋆ Alice's card C_A authorizes `db:write` on A's *test* database. D_B accepts it and mints $C'_B = \mathrm{att}_B(\mathrm{att}_A(C_A))$ authorizing `db:write` on B's *staging* database. Trace Theorem 4.1's proof sketch against this concrete case: at which step, if any, does "subset composition is a subset" stop being a statement about the *same* resource space? State precisely what kind of act minting C'_B actually is here, and what two things a correct D_B must additionally establish before minting it. *(solution on page 36)*

§5 Federated Revocation

8.2 Trace ⋆⋆ The "only-issuer-revokes" rule above is trustworthy only if a harbor's own local revocation record is *final* — once revoked, always revoked, even across an operational mishap at the relay. `proofs/relay/CardRevocation.tla` model-checks exactly this property (`RevocationFinal == badAccept = FALSE`) under three configurations. Read `proofs/relay/CardRevocation.md` and state, for each of the baseline, rollback, and epoch configurations, what changes in the model and what TLC found. *(solution on page 37)*

8.3 Trace ⋆⋆ Read `skills/harbor-results/scripts/sheaf_consistency_radius.py`, section `[0]` (function `section0`). By hand, using $r = |s|\sqrt{1 - R_{\text{eff}}(e)}$ and $R_{\text{eff}} = (n-1)/n$ on C_n, confirm the two numbers the script's closed-form check pins for C_6 and P_6 at lie magnitude 3.0. Then state what the script's `check(...)` call actually asserts for each case, and to what tolerance. *(solution on page 37)*

8.4 Open ⋆⋆⋆ Theorem 5.2 assumes a *single* equivocator and says outright that this is not decoration. Two colluding reporters sit on the same visible cycle K_c and each reports an offset; either offset alone would be detected. Using only the definition of the completion residual r as a minimum over global sections and free severed blocks, explain why a coalition can drive r to (numerically) zero even though neither liar's offset is small, and say what kind of evidence — if any — could still catch the coalition. *(solution on page 38)*

§9 Formal Model

8.5 Trace ⋆⋆ `proofs/bonded/federated/federated.pv` is a separate, already-mechanized ProVerif model from the cross-realm sketch above — it is Bonded Commons §7's federated multi-principal secrecy theorem, reused here because it is the formal backbone for reasoning about trust split across administrative principals. Read the model's header comment and state: which four principals it names, which single query it actually runs, and why running only the "daemon + email + KMS compromised" scenario is enough to cover the weaker three-principal scenarios too. *(solution on page 38)*

§12 Limitations and Open Questions

8.6 Open ⋆⋆⋆ §12.1 names two candidate resolutions for the correlated-equilibrium concept once two daemons, not one, are in play. State a stochastic/Bayesian game model of the two-harbor case in which the witness log could serve as a correlation device, name the informational event (a common-view or freshness event) under which you would try to prove a coarse correlated equilibrium, and give a safe degraded policy for the case where that event has not occurred. *(solution on page 39)*

8.7 Open ⋆⋆⋆ §12.2 leaves open whether trustless settlement is possible for the Federated Harbor's non-fungible, adjudication-dependent bonds. HTLCs solve the fungible case using a shared hash preimage. Explain, from the definition of an HTLC's atomicity condition, why that construction does not transfer to a bond whose "did the damage occur" predicate is subjective, and sketch what an impossibility argument for the general case would need to assume. *(solution on page 39)*

8.8 Open ⋆⋆⋆ §12.3 observes that the witness log cannot distinguish unanimous honest agreement from unanimous collusion. Using the one-shot deterrence inequality $p \cdot d \cdot L > G$ (audit probability, detection probability, loss on detection, collusive gain), state what would have to be true of a federation-layer audit mechanism for that inequality to be satisfiable against a cartel that controls a majority, but not all, of the witnesses in scope — and say plainly why no structural mechanism closes the case where the cartel controls *every* oracle and witness. *(solution on page 40)*

8.9 Open ⋆⋆⋆ §12.4 lists engineering discipline (many sponsors, small probationary scope, witness-log permanence) against the bonded-sponsor pattern's cold-start risk, but explicitly declines to call it a theorem. Describe a concrete measurement — stated as a metric over time, not a proof — that would let an operator tell the difference between "the sponsorship market is thin but open" and "incumbent sponsors have formed an invitation cartel." *(solution on page 40)*

