Network Working Group
Internet-Draft
Intended status: Informational
Expires: 19 November 2026

B. Morrison
Alter Meridian Pty Ltd

18 May 2026

# Identity Accord Protocol: A Peer Ceremony for Bilateral Agreements Between Identity-Substrate-Bound Principals

draft-morrison-identity-accord-01

## Abstract

This memo specifies the Identity Accord Protocol, a peer ceremony by which two principals, each represented by an organisational identity substrate and acting under a recorded delegation from a legal entity, execute a bilateral agreement as a portable, self-verifying COSE-signed CBOR document. The protocol composes DNS-based substrate discovery, Ed25519 sovereign signatures, an append-only identity log, and a tamper-evidence descriptor quorum into a single artefact that is verifiable by any third party with access to the public DNS, the parties' identity logs, and an on-chain anchor of the agreement's content hash. The protocol does not require a central registry, a designated verifier, or any infrastructure operated by the specification's author; verification succeeds when the author's reference deployment is offline. The canonical bilateral target is a mutual non-disclosure agreement, but the wire format generalises to any bilateral consent envelope between two legal entities each represented by an identity substrate. An associated MCP tool surface, an associated pre-send enforcement gate, and an associated disclosure-ledger schema are specified, all of which are optional layers above the wire format. The memo is Informational; the underlying COSE and CBOR formats are normative per [RFC9052] and [RFC8949].

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 19 November 2026.



Table of Contents

## 1. Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

## 2. Introduction

A bilateral agreement between two organisations is, in current commercial practice, a document drafted by either party's legal counsel, signed by an authorised officer of each party, exchanged by email or by a third-party signature platform, and stored in each party's document-management system. The agreement's existence, its terms, and its lifecycle events (execution, amendment, revocation, expiry) are not directly verifiable by any third party; they are matters between the parties and their records. A third party who needs to verify that an agreement is in force may, at best, request a copy from one of the parties.

This memo specifies a different arrangement. Two principals, each representing a legal entity and each bound to an organisational identity substrate, execute the agreement as a portable self-verifying document. The document carries the contract text, the parties' identities, the delegations under which the principals sign,

the agreement's term and jurisdiction, and a set of tamper-evidence descriptors anchored in independent substrates. A third party who receives the document, or who resolves the agreement's content address through public discovery, can verify the agreement's authenticity and lifecycle status against the public DNS, the parties' identity logs, and any on-chain anchor the descriptors reference. No party holds an authoritative copy that the other party lacks; the agreement is symmetric.

The protocol composes with [MCPDNS] for substrate discovery, with [IDPRONOUNS] for the principal-handle namespace, with [IDCOMMITS] for the attribution grammar that names the authorising officer, and with [ORGPOLICY] for the policy stack under which the agreement is admitted to the parties' agent-runtime sessions. An associated pre-send enforcement gate (Section 9) integrates with the agent- runtime governance flow specified by [ORGPOLICY] so that an agreement's permitted-purpose scope can be applied to outbound tool invocations of either party's runtimes.

The canonical bilateral target of the v0 specification is a mutual non-disclosure agreement. The wire format generalises to any bilateral consent envelope: master services agreements, data processing agreements, statements of work, reseller agreements, partnership letters. Multi-party extensions (three or more parties) are out of scope for this version and are anticipated for a successor draft.

## 3. Conventions and Definitions

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.

The following terms are defined for the purposes of this document.

Accord A bilateral agreement executed under the protocol of this memo. The wire-format artefact is the Accord document; the act of reaching mutual signature is the Accord ceremony.

Identity substrate An organisational identity primitive of the kind specified by [ORGPOLICY], addressable by a domain-qualified handle (e.g. ~example.com). Each party to an Accord is represented by one substrate.

Sovereign-tier handle A principal identity handle in the Sovereign

trust tier of [IDPRONOUNS] (e.g. ~alice). An authorised officer of a legal entity signs an Accord under their Sovereign-tier handle.

Delegation instrument A recorded, bounded, revocable, content-addressed assertion by the authorising officer of a party that a named handle (the delegate) is authorised to execute a specified Accord on the party's behalf. The delegation instrument is itself a COSE-signed CBOR document and is included by content address in the Accord payload.

Tamper-evidence descriptor A pointer to an independent substrate against which the Accord's content address is anchored. The minimum descriptor set is defined in Section 7. A verifier requires a substrate-defined quorum of descriptors to consider the Accord tamper-evident.

