ecma INTERNATIONAL

Standard ECMA-432

1st Edition / December 2025

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

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

is the registered trademark of Ecma International

COPYRIGHT PROTECTED DOCUMENT

© Ecma International 2025

Contents

Page

© Ecma International 2025

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 interactions 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.

There is a need for a standard common protocol that is used by humans to interact with an intelligent agent, and for intelligent agents to interact with each other. ECMA-430 specification proposes such a protocol, the Natural Language Interaction Protocol (NLIP).

This Standard describes the binding of NLIP protocol to a base transfer protocol which is using WebSocket.

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

© Ecma International 2025

COPYRIGHT NOTICE

© 2025 Ecma International

By obtaining and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions.

This document may be copied, published and distributed to others, and certain derivative works of it may be prepared, copied, published, and distributed, in whole or in part, provided that the above copyright notice and this Copyright License and Disclaimer are included on all such copies and derivative works. The only derivative works that are permissible under this Copyright License and Disclaimer are:

(i) works which incorporate all or portion of this document for the purpose of providing commentary or explanation (such as an annotated version of the document),

(ii) works which incorporate all or portion of this document for the purpose of incorporating features that provide accessibility,

(iii) translations of this document into languages other than English and into different formats and

(iv) works by making use of this specification in standard conformant products by implementing (e.g. by copy and paste wholly or partly) the functionality therein.

However, the content of this document itself may not be modified in any way, including by removing the copyright notice or references to Ecma International, except as required to translate it into languages other than English or into a different format.

The official version of an Ecma International document is the English language version on the Ecma International website. In the event of discrepancies between a translated version and the official version, the official version shall govern.

The limited permissions granted above are perpetual and will not be revoked by Ecma International or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and ECMA INTERNATIONAL DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

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

1 Scope

This specification defines how the Natural Language Interaction Protocol (NLIP) shall be implemented over the WebSocket protocol using CBOR (Concise Binary Object Representation, RFC 8949) for compact and efficient multimodal communication. It also describes a fallback to UTF-8 encoded JSON text frames for compatibility.

2 Conformance

A conformant implementation shall:

- Support full NLIP message schema as defined in the ECMA-430 – Natural Language Interaction Protocol.
- Encode/decode messages in CBOR format over binary WebSocket frames.
- Optionally fall back to UTF-8 JSON text frames for non-CBOR-capable peers.
- Support transmission of multimodal submessages, including raw binary content (e.g., audio, image).
- 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 6455, *WebSocket Protocol*
[https://datatracker.ietf.org/doc/html/rfc6455]

IETF RFC 7049, *CBOR Data Model*
[https://datatracker.ietf.org/doc/html/rfc7049]

IETF RFC 7230, *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*
[https://datatracker.ietf.org/doc/rfc7230/]

IETF RFC 7240, *Prefer Header for HTTP*
[https://datatracker.ietf.org/doc/rfc7240/]

IETF RFC 8949, *Concise Binary Object Representation (CBOR)*
[https://datatracker.ietf.org/doc/html/rfc8949]

© Ecma International 2025

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
CBOR
Binary serialization format for structured data.

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 WebSocket endpoint

The server end-point shall expose a WebSocket endpoint at:

wss://<host>:<port>/nlip/ws

A conformant implementation of NLIP over WebSocket will have the server end-point accessible using a URL defined as wss://<server_name>:port/nlip/ws.

6.1 Optional fallback endpoint

In case of server end-points that do not support CBOR, the server end-point may be accessible using a URL defined as wss://<host>:<port>/nlip/ws/text

7 Message transmission

Implementations may specify a maximum frame size that they support.

© Ecma International 2025

7.1 CBOR format

- Each WebSocket binary message or its constituent fragments shall contain a single NLIP message encoded in CBOR.
- CBOR Content fields may include:
  - String
  - Byte string (raw binary)
  - Array, Object (Map)
- Submessages are embedded in CBOR using the same schema.

7.2 Text fallback format

- If CBOR is not supported:
  - Each WebSocket binary message or its constituent fragments shall use UTF-8 encoded JSON in WebSocket text frames.
  - Binary data shall be base64-encoded.

8 Examples

This clause provides some normative examples of NLIP messages carried over WebSocket.

Example 1: text + audio (CBOR)

NLIP message (in Python pseudo-code before CBOR encoding)

```
{
  "MessageType": "Request",
  "Format": "structured",
  "Subformat": "application/json",
  "Content": {"intent": "weather_query"},
  "Submessages": [
    {
      "Label": "transcription",
      "Format": "text",
      "Subformat": "en-US",
      "Content": "What's the weather in Austin tomorrow?"
    },
    {
      "Label": "audio",
      "Format": "binary",
      "Subformat": "audio/wav",
      "Content": b'\x52\x49\x46\x46...'  # Raw binary WAV
    }
  ]
}
```

This is encoded as a **single CBOR binary frame**. The audio submessage uses a byte string directly and no base64 encoding.

© Ecma International 2025

Example 2: image processing request (CBOR)

```
{
  "MessageType": "Request",
  "Format": "binary",
  "Subformat": "image/jpeg",
  "Content": b'\xff\xd8\xff\xe0...',  // JPEG binary
  "Submessages": [
    {
      "Label": "description",
      "Format": "text",
      "Subformat": "en",
      "Content": "Process this image for defects"
    }
  ]
}
```

CBOR encoding allows this entire object to be transmitted compactly.

Example 3: text fallback (JSON over WebSocket text frame)

```
{
  "MessageType": "Request",
  "Format": "binary",
  "Subformat": "audio/wav;base64",
  "Content": "UklGRngAAABXQVZFZm10IBAAAAABAAEAESsAACJWAAACABAAZGF0YYAA...",
  "Submessages": [
    {
      "Label": "transcription",
      "Format": "text",
      "Subformat": "en-US",
      "Content": "What’s the current stock price of Tesla?"
    }
  ]
}
```

9 Message handling and framing

Feature	CBOR Frame	Text Fallback
Frame Type	Binary	Text
Encoding	CBOR	UTF-8 JSON
Binary Data Support	Native (byte string)	Base64 in JSON
Compression Support	Optional (via permessage-deflate)	Optional
Streaming Support	Chunking via multiple frames	Limited

10 Session management (optional)

- Each message may include a session ID in "MessageType" or custom field.

© Ecma International 2025

- An end-point may use WebSocket heartbeat for liveliness checks.
- An end-point may manage session states using a separate message with MessageType ”Control".

11 Error handling

- If CBOR decoding fails, a server end-point shall:
  - Send back a NLIP error message using the fallback endpoint.
  - Example:

```
{
 “MessageType”:”Error”,
  "Format": "text",
  "Subformat": "English",
  "Content": "CBOR decoding failed. Fallback to text recommended."
}
```

© Ecma International 2025

© Ecma International 2025