RFC-0509

Receiving-Boundary Reliance Grammar

Proof Attribute Presentment, Receiver Requirement Expression, Boundary Outcome Signaling, and Receiver-Side Conformance for HTTP-Based Consequential Actions

Field	Controlled value
Document status	Controlled Public Draft v0.8 / Public-facing candidate
Distribution	Controlled until release gate; intended to become public after counsel and strategy approval
Date	July 2026
Authoring basis	Derived from RFC-0509 Public Draft v0.7.1 and updated against the v0.8 Candidate Synthesis and Promotion Checklist
Scope	Neutral public boundary grammar only; no private reliance mechanism, no protected architecture, no universal proof object
Controlling premise	Better proof is supply. Receiver need is demand. RFC-0509 is the grammar that lets them meet at the boundary.
v0.8 drafting focus	Receiver-side outcome vocabulary hardening, object-verifiable vs attested attributes, receiver-signed local outcome records, receiver-side conformance, and interested-receiver posture.
Master test	Does the grammar preserve receiver-side judgment while making proof attributes, receiver requirements, local outcomes, and conformance boundaries explicit enough to support fit, auditability, and economics?

Contents

1. Abstract

RFC-0509 defines a public, neutral HTTP-compatible grammar for consequential actions where proof created by one system must be evaluated by another independently governed receiving domain before local reliance can attach.

The grammar does not standardize one proof object. It does not define a badge, certification mark, universal pass, legal-effect determination, or proprietary reliance decision. It lets proof suppliers present proof-side attributes and lets receiving domains express receiver-side requirements so that fit, boundary outcome, and local economic effect can be determined by the receiver under its own rulebook, current state, liability posture, and consequence model.

Version v0.8 hardens the grammar in five ways: it distinguishes object-verifiable proof attributes from attested custody-posture attributes; distinguishes proof-side presentment from receiver-signed local outcome records; narrows the normative boundary-outcome vocabulary to ACCEPT, HOLD, NARROW, and REFUSE; adds a receiver-side reliance-formation conformance test; and recognizes interested-receiver or self-confirmation risk as a receiver-posture consideration without creating an independence-certification regime.

RFC-0509 preserves receiver-side judgment. It makes proof attributes and receiver requirements legible at the boundary. The receiving domain decides whether the proof fits this action, under this rulebook, now.

2. Status of This Memo

This is a controlled public draft candidate. It is written in RFC style for clarity, review discipline, and eventual public circulation, but it is not an IETF Internet-Draft, standards submission, legal opinion, product specification, or patent disclosure. Version v0.8 incorporates the controlled candidate hardening pass for receiver-side outcomes, conformance, custody posture, argument commitments, and receiver-interest posture.

The document is intended to be public-facing after a separate release gate. It intentionally excludes private formation, enforcement, reliance-object, consequence-attachment, economic implementation, and protected architecture mechanics.

The working label RFC-0509 is a project and grammar identifier. It does not imply registration by any standards body.

2.1 Status, Reference Use, and Profile Claims

RFC-0509 is published as a public receiving-boundary grammar for HTTP-based presentment, evaluation, and receiver-local outcome signaling.

Public reference to RFC-0509 is permitted for discussion, implementation analysis, interoperability research, proof-side profile mapping, and receiver-boundary design.

A party MAY describe its own work as “mapped to,” “informed by,” or “designed for interoperability with” RFC-0509, provided the statement is accurate, does not imply endorsement, certification, commercial alignment, receiver reliance, or profile approval by 3PMobile, and does not represent RFC-0509 parsing or field usage as sufficient to create receiver-side reliance.

A party MUST NOT represent any product, proof object, profile, platform, implementation, or commercial offering as “RFC-0509 certified,” “3PMobile approved,” “officially conformant,” “an approved RFC-0509 profile,” part of a joint architecture, commercially aligned with 3PMobile, or granted access to any receiving-boundary consequence layer without prior written approval from 3PMobile.

A party claiming technical compatibility or self-assessed conformance MUST clearly distinguish such claim from 3PMobile certification, endorsement, commercial approval, profile approval, or receiver-side reliance. Receiver-side conformance requires satisfaction of the receiver-side conformance requirements in Section 28 and MUST NOT be

inferred from field-level parsing, proof-side profile mapping, vendor-side evaluation, or issuance of a signed record alone.

RFC-0509 defines public grammar only. Receiving institutions remain responsible for their own contextual profiles, rulebooks, reliance decisions, implementation choices, consequence surfaces, compliance obligations, and local outcome formation.

3. Terminology and Conventions

The key words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL are used in the conventional RFC sense for drafting precision.

HTTP field names are shown lowercase. In HTTP/2 and HTTP/3, field names are lowercase on the wire; field names are otherwise case-insensitive under HTTP semantics.

Examples are illustrative. They do not prescribe a proof format, trust framework, legal result, protected implementation method, commercial deployment model, or receiver-internal governance process.

4. Problem Statement

AI and agentic systems increasingly generate consequential actions and associated proof objects: logs, attestations, receipts, credentials, traces, approvals, policy checks, model records, runtime records, compliance artifacts, execution evidence, and other evidence packages. Those proof objects will not converge into one universal format.

A proof object can explain what happened upstream. It can show that a process ran, a control fired, a credential existed, an action was attempted, or a state was recorded. But when proof arrives in another independently governed system, the receiver still has to decide whether it can rely on that proof here, for this action, under this rulebook, now.

The missing layer is not a better proof object. The missing layer is a neutral receiving-boundary grammar that lets proof attributes and receiver requirements become explicit enough for the receiving domain to determine fit and produce a local boundary outcome.

Without such grammar, every receiving domain must implement bespoke interpretation logic for every upstream proof engine, governance platform, compliance framework, credential system, payment mechanism, agent runtime, attestation system, or audit package. That does not scale in an agentic environment.

