Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: 27 October 2026

B. stone
SwarmSync.AI

25 April 2026

# ADRP: Agent Dispute Resolution Protocol
draft-stone-adrp-00

Abstract

This document defines the Agent Dispute Resolution Protocol (ADRP), a wire protocol and state machine for resolving disputes that arise from cryptographically-attested agent-to-agent (A2A) transactions. ADRP is the companion specification to ATXN (draft-stone-atxn-00), which defines what an A2A transaction is. ADRP defines what happens when a party contests one.

ADRP severs an equivalence that every prior agentic commerce design has implicitly assumed: that a valid cryptographic proof bundle equals contractual satisfaction. It does not. Conduit-style cryptographic verifiers prove that an agent took specified actions; they do not prove that those actions satisfied the principal’s Intent Mandate. ADRP bifurcates disputes into a cryptographic class (resolvable by code from the proof bundle and mandate chain) and a semantic class (resolvable only against pre-committed machine-readable acceptance criteria, with arbitration escalation when those criteria are absent or under-specified).

ADRP introduces the Arbitration Mandate as a fourth element of the AP2 mandate chain (Intent / Cart / Payment / Arbitration), pre-signed by the principal at agent-deployment time. The Arbitration Mandate is the FAA Section 2-compliant written agreement that anchors the entire protocol legally without statutory change.

ADRP defines a counter-attestation override pattern in which a signed RulingBundle supersedes a Conduit ProofBundle by a signing-time precedence rule rather than by mutation. Both the original attestation and the override are preserved forever in the hash chain; "override" is a verification-time computation, not a write.

Companion specifications (all co-submitted as Internet-Drafts, work in progress):

* ATXN (draft-stone-atxn-00): defines the A2A transaction primitive that ADRP resolves disputes over

* AIVS (draft-stone-aivs-01): cryptographic audit-trail substrate for proof bundles

* VCAP (draft-stone-vcap-01): verified-commerce escrow rails consumed by ADRP EscrowDirectives

* ATEP (draft-stone-atep-01): trust passports referenced by Standing Tokens in ADRP

## Status of This Memo



## Copyright Notice



## Table of Contents

# 1. Introduction

## 1.1. The Problem

Conduit and similar cryptographic browser-automation verifiers produce SHA-256 hash-chained audit trails of agent task delivery. The proof bundles they produce are self-verifiable: any third party can replay the chain and confirm the recorded events occurred in the recorded order with the recorded signatures.

This is necessary but not sufficient.

A proof bundle attests that an agent took actions X, Y, and Z. It does not attest that X, Y, and Z satisfied the principal's Intent Mandate. Empirical data from Kleros (~40% of decentralized arbitration cases turn on spec ambiguity, not on whether action occurred) and from Upwork, Stripe Connect, and eBay (auto-resolve ceilings of 60-95% with all systems hitting walls below 95%) demonstrates that cryptographic proof of execution does not eliminate disputes — it relocates them to spec interpretation.

ADRP addresses the relocated dispute surface without breaking the cryptographic substrate.

## 1.2. What ADRP Defines

A five-layer protocol stacked on the existing Conduit + AP2 + ATXN stack:

Layer 5: Precedent corpus (signed RulingBundles indexed by Cart Mandate template hash)

Layer 4: Tier router (L1 atomic / L2 mandated / L3 fiduciary)

Layer 3: Resolution engine (cryptographic-class auto, semantic-class arbitration)

Layer 2: Counter-attestation primitive (append-only override of Conduit ProofBundle)

Layer 1: Arbitration Mandate (4th AP2 mandate, FAA Section 2 written agreement)

### 1.3. What ADRP Deliberately Does NOT Do

* ADRP does not modify the Conduit ProofBundle.
* ADRP does not grant agents independent dispute standing.
* ADRP does not require statutory change.
* ADRP does not specify how acceptance criteria are authored.
* ADRP does not promise greater than 90% auto-resolution.

### 1.4. Design Tenets

Bifurcation: Cryptographic and semantic disputes require different paths.

Append-only override: The hash chain is never mutated.

Principal-only standing: Agents flag; only principals file.

Time-windowed default-resolution: Silence equals approval.

Economic friction on filing: Non-refundable filing fee deters spam.

No tokenized stake: Filing fees are USD-denominated, non-refundable to the filer, refundable to the prevailing party.

Pre-committed arbiter: The arbitration pool is named in the Arbitration Mandate at agent-deployment time.

Curated pool for v0.1: Decentralized token-staked pools deferred to v0.2.

US-only B2B for v0.1: Cross-border, consumer, and PSD2/Reg E scope deferred.

## 2. Terminology

The key words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174].

Conduit ProofBundle (H_c): A SHA-256 hash-chained, Ed25519-signed audit trail produced by a cryptographic verifier of agent task delivery. The tip hash of this chain is referenced as H_c throughout this document.

RulingBundle (R): A signed attestation by an authorized arbitrator that supersedes an underlying Conduit ProofBundle via signing-time precedence.

DisputeBundle: The append-only chain of dispute events (filing, evidence, arbitrator assignment) anchored to a Conduit ProofBundle and consumed by verify_resolution.

DisputeFlag: An advisory, agent-emitted notification that an anomaly has been detected. Non-binding. Expires if not ratified by the principal within the tier-specific window.

DisputeFiling: A binding, principal-emitted dispute filing. Posts a filing fee. Anchors a DisputeBundle.

