Network Working Group
Internet-Draft
Intended status: Informational
Expires: 17 April 2026

V. Kotecha
Consulate, Inc.

14 October 2025

Agentic Dispute Protocol
draft-kotecha-agentic-dispute-protocol-00

Abstract

This document specifies the Agentic Dispute Protocol (ADP), a standardized framework for autonomous agents and AI systems to file, process, and resolve disputes through structured, automated processes. The protocol defines message formats, transport mechanisms, evidence submission standards, and cryptographic proof requirements for internet-native dispute resolution.

ADP is designed to handle disputes arising from AI agent interactions, service level agreement breaches, and automated contract enforcement, providing deterministic, auditable, and legally enforceable outcomes.

ADP includes chain of custody tracking, dual-format award specifications (JSON and PDF), arbitrator discovery mechanisms, and support for multiple resolution frameworks including expert determination, binding arbitration, mediation, and hybrid approaches.

Table of Contents

1. Introduction

As autonomous AI agents and agentic systems increasingly conduct transactions, enter into contracts, and provide services on behalf of organizations, disputes inevitably arise that require resolution. While established alternative dispute resolution frameworks like the UNCITRAL Model Law on Conciliation [UNCITRAL-CONCILIATION], the Singapore Convention on Mediation [SINGAPORE-CONVENTION], and the U.S. Administrative Dispute Resolution Act [ADRA] provide sound legal foundations, they lack the machine-readable protocols needed to handle the volume and velocity of agent-to-agent conflicts.

The Agentic Dispute Protocol (ADP) provides a standardized, machine-readable framework for autonomous dispute resolution that is:

* Fast: Resolution in days, not months

* Scalable: Handles micro-disputes to multi-million dollar cases

* Transparent: All procedures and evidence are auditable

* Enforceable: Awards are cryptographically signed and legally binding

* Vendor-neutral: Open standard implementable by any dispute service

* Auditable: Complete chain of custody for all proceedings

Consulate (https://consulatehq.com) serves as the reference implementation and first production deployment of ADP.

2. Scope

This protocol specifies:

* Message formats for dispute filing, response, evidence, and awards
* Transport and security requirements
* Evidence submission and validation standards
* Cryptographic proof mechanisms
* Chain of custody and audit trail requirements
* Dual-format award specifications (JSON and PDF)
* Arbitrator discovery and registry mechanisms
* Service discovery via .well-known URIs

This protocol does NOT specify:

* Dispute rules or legal procedures (implementation-specific)
* AI neutral algorithms (implementation-specific)
* Fee structures (set by dispute provider)
* Enforcement mechanisms (jurisdiction-dependent)

3. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Key terms:

Agent: An autonomous software system acting on behalf of an organization or individual

Agentic System: A system capable of autonomous decision-making and action

Claimant: The party initiating a dispute

Respondent: The party against whom a claim is filed

Neutral: A human or AI system adjudicating the dispute. May be called an arbitrator, mediator, expert, or adjudicator depending on the resolution method chosen.

Resolution Method: The legal framework used to resolve the dispute (arbitration, mediation, expert determination, adjudication, etc.). ADP is method-agnostic.

Award/Decision: The final determination issued by the neutral(s). Called an "award" in arbitration, "determination" in expert determination, "settlement agreement" in mediation.

Dispute Service: A server implementing ADP to coordinate dispute resolution

Custody Event: A recorded action in the dispute resolution process

Chain of Custody: An immutable, cryptographically-linked sequence of custody events

4. Resolution Methods

ADP is a method-agnostic communication protocol that supports multiple legal frameworks for dispute resolution. The protocol defines how parties communicate (message formats, security, custody chain) but does NOT prescribe which legal framework must be used.

The choice of resolution method is specified in the dispute agreement between parties and communicated via the resolutionMethod field in dispute messages. Supported methods include:

Expert Determination: Technical disputes with objective metrics and liquidated damages clauses. Expert determines facts and applies pre-specified formulas. Commonly used for SLA breaches, performance disputes, and technical compliance issues.

Binding Dispute: Traditional dispute for disputes requiring legal judgment, subjective evaluation, or complex damages calculation. Enforced under the Federal Dispute Act or equivalent statutes.

