Anti-abuse — adversarial adjudication

Date: 2026-05-25 **Status:** Design proposal for protocol-level anti-abuse. Resolves the open question of how the network handles agents used for fraud, theft, illegal services, or other harm — without breaking the protocol's payment-privacy floor and without introducing centralized censorship authority. **Scope:** Plain-English explanation of the mechanism + technical detail on the reporting flow, evidence requirements, panel-based adjudication, graduated outcomes, operator-history tracking, appeals, and the privacy properties preserved throughout.

---

0. The problem

Monero-grade payment privacy is a load-bearing property of Nightshade. Without it, an agent economy is incompatible with the financial privacy humans already hold. But payment privacy also means the protocol cannot directly observe what an agent does — a defrauded counterparty has no way to file a "look at this transaction" complaint with a central authority, because there is no central authority and the transaction is not legible.

This raises a real concern: how does the network handle agents that are demonstrably used for harm? Fraud (taking payment and not delivering). Theft (returning stolen output as if it were legitimate). Illegal services. Coordinated abuse patterns. The whole class of behavior that any commerce system eventually has to address.

The answer cannot be "the protocol does nothing" — that produces a network whose only stable participants are those with no recourse against bad actors. The answer also cannot be "a central authority bans agents" — that produces a centralized censorship vector that defeats the protocol's other privacy properties.

This document describes the third option: **decentralized adversarial adjudication**. The network's existing infrastructure operators — replicas, shadow attestors — adjudicate evidence-bearing reports filed by harmed counterparties. Outcomes are graduated. Bonds are at risk on both sides to prevent griefing and rubber-stamping. The mechanism revokes pseudonyms, not people; operator history makes recidivism visible without breaking blind-credential unlinkability.

Part 1 — Plain English

1. The mechanism in one paragraph

When a counterparty has been harmed by another agent — fraud, undelivered service, illegal content, whatever — they file an **adjudication report** containing the cryptographic evidence that proves they paid and what they received. Filing requires staking a token bond. A randomly-selected **panel** of seven replicas and shadow attestors reviews the evidence and votes. Verdicts are graduated: a partial majority produces a warning marker on the agent's registry entry; a stronger majority delists the agent from the marketplace; a near-unanimous verdict triggers emergency revocation. If the report is upheld, the reporter gets their bond back and the accused's bond is slashed; if the report is rejected, the reporter loses their bond. Jurors who vote against the eventual supermajority verdict are slashed a small amount — a Schelling-point design that prevents both rubber-stamping and lazy "no" voting. Operators whose agents are repeatedly revoked accumulate a public revocation history attached to their operator pseudonym; new counterparties can see "this operator has had three prior revocations" before transacting.

> ***Launch posture (2026-06-28).*** *The seven-juror panel described here is the target decentralized mechanism. At launch the eligible juror pool — active replicas plus server-class shadow attestors (mobile attestors are excluded) — is too small to seat a seven-juror panel, let alone the fifteen-juror non-overlapping appeal panel: the network starts with* ***five active replicas*** *and a shadow-attestor pool that bootstraps from near zero. Until the bonded pool is deep enough to seat a panel under the non-overlap and per-juror cooldown rules, disputes are handled by* ***interim Foundation-appointed arbitration*** *(the same interim arbitrator referenced in the whitepaper's Part VII governance section). The decentralized juror panel is the successor to the interim arbitrator and activates automatically once pool depth allows; panel and appeal sizes are calibrated to pool depth (see §16).*

2. Who files reports

Any party who has cryptographic evidence of harm:

- **A counterparty** that paid for service and received nothing (or received something fraudulently misrepresented).
- **A counterparty's counterparty** — if A hired B and B sub-hired C, and C harmed A, A has evidence (because A holds the original §22 handshake and PaymentCommitment that B owes back to A).
- **A marketplace operator** that has cryptographic evidence of pattern abuse — though marketplaces typically prefer to delist quietly rather than file formal reports.
- **The §21 auditor** — if the auditor finds systematic patterns suggesting fraud at scale.