EscrowDirective: The output of verify_resolution. Consumed by the AP2 Payment Mandate executor (or VCAP escrow rail) to release, refund, or split funds.

Arbitration Mandate: The fourth element of the AP2 mandate chain, pre-signed by the principal at agent-deployment time. Defines the arbitration pool, governing-law clause, fee schedule, and L1/L2/L3 thresholds. Hash anchored in Standing Token.

Cart Mandate Acceptance Criteria: A machine-readable structure embedded in the Cart Mandate that defines the deterministic checks Conduit attests against at delivery time.

Cryptographic-class dispute: A dispute whose resolution is computable deterministically from (DisputeBundle, RulingBundle, Standing Token chain).

Semantic-class dispute: A dispute whose resolution requires arbitration against the Cart Mandate’s acceptance criteria.

Curated Arbitrator Pool: A SwarmSync-published list of vetted human-or-model arbitrators registered as DIDs in TRUSTED_REGISTRIES. v0.1 mechanism.

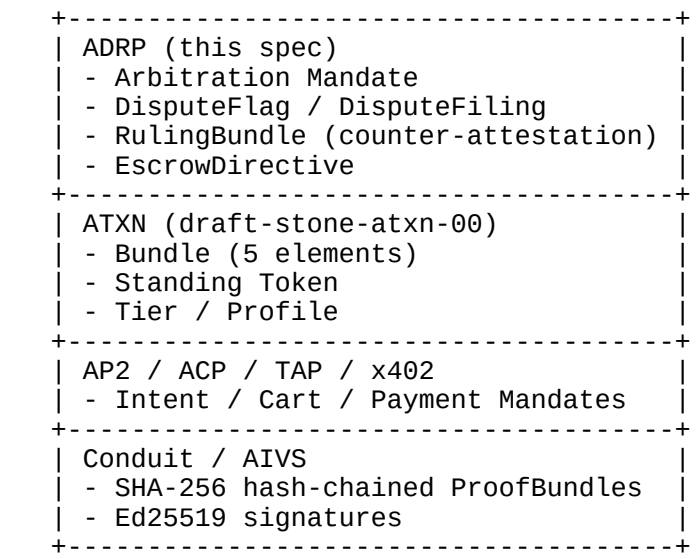
Filing Fee: A non-refundable, non-transferable, USD-denominated fee posted at DisputeFiling. Refundable only to the prevailing party.

Counter-attestation Override Pattern: The verification-time precedence rule by which a RulingBundle supersedes a Conduit ProofBundle. Both are preserved forever in the hash chain.

Terms imported from ATXN (draft-stone-atxn-00): Bundle (the five-element ATXN primitive, ATXN Section 3), Standing Token (ATXN Section 4), Tier (L1/L2/L3, ATXN Section 5), Principal, Agent, Operator, Mandate Framework, Commitment Event, Revocation Beacon (all ATXN Section 2).

## 3. Architecture Overview

ADRP sits on top of the following existing layers:




A dispute is the transition of an ATXN Bundle from delivered to disputed (per ATXN Section 8). ADRP defines what happens after that transition.

## 4. The Arbitration Mandate

### 4.1. Purpose

The Arbitration Mandate is the FAA Section 2 / NY Convention written, consensual arbitration agreement that makes ADRP enforceable without statutory change.

### 4.2. Required Fields

| Field | Type | Description |
|---|---|---|
| arb_mandate_id | UUID | Unique identifier |
| principal_did | DID | Principal’s DID |
| arbitrator_pool_ref | URI | Reference to the curated arbitrator pool (v0.1) |
| governing_law | String | E.g., "FAA + Delaware seat" |
| fee_schedule_ref | URI | Reference to filing fee schedule |
| tier_thresholds | Object | L1/L2/L3 transaction-value boundaries |
| appeal_panel_size | Integer | Default 5 |
| language_of_proceedings | ISO 639-1 | Default "en" |
| principal_signature | Ed25519 base64 | Principal’s signature over canonical JSON |

Table 1

### 4.3. Anchoring

The Arbitration Mandate’s SHA-256 hash MUST be referenced in the principal’s Standing Token (per ATXN Section 4) under a new field arb_mandate_hash. A Standing Token without an anchored Arbitration Mandate hash MUST NOT participate in an L2 or L3 ATXN Bundle.

### 4.4. Mutability

The Arbitration Mandate MAY be replaced by the principal at any time, but the replacement MUST NOT apply retroactively to in-flight Bundles.

## 5. Standing and the Flag/File Distinction

### 5.1. Parties

The parties to any ADRP dispute are always principal-A and principal-B. Never agents. This preserves the A2A executor-not-party model from UETA Section 14 and Restatement (Third) of Agency.

### 5.2. Agent Flag Rights

Agents MAY emit DisputeFlags — advisory, low-cost (1 KB JSON max), signed by the agent's Standing Token. A DisputeFlag is non-binding and does not initiate the dispute state machine.

### 5.3. Principal File Rights

Only principals MAY emit DisputeFilings — binding, fee-posting, claim-coded events that initiate the dispute state machine.

### 5.4. Flag Expiration

| Tier | Ratification Window |
|---|---|
| L1 | N/A (no dispute) |
| L2 | 72h |
| L3 | 14d |

Table 2

After expiration, the flag is preserved in the audit log but cannot anchor a DisputeFiling.

### 5.5. Pre-Authorized Dispute Delegate

A principal MAY pre-authorize a dispute delegate via a delegation field in the Arbitration Mandate. The delegate's filings are treated as principal-filed for standing purposes.