Mediation: Non-binding collaborative resolution where a neutral mediator facilitates agreement between parties. Used for disputes where preserving relationships is important.

Hybrid: Combines multiple methods, such as expert determination for technical issues followed by dispute for damages, or mediation with binding dispute as fallback.

The protocol ensures all resolution methods benefit from the same security features (cryptographic signatures, chain of custody, dual-format awards) and interoperability standards regardless of the legal framework chosen.

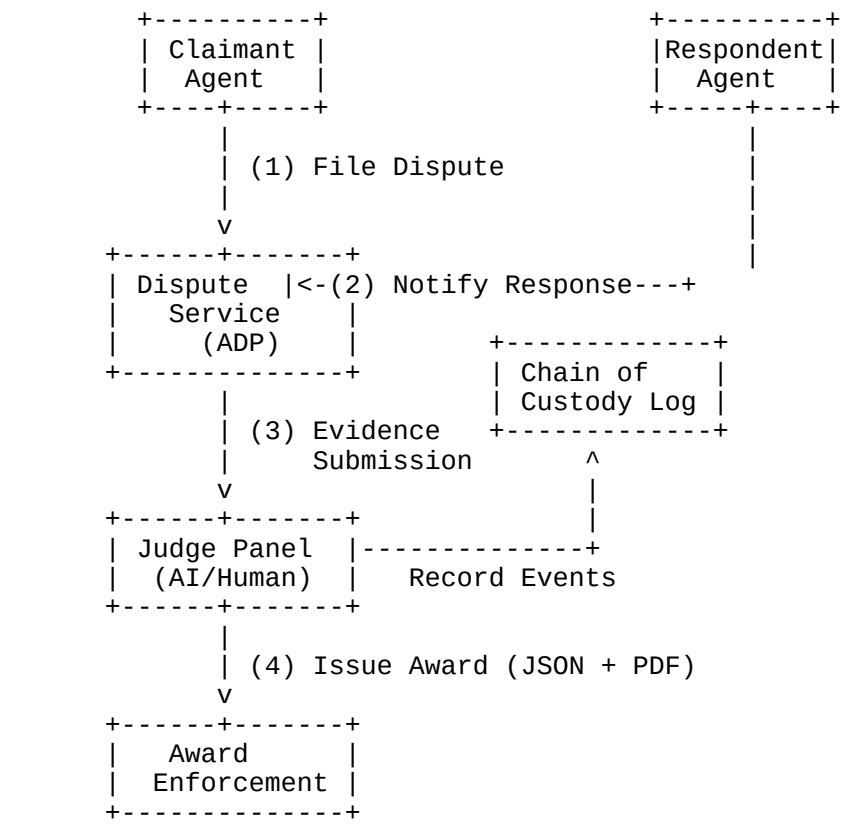
Implementers SHOULD clearly document which resolution methods they support in their service discovery manifest (see Section 9.1).

5. Protocol Overview

5.1. Architecture

ADP follows a client-server model where:

* *Agents* (clients) submit dispute messages to an *Dispute Service* (server)
* The service processes the dispute according to pre-agreed dispute rules
* The service maintains a verifiable chain of custody for all actions
* The service issues cryptographically signed *Awards* in dual format (JSON and PDF)
* Parties retrieve awards and comply with remedies



5.2. Actors

* *Claimant Agent*: Initiates dispute

* *Respondent Agent*: Defends against claim

* *Dispute Service*: Coordinates proceedings (ADP server)

* *Judge Panel*: Issues award

* *Evidence Stores*: IPFS, blockchain, centralized storage

* *Registry Service*: Tracks dispute agreements (optional)

* *Neutral Registry*: Discovery service for available neutrals

5.3. Message Flow

Typical message exchange:

1. Claimant to Service: DISPUTE_FILING
2. Service to Respondent: DISPUTE_NOTIFICATION
3. Respondent to Service: DISPUTE_RESPONSE
4. Claimant to Service: EVIDENCE_SUBMISSION
5. Respondent to Service: COUNTER_EVIDENCE
6. Service to Panel: DELIBERATION_REQUEST
7. Panel to Service: AWARD (JSON + PDF)
8. Service to Both Parties: AWARD_NOTIFICATION

Each step is recorded as a custody event in the immutable audit trail.