8.10 Open ⋆⋆⋆ §12.5 names two unresolved gaps side by side: a deployment-specific tail bound for equivocation propagation, and the unreconciled relationship between epidemic dissemination (Theorem 5.1) and sheaf-Laplacian diffusion (§5.4's closing paragraph). Explain why these are two *different* open problems rather than one problem stated twice, and name one measurement that would be needed before either could be closed. *(solution on page 40)*

8.11 Open ⋆⋆⋆ Standing back from the individual protocols: this paper repeatedly describes federation as "not a consensus protocol" (§2.4) while also proving a bucket-partition invariant by model-checking a bounded transition system (§6.2). Reconcile these two statements — what, precisely, does not require consensus, and what does the Conservation model-checking result actually presuppose about agreement? Then state, in your own words, what a reviewer should mean by calling the Federated Harbor "a project institution, not merely two machines sharing a repository." *(solution on page 41)*

14 Conclusion

The two-machine problem is the structural sequel to the local-swarm problem of Anchor and the trust-as-infrastructure argument of Bonded. The proposed answer is not a new substrate but a clean extension: keep each daemon sovereign inside its own machine, add a shared witness log that no harbor controls, hang revocation gossip and conditionally restricted custody off the witness log, and gate federation admission on bonded sponsorship rather than permissionless entry. Each primitive does exactly one job. Each primitive is honest about what it does not do.

The proof posture matches the prior papers' discipline. Where we mechanize, we mechanize (with the same ProVerif and TLA+ infrastructure). Where we bound a threat, we name the bound and price the bond against it. Where we leave a question open, we name it. The threat-band diagram of Figure 1 is the most concentrated summary of where each claim lives.

What this paper supplies is a clean federation surface that could compose with the local substrate: a transfer ceremony, witnessed revocation records, a conditional custody design, and bonded admission. The protocol and models make the missing implementation and service-level assumptions explicit; they do not establish that arbitrary harbors can already interoperate safely.

The remaining work—runtime custody enforcement, deployment conformance, a multi-principal equilibrium model, and a cartel-resistance bound—is material. The local substrate is real; the federation described here remains a design with partial mechanization.

Solutions to the exercises

8.1 *(exercise on page 34)* The proof sketch's move — "each attenuation is a subset operation in the Anchor algebra; subset composition is a subset" — is exactly right when att_A and att_B both narrow authority *within one shared capability/resource universe*. It stops being a statement about the same resource space at the second attenuation: att_B does not narrow a set of operations on A's test database, because D_B has no test database to narrow into. It grants a *new* authority, over a resource D_A never named, that D_B alone controls. Calling C'_B a "subset of C_A" silently assumes a mapping between the two harbors' resource namespaces that nothing in the four-message ceremony (§4.1) actually constructs.

What minting C'_B really is, in this case, is **token exchange and new authorization at** B, not attenuation of C_A. A correct D_B must additionally establish: (a) *A-side input eligibility* — a signed semantic/resource mapping saying that C_A's authority over A's test database is the kind of thing that may be exchanged at all, recorded with its dependency on A's issuing epoch root $R_A^{(e)}$ so a later revocation of C_A is traceable to C'_B; and (b) *B-side local authorization* — D_B's own policy minting a genuinely new child grant over its staging database, under D_B's sovereignty (Definition 2.1, item 2), rather than treating att_B as if it were narrowing A's grant.

One more gap rides along with this one and is worth naming here rather than assuming solved: Bob's harbor observing a failed acceptance check, or an inclusion proof against $R_B^{(e)}$, does not by itself establish that Alice's agent *caused* the failure (§6.3's damage-claim step). Settlement evidence must bind the exact work-unit, the base and result trees, the mediated effects, the acceptance policy in force, and the verifier that checked it; semantic causation — "α's write is why the check failed" — remains a claim for the adjudicator named in §6.1's 2-of-3 instantiation to weigh, not something the inclusion proof settles on its own. Theorem 4.1 is not false; it is scoped to the case its proof sketch actually covers, and this exercise is that scope made concrete.

8.2 *(exercise on page 35)* All three configurations share one module; only the `Rollback` and `Epoched` constants change. The committed results (`proofs/relay/CardRevocation.md`, "Results (TLC v1.8.0, Homebrew OpenJDK)"), quoted exactly:

```
baseline:  Model checking completed. No error has been found.   (13
    distinct states)
rollback:  Invariant RevocationFinal is violated.               (
    backup-restore re-accepts a revoked card)
epoch:     Model checking completed. No error has been found.   (76
    distinct states)
```