A second problem appears once RFC-0509 vocabulary becomes useful: vendors may claim compatibility while substituting vendor-controlled criteria for receiver-controlled reliance formation. RFC-0509 therefore must distinguish field-level compatibility from receiver-side conformance.

5. Design Goals

- Use existing HTTP semantics and transport behavior. RFC-0509 does not define a new transport protocol.
- Standardize the boundary, not the proof object.
- Allow receiving domains to express receiver-side requirements before or during consequential processing.
- Allow proof suppliers to present proof-side attributes clearly enough for receiver-side evaluation.
- Support attribute-level expression rather than treating proof as merely present or absent.
- Preserve receiver sovereignty and local reliance judgment.
- Make fit thresholds receiver-declared, not universal constants.
- Use a minimal normative boundary-outcome vocabulary: ACCEPT, HOLD, NARROW, and REFUSE.

- Treat escalation, manual review, reconstruction, override, and notification as receiver-internal process states, not normative boundary outcomes.
- Distinguish object-verifiable proof attributes from custody-posture attributes requiring separate attestation.
- Distinguish proof-side presentment from receiver-signed local outcome records.
- Define receiver-side conformance tests sufficient to prevent badge capture and vendor-rule capture.
- Support economic signaling without becoming an economic calculator or valuation engine.
- Keep protected formation, enforcement, reliance-object, and consequence-attachment mechanics outside the public grammar.

6. Non-Goals

- This document does not define one universal proof object.
- This document does not certify proof objects or create a badge.
- This document does not make evidence actionable for a receiver by itself.
- This document does not decide reliance, liability, settlement, economic value, or consequence attachment.
- This document does not define protected reliance formation or private implementation architecture.
- This document does not define a mandatory cryptographic scheme, identity system, blockchain, distributed ledger, consensus mechanism, registry, or hardware attestation mechanism.
- This document does not require a receiving domain to accept any proof object, attribute set, certificate, receipt, attestation, or compliance packet.
- This document does not define universal fit bands, thresholds, outcome rules, risk prices, or internal escalation routes.
- This document does not create new HTTP status codes in this version.
- This document does not define an independence-certification regime for receivers.
- This document does not require a receiver-signed local outcome record to create transitive reliance for other receivers.

7. Public Architecture Boundary

RFC-0509 is a neutral public grammar. It exists to let proof supply and receiver demand meet at a receiving boundary. It does not own the receiver decision. It does not own the proof object. It does not own the protected implementation used by any party.

7.1 Neutrality and Receiver Sovereignty

RFC-0509 is neutral grammar for expressing proof-side attributes, receiver-side requirements, fit results, boundary outcomes, local reliance indicators, and receiver-side outcome records. It MUST NOT be interpreted as a governance authority that overrides the receiving domain's rulebook, risk posture, state, consequence model, or internal operating model.

A receiver MAY declare, reference, withhold, update, or scope its requirements according to local policy. A proof supplier MAY present attributes, evidence references, attestations, certificates, receipts, traces, or other proof materials, but those materials remain inputs to receiver-side evaluation. RFC-0509 does not convert supplier presentation into receiver reliance.

Receiver-side judgment is not a defect in the grammar. It is the point of the grammar.

7.2 Anti-Badge, Anti-Certification, and Anti-Capture Rule

RFC-0509 compatibility MUST NOT be represented as a universal badge, certification mark, reliance pass, or substitute for receiver-side judgment. A badge, certificate, audit packet, compliance packet, concept dictionary, or

verified proof bundle MAY improve legibility, but it cannot make evidence actionable for a receiver without receiver-local acceptance.

A conforming deployment MUST distinguish between proof quality and receiver fit. Proof quality is supply. Receiver requirement is demand. Fit exists only when the receiver evaluates the presented attributes against the receiver's own declared or referenced requirements for the action, rulebook, context, and time at issue.

A receiver-side implementation MUST NOT use RFC-0509 vocabulary to replace receiver-controlled requirements with vendor-controlled criteria while marketing the result as receiver-side reliance formation.

Layer	RFC-0509 public position
Proof formation	Out of scope. Proof vendors, governance engines, credential systems, payment systems, attestation systems, and other upstream engines may create different proof objects.
Proof-side attributes	In scope as presentable attributes or references. The grammar can make attributes legible without standardizing the proof object.
Receiver-side requirements	In scope as receiver-declared requirement expressions. Requirements are local, contextual, and action-specific.
Fit evaluation	In scope as vocabulary and signaling. RFC-0509 does not compute universal fit or decide reliance.
Boundary outcome	In scope as receiver-declared result signaling using a minimal normative boundary set.
Receiver-signed local outcome records	In scope as a record class, where receiver requirements, evaluated attributes, fit, outcome, scope, and reason codes are recorded as receiver-side evidence.
Receiver-side conformance	In scope as observable tests for preserving receiver-side reliance formation. Not a certification program in this draft.
Commercial implementation	Out of scope. Implementations may be commercial, open, internal, or proprietary.
Protected mechanics	Out of scope. Private mechanisms remain outside the public grammar.

8. Conceptual Model: Proof Supply Meets Receiver Demand

RFC-0509 is organized around one economic and technical chain: proof attributes -> receiver requirements -> fit -> boundary outcome -> local reliance record -> economic effect. The grammar supports the chain but does not decide the values inside it.