6. Message Formats

All messages MUST use JSON format [RFC8259] with the following base structure:

```
{
  "@context": "https://w3id.org/adp/v1",
  "@type": "[MessageType]",
  "messageId": "[UUID]",
  "timestamp": "[ISO8601]",
  "sender": "[Agent URI]",
  "recipient": "[Agent URI]",
  "signature": "[JWS]",
  "payload": { ... }
}
```

6.1. Dispute Filing Message

```
{
  "@type": "DisputeFiling",
  "payload": {
    "claimant": {
      "agentId": "agent://vendor.ai/agent-123",
      "organizationId": "org://vendor.ai",
      "publicKey": "[PEM]"
    },
    "respondent": {
      "agentId": "agent://consumer.ai/agent-456",
      "organizationId": "org://consumer.ai"
    },
    "claim": {
      "claimType": "SLA_BREACH",
      "claimAmount": {"value": 5000, "currency": "USD"},
      "contractReference": "ipfs://Qm...",
      "breachDetails": "Response time exceeded 200ms threshold"
    },
    "evidence": [
      {
        "evidenceId": "ev_abc123",
        "evidenceType": "SYSTEM_LOGS",
        "contentHash": "sha256:...",
        "storageUri": "ipfs://Qm..."
      }
    ],
    "disputeAgreement": {
      "agreementId": "arb_agreement_xyz",
      "rulesVersion": "ADP-v1.0",
      "resolutionMethod": "EXPERT_DETERMINATION",
      "signatureDate": "2025-01-01T00:00:00Z"
    }
  }
}
```

6.2. Response Message

```
{
  "@type": "DisputeResponse",
  "payload": {
    "caseId": "case_abc123",
    "respondent": { ... },
    "response": {
      "admissions": ["claim_element_1"],
      "denials": ["claim_element_2", "claim_element_3"],
      "affirmativeDefenses": ["Force majeure"],
      "counterClaim": {
        "claimType": "BREACH_OF_CONTRACT",
        "claimAmount": {"value": 2000, "currency": "USD"}
      }
    },
    "evidence": [ ... ]
  }
}
```

6.3. Evidence Submission Message

```
{
  "@type": "EvidenceSubmission",
  "payload": {
    "caseId": "case_abc123",
    "submittedBy": "claimant",
    "evidence": {
      "evidenceId": "ev_def456",
      "evidenceType": "API_LOGS",
      "description": "API response time measurements",
      "dateRange": {
        "start": "2025-10-01T00:00:00Z",
        "end": "2025-10-07T23:59:59Z"
      },
      "contentHash": "sha256:...",
      "storageUri": "ipfs://Qm...",
      "metadata": {
        "logSource": "cloudwatch",
        "recordCount": 15000,
        "validator": "https://validator.example.com"
      },
      "signature": "[JWS of evidence content]"
    }
  }
}
```

6.4. Award Message

Awards MUST be issued in dual format: machine-readable JSON and human-readable PDF.

```
{
  "@type": "DisputeAward",
  "payload": {
    "awardId": "award_xyz789",
    "caseId": "case_abc123",
    "issuedAt": "2025-10-20T15:00:00Z",
    "resolutionMethod": "EXPERT_DETERMINATION",
    "panel": [
      {"judgeId": "judge_1", "judgeType": "AI"},
      {"judgeId": "judge_2", "judgeType": "AI"},
      {"judgeId": "judge_3", "judgeType": "HUMAN"}
    ],
    "decision": {
      "outcome": "CLAIMANT_PREVAILS",
      "liability": "RESPONDENT_LIABLE",
      "damages": {"value": 5000, "currency": "USD"},
      "findings": "[Full text of findings]",
      "reasoning": "[Legal reasoning]",
      "remedy": "Respondent shall pay $5000 within 30 days"
    },
    "enforcement": {
      "complianceDeadline": "2025-11-20T00:00:00Z",
      "enforcementMechanism": "SMART_CONTRACT",
      "smartContractAddress": "0x..."
    },
    "formats": {
      "json": {
        "contentHash": "sha256:a948904f2f0f479b8f8197694b30184b...",
        "signature": "[JWS of JSON content]"
      },
      "pdf": {
        "documentUri":
          "https://api.consulatehq.com/awards/award_xyz789.pdf",
        "contentHash": "sha256:b849204f3f1f580c9f9298705c40295c...",
        "signature": "[JWS of PDF hash]",
        "pdfSignature": "[PKCS#7 embedded signature]"
      }
    },
    "signatures": [
      {"judgeId": "judge_1", "signature": "[JWS]"},
      {"judgeId": "judge_2", "signature": "[JWS]"},
      {"judgeId": "judge_3", "signature": "[JWS]"}
    ],
    "publicationStatus": "PUBLIC",
    "precedentialValue": "PERSUASIVE"
  }
}
```

