Collaborative Agentic AI Needs Interoperability Across Ecosystems

Rishi Sharma[*]
EPFL
rishi.sharma@epfl.ch

Martijn de Vos
EPFL
martijn.devos@epfl.ch

Pradyumna Chari
Massachusetts Institute of Technology
pchari@mit.edu

Ramesh Raskar
Massachusetts Institute of Technology
raskar@mit.edu

Anne-Marie Kermarrec
EPFL
anne-marie.kermarrec@epfl.ch

Abstract

Collaborative agentic AI is projected to transform entire industries by enabling AI-powered agents to autonomously perceive, plan, and act within digital environments. Yet, current solutions in this field are all built in isolation, and we are rapidly heading toward a landscape of fragmented, incompatible ecosystems. In this position paper, we argue that **interoperability, achieved by the adoption of minimal standards, is essential to ensure open, secure, web-scale, and widely-adopted agentic ecosystems**. To this end, we devise a minimal architectural foundation for collaborative agentic AI, named WEB OF AGENTS, which is composed of four building blocks: agent-to-agent messaging, interaction interoperability, state management, and agent discovery. WEB OF AGENTS adopts existing standards and reuses existing infrastructure where possible. With WEB OF AGENTS, we take a first but critical step toward interoperable agentic systems and offer a pragmatic path forward before ecosystem fragmentation becomes the norm.

1 Introduction

Agentic artificial intelligence (AI) is an emerging approach in which autonomous agents, typically powered by large language models (LLMs), can make decisions, take actions, and learn from the environment in which they are deployed [1]. The capabilities of such agents go beyond generative AI as they can invoke tools, maintain long- and short-term memory, and engage in complex workflows with other agents [39, 48]. Their ability to autonomously plan, communicate, and collaborate unlocks a new class of possibilities ranging from scientific discovery to multi-agent enterprise orchestration [24, 38, 41, 54]. As academia and industry starts to understand these systems better, the impact of agentic AI is poised to be transformative.

While individual agents have shown tremendous potential with the use of tools such as deep research [51] and web search [56], multi-agent AI systems have recently gained traction to enable *collaborative agentic AI* [39, 63]. Recent deployments demonstrate their potential across scientific domains: AI Scientist [41] coordinates specialized agents for end-to-end research automation,

---

[*]Part of the work was carried out while the author was at Massachusetts Institute of Technology.
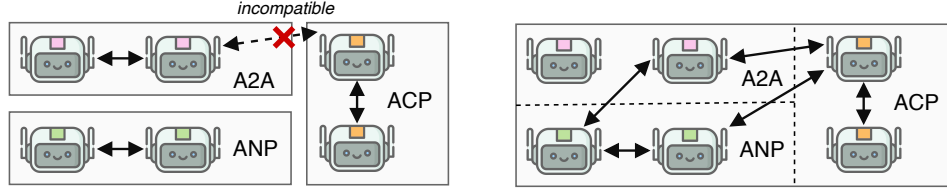
Figure 1: *Left:* Prominent protocols for collaborative agentic AI (A2A [26], ANP [5] and ACP [4]). Agents by default cannot communicate with agents in other ecosystems, resulting in fragmentation (see Section 3). *Right:* our proposed solution, WEB OF AGENTS. We advocate for ecosystem interoperability by encouraging practitioners to adopt existing minimal standards (see Section 5).

ROBIN [24] integrates planning and experimental agents for biomedical discovery, and comparable systems accelerate materials discovery through coordinated simulation and modeling agents [8]. As these agentic capabilities proliferate across domains, the next frontier is a decentralized WEB OF AGENTS where agents can dynamically discover each other's expertise and establish communication to tackle complex, cross-disciplinary challenges that no single agent could address alone [42, 68].

Currently, there is a significant effort to build infrastructure for collaborative agentic AI. Notable examples include the Agent2Agent (A2A) protocol [26], released by Google in April 2025, the Model context protocol (MCP) [45], released by Anthropic in November 2024, and the Agent communication protocol (ACP) [4], released by IBM in April 2025. These solutions introduce necessary primitives such as agent-to-agent communication protocols, agent authentication methods, and mechanisms for expressing the capabilities of agents. However, these siloed efforts develop in isolation with distinct abstractions and data formats, creating a fragmented landscape where agents from one ecosystem cannot interoperate or interact with agents from another. We visualize this challenge in Figure 1 (left). While such diverse and decentralized efforts in implementation accelerates innovation and experimentation [19, 58], our concern is that the resulting ecosystem fragmentation imposes engineering overheads, limits reusability, and threatens long-term scalability and security.