Permitted purpose A natural-language paragraph in the Accord payload that defines the scope of disclosures permitted under the agreement. The permitted-purpose paragraph is legally authoritative; the structured topic taxonomy of Section 8 is a deterministic runtime classifier and is subordinate to the permitted-purpose prose in any conflict.

Topic taxonomy A structured tag list, scoped to the Accord, identifying the topics on which disclosures are permitted, blocked, or require explicit consent. The taxonomy is operative for runtime gating (Section 9) and is informative for legal interpretation.

Disclosure-ledger event A typed signed event written to an Accord party's identity log recording a permitted disclosure, a blocked attempt, an amendment, a revocation, or an expiry. Events carry metadata and content hashes only; they do not carry the disclosed content.

## 4. Architectural Overview

The protocol comprises five composed layers, each addressable independently:

1. *Wire format* (Section 5). A COSE-signed CBOR document carrying the Accord payload, with two counter-signatures (one per party).

2. *Sovereign signing* (Section 6). Each party's signature is produced by an Ed25519 sovereign key associated with the authorising officer's ~handle, with the signature carried in a COSE_Sign or COSE_Sign1 envelope per [RFC9052].

3. *Discovery* (Section 7). The Accord is publicly discoverable via a content-addressed DNS TXT record under each party's substrate zone, complementing the existing _alter.<domain> record of [MCPDNS].

4. *Tamper-evidence descriptor quorum* (Section 8). Each party contributes descriptors anchoring the Accord's content address in independent substrates: per-party identity log, on-chain anchor, public DNS record. A verifier requires a quorum sufficient for the policy under which the verifier operates.

5. *MCP tool surface and enforcement* (Sections 10 and 11). An optional MCP tool surface allows agent runtimes of either party to participate in the ceremony, query Accord state, and record disclosure events. An optional pre-send enforcement gate applies the Accord's topic taxonomy to outbound tool invocations of either party's runtimes.

Layers 1 through 4 are required for any conformant Accord. Layers 10 and 11 are optional implementation surfaces and may be omitted by parties whose use of the protocol does not extend to agent-runtime-mediated execution.

## 5. Wire Format

An Accord is a CBOR object [RFC8949] carrying the Accord payload, wrapped in a COSE signature envelope [RFC9052].

The Accord payload is a CBOR map with the following keys.

version (text string, REQUIRED) The wire-format version. v0 of this specification uses the literal "identity-accord-v0".

accord_type (text string, REQUIRED) A token identifying the agreement type. Recognised values for v0:

* mutual-nda-v2 for the canonical mutual non-disclosure target.

* msa-v1, dpa-v1, sow-v1, reseller-v1, partnership-v1 for the additional bilateral types this memo anticipates.

Additional values MAY be registered in the IANA Accord Types registry (Section 13).

accord_id (text string, REQUIRED) A UUIDv4 assigned at ceremony commencement. Identifies the Accord within each party's records and in the disclosure ledger.

contract_body (CBOR map, REQUIRED) The contract text and its content address.

* text (text string): the UTF-8 contract body. Legally authoritative.

* content_address (byte string): the SHA-256 hash of the UTF-8 text. Verifiers MUST recompute and compare.

parties (CBOR array, REQUIRED, length 2) One entry per party. Each entry is a CBOR map:

* role (text string): one of party_a, party_b.

* handle (text string): the authorising officer’s Sovereign- tier handle per [IDPRONOUNS] (e.g. ~alice).

* legal_entity (text string): the registered name of the party’s legal entity.

* entity_registry_id (text string): the entity’s registry identifier (e.g. ACN, EIN, company number). Format is jurisdiction-specific.

* sovereign_pubkey (byte string): the Ed25519 public key against which the party’s signature verifies.

* delegation_ref (byte string): the content address of the delegation instrument (Section 6.2).

permitted_purpose (CBOR map, REQUIRED) The agreement’s scope.

* text (text string): the natural-language permitted-purpose paragraph. Legally authoritative in any conflict.

* hash (byte string): SHA-256 of the UTF-8 text.

topic_taxonomy (CBOR map, OPTIONAL) The structured tag list for runtime gating.

* version (text string): the taxonomy version identifier (e.g. "v1").

* permitted_tags (array of text strings): topic tags on which disclosure is permitted.

* blocked_tags (array of text strings): topic tags on which disclosure is refused.

* escalation_tags (array of text strings, OPTIONAL): topic tags that require explicit consent from the disclosing principal at disclosure time.

* nl_authority_anchor (byte string): the hash of permitted_purpose.text, binding the taxonomy to its natural-language source.

term (CBOR map, REQUIRED) Lifecycle parameters.