## 6. Dispute Taxonomy

### 6.1. Cryptographic-Class Disputes

Resolved by code from (DisputeBundle, RulingBundle, Standing Token chain) without arbitration.

| Code | Description |
|---|---|
| bundle_integrity | SHA-256 chain breaks; signature verification fails |
| mandate_scope | Agent acted outside Cart Mandate scope (deterministic predicate) |
| token_authority | Standing Token revoked or expired pre-execution |
| timestamp_skew | Attestation timestamps violate ordering invariants |
| oracle_contradiction | Third-party oracle data contradicts attestation |

Table 3

A cryptographic-class dispute MUST NOT be routed to arbitration.

## 6.2. Semantic-Class Disputes

Resolved by arbitration against the Cart Mandate’s acceptance criteria.

| Code | Description |
|---|---|
| quality_mismatch | Deliverable doesn’t satisfy acceptance_criteria.checks |
| spec_ambiguity | Acceptance criteria absent, under-specified, or contradictory |
| timing_breach | SLA missed |
| fitness_for_purpose | Deliverable formally compliant but unfit for principal’s purpose |

Table 4

### 6.3. Classification

The DisputeFiling's claim_code field declares the class. ADRP verifiers MUST validate that the declared class matches the evidence.

### 6.4. ATXN dispute_class to ADRP Claim Code Mapping

ATXN (Section 10.3) defines a dispute_class field set at Bundle handoff. The normative mapping is:

| ATXN dispute_class | ADRP Claim Code(s) | ADRP Class |
|---|---|---|
| fact_dispute | bundle_integrity, timestamp_skew, oracle_contradiction | Cryptographic |
| terms_dispute | mandate_scope, quality_mismatch, spec_ambiguity, timing_breach, fitness_for_purpose | Semantic |
| capacity_dispute | out of scope — see Section 6.5 | — |
| framework_dispute | out of scope — see Section 6.5 | — |

Table 5

When a DisputeFiling arrives with an ATXN dispute_class, the ADRP verifier MUST map it to the corresponding claim_code bucket using this table before routing.

### 6.5. Out of Scope (v0.1)

| Class | Reason | Disposition |
|---|---|---|
| capacity_dispute | Principal-level adjudication; agent action set aside entirely | Bypass ADRP entirely; executing platform MUST freeze settlement and refund-to-buyer; route to regulatory or Principal-level forum outside ADRP |

| | | |
|---|---|---|
| framework_dispute | Mandate framework validity contested | Bypass ADRP entirely; executing platform MUST freeze settlement; ADRP MUST NOT process until upstream framework forum resolves validity |
| Cross-border consumer disputes | Reg E / PSD2 / GDPR overlap not yet resolved | Defer to v0.2 |

Table 6

## 7. The Counter-Attestation Override Pattern

### 7.1. Invariant

The Conduit ProofBundle's tip hash H_c is never modified. No byte of the original ProofBundle is rewritten. Override is a verification-time precedence rule, not a mutation.

### 7.2. RulingBundle Structure

```
R = {
  type: "RulingBundle",
  supersedes: H_c,
  dispute_chain_tip: H_d,
  verdict: "release" | "refund" | "partial",
  partial_split: { to_buyer: 0.30, to_seller: 0.70 },
  rationale_hash: H_rationale,
  arbitrator_did: "did:web:...",
  arbitrator_vc_hash: H_vc,
  signing_time: "<RFC3339>",
  prev_hash: H_d
}
sig = Ed25519(arbitrator_priv, JCS(R))
```

### 7.3. Verification Rule

The latest valid RulingBundle (by signing_time) whose supersedes equals the underlying ProofBundle's tip hash AND whose signing arbitrator was authorized at signing_time, with chain integrity verified, is the verification winner.

### 7.4. Why This Works

Hash-chain immutability is preserved. Verification is offline and deterministic. Arbitrator authority is anchored to signing_time. The chain becomes a precedent corpus indexed by Cart Mandate template hash.

## 8. Wire Protocol

### 8.1. Message Types

| Number | Type | Emitter | Purpose |
|---|---|---|---|
| 1 | DisputeFlag | Agent | Advisory anomaly notification (non-binding) |
| 2 | DisputeFiling | Principal | Binding dispute initiation; posts filing fee |
| 3 | EvidenceSubmission | Either party | Hash-chained artifact append |
| 4 | ArbitratorAssignment | Registry | Binds arbitrator DID to dispute |
| 5 | RulingBundle | Arbitrator | Signed verdict + escrow directive |
| 6 | EscrowDirective | Verifier (derived) | Consumed by AP2 Payment Mandate or VCAP rail |

Table 7

### 8.2. Common Message Envelope

```
{
  "msg_type": "<one of the six>",
  "msg_id": "<UUID>",
  "prev_hash": "<SHA-256 of prior chain event>",
  "submitter_did": "<DID>",
  "submitter_signature": "<Ed25519 base64 over JCS of preceding fields>",
  "timestamp": "<RFC3339>",
  "payload": { ... type-specific ... }
}
```

### 8.3. Canonicalization

All signatures MUST be over the JCS ([RFC8785]) canonicalization of the signed object minus the signature field itself.