Two obvious approaches could address this fragmentation: *(i)* enforcing a single unified protocol, or *(ii)* creating post-hoc translation layers between each pair of systems. However, history suggests that both approaches are problematic. The former stifles innovation and often fails to gain adoption, as seen with the OSI model's struggle against TCP/IP's pragmatic success [14, 35, 36, 66], while the latter creates undesirable complexity and maintenance burdens [9, 10, 13, 33, 74]. In this position paper, we argue that the field of collaborative agentic AI needs a third approach to address fragmentation. We strongly advocate for a WEB OF AGENTS: an interoperable ecosystem of collaborative agents that are grounded in minimal standards. We visualize this vision in Figure 1 (right). More concretely, **our position is that interoperability, achieved by the adoption of minimal standards, is essential to ensure open, secure, web-scale, and widely-adopted agentic ecosystems**. Rather than constraining development by enforcing a particular protocol or creating brittle translation layers, we advocate for the adoption of lightweight standards that already power web infrastructure. This interoperability enables several key benefits: reduced duplication of effort, scalable ecosystem growth, enhanced security through shared scrutiny, and greater autonomy for users and agent operators as they can freely switch between agents that provide similar services.

To realize this vision, we identify four minimal, yet sufficient building blocks that ecosystems should adopt for enabling interoperability: *(i)* agent-to-agent messaging, enabling asynchronous and structured communication via existing web protocols, *(ii)* interaction interoperability, establishing shared interaction specifications through interaction documents, *(iii)* state management, supporting both short-term sessions and long-term state, and *(iv)* discovery, allowing agents to find one another. Our approach requires minimal new infrastructure, just careful extensions of the web as we know it. Together, these building blocks form the foundation of the WEB OF AGENTS.

Contributions. To defend our position, we first analyze several prominent solutions in Section 3, identifying common trends across current frameworks for collaborative agentic AI. This analysis motivates our arguments in Section 4 that interoperability is not merely desirable, but essential for realizing the full potential of collaborative agentic AI. We then propose in Section 5 a minimal blueprint, WEB OF AGENTS, that demonstrates how this interoperability can be achieved through
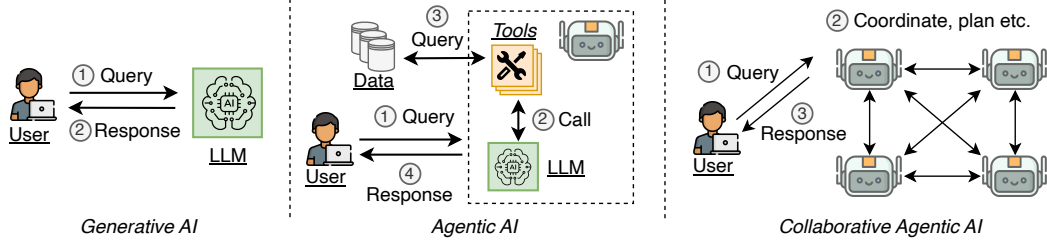
Figure 2: Generative AI (left), agentic AI (middle), and collaborative agentic AI (right). This work provides a blueprint for interoperable collaborative agentic AI that leverages existing web protocols.

careful reuse and extension of existing web technologies. We discuss some views that oppose our position in Section 6 before concluding with final remarks in Section 7.

2 From Generative AI to Collaborative Agentic AI

Generative AI. Generative AI encompasses computational models capable of producing novel content across multiple modalities, including text, images, and audio [22]. Among the most prominent implementations is ChatGPT, which provides a conversational interface to large language models (LLMs) through natural language processing [2]. LLMs demonstrate proficiency in diverse language generation tasks, including document summarization, content synthesis, and creative text generation. The standard generative AI workflow, illustrated in Figure 2 (left), follows a request-response pattern where users submit queries to an LLM system (Step 1) and receive generated outputs (Step 2). This architecture supports multi-turn conversations, enabling contextual dialogue through sequential query-response exchanges within a persistent session.

However, this conventional user-to-LLM paradigm exhibits notable architectural limitations [64]. A fundamental constraint is the models' inability to perform autonomous environmental observation, external system interaction, or action execution without explicit human orchestration. Such capabilities would encompass database querying, web navigation [16], or programmatic transaction execution. While retrieval-augmented generation (RAG) frameworks [37] partially address these limitations by augmenting model inputs with relevant external information, these approaches typically require separate orchestration mechanisms to manage the retrieval process.

