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

B. Morrison
Alter Meridian Pty Ltd

18 May 2026

Identity-Attributed Git Commits via Tier-Structured Trailers
draft-morrison-identity-attributed-commits-01

Abstract

This document defines a git commit trailer grammar for identity-attributed contributions using the ~handle identity primitive defined in [MCPDNS]. The grammar binds sovereign actors, automated bots, and AI instruments to specific commits via three tier-structured trailers (Acted-By, Executed-By, Drafted-With) and three optional cryptographic trailers (Identity-Signature, Identity-Key-Id, Identity-Anchor). The signature is computed with Ed25519 over the commit's tree hash rather than its commit hash, preserving attribution across rebase, cherry-pick, and squash merge operations. Conformant parsers reject cross-tier category errors (e.g., an Instrument-tier handle in an Acted-By slot) as malformed. The mechanism is provider-neutral, depends only on DNS [RFC1035] and the ~handle resolution algorithm of [MCPDNS], and requires no central authority or platform-specific verification service.

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.

Copyright Notice

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."

This Internet-Draft will expire on October 9, 2026.

2. Copyright Notice

3. Introduction

3.1. Problem Statement

Modern source-control workflows produce commits whose authorship is shared between human contributors, automated bots, and AI instruments operating under varying degrees of delegation. The prevailing mechanisms for attaching identity to a commit are fragmented and individually inadequate for this mixed reality:

* *Git Signed-off-by [DCO].* A legal attestation of contribution rights under the Developer Certificate of Origin. It carries no cryptographic identity proof, no tier distinction, and no resolution to a verifiable key. A Signed-off-by: line is whatever the committer types.

* *Git commit signing (git commit -S).* Cryptographically binding, but the key model is provider-locked: GPG keys uploaded to GitHub, SSH keys uploaded to GitLab, with each platform maintaining its own key directory. There is no DNS-resolved key path and no canonical identity-to-key mapping.

* *Sigstore / gitsign [GITSIGN].* A keyless signing path using short-lived certificates issued from OIDC identity tokens and recorded in the Rekor transparency log. The cryptography is sound, but the identity layer is bound to the operator of the OIDC provider. Migrating between providers re-roots identity. No tier structure exists for non-sovereign signers.

* *Anthropic's Co-Authored-By: Claude convention [ANTHROPIC-COAUTHOR].* An informal text convention for AI attribution. It is unverifiable, ungrammatical with respect to the underlying identity layer (the model is not a co-author in the sovereign sense), and offers no resolution path. Any committer can paste any string.

None of the above provides a provider-neutral, DNS-resolvable, tier-structured identity binding for the human/bot/AI contribution mix that has become typical of agent-augmented codebases.

3.2. Design Goals

This document defines a trailer grammar with the following goals:

1. *Provider-neutral.* No dependency on any specific identity provider, certificate authority, or transparency log operator.

2. *DNS-resolvable.* Public key material is reached via the ~handle resolution algorithm of [MCPDNS], which itself resolves to a DNS TXT record under the handle's policy zone.

3. *Tier-structured.* Three distinct trailer slots correspond to three distinct identity tiers: Sovereign (humans and organisations with cryptographic agency), Bot (autonomous agents under scoped delegation), and Instrument (AI models and tool classes that lack keys). Each slot accepts only handles from its corresponding tier.

4. *Cryptographically verifiable at the sovereign layer.* Sovereign attribution is bound by an Ed25519 signature whose public key is reachable from DNS without prior trust establishment.

5. *Category-safe against misattribution.* Conformant parsers reject cross-tier handle placement (e.g., an Instrument handle in an Acted-By slot) as a structural grammar violation, not a policy decision. Misattribution is detected at parse time.

3.3. Scope

This document specifies:

* The trailer grammar in ABNF [RFC5234].

* Multiplicity, placement, and ordering rules.

* The Ed25519 signature algorithm over the commit's tree hash.

* Verifier behaviour for accepting, rejecting, and surfacing attribution states.

* Security considerations specific to the trailer mechanism.

This document does NOT specify:

* The ~handle identity primitive itself. This is defined by [MCPDNS] and incorporated here by reference.

* The full doctrinal background of the tier taxonomy. Section 3 of this document restates the taxonomy in sufficient detail for the spec to be standalone-readable.

* Sovereign key custody, derivation, and recovery. These are out of scope for this document; implementations are expected to apply standard hardware-backed-key custody practice as summarised in Section 8.

* The IdentityLog transparency-log mechanism backing the optional Identity-Anchor trailer. A future document will define it.

4. Terminology

4.1. Requirements Language

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.

4.2. Definitions

Handle A ~-prefixed identifier per [MCPDNS]. Handles are the unit of identity addressing in this document. Resolution proceeds by extracting the policy zone from the handle and querying the zone's _alter underscore-prefixed TXT record.

Sovereign Tier Handle A handle representing a human individual or formal organisation with direct cryptographic agency. Holds its own private key. Can sign. Examples: ~alice, ~example.com, ~example-co.net.