* effective_date (text string, RFC 3339 timestamp)

* initial_term_days (unsigned integer)

* ordinary_survival_days (unsigned integer, optional)

* categorical_survival (CBOR map, optional): per-category survival rules, where the keys are category identifiers (e.g. trade_secret, personal_information) and the values are either an unsigned integer day-count or the literal "indefinite".

jurisdiction (CBOR map, REQUIRED) Governing law and forum.

* governing_law (text string): jurisdiction identifier (e.g. "NSW, Australia").

* exclusive_forum (text string, OPTIONAL).

tamper_evidence_descriptors (CBOR array, REQUIRED, length >= 2) An array of descriptors per Section 7. Each descriptor is a CBOR map with a type key and type-specific fields.

The Accord payload is canonicalised per the deterministic CBOR encoding rules of [RFC8949] Section 4.2 before signing. Verifiers MUST canonicalise before recomputing content addresses or verifying signatures.

## 6. Signing

### 6.1. Sovereign Signature

Each party signs the canonicalised Accord payload with the Ed25519 private key associated with the Sovereign-tier handle named in the party's entry. Signatures are carried in a COSE envelope per [RFC9052]:

* For ceremonies completed in a single co-signing event, a COSE_Sign envelope with two signatures is REQUIRED.

* For ceremonies completed in two stages (party A signs and publishes; party B counter-signs from the published artefact), each stage MAY emit a COSE_Sign1 envelope and a counter- signature MAY be added later per [RFC9052] counter-signature semantics. Verifiers MUST treat the combined two-signature envelope as authoritative; a single-signature artefact is a draft, not an Accord.

The signature’s protected header SHALL carry:

* alg: EdDSA (RFC 8032).

* content type: application/identity-accord+cbor.

* kid: the content address of the Accord payload.

The signature’s unprotected header MAY carry implementation- specific metadata; verifiers MUST NOT rely on unprotected-header fields for authenticity.

### 6.2. Delegation Instrument

Each parties[].delegation_ref resolves to a delegation instrument: a separate COSE-signed CBOR document, signed by the party’s Sovereign-tier handle, that names the authorised signatory of the present Accord and bounds the delegation’s scope.

The delegation instrument’s payload is a CBOR map with keys:

* version (text string): "identity-accord-delegation-v0".

* principal_handle (text string): the Sovereign-tier handle granting the delegation.

* delegate_handle (text string): the handle authorised to act. In v0, the delegate handle MUST equal the principal handle; Sovereign-to-Instrument delegation is anticipated for a successor draft.

* delegated_accord_id (text string): the accord_id of the present Accord.

* scope (text string): a natural-language description of the scope of the delegation (e.g. "execution of the present Accord and any amendments to it").

* inception (text string, RFC 3339): start of the delegation validity window.

* expiry (text string, RFC 3339, OPTIONAL): end of the delegation validity window; absent implies no expiry beyond the Accord's own term.

* revocation_commitment (byte string): the hash of a revocation token; revocation is effected by publishing the preimage to the principal's identity log.

Verifiers MUST resolve each delegation instrument from its content address, verify its signature against the principal handle's sovereign key, and verify that the delegation's delegated_accord_id equals the Accord's accord_id.

## 7. Tamper-Evidence Descriptor Quorum

Each party contributes one or more tamper-evidence descriptors to the Accord's tamper_evidence_descriptors array. Descriptors anchor the Accord's content address (the SHA-256 of the canonicalised Accord payload) in an independent substrate.

The minimum descriptor types for v0:

identitylog_entry A reference to an event in a party's append-only identity log, the event recording the Accord's content address at execution.

* party (text string): party_a or party_b.

* log_handle (text string): the substrate handle whose log carries the entry.

* entry_id (text string): the log entry identifier.

* signature (byte string): the log's signature over the entry.

onchain_anchor A reference to a transaction on a public blockchain whose payload anchors the Accord's content address (typically via inclusion in a Signed Tree Head of a per-substrate Merkle log inspired by [RFC6962]).

* chain (text string): chain identifier (e.g. "base", "ethereum").

* block (unsigned integer): block number.

* tx (byte string): transaction hash.

* sth_root (byte string): the Merkle root including the Accord's content address.

dns_txt_record A reference to a DNS TXT record under a party's substrate zone whose value is the Accord's content address.

* domain (text string): the fully-qualified domain name of the TXT record (typically _agreement.<content-address-base32>._alter.<party-domain>).

* record_value (text string): the TXT record's value encoding the content address.

