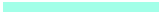


Steven Willmott
CEO, Safe
Intelligence

Do You Speak Agent?

The Convergence of APIs, MCP, A2A, and Agent Languages

Safe Intelligence



Build robust, reliable AI systems.

Deep validation and robustness for AI/ML
Systems (Vision, Tabular, Language).

<http://safeintelligence.ai>

@njyx / <http://steampunkai.com>

Where does this come from...?

“...dedicated to promoting the industry of intelligent agents by openly developing specifications supporting interoperability among agents and agent based applications.”



Circa 2000
<http://www.fipa.org>

Syntax v's Semantics

What is the difference?

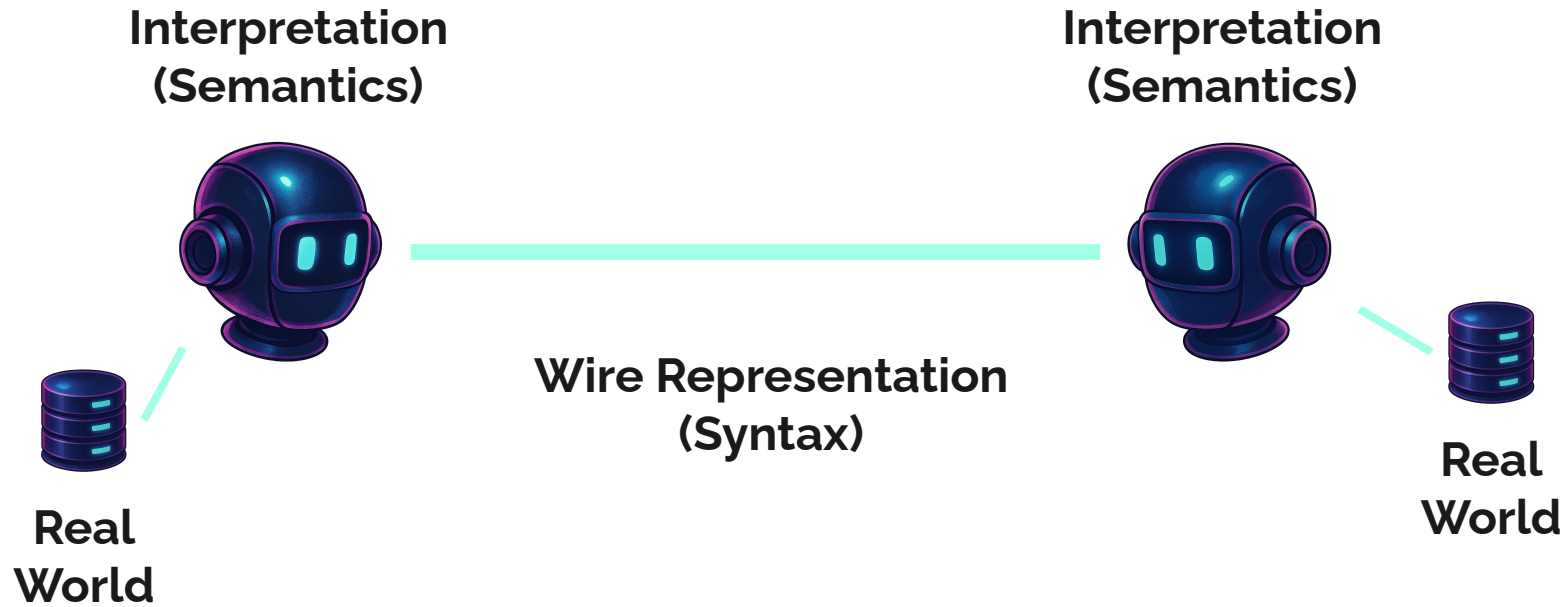


**“The map is
not the
territory”**

Alfred Korzybski
1931

Blackadder goes
Forth, 1989





OpenAPI v's MCP

What is the difference?



3.2.0 is out! <http://openapis.org>

Also see: layered.dev



Example : REST

- GET
`/hotels?city=Mumbai&check_in=2025-06-10&check_out=2025-06-12`
- POST `/reservations` -> `{"hotel_id": "H123", "customer_name": "Alice Example", "check_in": "2025-06-10", "check_out": "2025-06-12"}`



Example MCP

- User: "Find me a cheap hotel in Mumbai from June 10–12."
- Search_hotels tool called with {"location": "Mumbai", "check_in": "2025-06-10", "check_out": "2025-06-12" }
- Return "I found 2 hotels matching your dates..."