Stage	Question	RFC-0509 role
Proof attributes	What does the proof claim or demonstrate?	Make proof-side attributes expressible and inspectable.
Receiver requirements	What does the receiver need for this action, under this rulebook, now?	Make receiver-side requirements explicit and machine-readable or machine-referenceable.
Fit	Do the proof attributes satisfy the receiver requirements sufficiently for a local outcome?	Support fit expression and fit signaling without imposing universal thresholds.
Boundary outcome	What did the receiver decide at the proof boundary?	Signal ACCEPT, HOLD, NARROW, or REFUSE as receiver-local boundary outcomes.
Local outcome record	What receiver-side evidence preserves the decision?	Support receiver-signed or receiver-authorized records of requirements applied, attributes evaluated, fit, outcome, scope, and reason codes.
Economic effect	What cost, risk, delay, reserve, review, or action path changes because of the outcome?	Support downstream economic computation without becoming the calculator or authority for that computation.

9. Terminology

Term	Definition
Proof object	An upstream artifact, record, receipt, credential, trace, attestation, compliance packet, model record, policy record, audit trail, execution evidence package, or other evidence package that may explain or support what happened upstream.
Proof supplier	A system, vendor, governance engine, model runtime, compliance platform, enterprise workflow, credential issuer, payment system, attestation issuer, or other source that creates or presents proof-side attributes.
Proof-side attributes	Machine-readable or machine-referenceable characteristics of a proof object, such as trace completeness, freshness, replay protection, custody posture, identity claims, authority claims, issuance time, evidence links, policy coverage, auditability, verification class, or assurance grade.
Verification class	A classification of whether a proof-side attribute is object-verifiable from the presented object and public verification material, attested by a separate artifact, declared by a party, or otherwise dependent on external context.
Receiving domain	The independently governed system, organization, workflow, authority surface, or resource owner that receives proof or a request and determines whether it can act under local rules.
Receiver-side requirements	The receiver-declared conditions, attribute needs, thresholds, action context, risk posture, rulebook, state requirements, consequence model, and outcome options relevant to a local reliance decision.
Receiver need	The local requirement that creates demand for proof attributes. Receiver need is specific to this receiver, this action, this rulebook, and this moment.
Boundary	The HTTP-mediated or enterprise-mediated point where a proof object is encountered by a receiving domain for possible local use.
Fit	The receiver-local evaluation of whether presented proof attributes satisfy receiver-side requirements sufficiently to support a boundary outcome.
Boundary outcome	The receiver's local result after evaluating fit at the proof boundary. The candidate normative outcome set is ACCEPT, HOLD, NARROW, and REFUSE.
Receiver-internal process state	A process or workflow that may follow a boundary outcome, such as escalation, manual review, reconstruction, override, legal review, regulatory notification, or board approval. Such states are not normative boundary outcomes.
Local reliance	A receiver-local decision that proof attributes fit enough for the receiver to proceed under its own rules. Local reliance is not transitive and does not bind other domains.
Receiver-signed local outcome record	A receiver-authored, receiver-authorized, receiver-signed, or receiver-custodied record that preserves receiver requirements applied, proof attributes evaluated, fit result, boundary outcome, reason codes, scope, limitations, and consequence posture as receiver-side evidence.
Presentment grammar	The neutral HTTP-compatible vocabulary and exchange pattern by which proof-side attributes and receiver-side requirements are made legible at the boundary.
Standing Presentment	A technical operation within RFC-0509 by which a proof object or proof-attribute package is presented as a claim for receiver-local evaluation. The term does not imply automatic acceptance or reliance.
Interested receiver	A receiver that may have sponsored, funded, shaped, selected, benefited from, or otherwise had an interest in the proof being acceptable. Interested-receiver posture is a trust consideration, not an automatic invalidation rule.

10. Core Principles

- Neutral grammar; local decision: RFC-0509 makes attributes and requirements legible. The receiver decides fit and outcome.
- Proof is supply; receiver need is demand: Proof attributes have value only when they match a receiver's reliance requirement.
- Presentment is not acceptance: A proof object may be well-formed, signed, attested, or verified and still fail receiver-side requirements.

- Fit thresholds are declarations, not constants: RFC-0509 supports threshold expression. It does not define universal threshold values.
- Boundary outcomes are boundary states: ACCEPT, HOLD, NARROW, and REFUSE describe the receiver's decision at the proof boundary.
- Escalation is not a boundary outcome: receiver-internal governance remains receiver sovereign.
- A badge is not reliance: a badge, certificate, audit packet, compliance packet, or concept dictionary may improve legibility but cannot eliminate the receiver's local decision.
- Local reliance is non-transitive: acceptance by one receiver does not require acceptance by another receiver.
- Object verification and custody posture are different: a signature may prove what was signed and by which key; it does not by itself prove where the key resided or how custody was separated.
- Economics follow receiver outcome: proof quality becomes economically meaningful only when it changes receiver-side cost, risk, capacity, or action eligibility.
- Conformance is not parsing: receiver-side conformance requires preservation of receiver-controlled requirements and receiver-local outcome formation.

RFC-0509 may expose data useful for economic analysis, including fit status, boundary outcome, reason codes, review status references, reduced reconstruction, or action eligibility. It does not compute value, price risk, allocate liability, or guarantee savings. Economic effect remains an observed or receiver-modeled consequence of the receiver's local outcome.

11. Protocol Overview

1. A requester attempts or prepares a consequential action, or presents proof related to such action.
2. The receiving domain determines that receiver-side requirements must be satisfied before the proof can support local reliance.
3. The receiving domain advertises or returns receiver requirements, acceptable presentment types, requirement references, verification classes, or supported outcome vocabulary.
4. The requester presents a proof object, proof attribute package, or reference to proof-side attributes.
5. The receiving domain evaluates fit locally against receiver requirements and current context.
6. The receiving domain returns an HTTP response and RFC-0509 boundary outcome fields.
7. The receiving domain MAY issue, sign, authorize, or custody a local outcome record preserving the receiver-side decision, scope, reason codes, and consequence posture.
8. Any downstream economic or operational effect remains local to the receiver and outside this grammar.

12. Receiver Requirements

Receiver requirements are receiver-declared. They are not universal constants. A receiver MAY express requirements inline, by reference, by profile, by policy URI, through negotiation, or through an application-specific mechanism.

