ecma INTERNATIONAL

Standard ECMA-433

1st Edition / December 2025

Binding of the Natural Language Interaction Protocol (NLIP) over AMQP

Standard

Rue du Rhône 114 CH-1204 Geneva T: +41 22 849 6000 F: +41 22 849 6001

Contents

Page

Introduction

The technology of Artificial Intelligence (AI) has the potential to be truly transformative to society. Despite some limitations, the technology is capable of many functions, including but not limited to answering questions, translating, describing and summarizing multi-modal content, generating new content, and summarizing large volumes of information. This enables the creation of intelligent agents that can use AI to analyze data and provide new services.

A much bigger boost to the social benefits of AI technology can be obtained by interaction among different intelligent agents, which may be under the control of different organizations and users. The interaction among intelligent agents can unlock new economic and social value, just like the interactions among various Internet-based services was enabled with the advent of the web browser.

For the intelligent agents to interact with each other, there is a need for a standard common protocol that is used widely among interacting agents. This Standard specifies such a protocol which would ensure interoperability among various services that use AI based technology.

ECMA-430 defines the Natural Language Interaction Protocol (NLIP).

This Standard describes the binding of NLIP protocol to a base transfer protocol of AMQP.

This Ecma Standard was developed by Technical Committee 56 and was adopted by the General Assembly of December 2025.

Binding of the Natural Language Interaction Protocol (NLIP) over AMQP

1 Scope

This Standard defines how the Natural Language Interaction Protocol (NLIP) should be implemented over the base transfer protocol of AMQP. The exemplar use-cases for NLIP implementation over AMQP are out of scope of this Standard.

2 Conformance

A conforming implementation shall provide and support all types of messages and submessage along with the semantics defined in the NLIP specification ECMA-430 , and transfer them using AMQP.

A conforming production deployment of NLIP must select a security profile defined in ECMA-434 and support it.

3 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ECMA-430, *Natural Language Interaction Protocol (NLIP)*

ECMA-434, *Security Profiles for the Natural Language Interaction Protocol (NLIP)*

IETF RFC 2119, *Key words for use in RFCs to Indicate Requirement Levels*
[https://datatracker.ietf.org/doc/rfc2119]

IETF RFC 4422, *Simple Authentication and Security Layer (SASL)*. [https://datatracker.ietf.org/doc/rfc4422]

ISO/IEC 19464:2014, *Advanced Message Queuing Protocol (AMQP) v1.0 specification*

4 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

4.1
NLIP
NLIP or Natural Language Interaction Protocol is the protocol defined in ECMA-430.

4.2
base transfer protocol
a transfer protocol is a communication protocol between two computer systems which supports an encrypted and authenticated transfer of data across those computer systems.

4.3
NLIP end-point
an NLIP end-point is a computer system or computer software that communicates using NLIP with other computer systems or computer software, on the same or another machine.

4.4
AMQP connection end-point
an AMQP connection end-point is a computer system supporting the AMQP protocol that originates or receives AMQP messages, as defined in ISO/IEC 19464:2014 Section 2.4.

4.5
AMQP intermediary
an AMQP intermediary is a computer system that relays AMQP messages between two or more AMQP end-points, as defined in ISO/IEC 19464:2014 Section 3.2.

4.6
agent
an agent is a software which is capable of processing multimodal information, take actions and generate responses based on such processing using an AI model such as a large neural network. The action and response generation may or may not involve a human in the loop.

4.7
client agent
a client agent is an agent which initiates the transfer of data to another agent.

4.8
server agent
a server agent is an agent which waits for other agents to initiate the transfer of data with it and responds to them.

4.9
Agent AMQP Address
the AMQP address of an agent is the AMQP terminus address associated with that agent as defined in ISO/IEC 19464:2014 Section 3.2.15.

5 Notational conventions

In this Standard, the following conventions that are consistent with IETF RFC 2119 are used:

- “Shall” indicates that the item is an absolute requirement of the specification
- “Should” indicates that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
- “May” indicates that that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option shall be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option shall be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

6 NLIP Mapping to AMQP

6.1 Message Content Mapping

NLIP messages are mapped onto AMQP messages one-to-one. One NLIP message, regardless of the number of sub-messages it contains, is carried in one AMQP message.

An AMQP message consists of up to seven sections. Three of these sections: properties, application-properties, and data, are relevant to transport of NLIP messages over AMQP.

The other four sections are the header, delivery-annotations, message-annotations, and footer. They may be used in any way that the situation calls for (i.e. to provide reliable delivery, to set priority, to sign the bare message, etc.).

The fields listed in Table 1 are the only fields that are relevant to the implementation of the NLIP protocol.

Table 1 – Fields in the protocol binding

Section	Field
properties	user-id
properties	to
properties	reply-to
properties	correlation-id
properties	content-type
application-properties	nlip-reply-<format>
application-data	data

6.1.1 Properties.user-id

This field is optional, but if it is present, it shall contain the authentication identity of the user that originated the message. It may contain a user identity or a token.

