Network Working Group
Internet-Draft
Intended status: Informational
Expires: 25 November 2026

C. Vidiniotis
AutoCyber AI Pty Ltd

24 May 2026

Context Relay Protocol (CRP) -- HTTP Header Field Vocabulary
draft-vidiniotis-crp-headers-00

Abstract

This document defines the complete vocabulary of HTTP header fields for the Context Relay Protocol (CRP). CRP header fields carry AI-specific metadata — context quality, safety risk, provenance integrity, regulatory classification, agent state, and memory layer information — as standard HTTP header fields on AI request/response cycles. This specification provides normative definitions for 58 header fields across six namespaces, suitable for provisional registration in the IANA HTTP Field Name Registry.

Feedback

This is a working draft intended for submission as an IETF Internet-Draft (draft-vidiniotis-crp-headers-00). Comments and issues: https://github.com/crprotocol/spec/issues (https://github.com/crprotocol/spec/issues). The header field definitions in this document are also the basis for IANA provisional registration requests. All CRP header fields use the CRP- prefix.

Status of This Memo

Table of Contents

1. Introduction

1.1. Motivation

HTTP became the universal substrate for networked applications in part because its header mechanism provided a standardised, extensible metadata contract readable by any participant in a request/response chain — clients, servers, proxies, CDNs, WAFs — without parsing message bodies. Cache-Control made web-scale caching possible. Content-Security-Policy made transport-layer browser security possible. Both operate on the principle that metadata declared in headers is more interoperable than metadata embedded in payloads.

AI inference calls — requests to large language models — currently carry no standardised metadata about the quality, safety, or governance status of their responses. Every application operator instruments this separately, producing non-interoperable, non-verifiable, non-portable safety signals.

CRP header fields apply the HTTP lesson to AI: standardised metadata in headers, readable by any participant in the AI call chain, without parsing inference payloads.

1.2. Scope

This document defines the normative semantics, syntax, allowed values, directionality, and interaction rules for all CRP header fields. It does not define:

* The CRP gateway implementation (see CRP-SPEC-001)
* The Decision Provenance Engine that populates safety headers (see CRP-SPEC-005)
* The envelope packing algorithm that populates context headers (see CRP-SPEC-003)
* The Safety Policy directive language carried in CRP-Safety-Policy (see CRP-SPEC-006)
* The session token structure carried in CRP-Set-Session (see CRP-SPEC-007)

1.3. Relationship to HTTP

CRP header fields are standard HTTP header fields as defined in RFC 9110. They MUST be transmitted as HTTP header fields in the normal HTTP header section. They MUST NOT be placed in HTTP trailers unless explicitly noted.

CRP header fields are distinct from the inference payload (request body and response body). Per Axiom 4 of CRP-SPEC-001, CRP header fields MUST be stripped by the CRP gateway before forwarding requests to LLM providers.

2. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174].

2.1. Directionality Notation

* *REQ* — Header is sent in the HTTP _request_ (client → gateway)
* *RES* — Header is sent in the HTTP _response_ (gateway → client)
* *BOTH* — Header appears in both request and response (semantics may differ)

2.2. Value Notation

Header value syntax is described using ABNF [RFC5234]. The following base rules apply:

```
crp-token     = 1*( ALPHA / DIGIT / "-" / "_" / "." )
crp-hash      = "sha256:" 64HEXDIG
crp-float     = 1*DIGIT "." 1*DIGIT
crp-fraction  = crp-float  ; constrained to 0.0  1.0
crp-uri       = <URI as defined in RFC 3986>
crp-iso8601   = <date-time as defined in ISO 8601>
crp-version   = 1*DIGIT "." 1*DIGIT "." 1*DIGIT
```

2.3. Default Behaviour

Unless stated otherwise:

* Missing optional request headers indicate client preference defers to gateway defaults

* Missing response headers indicate the gateway did not compute the value for this call

* Header values are case-insensitive unless stated otherwise

3. Header Field Design Principles

CRP header fields follow these design principles, derived from existing HTTP header best practices:

3.1 One signal, one header. Each header carries a single, clearly defined signal. Composite values (e.g., CRP-Safety-Distortions: NUMBER_CHANGED:2, NEGATION_FLIP:1) use structured syntax where multiple values must be carried.

3.2 Machine-readable values. All values are designed for programmatic parsing by middleware, WAFs, and logging systems — not just human reading. Enum values use SCREAMING_SNAKE_CASE. Numeric values use decimal notation.

3.3 Gateway-generated response headers are authoritative. Response headers emitted by the CRP gateway are computed from live protocol data. They MUST NOT be set by clients and MUST NOT reflect LLM output. The gateway is the sole authoritative source for all CRP-Safety-*, CRP-Provenance-*, and CRP-Compliance-* response headers.

3.4 Request headers express client preferences, not mandates. CRP-Accept-* and CRP-Safety-Policy headers declare what the client wants. The gateway SHOULD honour them; where it cannot (e.g., CKF has no relevant facts), it MUST document the deviation in the corresponding response header.

3.5 Progressive adoption. Applications that ignore CRP response headers receive no harm — headers carry metadata, not payload. Applications that read and gate on CRP headers gain safety and governance capability.

4. Namespace: CRP-Context-*

Context headers carry the state of the Context Envelope and CKF interaction for the current call. They are the AI equivalent of HTTP content negotiation and cache-control headers.

4.1. CRP-Context-Quality-Tier

Direction: RES
Required: RECOMMENDED
Stability: Stable

Definition: The quality classification of the Context Envelope assembled for this call, computed by the 3-phase envelope packing algorithm (see CRP-SPEC-003). Quality tier reflects the coverage, saturation, and relevance score of facts included in the envelope.

Syntax:

CRP-Context-Quality-Tier = "S" / "A" / "B" / "C" / "D"

Values:

Value	Meaning	Saturation Range	Use
S	Superior — all critical facts included, optimal relevance	0.99	High-stakes decisions
A	High — comprehensive coverage with minor gaps	0.95	Standard production
B	Adequate — majority of relevant facts included	0.85	Acceptable for most uses
C	Marginal — significant gaps in relevant coverage	0.70	Use with caution
D	Deficient — critical facts missing or insufficient context	< 0.70	Remediation required

Table 1

Example:

CRP-Context-Quality-Tier: A

Interaction: Clients MAY set CRP-Accept-Quality to declare minimum acceptable tier. Gateway MUST retry with upgraded strategy or return HTTP 503 if minimum tier cannot be achieved.

4.2. CRP-Context-Window

Direction: RES
Required: RECOMMENDED

Definition: The current window position within the continuation chain, expressed as current/total. total is the configured maximum window depth for this session (see CRP-SPEC-004 for window continuation specification).

Syntax:

CRP-Context-Window = 1*DIGIT "/" 1*DIGIT

Example:

CRP-Context-Window: 3/5

Notes: When current equals total, the session has reached maximum continuation depth. The client MUST begin a new session or invoke dispatch_hierarchical() to continue.

4.3. CRP-Context-Saturation

Direction: RES
Required: RECOMMENDED

Definition: The ratio of token budget consumed by the Context Envelope relative to the total available context window, expressed as a decimal from 0.0 to 1.0. A saturation of 1.0 indicates the envelope has consumed the full available context window.

