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) — Core Specification
## draft-vidiniotis-crp-core-00

## Abstract

The Context Relay Protocol (CRP) defines a structured, language-agnostic protocol for managing AI context, safety governance, and compliance evidence in deployed large language model (LLM) systems. CRP operates as an HTTP-compatible sidecar protocol, enriching every AI request/response cycle with standardised headers carrying context quality, hallucination risk, provenance integrity, and regulatory classification metadata.

This document defines the foundational axioms, request/response model, sidecar architecture, and the normative relationship between CRP's subsystems: the Context Envelope, Contextual Knowledge Fabric (CKF), Decision Provenance Engine (DPE), and the Audit Chain.

## Feedback

This is a working draft of the CRP Core Specification, published for review and comment. Feedback may be submitted via email to contact@autocyberai.com (mailto:contact@autocyberai.com) or contact@crprotocol.io (mailto:contact@crprotocol.io), or at the CRP GitHub repository at github.com/crprotocol/spec (https://github.com/crprotocol/spec).

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 25 November 2026.



Table of Contents

1. Introduction . . . . . 2
1.1. Background . . . . . 3
1.2. Relationship to Existing Protocols . . . . . 3
1.3. Goals . . . . . 4
1.4. Non-Goals . . . . . 4
2. Terminology . . . . . 5
3. The Ten Axioms . . . . . 5
4. Protocol Architecture . . . . . 6
4.1. Overview . . . . . 6
4.2. Request Flow . . . . . 7
4.3. Memory Hierarchy . . . . . 8
5. Conformance Levels . . . . . 8
6. Security Considerations . . . . . 9
6.1. Header Injection . . . . . 9
6.2. Session Token Security . . . . . 9
6.3. HMAC Chain Integrity . . . . . 9
6.4. LLM Provider Credentials . . . . . 9
7. IANA Considerations . . . . . 9
8. References . . . . . 9
8.1. Normative References . . . . . 9
8.2. Informative References . . . . . 10
Change History . . . . . 10
Author's Address . . . . . 11

## 1. Introduction

### 1.1. Background

Large language model systems deployed in production lack a standardised mechanism for communicating the quality, safety, and compliance state of their outputs to consuming applications, intermediary services, and governance platforms. Each system operator builds bespoke instrumentation to capture hallucination risk, session state, and audit trails — leading to fragmented, non-interoperable approaches.

The Context Relay Protocol addresses this gap by defining:

* A *wire-level header vocabulary* (see CRP-SPEC-002) analogous to HTTP headers, carrying AI-specific metadata on every request/response.

* A *session state relay mechanism* (see CRP-SPEC-007) analogous to HTTP cookies, enabling stateless context continuity.

* A *safety policy directive language* (see CRP-SPEC-006) analogous to Content Security Policy, enabling declarative AI safety enforcement at the transport layer.

* A *provenance and audit chain* (see CRP-SPEC-011) enabling tamper-evident, cryptographically verifiable compliance evidence.

### 1.2. Relationship to Existing Protocols

CRP is designed to complement, not replace, existing AI agent protocols:

| Protocol | Role | CRP Relationship |
|---|---|---|
| MCP (Model Context Protocol) | Tool/resource access for agents | CRP governs the AI calls MCP agents make |
| A2A (Agent-to-Agent) | Inter-agent communication | CRP headers propagate safety state across A2A hops |
| OpenAI API | LLM inference | CRP gateway proxies OpenAI-compatible endpoints |
| HTTP/1.1, HTTP/2 | Transport | CRP headers are carried as standard HTTP header fields |

Table 1

### 1.3. Goals

1. Provide a universal, language-agnostic metadata contract for AI request/response cycles.

2. Enable safety enforcement at the transport layer, not the application layer.

3. Generate continuous compliance evidence without developer instrumentation.

4. Remain compatible with all major LLM providers and agent frameworks.

5. Be implementable as an RFC-based open standard.

### 1.4. Non-Goals

* CRP does not modify LLM model weights or training.

* CRP does not replace application-level business logic.

* CRP does not mandate a specific LLM provider.

* CRP does not define agent behaviour beyond the dispatch interface.

## 2. Terminology

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].

AI call: A single request/response cycle to a large language model.

CRP Gateway: An HTTP-compatible reverse proxy that implements the CRP protocol, sitting between a client application and one or more LLM providers.