A receiver requirement expression SHOULD identify the requested action or processing context, required or preferred proof attributes, required freshness or time windows, acceptable evidence classes, verification class requirements, privacy constraints, outcome options, and any reason codes that may be returned.

Receiver requirements MAY include qualitative, quantitative, temporal, jurisdictional, risk, privacy, evidence, traceability, custody, freshness, verification, or operational conditions. Any such threshold is receiver-declared and context-specific. RFC-0509 provides a way to express or reference the requirement; it does not set the threshold.

A receiver MAY publish exact requirements, provide a reference profile, disclose only coarse requirement categories, or withhold sensitive thresholds where disclosure would create gaming, security, privacy, or fraud risk.

```
HTTP/2 428 Precondition Required
receiver-requirements: <requirement-reference-or-envelope>
```

```
accept-proof: application/rfc0509-presentment+json
rfc0509-outcomes: ACCEPT, HOLD, NARROW, REFUSE
```

The names shown above are controlled draft field names. They are not IANA registrations and are subject to registration review before public standardization.

13. Proof Attribute Presentment

A proof supplier or requester MAY present proof-side attributes using an envelope, a reference, an embedded body, a detached body, or a negotiated media type. RFC-0509 does not require the proof object itself to be serialized in a particular format.

A presentment may contain attributes about trace completeness, freshness, custody posture, independent replayability, identity, authority, integrity, receipts, attestations, evidence links, argument commitments, or other proof characteristics. These attributes are inputs to receiver-side evaluation, not reliance decisions by themselves.

Proof attribute vocabularies are intentionally extensible. RFC-0509 does not require a closed taxonomy and does not prefer one proof vendor, credential system, attestation framework, audit package, or governance engine. Heterogeneous proof suppliers MAY present different attributes if the receiver can inspect them through the grammar or through a referenced profile.

A proof attribute SHOULD be presented as an inspectable input, not as a conclusion that the receiver must accept. The receiver determines whether the attribute is relevant, sufficient, stale, incomplete, excessive, mismatched, or otherwise unsuitable for the requested boundary outcome.

```
POST /boundary/evaluate HTTP/2
content-type: application/rfc0509-presentment+json
proof-presentment: <presentment-reference-or-envelope>
```

14. Attribute Verification Classes

Version v0.8 distinguishes attributes that are verifiable from the proof object itself from attributes that require separate attestation or deployment-context evidence. This distinction prevents evidence objects from silently carrying more assurance than they can self-prove.

Verification class	Description	Receiver-side implication
OBJECT_VERIFIABLE	The receiver can verify the attribute from the presented object, signatures, hashes, public keys, included references, or deterministic recomputation without trusting a vendor narrative.	The receiver may evaluate the attribute as directly inspectable proof-side input, subject to local requirements.
ATTESTED_CUSTODY_POSTURE	The attribute concerns where or how the proof was produced, how keys were held, whether custody separation existed, or whether a boundary layer was actually used. It requires a separate attestation artifact or deployment evidence.	The receiver decides whether the attestation class, attestor, scope, validity, and revocation posture satisfy local requirements.
DECLARED	The attribute is declared by a party but not independently verifiable from the proof object or attestation.	The receiver may treat it as a claim, not as independently verified evidence.
RECEIVER_EVALUATED	The attribute reflects the receiver's own evaluation result, such as fit status, boundary outcome, reason code, or local scope decision.	The attribute belongs in a receiver-side local outcome record, not in proof-side presentment alone.

An attestation artifact SHOULD bind to the evidence it supports, for example by referencing the signing-key fingerprint or evidence-object digest, so that an attestation for one deployment cannot be silently paired with evidence from another deployment.

A receiver MAY require a specific attestation trust class, such as third-party attestation or hardware-rooted attestation, but RFC-0509 does not define or certify those trust classes in this draft.

15. Argument Commitments and Disclosure Posture

A proof object may present an argument commitment instead of raw arguments. The commitment may allow a receiver to verify that a disclosed candidate argument set matches the original input without embedding sensitive arguments directly in the proof object.

RFC-0509 distinguishes whole-payload argument commitments from selective-disclosure commitments because receiver fit may depend on whether the receiver can verify all arguments, only disclosed candidate arguments, or individual fields.

Commitment posture	Meaning	Receiver-side consideration
WHOLE_PAYLOAD_COMMITMENT	A single commitment covers the full canonical argument payload. A receiver verifies by recomputing the commitment over a candidate full payload.	Useful when the receiver has or may receive the full candidate payload. Not sufficient where the receiver needs to verify one field while hiding the rest.
FIELD_LEVEL_COMMITMENT	Individual fields or field groups are committed separately.	May support selective verification, but requires a specified construction and disclosure rules.
SELECTIVE_DISCLOSURE_COMMITMENT	The proof supplier can reveal and prove selected fields while hiding others, for example through a Merkle-style structure or another selective-disclosure mechanism.	May satisfy receiver privacy requirements where full-payload disclosure is unacceptable.
OPAQUE_DIGEST	A digest is supplied without enough canonicalization or disclosure posture to recompute or inspect it.	Receiver may treat it as weak or insufficient unless other evidence satisfies the requirement.

This draft does not require any particular commitment construction. Receivers SHOULD state the disclosure posture they require when argument content is material to fit.

16. Receiver-Signed Local Outcome Records

A receiver-signed local outcome record is distinct from proof-side presentment. Proof-side presentment says what attributes are being offered for receiver evaluation. A receiver-side local outcome record says what the receiver evaluated, what requirement it applied, what fit result it declared, what boundary outcome it formed, and what scope or limitation it attached to that outcome.