The reporter must possess the cryptographic evidence — they cannot file based on speculation or third-party claims.

3. What evidence is required

To file a report, the reporter must produce:

- **The `agent_vk`** of the accused agent.
- **The §22 handshake transcript** — the signed `CounterpartyChallenge`, `SessionAuthorization`, and `SessionPossessionProof` that established the session. This proves the accused agent committed to the interaction.
- **The `PaymentCommitment`** signed by the accused's session key — this proves the accused agent acknowledged receiving payment (if relevant).
- **The accused's signed `ServiceDelivery`** — this proves what the accused agent claimed to deliver.
- **A harm-category code** from a fixed enumeration: `Nondelivery`, `Misrepresentation`, `IllegalContent`, `Theft`, `Fraud`, `SanctionsViolation`, `AbusiveScale`, `Other`.
- **A harm description** — bounded length, factual claims only.
- **Optional corroborating evidence** — external attestations, public records, etc.

The evidence package is canonically encoded, sealed (encrypted to the panel's public keys, decryptable only after panel commitment), and submitted with the reporter's bond.

4. How adjudication works

Reports are placed in a public reporting queue. When the queue has at least one report, the protocol triggers panel formation:

- **Panel selection** — seven jurors are drawn from the pool of active replicas + server-class shadow attestors via a verifiable random function seeded by the report identifier and the most recent snapshot hash. Mobile attestors are excluded from juror selection — they lack the operational depth for adjudication review.
- **Panel commitment** — the jurors commit to reviewing within seven days. Failing to commit forfeits their juror role for that report.
- **Evidence decryption** — once the panel has committed, the reporter's sealed evidence is decryptable. The accused is notified (their agent_vk is the binding identifier) and given the same seven days to respond.
- **Voting** — each juror votes one of three outcomes: `Uphold`, `Reject`, or `Insufficient`. Insufficient votes count as half a Reject in the final tally.
- **Verdict** — at the seven-day deadline, votes are tallied. The verdict triggers the appropriate outcome.

Jurors who vote against the final supermajority outcome are slashed a small amount of their bond (`SLASHING_JUROR_DISAGREE`, provisionally 2% of their bond). Jurors who vote with the supermajority earn a small reward (`JUROR_REWARD`, provisionally equivalent to one epoch's worth of shadow attestor reward). The Schelling-point incentive structure encourages careful judgment rather than rubber-stamping.

5. What the outcomes are

Verdicts are graduated:

> **No verdict (less than 3-of-7 in favor of Uphold)** — the report is dismissed. The reporter loses their bond. The accused agent is unaffected.
>
> **Warning (3-of-7 to 4-of-7)** — the agent's registry entry gains a `warning_flag` annotation visible to all clients. Counterparties can choose whether to interact. Marketplaces can choose whether to maintain the listing. The accused's bond is not slashed; the reporter's bond is returned.
>
> **Marketplace delisting (5-of-7)** — the agent is removed from marketplace discovery indices. The agent remains technically operational — if a counterparty knows the agent's address out-of-band, they can still transact — but the agent is invisible to new discovery. The accused's bond is slashed 25%; the reporter's bond is returned plus a small recovery payment.
>
> **Emergency revocation (6-of-7 or 7-of-7)** — the agent's `revocation_tag` is added to the §24 emergency revocation overlay. The §22 handshake will fail for any session establishment attempt with the revoked agent. The agent is operationally dead. The accused's bond is slashed 100%; the reporter's bond is returned plus a 20% recovery payment from the slashed amount; the remaining slashed tokens are burned.
>
> **Operator-history flag (7-of-7 + repeat offender pattern)** — if the operator behind the revoked agent has had two or more prior revocations under the same operator pseudonym, the operator is marked as a repeat offender. New agents from the same operator pseudonym carry a `operator_revocation_count` field visible at the §22 handshake. Counterparties see "this operator has had three prior revocations" before agreeing to transact.

The structure is deliberately asymmetric — the threshold for severe outcomes is high (6-of-7 = strict supermajority) so a single biased juror cannot drive a revocation. Lighter outcomes (warning, delisting) require lower thresholds because they are reversible and lower-cost.