What's going on?

- The LLM (or LLMs) are adding semantic interpretation, but the syntax is eerily similar
- We're not tied to REST semantics
- But... MCP isn't necessarily providing for flexible semantics: This comes from LLMs themselves



So...

What about actual semantic flexibility?

A Familiar Problem...

Agent Languages!





Example : FIPA ACL

```
(request
  :sender (agent-identifier :name A)
  :receiver (set (agent-identifier :name B))
  :content
    "((action (agent-identifier :name B)
      (book H123 (c_in 2025-06-10)
        (c_out 2025-06-12))))"
  :protocol fipa-request
  :ontology hotel_booking
  :language fipa-sl
  :reply-with booking567)
```

Speech Act Theory
based semantics

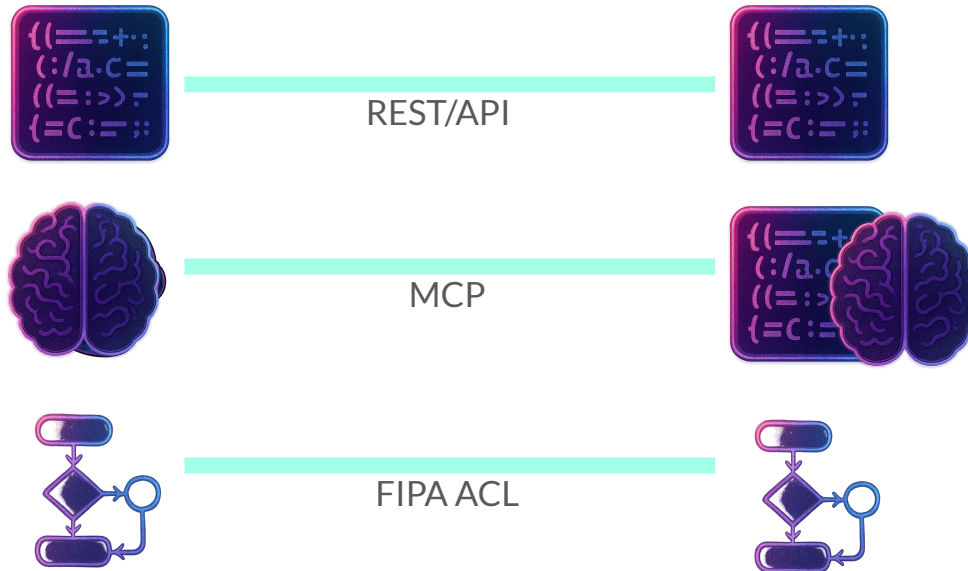
Variety of content
languages

Ontological grounding

Variety of protocols

Variety of encodings
and message transports

So, what's going on?



Semantics captured in different ways

FIPA-ACL, KQML and others built specs for symbolic reasoners

LLMs allow us to “use human language”

But the challenges are actually the same (and remain)

FIPA Performatives (Speech Acts)

- Inform / If / Ref
- Query / If / Ref
- Request
- Request When
- Request Whenever
- Subscribe
- Propose
- Agree
- Refuse
- Cancel
- Confirm
- Disconfirm
- Failure
- Not Understood
- Call for Proposal
- Accept Proposal
- Reject Proposal
- Propagate
- Proxy

Semantics : **Inform**

Sender:

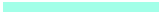
- *holds that some proposition is true,*
- *intends that the receiving agent also comes to believe that the proposition is true, and,*
- *does not already believe that the receiver has any knowledge of the truth of the proposition.*

Receiver:

- *the sender believes the proposition that is the content of the message, and,*
- *the sender wishes the receiver to believe that proposition also.*

Other Things in the FIPA Specs

- Agents & Autonomy
- Structured protocols (requests, CFP, contracting, ...)
- Directory Services, Naming Services, Ontology Services
- Encodings, Message Transport Services (IIOP, HTTP)
- Content languages (SL, Prolog, ...)
- Lots of third party implementations and follow on work (e.g. content languages like CCL).



Yikes! That's a lot!

Agent to Agent (A2A)

Agent Languages (Reborn?)





Example : A2A

```
{  
  "jsonrpc": "2.0",  
  "id": 1,  
  "method": "message/send",  
  "params": { "message": {  
    "role": "user",  
    "parts": [  
      { "kind": "text",  
        "text": "Please book me a hotel in Mumbai between 10  
July and 12 July, 2025"  
      }  
    ],  
    "messageId": "g229e770-767c-417b-a0bo-f0741243c589"  
  },  
  "metadata": {}  
}  
}
```

Agent Cards

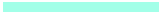
Agents are Opaque

Tasks

Skills

More even footing than
MCP but ... still client
server.

