

DMSC Working Group
Internet-Draft
Intended status: Informational
Expires: 14 August 2026

D. Verma
IBM
E. Bertino
Purdue University
A. Ratnaparkhi
eBay Inc.
S. Hughes
Service Now
L. Xing
Z. Li
X. Liao
J. Cui
University of Illinois at Urbana-Champaign
R. Topaloglu
Marist University
M. Rahouti
Fordham University
10 February 2026

Use of Natural Language for Agent Communication
draft-verma-dmsc-nlip-notes-00

Abstract

In current communication protocol designs, the prevalent practice is to define a strict Application Programming Interface (API) for different types of interactions among software entities. There is an explosion of APIs which makes interoperability hard, and the standards for interoperability fragile. Specifically for agent to agent communications, defining strict APIs for interactions such as registration, discovery, telemetry and debugging creates significant operational challenges. In this draft, we argue that a flexible design option -- where generative AI is used to create a flexible communication protocol for APIs, can lead to a common and flexible way for a single uniform API to be used between agents, and enables a better communication paradigm.

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 14 August 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. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

| | |
|--|---|
| 1. Conventions used in this document | 2 |
| 2. Terminology | 2 |
| 3. Introduction | 3 |
| 4. Other Modalities | 4 |
| 5. Existing Standard | 5 |
| 6. New Conventions | 5 |
| 7. Security Considerations | 6 |
| 8. IANA Considerations | 6 |
| 9. Acknowledgement | 6 |
| 10. References | 6 |
| 10.1. Normative References | 6 |
| 10.2. Informative References | 7 |
| Authors' Addresses | 7 |

1. Conventions used in this document

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 RFC 2119 [RFC2119] .

2. Terminology

The following terms are defined in this document:

- * NLIP: Natural Language Interaction Protocol

- * ntohs: Network to Host Short
- * htons: Host to Network Short

3. Introduction

Contemporary application-level protocols and Application Programming Interface (API) are designed by specifying a function name (or URI) and a data structure. All communicating endpoints are required to use the same data structure, and this data structure is transferred between them using a structured representation such as JSON. This design approach causes significant interoperability issues when the version upgrade of any software causes a change in the data structure that needs to be exchanged.

A more flexible communication paradigm can be obtained by leveraging the ability of large language models to translate between unstructured natural language and a structured representation such as JSON or the internal structure of a programming language effectively. The communicating parties can transmit a natural language representation among each other, and each endpoint can maintain its own internal representation for the structure being maintained for its tasks.

In this draft, we argue that this flexible exchange offers various benefits in agent to agent communications.

The natural language text exchanged between communicating parties has the same semantics as the structures maintained at various communication endpoints, but the format is natural language. A LLM at each end point can translate the wire format into the local structure format. This allows for the internal representation of structures at endpoints to be independent of each other. This independence allows for improved interoperability and the freedom for each end-point to upgrade its internal structures as needed.

As an example, let us consider the simple but illustrative case where one of the endpoints is an agent that is sending its communication endpoint information to the other endpoint which is a registrar providing registration services for agents. The registrar maintains information about each endpoint as a URL, while the agent maintains its internal information as a local host and port. The agent can register itself to the registrar using plain text - "I am running on host hostname on port 1430" or an equivalent natural language text representation. The registrar can use a local LLM to translate this text to an internal URL representation.

Using natural language provides two key advantages. It provides resilience against upgrades of the endpoints. Suppose the registrar is upgraded to a new version in which it supports each agent information as a structure consisting of a port, a protocol and a host address instead of the URL. This upgrade can be made without any impact to any of the agents. On the other hand, if the traditional approach of defining a fixed structure on the wire were to be used, the upgrade of the registrar would require changes in each of the agents interacting with it.

The second advantage is the simplification in the versioning of protocols. During any definition of a standard protocol, some key features or functions may be missed due to a variety of reasons. If a new protocol version is defined, the protocol needs to be defined to support a version number and a careful handling of mismatches in the structural differences across versions. By breaking the linkage between the internal representation of endpoint structures and the wire structure, natural language simplifies version management significantly.

The separation of the wire format from the internal structures can be viewed as a modern take on the concept of `ntohs` and `htons` - concepts developed during the early stages of computer communications development to promote interoperability between computer systems. When big-endian machines needed to talk to little-endian machines, these two macros translated between network and host structure formats. NLIP is providing the same functionality at the application level, leveraging the ability of LLMs to translate between unstructured natural text and structured representations.

It is recommended that two communicating parties exchange information about their internal representation with each other. When an agent registers with the registrar in the above example, the registrar can send a natural text representation of the internal information to the agent to validate that it has translated the text correctly to an internal representation and then translated it back. The agent can do its own internal translation and validate that the information is consistent with its internal representation. The same exchange also helps to validate if a field is missing or not incorporated properly.