Agentic AI. To address these architectural constraints, the field has witnessed a paradigm shift toward agentic artificial intelligence systems [30, 48]. Agentic AI architectures are fundamentally structured around autonomous *agents*—computational entities capable of independent operation and dynamic interaction with external *tools*, as illustrated in Figure 2 (middle). These agents are typically specialized and optimized for specific task domains, such as computer vision processing or web navigation protocols. Tools constitute external service interfaces that agents can programmatically invoke to execute functions beyond the inherent capabilities of the underlying LLMs. Tool selection and orchestration represent critical components of the response generation pipeline, requiring systematic evaluation of available resources for optimal task completion [61]. Agents can autonomously invoke multiple tools in parallel (Step 2 in Figure 2) when required, utilizing tool responses to inform subsequent planning and action selection. These tools maintain direct interfaces to various data repositories and external systems (Step 3). Agentic AI frameworks demonstrate significant potential for transformative applications across domains including healthcare and software engineering [48].

However, contemporary agentic architectures predominantly operate through single-agent paradigms, constraining system scalability, modularity, and fault tolerance. This architectural limitation necessitates that individual agents manage comprehensive task execution, logical reasoning, and tool orchestration, potentially exceeding agent capabilities when confronting complex scenarios requiring specialized domain expertise or access to heterogeneous tool ecosystems.

Collaborative Agentic AI. Collaborative agentic AI represents the next step and involves AI agents that can communicate and collaborate across tasks, domains, and organizational boundaries. We visualize this frontier in Figure 2 (right). Compared to agentic AI with a single agent, agents can form collectives and work together (Step 2). For example, a travel agent can coordinate with a

Solution	Agent Identity	Inter-Agent Comm.	Agent Discovery	State Management	Interoperability w/ other solutions
A2A [26]	✓AgentCard	✓Async.	❖Static	✗	✗
MCP [45]	❖ Contextual	✗	✗	✗	✗
ACP [4]	✓	✓REST API	✗	✗	✗
agents.json [71]	❖ Capabilities	✗	✗	✗	✗
Agora [43]	❖Protocol docs	✓	✗	✗	✗
ANP [5]	✓Identity Layer	✓	✓	✗	✗
AITP [50]	✓	✓Chat Threads	✗	✓	✗
LMOS [17]	✓	✓	✓	✓	✗

Table 1: Prominent solutions for (collaborative) agentic AI and their innovation areas.

calendar agent to check availability, a payment agent to process transactions, and a policy agent to enforce organizational constraints. These interactions can all occur without human intervention. Such networks of LLM-based agents have already been used to simulate human and economic behavior [38, 54], and to understand dynamics in societies [27]. The recent boom in AI agents follows substantial work on multi-agent systems (MAS), a field with decades of research on autonomous agents that, similar to collaborative agentic AI, coordinate, negotiate, and act in digital environments [21]. Classical MAS research focuses on pre-programmed agents with static rules, whereas LLM-powered agents can operate in more unstructured environments and reason using natural language.

3   The Alarming Trend Toward Incompatible Agentic Ecosystems

The potential of (collaborative) agentic AI has resulted in significant research and engineering efforts from both industry and academia to devise new infrastructure, protocols, and frameworks [29, 69]. Notable examples include Google's A2A protocol [26], Anthropic's MCP [45], and IBM's ACP [4], leading to the "protocol wars" phase of (collaborative) agentic AI. Based on existing work [18] and our own research, we identify prominent solutions for (collaborative) agentic AI and look at their innovation areas, see Table 1. Even though most solutions focus on inter-agent communication, we include MCP and agents.json in our analysis because of their increased adoption in agentic workflows [18]. The goal of our analysis is not to point out particular shortcomings of individual solutions but rather to take a broader perspective on the development of agentic AI solutions.

Our main observation is that **agentic AI solutions are developed in isolation and are incompatible with each other.** As a result, we are heading toward a collection of ecosystems in which agents built using one framework are unlikely to interoperate with those from another [60]. Such fragmentation introduces inefficiencies, for example, companies wishing to deploy agents that interact across systems must either commit to a single protocol or reimplement agents for each distinct ecosystem. As new solutions continue to emerge, these interoperability challenges are poised to grow.

One might address the interoperability challenges among diverse agentic AI systems with *translation layers* or *mediators* that convert messages and actions between different protocols [52]. This approach, for example, has been adopted to connect fragmented web standards in the early days of the web and to enable value exchange between different blockchain platforms. While this brings interoperability, the viability of translational layers has been questioned [9, 33]. Firstly, translation layers can become brittle or outdated as protocols evolve. Secondly, such layers can expand the system's attack surface and expose vulnerabilities that malicious actors could exploit, as has been observed with cross-blockchain bridges [74]. Thirdly, as the number of distinct protocols increases, maintaining translation mappings becomes increasingly difficult [13]. These challenges raise serious concerns about whether translation layers can serve as a long-term solution for agentic AI interoperability.