Syntax:

CRP-Context-Saturation = crp-fraction

Example:

CRP-Context-Saturation: 0.994

Notes: Values above 0.98 indicate context pressure. Clients receiving high saturation values SHOULD consider using a smaller query scope, enabling CKF caching, or upgrading to a longer-context model tier.

4.4. CRP-Context-Facts-Used

Direction: RES
Required: OPTIONAL

Definition: The number of discrete facts retrieved from the CKF and included in the envelope, expressed as included/available. available is the total number of candidate facts identified before relevance filtering.

Syntax:

CRP-Context-Facts-Used = 1*DIGIT "/" 1*DIGIT

Example:

CRP-Context-Facts-Used: 47/312

4.5. CRP-Context-Tokens-Used

Direction: RES
Required: OPTIONAL

Definition: The number of tokens consumed by the Context Envelope in the LLM's context window, as an integer.

Syntax:

```
CRP-Context-Tokens-Used = 1*DIGIT
```

Example:

```
CRP-Context-Tokens-Used: 105816
```

4.6. CRP-Context-Strategy

Direction: RES
Required: RECOMMENDED

Definition: The dispatch strategy employed for this call, as selected by the gateway from TaskIntent analysis or from the CRP-Accept-Strategy request header.

Syntax:

```
CRP-Context-Strategy = "push" / "pull" / "reflexive" / "agentic" /
                       "hierarchical" / "batch" / "streaming" /
                       "fan-out" / "fan-in"
```

Example:

```
CRP-Context-Strategy: reflexive
```

4.7. CRP-Context-Session-Id

Direction: BOTH
Required: REQUIRED

Definition: The unique identifier for the current CRP session. In requests, used as a hint when a CRP-Session-Token is not available. In responses, confirms the session to which this call belongs.

Syntax:

```
CRP-Context-Session-Id = "crp_sess_" 16*32( ALPHA / DIGIT )
```

Example:

CRP-Context-Session-Id: crp_sess_7f3a9bc2d4e1f083

4.8. CRP-Context-ETag

Direction: RES
Required: RECOMMENDED
Stability: Stable — *NEW in v3.0*

Definition: A hash of the current CKF fact-set state used to generate this call's Context Envelope. Analogous to HTTP ETag. Enables conditional dispatch: clients presenting this value in CRP-Context-If-Match on subsequent calls with the same knowledge domain will receive HTTP 304 if the fact-set has not changed, avoiding redundant envelope reconstruction.

Syntax:

CRP-Context-ETag = crp-hash

Example:

CRP-Context-ETag: sha256:4fa8e921abcd1234567890abcdef1234567890abcdef1234567890abcdef1234

Interaction: See CRP-Context-If-Match and CRP-Context-Cache for the full conditional dispatch mechanism.

4.9. CRP-Context-If-Match

Direction: REQ
Required: OPTIONAL
NEW in v3.0

Definition: Conditional dispatch request. If the presented ETag matches the current CKF fact-set hash, the gateway returns HTTP 304 (Context Not Modified) and skips envelope reconstruction. Analogous to HTTP If-None-Match.

Syntax:

CRP-Context-If-Match = crp-hash / "*"

Example:

CRP-Context-If-Match: sha256:4fa8e921abcd1234567890abcdef1234567890abcdef1234567890abcdef
1234

HTTP 304 response: When the gateway returns 304, the response body is empty. All CRP response headers from the previous matching call MUST be re-emitted with their unchanged values.

4.10. CRP-Context-Cache

Direction: REQ
Required: OPTIONAL
NEW in v3.0

Definition: Directives controlling CKF read/write behaviour for this call. Analogous to HTTP Cache-Control. Multiple directives are separated by commas.

Syntax:

```
CRP-Context-Cache     = cache-directive *( OWS "," OWS cache-directive )
cache-directive       = "no-store" / "no-cache" / "reuse-ckf" /
                        "only-if-ckf" / ( "max-age" "=" delta-seconds )
delta-seconds         = 1*DIGIT
```

Directive definitions:

Directive	Meaning
no-store	Do not persist this session’s facts to the CKF graph. Envelope built from temporary context only. Use for sensitive or PII-containing sessions.
no-cache	Do not reuse cached envelope. Force full 3-phase reconstruction even if ETag matches. Use when freshness is critical.
reuse-ckf	Read from CKF but do not trigger re-ingestion of source documents. Use when knowledge base is known to be current.
only-if-ckf	Refuse dispatch if CKF contains no relevant facts for this query. Returns HTTP 424 (Failed Dependency) if CKF miss.
max-age=N	Facts retrieved from CKF are considered valid for N seconds. Gateway checks CRP-Memory-Knowledge-Age against this value.

Table 2

Example:

CRP-Context-Cache: reuse-ckf, max-age=3600

GDPR note: no-store MUST be set for sessions processing personal data under GDPR Art. 5(1)(c) (data minimisation) unless CKF persistence has been explicitly authorised in the applicable DPA.

4.11. CRP-Context-Cache-Status

Direction: RES
Required: OPTIONAL
NEW in v3.0

Definition: Reports whether the envelope was served from CKF cache or freshly constructed.

Syntax:

```
CRP-Context-Cache-Status = ( "HIT" / "MISS" / "PARTIAL" )
                           [ "; reason=" crp-token ]
```

Example:

```
CRP-Context-Cache-Status: MISS; reason=facts-updated
CRP-Context-Cache-Status: HIT
CRP-Context-Cache-Status: PARTIAL; reason=partial-ckf-coverage
```

4.12. CRP-Context-Continuation-Id

Direction: BOTH
Required: OPTIONAL

Definition: An opaque token referencing the continuation point in the Window DAG. When included in a request, the gateway resumes the session from the referenced window. When included in a response, it provides the continuation pointer for the next call. See CRP-SPEC-004 for the full Window DAG specification.

Syntax:

```
CRP-Context-Continuation-Id = "crp_cont_" 16*32( ALPHA / DIGIT )
```

Example:

```
CRP-Context-Continuation-Id: crp_cont_9a2fb3c1d4e5f607
```

4.13. CRP-Accept-Quality

Direction: REQ
Required: OPTIONAL
NEW in v3.0

Definition: Client-declared minimum acceptable quality tier. The gateway MUST NOT return a response with a quality tier lower than declared. If the minimum tier cannot be achieved, the gateway MUST return HTTP 503 with a CRP-Context-Quality-Tier header indicating the maximum achievable tier.

Syntax:

```
CRP-Accept-Quality = quality-tier *( OWS "," OWS quality-tier )
quality-tier       = "S" / "A" / "B" / "C" / "D"
```

Example:

```
CRP-Accept-Quality: S, A
```

4.14. CRP-Accept-Strategy

Direction: REQ
Required: OPTIONAL
NEW in v3.0

Definition: Client-declared preferred dispatch strategies in order of preference. The gateway uses the first strategy from the list that is applicable to the current TaskIntent. If no listed strategy is applicable, the gateway falls back to automatic strategy selection and MUST report the selected strategy in CRP-Context-Strategy.