Context Envelope: The structured set of facts, knowledge fragments, and instructions assembled by the CRP gateway for injection into an LLM request.

Contextual Knowledge Fabric (CKF): The persistent fact graph (Tier 3 of the CRP memory hierarchy) from which Context Envelopes are assembled.

Decision Provenance Engine (DPE): The CRP module responsible for post-generation analysis of LLM outputs, producing hallucination risk scores, attribution analysis, and provenance records.

Safety Budget: A session-scoped counter representing remaining risk tolerance, decremented by each high-risk AI call. Exposed as CRP-Agent-Safety-Budget.

Window: A single AI call within a continuation chain. Windows are connected in a directed acyclic graph (DAG) for context enlargement across multiple calls.

## 3. The Ten Axioms

CRP’s design is governed by ten foundational axioms. All conformant CRP implementations MUST uphold these axioms.

Axiom 1 — Completeness: The Context Envelope MUST include all factual content necessary for the LLM to answer the query without reliance on parametric memory, where such content exists in the CKF.

Axiom 2 — Accuracy: Facts included in the envelope MUST be drawn from verified source material. The DPE MUST assess output accuracy against envelope content.

Axiom 3 — Relevance: The envelope packing algorithm MUST prioritise

facts by relevance score. Irrelevant facts MUST NOT consume token budget at the cost of relevant facts.

Axiom 4 — Transparency boundary: CRP headers MUST NOT be forwarded to LLM providers. The model MUST remain ignorant of the protocol layer.

Axiom 5 — Oversight capability: All CRP implementations MUST support human oversight triggering. The CRP-Oversight-Mode: halt directive MUST be honoured unconditionally.

Axiom 6 — Resource constraint awareness: The gateway MUST track token budget consumption and expose it via CRP-Context-Tokens-Used and CRP-Context-Window headers.

Axiom 7 — Provenance integrity: Every AI call MUST produce a tamper-evident audit record. HMAC chain integrity MUST be verifiable by any party holding the session key.

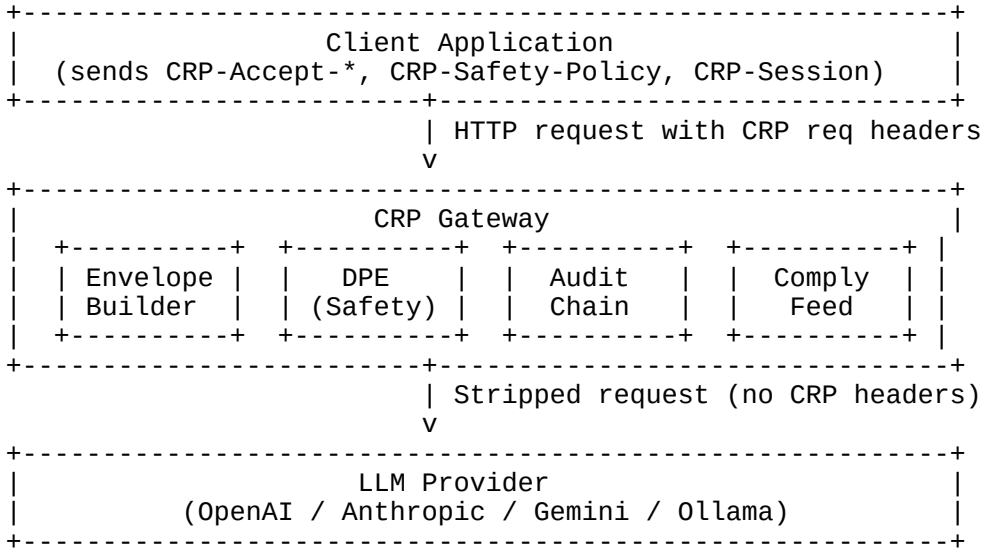
Axiom 8 — Continuity: Continuation sessions MUST preserve context quality across window boundaries. The CRP-Context-Continuation-Id header MUST enable stateless session relay.

Axiom 9 — Regulatory alignment: CRP outputs MUST be classifiable against EU AI Act, GDPR, NIST AI RMF, and ISO 42001 frameworks. Classification MUST be emitted as response headers.