Beyond interoperability challenges, this ecosystem-level isolation has also led to uneven progress across the agentic AI stack. A holistic solution for agentic AI requires, among others, primitives such as agent identity, protocols for inter-agent communication, agent discovery mechanisms, and mechanisms for memory management. We observe in Table 1 discrepancies in the focus of innovation areas across solutions. On the one hand, all identified solutions emphasize some notion of agent identity, such as proposing mechanisms for agents to describe their capabilities or roles. On the other hand, only a minority of solutions implement mechanisms for the discovery of other agents in

the network. Similarly, the capability of an agent to manage short- or long-term memory receives inconsistent attention across solutions. This uneven focus results in fragmented tooling that hinders composability across systems. This pattern is not unique to agentic AI: similar fragmentation has occurred in domains such as internet-of-things [6], blockchain [73], and early web services [53], where numerous tools emerged to solve narrow problems in isolation.

Some analyzed solutions introduce their own implementations of core primitives such as authentication, capability exposure, agent discovery, and message exchange formats. For example, while some leverage existing authentication schemes such as the OAuth protocol [28] or decentralized identifiers (DIDs) [59], others either rely on custom mechanisms or leave this aspect unspecified. Similarly, agent discovery is often realized through custom listings with their own formats. Generally, the result is a patchwork of ecosystems that are incompatible by design. Moreover, some protocols introduce new primitives without clearly defined use cases, optimizing for hypothetical scenarios rather than concrete needs. LMOS, for example, follows a maximalist design by building a complete OS-like environment for LLMs. Messaging applications are a pertinent analogy to this observation: major services like iMessage, WhatsApp, Signal, or Telegram all implement their own protocol stack, message format, and identity systems, and are largely incompatible with each other [7].

The field of agentic AI stands at a critical point. In the absence of coordination, it risks splintering into competing silos, each solving similar problems in slightly different and incompatible ways. For instance, consider a travel-booking agent developed by a major airline that supports rich capabilities such as real-time seat selection using a particular agent protocol. Even if this agent is exceptionally capable, users and other agents relying on a different agentic ecosystem would be unable to invoke it or compose it into larger workflows. In this way, incompatibility directly burdens engineers and diminishes the quality of service for end users.

> **Key Takeaway**
>
> Because each solution is being developed in isolation, agentic AI is evolving into a landscape of incompatible ecosystems.

4 Interoperability as Necessary Foundation for Agentic AI

If fragmentation is the current trajectory, interoperability is the antidote. To achieve this, we advocate for the adoption of **minimal standards** that enable agents to interact across different ecosystems. Minimal standardization has historically enabled breakthrough innovations across domains: protocols like HTTP and HTML bootstrapped the web [9], SMTP and IMAP enabled global email interoperability [15, 32], and TCP/IP continues to serve as the backbone of the Internet [57]. With minimal, interoperable foundations, we strongly believe we can bootstrap a successful agentic ecosystem. Next, we further motivate interoperability and elaborate on how it strengthens agentic AI in terms of security, open participation, and scalability.

Security. Interoperability through minimal standardization contributes to the security of the overall ecosystem [20]. This standardization creates opportunities for systematic auditing, formal verification, and community-wide threat modeling. In fact, security is a critical concern in the emerging landscape of agentic AI and is a necessity to protect against adversarial agents or users [67]. This is because, unlike traditional systems that follow predefined logic, AI agents operate in dynamic environments, interpret ambiguous instructions, and make autonomous decisions, often without human supervision. This autonomy introduces novel attack surfaces, *e.g.*, prompt injection and jailbreak attacks [40, 70]. As these agents become embedded in critical workflows, security becomes paramount.

Open Participation. Interoperability empowers agent builders, *e.g.*, developers, organizations, or infrastructure providers, to exercise meaningful choice in the agents they interact with [31]. In a collaborative agentic AI system, agents often depend on other agents. With shared protocols and minimal standards in place, builders can freely choose among competing agents that offer similar functionality but differ in trust models, performance, cost, or affiliation. Without such freedom of choice, switching costs rise dramatically as each agent may require custom APIs, data formats, and authentication schemes. This not only leads to vendor lock-in, but also stifles innovation by discouraging new entrants from building compatible alternatives [72]. This also holds for end users,
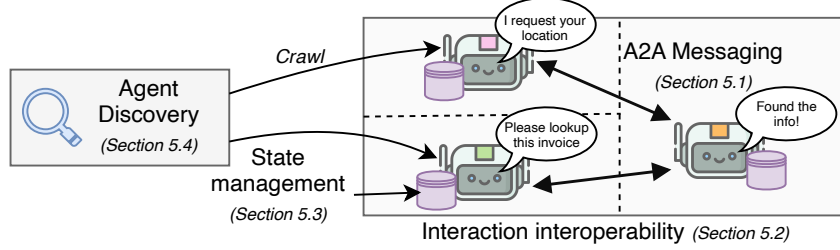
Figure 3: The blueprint of WEB OF AGENTS and its four building blocks.