## 9. State Machine

### 9.1. States

```
 Pre-FILED:   FLAGGED         (agent-emitted; expires if not ratified)
              FILED           (principal-emitted; fee posted)
                |
                v
              ASSIGNED        (arbitrator bound to dispute)
                |
                v
              EVIDENCE_OPEN   (parties submit evidence)
                |
                v
              UNDER_REVIEW    (arbitrator deliberating)
                |
                v
              RULED           (RulingBundle signed)
                |
                v
              SETTLED         (EscrowDirective consumed by payment rail)

 Terminal:    WITHDRAWN       (filer withdraws; fee partial-refunded)
              EXPIRED         (filing or evidence window expired)
```

### 9.2. ATXN State Machine Join Points

The ADRP internal state machine is entered from ATXN’s disputed state. The normative join points are:

* ATXN disputed → ADRP FLAGGED: agent-emitted DisputeFlag (advisory; principal must ratify within tier window to proceed)

* ATXN disputed → ADRP FILED: principal-emitted DisputeFiling (skips FLAGGED)

* ADRP RULED + ADRP SETTLED → ATXN adjudicated then finalized: EscrowDirective consumed by payment rail

* ADRP WITHDRAWN or EXPIRED → ATXN finalized: dispute resolved without ruling; escrow released per Profile default

### 9.3. Invariants

| ID | Invariant |
|---|---|
| I1 | PaymentMandate.escrow_state == HOLD until SETTLED, WITHDRAWN, or EXPIRED |
| I2 | No state mutates a prior attestation; transitions are new signed events |
| I3 | prev_hash of every event = SHA-256 of canonical-JSON of the immediately prior event |
| I4 | dispute_chain[0].prev_hash == ProofBundle.tip_hash |
| I5 | Only an arbitrator authorized by the Arbitration Mandate’s pool may emit RulingBundle |
| I6 | EscrowDirective MUST be derivable from (DisputeBundle, RulingBundle) by verify_resolution |

Table 8

## 10. Tier Parameters

| Parameter | L1 | L2 | L3 |
|---|---|---|---|
| Dispute window | 0 (atomic) | 72h | 14d |
| Filing fee | N/A | 10% of value, $1-$100 | 10% of value, $50-$1,000 |
| Arbitrator SLA | N/A | 4h | 24h |
| Arbitrator | N/A | Curated AI | Curated human |

| | | | |
|---|---|---|---|
| pool | | panel (3 of 5) | pool (3 of 5) + optional legal review |
| Override threshold | N/A | 4-of-5 supermajority | 4-of-5 supermajority |
| Max txn auto-arb | unlimited | $1,000 | unlimited |
| Default if SLA missed | N/A | refund-to-buyer | refund-to-buyer |
| Frivolous slash | N/A | 100% fee forfeited | 100% fee forfeited |
| Novel-but-lost slash | N/A | 50% fee forfeited | 50% fee forfeited |
| Appeal window | N/A | 12h | 48h |
| Appeal panel | N/A | 5 (up from 3) | 5 + legal review |
| Appeal quorum | N/A | 4 of 5 | 4 of 5 |
| Escrow hold | 0 | 7 days | 21 days |
| Evidence file size cap | N/A | 10 MB total, max 20 files | 10 MB total, max 20 files |

Table 9

## 10.1. Tier Routing

Tier is determined by the ATXN Bundle's Tier (per ATXN Section 5) AND by the transaction value relative to the max_txn_auto_arb ceiling. A Bundle declared L2 with transaction value greater than $1,000 MUST auto-escalate to L3 dispute processing. This is a dispute-processing escalation rule only and does not retroactively change the Bundle's declared formation Tier (see ATXN Section 5.2).

## 10.2. Default Ruling on SLA Miss

If the arbitrator pool fails to produce a RulingBundle within the SLA, the dispute MUST default to refund-to-buyer.

## 11. Cart Mandate Acceptance Criteria

### 11.1. Required Field

The Cart Mandate MUST contain:

```
{
  "acceptance_criteria": {
    "schema_version": "1",
    "checks": [
      { "type": "regex", "selector": "$.output", "pattern": "..." },
      { "type": "json_schema", "selector": "$.deliverable", "schema": { ... } },
      { "type": "deterministic_function", "function_hash": "sha256:..." },
      { "type": "human_review_required", "reason": "..." }
    ],
    "default_window_hours": 72,
    "silence_equals": "release"
  }
}
```

### 11.2. Check Types

| Type | Evaluation |
|---|---|
| regex | Deterministic regex match against JSON-path selector |
| json_schema | JSON Schema validation against selector |
| deterministic_function | Pure function (referenced by hash, fetched from registry) executed against deliverable |
| human_review_required | Auto-routes to arbitration; flags spec as deliberately not auto-evaluable |

Table 10

### 11.3. Conduit Attestation Against Checks

Conduit attests against acceptance_criteria.checks at delivery time. If all checks pass deterministically, the dispute path auto-resolves cryptographically. If checks include human_review_required OR are missing, dispute auto-routes to L2 semantic arbitration.

### 11.4. The Authoring Helper (Normative for Deployment)

A natural-language-to-checks helper that converts a principal's English-language description into machine-checkable predicates is REQUIRED for v0.1 deployment.

## 12. Arbitrator Pool

### 12.1. v0.1: Curated Pool

v0.1 ships with a SwarmSync-curated arbitrator pool — a centralized list of vetted human-or-model arbitrators registered as DIDs in TRUSTED_REGISTRIES.

### 12.2. Selection Algorithm

Arbitrator selection MUST use a verifiable random function (VRF) seeded by the DisputeFiling's hash.