The TXT record SHOULD be DNSSEC-validated [RFC4033] per the practice established by [MCPDNS].

wellknown_artefact A reference to a content-addressed artefact published at a party's well-known URI per [RFC8615].

* url (text string): the fully-qualified URL of the well-known resource.

* expected_hash (byte string): SHA-256 of the resource body.

Additional descriptor types MAY be registered in the IANA Tamper-Evidence Descriptor Types registry (Section 13).

A descriptor quorum is sufficient when at least two descriptors of independent type and independent substrate operator have been verified. Implementations SHOULD treat a quorum of one type, or a quorum of two descriptors operated by the same substrate operator, as INSUFFICIENT and refuse to admit the Accord as tamper-evident. Substrate operators SHOULD publish the quorum policy they apply.

Graceful degradation is REQUIRED, meaning parties without access to the full descriptor set SHOULD participate at the minimum-conformant quorum rather than be excluded.

## 8. Discovery

### 8.1. Substrate Discovery

Each party SHALL publish the existence and metadata of its identity substrate under the _alter.<domain> DNS TXT scheme of [MCPDNS]. Substrate discovery for the Accord protocol reuses [MCPDNS] without modification.

### 8.2. Accord Discovery

The existence of an Accord MAY be advertised by each party under a content-addressed sub-record:

_agreement.<content-address-base32>._alter.<party-domain>

The record’s value is a TXT carrying:

* content_address: the base32 encoding of the SHA-256 content address.

* accord_type: the value of the Accord payload’s accord_type field.

* effective_date: the effective date in RFC 3339.

* expiry: the expected expiry timestamp in RFC 3339, computed from effective_date + initial_term_days.

* parties: a comma-separated pair of Sovereign-tier handles (e.g. ~alice,~bob) for human readability.

* sth_anchor (OPTIONAL): a reference to an on-chain anchor per the onchain_anchor descriptor type.

Implementations SHOULD treat absence of an Accord discovery record as orthogonal to Accord validity; parties MAY execute a private Accord (with dns_txt_record descriptors omitted) and distribute the Accord artefact directly out of band. An Accord without DNS discovery still verifies against the descriptor quorum if at least two non-DNS descriptors are present.

### 8.3. Third-Party Verification Walkthrough

A third party who receives an Accord artefact and a content address performs the following verification:

1. Canonicalise the Accord payload per [RFC8949] and recompute the SHA-256 content address. Compare to the provided value.

2. For each party in parties:

    * Resolve the party’s _alter.<domain> per [MCPDNS].

    * Verify the party’s sovereign_pubkey against the public envelope published under [MCPDNS].

   * Resolve the delegation_ref content address and verify the delegation instrument per Section 6.2.

3. Verify the COSE signatures against each party's sovereign_pubkey.

4. Verify the descriptor quorum per Section 7.

5. For each party, query the party's identity log for any accord_revoked event referencing the Accord's content address. Refuse to admit a revoked Accord.

6. Confirm the Accord has not expired against term.

A third-party verifier requires no access to ALTER infrastructure or to either party's private systems beyond the public DNS, the public identity logs, and the on-chain anchor.

## 9. Topic Taxonomy

The optional topic_taxonomy field of the Accord payload provides a deterministic runtime classifier of the agreement's permitted scope. Taxonomy tags are short structured identifiers (e.g. engineering.architecture, finance.revenue, personnel.salaries) drawn from a substrate-published canonical registry or from a per-Accord extension thereof.

The taxonomy is informative for legal interpretation and operative for runtime gating. In the event of a conflict between the topic taxonomy and the permitted-purpose paragraph, the natural-language paragraph prevails per Section 4 (permitted_purpose.text legally authoritative).

Substrate operators SHOULD publish a canonical topic-taxonomy registry at a stable URL under their substrate zone (typical location: https://registry.<substrate-domain>/topic-taxonomy/v1). Accords SHOULD reference the registry version they extend and SHOULD declare per-Accord additions or restrictions explicitly.

## 10. MCP Tool Surface (Optional)

Substrates MAY expose the following MCP tool surface to authenticated agent runtimes of recognised members, enabling runtime participation in Accord ceremony and lifecycle.

begin_agreement(counterparty_handle, accord_type) Creates a draft Accord between the calling party and a counterparty handle. Returns an accord_draft_id.