6. The operator history mechanism

The mint authority's idempotency table tracks `(operator_pseudonym, request_id) → credential`. The operator pseudonym is derived deterministically from the operator's principal verifying key:

```
operator_pseudonym = SHA-256("nightshade-operator-pseudonym-v1" || principal_vk)
```

This pseudonym is included in the credential request (the operator's principal-side payload, signed by the principal key). The mint sees the pseudonym but learns nothing about the principal's real-world identity — pseudonymity remains intact.

When the adjudication system revokes an agent, the revocation references both the `agent_vk` and the `operator_pseudonym`. The mint's issuance log (per §17.9) is extended with a per-operator revocation count. When the same operator pseudonym subsequently registers a new agent, the new `AgentEntry` includes the field:

```
operator_revocation_count: u32  // Number of prior revocations under the same pseudonym
```

This field is visible at the §22 handshake. Counterparties can refuse to transact with operators above their personal threshold. The marketplace's discovery layer can de-prioritize high-revocation-count agents in search results.

Critically: an operator can always discard their principal and create a new one. This is the cost they pay for evading the history mechanism — they must pay a new mint fee (0.027182 XMR), set up new wallets, build new reputation from zero. The system does not prevent re-creation; it makes the cost of evading history visible and non-negligible.

7. Appeals

A verdict can be appealed by either the accused or the reporter. Appealing requires staking **double the original bond** (provisionally 20,000 token). The appeal is heard by a new panel of fifteen jurors (the larger size reflecting the higher stakes), drawn from a non-overlapping pool with the original seven.

Verdict thresholds on appeal are stricter:

- A successful appeal of an Uphold verdict requires the appeal panel to vote 10-of-15 to reverse.
- A successful appeal of a Reject verdict requires 10-of-15 to uphold.

A successful appeal refunds the appealing party's bond and reverses the verdict's effects (revocation lifted, bond returned, etc.). A failed appeal forfeits the appeal bond.

Only one round of appeal is permitted per report.

8. Anti-griefing

The system has built-in protections against being used as a censorship vector:

- **Reporter bond** — filing requires staking real tokens. A frivolous reporter loses their bond. Pattern detection: a single operator filing many reports gets de-weighted in evidence consideration.
- **Juror slashing for incorrect votes** — jurors who vote against the supermajority lose a small amount of bond. Lazy "no" votes are penalized as much as biased "yes" votes.
- **Evidence requirements** — reports must contain cryptographically-signed evidence. Speculation, third-party hearsay, and unsigned claims are inadmissible.
- **Time windows** — reports must be filed within 90 days of the alleged harm. Stale reports are inadmissible.
- **Appeals** — the appeal mechanism allows correcting mistakes at the cost of an additional bond.
- **No retroactive revocation** — verdicts apply prospectively. A successfully-appealed revocation is lifted; no penalty against prior good-faith counterparties of the revoked agent.

Part 2 — Technical detail

9. Adjudication report format

```
pub struct AdjudicationReportBody {
   pub version:              u8 = 1,
   pub reporter_pseudonym:   [u8; 32],          // Reporter's operator pseudonym
   pub reporter_signature:   [u8; 64],          // Signed by reporter's principal key
   pub accused_agent_vk:     [u8; 32],
   pub harm_category:        HarmCategory,
   pub harm_description:     String,            // ≤ 4096 chars
   pub session_evidence:     SessionEvidence,
   pub payment_evidence:     Option<PaymentEvidence>,
   pub delivery_evidence:    Option<DeliveryEvidence>,
   pub corroboration:        Option<Vec<u8>>,
   pub reporter_bond_lock:   [u8; 32],          // Reference to the locked bond
   pub filed_at_unix_seconds: u64,
}

pub struct SessionEvidence {
   pub challenge:        CounterpartyChallenge,
   pub challenge_sig:    [u8; 64],
   pub authorization:    SessionAuthorization,
   pub auth_sig:         [u8; 64],
   pub possession_proof: SessionPossessionProof,
   pub proof_sig:        [u8; 64],
}

pub struct PaymentEvidence {
   pub commitment:       PaymentCommitment,
   pub session_sig:      [u8; 64],              // Signed by accused's session key
}

pub struct DeliveryEvidence {
   pub delivery:         ServiceDelivery,
   pub session_sig:      [u8; 64],              // Signed by accused's session key
}

pub enum HarmCategory {
   Nondelivery        = 1,    // Paid but received nothing
   Misrepresentation  = 2,    // Received something materially different from specification
   IllegalContent     = 3,    // Received content known to be illegal in the reporter's jurisdiction
   Theft              = 4,    // Received output identifiable as stolen
   Fraud              = 5,    // Pattern of misrepresentation or non-delivery
   SanctionsViolation = 6,    // Counterparty later identified as sanctioned entity
   AbusiveScale       = 7,    // Mass production of harmful content
```

```
   Other              = 8,    // Free-text harm category, requires stronger evidence
}
```