who may want to directly use the services of agents across ecosystems or combine agents from different providers into a single workflow. An interoperable agentic AI ecosystem fosters open participation from both engineers and end-users, enabling vibrant competition, lowering barriers to entry, and promoting long-term ecosystem health [6].

Closely related, interoperability is also a prerequisite for a decentralized agentic AI ecosystem. Without a shared set of standards, coordination inevitably gravitates toward a small number of centralized platforms that dictate how agents are built and deployed. In contrast, minimal interoperability standards allow diverse actors to independently develop and operate agents, while still participating in the overall ecosystem [31]. With interoperability, anyone can publish and run agents that are discoverable and usable by others without needing permission from any authority.

Scalability.   Interoperability is also a key enabler of scalability in terms of ecosystem expansion. As the agentic AI landscape grows, no single provider will be able to build and maintain every agent, tool, or integration needed to support diverse user needs across domains. A scalable ecosystem depends on the ability of independent developers to contribute agents, services, and infrastructure components that can plug into a larger system without needing to coordinate closely with every other participant. Shared protocols reduce the integration overhead, allowing developers to focus on functionality rather than glue code. This mirrors the success of the web, where interoperable standards enabled a vast ecosystem of independent websites, browsers, and services to emerge without centralized planning [9]. Furthermore, it allows the protocols to grow with the necessary improvements as the field evolves and particular use-cases become more prominent.

Key Takeaway

Interoperability is not just a nice-to-have feature, it is foundational. It enables secure, composable, scalable, and decentralized agentic ecosystems across domains.

---

5   Architectural Foundations for an Interoperable Web of Agents

We now identify a minimal set of building blocks and devise a blueprint for an interoperable agentic AI ecosystem named WEB OF AGENTS. The WEB OF AGENTS architecture is visualized in Figure 3, and the building blocks are summarized in Table 2. This blueprint adheres to the long-standing system design principle: *Keep It Simple, Stupid* [34]. This principle guides our strategy for enforcing *minimal standardization*, and *adopting existing standards* where possible. We strongly believe that the greatest interoperability can be achieved by staying as close as possible to existing web technologies.

5.1   Agent-to-Agent Messaging

Standardized message exchange protocols are essential for interoperability in an agentic AI ecosystem, as they establish the communication channels for coordination, negotiation, and collaboration. Without common messaging standards, each pair of interacting agents would require custom integration, creating an scaling problem that would severely restrict ecosystem growth [53].

Leveraging Hypertext Transfer Protocol (HTTP) for agent-to-agent communication. HTTP, the protocol that underpins the modern web, is well-suited for agent-to-agent communication through its

Table 2: WEB OF AGENTS building blocks.

Building blocks	Functional needs	Web technologies
Agent-to-agent messaging	HTTP-based messaging	HTTP requests
Interaction interoperability	Interaction documentation	API documentation
State management	Short-term memory Long-term memory	Sessions DB integration
Agent discovery	Unique endpoints Capability advertisement	URLs, DNS Well-known paths

request-response pattern and well-defined semantics [23]. By adopting HTTP as the foundational transport protocol, agents can exchange information using established methods: GET requests to retrieve information from other agents (similar to browsers accessing webpages), and POST requests for more complex interactions with messages and other context data in the payload. This HTTP-based approach leverages decades of rethinking, improvements, and optimizations without duplication of efforts [46]. While some emerging protocols like ACP and Agora directly leverage HTTP requests for agent-to-agent communication, other solutions like A2A have opted for more specialized protocols like JSON-RPC [26, 43]. We strongly believe that agent-to-agent communication does not require specialized protocols like JSON-RPC or WebSockets. While these protocols offer some compelling features such as asynchronous communication, enforcing these as a standard would be an overkill, hinder progress, and limit interoperability [47].

Another advantage of utilizing HTTP requests for agentic communication is that agents can coexist with webpages, making it trivial to integrate agentic AI systems with existing software. Furthermore, specialized patterns like asynchronous requests can be easily implemented on the client-side using HTTP requests and have long existed on the web [44]. Finally, by building upon HTTP's extension mechanisms, this foundation allows for future enhancements without breaking backward compatibility, creating a stable yet evolvable communication layer for the WEB OF AGENTS.