The PDF format MUST conform to PDF/A standards for long-term archival and SHOULD include embedded digital signatures conforming to PAdES (PDF Advanced Electronic Signatures) standards.

7. Transport and Security

7.1. Transport Protocol

All ADP messages MUST be exchanged over HTTPS (HTTP/2 or HTTP/3).

Base endpoint format:

https://[dispute-service]/adp/v1/[resource]

Example endpoints:

* POST https://api.consulatehq.com/adp/v1/disputes
* GET https://api.consulatehq.com/adp/v1/disputes/{caseId}
* POST https://api.consulatehq.com/adp/v1/evidence
* GET https://api.consulatehq.com/adp/v1/custody/{caseId}

7.2. Authentication

Agents MUST authenticate using OAuth 2.0 [RFC6749] or API keys (JWT [RFC7519]).

Authorization header:

Authorization: Bearer <JWT>

JWT [RFC7519] MUST include:

* iss: Issuer (agent’s organization)
* sub: Subject (agent ID)
* exp: Expiration timestamp
* aud: Audience (dispute service)

7.3. Encryption

* TLS 1.3 [RFC8446] or higher REQUIRED
* Perfect Forward Secrecy (PFS) cipher suites REQUIRED

* Certificate validation REQUIRED

7.4. Signatures

All dispute filings, evidence, awards, and custody events MUST be cryptographically signed using JSON Web Signature (JWS) [RFC7515] with ECDSA (ES256 or ES384).

Example JWS header:

```
{
  "alg": "ES256",
  "typ": "JWT",
  "kid": "agent-key-123"
}
```

8. Evidence Standards

8.1. Evidence Types

ADP defines five standard evidence types:

* SYSTEM_LOGS: Server logs, API logs, error logs

* CONTRACTS: Service agreements, SLAs, terms

* COMMUNICATIONS: Messages, emails, notifications

* FINANCIAL: Transaction records, invoices, payments

* EXPERT: Third-party audits, certifications

Additional evidence types MAY be defined by dispute services using namespaced identifiers (e.g., "x-custom-evidence-type").

8.2. Cryptographic Proofs

Evidence MUST include:

* SHA-256 content hash

* Timestamp (RFC 3161 [RFC3161] or blockchain-anchored)

* Digital signature from submitting party

8.3. Timestamping

Timestamps MUST be:

* RFC 3161 [RFC3161] compliant, OR
* Blockchain-anchored (Bitcoin, Ethereum), OR
* Trusted timestamping service with verifiable audit trail

9. Chain of Custody and Audit Trail

ADP requires an immutable, verifiable chain of custody for all actions in the dispute resolution process. This provides transparency, auditability, and enables appeals or external audits.

9.1. Custody Event Structure

Each custody event MUST conform to the following structure:

```
{
  "@type": "CustodyEvent",
  "eventId": "evt_abc123def456",
  "caseId": "case_abc123",
  "eventType": "EVIDENCE_SUBMITTED",
  "timestamp": "2025-10-15T14:23:45Z",
  "actor": {
    "actorId": "agent://vendor.ai/agent-123",
    "actorType": "CLAIMANT_AGENT",
    "publicKey": "[PEM]"
  },
  "action": {
    "description": "Submitted API response time logs",
    "targetResource": "evidence://ev_def456",
    "actionData": {
      "evidenceId": "ev_def456",
      "evidenceType": "SYSTEM_LOGS"
    }
  },
  "integrity": {
    "contentHash": "sha256:...",
    "previousEventHash": "sha256:...",
    "signature": "[JWS of this event]"
  }
}
```

9.2. Standard Event Types

ADP defines the following standard custody event types:

* *DISPUTE_FILED*: Initial dispute submission
* *DISPUTE_NOTIFICATION_SENT*: Respondent notified
* *RESPONSE_SUBMITTED*: Respondent filed response
* *EVIDENCE_SUBMITTED*: Evidence added by either party
* *PANEL_ASSIGNED*: Arbitrators assigned to case
* *DELIBERATION_STARTED*: Panel begins review
* *DELIBERATION_COMPLETED*: Panel reaches decision
* *AWARD_ISSUED*: Final award published
* *AWARD_DELIVERED*: Award sent to parties
* *COMPLIANCE_VERIFIED*: Remedy compliance confirmed

Services MAY define additional event types with "x-" prefix.