Bot Tier Handle A handle representing an autonomous agent acting under scoped delegation from a sovereign. Holds a scoped key whose authority is bounded by the sovereign's published delegation policy. Can counter-sign within the delegation envelope. Examples: ~example-deps.bot, ~example-merge.bot.

Instrument Tier Handle A handle representing an AI model, API endpoint, or tool class. Does NOT hold cryptographic keys. Cannot sign. Exists as a DNS-resolvable descriptive label only, suitable for attaching provenance metadata to a contribution without making any identity claim that requires cryptographic backing. Examples: ~example-model-1, ~example-model-2, ~example-model-3.

Tree Hash The SHA-1 (or SHA-256 in git's newer object format) hash of a git tree object, as produced by git write-tree against the staged index, or equivalently by git cat-file -p <commit>^{tree} on an existing commit. The tree hash is a function of the committed content and is invariant under operations that preserve the tree (e.g., rebase, cherry-pick, squash merge into an empty parent).

Tier-Slot Grammar The constraint that a given trailer name accepts handles only from its corresponding tier. Cross-tier placement is a grammatical error, not a policy violation.

Conformant Verifier A consumer of commit trailers that implements the parsing, rejection, and signature-verification rules defined in Section 7.

5. Identity Tier Taxonomy (Informative Reference)

The trailer grammar in Section 4 partitions handles into three tiers. This section defines the taxonomy normatively for the purposes of this specification. Downstream attribution grammars that reuse the taxonomy reference this document.

Tier	Cryptographic Agency	Trailer Slot	Examples
Sovereign	Holds own key, signs	Acted-By:	~alice, ~example.com, ~example-co.net
Bot	Scoped delegated key	Executed-By:	~example-deps.bot, ~example-merge.bot
Instrument	No key, no signature	Drafted-With:	~example-model-1, ~example-model-2, ~example-model-3

Table 1

The tier of a given handle is determined by DNS metadata published under its _alter TXT record per [MCPDNS]. Implementations MAY treat the tier assignments above as authoritative when they correspond to DNS-published tiers; implementations MUST NOT promote or demote a handle's tier without re-resolving DNS.

The key invariant is that Instrument-tier handles cannot make attestational claims. An Drafted-With: trailer is informational provenance metadata, not a verifiable identity binding.

6. Trailer Grammar (Normative)

6.1. ABNF

The following ABNF [RFC5234] defines the syntax of each trailer. Implementations MUST accept exactly this grammar.

```
‘‘‘ acted-by-trailer = "Acted-By:" SP sovereign-handle CRLF executed-
by-trailer = "Executed-By:" SP bot-handle CRLF drafted-with-trailer =
"Drafted-With:" SP instrument-handle CRLF identity-signature =
"Identity-Signature:" SP "ed25519:" base64url-signature CRLF
identity-key-id = "Identity-Key-Id:" SP did-alter-uri CRLF identity-
anchor = "Identity-Anchor:" SP "identitylog://" timestamp "Z/sth/"
seq "#" commit-id CRLF

sovereign-handle = "~" handle-label bot-handle = "~" handle-label
".bot" instrument-handle = "~" handle-label ; tier determined by DNS
resolution per [MCPDNS]

handle-label = 1_63( ALPHA / DIGIT / "-" / "_" / "." ) did-alter-uri
= "did:alter:" sovereign-handle "#" key-id key-id = 1_64( ALPHA /
DIGIT / "-" / "_" ) base64url-signature = 86( base64url-char ) "==" ;
64-byte Ed25519 signature, base64url-encoded base64url-char = ALPHA /
DIGIT / "-" / "_" timestamp = date-fullyear "-" date-month "-" date-
mday "T" time-hour ":" time-minute ":" time-second seq = 1*DIGIT
commit-id = 40HEXDIG / 64HEXDIG ; SHA-1 or SHA-256 commit identifier
‘‘‘
```

The terminals ALPHA, DIGIT, HEXDIG, SP, and CRLF are imported from [RFC5234].

The bot-handle rule requires the .bot suffix, which makes the tier syntactically distinguishable for Bot trailers. Sovereign and Instrument handles share the same surface syntax; their tier distinction is enforced by DNS resolution per [MCPDNS] and by verifier-side rejection of cross-slot placement (Section 7).

6.2. Placement

Trailers MUST appear in the commit message footer block per the git trailer convention [GIT-TRAILERS]. The footer block is separated from the commit message body by exactly one blank line. Each trailer occupies one line of the footer block in the form Key: Value.

A commit message that places trailers anywhere other than the footer block (e.g., interleaved with body paragraphs) is malformed under this specification. Conformant verifiers MUST refuse to parse trailers from outside the footer block.

6.3. Ordering

Trailers SHOULD appear in the following canonical order:

1. Acted-By:
2. Executed-By:
3. Drafted-With:
4. Identity-Signature:
5. Identity-Key-Id:
6. Identity-Anchor:

Verifiers MUST accept trailers in any order, but emitters SHOULD follow the canonical order to support diff-based review. The canonical order is also the order most natural for a human reader: sovereign first, then delegate, then instruments, then proofs.