Baseline (`Rollback=FALSE, Epoched=FALSE`, `CardRevocation_baseline.cfg`): no adversary action is available, so a card in `revoked` stays revoked and `RevocationFinal` holds trivially over 13 reachable states. **Rollback** (`Rollback=TRUE, Epoched=FALSE`, `CardRevocation_rollback.cfg`): `RollbackRestore` can drop a card from the DB's `revoked` set while `everRevoked` (ground truth) stays set; `Accept` then only checks `revoked`, so the restored card is accepted again and `badAccept` fires — the invariant is violated *by design*, as the negative control for the attack. **Epoch** (`Rollback=TRUE, Epoched=TRUE`, `CardRevocation_epoch.cfg`): the mitigation adds an external epoch that `Revoke` bumps and that a restore does *not* roll back; `Accept` now additionally requires `cardEpoch[c] = epoch`, so a restored card's stale epoch disables acceptance even though `revoked` was rolled back, and the invariant holds again over the larger 76-state space (the mitigation adds reachable states without reintroducing the breach).

The rollback configuration's counterexample is not a bug in the model; it *is* the ADR-0018 Attack Vector 2 the model exists to demonstrate. A relay revokes card c (an agent's leaked capability); the relay's database is later restored from a backup taken before the revocation; the restored `revoked` set no longer contains c; the relay's own `Accept` check, consulting only that set, admits a presentation of c it had already promised to refuse. TLC's counterexample trace is exactly `Issue(c)` → `Revoke(c)` → `RollbackRestore(c)` → `Accept(c)`, four transitions, `badAccept` true at the end. The failure is the intended result because a baseline or an untested mitigation would give false comfort: it is precisely the negative control that lets the epoch configuration's clean pass mean something, the same discipline §5.4's negative-control ProVerif line and mutation suite apply. This chapter's own per-epoch roots $R_X^{(e)}$ (§2.3) are the same design move at the federation layer: an external, monotonic anchor that a local rollback cannot touch is what makes "only-issuer-revokes" something a cross-harbor verifier can actually rely on, rather than merely something the issuer asserts.

8.3 *(exercise on page 35)* By hand: C_6 has $R_{\text{eff}} = 5/6$, so $r = 3\sqrt{1 - 5/6} = 3/\sqrt{6} = 1.224745$

[verified]. P_6's relayed edge is a bridge (a path has no route between an edge's endpoints other than the edge itself), so $R_{\text{eff}} = 1$ and $r = 3\sqrt{1-1} = 0$ exactly [verified].

The script's own printed lines (`skills/harbor-results/scripts/sheaf_consistency_radius.py`, `section0`), quoted in the worked example above [internal, `sheaf_consistency_radius.py`]: the C_6 line's two computed quantities (the least-squares solve's r and the closed form's `pred`) are asserted equal to within 10^{-9} of each other and within 10^{-6} of 1.224745 (`check(abs(r - pred) < 1e-9 and abs(r - 1.224745) < 1e-6, ...)`) — the numerical solve and the hand formula are checked against each other, not merely printed side by side. The P_6 line's assertion is different in kind: `check(rP < TOL_SILENT and abs(ReffP - 1.0) < 1e-9, ...)` with `TOL_SILENT`$= 10^{-9}$, i.e. it asserts the residual is *below a silence threshold*, not equal to a specific float — the honest way to assert "zero" about a least-squares solve that carries ordinary floating-point roundoff rather than landing on the exact bit pattern for 0.0.

8.4 *(exercise on page 35)* r is defined as the *best-fit* residual: the smallest disagreement left over after choosing the global section x and the free severed blocks g_{sev} that explain as much of the observed data as any choice can. It is not a per-liar quantity; it is a property of the whole observed cochain g_K. If two equivocators on a common cycle choose offsets o_1 and o_2 such that $o_1 + o_2$ happens to be a coboundary — i.e. the *sum* of their two lies is indistinguishable from a single consistent state change — then the best-fit global section can absorb $o_1 + o_2$ entirely, leaving a near-zero residual, exactly the way a lone liar's offset is absorbed when it lies off a visible cycle (the cut-edge case above). Nothing in the definition of r requires the offsets to be individually small for their *sum* to be small or zero; it requires only that the sum land in the image of δ, the space the completion can always explain away. This is why the single-equivocator hypothesis is decisive rather than a simplifying nicety: the moment a second colluding reporter is admitted on the same cycle, the detector's soundness guarantee (Theorem 5.2's "if and only if") no longer applies to the *pair*, only to each alone under the counterfactual that the other is honest.