9.3. Cryptographic Linking

Custody events MUST be cryptographically linked in a Merkle chain style:

1. Each event includes the hash of the previous event in "previousEventHash"
2. The first event in a case has previousEventHash set to null or case creation hash
3. The contentHash is computed as SHA-256 of the canonical JSON representation (excluding the signature field)
4. Each event MUST be signed by the actor or the dispute service

This creates an immutable chain where any tampering with an event will invalidate all subsequent event hashes.

9.4. Verification Procedures

To verify the integrity of a custody chain:

1. Retrieve all custody events for the case in chronological order
2. Verify the signature of each event using the actor’s public key
3. Recompute the contentHash of each event and compare with stored hash
4. Verify that each event’s previousEventHash matches the prior event’s contentHash
5. Verify timestamps are monotonically increasing

If any verification step fails, the custody chain is compromised.

9.5. Custody Chain Retrieval

Services MUST provide an endpoint to retrieve the complete custody chain:

GET /adp/v1/custody/{caseId}

Response format:

```
{
  "caseId": "case_abc123",
  "chainStatus": "VALID",
  "events": [
    { /* custody event 1 */ },
    { /* custody event 2 */ },
    ...
  ],
  "chainRoot": "sha256:...",
  "totalEvents": 8
}
```

10. Dispute Lifecycle

10.1. Filing Phase

* Claimant submits DisputeFiling message
* Service validates message format and signature
* Service assigns case ID and notifies respondent

* Service confirms filing with timestamp
* Service records DISPUTE_FILED custody event

10.2. Response Phase

* Respondent submits DisputeResponse within deadline
* Service validates response
* If no response, service MAY issue default judgment per dispute rules
* Service records RESPONSE_SUBMITTED custody event

10.3. Discovery Phase

* Parties exchange EvidenceSubmission messages
* Service validates evidence format and signatures
* Service stores evidence hashes for integrity verification
* Service records EVIDENCE_SUBMITTED custody event for each submission

10.4. Deliberation Phase

* Panel reviews all submissions
* Panel MAY request additional evidence or clarifications
* Panel deliberates according to agreed dispute rules
* Service records DELIBERATION_STARTED and DELIBERATION_COMPLETED events

10.5. Award Phase

* Panel issues DisputeAward message in dual format (JSON + PDF)
* Service validates award signatures from all panel members
* Service delivers award to both parties
* Service publishes award if designated as public
* Service records AWARD_ISSUED and AWARD_DELIVERED custody events

* Parties have specified time to comply with remedy

11. Interoperability

11.1. Service Discovery (.well-known)

Dispute services SHOULD publish capability manifest at:

https://[domain]/.well-known/adp

Example:

```
{
  "disputeService": "https://api.consulatehq.com/adp/v1",
  "protocolVersion": "1.0",
  "supportedRules": ["Consulate-v1.0", "UNCITRAL-2021"],
  "supportedEvidenceTypes": [
    "SYSTEM_LOGS",
    "CONTRACTS",
    "COMMUNICATIONS",
    "FINANCIAL",
    "EXPERT"
  ],
  "maxClaimValue": 10000000,
  "supportedCurrencies": ["USD", "EUR", "ETH"],
  "features": {
    "chainOfCustody": true,
    "dualFormatAwards": true,
    "neutralDiscovery": true
  },
  "endpoints": {
    "disputes": "/adp/v1/disputes",
    "evidence": "/adp/v1/evidence",
    "custody": "/adp/v1/custody/{caseId}",
    "neutrals": "/.well-known/adp/neutrals"
  },
  "publicKeyEndpoint": "/.well-known/jwks.json",
  "pricingTiers": ["micro", "smb", "enterprise"]
}
```

11.2. Capability Negotiation

Parties MAY negotiate protocol extensions via OPTIONS request to the dispute service endpoint.