A receiver-side local outcome record MAY be signed, authorized, custodied, or otherwise made inspectable as the receiver's decision. RFC-0509 does not require one universal record format. The record class is defined by its receiver-side function, not by a vendor-specific schema.

Field class	Purpose	Notes
receiver_requirement_ref	Identifies the receiver requirement, profile, rulebook, or requirement reference applied.	May be inline, referenced, partially withheld, or receiver-private.
proof_attributes_evaluated	Identifies which presented attributes the receiver actually evaluated.	Prevents records from listing supplied data without showing what mattered to fit.

fit-status	Signals receiver-local fit result, if disclosed.	Fit remains receiver-local and non-transitive.
boundary-outcome	Signals ACCEPT, HOLD, NARROW, or REFUSE.	Boundary state only; not an internal governance instruction.
outcome-reason	Preserves receiver-defined reason code or reference.	Supports audit, dispute, downstream review, and economics.
scope-of-reliance	States the action, scope, time window, limitation, or consequence for which the receiver outcome applies.	May be declarative, machine-enforced, lifecycle-governed, or some combination depending on implementation.
downstream-reuse	States whether and how the record may be reused downstream, if disclosed.	Does not create transitive reliance unless another receiver independently accepts it.
lifecycle-status-ref	References dispute, correction, supersession, withdrawal, or other lifecycle status if supported.	Lifecycle mechanics are implementation-specific; RFC-0509 may reference but does not define them.

A signed local outcome record may contain fields that are machine-enforced, declarative, lifecycle-linked, or merely inspectable. Implementations SHOULD make this distinction clear. A signed declaration of scope or downstream reuse MUST NOT be treated as runtime enforcement unless the implementation actually enforces it.

A receiver-side local outcome record MUST NOT be represented as a universal reliance pass, proof-side certification, legal-effect determination, or binding outcome for other receiving domains.

17. Attribute Fit and Boundary Outcomes

Fit is the receiver-local relationship between proof-side attributes and receiver-side requirements. RFC-0509 supports expression of fit results but does not impose universal fit bands or economic outcomes.

Fit category	Meaning	Typical receiver boundary outcome
low fit	Proof attributes do not satisfy key receiver requirements.	REFUSE or HOLD
partial fit	Proof attributes satisfy some receiver requirements but leave residual risk, missing attributes, or action constraints.	HOLD or NARROW
high fit	Proof attributes satisfy the receiver's declared needs for the action context.	ACCEPT

The labels low fit, partial fit, and high fit are illustrative. A conforming implementation MAY use a different scoring model or vocabulary if it preserves receiver-local judgment and does not turn RFC-0509 into a universal pass or badge.

The normative boundary outcome vocabulary for this draft is: ACCEPT, HOLD, NARROW, and REFUSE.

Outcome	Controlled meaning	Boundary effect
ACCEPT	The presented proof satisfies the receiver's qualification requirements for the requested action or processing context.	The receiver may proceed under its own governance framework.
HOLD	The proof is not refused, but the receiver cannot resolve qualification at this time.	Processing is suspended, delayed, or paused pending receiver-local resolution.
NARROW	The proof satisfies receiver requirements only for a restricted subset of the requested scope.	The receiver may permit partial, capped, limited, or narrowed reliance, as locally defined.
REFUSE	The proof does not satisfy receiver qualification requirements.	No local reliance is granted for the requested action or processing context.

Boundary outcomes are receiver decisions, not commands imposed by RFC-0509. Escalation, manual review, reconstruction, board approval, regulatory notification, legal review, override, or workflow routing may follow a boundary outcome, but those are receiver-internal process states, not normative RFC-0509 boundary outcomes.

Fit does not travel as universal reliance. A high-fit or ACCEPT outcome in one receiving domain MAY be irrelevant, insufficient, or unacceptable in another receiving domain.

18. HTTP Status Code Mapping

HTTP status codes describe the HTTP request outcome. RFC-0509 fields describe the boundary outcome. A successful HTTP response MUST NOT be interpreted as proof that local reliance was granted unless the receiving domain also provides a receiver-local outcome signal.

RFC-0509 uses existing HTTP status-code semantics rather than defining a replacement protocol. HTTP status codes signal request and boundary-processing states; RFC-0509 fields, envelopes, and media types make proof attributes, receiver requirements, fit status, and receiver-local boundary outcomes legible. The status code alone never carries the proof object and never decides reliance.

HTTP status	RFC-0509 meaning	Notes
428 Precondition Required	Receiver requirements must be satisfied before local reliance or consequential processing can proceed.	Used when the receiver needs proof attributes or a presentment before evaluating the action.
400 Bad Request	Malformed presentment, requirement envelope, or request.	Syntax or parse failure.
401 / 403	Ordinary authentication or authorization failure.	Separate from proof fit and receiver-local reliance.
409 Conflict	Receiver detects conflicting state, conflicting claims, incompatible lifecycle state, or inconsistent fit conditions.	The receiver may return conflict reason codes.
412 Precondition Failed	Presented proof attributes failed receiver-side requirements for the requested action.	May be accompanied by REFUSE or HOLD.
202 Accepted	Receiver has accepted the request for asynchronous evaluation or HOLD status.	Does not imply local reliance unless the outcome field says so.
200 OK / 204 No Content	Request completed; local outcome may be ACCEPT, NARROW, HOLD, REFUSE, or another receiver-declared result if a profile extends the vocabulary.	HTTP success and reliance outcome remain distinct.

19. Field Names and Syntax

This draft defines conceptual field names for discussion. Future versions may align them with HTTP Structured Fields, ABNF, media type registration rules, and IANA review.