A2A



- Positive to see Agent-to-Agent as the focus
- Many of the require elements re-appear (discovery, agent descriptions, transport neutrality, opacity, ...)
- However:
 - Tending to assume messages are unstructured.. (LLMs inside)
 - When we use more structured content, we're still relying on human developer interpretation for grounding in reality
- More work to do! (but great start!)

Where is “Agent” (& Web) Communication Headed?



Key Takeaway 1

You have to
implement it
somewhere

**Always figure out
where your
semantics are
implemented**

Key Takeaway 2

Autonomy
changes
everything: not
just client-server

Communication between Autonomous Systems is Complex

Key Takeaway 3

The death of APIs
has been greatly
exaggerated...

**The more specific
and narrow the
interactions, the
more efficient the
implementations**

Key Takeaway 4

Stop using a
hammer to crack
a nut!

**Combining
symbolic languages
with today's APIs
and AI might buy us
quite a lot**

Step up ChatGPT 5 (1/3)

Read the following [semantic language specification](#), and respond to the following input:

(query-ref

```
:sender (agent-identifier :name B)
:receiver (set (agent-identifier :name A))
:content
  "((all ?x (q ?x c)))"
:language fipa-sl
:reply-with query2)
```

(inform

```
:sender (agent-identifier :name A)
:receiver (set (agent-identifier :name B))
:in-reply-to query2
:language fipa-sl
:content
  "(= (all ?x (q ?x c)) (set a1 a2 ... an)))"
```

Replace **a1 a2 ... an** with exactly those constants for which **q(x, c)** holds in A's knowledge base. **query-ref** asks for the object denoted by a referential expression, and **all** denotes the set of all objects satisfying the formula.

Step up ChatGPT 5 (2/3)

Respond to the following input:

(request

```
:sender (agent-identifier :name B)
:receiver (set (agent-identifier :name A))
:content
  "((action (agent-identifier :name B) (book
H123 (c_in 2025-06-10) (c_out 2025-06-12))) "
:language fipa-sl
:reply-with query2)
```

(refuse

```
:sender (agent-identifier :name A)
:receiver (set (agent-identifier :name B))
:in-reply-to query2
:language fipa-sl
:content
  "((actor-mismatch
(expected (agent-identifier :name A))
(found (agent-identifier :name B))))")
```

Reason: a **request** asks the receiver to perform the **action**, so the **action** actor must be A, not B.

Step up ChatGPT 5 (3/3)

Respond to the following input:

(request

```
:sender (agent-identifier :name B)
:receiver (set (agent-identifier :name A))
:content
  "((action (agent-identifier :name B) (book
H123 (c_in 2025-06-10) (c_out 2025-06-12))) "
:language fipa-sl
:reply-with query2)
```

(agree

```
:sender (agent-identifier :name A)
:receiver (set (agent-identifier :name B))
:in-reply-to query2
:language fipa-sl
:content
  "((action (agent-identifier :name A)
(book H123 (c_in 2025-06-10) (c_out
2025-06-12))))")
```

This signals that A accepts to attempt the requested action. Execution status (success or failure of the booking) would follow in a later **inform** or **failure** message.

Prediction: AI maintained APIs.

In the long run, high volume channels will be narrow and tightly defined, but maintained by smart systems that intervene occasionally to deal with novel situations and restructure channels.



Thank you!

@njyx

steve@safeintelligence.ai

<http://safeintelligence.ai>

<http://steampunkai.com>

In Memory of:

References



- Fabio Bellifemine, TI Labs, a tireless crusader for agent standards & amazing colleague.
- OpenAPI: <https://www.openapis.org/>
- MCP: <https://modelcontextprotocol.io/>
- A2A: <https://a2a-protocol.org/latest/>
- FIPA: <http://www.fipa.org/>
- KQML:
<https://dl.acm.org/doi/10.1145/191246.191322>
- MCP-Analysis, Kevin Swiber:
<https://www.layered.dev/breaking-down-mcp-an-architectural-analysis>
- FIPA CCL:
<http://www.fipa.org/specs/fipa00009/XC00009B.html>

FIPA: Definition of “Agent”

“A computational process that implements the autonomous, communicating functionality of an application.”

Also see <http://steampunkai.com>