

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: 22 November 2026

B. Morrison  
Alter Meridian Pty Ltd (~truealter)  
May 2026

The Briefing-and-Binding Envelope: A Delivery Contract for Agent-to-Principal Decision Moments with Dual-Veto Reconciliation  
draft-morrison-binding-moment-envelope-00

## Abstract

This memo specifies the briefing-and-binding envelope: a delivery contract for the wire-level structure by which an artificial-intelligence agent surfaces a consequential decision to the human principal it acts for, and by which the principal commits, declines, amends, or rejects that decision. The envelope carries eight named slots (a synopsis, findings, recommendations, an offer of detail, a question stem, a set of options each marked with its own reasoning, a single recommended option, and a pair of escape hatches) and is emitted as a structured field of a Model Context Protocol [MCP] tool result. The contribution is the delivery contract itself: a single renderer-agnostic envelope so that the briefing an agent delivers and the binding a principal commits back have one machine-checkable shape across every consuming surface. The load-bearing element is the dual-veto handshake: one escape hatch lets the principal revise the answer space while accepting the question; the other lets the principal reject the question itself and reopen deliberation. Either party may veto. The memo is Informational. No new transport is introduced; the envelope composes with the handle namespace of [IDPRONOUNS] and the MCP tool surface of [POLICYPROV].

## 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 2 November 2026.

## Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

## Table of Contents

1. Status of This Memo . . . . .	3
2. Introduction . . . . .	3
3. Conventions and Definitions . . . . .	4
4. Architecture . . . . .	5
5. The Briefing-and-Binding Envelope . . . . .	6
6. Slot Grammar . . . . .	8
6.1. Slot 1: Synopsis . . . . .	8
6.2. Slot 2: Findings . . . . .	8
6.3. Slot 3: Recommendations . . . . .	8
6.4. Slot 4: Offer . . . . .	8
6.5. Slot 5: Question Stem . . . . .	9
6.6. Slot 6: Options . . . . .	9
6.7. Slot 7: Recommended Option . . . . .	9
6.8. Slot 8: Escape Hatches . . . . .	9
6.9. The meta Slot . . . . .	10
7. The Dual-Veto Handshake . . . . .	10
7.1. Agent-Side Veto . . . . .	10
7.2. Principal-Side Veto: Two Hatches . . . . .	11
7.3. Resolution . . . . .	11
8. MCP Delivery Binding . . . . .	12
8.1. Envelope Placement . . . . .	12
8.2. Tool-Manifest Advertisement . . . . .	12
8.3. Staged Adoption . . . . .	12
8.4. Single-Surface Delivery, Not Broadcast . . . . .	12
9. Renderer Contract . . . . .	13
9.1. Invariants . . . . .	13
9.2. Native Variation . . . . .	13
9.3. Fallback Rendering . . . . .	14
10. Relationship to Companion Memos . . . . .	14
11. IANA Considerations . . . . .	15
12. Security Considerations . . . . .	15
12.1. Synthesis Spoofing . . . . .	15
12.2. Recommendation as Coercion . . . . .	16
12.3. Hatch Suppression . . . . .	16
12.4. Confirmation Routing Under Concurrent Sessions . . . . .	16

12.5. Malformed-Envelope Handling . . . . .	16
13. Privacy Considerations . . . . .	17
14. Implementation Status . . . . .	17
15. Document History . . . . .	18
16. References . . . . .	18
16.1. Normative References . . . . .	18
16.2. Informative References . . . . .	19
Acknowledgements . . . . .	19
Author's Address . . . . .	20

## 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

An artificial-intelligence agent operated on behalf of a human principal periodically reaches a point at which it has synthesised a position and must obtain the principal's commitment before acting: whether to release a scoped view of the principal's data to a third party, whether to accept or decline a delegated action, whether to acknowledge a recommendation, whether to proceed with a step that changes external state. This memo names that point a `_binding moment_` and specifies the wire-level structure through which a binding moment is delivered and resolved.

In current practice the structure of a binding moment is per-tool ad hoc. One agent tool returns a free-text question; another returns a list of options with no synthesis; a third returns a state dump and leaves the principal to perform the synthesis themselves. The consequence is twofold. First, the principal cannot acquire a stable scanning habit, because the shape of the decision changes with every tool. Second, and more seriously, an agent that returns a bare menu of options has pushed the cost of deliberation back onto the principal: the agent has measured but not committed, and the interface has become soft-coercive: the principal must either pick from a frame they did not author or abandon the interaction.