The compendium's own sweep states the shape of this exactly: a solo equivocator on an 8-cycle is bounded below at $r \geq |s|/\sqrt{8}$, while a coordinating coalition of two on the same cycle drives the same statistic to $r \approx 6 \times 10^{-15}$ — numerically the silence of a true coboundary, not a small-but-detectable residual [internal, `sheaf_consistency_radius.py`]. What could still catch the coalition is exactly what r is not: per-report attribution rather than a single scalar. Signature-level forensics can ask whether the *same* reporter's two prefixes are individually self-consistent (an equivocator still signs two incompatible things even if the aggregate looks clean); cross-checking against a third, independent visible cycle through only one of the two colluders can isolate that one's contribution if such a cycle exists; and widening K — comparing more edges directly rather than relying on relayed reports — shrinks the coalition's room to choose a cancelling pair in the first place. None of this is a proof that coalitions are caught in general; §12.3 names the cross-harbor cartel question as open for exactly this reason, and this exercise is that same boundary stated at the level of one theorem rather than the whole federation.

8.5 *(exercise on page 35)* The four principals are the daemon's local DB (D, read-only), the KMS (K, read-only key witness), the email provider (E, read access to the user's mailbox), and the passphrase (P, out-of-band, held only in the user's head — deliberately not modeled as compromisable, since §7 does not defend against the user revealing their own passphrase). The single query the file states is `query attacker(account_root)`: can the attacker ever derive the reconstructed vault. The model has no destructor for `combine4`, so the four shares can only be combined by the honest user's own process, never inverted by an attacker who merely holds some subset of them.

Only one scenario is actually run — the top-level `process` compromises D, E, and K simultaneously (three of four), leaving P uncompromised. The file's own comment states why

this subsumes the weaker cases: "if the vault is safe with daemon+email+KMS public, it is also safe with any subset of those compromised." An attacker who additionally lacks one of D, E, or K is strictly weaker than the modeled attacker, so a query that holds against the strongest three-of-four attacker holds against every subset of it for free; running the weaker scenarios separately would add nothing this run does not already dominate.

Pinned verdict (`proofs/bonded/federated/federated.run.log`), quoted exactly:

```
Starting query not attacker(account_root[])
RESULT not attacker(account_root[]) is true.
------------------------------------------------------------
Verification summary:

Query not attacker(account_root[]) is true.
```

What this does *not* cover, stated as plainly as §4.5 states its own draft status: all four principals compromised at once (explicitly out of scope — the four-eyes attacker), and the §7.5 Shamir 3-of-4 recovery path, which is the honest user's own reconstruction option, not an attacker capability. Neither this model nor the cross-realm sketch above substitutes for the other: one is a completed secrecy result about who must collude for a vault to fall, the other is this chapter's own in-progress model of cross-realm card authenticity.

8.6 *(exercise on page 35)* This is named open work (§12.1); per the memo's own framing, calling the witness log "the" correlation device without stating information and incentive constraints is not yet a result, so a defensible design obligation is what this exercise asks for, not a closed theorem. A stochastic/Bayesian game model: each harbor's daemon is a player with a private local view of its own event log, and a public signal that arrives only when both harbors' epoch roots for a common epoch e are simultaneously present in the witness log (a *common-view event*). Local daemon correlation devices — each daemon still correlates the agents inside its own harbor exactly as in Bonded — feed a delayed public signal that only exists conditional on that event. The proof obligation is a coarse correlated equilibrium *conditional on the common-view/freshness event*: during a window where both roots are mutually current, the witness log's joint state is common knowledge and can play Bonded's correlation role across the boundary; the moment freshness lapses (a partition, or one harbor lagging), no such joint signal exists, and the equilibrium claim cannot be made. The safe degraded policy for that lapsed case is to fall back to *local* correlated equilibrium within each harbor plus ordinary Nash reasoning about the other harbor's declared but unconfirmed state — i.e. treat an unconfirmed cross-harbor claim the way §5.2's authoritative rule already treats an unconfirmed revocation: trusted once witnessed, not before. Whether this weakens Bonded's result at the federation layer (the paper's second candidate resolution) or merely restates it under a freshness side-condition (the first) is exactly the open question; this sketch does not resolve it, it only makes the two candidates precise enough to attack.