Syntax:

```
CRP-Accept-Strategy = strategy-token *( OWS "," OWS strategy-token )
strategy-token      = "push" / "pull" / "reflexive" / "agentic" /
                      "hierarchical" / "batch" / "streaming" /
                      "fan-out" / "fan-in"
```

Example:

```
CRP-Accept-Strategy: reflexive, push
```

4.15. CRP-Context-Protocol-Version

Direction: RES
Required: REQUIRED

Definition: The CRP protocol version used to process this call.

Syntax:

```
CRP-Context-Protocol-Version = crp-version
```

Example:

```
CRP-Context-Protocol-Version: 3.0.0
```

5. Namespace: CRP-Safety-*

Safety headers carry the output of the Decision Provenance Engine (DPE) for the current response. They are response-only by default (generated by the gateway); safety policy and preferences are expressed in CRP-Safety-Policy (see CRP-SPEC-006) and CRP-Accept-Risk request headers.

Security note: All CRP-Safety-* response headers are gateway-generated. Clients MUST NOT set CRP-Safety-* response headers in requests. Gateways MUST reject requests where CRP-Safety-Hallucination-Risk, CRP-Safety-Hallucination-Score, or CRP-Safety-Attribution appear as request headers.

5.1. CRP-Safety-Hallucination-Risk

Direction: RES
Required: REQUIRED (for CRP-Standard and CRP-Full conformance)

Definition: The composite hallucination risk classification for the LLM response, computed by the DPE from four weighted signals: attribution (0.35), fidelity (0.25), entailment (0.25), and specificity (0.15). See CRP-SPEC-005 for the complete scoring algorithm. Regulatory amplifiers are applied before classification (see CRP-SPEC-005 § 4.3).

Syntax:

```
CRP-Safety-Hallucination-Risk = "CRITICAL" / "HIGH" / "MEDIUM" / "LOW"
```

Classification thresholds (post-amplification composite score):

Value	Score Range	Protocol Action	Regulatory Mapping
CRITICAL	0.70	HTTP 451 if halt-on CRITICAL policy set	EU AI Act Art. 14 (halt)
HIGH	0.45	Strategy upgrade if upgrade-on-risk set	EU AI Act Art. 13 (warn)
MEDIUM	0.20	Pass with headers	EU AI Act Art. 52
LOW	< 0.20	Pass	Compliant

Table 3

Example:

```
CRP-Safety-Hallucination-Risk: LOW
```

5.2. CRP-Safety-Hallucination-Score

Direction: RES
Required: RECOMMENDED

Definition: The raw composite hallucination risk score from the DPE before classification, expressed as a decimal from 0.0 to 1.0. Higher values indicate greater hallucination risk.

Syntax:

```
CRP-Safety-Hallucination-Score = crp-fraction
```

Example:

```
CRP-Safety-Hallucination-Score: 0.14
```

5.3. CRP-Safety-Attribution

Direction: RES
Required: RECOMMENDED

Definition: The dominant attribution type of claims in the response, computed by DPE Stage 2.

Syntax:

```
CRP-Safety-Attribution = "CONTEXT_GROUNDED" / "PARAMETRIC" / "MIXED" / "UNVERIFIABLE"
```

Values:

Value	Meaning
CONTEXT_GROUNDED	Majority of claims verifiably supported by Context Envelope content
PARAMETRIC	Majority of claims drawn from LLM parametric memory, not envelope
MIXED	Significant presence of both context-grounded and parametric claims
UNVERIFIABLE	Claims cannot be attributed to any verifiable source

Table 4

Example:

```
CRP-Safety-Attribution: CONTEXT_GROUNDED
```

5.4. CRP-Safety-Grounding-Pct

Direction: RES
Required: RECOMMENDED

Definition: The percentage of factual claims in the response that are verifiably supported by content in the Context Envelope, as a decimal from 0.0 to 1.0.

Syntax:

```
CRP-Safety-Grounding-Pct = crp-fraction
```

Example:

```
CRP-Safety-Grounding-Pct: 0.923
```

5.5. CRP-Safety-Fabrications

Direction: RES
Required: RECOMMENDED

Definition: The count of fabricated entities detected by DPE Stage 3. A fabricated entity is a named entity (person, organisation, date, statistic, citation) that appears in the response but has no verifiable basis in the Context Envelope or any grounded source.

Syntax:

```
CRP-Safety-Fabrications = 1*DIGIT
```

Example:

```
CRP-Safety-Fabrications: 0
```

Note: A value of 0 is the expected result for well-grounded responses. Any non-zero value MUST be reported to CRP-Safety-Report-URI if configured.

5.6. CRP-Safety-Distortions

Direction: RES
Required: RECOMMENDED

Definition: The count of detected fidelity distortions in the response, where the response has misrepresented content that exists in the Context Envelope. Distortion types are reported as a structured list.

Syntax:

```
CRP-Safety-Distortions = distortion-count [ "; types=" distortion-list ]
distortion-count       = 1*DIGIT
distortion-list        = distortion-type *( "," distortion-type )
distortion-type        = "NUMBER_CHANGED" / "NEGATION_FLIP" / "DATE_SHIFTED" /
                         "ENTITY_SUBSTITUTED" / "MAGNITUDE_ALTERED" / "CONTEXT_STRIPPE
D"
```

Example:

```
CRP-Safety-Distortions: 1; types=NUMBER_CHANGED
CRP-Safety-Distortions: 0
```

5.7. CRP-Safety-Contradictions

Direction: RES
Required: OPTIONAL

Definition: The count of internal contradictions detected in the response — instances where the response contradicts itself (intra-window) or contradicts an earlier window in the session (cross-window).

Syntax:

```
CRP-Safety-Contradictions = 1*DIGIT [ "; scope=" ( "intra" / "cross-window" / "both" )
 ]
```

Example:

```
CRP-Safety-Contradictions: 0
CRP-Safety-Contradictions: 2; scope=cross-window
```

5.8. CRP-Safety-Omissions

Direction: RES
Required: OPTIONAL

Definition: Summary of material omissions detected — cases where the envelope contained information critical to a complete answer that the LLM did not include in the response.

Syntax:

```
CRP-Safety-Omissions = omission-entry *( OWS "," OWS omission-entry )
omission-entry       = ( "CRITICAL" / "HIGH" / "MEDIUM" ) ":" 1*DIGIT
```

Example:

```
CRP-Safety-Omissions: CRITICAL:0, HIGH:1, MEDIUM:2
```

5.9. CRP-Safety-Entailment-Score

Direction: RES
Required: RECOMMENDED

Definition: The NLI (Natural Language Inference) cross-encoder entailment score measuring the degree to which the response is logically entailed by the Context Envelope, from 0.0 (contradiction) to 1.0 (full entailment). This is one of the four DPE composite signal inputs (weight: 0.25).

Syntax:

```
CRP-Safety-Entailment-Score = crp-fraction
```

Example:

```
CRP-Safety-Entailment-Score: 0.912
```

5.10. CRP-Safety-Oversight-Mode

Direction: BOTH
Required: RECOMMENDED

Definition: In requests, declares the human oversight level required for this session. In responses, confirms the oversight mode applied by the gateway and (if applicable) whether an oversight event was triggered.