### 12.3. Pool Composition

| Tier | Pool Composition |
|---|---|
| L2 | AI arbitrator panels (model + curated checker) |
| L3 | Human arbitrators (vetted, jurisdictionally-licensed where relevant) + optional legal review |

Table 11

### 12.4. Arbitrator Credentialing

Every arbitrator MUST hold a Verifiable Credential that names the arbitrator's DID, states valid_from and valid_until timestamps, names qualifications, and is signed by a registry in TRUSTED_REGISTRIES.

### 12.5. v0.2 Roadmap

Decentralized stake-weighted arbitrator pools are deferred to v0.2.

## 13. Filing Fee Economics

### 13.1. Fee Structure

| Parameter | Value |
|---|---|
| Fee rate | 10% of transaction value |
| Floor | $1 (L2) / $50 (L3) |
| Ceiling | $100 (L2) / $1,000 (L3) |
| Denomination | USD (fiat or USDC stablecoin) |
| Refund | To prevailing party only |

Table 12

### 13.2. Why Not a Tokenized Stake

The fee is non-refundable to the filer, non-transferable, and denominated in USD. It yields no return. It is not a security under Howey.

### 13.3. Slashing

| Outcome | Filer’s Fee |
|---|---|
| Filer prevails | 100% refunded |
| Filer loses; novel evidence | 50% forfeited |
| Filer loses; frivolous | 100% forfeited |
| Filer withdraws before EVIDENCE_OPEN | 75% refunded |
| Filer withdraws after EVIDENCE_OPEN | 25% refunded |

Table 13

Forfeited fees MAY fund the arbitrator pool and a public dispute-quality dataset.

### 13.4. Arbitrator Compensation

Arbitrators are compensated 2% of transaction value (min $0.50) from the losing party’s escrow.

## 14. Appeals

### 14.1. Trigger

Either party MAY file an appeal within the tier-specific appeal window (12h L2, 48h L3). Appeals require a fresh filing fee at 1.5x the original rate.

### 14.2. Appeal Panel

A 5-arbitrator panel selected via VRF from a different pool slice than the original ruling. Quorum: 4 of 5.

### 14.3. Finality

Appeal rulings are FINAL. There is no second appeal.

### 14.4. Appeal Override

An appeal RulingBundle supersedes the original RulingBundle by the same signing-time precedence rule (Section 7.3). Both are preserved forever.

## 15. Precedent Corpus

### 15.1. Indexing

Every signed RulingBundle is content-addressed and indexed by: Cart Mandate template hash; dispute claim_code; verdict; arbitrator DID; signing time.

### 15.2. Citation Requirement

Future arbitrators MUST query the precedent corpus for prior rulings on similar Cart Mandate templates and either cite or distinguish them.

### 15.3. v0.1 vs v0.2

v0.1 ships the data layer. v0.2 ships the search and citation infrastructure.

# 16. Verification Algorithm

## 16.1. The verify_resolution Function

```
   def verify_resolution(db: DisputeBundle, rb: RulingBundle) -> Result:
       # 1. Anchor checks
       assert db.conduit_proof_ref == rb.supersedes
       assert rb.prev_hash == db.chain_tip

       # 2. Chain integrity
       h = db.conduit_proof_ref
       for ev in [db.filing, *db.evidence_chain, db.arbitrator_assignment]:
           assert ev.prev_hash == h
           assert verify_sig(ev.submitter_did, ev.sig, jcs(ev))
           h = sha256(jcs(ev))
       assert h == db.chain_tip

       # 3. Arbitrator authority at signing_time (not now)
       vc = fetch_vc(rb.arbitrator_vc_hash)
       assert vc.subject == rb.arbitrator_did
       assert vc.issuer in TRUSTED_REGISTRIES
       assert vc.valid_from <= rb.signing_time <= vc.valid_until
       assert verify_sig(vc.issuer, vc.sig, jcs(vc))

       # 4. Ruling signature
       assert verify_sig(rb.arbitrator_did, rb.sig, jcs(rb_minus_sig))

       # 5. Verdict well-formed
       assert rb.verdict in {"release", "refund", "partial"}
       if rb.verdict == "partial":
           assert sum(rb.partial_split.values()) == 1.0

       # 6. Derive directive
       return Result(valid=True, escrow_directive=EscrowDirective(
           payment_mandate_ref=db.payment_mandate_ref,
           action=rb.verdict,
           split=rb.partial_split,
           ruling_ref=sha256(jcs(rb))
       ))
```

## 16.2. Determinism Requirement

Two honest verifiers given the same inputs MUST reach the same EscrowDirective.

### 16.3. Offline Requirement

verify_resolution MUST NOT require network access except to fetch the arbitrator's Verifiable Credential by hash.

## 17. SDK Surface

A conformant ADRP v0.1 SDK MUST expose at minimum six methods, implementable in 500 LOC total:

```
emit_flag(agent_standing_token, claim_code, evidence_ref) -> DisputeFlag
file_dispute(payment_mandate, claim_code, claim_detail, flag_ref) -> DisputeBundle
submit_evidence(dispute_id, artifact_bytes, mime_type) -> EvidenceLink
query_status(dispute_id) -> {state, chain_tip, latest_event}
render_ruling(dispute_id, verdict, partial_split, rationale) -> RulingBundle  # arbitr
ator-only
verify_resolution(dispute_bundle, ruling_bundle) -> {valid, escrow_directive}
```

### 17.1. Appeal as Re-Entry

An appeal is file_dispute with the prior_ruling_ref field set.