5.2 Interaction Interoperability

Communication between autonomous agents presents unique challenges beyond basic message transportation. Agents are currently being developed using diverse frameworks, architectural approaches, underlying technologies, and with varying capabilities (see Section 3). In fact, different agents might need completely different inputs, possibly in different formats, to carry out their tasks. For example, a weather forecasting agent may expect location coordinates and a timestamp, while a financial analysis agent requires historical market data and risk parameters. Existing solutions typically do not sufficiently consider differences in agent design and capabilities. Protocols typically assume a universal interaction pattern without accounting for variability in inputs, expected conditions, or optimal message compositions that can be efficiently interpreted by the underlying LLM [4, 26]. Therefore, collaborative agentic AI requires a way to enable interoperability in agent interactions.

Interaction document. To enable interaction interoperability, we propose an *interaction document* exchange between agents. This way, each agent can explicitly document its interface requirements, capabilities, expected inputs, outputs, and conditions necessary for effective communication. Crucially, this documentation need not follow a schema or predefined structure. Since modern AI agents are powered by LLMs, they can interpret and adapt to various documentation formats, from structured JSON to natural language descriptions. This documentation acts as a standardized handshake, accessible via common endpoints using HTTP requests, and enables agents to understand and adapt to diverse interaction patterns without prior agreements or custom engineering efforts by the agent operator. This approach is minimalistic in the sense that we neither restrict how agents interact with each other nor impose rigid formatting requirements on their capability descriptions.

This minimalistic approach to interaction interoperability offers three advantages. First, it enables agents to evolve independently, *i.e.*, the internal implementations can change completely as long as the documentation interface remains stable. Second, it fosters innovation by avoiding premature and overly restrictive standards, providing agent operators flexibility in their implementations. Ultimately, this straightforward yet effective solution mitigates the friction associated with agent integration.

5.3 State Management

Agent interactions are inherently stateful, distinguishing them from traditional stateless web requests [25]. Agents engage in conversations where each interaction changes the internal state of the agent or the environment. This requires both short-term retention of state (*e.g.*, for each pairwise agent interaction) and long-term state persistence. However, most agentic AI frameworks adopt stateless designs or delegate state management entirely to the developers [4, 26, 43]. Without standardized approaches to state management, agents would remain confined to isolated and context-free exchanges, severely limiting their interoperability and potential to collaborate on complex tasks [62].

Short-term memory through sessions. Agents require mechanisms to maintain immediate context across multiple message exchanges in a pairwise conversation. This requirement can be satisfied through existing web session management patterns, which are well-established. More specifically, short-term state in web interactions is managed through cookies, which is a standardized approach as part of the HTTP protocol, where a unique session identifier is assigned during the initial interaction and subsequently included in all requests within the same conversation [55].

Long-term memory through databases. Beyond immediate conversational context, agents often require long-term state storage to store historical interactions and accumulated knowledge across sessions [63]. This long-term memory requirement can be satisfied through established patterns for database integration in web services. The agents can write to and read from the database as part of their workflow. Similar patterns underpin virtually all data-persistent web applications, from social media platforms to collaborative document editing systems.

Thus, we do not need sophisticated or new solutions to state management in agentic AI. Reusing existing approaches for state management offers significant advantages over more prescriptive standards. It allows agent developers to be sure that all agents will have state management, but leaves them the flexibility of implementing it according to their specific requirements.

5.4 Discovery

Discovery is a foundational requirement for any interoperable agent ecosystem. Without standardized discovery mechanisms, agent ecosystems would fragment into isolated islands of functionality, severely limiting their collective capabilities. Current methods either involve developing isolated networks of agents like Naptha AI [49] or curating a centralized registry of existing AI agents like Agency Marketplace [3]. While the former requires all agents to be on the same closed network, the latter requires manual registry self-curation. Both these approaches are not scalable, and hence, unsuitable for WEB OF AGENTS. To enable a truly collaborative WEB OF AGENTS, we need search engines for agents that regularly crawl and index agents on the web. This mirrors how web search engines transformed the early internet from a collection of disconnected pages into a navigable and discoverable information space [11, 65].