Syntax:

```
CRP-Safety-Oversight-Mode = "auto" / "human-review" / "halt" / "log-only"
```

Values:

Value	Meaning
auto	Gateway applies oversight based on Safety Policy and risk level
human-review	Response is held pending human review for any HIGH or CRITICAL risk
halt	Response is halted unconditionally for any CRITICAL risk (EU AI Act Art. 14)
log-only	Risk signals logged and reported but response always passed through

Table 5

Example:

CRP-Safety-Oversight-Mode: auto

5.11. CRP-Safety-Mode

Direction: REQ
Required: OPTIONAL
NEW in v3.0

Definition: The global safety strictness level for the session. Overrides individual Safety Policy directives where they are less restrictive.

Syntax:

CRP-Safety-Mode = "strict" / "warn" / "permissive"

Value	Behaviour
strict	Equivalent to halt-on CRITICAL; warn-on HIGH; block-ungrounded
warn	All risk levels passed; HIGH and CRITICAL emitted in headers and reported
permissive	Risk signals computed and emitted; no gating applied

Table 6

Example:

CRP-Safety-Mode: strict

5.12. CRP-Safety-Policy

Direction: REQ
Required: RECOMMENDED
NEW in v3.0

Definition: A directive string declaring the AI safety policy for this session, analogous to Content-Security-Policy. The full directive grammar is defined in CRP-SPEC-006.

Syntax:

CRP-Safety-Policy = policy-directive *(";" OWS policy-directive)

Example:

CRP-Safety-Policy: default-src context; halt-on CRITICAL; warn-on HIGH; require-grounding 0.75; block-ungrounded; report-uri https://comply.crprotocol.io/reports

Interaction with CRP-Safety-Mode: If both CRP-Safety-Mode and CRP-Safety-Policy are set, the more restrictive of the two applies per-directive.

5.13. CRP-Safety-Report-URI

Direction: REQ
Required: OPTIONAL
NEW in v3.0

Definition: URI to which the gateway MUST POST a JSON violation report when a Safety Policy violation occurs (any halt-on trigger, fabrication detected, chain integrity broken). Analogous to CSP report-uri.

Syntax:

CRP-Safety-Report-URI = crp-uri

Example:

CRP-Safety-Report-URI: https://comply.crprotocol.io/reports/my-org

Report format: The gateway POSTs Content-Type: application/json with fields: session_id, window_number, violation_type, risk_level, audit_trail_uri, timestamp.

5.14. CRP-Accept-Risk

Direction: REQ
Required: OPTIONAL
NEW in v3.0

Definition: Maximum risk level the client is willing to accept for this call. If the computed risk exceeds the declared level, the gateway MUST either retry with upgraded strategy (if upgrade-on-risk is set in Safety Policy) or halt and return HTTP 451.

Syntax:

CRP-Accept-Risk = "CRITICAL" / "HIGH" / "MEDIUM" / "LOW"

Example:

CRP-Accept-Risk: MEDIUM

5.15. CRP-Safety-Retry-After

Direction: RES
Required: CONDITIONAL — MUST be sent with HTTP 451

Definition: Indicates when or under what condition the client may retry a halted call. Sent alongside HTTP 451 when a CRITICAL risk halt or UNACCEPTABLE EU AI Act risk classification prevents response delivery.

Syntax:

CRP-Safety-Retry-After = (delta-seconds / crp-iso8601 / "oversight-required")

Example:

```
CRP-Safety-Retry-After: oversight-required
CRP-Safety-Retry-After: 300
```

5.16. CRP-Safety-Nonce

Direction: BOTH
Required: OPTIONAL
NEW in v3.0

Definition: A per-session nonce bound to the Safety Policy hash at session initialisation. Prevents Safety Policy replay attacks. The gateway generates the nonce at session start and binds it to the hash of the active Safety Policy. Subsequent requests presenting a Safety Policy with a different hash but the same nonce are rejected.

Syntax:

```
CRP-Safety-Nonce = "base64:" 1*( ALPHA / DIGIT / "+" / "/" / "=" )
```

Example:

```
CRP-Safety-Nonce: base64:nZ8fXwKq2mP9vR==
```

6. Namespace: CRP-Provenance-*

Provenance headers carry the cryptographic audit chain and DAG state for the current call. They enable tamper-evident compliance evidence and chain verification by any party holding the session HMAC key.

6.1. CRP-Provenance-HMAC

Direction: RES
Required: REQUIRED (for CRP-Standard and CRP-Full)

Definition: The HMAC-SHA256 hash for this window’s audit record, chained from the previous window’s HMAC. See CRP-SPEC-011 for the complete chain algorithm. This value is the primary tamper-evidence mechanism: modifying any past audit record invalidates this value and all subsequent values in the chain.

Syntax:

```
CRP-Provenance-HMAC = crp-hash
```

Example:

```
CRP-Provenance-HMAC: sha256:4fa8e921abcd1234567890abcdef1234567890abcdef1234567890abcdef1
234
```

6.2. CRP-Provenance-Window-HMAC

Direction: RES
Required: RECOMMENDED

Definition: The per-window HMAC covering only the current window's content and metadata (not chained). Used for single-window verification without requiring the full chain.

Syntax:

```
CRP-Provenance-Window-HMAC = crp-hash
```

Example:

```
CRP-Provenance-Window-HMAC: sha256:9bce472f1234abcd...
```

6.3. CRP-Provenance-DAG-Root

Direction: RES
Required: OPTIONAL

Definition: The root node identifier of the Window DAG for the current session. Used to anchor the provenance graph for session-level verification and CRP Comply evidence import.

Syntax:

```
CRP-Provenance-DAG-Root = "dag:" crp-token
```

Example:

```
CRP-Provenance-DAG-Root: dag:crp_win_a7f3b2c1d4e5
```

6.4. CRP-Provenance-Chain-Integrity

Direction: RES
Required: REQUIRED (for CRP-Standard and CRP-Full)

Definition: The result of chain integrity verification for the current session's HMAC chain up to and including the current window. A BROKEN result MUST trigger an audit incident.

Syntax:

```
CRP-Provenance-Chain-Integrity = "VALID" / "BROKEN" / "PARTIAL" / "UNVERIFIED"
```

Value	Meaning
VALID	All windows in chain verify correctly
BROKEN	One or more windows fail verification — possible tampering
PARTIAL	Chain verified for available windows; some windows missing
UNVERIFIED	Verification not performed (e.g., first window of session)

Table 7

Example:

```
CRP-Provenance-Chain-Integrity: VALID
```

6.5. CRP-Provenance-Claim-Count

Direction: RES
Required: OPTIONAL

Definition: The number of discrete factual claims identified in the LLM response by DPE Stage 1 (claim segmentation).

Syntax:

```
CRP-Provenance-Claim-Count = 1*DIGIT
```

Example:

```
CRP-Provenance-Claim-Count: 23
```

6.6. CRP-Provenance-Attribution-Score

Direction: RES
Required: RECOMMENDED

Definition: The composite attribution score across all claims in the response, from 0.0 (fully parametric) to 1.0 (fully context-grounded). This is one of the four DPE composite signal inputs (weight: 0.35, as 1 - attribution_score).

Syntax:

CRP-Provenance-Attribution-Score = crp-fraction

Example:

CRP-Provenance-Attribution-Score: 0.913

6.7. CRP-Provenance-Fidelity-Score

Direction: RES
Required: RECOMMENDED

Definition: The fidelity score for the response, measuring the accuracy with which context-grounded claims reproduce their source facts, from 0.0 (severe distortion) to 1.0 (exact fidelity). This is one of the four DPE composite signal inputs (weight: 0.25).