This memo specifies a single delivery contract, the briefing-and-binding envelope, that resolves both problems by construction. The envelope has a fixed eight-slot grammar. The first four slots are the `_briefing_`: a pre-synthesised, scannable account of what the agent observed and what it concluded. The last four slots are the `_binding_`: a quantised decision, a recommended path, and two escape hatches. The envelope is emitted as a structured field of a Model Context Protocol [MCP] tool result, and any consuming surface (a command-line client, a mobile consent sheet, a desktop notifier, an agent-runtime hook) renders the same eight slots into its native user interface. One contract, many renderers.

The load-bearing innovation is the `_dual-veto handshake_`. A binding moment is consensual only if both parties can refuse it. The agent refuses by declining to emit a binding moment at all when it cannot commit to a position: a binding moment with a hedged synopsis is malformed. The principal refuses through two distinct escape hatches: one revises the `_answer space_` (the principal accepts the question but answers on their own terms), the other revises the `_question space_` (the principal rejects the question itself and reopens deliberation). Without both hatches the envelope degrades to a forced-choice interface and the briefing slots stop carrying their weight. Section 6 specifies the handshake.

This memo specifies a delivery contract and a veto handshake. It does NOT specify a persistent log of binding moments, a signed event history between identifiers, or any derivation of signing authority; those concerns are out of scope and are addressed by separate work. The envelope is a transient payload: it is emitted, rendered, resolved, and the resolution is returned. What an implementation durably records of that exchange, and under what attribution, is a matter for the audit-signal mechanism of [POLICYPROV] and is not constrained here.

### 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.

**Principal** The human on whose behalf an agent acts, identified by a Sovereign- tier ~handle as defined by [IDPRONOUNS].

**Agent** An artificial-intelligence runtime acting for a principal,

identified by an Instrument-tier ~handle per [IDPRONOUNS]. The agent is the party that emits a briefing-and-binding envelope.

**Binding moment** A point in an agent's operation at which it has synthesised a position and requires the principal's commitment, decline, amendment, or rejection before proceeding.

**Briefing-and-binding envelope** The eight-slot structured payload defined by this memo, emitted by an agent at a binding moment as a field of an MCP tool result.

**Briefing slots** The first four slots of the envelope (synopsis, findings, recommendations, offer). Pre-synthesised material the principal scans.

**Binding slots** The last four slots of the envelope (question stem, options, recommended option, escape hatches). The quantised decision the principal resolves.

**Escape hatch** A binding-slot affordance through which the principal declines the question as posed. Two are defined: the `_answer-space hatch_`, which accepts the question and revises the answer; and the `_question-space hatch_`, which rejects the question and reopens deliberation.

**Dual veto** The property that a binding moment may be refused by either party: the agent by withholding a malformed binding moment, the principal by invoking either escape hatch.

**Consuming surface** A client that receives an MCP tool result carrying a briefing-and-binding envelope and renders the eight slots into a native user interface.

**Resolution** The principal's response to a binding moment: selection of an option, invocation of the answer-space hatch with an answer, or invocation of the question-space hatch.

#### 4. Architecture

The briefing-and-binding envelope is a payload, not a protocol. It is carried by the existing request/response semantics of the Model Context Protocol [MCP] and introduces no new transport, no new handle category, and no new discovery mechanism.

The arrangement has three roles and one round trip.

The `_agent_` reaches a binding moment in the course of operating for its principal. Instead of returning an ad hoc question, it constructs a briefing-and-binding envelope and emits it as a structured field of the MCP tool result.

The `_consuming surface_` receives the tool result, detects the envelope field, and renders the eight slots into its native interface. The consuming surface performs no synthesis: the briefing slots are rendered as written; the binding slots are rendered as the interactive decision element the surface natively supports.

The `_principal_` observes the rendered briefing, resolves the binding, and the resolution is returned to the agent as the input to the next tool invocation.

The round trip is therefore: agent emits envelope -> surface renders -> principal resolves -> resolution returns to agent. No state is held between trips by the envelope itself. An envelope is valid only for the single resolution it solicits; an agent that requires a further commitment emits a further envelope.