8.7 *(exercise on page 35)* An HTLC's atomicity rests on a single objective, third-party-checkable bit: does a preimage of a fixed hash appear on-chain before a deadline. Both parties' payoffs are a deterministic function of that one bit, so the same contract executes identically for anyone who can read the chain. The Federated Harbor's damage predicate (§6.3's claim step) is not a single objective bit revealed by one party: whether α's action at B actually caused the acceptance-criteria failure is a semantic, evidence-weighing judgment over a work-unit's base and result trees, exactly the causation gap Exercise 8.1's solution names. A deterministic trustless contract must settle identically whenever its on-ledger inputs are identical; if two possible worlds produce the same signed evidence trail but differ in whether α was actually at fault, no deterministic on-ledger rule can be correct in both worlds at once. That is the shape of the impossibility this exercise is asked to sketch, not merely a practical

gap: it would need to assume exactly two worlds with identical observable transcripts and provably different ground truth, and show every candidate deterministic settlement rule is wrong in at least one of them. Progress, as §12.2 and §6.1's 2-of-3 candidate both note, means adding something outside pure on-ledger determinism — an oracle, a TEE, an optimistic dispute window, or a bonded human/arbiter adjudicator — each of which reintroduces a trust assumption rather than removing one. This paper gives neither the construction nor the impossibility proof; it gives the escrow's conditional bound (Design Invariant 6.1) as the honest fallback while the question stays open.

8.8 *(exercise on page 35)* For $p \cdot d \cdot L > G$ to be satisfiable against a majority-but-not-total cartel, the audit mechanism needs an independent source of p and d that the cartel does not itself control: cross-harbor adversarial audits sampled from outside the colluding set, witness diversification so that a majority of witnesses being compromised still leaves an honest witness able to observe and attest to the true state, and principal-wide collateral so that L scales with what the colluding *principals* (not just their harbor identities) have at stake, per §12.3's list. Concavity of repeated-game payoffs in reputation credit raises the effective G a cartel forfeits by being caught once, which is what makes $p \cdot d \cdot L > G$ achievable at plausible p rather than requiring p near 1. All of this requires that at least one witness and one audit path lie genuinely outside the cartel's control — the mechanism's leverage comes entirely from an outside observer existing.

That is exactly why no structural mechanism closes the all-witnesses case: if the cartel controls every oracle and witness in scope, there is no outside observer left to supply p or d at all, and the inequality has no independent terms to be satisfied with. This is not a gap this paper failed to close; it is the same limit every audit-based mechanism in the literature hits, stated honestly rather than argued around. §10's centralization-gravity discussion is the operational form of the same fact: the fewer independent witnesses a federation actually has, the closer it sits to the all-controlled boundary where no deterrence inequality can help.

8.9 *(exercise on page 35)* Four measurements, each cheap to compute off the witness-logged admission records §7 already produces because enrollment is permanent and public: **acceptance rate** (admitted harbors / sponsorship requests, over a trailing window); **sponsor concentration** (the Herfindahl index, or simply the top-sponsor's share, of successful sponsorships); **time-to-entry** (elapsed time from first sponsorship request to admission, for accepted applicants); and **false-rejection rate** (requests declined that a later, independent sponsor accepted without incident, as a proxy for rejections that were not actually risk-justified). A thin-but-open market shows a low acceptance rate *and* low sponsor concentration *and* a time-to-entry that is long but not systematically longer for any identifiable class of applicant. An invitation cartel shows the opposite signature on the second and fourth: concentration rising over time even as total admissions grow, and a persistent gap in acceptance rate or time-to-entry between applicants introduced by an incumbent sponsor and applicants without one. None of these four numbers is a proof of openness by itself — a legitimately small, honest market can also show high concentration simply because there are few sponsors to begin with — which is exactly why §12.4 calls the structural pushback engineering discipline rather than theorem: the measurement tells an operator when to worry, it does not certify the absence of the failure mode.

8.10 *(exercise on page 36)* They are different problems because they sit on different sides of the same boundary. The deployment-specific tail bound is an *empirical* gap: Theorem 5.1 already gives an expected $\Theta(\log m)$ result under a stated model (a complete overlay, reliable synchronous rounds), and what is missing is fitting that model's parameters — or a weaker model's — against a real peer graph's loss rates, fan-out, and partition frequency, exactly as §12.5's solution direction calls for instrumenting real peer graphs and fitting tail quantiles before pricing a bond from propagation risk. The sheaf-Laplacian question is a *mathematical*

gap: §5.4's closing paragraph is explicit that no relaxation-time bound has been derived or checked, because the Hansen–Ghrist rate governs continuous-time *linear* diffusion over real stalks, while anti-entropy gossip over signed append-only logs under a Byzantine equivocator is discrete, asynchronous, adversarial, and monotone-join rather than linear-averaging — closing it means either proving the two convergence stories are secretly the same rate, or proving they are provably different and explaining why the sheaf spectral gap is the wrong handle. Even a complete answer to one gap would not touch the other: a perfectly fitted empirical tail bound says nothing about whether $\lambda_2(\Delta_{\mathcal{F}})$ predicts it, and a proved reconciliation of the two convergence models would still need real peer-graph data to become a number an operator can price a bond against. The measurement needed before either can be closed is the same first step: real anti-entropy round traces from an actual multi-harbor deployment, since §12.5's empirical gap needs them directly and any attempt to validate a sheaf-diffusion reconciliation against gossip behavior needs the same trace data to compare against.

