Independent Submission
Internet-Draft
Intended status: Experimental
Expires: 27 September 2026

A. Sogomonian, Ed.
Artificial Intelligence Internet Foundation (AIIF)

26 March 2026

Architecture for the Artificial Intelligence Internet Protocol
draft-sogomonian-aiip-architecture-04

Abstract

This document defines the architectural model for the Artificial Intelligence Internet Protocol (AIIP). AIIP defines a dedicated autonomous access plane for execution-capable systems, enabling delegated, stateless execution of real-world actions using a resolve-invoke-execute-receipt pattern over authenticated transports.

AIIP is not a discovery, registration, orchestration, or control-plane protocol. It defines the architectural boundary at which autonomous execution is recognized and cryptographically verifiable through execution receipts.

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 27 September 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

1. Introduction

The Internet has historically been optimized for human interaction, document retrieval, and request-response semantics. Autonomous systems, however, require a different access model, one focused on delegated execution, safety boundaries, and post-execution accountability.

AIIP provides a uniform architectural model for invoking and executing actions by autonomous systems with verifiable outcomes. The architecture follows a resolve-invoke-execute-receipt pattern.

Native AIIP communication is designed for machine-to-machine operation and does not assume HTTP semantics, browser compatibility, or synchronous request-response behavior. HTTPS gateways are optional and non-authoritative.

From the perspective of autonomous systems, AIIP functions as a native entry point to Internet capabilities, separate from the human-oriented Web access model.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Scope and Non-Goals

AIIP defines:

* Invocation semantics for autonomous actions
* Execution boundaries for state-changing operations
* Cryptographically verifiable execution receipts
* Operation under intermittent or absent connectivity
* Delegation-based authorization for execution attempts

AIIP explicitly does NOT define:

* Agent registration or discovery
* Capability catalogs or negotiation
* Workflow orchestration or task planning
* Real-time control loops or enforcement
* Human-facing user interface behavior

3. AIIP Autonomous Access Plane

AIIP defines a distinct autonomous access plane for execution-capable systems. This access plane is separate from the human-oriented Web and independent of control-plane or discovery mechanisms.

An AIIP access point is a network-visible execution interface that accepts invocation requests, validates authorization, performs or brokers execution, and produces verifiable execution receipts.

AIIP provides a distinct network access model for autonomous systems. In this model, autonomous systems interact through AIIP-native invocation and execution semantics rather than through HTTP-based interfaces designed primarily for human users.

The AIIP autonomous access plane begins at invocation and ends at execution receipt generation. Any function operating entirely outside this boundary is outside the scope of AIIP.

AIIP does not extend HTTP. It defines a native, machine-oriented execution plane that may be carried over authenticated transports such as TLS [RFC8446] or QUIC [RFC9000].

AIIP is designed to coexist with discovery, registration, policy, and orchestration mechanisms, which may provide inputs to AIIP without constraining its execution semantics.

4. Architectural Layering

AIIP occupies a distinct architectural layer between higher-level autonomous decision logic and authenticated transport. This can be expressed conceptually as follows:

* Application, orchestration, and planning layers (out of scope)

* Discovery and registration mechanisms (out of scope)

* AIIP autonomous access plane (this specification)

* Authenticated transport (for example, TLS or QUIC)

Discovery and registration mechanisms may provide identifiers, endpoints, or trust material consumed by AIIP, but AIIP does not depend on their semantics.

5. Architectural Model

AIIP follows a resolve-invoke-execute-receipt pattern.

5.1. Resolve

Prior to invocation, an autonomous system may obtain an execution target, access point identifier, trust material, or delegation grant by means outside the scope of this specification.

5.2. Invoke

The invoking system submits a cryptographically authenticated request to attempt an action under a delegation grant.

5.3. Execute

The receiving environment validates the invocation context and, if authorized and feasible, performs the requested state-changing operation.

5.4. Receipt

Upon execution, the environment generates an execution receipt that provides verifiable evidence of the execution outcome.

6. Delegation and Authorization

AIIP assumes that execution attempts occur under delegated authority. A delegation grant authorizes an autonomous system to attempt specific actions against a defined scope under stated constraints.

Delegation grants SHOULD be narrowly scoped, time-limited, and bound to an identified subject. Implementations MAY include additional constraints such as maximum executions, time windows, geographic restrictions, or environmental preconditions.

Grant encoding and distribution are out of scope for this document. However, the architecture assumes that the executing environment is able to verify grant authenticity and applicability before execution.