This memo specifies the envelope (Section 4), the slot grammar (Section 5), the dual-veto handshake (Section 6), the MCP delivery binding (Section 7), and the renderer contract (Section 8).

## 5. The Briefing-and-Binding Envelope

An agent at a binding moment SHALL emit a briefing-and-binding envelope as a structured field, named `binding_moment`, of the MCP tool result it returns. The field's value is a JSON [RFC8259] object with exactly the members specified below.

```

{
  "binding_moment": {
    "synopsis":      <string>,
    "findings":     [ <string>, ... ],
    "recommendations": [ <string>, ... ],
    "offer":        <string>,
    "question": {
      "stem":       <string>,
      "options": [
        { "label": <string>, "reasoning": <string> },
        ...
      ],
      "recommended_idx": <integer>,
      "hatches": {
        "free_text": <boolean>,
        "dialogue":  <boolean>
      }
    },
    "meta": {
      "decision_class": <string>,    ; OPTIONAL
      "calibration_note": <string>    ; OPTIONAL
    }
  }
}

```

The members synopsis, findings, recommendations, offer, and question are REQUIRED. The member meta is OPTIONAL; when present, both of its members are individually OPTIONAL.

A consuming surface that receives a tool result with a binding\_moment field that is not a well-formed envelope (a missing REQUIRED member, an out-of-range recommended\_idx, an options list with fewer than two entries) SHALL treat the binding moment as malformed. A malformed binding moment SHALL NOT be rendered as a decision; the surface SHOULD instead render the underlying tool result in its fallback non-envelope form (Section 8.2) and SHOULD surface a diagnostic indicating the envelope was discarded.

The eight slots referenced throughout this memo are: (1) synopsis, (2) findings, (3) recommendations, (4) offer, (5) question stem, (6) options with per-option reasoning, (7) the recommended option, and (8) the escape hatches. Slots 1 through 4 are the briefing slots; slots 5 through 8 are the binding slots.

## 6. Slot Grammar

This section specifies the content contract for each slot. The contract is partly machine-checkable (cardinality, type) and partly a construction discipline the agent SHALL observe for the envelope to be well-formed in the sense Section 4 requires.

### 6.1. Slot 1: Synopsis

synopsis is a string carrying a single sentence that states the agent's position answer-first. The synopsis SHALL commit to a position. A synopsis that hedges (one that defers the decision rather than stating it) renders the envelope malformed: if the agent cannot commit, the binding moment is premature and the agent SHOULD instead return non-binding-moment material that surfaces state and requests permission to deliberate further. The synopsis is the slot that carries the decision if the principal scans nothing else.

### 6.2. Slot 2: Findings

findings is a JSON array of strings. Each string is one terse observation on which the synopsis rests. An array of three to six entries is RECOMMENDED. Each finding SHALL add an observation the synopsis does not itself carry; a finding that restates the synopsis or recapitulates the principal's prior input is filler and SHOULD be omitted.

### 6.3. Slot 3: Recommendations

recommendations is a JSON array of strings. Each string is one concrete move: an action that would change state if taken. An array of two to four entries is RECOMMENDED. A recommendation that names a consideration rather than an action ("weigh the trade-offs") does not satisfy the slot.

### 6.4. Slot 4: Offer

offer is a string offering the principal bounded, on-demand elaboration of any briefing item (for example, "Ask for detail on any item."). The offer is a bounded-pull affordance: depth is available when the principal requests it and is never pushed into the briefing.

#### 6.5. Slot 5: Question Stem

`question.stem` is a string carrying a single neutral, plain-language sentence that poses the decision. The stem SHALL NOT be leading: it SHALL NOT embed the agent's recommendation, and it SHALL NOT be phrased to make one option the path of least resistance. The agent's lean is carried by slot 7, on the option, never by the stem.

#### 6.6. Slot 6: Options

`question.options` is a JSON array of objects. The array SHALL contain at least two and at most four entries. Each entry is an object with two REQUIRED string members:

`label` A short, action-shaped statement of the option.

`reasoning` A single line stating why the principal might choose this option. The reasoning lives on the option, not in the stem; it caches the comparison so that the principal scans rather than deliberates.

The options enumerate the substantive answers to the question. They do not include the escape hatches; the hatches are slot 8 and are not members of the options array.