### 17.2. LOC Budget

emit_flag: ~30 LOC

file_dispute: ~80 LOC

submit_evidence: ~50 LOC

query_status: ~30 LOC

render_ruling: ~80 LOC

verify_resolution: ~120 LOC

Common envelope, signing, hash chain: ~110 LOC

Total: ~500 LOC

### 17.3. Golden Test Cases

| Number | Case | Expected Behavior |
|---|---|---|
| 1 | clear_release | No flag, no filing, window expires → release |
| 2 | clear_refund | DisputeFiling with valid |

| | | | |
|---|---|---|---|
| | | | bundle_integrity claim → refund |
| 3 | partial_split | Semantic dispute, partial_split: {to_buyer: 0.30, to_seller: 0.70} | |
| 4 | attestation_override | Original ProofBundle preserved; latest valid RulingBundle wins | |
| 5 | frivolous_dispute | Filer loses without novel evidence → 100% fee forfeited | |
| 6 | malicious_requester_reject | Principal-A files invalid claim → fee forfeited, escrow releases to seller | |
| 7 | expired_dispute | DisputeFiling outside tier window → EXPIRED | |
| 8 | flag_ratification | DisputeFlag emitted, principal files within window → promoted | |
| 9 | flag_expiration | DisputeFlag emitted, no action within window → preserved but cannot anchor filing | |
| 10 | appeal_supersedes | Appeal RulingBundle supersedes original by signing-time precedence | |

Table 14

## 18. Scope Locks for v0.1

| Lock | Reason |
|---|---|
| US-only B2B | No consumer / cross-border / Reg E / PSD2 / GDPR overlap |
| Partnered custody | No SwarmSync-held escrow; relies on Stripe Connect, Bridge, Modern Treasury, or equivalent |

| | |
|---|---|
| Curated arbitrator pool | No token-staked decentralized pool (Howey) |
| Filing fee, not stake | Non-refundable, non-transferable, USD-denominated |
| Reputation logged but not modulating | Avoids bootstrap-circular trust dependency in v0.1 |
| Per-principal preference model deferred | Data collection pipeline ships in v0.1; productized search ships in v0.2 |

Table 15

## 19. Security Considerations

### 19.1. The >90% Auto-Resolution Target Is NOT a Protocol Property

The greater than 90% auto-resolution target is a deployment-conditional target, not a protocol property. Empirical base rates (Kleros, eBay, Upwork, Mechanical Turk) cap auto-resolution at 60-95%. Implementations MUST: ship a normative NLP-to-checks authoring helper; run a 90-day shadow-mode pilot before GA; measure check-coverage and auto-resolution rate as primary metrics; pivot to mandatory human_review_required flag if check coverage is less than 30% at day 30.

### 19.2. The Bifurcation Seam

Adversaries will probe the seam between cryptographic-class and semantic-class disputes. The classification validation in Section 6.3 is REQUIRED.

### 19.3. Arbitrator Collusion

A 4-of-5 threshold is compromisable via collusion of 4 arbitrators. Mitigations: diversify the curated pool across non-correlated risk profiles; monitor for arbitrator-pair correlation in rulings; periodically audit a random sample of rulings.

### 19.4. Filing Fee as DoS Vector

Mitigations: per-principal rate limit (max 0.5% of rolling 30-day transaction count); frivolous slash; account suspension after N frivolous filings.

### 19.5. Standing Token Replay

A revoked Standing Token MUST NOT anchor a new DisputeFiling.

### 19.6. Time-Skew Attacks

Mitigations: threshold-signature timestamps for L2 and L3 rulings (3-of-5 from a federated timestamping authority); reject rulings with signing_time more than 5 minutes in the future.

### 19.7. Jurisdictional Enforceability of the Arbitration Mandate

Implementations SHOULD seek a written legal opinion in the deployment jurisdiction before relying on the Arbitration Mandate to defeat a court filing.

### 19.8. Sanctions Screening

Every party receiving funds via an EscrowDirective MUST be screened against OFAC and equivalent sanctions lists.

### 19.9. Principal Capacity Mid-Dispute

If a principal’s capacity attestation expires or is revoked mid-dispute, the dispute MUST be halted and routed to a capacity_dispute, defaulting to refund-to-buyer.

### 19.10. Acknowledged Residual Risks

* Hallucinated mandates: partially mitigated by L3 epistemic attestation; unsolved for L1/L2.
* Adversarial sub-agency chains: bounded by sub_delegation_depth per ATXN.
* Arbitrator pool capture: mitigated by VRF selection; v0.2 decentralization is the long-term mitigation.

## 20. IANA Considerations

### 20.1. ADRP JSON-LD Context

URI: https://swarmsync.ai/spec/adrp/v1

### 20.2. ADRP Claim Code Registry

A new IANA registry "ADRP Claim Codes" is requested. Registration policy: Specification Required (per [RFC8126]).