6.4. Multiplicity Rules

The following multiplicity constraints apply to a single commit:

* *Acted-By:* - Exactly one trailer per signed commit. A squash-merged commit MAY contain multiple Acted-By: trailers aggregating the contributor handles of the squashed commits; this is the only case in which multiple Acted-By: trailers are permitted. Verifiers MUST treat each aggregated Acted-By: as a separate sovereign attribution that requires its own signature pair if cryptographic verification is desired.
* *Executed-By:* - At most one trailer per commit. A commit is executed by at most one bot in a single delegation context.
* *Drafted-With:* - Zero or more trailers per commit. Multi-instrument drafting (e.g., a commit drafted partly with ~example-model-1 and partly with ~example-model-2) is permitted and expected. Multiple Drafted-With: trailers on a single commit form an unordered set; order of appearance is not semantically significant and verifiers MUST NOT attribute differential authority to earlier-appearing entries.

* *Identity-Signature: and Identity-Key-Id:* - These two trailers MUST appear together or not at all. An Identity-Signature: without an Identity-Key-Id: is malformed, and vice versa. When present, they bind to the most recent preceding Acted-By: trailer in the trailer block.

* *Identity-Anchor:* - OPTIONAL in this version of the specification. Implementations targeting Rung-3-compliant attribution (transparency-log-anchored) MUST emit it; all other implementations MAY omit it. Future revisions of this document may upgrade the requirement.

7. Signature Algorithm (Normative)

7.1. Algorithm

The signature algorithm is Ed25519 [RFC8032], which uses SHA-512 internally and produces a 64-byte signature over an arbitrary input message. Implementations MUST use Ed25519 and MUST NOT use Ed25519ph or Ed25519ctx variants.

7.2. Signed Payload

The signed payload is the raw byte representation of the commit's tree hash:

* For repositories using SHA-1 git objects, the payload is the 20-byte SHA-1 tree hash.

* For repositories using SHA-256 git objects, the payload is the 32-byte SHA-256 tree hash.

The tree hash is obtained by git write-tree at signing time (operating on the staged index) or equivalently by git cat-file -p <commit>^{tree} on an existing commit. The hash is signed in its raw binary form, not as a hex-encoded string.

7.3. Rationale for Tree-Hash Signing

The decision to sign the tree hash rather than the commit hash is load-bearing for the operational viability of the scheme.

A commit hash is a function of the commit's tree, its parent commits, its author, its committer, its timestamps, and its message - including, recursively, any trailers in the message. Signing the commit hash directly creates a chicken-and-egg problem (the trailer would be part of the input to its own signature) and, more fundamentally, invalidates the signature on any history-rewriting operation: rebase, cherry-pick, squash merge, amend, and filter-branch all change the commit hash while preserving the tree.

A tree hash is a function of the committed content alone. It is stable across rebase, cherry-pick, and squash merge into an empty parent (the squash result has the same tree as the union of the input trees if no conflicts arose). Signing the tree hash preserves attribution across the full range of git workflows that modern teams depend on, at the cost of being unable to distinguish between two commits with the same tree but different histories.

This trade-off is acceptable: git's own merkle structure ensures content integrity, the parent chain is independently auditable through git itself, and the cases in which two distinct commits share a tree are precisely the cases in which attribution should be preserved (a clean rebase is the same content by the same author).

Where stronger anchoring is required, the optional Identity-Anchor: trailer binds the signature to a specific commit-id within a transparency log entry, recovering commit-level identity at the cost of an external dependency.

7.4. Signature Format

The signature is encoded for placement in the trailer as:

ed25519:<base64url-signature>

The base64url encoding follows [RFC4648] Section 5 (URL- and filename-safe alphabet) without line breaks. A 64-byte Ed25519 signature encodes to 86 base64url characters plus two = padding characters, for a total of 88 characters in the trailer value following the ed25519: prefix.

7.5. Key Derivation and Rotation

Sovereign keys are derived out-of-band; their public components are published under the sovereign's _alter DNS record per [MCPDNS]. Key derivation, custody, and recovery procedures are out of scope for this document. This document treats the sovereign key as a pre-existing Ed25519 keypair whose public component is reachable via the DNS-resolved path of Section 6.1.

Key rotation is supported by the Identity-Key-Id: trailer, which identifies which key was used to sign a given commit. A sovereign's DNS record MAY publish multiple historical keys indexed by key-id, allowing verifiers to validate older commits against the key that was current at the time of signing even after the sovereign has rotated their primary signing key.

8. DNS Resolution (Normative Reference)

8.1. Sovereign Key Resolution

The sovereign handle's public key is resolved via the [MCPDNS] _alter.<zone> DNS record mechanism. Verifiers MUST use the resolution algorithm specified in [MCPDNS] to obtain the public key corresponding to the key-id named in the Identity-Key-Id: trailer.