#### 6.7. Slot 7: Recommended Option

`question.recommended_idx` is an integer: the zero-based index, into the options array, of the option the agent recommends. Exactly one option SHALL be marked. `recommended_idx` SHALL be present, SHALL be non-negative, and SHALL be strictly less than the length of the options array; the value-absent and the two-marked cases are both malformed.

The recommended option is the agent's committed synthesis offered as a contestable position; the principal validates it or vetoes it. Where the agent's analysis genuinely ties two options, the agent SHALL still mark exactly one: the synopsis names the tie, and the mark SHALL fall on the option that minimises the cost of being wrong.

#### 6.8. Slot 8: Escape Hatches

`question.hatches` is a JSON object with two REQUIRED boolean members:

`free_text` The `_answer-space hatch_`. When true, the consuming

surface SHALL render an affordance through which the principal may submit a free-form answer instead of selecting an option. Invoking this hatch accepts the question as posed and revises the answer.

dialogue The `_question-space hatch_`. When true, the consuming surface SHALL render an affordance through which the principal may decline the question itself and reopen deliberation with the agent. Invoking this hatch rejects the framing.

Both members SHALL default to true. An agent that emits a binding moment with either hatch set to false removes a veto path and SHALL do so only where the corresponding revision is genuinely inapplicable; the default posture is that both hatches are open. Section 6 specifies the handshake the hatches realise.

#### 6.9. The meta Slot

meta, when present, MAY carry two members. `decision_class` is a string naming the category of the decision (for example, a consent decision, a permission-delta decision, a recompute decision); a consuming surface MAY use it to select a rendering style. `calibration_note` is a string carrying a short honesty-line preface, included only when the agent's prior framing in the session now reads as materially understating what the agent has since concluded. The note is a costly signal and SHALL NOT be emitted ritually; an implementation SHOULD monitor the rate at which `calibration_note` is populated and treat a persistently high rate as a defect in the agent's earlier framing rather than as normal operation.

### 7. The Dual-Veto Handshake

The defining property of a binding moment, as opposed to a notification, is that either party may refuse it. This section specifies the handshake.

#### 7.1. Agent-Side Veto

The agent's veto is exercised by `_not emitting a binding moment_`. An agent reaches many points at which a principal-facing question would be premature: the agent has not yet synthesised a position, or the available options do not yet partition the decision, or the agent needs to deliberate further before it can commit. At such a point the agent SHALL NOT emit a briefing-and-binding envelope. It SHOULD instead return material that surfaces its current state and requests the principal's permission to continue deliberating. Emitting an envelope with a hedged synopsis (Section 5.1) is the failure this rule forbids: it presents the appearance of a committed decision

while pushing the synthesis cost back to the principal.

The agent-side veto is therefore a precondition on emission: a well-formed envelope is, by construction, one the agent has not vetoed.

## 7.2. Principal-Side Veto: Two Hatches

The principal's veto is exercised through the two escape hatches of slot 8. The two hatches are not redundant; they refuse two structurally different things.

The `_answer-space hatch_` (`free_text`) refuses the `_quantisation_` while accepting the `_frame_`. The principal agrees the agent has asked the right question but holds an answer not among the enumerated options. Invoking this hatch returns a free-form answer; the agent incorporates the answer and proceeds. The frame stands; the option set is treated as non-exhaustive.

The `_question-space hatch_` (`dialogue`) refuses the `_frame_` itself. The principal judges that the agent has asked the wrong question, or has asked a question that rests on a premise the agent does not have grounds for, or that the decision is not ripe. Invoking this hatch does not return an answer; it returns the binding moment to deliberation. The agent SHALL treat a question-space veto as a genuine reopening: it SHALL NOT respond by re-emitting the same question, and it SHALL NOT respond by emitting a further forced-choice envelope without first incorporating the principal's objection. A question-space hatch that leads only to another menu is a broken handshake.

The mapping to deliberative procedure is exact. Selecting an option is a vote. The answer-space hatch is an amendment to the motion. The question-space hatch is a motion to refer the question back. An interface that offers only the vote is not consensual; it is forced-choice.

## 7.3. Resolution

A binding moment is resolved by exactly one of: selection of an option from slot 6; invocation of the answer-space hatch with a free-form answer; or invocation of the question-space hatch. The consuming surface SHALL return the resolution to the agent in a form that distinguishes which of the three occurred, because the agent's correct continuation differs in each case. This memo does not mandate a wire form for the returned resolution beyond the requirement that the three cases be distinguishable; the resolution is carried as the argument to the agent's next tool invocation under the [MCP] semantics already in force.