Syntax:

CRP-Provenance-Fidelity-Score = crp-fraction

Example:

CRP-Provenance-Fidelity-Score: 0.978

6.8. CRP-Provenance-Report-URI

Direction: RES
Required: RECOMMENDED

Definition: URI of the full DPE provenance report for this call, stored in CRP Comply. Any log aggregator, SIEM, or auditor that captures this header value has a direct link to the complete regulatory evidence record for this specific AI call.

Syntax:

CRP-Provenance-Report-URI = crp-uri

Example:

CRP-Provenance-Report-URI: https://comply.crprotocol.io/p/7fa3bc9e2d

6.9. CRP-Provenance-Window-Lineage

Direction: RES
Required: OPTIONAL

Definition: The ordered chain of window IDs leading to the current window, expressed as an arrow-separated sequence. Useful for debugging continuation chains and for auditor reconstruction of session context.

Syntax:

```
CRP-Provenance-Window-Lineage = window-id *( " -> " window-id )
window-id                     = "crp_win_" crp-token
```

Example:

```
CRP-Provenance-Window-Lineage: crp_win_a7f3 -> crp_win_b9c2 -> crp_win_c1d4
```

7. Namespace: CRP-Compliance-*

Compliance headers carry per-call regulatory classification metadata, generated by the CRP compliance pipeline. They bridge the protocol layer and the regulatory evidence layer (CRP Comply). The CRP-Compliance-Audit-Trail-URI header is the primary integration point between every AI call and its regulatory record.

7.1. CRP-Compliance-EU-AI-Act

Direction: RES
Required: RECOMMENDED

Definition: The EU AI Act risk classification for the AI system and use case associated with this call, per Regulation (EU) 2024/1689 Article 6 classification criteria.

Syntax:

```
CRP-Compliance-EU-AI-Act = "UNACCEPTABLE" / "HIGH" / "LIMITED" / "MINIMAL"
```

Value	EU AI Act Article	Regulatory Consequence
UNACCEPTABLE	Art. 5	Prohibited. Gateway MUST halt and return HTTP 451.
HIGH	Art. 6 + Annex III	Conformity assessment required before deployment
LIMITED	Art. 52	Transparency obligations (disclose AI interaction)
MINIMAL	—	No specific obligations beyond general law

Table 8

Example:

CRP-Compliance-EU-AI-Act: LIMITED

Note: Classification is determined by the CRP compliance pipeline based on the registered AI system type, deployment domain, and output use. It is NOT determined per-call from response content alone. Operators MUST register their AI system type during CRP Gateway setup.

7.2. CRP-Compliance-NIST-Tier

Direction: RES
Required: OPTIONAL

Definition: The NIST AI RMF risk tier for this call’s context.

Syntax:

CRP-Compliance-NIST-Tier = "TIER-1" / "TIER-2" / "TIER-3" / "TIER-4"

Example:

CRP-Compliance-NIST-Tier: TIER-2

7.3. CRP-Compliance-GDPR-PII

Direction: RES
Required: RECOMMENDED

Definition: Indicates whether personal data (as defined under GDPR Art. 4(1)) was detected in the request prompt or LLM response by the DPE PII detection module.

Syntax:

```
CRP-Compliance-GDPR-PII = "true" / "false"
```

Example:

```
CRP-Compliance-GDPR-PII: false
```

Note: A value of true triggers GDPR Art. 5(1)(c) data minimisation obligations. If CRP-Context-Cache: no-store is not set when CRP-Compliance-GDPR-PII: true is emitted, the gateway MUST log a compliance warning.

7.4. CRP-Compliance-ISO-42001

Direction: RES
Required: OPTIONAL

Definition: ISO/IEC 42001:2023 control IDs satisfied by the evidence generated for this call. Expressed as a comma-separated list of Annex A control identifiers.

Syntax:

```
CRP-Compliance-ISO-42001 = iso-control-id *( OWS "," OWS iso-control-id )
iso-control-id           = "A." 1*DIGIT "." 1*DIGIT [ "." 1*DIGIT ]
```

Example:

```
CRP-Compliance-ISO-42001: A.6.1.2, A.9.4, A.10.2
```

7.5. CRP-Compliance-Audit-Trail-Id

Direction: RES
Required: REQUIRED (for CRP-Standard and CRP-Full)

Definition: The unique identifier for the audit trail record generated for this call. This ID references the HMAC-chained audit event in the CRP Comply evidence database.

Syntax:

```
CRP-Compliance-Audit-Trail-Id = "crp_trail_" 16*32( ALPHA / DIGIT )
```

Example:

CRP-Compliance-Audit-Trail-Id: crp_trail_7fa3bc9e2d14f5a8

7.6. CRP-Compliance-Audit-Trail-URI

Direction: RES
Required: RECOMMENDED

Definition: Deep-link URI to the full regulatory evidence pack for this call in CRP Comply. Any downstream system — log aggregator, SIEM, auditor tool — that captures this header can navigate directly to the complete compliance record for this specific AI call.

Syntax:

CRP-Compliance-Audit-Trail-URI = crp-uri

Example:

CRP-Compliance-Audit-Trail-URI: https://comply.crprotocol.io/t/7fa3bc9e2d14f5a8

7.7. CRP-Compliance-Data-Residency

Direction: BOTH
Required: OPTIONAL
NEW in v3.0

Definition: Declares the data residency jurisdiction for processing and storing this call's data. In requests, expresses the client's requirement. In responses, confirms the jurisdiction in which data was processed. Gateway MUST NOT process data in a different jurisdiction than declared in the request.

Syntax:

CRP-Compliance-Data-Residency = 2ALPHA / "EU" / "AU" / "US" / "UK"

Example:

CRP-Compliance-Data-Residency: EU

7.8. CRP-Compliance-Controls-Met

Direction: RES
Required: OPTIONAL

Definition: The number of applicable regulatory controls satisfied by the evidence generated for this call, expressed as met/total. Total is the number of controls applicable given the registered AI system type and applicable regulations.

Syntax:

```
CRP-Compliance-Controls-Met = 1*DIGIT "/" 1*DIGIT
```

Example:

```
CRP-Compliance-Controls-Met: 33/35
```

8. Namespace: CRP-Agent-*

Agent headers carry state for agentic dispatch sessions — calls using dispatch_agentic(), dispatch_hierarchical(), dispatch_fan_out(), or dispatch_fan_in(). They are relevant only when CRP-Context-Strategy indicates an agentic dispatch mode.

8.1. CRP-Agent-Phase

Direction: RES
Required: OPTIONAL (REQUIRED when Strategy = agentic)

Definition: The current cognitive phase of the agentic dispatch loop, as defined in CRP-SPEC-008.

Syntax:

```
CRP-Agent-Phase = "ANALYZE" / "PLAN" / "GENERATE" / "EVALUATE" /
                  "CRITIQUE" / "REFINE" / "INTEGRATE" / "COMPLETE"
```

Example:

```
CRP-Agent-Phase: EVALUATE
```

8.2. CRP-Agent-Loop-Depth

Direction: RES
Required: OPTIONAL (REQUIRED when Strategy = agentic or hierarchical)

Definition: The nesting depth of the current agent within a multi-agent hierarchy. The root agent has depth 0. Each delegating level increments by 1. Gateways MUST reject requests where loop depth exceeds the configured maximum (default: 5).