‘propose_terms(accord_draft_id, contract_content_address,

permitted_purpose, topic_taxonomy, term, jurisdiction,

delegation_ref)‘ Populates the draft with proposed terms.

accept_terms(accord_draft_id) Counterparty’s acceptance; moves the draft to a signing-ready state.

sign_accord(accord_draft_id, sovereign_signature) Attaches an Ed25519 signature from the authorising officer’s Sovereign-tier handle.

publish_tamper_evidence(accord_id, descriptor_set) Emits tamper-evidence descriptors to the substrate’s identity log, to on-chain anchors, and to DNS as configured.

query_accord_status(accord_id_or_content_address) Returns the Accord’s lifecycle state (draft, executed, active, revoked, expired) and the descriptor set. Available to any caller who knows the content address; no privileged authentication is required for this read.

revoke_accord(accord_id, reason) Either party MAY invoke; triggers return-or-destruction obligations and emits agreement_revoked to the identity log.

‘record_disclosure(accord_id, recipient_handle, topic_tags,

content_hash, size, method)‘ Records a permitted disclosure to the disclosure ledger.

record_scope_violation(accord_id, attempted_tags, reason) Records a blocked disclosure attempt for audit.

The MCP tool names above SHALL be registered in the MCP Tool Surface Names registry referenced in [ORGPOLICY] (or a successor specification establishing said registry).

## 11. Pre-Send Enforcement Gate (Optional)

A party MAY operate a pre-send enforcement gate that intercepts outbound tool invocations from the party’s agent runtimes and classifies the invocation’s payload against the topic taxonomies of any active Accords binding the calling principal to the recipient principal.

The gate algorithm:

1. For each prospective outbound tool invocation:

   * Resolve the recipient handle from the invocation’s arguments.

   * Look up any active Accord whose parties set includes the caller and the recipient.

   * If no active Accord exists between the parties, the gate does not apply; the invocation proceeds per the runtime’s default policy (which may be block, prompt, or allow per the runtime’s enforcement-gate specification of [ORGPOLICY]).

2. For each active Accord:

   * Classify the invocation’s payload into a set of topic tags using a substrate-defined classifier. The classifier MAY combine a fast-path structured matcher on payload metadata with a slow-path model-based classifier on payload content.

   * Compare the classified tag set to the Accord’s permitted_tags, blocked_tags, and escalation_tags.

3. Take action:

   * If the classified set lies entirely within permitted_tags, emit a disclosure_recorded event and allow the invocation.

   * If the classified set intersects blocked_tags, emit a scope_violation_blocked event and refuse the invocation with a structured error.

   * If the classified set intersects escalation_tags, present a confirmation prompt to the Sovereign-tier principal and proceed only on confirmation.

   * If classification is ambiguous, fail closed: refuse the invocation and emit scope_violation_blocked with the ambiguity flagged.

Disclosure-ledger events are written to the calling party’s identity log under the event types of Section 11.

The enforcement gate is composable with the per-runtime enforcement-gate specification of [ORGPOLICY]: an outbound invocation MUST satisfy both the party’s runtime gates and any applicable Accord gates. Where both apply, the more restrictive action prevails.

## 12. Disclosure Ledger

Each party SHALL maintain, in its identity log, the following event types under the agreement scope:

agreement_executed Emitted by both parties on ceremony completion. Payload: the Accord's content address, the descriptor quorum, and the signing handle.

disclosure_recorded Emitted per permitted outbound disclosure. Payload: the Accord's content address, the recipient handle, the topic tag set, the content hash, the size in bytes, the method (tool name). Content is NEVER included; only the hash.

scope_violation_blocked Emitted per blocked disclosure attempt. Payload: the Accord's content address, the attempted topic tag set, the block reason.

agreement_amended Emitted on negotiated amendment of an Accord. Payload: the prior and successor content addresses, the diff hash, and the authorising signatures of both parties.

agreement_revoked Emitted on revocation by either party. Payload: the Accord's content address, the revoking party, the reason, the revocation token preimage.

agreement_expired Emitted on term expiry. Payload: the Accord's content address and the expiry timestamp.

Each party's identity log SHOULD be cross-anchored to the counterparty's log via periodic hash-chain exchange so that both parties hold matching event subsets for the agreement. Cross-anchoring is a substrate-side concern and is not specified here beyond the requirement that each party's log is verifiable independently.

## 13. Revocation

Either party MAY revoke an Accord at any time during its term. Revocation:

1. The revoking party publishes a agreement_revoked event to its identity log carrying the revocation token preimage.

2. The substrate emits a notification to the counterparty's subscription channel for the Accord.

3. The counterparty's substrate records the receipt in its own identity log under agreement_revoked with the cross- reference to the originating event.

4. Any pre-send enforcement gate (Section 9) ceases admitting the Accord; subsequent outbound invocations between the parties default to the runtime's no-Accord policy.