Verifiers MUST require DNSSEC [RFC4034] validation on the _alter.<zone> lookup when DNSSEC is available for the zone. For zones lacking DNSSEC deployment, verifiers MAY accept the HTTPS .well-known fallback resolution path defined in [MCPDNS], provided the TLS chain validates against the policy domain.

8.2. Instrument Metadata Resolution

Instrument-handle metadata (provider, version, deprecation status, capability profile) is resolved via the same _alter mechanism, but the resolved record is descriptive only. Verifiers SHOULD treat Instrument metadata as informational provenance and MUST NOT treat any field of an Instrument record as an attestational claim. Instrument handles cannot cryptographically sign commits; their DNS records advertise what the model is, not that the commit was authorised by it.

9. Verifier Behaviour (Normative)

A conformant verifier MUST perform the following steps in order:

1. *Parse all trailers from the footer block.* Trailers appearing outside the footer block MUST be ignored.

2. *Reject cross-slot category errors.* For each trailer, resolve the handle's tier per [MCPDNS] (or, where DNS resolution is unavailable, fall back to the syntactic tier indicators of Section 4.1). If any handle appears in a slot other than its tier's slot - for example, an Instrument-tier handle in an Acted-By: slot, or a Sovereign-tier handle in a Drafted-With: slot - the commit is malformed and the verifier MUST reject it as a category error. The error message SHOULD identify the offending trailer by name.

3. *Verify signatures, if present.* If Identity-Signature: and Identity-Key-Id: are present, the verifier MUST:

   a. Extract the key-id from the Identity-Key-Id: trailer. b. Resolve the corresponding public key by querying the Acted-By: handle's _alter record per Section 6.1. c. Compute the commit's tree hash via git cat-file or an equivalent. d. Verify the Ed25519 signature against the tree hash using the resolved public key.

   If signature verification fails, the verifier MUST mark the commit as unverified and MUST NOT report it as having a valid sovereign attribution.

4. *Verify the transparency anchor, if present.* If Identity-Anchor: is present, the verifier SHOULD verify the anchor against the referenced log according to the log's own verification protocol. Failure to verify the anchor MUST be surfaced to the user but MUST NOT silently downgrade the commit's verified status.

A conformant verifier SHOULD additionally:

1. *Cache handle-to-key resolutions.* DNS lookups for the same handle within a single verification pass should be performed at most once. Cache TTL SHOULD respect the DNS record TTL.

2. *Distinguish attribution states in user-facing output.* Verifiers SHOULD present three distinct states to users:

   * verified - Acted-By: present with a valid Identity-Signature: resolving to the published key.

   * claimed - Acted-By: present without a signature, or with a signature whose key cannot be resolved.

   * anonymous - no Acted-By: present.

   Conflating these states is a security defect.

10. Rung 2 Extension - Mode-Attributed Commits

10.1. Motivation

The trailer grammar of Section 4 attributes each commit to individual identities: one or more Sovereign actors, at most one Bot, zero or more Instruments. A class of contributions falls outside this model. Joint manifestations produced by a mode - a bounded, DNS-addressable composition of Sovereigns operating under declared threshold consent - are not authored by any single ~handle. Examples include pair-programming commits where two Sovereigns contributed indistinguishably, working-group decisions ratified by a quorum, AI-majority outputs produced by a mode whose signing authority is defined at the mode level rather than at any member’s level, and cross-organisation commits ratified jointly by two or more modes.

Rung 1 of this specification has no surface for such attributions. A committer must either pick one member arbitrarily as Acted-By:, which misattributes, or omit Acted-By: entirely, which drops the commit to the claimed state. Rung 2 closes this gap by adding a new trailer class, Acted-By-Mode:, whose value is a mode handle and whose semantics are threshold-attestational rather than individual-attestational.

10.2. Trailer: Acted-By-Mode

The following ABNF extends Section 4.1:

acted-by-mode-trailer = "Acted-By-Mode:" SP mode-handle CRLF mode-handle = "~" org-label "." handle-label org-label = 1*63(ALPHA / DIGIT / "-" / "_")

The mode-handle production is syntactically distinct from the Sovereign, Bot, and Instrument handles of Section 4.1 by the required two-label form ~<org>.<handle>. The organisational prefix names the hosting zone of the mode; the inner label names the mode within that zone. A mode handle MUST NOT bear the .bot suffix, and MUST NOT be a bare single-label Sovereign handle.

The semantics of Acted-By-Mode: are threshold-attestational: the trailer asserts that the commit was authored under the compositional consent standing of the named mode, not under the signing authority of any single member. A commit MAY carry both an individual Acted-By: trailer (naming the Sovereign who physically performed the commit operation) and an Acted-By-Mode: trailer (naming the mode under whose standing the work was performed); the combination attests "member M committed on behalf of mode O under compositional consent". A commit MUST be permitted to carry Acted-By-Mode: as its sole Acted-By-* trailer class when the work is purely mode-coupled and no single Sovereign claims primary authorship.

10.3. Tier Resolution

Rung 1 uses the syntactic tier heuristics of Section 4.1 (the .bot suffix; the provider-prefix convention for Instrument handles) for verifier dispatch. Rung 2 replaces these heuristics with DNS-based tier resolution against the handle's _alter.<zone> TXT record per [MCPDNS].