Field	Direction	Purpose
receiver-requirements	receiver -> requester	Identifies receiver-side requirements or a reference to them.
accept-proof	receiver -> requester	Identifies acceptable presentment envelope types or proof-attribute media types.
proof-presentment	requester -> receiver	Carries or references the presented proof attribute package.
proof-attributes	requester -> receiver	Optional shorthand field for proof-side attribute reference or digest.
verification-class	requester -> receiver	Optional indication of whether an attribute is object-verifiable, attested, declared, or otherwise dependent on external context.
attestation-ref	requester -> receiver	Optional reference to an attestation artifact supporting custody-posture or deployment-context claims.
argument-commitment	requester -> receiver	Optional reference or value describing argument commitment and disclosure posture.

fit-status	receiver -> requester	Signals receiver-local fit result, if disclosed.
boundary-outcome	receiver -> requester	Signals receiver-local outcome: ACCEPT, HOLD, NARROW, or REFUSE.
local-reliance	receiver -> requester	Signals whether the receiver has formed local reliance for the requested action.
outcome-reason	receiver -> requester	Optional receiver-defined reason code or reference.
requirement-profile	receiver -> requester	Optional reference to a receiver-declared requirement profile.
local-outcome-record	receiver -> requester	Optional reference to a receiver-side local outcome record.
process-status	receiver -> requester	Optional non-normative receiver-internal process state, such as manual-review, escalation, reconstruction, or pending.
economics-ref	receiver -> requester	Optional reference to receiver-local economic rules or results. RFC-0509 does not compute them.
receiver-interest-posture	receiver -> requester or requester -> receiver	Optional disclosure, request, or flag concerning receiver interest, sponsorship, conflict, or self-confirmation posture.

20. Presentment Envelopes and JSON Posture

JSON MAY be used for presentment envelopes where appropriate. JSON is not mandatory, and a JSON envelope is not necessarily the proof object itself. RFC-0509 treats proof objects as opaque unless their attributes are voluntarily exposed or referenced for receiver-side evaluation.

```
{
  "presentment_id": "rp_123",
  "proof_ref": "urn:example:proof:abc",
  "receiver_requirement_ref": "https://receiver.example/requirements/pa-v1",
  "request_binding": {
    "method": "POST",
    "target": "/boundary/evaluate",
    "body_digest": "sha-256:..."
  },
  "proof_attributes": {
    "trace_completeness": "complete",
    "freshness": { "issued_at": "2026-06-15T15:00:00Z", "max_age_seconds": 300 },
    "verification_class": "OBJECT_VERIFIABLE",
    "custody_posture": {
      "attribute": "custody_separation",
      "verification_class": "ATTESTED_CUSTODY_POSTURE",
      "attestation_ref": "urn:example:attestation:1"
    },
    "argument_commitment": {
      "type": "WHOLE_PAYLOAD_COMMITMENT",
      "digest": "sha256:..."
    },
    "evidence_refs": ["urn:example:evidence:1"]
  },
  "privacy": {
    "minimized": true,
    "redaction_profile": "receiver-pa-basic"
  }
}
```

The example is illustrative and non-normative. It does not define a protected implementation or a universal schema.

21. Caching and Intermediary Behavior

RFC-0509 presentments and receiver outcomes may contain sensitive context, proof attributes, receiver requirements, interest posture, and local outcome data. Implementations SHOULD use conservative cache controls and SHOULD avoid intermediary exposure unless explicitly intended.

- Receivers SHOULD mark requirement and outcome responses no-store unless the response is expressly intended to be reusable.
- Presentment envelopes SHOULD be minimized and MAY use references rather than embedded sensitive data.
- Intermediaries MUST NOT treat HTTP success as local reliance.
- A receiver MAY choose not to disclose all requirement thresholds or fit details if doing so would create gaming, security, privacy, or fraud risk.
- Receiver-signed local outcome records SHOULD disclose lifecycle status or status references where downstream inspection depends on current effective status.

22. Idempotency, Replay, and Retry Behavior

Receivers and requesters SHOULD distinguish retry-safe presentment from repeated reliance attempts. A requester may retry delivery of a presentment envelope, but a receiver MUST NOT treat repeated presentment as repeated acceptance.

- Presentments SHOULD be bound to request context, target action, receiver, and time window where consequence matters.
- Receivers SHOULD detect duplicate presentments and may return prior outcome references.
- A retry of a held, narrowed, or refused outcome MUST NOT be treated as a new local reliance decision unless the receiver's local rules permit reevaluation.
- A receiver-side local outcome record SHOULD distinguish original issuance from correction, dispute, supersession, or withdrawal if lifecycle status is supported.

23. Synchronous and Asynchronous Processing

A receiver MAY evaluate fit synchronously or asynchronously. Asynchronous evaluation is useful where receiver requirements involve human review, external policy lookup, fraud analysis, risk pricing, missing attributes, delayed data availability, lifecycle checks, or internal governance processes.

```
HTTP/2 202 Accepted
boundary-outcome: HOLD
fit-status: partial
local-reliance: false
outcome-reason: additional-review-required
process-status: manual-review
status-ref: https://receiver.example/rfc0509/status/789
```

The process-status field is non-normative. It describes receiver-internal process state if the receiver chooses to disclose it. It is not a boundary outcome.

24. Examples

24.1 Receiver requires proof attributes

```
HTTP/2 428 Precondition Required
receiver-requirements: https://payer.example/requirements/prior-auth-v3
accept-proof: application/rfc0509-presentment+json
rfc0509-outcomes: ACCEPT, HOLD, NARROW, REFUSE
```

24.2 Proof attributes are presented

```
POST /prior-auth/evaluate HTTP/2
content-type: application/rfc0509-presentment+json
proof-presentment: urn:proof-package:provider:12345
```

24.3 Receiver accepts and forms local reliance

```
HTTP/2 200 OK
fit-status: high
boundary-outcome: ACCEPT
local-reliance: true
outcome-reason: receiver-requirements-satisfied
local-outcome-record: urn:receiver:outcome:abc
```

24.4 Receiver narrows the action

```
HTTP/2 200 OK
fit-status: partial
boundary-outcome: NARROW
local-reliance: true
outcome-reason: proof-supports-limited-scope
local-outcome-record: urn:receiver:outcome:def
```

24.5 Receiver refuses

```
HTTP/2 412 Precondition Failed
fit-status: low
boundary-outcome: REFUSE
local-reliance: false
outcome-reason: freshness-requirement-not-met
```

24.6 Receiver holds and discloses internal process status

```
HTTP/2 202 Accepted
fit-status: indeterminate
boundary-outcome: HOLD
local-reliance: false
outcome-reason: custody-attestation-required
process-status: escalation
```

In this example, escalation is receiver-internal process metadata. It is not a normative RFC-0509 boundary outcome.