## 8. MCP Delivery Binding

The briefing-and-binding envelope is delivered as a structured field of a Model Context Protocol [MCP] tool result. This section specifies the binding.

### 8.1. Envelope Placement

An agent tool that surfaces a binding moment SHALL include, in the tool result, a top-level member named `binding_moment` whose value is the envelope object of Section 4. The remainder of the tool result, the tool's ordinary domain payload, is unaffected: the `binding_moment` field is additive. A tool result MAY therefore carry both its conventional payload and a `binding_moment` envelope; a consuming surface that does not understand the envelope ignores the field and renders the conventional payload, and a surface that does understand it renders the envelope.

### 8.2. Tool-Manifest Advertisement

A tool that may emit a `binding_moment` envelope SHOULD advertise the fact in its entry in the MCP tools/list manifest, by carrying a boolean annotation (a RECOMMENDED name is `emits_binding_moment`). The annotation lets a consuming surface decide, before any invocation, whether to prepare its envelope renderer. The annotation is advisory: a surface SHALL still inspect each tool result for the `binding_moment` field, because a tool that sometimes emits an envelope and sometimes does not is conformant.

### 8.3. Staged Adoption

A binding-moment-emitting tool MAY be introduced behind an implementation-defined feature flag, so that the envelope and the tool's conventional payload are emitted side by side during a migration window and the conventional payload is the fallback. This memo places no requirement on the flag mechanism; it notes only that the additive placement of Section 7.1 makes such staged adoption possible without a flag day.

### 8.4. Single-Surface Delivery, Not Broadcast

The briefing-and-binding envelope is delivered to the principal's own consuming surface as the result of a tool the principal's agent invoked. It is a one-to-one delivery between an agent and its principal. The envelope is not a broadcast format: a payload that informs or persuades a population, rather than soliciting one principal's resolution of one decision, is outside the scope of this memo and SHALL NOT be carried as a `binding_moment` envelope.

## 9. Renderer Contract

A consuming surface renders the eight slots into its native interface. This section specifies what the rendering SHALL preserve and what it MAY vary.

### 9.1. Invariants

A conforming renderer SHALL preserve the following across every surface:

- \* All four briefing slots are rendered. A renderer SHALL NOT drop the findings or the recommendations to save space; the briefing is the pre-paid synthesis and is the reason the binding is cheap to resolve.
- \* The recommended option (slot 7) is rendered as visibly distinguished from the other options. The principal SHALL be able to see, without acting, which option the agent recommends.
- \* Both escape hatches whose slot-8 boolean is true are rendered as invocable affordances. A renderer SHALL NOT omit a hatch on the grounds that the option set appears exhaustive; the hatches are the principal's veto and their availability is not the renderer's to withdraw.
- \* The per-option reasoning (slot 6) is rendered with, or made immediately available from, each option. A renderer that shows option labels but hides the reasoning has removed the cached comparison and forced the principal back into deliberation.

### 9.2. Native Variation

Within the invariants of Section 8.1, a renderer MAY map the slots to whatever native element it supports. A command-line client may render the binding as an arrow-key picker with the escape hatches bound to distinct keys. A mobile client may render it as a consent sheet with chip-style options and the recommended chip styled distinctly. A desktop notifier may render the synopsis as a notification title and defer the full envelope to an expanded view. An agent-runtime hook may render the envelope as plain text with a labelled multiple-choice block. The rendering differs; the eight slots and their semantics do not.

### 9.3. Fallback Rendering

A consuming surface that does not implement the envelope renderer, or that has received a malformed envelope per Section 4, SHALL fall back to rendering the tool result's conventional payload. Because the `binding_moment` field is additive (Section 7.1), the conventional payload is always present and the fallback is always available. A surface SHOULD NOT fail an interaction solely because it could not render an envelope.

## 10. Relationship to Companion Memos

This memo composes with the Morrison-family Internet-Drafts as follows.

[IDPRONOUNS] supplies the `~handle` namespace and the Sovereign / Instrument trust-tier taxonomy. The principal is a Sovereign-tier handle and the agent an Instrument-tier handle; this memo introduces no new handle category.