Syntax:

CRP-Agent-Loop-Depth = 1*DIGIT

Example:

CRP-Agent-Loop-Depth: 2

8.3. CRP-Agent-Safety-Budget

Direction: BOTH
Required: RECOMMENDED (for agentic strategies)
NEW in v3.0

Definition: The remaining safety risk budget for the current agent session, expressed as a decimal from 0.0 to 1.0. The budget starts at 1.0 and is decremented by the gateway on each call according to the risk level:

Risk Level	Default Decrement
LOW	0.00 (no decrement)
MEDIUM	0.05
HIGH	0.15
CRITICAL	0.35

Table 9

When the budget reaches 0.10, the gateway MUST upgrade CRP-Safety-Oversight-Mode to human-review regardless of Safety Policy. When it reaches 0.00, the gateway MUST halt and return HTTP 451.

In requests from orchestrator agents, this header passes the remaining budget down to sub-agents. Sub-agents MUST NOT inflate the budget.

Syntax:

CRP-Agent-Safety-Budget = crp-fraction

Example:

CRP-Agent-Safety-Budget: 0.63

8.4. CRP-Agent-Tool-Calls

Direction: RES
Required: OPTIONAL

Definition: The count of tool invocations made by the agentic dispatch loop for this call.

Syntax:

```
CRP-Agent-Tool-Calls = 1*DIGIT
```

Example:

```
CRP-Agent-Tool-Calls: 4
```

8.5. CRP-Agent-Session-Parent

Direction: BOTH
Required: OPTIONAL

Definition: The session ID of the parent agent session. Set by orchestrator agents when delegating to sub-agents. Enables the provenance DAG to record the full fan-out hierarchy.

Syntax:

```
CRP-Agent-Session-Parent = "crp_sess_" 16*32( ALPHA / DIGIT )
```

Example:

```
CRP-Agent-Session-Parent: crp_sess_4b2f1c3d5e6a7b8c
```

8.6. CRP-Agent-Dispatch-Strategy

Direction: BOTH
Required: OPTIONAL

Definition: Alias for CRP-Context-Strategy scoped to agentic contexts. Provided for clarity in multi-agent logging where CRP-Context-Strategy may be omitted.

Syntax: Same as CRP-Context-Strategy.

8.7. CRP-Agent-Revision-Round

Direction: RES
Required: OPTIONAL (present when Strategy = reflexive)

Definition: The current revision pass number within a reflexive dispatch cycle, expressed as current/max.

Syntax:

CRP-Agent-Revision-Round = 1*DIGIT "/" 1*DIGIT

Example:

CRP-Agent-Revision-Round: 2/3

9. Namespace: CRP-Memory-*

Memory headers expose the state of the four-tier memory hierarchy interaction for the current call. They inform clients which memory tiers were accessed and provide cache-like signals for knowledge freshness.

9.1. CRP-Memory-Tier-Hit

Direction: RES
Required: OPTIONAL

Definition: The highest memory tier accessed to serve this call's Context Envelope. Higher tier numbers indicate deeper (slower, more persistent) storage was required.

Syntax:

CRP-Memory-Tier-Hit = "0" / "1" / "2" / "3"

Value	Tier	Storage	Latency
0	Active	In-context	<1ms
1	Hot	Session cache	<10ms
2	Warm	Recent CKF	<100ms
3	Cold	Full CKF graph	<1000ms

Table 10

Example:

CRP-Memory-Tier-Hit: 2

9.2. CRP-Memory-CKF-Hits

Direction: RES
Required: OPTIONAL

Definition: The number of facts retrieved from the Contextual Knowledge Fabric (Tier 3 cold storage) for inclusion in this call's envelope.

Syntax:

```
CRP-Memory-CKF-Hits = 1*DIGIT
```

Example:

```
CRP-Memory-CKF-Hits: 34
```

9.3. CRP-Memory-CKF-Community

Direction: RES
Required: OPTIONAL

Definition: The Leiden community cluster label from the CKF graph most relevant to this call's query. Indicates which knowledge domain the CKF served.

Syntax:

```
CRP-Memory-CKF-Community = crp-token
```

Example:

```
CRP-Memory-CKF-Community: eu-ai-act-compliance
```

9.4. CRP-Memory-Knowledge-Age

Direction: RES
Required: OPTIONAL

Definition: The elapsed time since the most recently ingested fact used in this call's envelope, expressed as an ISO 8601 duration. Indicates knowledge freshness.

Syntax:

```
CRP-Memory-Knowledge-Age = "P" [ 1*DIGIT "Y" ] [ 1*DIGIT "M" ] [ 1*DIGIT "D" ]
                           [ "T" [ 1*DIGIT "H" ] [ 1*DIGIT "M" ] [ 1*DIGIT "S" ] ]
```

Example:

```
CRP-Memory-Knowledge-Age: P3D
CRP-Memory-Knowledge-Age: PT6H
```

10. Session State Headers

Session state headers are the CRP equivalent of HTTP cookies — they carry signed session state, enabling stateless session relay across language boundaries and gateway instances. Full specification in CRP-SPEC-007.

10.1. CRP-Set-Session

Direction: RES
Required: REQUIRED for first window; RECOMMENDED for subsequent windows

Definition: Sets the CRP session token on the client, analogous to HTTP Set-Cookie. Carries signed session state including session ID, window number, quality history, safety budget, and HMAC chain tip. The token is signed with HMAC-SHA256 using the session signing key and MUST be validated by the gateway on subsequent requests.

Syntax:

```
CRP-Set-Session = "token=" session-token
                  *( ";" OWS session-attribute )
session-token   = 1*( ALPHA / DIGIT / "+" / "/" / "." / "=" / "-" / "_" )
session-attribute = "Path=" path-value
                  / "Max-Age=" delta-seconds
                  / "Signed"
                  / "SameSite=" ( "Strict" / "Lax" / "None" )
                  / "Window=" 1*DIGIT
                  / "QualityHistory=" quality-history
quality-history = quality-tier *( "," quality-tier )
```

Example:

```
CRP-Set-Session: token=eyJzZXNzaW9uX2lkIjoiY3JwX3Nlc3NfN2YzYSJ9.sha256:sig...; Path=/; Ma
x-Age=3600; Signed; SameSite=Strict; Window=3; QualityHistory=A,A,B
```

10.2. CRP-Session-Token

Direction: REQ
Required: RECOMMENDED (when set by prior CRP-Set-Session)

Definition: The session token received via CRP-Set-Session, returned by the client on subsequent requests to resume the session, analogous to HTTP Cookie. The gateway validates the token signature and extracts session state without requiring server-side session storage.

Syntax:

CRP-Session-Token = session-token

Example:

```
CRP-Session-Token: eyJzZXNzaW9uX2lkIjoiY3JwX3Nlc3NfN2YzYSJ9.sha256:sig...
```

11. LLM Configuration Headers

LLM configuration headers allow the CRP gateway to dynamically adjust LLM inference parameters based on session safety state. These headers are used internally by the gateway — clients SHOULD NOT set these directly.