8.11 *(exercise on page 36)* What does not require consensus is the *event history*: harbors never need to agree on a single global order of issuances, revocations, or transfers. Each harbor's epoch roots only extend (per-harbor monotonicity), each root is cross-witnessed rather than centrally ordered, and equivocation is detectable rather than prevented by agreement (§2.4, §2.3). What the Cross-Harbor Bucket Partition result (Design Invariant 6.2) presupposes is narrower and different in kind: it is a model-checked invariant of one abstract transition system with a fixed set of buckets per bond, checked at a stated finite bound ($|F| = 4$, bond depth 16, §9.2). Apalache is not requiring the *harbors* to agree on anything at runtime; it is verifying that the *rules* governing one bond's bucket moves cannot, on paper, create or destroy value, for every reachable state up to the bound. Consensus among machines and exhaustive checking of a specification are different axes entirely — the first is a runtime coordination requirement the federation deliberately avoids, the second is an offline proof technique applied to a design the federation still has to implement correctly and run at unbounded scale. Nothing here claims the deployed custody ledger enforces the bound at runtime; that is exactly Design Invariant 6.1's conditional custody-ledger assumption, stated separately.

Calling the Federated Harbor "a project institution, not merely two machines sharing a repository" means the paper's actual claim is broader than any one protocol: an authoritative event/work-unit graph, per-principal policy, local effect mediators, causal messages, evidence, roles, and explicit governance over conflicts, of which the transfer ceremony, revocation mesh, and settlement escrow are three essential pieces, not the whole structure. A roadmap or a stated intention is versioned forecast, not truth, and a contradiction detector (the witness log's equivocation check) produces evidence for a *named principal or role* to decide against, rather than deciding by itself. The paper's own posture — consensus only where it is cheap and structurally necessary (the small authoritative metadata core: epoch roots, admission records), federation everywhere else (the actual work artifacts) — is the concrete answer to the framing question of §1: two machines needed more than a shared repository, and what they needed is exactly the institution this paper specifies one layer of.

References

[1] Erich Owens. The Anchor Protocol: A Mechanically Analyzed Control Plane for Local Agent Swarms. Curiositech LLC Technical White Paper, May 2026. Cited on pages 6 and 31.

[2] Erich Owens. The Bonded Commons: Pre-Transactional Trust Infrastructure for Multi-Agent Systems. Curiositech LLC Technical White Paper, May 2026. Cited on pages 6, 8, 32, and 44.

[3] Erich Owens. The Federated Harbor: Proposal and Annotated Bibliography. Curiositech LLC working document, May 2026. Cited on pages 7, 17, 33, and 45.

[4] Erich Owens. The Cohomology of Equivocation: Detecting Split-View Lies in Federated Witness-Log Gossip by Sheaf Consistency. The Harbor Research Program, Paper 7, August 2026. Cited on page 18.

[5] Arnar Birgisson, Joe Gibbs Politz, Úlfar Erlingsson, Ankur Taly, Michael Vrable, and Mark Lentczner. Macaroons: Cookies with Contextual Caveats for Decentralized Authorization in the Cloud. In *Network and Distributed System Security Symposium (NDSS)*, 2014. Cited on page 31.

[6] Jack B. Dennis and Earl C. Van Horn. Programming Semantics for Multiprogrammed Computations. *Communications of the ACM*, 9(3):143–155, 1966. Cited on page 31.

[7] Mark S. Miller. Robust Composition: Towards a Unified Approach to Access Control and Concurrency Control. PhD thesis, Johns Hopkins University, 2006. Cited on page 31.

[8] Danny Dolev and Andrew Yao. On the Security of Public Key Protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983. Cited on page 10.

[9] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. High-Speed High-Security Signatures. *Journal of Cryptographic Engineering*, 2(2):77–89, 2012. Cited on page 10.

[10] Ben Laurie, Adam Langley, and Emilia Kasper. Certificate Transparency. IETF RFC 6962, June 2013. Cited on pages 9, 15, 31, and 32.