A mode handle MUST resolve to an _alter record whose capability declaration includes cap=mode (or an equivalent capability token established in a future revision of [MCPDNS]). The mode record carries, at minimum:

* type=mode - asserts mode-tier classification.

* threshold=<num>/<den> - declares the signing threshold required for the mode to attest to a commit (for example, threshold=2/3 requires two of three member signatures).

* members=<uri> - a reference to the mode's member attestation keylist, itself a DNS-published or HTTPS-resolved JSON document listing member Sovereign handles and their currently-valid signing-key identifiers.

Verifiers MUST resolve the mode record via DNSSEC where available and MUST fall back to the HTTPS .well-known path of [MCPDNS] only under the same conditions that apply to Sovereign key resolution in Section 6.1.

10.4. Verification

Section 7 is extended for Rung 2 as follows. For each Acted-By-Mode: trailer on a commit, a conformant Rung 2 verifier MUST:

1. Resolve the mode handle per Section N.3 above.

2. Fetch the threshold-attestation metadata (threshold, members).

3. Enumerate the member signatures present on the commit - that is, the set of Identity-Signature: trailers whose corresponding Identity-Key-Id: binds to a Sovereign handle listed in the mode's member keylist.

4. Verify that the count of valid member signatures satisfies the declared threshold.

Under the Rung 2 hard-gate profile, a commit bearing Acted-By-Mode: whose member signature set does not satisfy the declared threshold MUST be marked unverified. Under the Rung 2 warn-only profile (the rollout default), verification is limited to parse-only and slot-category correctness; threshold satisfaction is surfaced as informational but does not downgrade the commit's verified status.

10.5. Slot Exclusivity

Acted-By-Mode: accepts mode-tier handles ONLY. Verifiers MUST reject Sovereign, Bot, and Instrument handles appearing in this slot as cross-slot category errors per the rules of Section 7, Step 2. The existing Acted-By: slot continues to accept Sovereign handles ONLY, regardless of whether an Acted-By-Mode: is also present.

10.6. Interaction with Co-Authored-By

The Co-Authored-By: convention of [ANTHROPIC-COAUTHOR] and the equivalent GitHub commits-UI trailer remain unchanged by this extension. Commits bearing Acted-By-Mode: SHOULD include a Co-Authored-By: line rendering the mode handle in human-readable form, so that the authoring mode is visible in commit-UI surfaces that do not parse the identity trailer grammar natively.

10.7. Rung 2 Rollout

Rung 2 MUST be deployed in two sub-phases. In the parse-only sub-phase, conformant hooks and verifiers accept and recognise the Acted-By-Mode: trailer, enforce slot-exclusivity per Section N.5, and surface member signature counts informationally. They do NOT enforce the threshold. The parse-only sub-phase is permitted before the _alter. resolver has shipped in the referenced backend implementation.

In the signature-verification sub-phase, verifiers additionally enforce that the member signature set satisfies the mode's declared threshold. The signature-verification sub-phase MUST NOT be enabled before the _alter. resolver is in production and the mode record schema of Section N.3 is stable.

11. Security Considerations

11.1. Sovereign Key Compromise

If a sovereign's signing key is compromised, the sovereign rotates the key and publishes the new key under a new key-id in their _alter record. The previous key SHOULD remain published as a historical record so that commits signed during its validity period continue to verify. Sovereigns SHOULD also publish revocation metadata distinguishing keys that were rotated for hygiene from keys that were rotated due to compromise; verifiers encountering a compromise-revoked key SHOULD warn the operator that any commit signed by that key is suspect even if the signature still validates mathematically.

11.2. Instrument Handle Spoofing

Because Instrument handles cannot sign, the Drafted-With: trailer is an unverified provenance claim. A malicious committer can always paste Drafted-With: ~example-model-1 into a commit they hand-wrote. Implementations MUST treat Instrument attribution as informational, not attestational, and MUST NOT extend trust decisions on the basis of an Instrument trailer alone. This is explicit by design: the Instrument tier is a documentation mechanism, not an attestation mechanism. The protection against Instrument-trailer abuse is the sovereign signature on Acted-By:, which binds a real cryptographic identity to the overall commit and to the committer's claim about what tools they used.

11.3. DNS Poisoning

A successful DNS poisoning attack against the _alter.<zone> zone could redirect verifiers to a substitute public key under the attacker's control. This risk is mitigated by:

* DNSSEC validation when available. Verifiers SHOULD require DNSSEC on the policy zone and MAY refuse to verify against an unsigned zone.

* The HTTPS .well-known fallback path defined in [MCPDNS], which terminates the trust chain at the TLS certificate of the policy domain.

* Independent transparency-log anchoring via the optional Identity-Anchor: trailer, which provides a second source of truth that is unaffected by DNS poisoning.

