

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

B. Morrison
Alter Meridian Pty Ltd
18 May 2026

Identity Pronouns: A Reference-Axis Extension to ~handle Identity
Systems
draft-morrison-identity-pronouns-01

Abstract

This document defines an identity pronoun grammar as a reference axis orthogonal to the ~handle identity tier taxonomy defined in [MCPDNS] and [IDCOMMITS]. A pronoun is a session-scoped reference that resolves client-side to a concrete handle using local session state before any cryptographic, DNS, or federation operation. The entity-class taxonomy (Sovereign, Bot, Instrument) is unchanged; this specification introduces Absolute vs Pronoun as an orthogonal axis. A pronoun MUST NOT appear in a capability token, in a DNS record, in an Accord signature, or in any inter-organisational protocol payload. The reference implementation defines a single Wave-1 pronoun, ~org, that resolves to the concrete handle of the organisation bound to the caller's current session. An appendix defines a relative-path pronoun grammar (e.g. ~./architect, ~../weaver) as a non-normative design surface for future work. The mechanism is provider-neutral, introduces no new cryptographic primitive, and imposes zero new load on DNS, capability-token issuers, or federated resolvers.

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

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. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Status of This Memo	3
2. Introduction	3
3. Conventions and Definitions	4
4. The Reference Axis	4
4.1. Two Orthogonal Axes	4
4.2. Wire Invariant	5
4.3. Resolution Algorithm	6
5. The ~org Pronoun (Wave 1)	7
5.1. Definition	7
5.2. Session Binding	7
5.3. Resolution Failures	7
5.4. Example (non-normative)	8
6. Pronoun Lexicon Extensibility	8
6.1. Per-Client Lexicon	8
6.2. Per-Org Lexicon (Future Work)	8
7. Relative-Path Pronoun Grammar (Appendix, Non-Normative)	8
8. IANA Considerations	9
9. Security Considerations	10
9.1. Pronoun Spoofing	10
9.2. Pronoun Leakage	10
9.3. Binding Freshness	10
9.4. Cross-Tenant Confusion	10
10. Implementation Status	11
11. References	11
11.1. Normative References	11
11.2. Informative References	12
Appendix A. Acknowledgements	12
Author's Address	12
Author's Address	12

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

The `~handle` identity primitive defined in [MCPDNS] binds a textual identifier to a cryptographic principal via DNS TXT records under an `_alter.` prefix. The tier taxonomy subsequently locked in [IDCOMMITS] partitions handles into three parser-enforced entity classes -- Sovereign, Bot, and Instrument -- each with distinct capabilities and invariants. The three-tier taxonomy answers the question: `_what kind of thing does this handle bind to?`

This document answers a second, orthogonal question: `_how is the handle referenced?` A reference may be absolute (a literal handle transmitted on the wire) or pronominal (a session-scoped reference resolved by the client before transmission). Every pronoun resolves to one of the three entity classes at evaluation time; the entity taxonomy is unchanged.

The motivating use case is multi-organisational identity. A human agent who is a member of two organisations requires a pronoun that resolves to "the organisation bound to my current session" without hard-coding either organisation's concrete handle into the agent's command surface. A shell user relies on the same pattern when `~` [POSIX-TILDE] resolves to `$HOME` for the invoking user rather than any specific directory path. Git [GIT-REVISIONS] relies on `HEAD` and `@{upstream}` as first-class reflexive references. CSS [CSS-SELECTORS-4] relies on `:root` and `:host` as grammar-level contextual references. Kubernetes [K8S-SUBJECTACCESSREVIEW] relies on `:self` as a protocol-level reflexive subject.

Each of these precedents shares three properties. The pronoun is (1) first-class in the surface grammar, (2) resolved before the underlying protocol sees it, and (3) absent from any payload that crosses a trust boundary. This specification adopts the same three properties for the ~handle system.

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:

- * ***Handle.*** A textual identifier of the form ~ followed by a label, as defined in [MCPDNS].
- * ***Absolute handle.*** A handle whose textual form is identical to the handle it identifies (e.g. ~alice, ~example.com).
- * ***Pronoun.*** A handle whose textual form is a reserved reference expression that resolves to a concrete absolute handle via a defined resolution algorithm (e.g. ~org, ~me).
- * ***Concrete handle.*** The absolute handle produced by resolving a pronoun; in contrast to the pronoun's surface form.
- * ***Resolution.*** The operation of replacing a pronoun with its concrete handle in a given context.
- * ***Session state.*** The set of bindings available to a client at the time of resolution, including but not limited to the caller's sovereign handle, the caller's currently-bound organisation, and any pronoun lexicon loaded by the caller's client.
- * ***Wire.*** Any transport boundary crossed by an identity-bearing payload, including capability tokens, DNS records, MCP tool invocations, HTTP requests to federated peers, and signatures attached to Accord documents.