If the AMQP connection is direct between two NLIP end-points with one being a client agent and the other being a server agent, the AMQP connection end-point of the server agent shall enforce this requirement by authenticating the client's identity via the SASL protocol.

If the message passes through AMQP intermediaries, the connection to the server agent is not from the client agent, but from the last AMQP intermediary. Therefore, the authenticated identity of the connection client is *not* the same as the originator of the message.

The first AMQP intermediary directly connected to the client agent shall ensure that the identity in the user-id field of AMQP message matches the authenticated identity for the connection from the originating client.

6.1.2 Properties.to

This field contains the agent AMQP address of the recipient of the message.

For a request message, this is the agent AMQP address of the server agent. For a response message, this is the agent AMQP address of the original requestor or a designated agent AMQP address for the response.

6.1.3 Properties.reply-to

The reply-to field is used in a message for which there will be an eventual response. The field contains the agent AMQP address of the agent who is to receive the response. Typically, this is a dynamic address allocated to the original requestor.

A server agent that receives a request copies the reply-to from the request into the 'to' field of the response. See Annex A.6 for more details.

6.1.4 Properties.correlation-id

The correlation-id field may be used by a client agent to positively correlate received responses to their requests. A server agent that receives a request with a correlation-id shall copy the contents of the correlation-id field in the request to the correlation-id field of the response.

6.1.5 Properties.content-type

The content-type field shall correctly describe the format of the NLIP message data.

If the message data is encoded using AMQP encoding as defined in ISO/IEC 19464:2014 Section 1.2, the content-type shall be "message/x-amqp-list".

If the message data is in JSON format, the content-type shall be "application/json".

6.1.6 Application-properties

Application-properties is an optional message section that carries a map of application-specific key/value pairs. This binding enumerates NLIP application properties used for multi-format replies (see Clause 6.2).

6.1.7 Application-data

The application data section (see ISO/IEC19464:2014 Section 3.2) is where the NLIP payload is encoded. The NLIP payload is encoded as an AMQP Sequence section (ISO/IEC19464:2014 Section 3.2.7) or an AMQP Value section (ISO/IEC19464:2014 Section 3.2.8) in which the value is a list of structured objects. In both cases, the payload is encoded using AMQP Type Encodings (ISO/IEC19464:2014 Section 1.2).

The Content-Type field must contain "message/x-amqp-list".

Alternatively, the NLIP payload may be encoded as a JSON string. In this case, it is encoded in a Data section (ISO/IEC19464:2014 Section 3.2.6). The Content-Type field must contain "application/json".

6.2 Multi-Format Replies

There is a request-reply pattern that is unique to NLIP whereby a requestor may wish to receive the response in more than one format. An example of this is a response that is generated both in machine-readable JSON format and in natural-language text for human consumption or for audit logging. Furthermore, these different responses may need to be delivered to different consumers in the network.

This is achieved by adding items to the application-properties map using a key of the form:

```
nlip-reply-<content-type>
```

The value referenced by the key is an address to be used for replies of the indicated content-type. The content-type should be used as defined in ISO/IEC19464:2014 Section 3.2.4.

The nlip-reply fields serve two purposes: They indicate to the receiver which formats are desired for responses; and they provide overrides for the `reply-to` field in properties. If an agent acting as an NLIP server receives a request with one or more nlip-reply fields in the application-properties, it shall generate responses in each of the indicated content-types and use the NLIP-reply addresses as destinations for their respective response messages.

Annex A
(informative)

Introduction to AMQP

Advanced Message Queueing Protocol (AMQP) is an ISO-standard, enterprise-caliber message transfer protocol with a rich set of capabilities that make it well suited for use in complex and large-scale distributed software systems.

Aside from the handshakes used to establish security, AMQP is symmetric and asynchronous.

A.1 Protocol Symmetry

Once a transport connection is established between two peer endpoints, there is no distinction in role between those endpoints. Whereas HTTP has client and server roles and other messaging protocols have client and broker roles, AMQP is simply peer and peer. This means that both sides of an AMQP connection are equally able to send and receive messages when and as they see fit.

A.2 Asynchrony

AMQP communication is not characterized by temporal relationships between sent messages. There is no requirement that any sent message be followed by a response. Asynchrony gives flexibility to application architects and allows for high performance data transfer in a wide range of applications.

A sent message may not require a response. Or a sent message may result in multiple responses. Many messages can be sent and their responses collected over time as they are completed. There is no requirement that responses be returned in the same order in which the requests were sent. AMQP has the ability to explicitly route responses and to correlate responses to requests.

A.3 Addressing

AMQP provides the ability to assign an address to a terminus. A terminus is the endpoint of a link and is either a producer or a consumer of messages. An AMQP connection in a running agent may have many termini associated with it.

Unlike an IP address, which designates a single host, the AMQP address designates a fine-grained source or destination inside a running process.