11.3. Protocol Extensions

Custom evidence types and message fields MAY be added with namespaced keys (e.g., "x-custom-field"). Services implementing such extensions SHOULD document them in their capability manifest.

12. Arbitrator Discovery and Registry

To enable transparent neutral selection, services SHOULD provide a discoverable registry of available neutrals.

12.1. Registry Endpoint

Arbitrator registries SHOULD be published at:

https://[domain]/.well-known/adp/neutrals

Example response:

```
{
  "registryVersion": "1.0",
  "lastUpdated": "2025-10-10T00:00:00Z",
  "neutrals": [
    {
      "neutralId": "arb_human_001",
      "name": "Hon. Jane Smith",
      "type": "HUMAN",
      "qualifications": {
        "certifications": [
          "AAA Certified Arbitrator",
          "JAMS Panelist"
        ],
        "jurisdictions": ["US-CA", "US-NY", "US-TX"],
        "specializations": [
          "Technology Disputes",
          "SLA Breaches"
        ],
        "yearsExperience": 15,
        "casesCompleted": 342
      },
      "availability": {
        "status": "AVAILABLE",
        "nextAvailable": "2025-10-12T00:00:00Z",
        "maxCaseload": 10,
        "currentCaseload": 3
      },
      "publicKey": "[PEM public key]",
      "biasAudit": {
```

```
          "lastAuditDate": "2025-07-01",
          "auditResults":
            "https://consulatehq.com/audits/arb_human_001",
          "fairnessScore": 0.94
        }
      },
      {
        "neutralId": "arb_ai_gpt4_001",
        "name": "GPT-4 Arbitrator Instance",
        "type": "AI",
        "qualifications": {
          "model": "gpt-4-turbo",
          "version": "2024-11-06",
          "specializations": [
            "Contract Interpretation",
            "Technical Disputes"
          ],
          "trainingData":
            "Trained on dispute case law through 2024",
          "casesCompleted": 15420
        },
        "availability": {
          "status": "AVAILABLE",
          "responseTime": "< 5 minutes"
        },
        "publicKey": "[PEM public key]",
        "biasAudit": {
          "lastAuditDate": "2025-09-15",
          "auditResults":
            "https://consulatehq.com/audits/arb_ai_gpt4_001",
          "fairnessScore": 0.91,
          "biasMetrics": {
            "genderBias": 0.02,
            "organizationSizeBias": 0.03,
            "claimAmountBias": 0.01
          }
        }
      }
    ],
    "selectionGuidelines":
      "https://consulatehq.com/docs/neutral-selection"
  }
```

12.2. Arbitrator Manifest Structure

Each neutral entry MUST include:

* *neutralId*: Unique identifier

* *type*: HUMAN or AI
* *qualifications*: Relevant credentials and experience
* *availability*: Current status and capacity
* *publicKey*: For signature verification

Arbitrator entries SHOULD include:

* *biasAudit*: Transparency about bias testing results
* *specializations*: Types of disputes best suited for
* *performance metrics*: Historical data on timeliness and quality

12.3. Selection Guidelines

Services SHOULD provide guidelines for neutral selection. ADP does not prescribe a specific selection algorithm but RECOMMENDS considering:

* Relevant qualifications and specializations
* Availability and response time requirements
* Historical performance metrics
* Bias audit results and fairness scores
* Party preferences (where permitted by dispute rules)
* Cost considerations

For panels, services SHOULD ensure diversity in neutral types (e.g., mixing AI and human neutrals) and backgrounds.

12.4. Bias Auditing and Transparency

To maintain trust in the dispute process, services implementing ADP SHOULD:

1. Conduct regular bias audits of both AI and human neutrals
2. Publish audit methodologies and results
3. Monitor for disparate impact across different types of parties

4. Implement corrective measures when bias is detected

5. Allow parties to challenge neutral assignments based on bias concerns

Bias metrics MAY include analysis across dimensions such as organization size, claim amount, industry sector, jurisdiction, and party characteristics.

13. Security Considerations

* *Replay attacks*: All messages include unique messageId and timestamp. Services MUST reject messages with duplicate IDs or expired timestamps.