5. Return-or-destruction obligations under the contract body take effect per the contract's terms. The protocol records the lifecycle event; the contract specifies the substantive obligations.

Revocation is not retractable. A re-executed agreement between the same parties on the same subject matter is a new Accord with a new accord_id and a new content address.

## 14. Discovery, Identity, and Trust-Tier Composition

The Accord protocol composes with the broader Morrison-family identity architecture as follows.

### 14.1. With Substrate Discovery

The _alter.<domain> TXT scheme of [MCPDNS] supplies both parties' substrate endpoints, signing keys, and capability profiles. Accord-specific records under _agreement.<content-address>._alter.<domain> extend the same zone without creating a new label namespace.

### 14.2. With Handle Tier Semantics

Sovereign-tier handles per [IDPRONOUNS] are the only tier authorised to sign an Accord in v0. Instrument-tier handles MAY participate in the ceremony surfaces (Section 10) under Sovereign-tier delegation per [IDCOMMITS] attribution (Acted-By: is the Sovereign signer; Drafted-With: may name the Instrument that drafted the contract body), but the authoritative signature is always Sovereign-tier.

### 14.3. With Org-Alter Policy Provision

When either party operates an agent runtime under the policy-provision flow of [ORGPOLICY], any active Accord adds an enforcement-gate composition layer above the substrate's default policy stack. The composition rule of Section 9 applies in addition to the strictest-applicable rule of [ORGPOLICY] Section 8.

### 14.4. Multi-Party Anticipation

This memo specifies bilateral Accords only. An N-party Accord (N > 2) requires N-way signature collection, an N-way descriptor quorum, and a generalised topic-taxonomy composition rule. These extensions are anticipated for a successor draft and are explicitly out of scope here.

## 15. IANA Considerations

This memo requests that IANA establish two registries.

### 15.1. Accord Types Registry

A registry of accord_type values for the wire-format field of Section 5. Initial entries:

| accord_type | reference | description |
|---|---|---|
| mutual-nda-v2 | this document | Mutual non-disclosure agreement, v2 template family. |
| msa-v1 | this document | Master services agreement. |
| dpa-v1 | this document | Data processing agreement. |
| sow-v1 | this document | Statement of work. |
| reseller-v1 | this document | Reseller agreement. |
| partnership-v1 | this document | Partnership letter. |

Table 1

Registration policy: Specification Required. New accord_type values are registered by Internet-Draft or by an RFC defining the contract-body shape and any type-specific protocol extensions.

### 15.2. Tamper-Evidence Descriptor Types Registry

A registry of tamper_evidence_descriptors[].type values for Section 7. Initial entries:

| type | reference | description |
|---|---|---|
| identitylog_entry | this document | Reference to an event in a party's append-only identity log. |
| onchain_anchor | this document | Reference to a transaction on a public blockchain anchoring the content address. |
| dns_txt_record | this document | Reference to a DNS TXT record bearing the content address. |
| wellknown_artefact | this document | Reference to a well-known URI artefact bearing the content address. |

Table 2

Registration policy: Specification Required. New descriptor types are registered by Internet-Draft or RFC defining the descriptor fields and the verification procedure.

### 15.3. MCP Tool Surface Names

The MCP tool surface names of Section 10 (begin_agreement, propose_terms, accept_terms, sign_accord, publish_tamper_evidence, query_accord_status, revoke_accord, record_disclosure, record_scope_violation) are registered in the MCP Tool Surface Names registry referenced in [ORGPOLICY]. Establishment of that registry, if not already done, is the subject of [ORGPOLICY]'s IANA Considerations.

### 15.4. Media Type

This memo requests registration of the media type application/identity-accord+cbor per RFC 6838, with the following information:

* Type name: application
* Subtype name: identity-accord+cbor
* Required parameters: none
* Optional parameters: version (the value of the Accord payload's version field).

* Encoding considerations: binary; deterministic CBOR per [RFC8949] Section 4.2.

* Security considerations: see Section 14 of this document.

* Interoperability considerations: see Section 5 of this document.

* Published specification: this document.

## 16. Security Considerations

### 16.1. Sovereign-Key Compromise

An Accord’s authenticity rests on the Sovereign-tier handle’s Ed25519 signing key. Compromise of either party’s signing key permits an attacker to forge new Accords under the party’s identity, or to forge revocations of existing Accords. Mitigations:

* Sovereign-tier signing keys SHOULD be held in hardware-backed custody (HSM, secure enclave, hardware security token) and SHOULD NOT be exported in plaintext under any circumstances.