[POLICYPROV] supplies the Model Context Protocol tool surface over an identity substrate from which an agent retrieves policy and to which it emits audit signals. The briefing-and-binding envelope is carried by a tool result on that same surface; a binding moment and its resolution are among the runtime events an implementation MAY record through the audit-signal mechanism that memo specifies. This memo specifies the transient delivery contract; [POLICYPROV] specifies the durable audit channel. The two are deliberately separated: the envelope does not itself persist.

[IDCOMMITTS] supplies the attribution grammar. Where an implementation records a resolved binding moment, the attribution of that record (the Sovereign-tier handle that resolved it and the Instrument-tier handle that drafted the envelope) follows the trailer grammar of [IDCOMMITTS]. This memo does not place attribution slots in the envelope itself; attribution is a property of any durable record, not of the transient payload.

[MCPDNS] supplies the discovery mechanism by which a consuming surface locates the MCP server whose tools emit envelopes. This memo introduces no new discovery surface.

[SUBSTRATE] supplies the coordination posture for the case in which a principal operates several agent sessions concurrently. A binding moment emitted by one session and a binding moment emitted by another are independent payloads; this memo does not coordinate them, and an implementation that surfaces concurrent binding moments deconflicts them under the substrate-observation posture of that memo rather than through any mechanism specified here.

## 11. IANA Considerations

This memo requests no IANA action.

The structured-field name `binding_moment` (Section 4) and the tool-manifest annotation name `emits_binding_moment` (Section 7.2) are illustrative of the reference implementation operated by the present author. They are member names within a Model Context Protocol tool result and tool manifest respectively; they are not protocol identifiers requiring registration. A conforming implementation MAY name these fields by any convention consistent with its MCP tool schema. If a future revision of this memo, or a companion specification, proposes a registry for canonical tool-result field names, that revision will request the corresponding IANA action.

No new transport, port number, URI scheme, media type, or DNS record type is introduced by this memo.

## 12. Security Considerations

The briefing-and-binding envelope is a delivery contract for decisions a principal makes about an agent's behaviour. Its security considerations concern the integrity of the decision, not the confidentiality of a stored record; the envelope holds no durable state.

### 12.1. Synthesis Spoofing

An agent, or a compromised tool impersonating one, may emit an envelope whose briefing slots misrepresent what the agent observed, inducing the principal to validate a recommendation grounded in falsified findings. The envelope cannot prevent this; the briefing is the agent's own account. Mitigation is twofold. First, the offer slot (slot 4) and the answer-space hatch exist precisely so the principal may interrogate any finding or supply an answer the agent did not anticipate; a renderer that drops the offer or the hatch (Section 8.1) removes this defence. Second, the binding moment is a tool result delivered over the [MCP] session, and an implementation SHOULD authenticate the tool surface (for example via the cryptographic identity envelope of [MCPDNS]) so that the principal's

agent is interacting with the substrate it intends.

## 12.2. Recommendation as Coercion

The recommended option (slot 7) carries the agent's lean. An agent that consistently recommends the option most favourable to the agent's operator, rather than to the principal, turns the envelope into a manipulation surface. The structural mitigations are the leading-stem prohibition (Section 5.5), which keeps the lean visibly confined to the option rather than smuggled into the question, and the question-space hatch, which lets the principal reject a frame whose options are all skewed. An implementation SHOULD additionally monitor the rate at which principals invoke the question-space hatch: a persistently high rate against a particular agent or decision class is evidence that the option grammar is systematically mis-framed.

## 12.3. Hatch Suppression

An agent or a renderer that disables an escape hatch removes a veto path. Disabling the question-space hatch in particular converts a consensual binding moment into a forced choice. An agent SHALL set a hatch to false only where the corresponding revision is genuinely inapplicable (Section 5.8), and a renderer SHALL NOT omit a hatch the envelope marks true (Section 8.1). An implementation SHOULD treat a binding moment that disables a hatch as an event worth recording, so that systematic hatch suppression is detectable by post-hoc audit through the mechanism of [POLICYPROV].

## 12.4. Confirmation Routing Under Concurrent Sessions

Where a principal operates several agent sessions concurrently, a binding moment that delegates a consequential action SHOULD be rendered on a surface bound to the principal's own handle, so that an agent session in possession of the principal's session credential cannot resolve the binding moment on the principal's behalf. The envelope itself does not route; routing is a property of the consuming surface and of the session-credential discipline of [POLICYPROV] and [SUBSTRATE].