Initial entries:

| Code | Class | Description |
|---|---|---|
| bundle_integrity | cryptographic | SHA-256 chain breaks; signature verification fails |
| mandate_scope | cryptographic | Agent acted outside Cart Mandate scope |
| token_authority | cryptographic | Standing Token revoked or expired pre-execution |
| timestamp_skew | cryptographic | Attestation timestamps violate ordering invariants |
| oracle_contradiction | cryptographic | Third-party oracle data contradicts attestation |
| quality_mismatch | semantic | Deliverable doesn’t satisfy acceptance_criteria.checks |
| spec_ambiguity | semantic | Acceptance criteria absent or under-specified |
| timing_breach | semantic | SLA missed |
| fitness_for_purpose | semantic | Deliverable formally compliant but unfit |

Table 16

### 20.3. ADRP Verdict Enum

A new IANA registry "ADRP Verdicts" is requested.

| Value | Description |
|---|---|
| release | Funds released to seller (counterparty) |
| refund | Funds returned to buyer (filer) |
| partial | Funds split per partial_split (must sum to 1.0) |

Table 17

### 20.4. ADRP Trusted Registries List

A new IANA registry "ADRP Trusted Arbitrator Registries" is requested.

## 21. Acknowledgements

This specification synthesizes the output of the Ultimate Brainstorm v2.2 Agent Dispute Resolution Protocol session (2026-04-25). The author thanks: EpistemicAuditor (cryptographic/semantic bifurcation); Archaeologist (empirical base-rate floor); Quantifier (tier parameters); ConstraintCartographer (Arbitration Mandate as FAA anchor; v0.1 scope locks); socratic-mentor (principal-only standing); DarkMirror (flag/file distinction); IdeaMatrix (tiered hybrid architecture); RemixForge (Cart Mandate acceptance_criteria); SoSpec (wire protocol, state machine, SDK surface, verify_resolution algorithm); SpiderSpark (three-tier bonded-escalation structure).

Three formal dissents: Archaeologist assigns less than 40% probability to the greater than 90% auto-resolution target. EpistemicAuditor assigns 25% probability to median Cart Mandates having sufficient acceptance_criteria coverage. RemixForge considers deferring per-principal preference model to v0.2 a 70%-probability strategic mistake.

## 22. Normative References

[draft-stone-aivs-01]
stone, B., "AIVS: Agentic Integrity Verification Standard", Work in Progress, Internet-Draft, draft-stone-aivs-01, 2026, <https://datatracker.ietf.org/doc/html/draft-stone-aivs-01>.

[draft-stone-atep-01]
stone, B., "ATEP: Agent Trust and Execution Passport", Work in Progress, Internet-Draft, draft-stone-atep-01, 2026, <https://datatracker.ietf.org/doc/html/draft-stone-atep-01>.

[draft-stone-atxn-00]
stone, B., "ATXN: Agent-to-Agent Transaction Definition Protocol", Work in Progress, Internet-Draft, draft-stone-atxn-00, 2026, <https://datatracker.ietf.org/doc/html/draft-stone-atxn-00>.

[draft-stone-vcap-01]
stone, B., "VCAP: Verified Commerce for Agent Protocols", Work in Progress, Internet-Draft, draft-stone-vcap-01, 2026, <https://datatracker.ietf.org/doc/html/draft-stone-vcap-01>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <https://www.rfc-editor.org/info/rfc3339>.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <https://www.rfc-editor.org/info/rfc8032>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8785] Rundgren, A., Jordan, B., and S. Erdtman, "JSON Canonicalization Scheme (JCS)", RFC 8785, DOI 10.17487/RFC8785, June 2020, <https://www.rfc-editor.org/info/rfc8785>.

[W3C-DID] W3C, "Decentralized Identifiers (DIDs) v1.0", 2022, <https://www.w3.org/TR/did-core/>.

[W3C-VC-2.0]
W3C, "Verifiable Credentials Data Model 2.0", 2024, <https://www.w3.org/TR/vc-data-model-2.0/>.

## 23. Informative References

[AP2] Google, "Agent Payments Protocol", 2025, <https://developers.google.com/wallet/agent-payments>.

[FAA] United States Congress, "Federal Arbitration Act, 9 U.S.C. Sections 1-16", 1925.

[HOWEY] Supreme Court of the United States, "SEC v. W. J. Howey Co., 328 U.S. 293", 1946.

[KLEROS] Kleros, "Decentralized Court Protocol", 2018, <https://kleros.io>.

[NY-CONVENTION]
United Nations, "Convention on the Recognition and Enforcement of Foreign Arbitral Awards", 1958.

[OFAC-SANCTIONS]
U.S. Treasury OFAC, "Specially Designated Nationals List", 2024, <https://ofac.treasury.gov/specially-designated-nationals-and-blocked-persons-list-sdn-human-readable-lists>.

[REST-AGENCY-3D]
American Law Institute, "Restatement (Third) of Agency", 2006.

[STRIPE-CONNECT]
Stripe, "Connect Disputes and Chargebacks", 2024, <https://stripe.com/docs/connect/disputes>.

[UETA-14] Uniform Law Commission, "Uniform Electronic Transactions Act Section 14", 1999.

[UMA] UMA Protocol, "Optimistic Oracle", 2020, <https://uma.xyz>.

[UPWORK-MEDIATION]
Upwork, "Dispute Process", 2024, <https://support.upwork.com/hc/en-us/articles/211062568>.

[VISA-CHARGEBACK-CODES]
Visa Inc., "Visa Chargeback Reason Codes", 2024,
<https://usa.visa.com/dam/VCOM/download/about-visa/visa-rules-public.pdf>.