11.4. Tree-Hash Collision

Most git repositories currently use SHA-1 for tree hashing. SHA-1 is cryptographically weakened (SHAttered, 2017) for collision resistance, and tree-hash signing inherits that weakness. Implementations operating in high-assurance contexts SHOULD migrate to SHA-256 git objects, which use SHA-256 for the tree hash and eliminate the SHA-1 weakness. Until such migration is complete, verifiers SHOULD record both the tree hash and the commit hash in any local audit log so that any future SHA-1 collision attack against the verifier’s history is detectable ex post.

11.5. Squash-Merge Trailer Aggregation Race

The aggregation of contributor Acted-By: trailers into a squash-merged commit is an implementation responsibility of the hosting platform or merge tool. If aggregation is skipped or fails silently, the trailers from individual contributor commits are lost, and the merge commit appears to have a single sovereign when it actually had several. Implementations performing squash merges MUST validate that contributor trailers have been aggregated before completing the merge, and SHOULD refuse to complete a squash that loses trailer attribution. This is an implementation concern, not a protocol-level issue, but it is listed here because the operational consequence of a missing trailer is a silent loss of attribution.

11.6. Key Custody at the Commit-Signing Boundary

The pre-commit hook (or analogous integration point) that invokes the signing operation is a trust-sensitive boundary: the hook runs in the unprivileged developer process and may have access to the sovereign’s private key. Implementations SHOULD route signing through a privileged helper - for example, a unix domain socket exposed by a dedicated signing daemon, or a hardware authenticator using WebAuthn PRF - rather than reading the private key directly from unprivileged process memory. Direct key handling in the developer process is acceptable for prototyping but MUST NOT be relied upon in production deployments where commit attribution carries weight.

11.7. Negative-Attribution Risk

A committer may deliberately omit the Drafted-With: trailer to conceal AI-instrument involvement in a contribution. This is detectable only by out-of-band evidence and is not addressable at the protocol layer. Where AI-disclosure obligations exist (for example, in regulated software development contexts), they SHOULD be enforced at the policy layer with this protocol providing the truthful path for honest committers, not the verification path for dishonest ones.

12. IANA Considerations

12.1. Git Trailer Name Registration

At the time of writing, IANA does not maintain a registry of git commit trailer names. If such a registry is established, this document requests registration of the following trailer names with reference to this specification:

* Acted-By
* Executed-By
* Drafted-With
* Identity-Signature
* Identity-Key-Id
* Identity-Anchor

Until a formal registry exists, this document recommends that implementers coordinate via the ALTER discovery community and treat the trailer names defined here as reserved for the identity-attributed commit grammar.

12.2. URI Scheme Dependencies

This document depends on the did:alter: URI scheme via the Identity-Key-Id: trailer. The alter: URI scheme is the subject of IANA considerations in [MCPDNS]; this document does not separately register it.

The identitylog:// URI scheme used by the optional Identity-Anchor: trailer is reserved by this document for future registration when a normative IdentityLog specification is published. Implementations encountering identitylog:// URIs without a registered scheme MUST treat the anchor as an opaque reference and SHOULD NOT attempt resolution.

12.3. No Other IANA Actions

This document requests no other IANA actions.

13. Relationship to Existing Standards

The trailer grammar defined here is intended to coexist with prior commit-attribution mechanisms rather than to replace them.

Mechanism	Purpose	Coexistence with this spec
Git Signed-off-by [DCO]	Legal attestation of contribution rights	Orthogonal. A commit MAY carry both a Signed-off-by: and an Acted-By: trailer. They answer different questions.
Git commit signing (git commit -S)	Cryptographic identity via GPG/SSH key directories	Orthogonal. A commit MAY be both GPG-signed and Acted-By-signed. Verifiers handle each path independently.
Sigstore / gitsign [GITSIGN]	Keyless cryptographic identity via OIDC	Architecturally adjacent. Different identity provider model (OIDC + Rekor vs DNS-resolved DID + IdentityLog). May coexist.
Anthropic Co-Authored-By: Claude [ANTHROPIC-COAUTHOR]	Informal AI co-authorship convention	Superseded for AI attribution by Drafted-With: (Instrument tier). Implementations MAY emit both during a transition window.

Linux kernel Assisted-by [LINUX-AI-ASSIST]	Disclosure-only attribution of AI assistance in kernel contributions; legal liability remains with a human via DCO (Signed-off-by)	Architecturally adjacent and complementary. Assisted-by: discloses AI involvement in prose-readable form; Drafted-With: binds the same involvement to a DNS-resolvable Instrument ~handle with machine-verifiable metadata. Both MAY appear on the same commit. Bidirectional mapping and a canonical emission pattern are defined in Appendix A.

Table 2

The Instrument tier is novel to this specification and has no analogue in any existing mechanism. Sigstore identifies signers; the DCO attests to legal rights; the Anthropic convention is a plain-text marker. None expresses the structural distinction between a sovereign actor, a delegated bot, and a non-signing instrument. This distinction is the load-bearing contribution of the present document.