[11] Zachary Newman, John Speed Meyers, and Santiago Torres-Arias. Sigstore: Software Signing for Everybody. In *ACM Conference on Computer and Communications Security (CCS)*, 2022. Cited on pages 9, 30, and 32.

[12] Santiago Torres-Arias, Hammad Afzali, Trishank Karthik Kuppusamy, Reza Curtmola, and Justin Cappos. in-toto: Providing farm-to-table guarantees for bits and bytes. In *USENIX Security Symposium*, 2019. Cited on pages 31 and 32.

[13] Linux Foundation / OpenSSF. SLSA Specification v1.0: Supply-chain Levels for Software Artifacts, 2023. Cited on pages 31 and 32.

[14] Scott A. Crosby and Dan S. Wallach. Efficient Data Structures for Tamper-Evident Logging. In *USENIX Security Symposium*, 2009. Cited on page 32.

[15] Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry. Epidemic Algorithms for Replicated Database Maintenance. In *ACM Symposium on Principles of Distributed Computing (PODC)*, 1987. Cited on pages 15, 15, 16, 23, and 44.

[16] Maurice Herlihy. Atomic Cross-Chain Swaps. In *ACM Symposium on Principles of Distributed Computing (PODC)*, 2018. Cited on pages 7, 24, 32, and 33.

[17] John R. Douceur. The Sybil Attack. In *International Workshop on Peer-to-Peer Systems (IPTPS)*, 2002. Cited on pages 10 and 26.

[18] Eve Maler and Drummond Reed. The Venn of Identity: Options and Issues in Federated Identity Management. *IEEE Security & Privacy*, 6(2):16–23, 2008. Cited on pages 11, 27, 27, 30, 30, 31, and 33.

[19] Christine Lemmer-Webber, Jessica Tallon, Erin Shepherd, Amy Guy, and Evan Prodromou (eds.). ActivityPub. W3C Recommendation, 2018. Cited on page 31.

[20] The Matrix.org Foundation. The Matrix Specification. `https://spec.matrix.org`. Cited on page 32.

[21] Carl M. Ellison, Bill Frantz, Butler Lampson, Ron Rivest, Brian M. Thomas, and Tatu Ylonen. SPKI Certificate Theory. IETF RFC 2693, 1999. Cited on page 32.

[22] Philip R. Zimmermann. The Official PGP User's Guide. MIT Press, 1995. Cited on page 27.

[23] Alfarez Abdul-Rahman. The PGP Trust Model. *EDI-Forum: The Journal of Electronic Commerce*, 10(3):27–31, 1997. Cited on page 30.

[24] UCAN Working Group. UCAN Specification 1.0: User Controlled Authorization Network, 2024. Cited on page 31.

[25] Roland Hedberg (ed.), Michael B. Jones, Andreas Åkre Solberg, John Bradley, Giuseppe De Marco, and Vladimir Dzhuvinov. OpenID Federation 1.0. OpenID Foundation Implementer's Draft 4, July 2024. Cited on page 31.

[26] Christopher Goes. The Interblockchain Communication Protocol: An Overview. arXiv:2006.15918, 2020. Cited on page 32.

[27] Philipp Schindler, Aljosha Judmayer, Nicholas Stifter, and Edgar Weippl. ETHDKG: Distributed Key Generation with Ethereum Smart Contracts. IACR ePrint 2019/985. Cited on page 7.

[28] Andrew Tobin and Drummond Reed. The Inevitable Rise of Self-Sovereign Identity. Sovrin Foundation White Paper, 2017. Cited on page 7.

[29] Avery Pennarun et al. How Tailscale Works. Tailscale Engineering Blog, March 2020. Cited on page 30.

[30] Juan Brackeva et al. Headscale: Open-Source Tailscale Coordination Server. GitHub project, 2020–present. Cited on page 30.

[31] Bruno Blanchet. Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif. *Foundations and Trends in Privacy and Security*, 1(1-2):1–135, 2016. Cited on page 29.

[32] Universal Commerce Protocol. Specification and reference implementation (Apache-2.0), Google, Shopify, et al., 2026. `https://ucp.dev`. Cited on page 32.

[33] Agent Payments Protocol (AP2): Mandates. Verifiable-credential authorization chain: SD-JWT+kb, JWS detached content, JCS canonicalization, 2026. `https://ap2-protocol.org`. Cited on page 32.

[34] Samson Abramsky and Adam Brandenburger. The operational-sheaf-theoretic approach to contextuality and non-locality. *New Journal of Physics*, 13(11):113036, 2011. Cited on page 19.