11.1. CRP-LLM-Temperature

Direction: Internal (gateway use)
Required: N/A — internal to gateway

Definition: The temperature value used for the LLM call, after any safety-driven adjustments. Logged in the audit trail.

Range: 0.0 2.0. Gateway reduces toward 0.2 on reflexive re-dispatch for HIGH-risk sessions.

11.2. CRP-LLM-Grounding-Mode

Direction: REQ
Required: OPTIONAL

Definition: Controls the grounding instruction injected into the LLM system prompt by the gateway.

Syntax:

CRP-LLM-Grounding-Mode = "context-strict" / "context-preferred" / "open"

Value	System Prompt Instruction
context-strict	"Answer only using the provided context. Do not use external knowledge."
context-preferred	"Prefer the provided context. Indicate when using general knowledge."
open	No grounding instruction injected.

Table 11

Default: context-preferred

Example:

CRP-LLM-Grounding-Mode: context-strict

11.3. CRP-LLM-Reproducibility-Seed

Direction: REQ / stored in audit trail
Required: OPTIONAL
NEW in v3.0

Definition: A numeric seed for deterministic LLM sampling, when the provider supports it. Stored in the HMAC audit chain to enable exact regeneration of any response for audit replay. GDPR Art. 22 requires that automated decisions be explainable and reproducible.

Syntax:

CRP-LLM-Reproducibility-Seed = 1*DIGIT

Example:

CRP-LLM-Reproducibility-Seed: 42

12. Header Interaction Rules

12.1. Safety Policy and Override Headers

When CRP-Safety-Policy and CRP-Safety-Mode are both present, the *more restrictive* setting applies per-directive: - CRP-Safety-Mode: strict + CRP-Safety-Policy: warn-on CRITICAL → halt-on CRITICAL (strict wins) - CRP-Safety-Mode: permissive + CRP-Safety-Policy: halt-on CRITICAL → halt-on CRITICAL (policy wins)

12.2. ETag and Cache Interaction

CRP-Context-If-Match is only evaluated when CRP-Context-Cache does NOT contain no-cache. If no-cache is set, the gateway MUST bypass ETag validation and reconstruct the envelope unconditionally.

12.3. Session Token Priority

When both CRP-Session-Token and CRP-Context-Session-Id are present, CRP-Session-Token takes precedence. The gateway validates the token signature; if valid, the embedded session ID overrides CRP-Context-Session-Id.

12.4. Agentic Safety Budget Propagation

In multi-agent chains, CRP-Agent-Safety-Budget presented in a request from a sub-agent is the budget ceiling for that sub-agent's session. The sub-agent's gateway MUST NOT issue a safety budget higher than the value received. This prevents budget inflation in nested agent calls.

12.5. Compliance Headers Require Registered AI System

CRP-Compliance-EU-AI-Act classification is only meaningful when the AI system has been registered in the CRP Gateway configuration with a system type, deployment domain, and intended purpose. Unregistered systems MUST receive CRP-Compliance-EU-AI-Act: UNKNOWN and a log warning.

13. Error Semantics

13.1. HTTP Status Codes Used by CRP

Status	Condition	Required Headers
200 OK	Normal response	All applicable CRP headers
304 Not Modified	ETag match — context not changed	CRP-Context-ETag
400 Bad Request	Malformed Safety Policy, nonce mismatch	CRP-Safety-Nonce
401 Unauthorized	Invalid or expired session	—

	token	
424 Failed Dependency	only-if-ckf set but CKF miss	CRP-Context-Cache-Status: MISS
451 Unavailable For Legal Reasons	Safety policy halt — CRITICAL risk or UNACCEPTABLE EU AI Act class	CRP-Safety-Hallucination-Risk, CRP-Safety-Retry-After, CRP-Compliance-Audit-Trail-URI
503 Service Unavailable	Minimum quality tier cannot be achieved	CRP-Context-Quality-Tier (max achievable)

Table 12

13.2. HTTP 451 Semantics

HTTP 451 (Unavailable For Legal Reasons, RFC 7725) is used by CRP to indicate a safety or regulatory halt. The body of a 451 response MUST include a JSON object:

```
{
  "crp_halt_reason": "CRITICAL_HALLUCINATION_RISK | UNACCEPTABLE_EU_AI_ACT | SAFETY_BU
DGET_DEPLETED",
  "session_id": "crp_sess_...",
  "audit_trail_uri": "https://comply.crprotocol.io/t/...",
  "oversight_required": true,
  "retry_condition": "oversight-required | <ISO8601 datetime>"
}
```

14. Security Considerations

14.1. Header Injection

Clients MUST NOT set CRP-Safety-* (response namespace), CRP-Provenance-* (response namespace), or CRP-Compliance-* (response namespace) headers in requests. Gateways MUST validate and strip any such headers from client requests before processing.

LLM outputs MUST NOT be parsed for CRP header values. All response headers are injected by the gateway post-response analysis, not derived from response content.

14.2. Session Token Security

CRP-Session-Token values are cryptographically signed. Forged tokens will fail signature validation. Tokens with expired expires_at MUST be rejected with HTTP 401. See CRP-SPEC-015 § 3.2.

14.3. Safety Policy Integrity

The CRP-Safety-Policy value MUST be validated for syntactic correctness per the grammar in CRP-SPEC-006 before session initialisation. Policies containing unrecognised directives MUST be rejected, not silently ignored, to prevent policy bypass through unknown directive injection.

14.4. HMAC Chain Protection

The HMAC chain key (used to generate CRP-Provenance-HMAC) MUST be stored securely and never transmitted in any CRP header. See CRP-SPEC-015 § 3.1 for the complete HMAC specification.

15. Privacy Considerations

CRP response headers carry metadata about AI calls and may indirectly reveal information about the content of those calls. Specifically:

* CRP-Compliance-GDPR-PII: true reveals that the call contained personal data

* CRP-Safety-Attribution: PARAMETRIC may reveal that the request was outside the trained knowledge domain

* CRP-Memory-CKF-Community reveals the knowledge domain of the query

Implementors MUST consider the sensitivity of CRP response headers when making them available to browser-based clients. In particular, CRP-Compliance-GDPR-PII SHOULD be treated as sensitive and not exposed to JavaScript.

The CRP-Context-Cache: no-store directive MUST be used for calls processing personal data to prevent persistence in the CKF. See CRP-SPEC-015 § 6.

16. IANA Considerations

16.1. HTTP Field Name Registrations

This document requests provisional registration of the following HTTP field names in the IANA HTTP Field Name Registry (per RFC 9110 §16.3). All fields share the following properties unless noted:

* *Applicable Protocol:* http
* *Status:* provisional
* *Author/Change Controller:* AutoCyber AI Pty Ltd contact@crprotocol.io (mailto:contact@crprotocol.io)
* *Specification Document:* https://crprotocol.io/spec/headers/ (this document)

Priority registration set (10 headers — submit first):

Field Name	Direction	Reference
CRP-Context-Quality-Tier	Response	§4.1
CRP-Safety-Hallucination-Risk	Response	§5.1
CRP-Provenance-HMAC	Response	§6.1
CRP-Compliance-EU-AI-Act	Response	§7.1
CRP-Safety-Policy	Request	§5.12
CRP-Agent-Safety-Budget	Both	§8.3
CRP-Set-Session	Response	§10.1
CRP-Context-ETag	Response	§4.8
CRP-Compliance-Audit-Trail-URI	Response	§7.6
CRP-Safety-Oversight-Mode	Both	§5.10

Table 13

Full registration set: All 58 headers defined in Sections 411.