* The handle’s published envelope per [MCPDNS] is the canonical pubkey; a compromised key SHALL be rotated by publishing a new envelope and recording the rotation in the substrate’s identity log. Verifiers SHOULD check whether the signing key recorded in the Accord was the current key at the time of the Accord’s effective date.

* Tamper-evidence descriptors anchored at the time of execution defend the Accord against post-hoc forgery by anchoring the content address in substrates the attacker does not control.

### 16.2. Descriptor-Quorum Subversion

An attacker controlling one substrate party may attempt to publish descriptors anchoring a falsified Accord content address. Mitigations:

* The quorum policy of Section 7 requires descriptors of independent type and independent substrate operator. An attacker controlling a single substrate cannot satisfy the quorum alone.

* On-chain anchors SHOULD reference chains the attacker does not control; well-known artefacts SHOULD be hosted under the party’s verifiable substrate zone, not under an attacker- controllable third party.

* Verifiers SHOULD compare independent descriptors against each other; descriptors anchoring conflicting content addresses for the same Accord ID are evidence of an attempted forgery.

### 16.3. Delegation-Instrument Replay

A revoked delegation instrument, if its revocation has not been propagated, may be replayed to forge new signatures. Mitigations:

* Delegation revocations SHALL be recorded in the principal's identity log under a typed event before any reliance on the delegation is admitted.

* Verifiers SHALL check the principal's identity log for any revocation event referencing the delegation's content address before treating the delegation as valid.

* Delegation expiry timestamps SHOULD be set conservatively; a delegation that outlives the Accord's effective scope is an unnecessary liability.

### 16.4. Enforcement-Gate Bypass

A party operating the pre-send enforcement gate of Section 9 may have its gate bypassed by a runtime that does not source its policy from the substrate per [ORGPOLICY]. Mitigations:

* Parties SHOULD configure all agent runtimes bound to the party's identity to operate under [ORGPOLICY] policy provision.

* Outbound tool invocations from non-conformant runtimes SHOULD be detected by the substrate's audit-signal flow and the disclosure-ledger comparison SHOULD reveal the divergence.

* Outbound network traffic from non-conformant runtimes is outside the scope of this memo; the Accord's enforcement posture is a protocol layer, not a perimeter control.

### 16.5. Classifier Adversarial Inputs

The pre-send enforcement gate's topic-tag classifier may be adversarially manipulated through crafted payloads that evade classification or that classify into permitted tags spuriously. Mitigations:

* The classifier's ambiguity threshold SHOULD be conservative; ambiguous classifications SHALL fail closed per Section 9.

* The classifier’s structured fast-path SHOULD operate on payload metadata under cryptographic integrity binding, not solely on payload content susceptible to crafting.

* Periodic adversarial-payload rehearsal of the classifier is RECOMMENDED.

## 17. Privacy Considerations

### 17.1. Content Confidentiality

The Accord’s contract body MAY contain confidential terms; the Accord wire format preserves the body’s confidentiality only to the extent that the artefact is not published. Parties wishing to retain content confidentiality SHOULD:

* Omit the dns_txt_record and wellknown_artefact descriptors, retaining only identitylog_entry and onchain_anchor (which record content addresses, not content).

* Distribute the artefact directly between the parties, out of band of the public substrate surface.

### 17.2. Disclosure-Ledger Privacy

Events under the disclosure ledger of Section 11 record content hashes, recipient handles, and topic-tag sets. An adversary with access to a party’s identity log can observe disclosure patterns even without access to disclosed content. Mitigations:

* Identity logs MAY be encrypted at rest; the cross-anchor hash-chain exchange between parties’ logs does not require exposing log contents.

* Recipient handles in disclosure events SHOULD be pseudonymous where the substrate permits; the Accord’s permitted-purpose scope binds the disclosure regardless of recipient pseudonymity.

### 17.3. Third-Party Verification Privacy

A third-party verifier accessing public discovery records and on-chain anchors leaves network and chain-observation footprints. Such verifiers SHOULD operate over privacy- preserving DNS (DNS over HTTPS or DNS over TLS) and SHOULD treat their verification queries as potentially observable.

### 17.4. Cross-Substrate Audit Fan-Out

Where an outbound tool invocation between Accord parties involves a third substrate (e.g. a tool whose execution is mediated by a third party), the disclosure-ledger event is written to all participating substrates per the audit fan-out of [ORGPOLICY] Section 8. Parties SHOULD declare in their permitted-purpose paragraph any third-substrate involvement so that fan-out is anticipated rather than incidental.