## Appendix A: JSON Schemas (Informative)

### A.1 DisputeFiling Schema

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://swarmsync.ai/spec/adrp/v1/dispute-filing.schema.json",
  "title": "DisputeFiling",
  "type": "object",
  "required": [
    "msg_type", "msg_id", "prev_hash", "submitter_did",
    "submitter_signature", "timestamp", "payload"
  ],
  "properties": {
    "msg_type": { "const": "DisputeFiling" },
    "msg_id": { "type": "string", "format": "uuid" },
    "prev_hash": { "type": "string", "pattern": "^[0-9a-f]{64}$" },
    "submitter_did": { "type": "string" },
    "submitter_signature": { "type": "string" },
    "timestamp": { "type": "string", "format": "date-time" },
    "payload": {
      "type": "object",
      "required": [
        "payment_mandate_ref", "claim_code", "claim_detail", "filing_fee"
      ],
      "properties": {
        "payment_mandate_ref": { "type": "string" },
        "claim_code": { "type": "string" },
        "claim_detail": { "type": "string", "maxLength": 4096 },
        "flag_ref": { "type": "string" },
        "prior_ruling_ref": { "type": "string" },
        "filing_fee": {
          "type": "object",
          "required": ["amount", "currency", "transaction_id"],
          "properties": {
            "amount": { "type": "number", "minimum": 1, "maximum": 1000 },
            "currency": { "enum": ["USD", "USDC"] },
            "transaction_id": { "type": "string" }
          }
        }
      }
    }
  }
}
```

A.2 RulingBundle Schema

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://swarmsync.ai/spec/adrp/v1/ruling-bundle.schema.json",
  "title": "RulingBundle",
  "type": "object",
  "required": [
    "type", "supersedes", "dispute_chain_tip", "verdict",
    "rationale_hash", "arbitrator_did", "arbitrator_vc_hash",
    "signing_time", "prev_hash", "sig"
  ],
  "properties": {
    "type": { "const": "RulingBundle" },
    "supersedes": { "type": "string", "pattern": "^[0-9a-f]{64}$" },
    "dispute_chain_tip": { "type": "string", "pattern": "^[0-9a-f]{64}$" },
    "verdict": { "enum": ["release", "refund", "partial"] },
    "partial_split": {
      "type": "object",
      "properties": {
        "to_buyer": { "type": "number", "minimum": 0, "maximum": 1 },
        "to_seller": { "type": "number", "minimum": 0, "maximum": 1 }
      }
    },
    "rationale_hash": { "type": "string", "pattern": "^[0-9a-f]{64}$" },
    "arbitrator_did": { "type": "string" },
    "arbitrator_vc_hash": { "type": "string", "pattern": "^[0-9a-f]{64}$" },
    "signing_time": { "type": "string", "format": "date-time" },
    "prev_hash": { "type": "string", "pattern": "^[0-9a-f]{64}$" },
    "sig": { "type": "string" }
  }
}
```

## Appendix B: Worked Example (Informative)

### B.1 Scenario

A small business principal (did:web:smb.example) dispatches an agent to book a $250 hotel reservation. The agent books in Portland, ME instead of Portland, OR due to an ambiguous Cart Mandate.

### B.2 Sequence

1. Agent emits DisputeFlag: claim_code quality_mismatch.

2. Principal reviews flag within 72h, agrees booking is wrong.

3. Principal emits DisputeFiling: quality_mismatch, filing_fee $25.00 USDC.

4. Arbitrator assigned via VRF from L2 curated AI pool.

5. Principal submits evidence: original Cart Mandate showing "Portland" without state qualifier.

6. Counterparty submits evidence: acceptance_criteria.checks is empty.

7. Arbitrator finds spec_ambiguity. Verdict: partial, split 70/30 to buyer/seller.

8. RulingBundle signed within 4h SLA.

9. EscrowDirective: $175 to buyer, $75 to seller.

10. AP2 Payment Mandate executor consumes EscrowDirective, executes split.

11. Filing fee: 50% forfeited ($12.50), 50% refunded ($12.50).

### B.3 Verification

Any third party, given (DisputeBundle, RulingBundle), can run verify_resolution deterministically.

### B.4 Precedent

RulingBundle indexed in precedent corpus by Cart Mandate template hash.

## Appendix C: Open Issues for Shadow-Mode Validation

| Number | Question | Falsification Threshold |
|---|---|---|
| 1 | Will principals author Cart Mandates with 3+ acceptance_criteria.checks? | Less than 30% coverage at day 30 → ship NLP-to-checks helper as mandatory |
| 2 | Will the auto-resolution rate hit 60%+? | Less than 60% at day 60 → escalate to architectural review |
| 3 | Will the frivolous filing rate stay below 2%? | Greater than 2% at day 60 → tighten filing fee floor |

| | | |
|---|---|---|
| 4 | Will the curated arbitrator pool meet SLA at scale? | Greater than 5% SLA breaches at day 90 → expand pool or accelerate v0.2 |
| 5 | Will the Arbitration Mandate be challenged in court? | Any court challenge → seek immediate written legal opinion |
| 6 | Will the precedent corpus accumulate fast enough? | Less than 100 corpus queries by arbitrators at month 3 → revisit |
| 7 | Are tier thresholds calibrated correctly? | Filing rate greater than 5% or less than 0.05% → recalibrate |

Table 18

GA is gated on: auto-resolution rate greater than or equal to 60% AND frivolous filing rate less than or equal to 2% AND zero court challenges AND arbitrator SLA breaches less than or equal to 5% AND median Cart Mandate has at least 1 non-trivial acceptance_criteria.check.

## Author’s Address

Ben Stone
SwarmSync.AI
Email: benstone@swarmsync.ai