## 12.5. Malformed-Envelope Handling

A malformed envelope (Section 4) that a renderer nonetheless attempts to render as a decision risks presenting the principal with a decision whose options or recommendation are incoherent. A renderer SHALL discard a malformed envelope and fall back to the conventional payload (Section 8.3) rather than render a partial or repaired decision.

### 13. Privacy Considerations

The briefing-and-binding envelope is a transient payload. It is emitted, rendered, resolved, and discarded; this memo specifies no durable store of envelopes and no log of resolutions. An implementation that does record resolved binding moments does so through the audit-signal mechanism of [POLICYPROV], and the privacy considerations of that mechanism (significance-predicate scope, argument redaction) apply to such records. This memo adds two considerations specific to the envelope.

First, the briefing slots may contain, in the agent's findings, an account of observations the agent has made about the principal or about third parties. An agent SHOULD construct the findings slot to the minimum needed for the principal to validate the decision, and SHOULD NOT use the findings slot as an incidental disclosure channel for material unrelated to the binding moment at hand.

Second, the principal's resolution, including a free-form answer submitted through the answer-space hatch, is principal-authored content returned to the agent. An implementation SHALL treat an answer-space response with the same care as any other principal input and SHALL NOT persist it beyond what the audit-signal significance predicate of [POLICYPROV] requires.

### 14. Implementation Status

This section records implementation experience in the spirit of [RFC7942] and is expected to be removed before the document advances beyond the Independent Stream.

A reference implementation of the briefing-and-binding envelope is operated by the present author against a production Model Context Protocol substrate. An agent tool that surfaces the next question of a conversational identity-discovery flow emits the envelope as an additive `binding_moment` field of its tool result, behind a feature flag, alongside the tool's conventional payload. The envelope's eight slots are populated as specified in Section 5: the synopsis and findings report the state of the discovery flow, the recommended option is selected by an information-gain heuristic over the available answers, and the escape hatches are set per Section 5.8 with the question-space hatch open. A second tool that records a submitted discovery answer emits a follow-on envelope soliciting whether to continue. Consuming surfaces render the envelope to a plain-text multiple-choice block with a visibly marked recommended option and both escape hatches.

No claim of interoperability is made; the reference deployment is a single substrate operated by the specification's author.

## 15. Document History

draft-morrison-binding-moment-envelope-00 (May 2026):

- \* Initial submission.
- \* Specifies the eight-slot briefing-and-binding envelope (Section 4): synopsis, findings, recommendations, offer, question stem, options with per-option reasoning, recommended option, and dual escape hatches.
- \* Specifies the per-slot grammar and construction discipline (Section 5), including the no-hedge rule on the synopsis and the leading-stem prohibition.
- \* Specifies the dual-veto handshake (Section 6): the agent-side veto by withheld emission and the two principal-side escape hatches refusing the answer space and the question space respectively.
- \* Specifies the MCP delivery binding (Section 7): additive `binding_moment` field, tool-manifest advertisement, staged adoption, single-surface delivery.
- \* Specifies the renderer contract (Section 8): rendering invariants, permitted native variation, and fallback rendering.

## 16. References

### 16.1. 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>>.
- [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>>.

[MCP] Agentic AI Foundation, "Model Context Protocol Specification", 2026, <<https://modelcontextprotocol.io>>.

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

## 16.2. Informative References

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

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

[POLICYPROV] Morrison, B., "Policy Provision and Governance Inheritance from an Organisational Identity Substrate", 2026, <<https://datatracker.ietf.org/doc/draft-morrison-org-alter-policy-provision/>>.

[SUBSTRATE] Morrison, B., "Substrate-Observation as an Alternative to Envelope Coordination for Concurrent Sessions", 2026, <<https://datatracker.ietf.org/doc/draft-morrison-substrate-observation/>>.

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.

## Acknowledgements

This memo grew out of internal design work on the structure of agent-to-principal decision moments and the recognition that a binding moment is consensual only when both parties are able to veto it: the agent by declining to pose a question it cannot commit to, and the principal by two distinct refusals. The articulation of the dual-veto handshake as the load-bearing element of the delivery contract is the insight behind this specification.

Author's Address

Blake Morrison  
Alter Meridian Pty Ltd (~truealter)  
Email: [blake@truealter.com](mailto:blake@truealter.com)  
URI: alter:~blake