4. The Reference Axis

4.1. Two Orthogonal Axes

The ~handle identity system has two orthogonal axes. The first axis is ***entity class***, defined in [IDCOMMITS]:

Sovereign | Bot | Instrument

The second axis, defined by this document, is **reference type**:

Absolute | Pronoun

Every handle is a point in the two-dimensional space defined by these axes. An Absolute Sovereign is a concrete handle such as `~alice`. A Pronoun Sovereign is a reference such as `~org` that resolves to a concrete Sovereign handle at evaluation time. An Absolute Bot is `~example-deps.bot`. A Pronoun Bot is a reference such as `~my-runtime.bot` (non-normative example) that resolves to the concrete Bot handle of the caller's currently-active runtime daemon.

The two axes are orthogonal: a pronoun's entity class is a property of the concrete handle produced by resolution, not of the pronoun itself.

4.2. Wire Invariant

A pronoun **MUST NOT** appear in a typed handle field of any structured payload that crosses a protocol trust boundary. Before any of the following operations, a pronoun appearing in a handle field **MUST** be resolved to its concrete handle:

1. Any DNS query or DNS record publication under the `_alter.` prefix or any other DNS label used for `~handle` resolution.
2. Issuance or verification of any capability token whose subject, audience, or resource identifier references a handle.
3. Any cryptographic signing or verification operation over an identity-bearing payload, including but not limited to Accord documents [IDACCORD], Identity-Attributed commit trailers [IDCOMMITTS], attestation statements, and IdentityRank claims.
4. Any MCP tool invocation, HTTP request to a federated peer's `.well-known/alter` endpoint, or other protocol payload in which a handle appears as a typed field.
5. Any write to an append-only identity log, continuous identity-field record, or Signed Tree Head anchor.
6. Any on-chain reference, smart-contract call, or blockchain anchoring operation (e.g. x402 micropayment routing) that encodes a handle as a contract parameter.

This enumeration is non-exhaustive. The governing principle is that a pronoun is a property of the caller's session context and MUST NOT be transmitted to any principal who does not share that session context.

The Wire Invariant applies to handle fields in structured payloads. A handle field is a field typed as "handle" in a payload schema or a positional argument documented as carrying a handle. The invariant does NOT apply to natural-language content -- for example, the body of an alter-to-alter message, a user prompt, or a documentation string -- that incidentally contains a pronoun token. A client MUST NOT rewrite pronoun tokens appearing inside free-text content; such content is conveyed verbatim and a remote reader is responsible for recognising that a pronoun in free text is scoped to the author's session, not the reader's. A client MAY surface a usability warning when free-text content contains pronoun tokens, but MUST NOT transform the content.

This invariant is the architectural rationale for defining pronouns in this document rather than extending the entity-class taxonomy. Pronouns are a property of the client's presentation layer; concrete handles are the property of the protocol.

A conformant implementation MUST reject any payload received over the wire in which a pronoun appears in a typed handle field. This rejection is a category error analogous to the cross-tier rejection defined in [IDCOMMITS].

4.3. Resolution Algorithm

Resolution is performed by the client immediately prior to the first operation listed in the Wire Invariant above. The resolution algorithm proceeds in three phases.

Phase 1 is lexicon lookup. The client maintains a pronoun lexicon mapping reserved textual forms to resolver functions. If the handle's textual form does not match any entry in the loaded lexicon, the handle is treated as absolute and no further resolution is performed.

Phase 2 is context binding. The resolver function is invoked with the current session state. Session state MUST include at minimum the caller's sovereign handle; it MAY include the caller's currently-bound organisation, the caller's currently-held seats, the current thread identifier, and any other context the pronoun's resolver requires.

Phase 3 is substitution. The resolver returns a concrete handle or an error. On success, the pronoun is replaced with the concrete handle in the payload about to be transmitted. On error, the operation that triggered resolution MUST be aborted; the pronoun MUST NOT be transmitted unresolved as a fallback.

A conformant resolver MUST be deterministic within a single session: two successive resolutions of the same pronoun within the same session state MUST yield the same concrete handle.

5. The ~org Pronoun (Wave 1)

5.1. Definition

~org is a Sovereign-class pronoun that resolves to the concrete sovereign handle of the organisation currently bound to the caller's session.