Axiom 10 — Provider neutrality: CRP MUST support any LLM provider exposing an OpenAI-compatible API. Provider selection MUST be transparent to consuming applications.

## 4. Protocol Architecture

### 4.1. Overview




## 4.2. Request Flow

1. Client sends HTTP request to CRP gateway with optional CRP request headers.

2. Gateway authenticates request using CRP API key.

3. Gateway checks CRP-Context-If-Match — returns 304 if ETag matches.

4. Gateway assembles Context Envelope from CKF (3-phase: select, rank, pack).

5. Gateway selects dispatch strategy from CRP-Accept-Strategy or TaskIntent detection.

6. Gateway strips all CRP headers, forwards packed request to LLM provider.

7. LLM provider returns raw completion.

8. Gateway runs DPE pipeline on completion (13 modules).

9. Gateway evaluates completion against CRP-Safety-Policy.

10. If policy violation (e.g., halt-on CRITICAL): returns HTTP 451, fires report-uri webhook.

11. Gateway injects all response CRP headers.

12. Gateway updates HMAC chain, issues updated CRP-Set-Session token.

13. Gateway streams audit event to CRP Comply (if configured).

14. Gateway returns response with CRP headers to client.

### 4.3. Memory Hierarchy

CRP implements a four-tier memory hierarchy:

| Tier | Name | Latency | Persistence | CRP Header |
|---|---|---|---|---|
| 0 | Active (in-context) | <1ms | Call-scoped | CRP-Context-Window |
| 1 | Hot (session cache) | <10ms | Session-scoped | CRP-Context-Session-Id |
| 2 | Warm (recent CKF) | <100ms | Cross-session | CRP-Memory-CKF-Hits |
| 3 | Cold (full CKF graph) | <1000ms | Persistent | CRP-Memory-Knowledge-Age |

Table 2

## 5. Conformance Levels

CRP defines three conformance levels:

CRP-Basic: Implements core headers (CRP-Context-Quality-Tier, CRP-Safety-Hallucination-Risk, CRP-Provenance-HMAC), session tokens, and HTTP 451 halt. This is the minimum viable governance level.

CRP-Standard: Implements all 58 headers, Safety Policy directives, ETag caching, agentic dispatch headers, and compliance headers. Required for CRP Comply integration.

CRP-Full: Implements all of Standard plus streaming safety enforcement, stop-sequence injection, multi-agent safety budget propagation, and SIEM export. Required for CRP Certification.

## 6. Security Considerations

### 6.1. Header Injection

CRP headers on responses MUST be generated by the CRP gateway, not by LLM output. Implementations MUST validate that no CRP-prefixed headers are present in raw LLM responses before injection.

### 6.2. Session Token Security

The CRP-Set-Session token MUST be signed with HMAC-SHA256 using a session key not derivable from the token payload. Tokens MUST include an expiry and MUST NOT be accepted after expiry.

### 6.3. HMAC Chain Integrity

The HMAC chain MUST be computed as: HMAC-SHA256(window_content || previous_HMAC, session_key). Any break in the chain (verified via CRP-Provenance-Chain-Integrity: BROKEN) MUST trigger an audit incident.

### 6.4. LLM Provider Credentials

CRP gateways that vault LLM provider credentials MUST store keys encrypted at rest. Client applications MUST NOT be required to hold LLM provider credentials when using a CRP gateway.

## 7. IANA Considerations

This document requests registration of the CRP- prefix in the HTTP Field Name Registry at https://www.iana.org/assignments/http-fields (https://www.iana.org/assignments/http-fields) per [RFC9110] Section 16.3.

A complete list of headers for registration is provided in CRP-SPEC-002 (Header Specification).

## 8. References

### 8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

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

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

[RFC6265] Barth, A., "HTTP State Management Mechanism", RFC 6265, DOI 10.17487/RFC6265, April 2011, <https://www.rfc-editor.org/info/rfc6265>.

## 8.2. 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.

## Change History

| Version | Date | Changes |
|---|---|---|
| 1.0.0 | 2024-01-01 | Initial protocol release |
| 2.0.0 | 2024-06-01 | DPE integration, HMAC chain |
| 3.0.0 | 2026-05-24 | Header specification, Safety Policy, Session Token |

Table 3

Author’s Address

Constantinos Vidiniotis
AutoCyber AI Pty Ltd
Email: contact@crprotocol.io