[35] Jakob Hansen and Robert Ghrist. Toward a spectral theory of cellular sheaves. *Journal of Applied and Computational Topology*, 3(4):315–358, 2019. Cited on page 23.

[36] Michael Robinson. Sheaves Are the Canonical Data Structure for Sensor Integration. *Information Fusion*, 36:208–224, 2017. Cited on page 21.

[37] Giovanni Carù. On the Cohomology of Contextuality. In *Proc. 13th International Conference on Quantum Physics and Logic (QPL 2016)*, EPTCS 236, pp. 21–39, 2017. Cited on page 23.

[38] Peiyao Sheng, Gerui Wang, Kartik Nayak, Sreeram Kannan, and Pramod Viswanath. BFT Protocol Forensics. In *Proc. ACM Conference on Computer and Communications Security (CCS)*, pp. 1722–1743, 2021. Cited on page 23.

[39] Daniel A. Spielman and Shang-Hua Teng. Nearly-Linear Time Algorithms for Graph Partitioning, Graph Sparsification, and Solving Linear Systems. In *Proc. 36th ACM Symposium on Theory of Computing (STOC)*, pp. 81–90, 2004. Cited on page 22.

[40] Eric Bach. Sheaf Cohomology is #P-hard. *Journal of Symbolic Computation*, 27(4):429–433, 1999. Cited on page 22.

A Verification Status

Artifact registry. The cross-paper status table in Bonded Commons [2] carries the items that are shared across the series; this appendix lists only the items specific to the Federated Harbor.

Table 2: Federated Harbor verification status. **closed**: model verified and runtime conformance test passes. PARTIAL: design specified, mechanization in draft. *open*: known unmechanized; threat band visible in Figure 1 (cobalt).

Claim	Method	Result	Artifact
Single-harbor capability authenticity (inherited)	ProVerif 2.05	**closed**	Anchor `harbor_card_v*.pv`
Cross-realm authenticity (§4)	ProVerif (draft)	PARTIAL pending witness-log freshness assumption	`fh-xfer.pv` (draft)
Attenuation monotonicity across boundary (Lem. 4.1)	ProVerif + Anchor sub-result	PARTIAL	`fh-att.pv` (draft)
Federated revocation dissemination (Prop. 5.1)	Conditional epidemic model [15]	PARTIAL expected asymptotic result; no partition deadline	`fh-conv.tex` (proof sketch)
Cross-witness equivocation evidence	Signed-root comparison; dissemination model	PARTIAL verification immediate once views meet; no hard delivery bound	—
Escrow extraction bound (Prop. 6.1)	Transition-system case analysis	PARTIAL conditional on non-bypassable custody; no runtime conformance	`fh-escrow.pv` (draft)
Cross-harbor Conservation (§9.2)	TLA+ / Apalache, $\|F\| \leq 4$	PARTIAL model-checked at scale	`CrossHarbor Conservation.tla`
Sheaf detection of equivocation (Thm. 5.2)	Pre-registered statistical harness, 200 trials/arm	PARTIAL gates pass at the stated scope (single equivocator, real stalks, prefix restrictions); not mechanized	`sheaf_harness_v2.py`
Sheaf-Laplacian relaxation-time bound (§5.4)	—	*open* no derivation and no citation; the gossip and linear-diffusion models are not reconciled	—
Trustless settlement for non-fungible bonds	—	*open* no construction or impossibility proof	—

(continued)

Claim	Method	Result	Artifact
Multi-principal correlated-equilibrium extension	—	*open*	—
Cross-federation cartel-resistance bound	—	*open*	—

What this table means. Every row in the PARTIAL band is committed to and on the active workplan. Every row in the *open* band is a research question we cannot resolve from the desk; the proposal [3] names collaborators whose contributions are necessary to close them. We err toward listing fewer items as closed rather than more, in the same posture as Anchor and Bonded.

B ProVerif Models (draft)

For brevity we omit the full model text in this pre-print; the draft `fh-xfer.pv` extends Anchor's `harbor_card_v3_delegation.pv` with a second signing key sk_B, an epoch root binding, and a cross-realm authenticity query of the shape sketched in §9.1. The full model will accompany the revised version once the witness-log freshness assumption is mechanized rather than assumed.

C TLA+ Specification (draft)

The full `CrossHarborConservation.tla` extends Bonded's Conservation specification with per-harbor pools, an escrow pool, and an inter-harbor transit relation. Apalache symbolic model-checking at $|F| = 4$ harbors and bond depth 16 establishes that Conservation holds across all reachable states under the transition relation. The full specification accompanies the revised version.