14. Acknowledgments

The author thanks colleagues at Alter Meridian Pty Ltd for the framing of identity tiers, and external adversarial reviewers for pressure-testing the tier-slot grammar and the cross-tier rejection rules. Additional contributors will be named at review time.

15. References

15.1. Normative References

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4034] Arends, R., et al., "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, January 2017.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017.

[RFC8552] Crocker, D., "Scoped Interpretation of DNS Resource Records through 'Underscored' Naming of Attribute Leaves", BCP 222, RFC 8552, March 2019.

[RFC8785] Rundgren, A., et al., "JSON Canonicalization Scheme (JCS)", RFC 8785, June 2020.

[MCPDNS] Morrison, B., "Discovery of Model Context Protocol Servers via DNS TXT Records", draft-morrison-mcp-dns-discovery, work in progress.

15.2. Informative References

[RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, July 2016.

[DCO] "Developer Certificate of Origin v1.1", https://developercertificate.org/, 2004.

[GIT-TRAILERS] "git-interpret-trailers(1)", https://git-scm.com/docs/git-interpret-trailers.

[GITSIGN] "gitsign: Keyless Git Signing", https://docs.sigstore.dev/cosign/signing/gitsign/.

[SIGSTORE] "Sigstore: Software Signing for Everybody", https://www.sigstore.dev/.

[ANTHROPIC-COAUTHOR] Anthropic, "Co-Authored-By: Claude - convention for AI-assisted commits".

[LINUX-AI-ASSIST] The Linux Kernel Contributors, "AI Coding Assistants", Linux kernel Documentation, Documentation/process/coding-assistants.rst, 2024.

[MORRISON-IFT] Morrison, B., "Identity Field Theory: Toward a Physics of Being Known", https://doi.org/10.6084/m9.figshare.31951383, 2026.

16. Appendix A. Interoperability with Linux Kernel Assisted-by Policy

16.1. A.1. Scope

The Linux kernel project has adopted a formal policy for AI-assisted contributions in Documentation/process/coding-assistants.rst [LINUX-AI-ASSIST]. The policy reserves the DCO Signed-off-by: trailer to human developers exclusively and introduces a disclosure trailer, Assisted-by:, to document AI involvement. Because the kernel is the most widely deployed code review culture in the world, its conventions set a de facto norm for AI-attribution in free and open source contributions more broadly.

This Appendix defines how implementations of the present specification SHOULD interoperate with the Linux kernel Assisted-by: trailer. Both mechanisms are disclosure mechanisms for AI involvement and MAY coexist on the same commit. They are not substitutes.

16.2. A.2. Format Comparison

The Linux kernel Assisted-by: trailer has the shape:

Assisted-by: AGENT_NAME:MODEL_VERSION [TOOL1] [TOOL2]

AGENT_NAME is a human-readable vendor or product label (e.g. "Claude", "Copilot"). MODEL_VERSION is a human-readable version string (e.g. "opus-4-6", "3.5-Sonnet"). The bracketed tokens are zero or more specialised analysis utilities employed during the AI-assisted authoring process (e.g. "coccinelle", "sparse", "smatch", "clang-tidy"). Standard developer tools such as git, gcc, and make are excluded by policy.

The trailer defined in Section 4 of this document has the shape:

Drafted-With: ~<instrument-handle>

~<instrument-handle> is an Instrument-tier ~handle (Section 3) that resolves via DNS (Section 6.2) to machine-readable metadata describing the AI instrument, including a structured vendor, model, and tool manifest.

16.3. A.3. Mapping: Drafted-With -> Assisted-by

An implementation that emits both trailers MAY derive the Assisted-by: value from Drafted-With: as follows.

1. Resolve ~<instrument-handle> via the DNS TXT record at _instrument.<domain> as defined in Section 6.2. The record SHOULD include at minimum the fields vendor, model, and tools (an ordered list).

2. Format the Assisted-by: value as:

   Assisted-by: <vendor>:<model> [tool1] [tool2] ...

   where <vendor> and <model> are taken verbatim from the resolved DNS TXT record and each [toolN] is drawn from the tools list.

3. If the tools list is empty, the bracketed suffix is omitted.

4. Implementations MUST NOT include standard developer tools (such as git, gcc, or make) in the tools field of the Instrument metadata record. This exclusion is a policy alignment with [LINUX-AI-ASSIST] and is enforced at the metadata-publication boundary, not at the verifier.

Example. Given an Instrument DNS TXT record for ~example-model-1 that resolves to:

vendor=ExampleVendor;model=example-model-1;tools=MCP

the emitted trailer pair is:

Drafted-With: ~example-model-1 Assisted-by: ExampleVendor:example-model-1 [MCP]

16.4. A.4. Mapping: Assisted-by -> Drafted-With

The reverse mapping (from a Assisted-by: trailer to an Instrument ~handle) is NOT generally computable without additional information, because the Assisted-by: format does not carry a DNS-resolvable identifier. Two degraded modes are available.