5.2. Session Binding

The session state that binds ~org is established at authentication time (typically by a command of the form `alter login <handle>` or equivalent). The binding is recorded in a session-state artefact local to the caller's environment -- on reference implementations, a file at the path `$HOME/.config/alter/session.json` is RECOMMENDED -- and MUST contain at least the field `current_org` whose value is the concrete sovereign handle of the bound organisation.

A client MAY provide a switch command (e.g. `alter switch <handle>`) that rewrites `current_org` without requiring re-authentication, provided the caller is already a verified member of the target organisation.

5.3. Resolution Failures

~org resolution MUST fail, and the triggering operation MUST be aborted, if any of the following hold:

- * The session-state artefact is absent or unreadable.
- * The `current_org` field is absent, empty, or malformed.
- * The caller is no longer a verified member of the bound organisation (e.g. revocation has occurred since session start).

A resolution failure is distinct from an absolute handle referencing a non-existent organisation. The latter fails at protocol layer; the former fails at client layer before any wire operation.

5.4. Example (non-normative)

A member bound to `~alter` via `alter login ~alter` invokes a messaging command:

```
alter message send --to ~org --body "what is due this week?"
```

The client resolves `~org` to `~alter` by reading `current_org` from session state. The outbound capability-token subject field carries `~alter`; the pronoun `~org` does not appear on the wire.

The same member subsequently invokes `alter switch ~example-co` (an organisation they are a member of via a separate invitation). The next invocation of the same command resolves `~org` to `~example-co` instead. Neither invocation requires the member to retype the concrete handle.

6. Pronoun Lexicon Extensibility

6.1. Per-Client Lexicon

An implementation MAY load a pronoun lexicon from any source trusted by the operator. The reference implementation ships a minimal lexicon containing `~org` only. Additional pronouns MAY be added by the client without coordination with any other party, provided the Wire Invariant is preserved.

6.2. Per-Org Lexicon (Future Work)

A future revision of this document is expected to define a DNS-based mechanism by which an organisation MAY publish a pronoun lexicon as part of its `_alter.<domain>` record, enabling members of that organisation to load org-specific pronouns (e.g. `~weaver` resolving to the current holder of the Weaver seat within that org). Such pronouns MUST obey the Wire Invariant: the DNS record carries the lexicon metadata, not resolved values, and resolution remains a client-side operation.

This extension is out of scope for the present document. The protocol-version constant defined by [MCPDNS] is not incremented.

7. Relative-Path Pronoun Grammar (Appendix, Non-Normative)

This appendix describes a relative-path pronoun grammar as a design surface for future work. The grammar is non-normative in the present document; implementations SHOULD NOT emit these forms in Wave 1.

The grammar borrows from POSIX path semantics [POSIX-TILDE]:

- * `~/<label>` -- resolve `<label>` within the scope of the caller's current organisation (e.g. `~/architect` resolves to the current holder of the Architect seat within `current_org`).
- * `~../<label>` -- resolve `<label>` within the scope of the parent organisation in the caller's current Accord chain (e.g. `~../compliance` resolves to the compliance role of the Accord counter-party).
- * `~/<label>` is reserved and MUST NOT be used; the syntax would be ambiguous with respect to absolute handle forms.

Relative-path pronouns, if standardised in a future revision, obey the Wire Invariant identically to `~org`: resolution happens client-side and the concrete handle is what crosses any wire.

Rationale for filing the grammar as a design surface rather than a Wave-1 normative construct. The relative-path grammar's purpose is **scope-disambiguation across multiple loaded lexicons**, not mere capability. A per-org lexicon (Section 5.2) addressing a pronoun such as `~architect` is sufficient when a member is bound to a single organisation. In a multi-organisational scenario, however, a member whose client loads the lexicons of two organisations both defining `~architect` faces a collision: which organisation's architect does the bare pronoun resolve to? A relative-path form (`~/architect` = architect of current organisation; `~../architect` = architect of the Accord counter-party) makes the scope explicit at the call site, eliminating the ambiguity without requiring the client to adopt a precedence rule that may not match caller intent.

The grammar is deferred to a future revision for three reasons. First, no implementation deployment has yet exercised the multi-lexicon scenario that motivates the grammar. Second, the parent-organisation scope-chain semantics (`~../`) require coordination with the Accord Protocol [IDACCORD] that has not yet been specified in sufficient detail. Third, filing the grammar early without implementation evidence risks committing to path-traversal semantics that a future implementation may reveal to be incorrect. Filing the grammar as a non-normative appendix stakes the design space -- preventing a competing specification from claiming it -- without committing any implementation to it.