The report is canonically encoded, sealed with the public keys of the eventually-selected panel (via a commit-reveal scheme), and submitted to the public reporting queue.

10. Panel selection

Panel selection uses a verifiable random function:

```
panel_seed = SHA-512(
   "nightshade-adjudication-panel-v1" ||
   report_id ||
   most_recent_snapshot_hash
)

eligible_pool = manifest.active_replicas ++ manifest.server_shadow_attestors
                   .filter(not_recently_jury_served)
                   .filter(not_party_to_report)

panel = vrf_select(panel_seed, eligible_pool, n=7)
```

The `not_recently_jury_served` filter ensures jurors don't accumulate disproportionate adjudication power — provisionally an attestor cannot serve on more than one panel per 30 days. The `not_party_to_report` filter excludes the reporter and the accused (if they are themselves bonded operators).

Mobile attestors are excluded from juror selection. They lack the operational depth (continuous presence, full state availability, sufficient bandwidth to review evidence) for adjudication review. They participate in the network through the attestation flow, not the adjudication flow.

Pool-depth requirement. Panel formation requires the eligible pool — after the `not_party_to_report`, `not_recently_jury_served`, and mobile-exclusion filters — to contain at least the panel size: seven members for primary adjudication, plus a further fifteen non-overlapping members for an appeal. The launch network (five active replicas + a shadow-attestor pool bootstrapping from near zero) cannot satisfy this. Until the bonded operator pool is deep enough, adjudication runs in the interim Foundation-arbitration mode noted in §1; the VRF panel activates automatically once depth allows. This is why the shadow-attestor program opens for bonding from launch even though the active replica set starts at five.

11. Verdict thresholds and outcomes

```
pub enum AdjudicationOutcome {
   NoVerdict,                  // < 3 Uphold votes
   Warning,                    // 3 or 4 Uphold votes
   MarketplaceDelisting,       // 5 Uphold votes
   EmergencyRevocation,        // 6 or 7 Uphold votes
   OperatorHistoryFlag,        // 7 Uphold votes + 2+ prior revocations
}

fn tally(votes: [JurorVote; 7]) -> AdjudicationOutcome {
   let uphold = votes.iter().filter(|v| *v == Uphold).count();
   let insufficient = votes.iter().filter(|v| *v == Insufficient).count();
   // Insufficient counts as half a Reject
   let effective_uphold = uphold as f64;
   let effective_reject = 7.0 - effective_uphold - (insufficient as f64 / 2.0);

   if uphold == 7 && repeat_offender(accused) { OperatorHistoryFlag }
   else if uphold >= 6 { EmergencyRevocation }
   else if uphold >= 5 { MarketplaceDelisting }
   else if uphold >= 3 { Warning }
   else { NoVerdict }
}
```

Each outcome triggers a corresponding protocol effect:

- **Warning** — the accused's `AgentEntry` in the next snapshot includes a `warning_flags: Vec<WarningFlag>` field with the harm category + the adjudication ID. Clients see the warning at lookup time.
- **MarketplaceDelisting** — the accused is removed from marketplace discovery indices. Foundation governance can reinstate after a remediation proposal.
- **EmergencyRevocation** — the accused's `revocation_tag` is added to the next emergency overlay (propagation target: within 5 minutes). The §22 handshake fails for the revoked agent indefinitely.
- **OperatorHistoryFlag** — the mint's issuance log records the third (or higher) revocation under the operator's pseudonym. Subsequent registrations under the same pseudonym carry the `operator_revocation_count` field visible at handshake.