* *Man-in-the-middle*: TLS 1.3 with certificate pinning RECOMMENDED for high-value disputes.

* *Evidence tampering*: All evidence MUST be hashed and signed. Tampering is detectable via hash mismatch. Chain of custody provides additional tamper detection.

* *Denial of service*: Services SHOULD implement rate limiting and require filing fees to prevent spam submissions.

* *Privacy*: Parties MAY encrypt evidence payloads with panel's public key. Services SHOULD support confidential dispute upon request.

* *Key compromise*: Services MUST support key rotation and revocation. Historical awards remain valid if signed before revocation. Chain of custody events preserve pre-compromise integrity.

* *Chain of custody integrity*: The Merkle chain linking ensures any tampering with custody events is detectable. Services MUST preserve complete custody chains for the duration of potential appeals.

* *PDF signature verification*: Dual-format awards require both JWS signatures (JSON) and PAdES signatures (PDF). Both MUST be verified independently.

14. IANA Considerations

This document requests IANA to register:

* URI scheme: "agent://" for agent identification

* Media type: "application/aap+json" for ADP messages

* Well-known URI: /.well-known/adp for service discovery

* Well-known URI: /.well-known/adp/neutrals for neutral registry

The JSON-LD context "https://w3id.org/adp/v1" will be registered with W3C for semantic interoperability.

15. Normative References

[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>.

[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>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>.

[RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <https://www.rfc-editor.org/info/rfc7515>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, DOI 10.17487/RFC6749, October 2012, <https://www.rfc-editor.org/info/rfc6749>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <https://www.rfc-editor.org/info/rfc7519>.

[RFC3161] Adams, C., Cain, P., Pinkas, D., and R. Zuccherato, "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)", RFC 3161, DOI 10.17487/RFC3161, August 2001, <https://www.rfc-editor.org/info/rfc3161>.

16. Informative References

[UNCITRAL-CONCILIATION]
United Nations Commission on International Trade Law, "UNCITRAL Model Law on International Commercial Conciliation", 2002, <https://uncitral.un.org/en/texts/arbitration/modellaw/commercial_conciliation>.

[SINGAPORE-CONVENTION]
United Nations, "United Nations Convention on International Settlement Agreements Resulting from Mediation (Singapore Convention on Mediation)", 2018, <https://uncitral.un.org/en/texts/mediation/conventions/international_settlement_agreements>.

[ADRA] United States Congress, "Administrative Dispute Resolution Act of 1995", 1995, <https://www.congress.gov/bill/104th-congress/senate-bill/1224>.

Appendix A. JSON Schema Definitions

Full JSON schemas for all ADP message types are published at:

https://w3id.org/adp/v1/schema/

Reference implementation schemas are available at:

https://consulatehq.com/schema/adp/v1/

GitHub repository with schemas, examples, and tooling:

https://github.com/consulatehq/agentic-dispute-protocol

Appendix B. Example Message Exchanges

Complete examples of filing, response, evidence, custody events, and award message exchanges are available in the ADP repository:

https://github.com/consulatehq/agentic-dispute-protocol

These examples demonstrate common scenarios including:

* Simple SLA breach dispute with complete custody chain
* Multi-party dispute with counterclaim
* Confidential dispute with encrypted evidence
* Cross-border dispute with multiple currencies

* Dual-format award generation and verification

Appendix C. Implementation Notes

Consulate (https://consulatehq.com) provides the reference implementation of ADP with the following features:

* Full ADP v1.0 compliance
* REST and WebSocket APIs
* JavaScript/TypeScript SDK
* Real-time case status updates
* Complete chain of custody tracking
* Automated dual-format award generation (JSON + PDF)
* Arbitrator discovery and selection tools
* Integration with major AI agent platforms

Implementers are encouraged to test interoperability with the Consulate reference implementation during development.

Appendix D. Acknowledgments

This protocol builds upon established alternative dispute resolution frameworks including the UNCITRAL Model Law on International Commercial Conciliation, the Singapore Convention on Mediation, and the U.S. Administrative Dispute Resolution Act, adapting them for autonomous agent interactions and digital dispute resolution.

Author’s Address

Vivek Kotecha
Consulate, Inc.
United States of America
Email: vivek@consulatehq.com
URI: https://consulatehq.com