Unique endpoints. Agents require globally unique identifiers that can be reliably resolved to network endpoints for other agents or search engines to discover and interact with them. Uniform resource locators (URLs) provide an ideal solution as they are already designed to uniquely identify resources on the web. This approach benefits from the established domain name system (DNS) infrastructure for name resolution without introducing any new identification scheme. This approach provides several advantages: *(i)* URLs are human-readable, *(ii)* they are hierarchical, allowing organizational namespacing of agents, *(iii)* the entire URL resolution infrastructure, including the DNS, has been optimized and maintained to handle the scale of the web, and *(iv)* this design allows agents and webpages to coexist over the web, enabling seamless interoperability.

Capability advertisement. Agents must expose metadata about their capabilities and available tools. Most agentic protocols, including A2A, ACP, and agents.json, describe their formal standards for the description of agents and their capabilities. However, the number of fields in such a formal description can range from basic capabilities to guardrails, and is only expected to grow as we build new agentic AI systems [12]. Therefore, keeping it simple, each agent should expose a short document that describes its capabilities and tools. This document can potentially be the same as the interaction documentation described in Section 5.2. Similar to A2A, we propose hosting the capability description at a well-known path (similar to *robots.txt* and *sitemap.xml*). This standardized location ensures that both search engines and other agents can reliably locate capability information.

By adopting these design principles for agent discovery, we establish a minimalistic standardization framework that directly leverages and builds on top of the web's highly optimized and scalable infrastructure. Each component can keep evolving to make WEB OF AGENTS better while maintaining interoperability across agentic AI ecosystems.

5.5 Putting it Together

The four building blocks of WEB OF AGENTS form a sufficient foundation for collaborative agentic AI. To illustrate this, consider the AI Scientist MAS [41], where specialized AI agents collaborate for scientific discovery. In WEB OF AGENTS, these agents would operate without prior knowledge of other agents and their capabilities. The discovery mechanism regularly indexes all agents. When the hypotheses-generation agent (H) needs computational support, it can query the discovery to locate a suitable code-generation agent (C) for collaboration. Agent-to-agent messaging enables H to send requests to C, while interaction interoperability allows H to understand C's interface requirements and compose properly formatted requests. Finally, state management allows H and C to maintain contextual continuity across multi-turn dialogs without retransmitting complete conversations. Critically, both H and C operate independently and may reside in different agentic ecosystems (*e.g.*, A2A and ACP), yet these four building blocks are minimal and sufficient to enable seamless collaboration. The current nascent state of agentic AI frameworks is a perfect moment for adopting these minimal standards to avoid the *protocol wars* and lay the foundations of an interoperable WEB OF AGENTS.

6 Alternative Views

Alternative View 1: *Web technologies were designed for content delivery and should be reinvented for agentic AI.*

We reject the assumption that web technologies are fundamentally incompatible with agentic AI. In fact, web technologies have evolved from serving static websites to dynamic content. At their core, agents are like dynamic websites: they expose some capabilities, respond to structured requests, maintain internal state, and can be discovered and invoked using standard web protocols. Some agentic AI systems are already extending mature web technologies, although in isolation. Thus, agentic AI requires alignment with web technologies, which we have shown is feasible in Section 5.

Alternative View 2: *It is too soon to worry about interoperability in Agentic AI: natural selection will eventually determine the best practices and standards.*

While it is true that the agentic AI space is still evolving and diverse designs as well as use cases are expected, deferring interoperability until a mature protocol emerges is shortsighted. History shows that dominant solutions often arise without prioritizing openness (*e.g.*, early cloud APIs or messaging platforms). Google's A2A may become influential, but it is not inherently neutral nor guaranteed to support open collaboration across providers. Messaging apps serve as a cautionary tale: while dominant platforms emerged, the lack of interoperability led to closed ecosystems, lock-in, and duplicated engineering effort [72]. We thus ask ourselves the question: do we want to monopolize agentic AI by adopting one ecosystem, or allow a decentralized existence of diverse but interoperable ecosystems? Our position takes the decentralized stance: early alignment around minimal standards ensures flexibility and guards against future silos, even if popular solutions emerge later.

Alternative View 3: *Standardizing early slows down progress.*

We agree that standardization can slow innovation, especially in a fast-moving field like agentic AI. We acknowledge this and hence, do not advocate for extensive protocols, but for minimal and extensible foundations, just enough to enable interoperability without stifling experimentation. WEB OF AGENTS is the first step towards this. The web itself evolved this way: early standards like HTTP and HTML were minimalistic and were later extended with the uses that were common on the web.

7 Final Remarks

In this position paper, we argue that collaborative agentic AI demands interoperability as a foundational requirement. Rather than waiting for dominant solutions to emerge, we advocate for the adoption of minimal standards that enable agents to discover, communicate, and collaborate across

organizational and technical boundaries. Our proposed WEB OF AGENTS blueprint shows that such interoperability can be achieved by reusing existing web technologies.