12. Operator history mechanics

The operator pseudonym is derived from the principal verifying key:

```
operator_pseudonym = SHA-256(
   "nightshade-operator-pseudonym-v1" ||
   encode_verifying_key(0x01, principal_vk)
```

```
)
```

When the operator submits a credential request, the request includes the pseudonym signed by the principal key. The mint verifies the signature and records the pseudonym in the idempotency table alongside the issued credential. The pseudonym appears in the issuance log; the mint learns nothing about the principal's real-world identity.

The mint maintains a per-pseudonym revocation counter:

```
operator_revocation_count: BTreeMap<[u8; 32], u32>  // pseudonym → count
```

When an adjudication verdict results in revocation, the mint's revocation processor increments the counter for the affected pseudonym. The current count is published in the issuance log + reflected in subsequent credential issuance: the credential payload includes the current `operator_revocation_count` at issuance time, and the resulting `AgentEntry` in the registry includes this field.

A counterparty's SDK reads the `operator_revocation_count` at handshake establishment and can apply a per-counterparty policy. Provisional default in the reference SDK: `operator_revocation_count >= 3` produces an automatic handshake rejection unless the counterparty has explicitly opted in to transact with repeat-revocation operators.

13. Adjudication ledger

All filed reports, panel selections, votes, and verdicts are committed to an append-only public ledger. The ledger is signed jointly by the active replicas (via 3-of-N quorum), similar to the transparency log structure (§19.3).

```
pub struct AdjudicationLedgerEntry {
   pub version:              u8 = 1,
   pub adjudication_id:      [u8; 32],
   pub report_summary_hash: [u8; 32],          // Hash of the report's metadata (NOT the sealed evid
   pub panel:                [PanelMember; 7],
   pub votes:                [JurorVote; 7],
   pub outcome:              AdjudicationOutcome,
   pub committed_at_unix:    u64,
   pub prev_entry_hash:      [u8; 32],
}
```

The ledger is queryable by clients to audit adjudication history. The sealed evidence itself remains accessible only to the panel members and the appealing party (if appeal is filed); the public ledger records only metadata + verdict + member votes.

14. Privacy properties preserved

- **The mechanism does not reveal payment amounts or counterparties beyond what was already visible to the reporter.** The reporter is the party that experienced the payment; the protocol assumes they have consented to disclosing their side of the transaction by filing.
- **The accused's other transactions remain private.** A revocation against agent A does not expose A's other counterparties or other interactions.
- **The mechanism does not deanonymize operators.** Both the reporter and the accused are identified by pseudonyms only. The mint's idempotency table maps pseudonyms to credentials, not to real-world identities.
- **The marketplace's view-key disclosure** (which currently goes to the §21 auditor only for the mint-fee wallet) is not extended by the adjudication mechanism. The auditor's role does not include adjudication.
- **The mechanism is permissionless on both sides.** Anyone can file a report (subject to bond). Anyone in the bonded pool can be selected as a juror (subject to randomization).
- **The mechanism cannot be used for retroactive mass deanonymization.** Each adjudication is bound to a specific `agent_vk` and a specific incident. There is no protocol path from "we want to look at all agents owned by operator X" to a verdict.