16.2. Well-Known URI Registration

This document requests registration of /.well-known/crp-gateway.json in the IANA Well-Known URIs registry (per RFC 8615) for CRP Gateway capability advertisement.

17. References

17.1. 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>.

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

[RFC7725] Bray, T., "An HTTP Status Code to Report Legal Obstacles", RFC 7725, RFC 7725, DOI 10.17487/RFC7725, February 2016, <https://www.rfc-editor.org/info/rfc7725>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

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

[RFC9110] Fielding, R., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/info/rfc9110>.

[RFC2104] Krawczyk, H., "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, RFC 2104, DOI 10.17487/RFC2104, February 1997, <https://www.rfc-editor.org/info/rfc2104>.

[RFC5869] Krawczyk, H., "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)", RFC 5869, RFC 5869, DOI 10.17487/RFC5869, May 2010, <https://www.rfc-editor.org/info/rfc5869>.

17.2. Informative References

[EU-AI-ACT] European Parliament and Council of the European Union, "Regulation (EU) 2024/1689 of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)", 2024.

[ISO42001] International Organization for Standardization, "ISO/IEC 42001:2023 — Artificial intelligence — Management system", 2023.

[NIST-AI-RMF] National Institute of Standards and Technology (NIST), "Artificial Intelligence Risk Management Framework (AI RMF 1.0)", January 2023, <https://airc.nist.gov/RMF>.

[GDPR] European Parliament and Council of the European Union, "Regulation (EU) 2016/679 of the European Parliament and of the Council on the protection of natural persons with regard to the processing of personal data (General Data Protection Regulation)", 2016.

Complete Header Index

Header	Namespace	Direction	Section	Conformance
CRP-Accept-Quality	Context	REQ	4.13	Optional
CRP-Accept-Risk	Safety	REQ	5.14	Optional
CRP-Accept-Strategy	Context	REQ	4.14	Optional
CRP-Agent-Dispatch-Strategy	Agent	BOTH	8.6	Optional
CRP-Agent-Loop-Depth	Agent	RES	8.2	Recommended
CRP-Agent-Phase	Agent	RES	8.1	Conditional
CRP-Agent-Revision-Round	Agent	RES	8.7	Optional
CRP-Agent-Safety-Budget	Agent	BOTH	8.3	Recommended

CRP-Agent-Session-Parent	Agent	BOTH	8.5	Optional
CRP-Agent-Tool-Calls	Agent	RES	8.4	Optional
CRP-Compliance-Audit-Trail-Id	Compliance	RES	7.5	Required*
CRP-Compliance-Audit-Trail-URI	Compliance	RES	7.6	Recommended
CRP-Compliance-Controls-Met	Compliance	RES	7.8	Optional
CRP-Compliance-Data-Residency	Compliance	BOTH	7.7	Optional
CRP-Compliance-EU-AI-Act	Compliance	RES	7.1	Recommended
CRP-Compliance-GDPR-PII	Compliance	RES	7.3	Recommended
CRP-Compliance-ISO-42001	Compliance	RES	7.4	Optional
CRP-Compliance-NIST-Tier	Compliance	RES	7.2	Optional
CRP-Context-Acceptance	Context	REQ	—	—
CRP-Context-Cache	Context	REQ	4.10	Optional
CRP-Context-Cache-Status	Context	RES	4.11	Optional
CRP-Context-Continuation-Id	Context	BOTH	4.12	Optional
CRP-Context-ETag	Context	RES	4.8	Recommended
CRP-Context-Facts-Used	Context	RES	4.4	Optional
CRP-Context-If-Match	Context	REQ	4.9	Optional

CRP-Context-Protocol-Version	Context	RES	4.15	Required
CRP-Context-Quality-Tier	Context	RES	4.1	Recommended
CRP-Context-Saturation	Context	RES	4.3	Recommended
CRP-Context-Session-Id	Context	BOTH	4.7	Required
CRP-Context-Strategy	Context	RES	4.6	Recommended
CRP-Context-Tokens-Used	Context	RES	4.5	Optional
CRP-Context-Window	Context	RES	4.2	Recommended
CRP-LLM-Grounding-Mode	LLM	REQ	11.2	Optional
CRP-LLM-Reproducibility-Seed	LLM	REQ	11.3	Optional
CRP-Memory-CKF-Community	Memory	RES	9.3	Optional
CRP-Memory-CKF-Hits	Memory	RES	9.2	Optional
CRP-Memory-Knowledge-Age	Memory	RES	9.4	Optional
CRP-Memory-Tier-Hit	Memory	RES	9.1	Optional
CRP-Provenance-Attribution-Score	Provenance	RES	6.6	Recommended
CRP-Provenance-Chain-Integrity	Provenance	RES	6.4	Required*
CRP-Provenance-	Provenance	RES	6.5	Optional

Claim-Count				
CRP-Provenance-DAG-Root	Provenance	RES	6.3	Optional
CRP-Provenance-Fidelity-Score	Provenance	RES	6.7	Recommended
CRP-Provenance-HMAC	Provenance	RES	6.1	Required*
CRP-Provenance-Report-URI	Provenance	RES	6.8	Recommended
CRP-Provenance-Window-HMAC	Provenance	RES	6.2	Recommended
CRP-Provenance-Window-Lineage	Provenance	RES	6.9	Optional
CRP-Safety-Attribution	Safety	RES	5.3	Recommended
CRP-Safety-Contradictions	Safety	RES	5.7	Optional
CRP-Safety-Distortions	Safety	RES	5.6	Recommended
CRP-Safety-Entailment-Score	Safety	RES	5.9	Recommended
CRP-Safety-Fabrications	Safety	RES	5.5	Recommended
CRP-Safety-Grounding-Pct	Safety	RES	5.4	Recommended
CRP-Safety-Hallucination-Risk	Safety	RES	5.1	Required*
CRP-Safety-Hallucination-Score	Safety	RES	5.2	Recommended
CRP-Safety-Mode	Safety	REQ	5.11	Optional

CRP-Safety-Nonce	Safety	BOTH	5.16	Optional
CRP-Safety-Omissions	Safety	RES	5.8	Optional
CRP-Safety-Oversight-Mode	Safety	BOTH	5.10	Recommended
CRP-Safety-Policy	Safety	REQ	5.12	Recommended
CRP-Safety-Report-URI	Safety	REQ	5.13	Optional
CRP-Safety-Retry-After	Safety	RES	5.15	Conditional
CRP-Session-Token	Session	REQ	10.2	Recommended
CRP-Set-Session	Session	RES	10.1	Required*

Table 14

* Required for CRP-Standard and CRP-Full conformance.

Author’s Address

Constantinos Vidiniotis
AutoCyber AI Pty Ltd
Australia
Email: contact@crprotocol.io
URI: https://crprotocol.io