7. Invocation, Execution, and Architectural Boundary

An invocation is a cryptographically authenticated request to attempt an action under a delegation grant.

An invocation message typically includes a target identifier, an action, input parameters, a delegation grant or reference to one, and cryptographic metadata such as timestamps, nonces, or signatures. The exact encoding is out of scope for this document.

Execution is the successful completion of a state-changing operation by the target environment.

AIIP defines the sole architectural boundary at which autonomous execution is recognized. An action lacking a valid AIIP execution receipt MUST NOT be treated as executed for protocol purposes.

8. Execution Receipts

Each successful execution MUST produce an execution receipt. A failed execution attempt MAY also produce a signed receipt or equivalent verifiable outcome record.

Receipts provide cryptographic evidence that an execution occurred, together with the execution outcome as determined by the executing environment.

A receipt typically includes an invocation reference, execution status, result commitment such as a hash, timestamp, identity of the executing environment, and a cryptographic signature.

Receipts enable auditability, compliance, dispute resolution, and post hoc verification.

Receipt encoding is out of scope for this document. Semantic validity is independent of serialization format.

Canonicality

Receipt validity MUST NOT depend on field ordering, whitespace, presentation details, or other serialization artifacts.

9. Transport Considerations

AIIP messages MUST be carried over an authenticated transport or protected by equivalent cryptographic mechanisms that provide peer authentication, integrity protection, and confidentiality appropriate to the deployment environment.

This document does not define a mandatory single transport binding. Example transports include TLS [RFC8446] and QUIC [RFC9000]. Future specifications may define concrete bindings and negotiation behavior.

Transport success alone does not constitute execution. Protocol-level recognition of execution depends on receipt generation and validation.

10. Offline and Disconnected Operation

AIIP supports authorized execution in the absence of continuous connectivity. Offline operation represents execution under pre-authorized constraints, not deferred transmission or request caching.

Implementations supporting offline execution SHOULD bound that capability using explicit validity intervals, execution limits, environmental constraints, or comparable controls.

Execution receipts generated during offline operation MUST remain verifiable once connectivity is restored.

11. Optional Gateways

AIIP gateways MAY translate between AIIP and non-AIIP systems, including HTTP-based systems.

Gateways are non-authoritative with respect to execution. A gateway MUST NOT alter the semantic content of a receipt and MUST NOT claim that transport delivery alone is equivalent to execution.

Unless explicitly acting as the execution environment itself, a gateway MUST NOT generate an authoritative execution receipt on behalf of a different execution environment.

12. Future Work

Future work may define concrete transport bindings, receipt encodings, deployment profiles, registries, media types, URI schemes, or other operational extensions needed to support interoperable AIIP deployments.

Such extensions require separate specification and community review.

13. Security Considerations

AIIP relies on authenticated transports and cryptographic mechanisms to provide integrity, authenticity, and non-repudiation properties for invocations and execution receipts.

Implementations MUST consider threats including replay of invocation messages, forgery or misuse of delegation grants, impersonation of invoking agents or execution environments, tampering with receipts or result commitments, abuse of offline execution authority beyond its intended scope, and unauthorized modification or observation by intermediary systems.

Implementations SHOULD use nonces, timestamps, audience restriction, narrow grant scope, expiration controls, and cryptographic signatures to mitigate replay and forgery risks.

Execution environments MUST verify that the presented authorization is applicable to the requested action and target before execution.

Because gateways are non-authoritative, trust decisions MUST be bound to the actual execution environment and its receipt rather than to an intermediary that merely relays or translates requests.

14. Privacy Considerations

AIIP deployments can expose information about agent identity, delegated authority, execution timing, execution targets, and action outcomes. Correlation of receipts or invocation metadata across contexts can reveal behavior patterns even when payload content is protected.

Implementations SHOULD minimize unnecessary disclosure, limit retained metadata, and apply data minimization principles to both invocations and receipts. Deployments MAY use privacy-enhancing techniques such as selective disclosure, compartmentalized identifiers, or zero-knowledge mechanisms where suitable.

15. IANA Considerations

This document has no IANA actions.

16. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", 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", RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC9000] Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

17. Informative References

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <https://www.rfc-editor.org/info/rfc3552>.

[RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <https://www.rfc-editor.org/info/rfc7258>.

Author’s Address

Aram Sogomonian (editor)
Artificial Intelligence Internet Foundation (AIIF)
United States of America
Email: aiinternetfoundation@icloud.com