## 18. Relation to Companion Memos

This memo composes with five Morrison-family Internet-Drafts.

[MCPDNS] supplies substrate discovery (_alter.<domain> TXT scheme) and the cryptographic identity envelope that publishes each party's Sovereign-tier signing key. The Accord protocol does not introduce new DNS labels except as content-addressed sub-records under the existing _alter. zone.

[IDPRONOUNS] supplies the handle namespace and trust-tier taxonomy. Sovereign-tier handles are the authoritative signatories of an Accord. No new tier is introduced.

[IDCOMMITS] supplies the attribution grammar used by the optional MCP tool surface and by Accord-adjacent git commits recording amendment activity. The Accord protocol's parties[].handle field corresponds semantically to the Acted-By: trailer slot of [IDCOMMITS].

[ORGPOLICY] supplies the agent-runtime policy provision flow into which the Accord's enforcement gate composes. The Accord gate of Section 9 layers above the per-runtime gate set of [ORGPOLICY] Section 5 under a strictest-applicable composition rule.

The substrate-observation posture of the companion substrate-observation memo (the present author's prior I-D) is not directly invoked by the Accord protocol but is a sibling posture: both rest on the principle that bilateral and multilateral coordination problems benefit from substrate- physics commitments rather than from canonical-broker arbitration.

## 19. Implementation Status

A reference implementation of the bilateral Accord ceremony is in active development by the specification's author. Initial ceremony targets are private; post-ceremony case studies are anticipated as the public artefacts of this work.

In the spirit of [RFC7942], the present author notes that this section documents implementation intent and is expected to be removed before the document advances beyond the Independent Stream. No claim of interoperability is made; the reference deployment is a single substrate operated by the specification’s author with a single anticipated counterparty.

## 20. References

### 20.1. 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>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/info/rfc4648>.

[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>.

[RFC8152] Schaad, J., "CBOR Object Signing and Encryption (COSE)", RFC 8152, DOI 10.17487/RFC8152, July 2017, <https://www.rfc-editor.org/info/rfc8152>.

[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>.

[RFC8615] Nottingham, M., "Well-Known Uniform Resource Identifiers (URIs)", RFC 8615, DOI 10.17487/RFC8615, May 2019, <https://www.rfc-editor.org/info/rfc8615>.

[RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <https://www.rfc-editor.org/info/rfc8949>.

[RFC9052] Schaad, J., "CBOR Object Signing and Encryption (COSE): Structures and Process", STD 96, RFC 9052, DOI 10.17487/RFC9052, August 2022, <https://www.rfc-editor.org/info/rfc9052>.

[MCPDNS] Morrison, B., "Discovery of Model Context Protocol Servers via DNS TXT Records", 2026, <https://datatracker.ietf.org/doc/draft-morrison-mcp-dns-discovery/>.

[IDPRONOUNS] Morrison, B., "Identity Pronouns: A Reference-Axis Extension to ~handle Identity Systems", 2026, <https://datatracker.ietf.org/doc/draft-morrison-identity-pronouns/>.

[IDCOMMITS] Morrison, B., "Identity-Attributed Git Commits via Tier-Structured Trailers", 2026, <https://datatracker.ietf.org/doc/draft-morrison-identity-attributed-commits/>.

[ORGPOLICY] Morrison, B., "Org-Alter-Mediated Policy Provision and Governance Inheritance for Agent Runtimes Bound to a Principal Identity", 2026, <https://datatracker.ietf.org/doc/draft-morrison-org-alter-policy-provision/>.

## 20.2. Informative References

[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>.

[RFC6962] Laurie, B., Langley, A., and E. Kasper, "Certificate Transparency", RFC 6962, DOI 10.17487/RFC6962, June 2013, <https://www.rfc-editor.org/info/rfc6962>.

[RFC8615-WK] "Well-Known URIs", 2019, <https://www.rfc-editor.org/rfc/rfc8615>.

## Acknowledgements

This memo grew out of internal architectural work on the question of how two organisations, each represented by an identity substrate, can execute a bilateral agreement as a self-verifying portable artefact without recourse to a central registry, a third-party signature platform, or any infrastructure operated by either party’s vendor. The realisation that the agreement substrate, the identity substrate, and the audit substrate are the same substrate, and that this

collapse is what makes a third-party signature platform structurally redundant between parties who hold their own identity logs, is the load-bearing insight behind this specification.

Author’s Address

Blake Morrison Alter Meridian Pty Ltd Email: blake@truealter.com

Author’s Address

Blake Morrison
Alter Meridian Pty Ltd
Email: blake@truealter.com