8. IANA Considerations

This document introduces no new IANA registry. Pronoun lexicons are client-local artefacts; they are not enumerated by any registry.

A future revision of this document that defines a DNS-based per-org lexicon mechanism (see Section 5.2) will require a lexicon TXT record sub-field to be registered in the `_alter.` record schema defined by [MCPDNS]. That registration is out of scope for the present document.

9. Security Considerations

9.1. Pronoun Spoofing

An attacker capable of writing to the caller's session-state artefact (e.g. the `session.json` file under `$HOME/.config/alter/`) MAY cause `~org` to resolve to an attacker-controlled handle. This attack is a local privilege-escalation class and is outside the scope of this specification; implementations SHOULD protect session-state artefacts with the same filesystem permissions applied to other credential artefacts.

9.2. Pronoun Leakage

An implementation that mistakenly transmits a pronoun on the wire (e.g. by failing to resolve before token issuance) creates two hazards: (1) the receiving peer rejects the payload as specified in Section 3.2, exposing the caller's misconfiguration; (2) the pronoun's textual form may leak information about the caller's client topology. Conformant implementations MUST resolve pronouns before any wire operation; test suites SHOULD include explicit coverage of Wire Invariant violations.

9.3. Binding Freshness

`~org` resolution depends on session state that MAY have been written arbitrarily long before the current invocation. A client MAY cache the membership proof; if the caller has been revoked from the bound organisation since session start, a stale binding could cause a resolution that the organisation's membership system would reject at protocol layer. Implementations SHOULD re-verify membership freshness at resolution time when the resolved handle is about to be used for a high-privilege operation (e.g. Accord signature).

9.4. Cross-Tenant Confusion

A member of two organisations who invokes a command against `~org` must be unambiguous about which organisation is bound. Clients SHOULD surface the currently-bound organisation in user-visible status (a status line, prompt decoration, or equivalent) so that the member cannot mistakenly send a message intended for one organisation to the other.

10. Implementation Status

This section is to be removed before final publication.

A reference implementation of this specification is planned in the ALTER open-source codebase; no implementation of the pronoun resolver exists at the time of this -00 revision. Implementation is expected to be distributed across three surfaces:

1. The alter-cli command-line interface will maintain the current_org field in a session.json file under \$HOME/.config/alter/ via alter login and alter switch.
2. The personal alter MCP bridge (mcp-alter) will load the minimal Wave-1 lexicon and perform resolution prior to any outbound MCP call.
3. Client-side slash-command wrappers (e.g. the Claude Code /msg hook) will consume the resolver via the personal alter bridge and MUST NOT implement their own pronoun grammar.

11. References

11.1. 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>>.
- [MCPDNS] Morrison, B., "Discovery of Model Context Protocol Servers via DNS TXT Records", 2026, <<https://datatracker.ietf.org/doc/draft-morrison-mcp-dns-discovery/>>.
- [IDACCORD] Morrison, B., "Identity Accord Protocol", 2026, <<https://datatracker.ietf.org/doc/draft-morrison-identity-accord/>>.
- [IDCOMMITTS] Morrison, B., "Identity-Attributed Git Commits via Tier-Structured Trailers", 2026, <<https://datatracker.ietf.org/doc/draft-morrison-identity-attributed-commits/>>.

11.2. Informative References

[POSIX-TILDE]

"IEEE Std 1003.1-2017, Shell Command Language, Section 2.6.1 Tilde Expansion", 2017, <<https://pubs.opengroup.org/onlinepubs/9699919799/>>.

[GIT-REVISIONS]

"gitrevisions(7) -- specifying revisions and ranges for Git", 2024, <<https://git-scm.com/docs/gitrevisions>>.

[CSS-SELECTORS-4]

"Selectors Level 4 (:root, :host)", 2022, <<https://www.w3.org/TR/selectors-4/>>.

[K8S-SUBJECTACCESSREVIEW]

"Kubernetes Authorization -- SelfSubjectAccessReview", 2024, <<https://kubernetes.io/docs/reference/access-authn-authz/authorization/>>.

Appendix A. Acknowledgements

This document grew out of a design conversation identifying that the three-tier handle taxonomy of [IDCOMMITTS] answered `_what` kind of thing_ a handle binds to but did not answer `_how` the handle is referenced_. The realisation that pronouns are a reference axis orthogonal to entity class -- rather than a fourth tier -- 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