Mode 1 (canonical registry). A verifier MAY maintain a local vendor-to-handle canonicalisation table (e.g. Anthropic -> namespace prefix ~cc-, with model-string rules). When the AGENT_NAME:MODEL_VERSION tuple resolves under such a table, the verifier MAY treat the canonicalised ~handle as an inferred Instrument binding. No normative canonicalisation table is specified here; a companion informational document may define one if and when community practice converges.

Mode 2 (informational only). When no canonical mapping exists, the verifier MUST treat Assisted-by: as prose-level disclosure only. In particular, the verifier MUST NOT promote the commit to any trust tier on the basis of Assisted-by: alone, and MUST NOT treat the AGENT_NAME:MODEL_VERSION tuple as an authenticated assertion.

Implementations of this specification SHOULD prefer emission of both trailers at authoring time over reverse mapping at verification time, because authoring-side emission preserves the DNS-resolvable metadata that the reverse mapping loses.

16.5. A.5. DCO and Acted-By: Liability Boundary

The Linux kernel policy reserves Signed-off-by: to human developers on the grounds that only a human can legally certify DCO compliance. The present specification’s Acted-By: trailer is similarly reserved to Sovereign-tier ~handle bindings (Section 3), which are human-bound in the trust model this specification is built on.

Signed-off-by: and Acted-By: are therefore complementary, not redundant:

* Signed-off-by: is a legal attestation bound to an email identity. It is evaluated by human reviewers against the DCO text and is not cryptographically verifiable at commit time.

* Acted-By: is a cryptographic identity binding rooted in DNS and a sovereign key (Sections 5 and 6). It is evaluated by verifier software and produces a machine-verifiable result independent of DCO evaluation.

Neither trailer supersedes the other. Contributors to projects that require DCO compliance (such as the Linux kernel) MUST continue to provide Signed-off-by:. Contributors who also operate within the ALTER identity framework SHOULD additionally provide Acted-By:.

16.6. A.6. Recommended Emission Pattern

A contributor operating on both stacks is RECOMMENDED to emit the following trailer block:

Signed-off-by: <Full Name> <email> Acted-By: ~<sovereign-handle> Assisted-by: <vendor>:<model> [tool1] [tool2] ... Drafted-With: ~<instrument-handle> Identity-Signature: ed25519:<base64url> Identity-Key-Id: did:alter:~<sovereign-handle>#key-<yyyy-mm>

Verifiers of either specification handle each line independently. A verifier that only understands the DCO model (such as a kernel maintainer tree) will evaluate Signed-off-by: and Assisted-by: and ignore the remaining trailers. A verifier that implements the present specification will evaluate Acted-By:, Drafted-With:, and the Identity-* trailers and ignore the kernel-specific trailers. Neither verifier fails closed on the presence of the other’s trailers.

16.7. A.7. Roundtripping Concerns

The Assisted-by: format does not preserve full Instrument DNS metadata (notably the DID key identifier and any rotation fingerprint). Implementations performing archival or mirroring MUST NOT attempt to reconstruct Drafted-With: from Assisted-by: in a way that loses fidelity; the original Drafted-With: trailer, if present, MUST be preserved verbatim.

16.8. A.8. Open Items

The following items are out of scope for the present document and are candidates for a follow-up Informational draft:

* A canonical vendor-to-~handle registry, with rules for conflict resolution across competing canonicalisations.

* Normative handling of Assisted-by: trailers whose AGENT_NAME:MODEL_VERSION tuple contains characters outside the Instrument-tier handle grammar (Section 3).

* Extension of the Linux kernel exclusion rule (forbidding git, gcc, make in Assisted-by:) to the Instrument tools field as a normative requirement rather than a metadata-publisher policy.

17. Author’s Address

Blake Morrison Alter Meridian Pty Ltd

Email: blake@truealter.com URI: https://truealter.com

18. References

18.1. Normative References

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[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>.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <https://www.rfc-editor.org/info/rfc4034>.

[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>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[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>.

[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>.

[RFC8552] Crocker, D., "Scoped Interpretation of DNS Resource Records through "Underscored" Naming of Attribute Leaves", BCP 222, RFC 8552, DOI 10.17487/RFC8552, March 2019, <https://www.rfc-editor.org/info/rfc8552>.

[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>.

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

18.2. Informative References

[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>.

[SIGSTORE] "Sigstore: Software Signing for Everybody", 2022, <https://www.sigstore.dev/>.

[GITSIGN] "gitsign: Keyless Git Signing", 2023, <https://docs.sigstore.dev/cosign/signing/gitsign/>.

[DCO] "Developer Certificate of Origin v1.1", 2004, <https://developercertificate.org/>.

[GIT-TRAILERS] "git-interpret-trailers(1)", n.d., <https://git-scm.com/docs/git-interpret-trailers>.

[ANTHROPIC-COAUTHOR] Anthropic, "Co-Authored-By: Claude - convention for AI-assisted commits", 2025, <https://docs.anthropic.com/claude/docs/co-authored-by-convention>.

[MORRISON-IFT] Morrison, B., "Identity Field Theory: Toward a Physics of Being Known", 2026, <https://doi.org/10.6084/m9.figshare.31951383>.

Author’s Address

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