4. Other Modalities

Natural language is not the only modality which is needed for general communication among software entities. Some agents may require other modality such as images, video or audio/speech as part of their operation. Nevertheless, the same principle of separating the wire format from the internal structures and capabilities of the software application can be used.

For each modality, the wire exchange format can specify the modality of the exchange. However, it can leave the task of interpreting the contents and internal structure of the exchanged information to the local AI model. The agent can use its local AI model to interpret the format and process it.

This is an analogue of the way humans communicate using the different senses. Our eyes process light/visual modality, our ears process the sound/speech modality etc. Our internal intelligence processes these signals without relying on the strict adherence to internal structures. We can simply identify the information modality and let the agent process the information.

5. Existing Standard

There is an existing standard which is designed based on this principle of using natural language for exchange among agents. This standard - Natural Language Interaction Protocol (NLIP) [ECMA430]-- defines a way for exchanged information to simply define the modality, and let interacting agents interpret the exchanged content without requiring a strict structure on the wire.

Experiences implementing NLIP based services and the standard have shown that the approach is viable, has good performance and is secure. For development and debugging, the NLIP protocol allowed for a single interface for trouble-shooting, and our performance evaluation showed that the protocol added little to no overhead, and in many cases out-performed registration functions using a more traditional API.

As the IETF community works towards defining conventions for agent to agent communications, we want to bring this design approach to their attention, since leveraging a protocol like NLIP will provide tremendous flexibility and operational simplicity to software implementing the functions of agents.

6. New Conventions

With the presence of existing standard of NLIP, the task of defining common functions for any agent to agent communications becomes that of defining the semantics of the information that should be transferred, as opposed to defining a rigid structure for the interaction. When agents need to communicate with each other, they need to perform functions such as discover other agents, register themselves with a registrar, or obtain the governing policies for security or data management from other agents. While NLIP provides the basic envelop for transfer of the information, new conventions or standards would need to be developed for each of these functions.

The key difference from traditional approach is that one does not define a rigid structure (e.g. a JSON schema or a XML structure) for interaction but only the semantic content of the information to be exchanged.

We would consider it a task of the IETF to define the exact semantics. As an example for the task of registration, the semantic specification may define that the port number and server host name of the agent should be identified, along with the identity of the owning organization and the provider of security certificate. This information can be expressed in natural language and converted into the local structured representation.

Similarly, for the task of discovery, the agent looking for specific type of remote agent can describe their requirements in natural language, instead of defining a rigid schema for discovery of remote agents. Each agent can convert the requirements to their local structure to decide whether or not they match the requirements. IETF needs to define the semantic content for discovery requests -- e.g. specify that agents must define the type of capabilities they are looking for - such as image analysis, speech to text conversion, security requirements, performance constraints, or billing limits etc. There is no need to define a specific structure for the task.

By following this convention, the standardization process can be made more streamlined and agents implementing the protocol interoperate better with other agents.

7. Security Considerations

This document should not affect the security of the Internet.

8. IANA Considerations

This document includes no request to IANA.

9. Acknowledgement

This template uses extracts from templates written by Pekka Savola, Elwyn Davies and Henrik Levkowetz.

10. References

10.1. Normative References

- [ECMA430] Ecma International, "Natural Language Interaction Protocol", Standard ECMA-430, Edition 1, December 2025, <<https://www.ecma-international.org>>.

[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>>.

10.2. Informative References

Authors' Addresses

Dinesh Verma
IBM
P.O. Box 218, Yorktown Heights
New York, 10598
United States of America
Email: dverma@us.ibm.com

Elisa Bertino
Purdue University
CS Department
West Lafayette , IN 47907
United States of America
Email: bertino@purdue.edu

Abhay Ratnaparkhi
eBay Inc.
7700 West Parmer Lane
Austin, 78729
United States of America
Email: abratnaparkhi@ebay.com

Sean Hughes
Service Now
2225 Lawson Lane
Santa Clara , CA 95054
United States of America
Email: sean.hughes@servicenow.com

Luyi Xing
University of Illinois at Urbana-Champaign
201 N Goodwin Ave
Urbana , IL 61801
United States of America
Email: lxing2@illinois.edu

Zichuan Li
University of Illinois at Urbana-Champaign
201 N Goodwin Ave
Urbana , IL 61801
United States of America
Email: zichuan7@illinois.edu

Xiaoqing Liao
University of Illinois at Urbana-Champaign
201 N Goodwin Ave
Urbana , IL 61801
United States of America
Email: xjliao@illinois.edu

Jian Cui
University of Illinois at Urbana-Champaign
201 N Goodwin Ave
Urbana , IL 61801
United States of America
Email: jiancui3@illinois.edu

Rasit Topaloglu
Marist University
3399 North Road
Poughkeepsie , NY 12601
United States of America
Email: rasit.topaloglu@marist.edu

Mohamed Rahouti
Fordham University
113 W 60th Street
New York , NY 10023
United States of America
Email: mohamed@fordham.edu