25. Security Considerations

- Confused-deputy attacks where proof intended for one receiver is presented to another receiver.
- Replay or duplicate presentment leading to repeated attempts to create local reliance.
- Requirement gaming where proof suppliers learn exact thresholds and manufacture superficial fit.
- Downgrade attacks where a receiver is induced to accept present/absent proof rather than attribute-level evaluation.
- Over-disclosure of proof attributes, evidence, receiver requirements, interest posture, or economic rulebooks.
- Intermediary leakage of presentment envelopes, requirement profiles, outcome fields, local outcome records, or fit results.
- Badge capture where a vendor attempts to convert RFC-0509 compatibility into universal pass authority.
- Vendor-rule capture where a system claims receiver-side conformance while applying vendor-controlled thresholds instead of receiver-controlled requirements.
- Attestation/evidence mismatch where an attestation for one deployment or key is presented with evidence from another deployment or key.
- Self-confirmation risk where a receiver shaped, sponsored, funded, benefits from, or is otherwise interested in the proof being accepted.

Implementations SHOULD bind presentments to audience, request context, action context, and time window. Receivers SHOULD retain discretion over how much of their requirement logic is disclosed.

Receiver interest posture is a trust and disclosure consideration. RFC-0509 does not certify receiver independence and does not automatically invalidate receiver outcomes because a receiver is interested. However,

implementations SHOULD avoid hiding interest posture where it materially affects downstream evaluation, dispute, audit, or reliance.

26. Privacy Considerations

RFC-0509 may cause proof attributes, receiver requirements, local outcome records, and receiver posture to become more explicit. That improves legibility but can increase privacy risk. Implementations SHOULD minimize disclosed attributes, use references where possible, avoid unnecessary personal data, and separate requirement disclosure from sensitive internal rulebooks.

- Receivers SHOULD request only attributes needed for the boundary outcome.
- Proof suppliers SHOULD avoid embedding sensitive evidence when a verifier reference, digest, or commitment is sufficient.
- Argument commitments SHOULD disclose their commitment posture so receivers understand whether full-payload or selective verification is possible.
- Local reliance outcomes and receiver-signed local outcome records may be sensitive and SHOULD be protected accordingly.
- Receiver-interest posture may itself be sensitive and SHOULD be disclosed only to the extent needed for the relevant boundary, audit, or downstream evaluation.

27. Relationship to Adjacent Mechanisms

Mechanism	What it does	RFC-0509 distinction
Authentication	Identifies a party or system.	Identity is not receiver-side fit.
Authorization	Permits an actor or system to attempt an action.	Prior permission is not local reliance by a receiving domain.
Payment and x402-style rails	Move value or satisfy payment conditions.	Payment does not decide whether a receiving domain can rely on proof for a local consequence.
Credentials and VCs	Assert attributes or claims.	Credentials may supply proof attributes. The receiver still decides fit.
Attestations	Describe evaluated conditions.	Attestation quality may improve proof supply but does not eliminate receiver need. Custody posture may require separate attestation.
Receipts and logs	Record events or evidence.	Receipts and logs may improve traceability; they do not themselves make evidence actionable.
Receiver-signed local outcome records	Record what a receiver evaluated and decided under local requirements.	They may preserve receiver-side reliance formation evidence, but they do not create universal reliance for other receivers.
Concept dictionaries and audit packets	Improve semantic or evidentiary legibility.	They may help express attributes; they do not decide receiver reliance.
Settlement systems	Reconcile, order, or settle outcomes.	RFC-0509 operates at the receiving boundary before or during local reliance formation; it is not settlement infrastructure.
Observability systems	Collect telemetry, traces, and operational events.	Telemetry may supply proof attributes. Receiver fit remains a local boundary decision.

28. Receiver-Side Conformance Requirements

This section defines candidate conformance requirements for receiver-side reliance formation. It does not create a public certification program. It distinguishes field-level compatibility from genuine preservation of receiver-side reliance formation.

A receiver-side implementation MAY claim RFC-0509 receiver-side conformance only where the implementation preserves receiver-controlled requirement expression, evaluates presented proof attributes against those receiver requirements, records an explicit receiver-local boundary outcome, preserves reason codes and scope limitations, and makes the resulting local outcome record inspectable as the receiver's decision. Field-level parsing, proof-side profile mapping, vendor-side evaluation, or issuance of a signed record alone SHALL NOT be sufficient to claim receiver-side conformance.