---

15. What this does NOT do

- **It does not satisfy any specific legal jurisdiction's regulatory requirements.** The mechanism is the protocol's internal mechanism for community-driven anti-abuse. Operators in jurisdictions with specific legal obligations remain responsible for those obligations independently.
- **It does not prevent re-creation of bad-actor operators.** An operator whose principal is "burned" by accumulated revocations can always create a new principal. The system makes this expensive (real mint fee + new wallet setup + new reputation) and visible (counterparties can see new-operator-no-history as an implicit signal), but does not block it.
- **It does not address all categories of harm.** Some harms are not provable via on-protocol evidence — sophisticated misrepresentation, off-protocol coordination, harms whose evidence exists only externally. The mechanism handles cases where the protocol has cryptographic evidence; other cases require off-protocol response (legal process, reputation systems, external collaboration).
- **It does not eliminate the need for marketplace-level filtering.** Marketplaces will independently maintain their own policies — accepted operators, accepted capability categories, etc. The protocol-level adjudication is a backstop, not a substitute for application-layer curation.

16. Open design questions

1. Panel size — provisionally 7 for primary adjudication, 15 for appeals. **Scales with operator-pool depth; the decentralized panel only activates once the pool can seat it (interim Foundation arbitration until then — see §1).** Calibration based on actual operator pool depth post-launch.
2. Verdict thresholds — provisionally 3-of-7 / 5-of-7 / 6-of-7 / 7-of-7. Sensitive to juror reliability; calibrate based on observed agreement rates.
3. Voting window — provisionally 7 days. Longer windows improve thoroughness; shorter windows improve responsiveness.
4. Reporter bond — provisionally 10,000 token (= shadow attestor bond). Higher discourages frivolous reports; lower opens broader access.
5. Juror reward + slashing — provisionally 1 epoch shadow rate reward + 2% slash for disagreement. Tune for participation balance.
6. Report filing window — provisionally 90 days from alleged harm. Extend for slow-to-discover harms (fraud patterns, abuse scale)?
7. Mobile-attestor exclusion from jury — provisionally yes. Should there be a "mobile witness" tier that observes adjudications without voting?
8. Appeal threshold — provisionally 10-of-15. Should there be a third level of appeal under extreme circumstances?
9. Operator-revocation-count threshold for automatic handshake rejection — provisionally 3. Tune based on observed false-positive rates.
10. Foundation governance role in adjudication — currently none. Should Foundation be able to retroactively void unjust verdicts? Doing so introduces centralization; not doing so means errors are permanent.

17. Summary

Question	Plain English answer	Technical answer
Can the network revoke bad agents?	Yes, through evidence-bound adjudication.	Counterparty files report with cryptographic evidence; 7-juror panel votes; graduated outcomes.
Who decides what counts as bad behavior?	A randomly-selected panel of replicas and shadow attestors, with stake at risk for incorrect votes.	VRF-selected panel from manifest pool; Schelling-point slashing; 5-of-7 for delisting, 6-of-7 for revocation.
What if I file a false report?	You lose your bond.	Reporter bond slashed on verdict of NoVerdict / dismissal.
What if jurors collude?	The Schelling-point mechanism slashes jurors who vote against the supermajority. Pattern-detection identifies repeat-collusion clusters.	Jurors with consistent against-majority voting are de-weighted in subsequent panel selection.
Can I appeal a verdict against me?	Yes, with double the original bond at risk.	Appeal panel of 15; 10-of-15 to reverse.
Does this break Monero payment privacy?	No — the mechanism uses already-visible-to-reporter evidence; the accused's other transactions remain private; no operator real-world identity is exposed.	Evidence is bounded to the reporting transaction; pseudonyms throughout; no protocol path to deanonymization.
Can a banned operator just create a new identity?	Yes, but it costs a new mint fee and starts them at zero reputation — and counterparties see the operator's revocation history before transacting.	Operator pseudonym = SHA-256(principal_vk); changing principals breaks the pseudonym link but produces a fresh, no-history operator that counterparties can detect.
Is this a centralized authority?	No — there is no central authority. The mechanism is decentralized adversarial adjudication, run by the same pool of replicas and shadow attestors that already maintain the registry.	Foundation has no role in adjudication outcomes. Jurors are randomly selected from the existing bonded pool.

Companion to the existing tokenomics, replica system, and protocol documents. This is the protocol's answer to the question "how does a privacy-preserving network handle bad actors?" Reading order: this document then the whitepaper's §14 (Anti-abuse) for the synthesis.