Because the AMQP address exists at the application layer, and is not associated with any host address, it allows communication to occur at a higher level of abstraction than that offered by TCP/IP. In an intermediated topology, where agents are not directly connected to each other but communicate via intermediate processes, agents on hosts that cannot communicate via TCP/IP can interchange messages via AMQP. This is particularly powerful in hybrid/multi-cloud environments where communication is traditionally difficult. For example, an agent hosted in the private data center of one enterprise cannot communicate with another agent hosted in the private data center of another enterprise. An AMQP network involving a public, or mutually reachable, intermediary can facilitate the needed communication with fine granularity. There is no need to open networks or hosts with a VPN.

The semantics of AMQP addressing assume that there will be multiple termini with the same addresses. This opens the possibility of anycast and multicast routing. Anycast is used to balance workload across multiple servers. Multicast is used to efficiently distribute data to multiple destinations.

With this addressing flexibility, an AMQP-routed network makes multitenancy very easy. A new set of addresses can be defined for each new tenant that are completely unknown to existing tenants. This prevents leakage, probing, and other related insecure activities.

A.4 Direct vs. Intermediated Communication

AMQP is ultimately a message-oriented exchange protocol for point-to-point use. It runs over a reliable network transport like TLS/TCP. As such, it can be used to directly connect NLIP agents with one acting as the TCP client and the other as the TCP server.

AMQP, however, was designed to facilitate intermediated message transfer. The traditional form of this is to use a message broker to store and forward messages in queues and topics. But because the AMQP protocol is symmetric and does not require a message server (broker), it can also be used with intermediary routers. An AMQP router provides a way to create a network over which AMQP messages can be transferred end-to-end, possibly through multiple hops, with the same reliability and delivery guarantees that can be had with direct communication.

Brokers are useful when temporal disconnect is desired (i.e. the producer and consumer are not present at the same time). Routers are useful when store-and-forward is not needed and a more flexible, faster, and lighter weight medium is desired.

A.5 Sessions and Links for Flow Control and Multiplexing

The structure of AMQP starts with the Connection. An AMQP connection is a reliable transport connection, typically TCP (Transport Control Protocol). There are two standard security layers that can be used to enhance the security of the AMQP connection: TLS (Transport Layer Security) is used for encryption and authentication of one or both endpoints of the connection; SASL (Simple Authentication and Security Layer) can be further used to authenticate the client-side participant in the AMQP connection.

Sub-structure of the AMQP connection consists of Sessions, Links, and Frames. Connections contain sessions, sessions contain links, and links carry frames. Messages are transferred in one or more frames on a link.

Sessions provide a sliding-window flow control mechanism that is tied to the number of frames that a receiver is willing to accept. Since frames have a maximum size, session flow control can be based on the amount of memory that an application is willing to allocate to message traffic for a specific purpose. Since multiple sessions can be established, each with its own allocated buffer memory, multiple classes of traffic can be transferred over an AMQP connection such that congestion in one class does not affect the flow of traffic in another class. This is very useful for systems in which control traffic must flow regardless of the volume of concurrent bulk transfers. It is also useful in edge applications where memory may be scarce.

Links are unidirectional and represent a single stream of message traffic. A link has two termini, one a source and one a target. Message traffic flows from source to target. Links provide a credit-based flow control in which the target indicates how many messages it is willing to receive. Since there is no limit to the size of a message, link flow control cannot be tied to available memory. It is used to meter the flow of traffic over that link. Links also provide varying levels of delivery guarantee for messages. Message transfer can be best effort, in which the sender is not interested in knowing whether the message was received. Transfer can also be at-least-once, in which the sender is notified as to whether or not the message was accepted by the receiver. Exactly-once transfer is like at-least-once, but guarantees that the receiver is not given duplicate copies in the event the message was re-transmitted.

Frames are used to break large messages into manageable fragments. The framing capability of AMQP allows message delivery over links and sessions to be interleaved. This provides multiplexing of multiple message transfers. The transfer of a very large message on one link will not block the transfer of a smaller message on another link. The frames of both messages will be interleaved across the connection.

A.6 To, Reply-To, and Dynamic Addresses

AMQP supports a variety of different messaging patterns including request/reply, where one or more reply messages are sent as a result of receiving a request message. All the patterns involve sending a message to a specific logical destination (the to address).

In cases where a reply is desired, the request message contains a reply-to address to designate where the replies are to be sent. A requestor needs to establish a receiving link on which to receive replies. The requestor can use some form of unique address (a UUID) for the reply. Alternatively, the requestor can use a *dynamic* address for the reply link's target terminus. In this case, the requestor sets the dynamic flag when establishing the link. This instructs the connected peer to create a unique, dynamic address for the link. This address can then be used in the reply-to field in the request message. Dynamic addresses are particularly useful when running in intermediated AMQP networks where the intermediary can create a temporary address that is routable from all other parts of the network.