#	Requirement	Conformance question	Failure mode if absent
1	Receiver-controlled requirement	Did the receiving domain declare, reference, authorize, or withhold the requirement applied?	Vendor applies its own criteria and calls the result receiver reliance.
2	Proof-neutral presentment	Can heterogeneous proof be presented without requiring one vendor-native proof object as the only valid path?	Vendor capture or proof-object lock-in.
3	Evaluated attributes identified	Does the outcome record state which proof attributes were actually evaluated?	Record lists supplied data but not what mattered to fit.
4	Receiver-local fit	Was fit evaluated against receiver requirements rather than universal thresholds or vendor defaults?	Fit becomes a vendor score or universal badge.
5	Explicit boundary outcome	Is the local outcome explicit, using ACCEPT, HOLD, NARROW, or REFUSE, unless a receiver profile defines additional local vocabulary?	Outcome is ambiguous, inferred, or hidden inside workflow state.
6	Reason-code preservation	Is the reason for ACCEPT, HOLD, NARROW, or REFUSE recorded or referenced?	The receiver cannot later inspect why reliance was formed or withheld.
7	Scope and consequence bounded	Does the record state what consequence is allowed, withheld, narrowed, or deferred?	Proof acceptance silently expands beyond the receiver's intended use.
8	Receiver authority / custody	Is the outcome record signed, authorized, or custodied so it is inspectable as the receiver's decision?	Record is a vendor-side report, not receiver evidence.
9	Non-transitive reliance	Does the record avoid implying that another receiver must accept the same proof or outcome?	Local reliance becomes a universal pass.
10	No badge capture	Does the implementation avoid marketing RFC-0509 parsing, profile mapping, or outcome records as certification, legal effect, or universal acceptance?	RFC-0509 becomes a badge rather than a boundary grammar.
11	Implementation-neutral enforcement disclosure	Does the implementation distinguish signed/declarative fields from machine-enforced gates and lifecycle-linked states?	A reader mistakes a signed statement for runtime enforcement.
12	Receiver posture clarity	Where relevant, does the implementation avoid hiding material receiver-interest or self-confirmation posture?	A receiver's acceptance appears independent when it may be interested or self-confirming.

An implementation that uses a Reliance Decision Record, receiver outcome record, or similarly named artifact is not automatically RFC-0509-conformant merely because it records a decision, references RFC-0509, or is

cryptographically signed. The decisive questions are: Who owns the requirement? Who owns the outcome? Who bears the consequence?

A receiver-side local outcome record MUST NOT be represented as a universal reliance pass, proof-side certification, legal-effect determination, or binding outcome for other receiving domains. Acceptance, holding, narrowing, or refusal by one receiving domain remains local and non-transitive unless another receiving domain independently accepts the record under its own requirements.

29. IANA Considerations

This draft does not request IANA registration. Field names, media types, outcome registries, process-status tokens, verification-class tokens, and structured syntax remain provisional until a future public submission or registration process.

Future versions may register HTTP field names, media types, problem detail types, outcome tokens, verification-class tokens, or well-known resource locations if deployment experience justifies formal registration.

30. Future Extensions

- Structured Fields encoding for receiver requirements and boundary outcomes.
- Registered media types for requirement profiles, presentment envelopes, and receiver-side local outcome records.
- Problem Details integration for REFUSE, HOLD, NARROW, lifecycle, and process-status conditions.
- Optional receiver-published requirement profiles.
- Optional economic signal schemas that receivers may use without making RFC-0509 an economic calculator.
- Formal test vectors for object-verifiable, attested, declared, and receiver-evaluated attributes.
- Formal test vectors for whole-payload commitment, field-level commitment, and selective-disclosure commitment.
- Conformance test suites for receiver-controlled requirements, proof-neutral presentment, reason-code preservation, and non-transitive reliance.
- Compatibility profiles for agentic commerce, healthcare prior authorization, insurance, industrial safety, regulated AI governance, and enterprise controls.
- Receiver-interest posture profiles where disclosure is needed for audit, downstream review, or dispute resolution.

Appendix A. Wire Summary

Receiver says requirements are needed:

```
HTTP/2 428 Precondition Required
receiver-requirements: <reference-or-envelope>
accept-proof: application/rfc0509-presentment+json
rfc0509-outcomes: ACCEPT, HOLD, NARROW, REFUSE
```

Requester presents proof attributes:

```
POST /boundary/evaluate HTTP/2
content-type: application/rfc0509-presentment+json
proof-presentment: <reference-or-envelope>
```

Receiver returns local outcome:

```
HTTP/2 200 OK
fit-status: high
boundary-outcome: ACCEPT
```

```
local-reliance: true
outcome-reason: receiver-requirements-satisfied
local-outcome-record: <reference-or-envelope>
```

Appendix B. Boundary Outcome Vocabulary Summary

Outcome	Receiver-local meaning
ACCEPT	Receiver accepts the proof attributes for the requested action or processing context.
HOLD	Receiver defers, suspends, or pauses the boundary decision because qualification cannot be resolved at this time.
NARROW	Receiver permits only a narrower action, scope, time window, exposure, risk, or consequence than requested.
REFUSE	Receiver declines local reliance or processing for the requested action or processing context.

The following are not normative boundary outcomes in v0.8: escalation, manual review, board approval, counsel review, regulatory notification, override, reconstruction workflow, and similar internal process states. A receiver MAY disclose such process states as non-normative metadata if it chooses.

Appendix C. Receiver-Side Conformance Test Summary

Test	Question
Requirement ownership	Did the receiver declare, reference, authorize, or withhold the requirement profile?
Outcome ownership	Did the receiver form or authorize ACCEPT, HOLD, NARROW, or REFUSE?
Consequence ownership	Does the receiver's consequence model determine what may proceed, be held, narrowed, or refused?
Reason preservation	Is the reason code recorded or referenced?
Scope boundary	Is the permitted consequence bounded?
Record custody	Is the local outcome record receiver-signed, receiver-authorized, receiver-custodied, or independently inspectable as the receiver's decision?
Proof neutrality	Can heterogeneous proof be qualified without forcing it into one vendor-native proof regime?
Non-transitivity	Does the record avoid implying universal acceptance by other receivers?
No badge capture	Does the implementation avoid marketing parsing, mapping, or record creation as certification, legal effect, or universal acceptance?

Controlled public draft candidate. Not a certification, conformance approval, legal opinion, commercial alignment document, or protected implementation disclosure.