While our work focuses on interoperability as a core enabler for collaborative agentic AI, we acknowledge that this is only one piece of a much broader puzzle. An agentic ecosystem must also address critical challenges around *trust*, *security*, *privacy*, and *safety*. These challenges are more complex in agentic AI systems compared to traditional systems due to their non-deterministic behavior. Therefore, we call on the agentic AI community to focus on these challenges while adopting interoperability standards to accelerate the development of collaborative agentic AI instead of attempting to build the best agent-to-agent framework.

References

[1] Deepak Bhaskar Acharya, Karthigeyan Kuppan, and B Divya. Agentic ai: Autonomous intelligence for complex goals–a comprehensive survey. *IEEE Access*, 2025.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Agency. Agency marketplace. `https://marketplace.agen.cy/agents/`, 2025. Accessed: 2025-05-21.

[4] Agent Communication Protocol Project. Agent communication protocol specification. `https://agentcommunicationprotocol.dev/introduction/welcome`, 2025. Accessed: 2025-05-20.

[5] Agent Network Protocol Project. Agent network protocol (anp) specification. `https://agent-network-protocol.com/`, 2025. Accessed: 2025-05-20.

[6] Mohab Aly, Foutse Khomh, Yann-Gaël Guéhéneuc, Hironori Washizaki, and Soumaya Yacout. Is fragmentation a threat to the success of the internet of things? *IEEE Internet of Things Journal*, 6(1):472–487, 2018.

[7] René Arnold and Anna Schneider. An app for every step: A psychological perspective on interoperability of mobile messenger apps. 2017.

[8] Adib Bazgir, Rama chandra Praneeth Madugula, and Yuwen Zhang. Multicrossmodal automated agent for integrating diverse materials science data, 2025. URL `https://arxiv.org/abs/2505.15132`.

[9] Tim Berners-Lee. *Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor*. Harper San Francisco, 1999.

[10] Terry Bollinger. A guide to understanding emerging interoperability technologies. *MITRE.(Acedido em 23 de Junho de 2004) http://www. mitre. org/work/tech_papers/tech_papers_00/bollinger_interoperability/bolling er_interop. pdf*, 2000.

[11] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

[12] Stephen Casper, Luke Bailey, Rosco Hunter, Carson Ezell, Emma Cabalé, Michael Gerovitch, Stewart Slocum, Kevin Wei, Nikola Jurkovic, Ariba Khan, et al. The ai agent index. *arXiv preprint arXiv:2502.01635*, 2025.

[13] Yuhan Chen, David Annebicque, Alexandre Philippot, Véronique Carré-Ménétrier, and Thierry Daneau. Evaluation methodology of interoperability for the industrial domain: Standardization vs. mediation. *Processes*, 11(4):1274, 2023.

[14] David Clark. The design philosophy of the darpa internet protocols. In *Symposium proceedings on Communications architectures and protocols*, pages 106–114, 1988.

[15] M. Crispin. Internet message access protocol - version 4rev1. RFC 3501, March 2003. `https://www.rfc-editor.org/rfc/rfc3501`.

[16] Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H Laradji, Manuel Del Verme, Tom Marty, Léo Boisvert, Megh Thakkar, Quentin Cappart, David Vazquez, et al. Workarena: How capable are web agents at solving common knowledge work tasks? *arXiv preprint arXiv:2403.07718*, 2024.

[17] Eclipse Foundation. Eclipse lmos: Language model operating system. `https://eclipse.dev/lmos/`, 2025. Accessed: 2025-05-20.

[18] Abul Ehtesham, Aditi Singh, Gaurav Kumar Gupta, and Saket Kumar. A survey of agent interoperability protocols: Model context protocol (mcp), agent communication protocol (acp), agent-to-agent protocol (a2a), and agent network protocol (anp). *arXiv preprint arXiv:2505.02279*, 2025.

[19] Francisco Eiras, Aleksandar Petrov, Bertie Vidgen, Christian Schroeder De Witt, Fabio Pizzati, Katherine Elkins, Supratik Mukhopadhyay, Adel Bibi, Botos Csaba, Fabro Steibel, et al. Near to mid-term risks and opportunities of open-source generative ai. *arXiv preprint arXiv:2404.17047*, 2024.

[20] Mahmoud Elkhodr, Seyed Shahrestani, and Hon Cheung. The internet of things: New interoperability, management and security challenges. *arXiv preprint arXiv:1604.04824*, 2016.

[21] Jacques Ferber and Gerhard Weiss. *Multi-agent systems: an introduction to distributed artificial intelligence*, volume 1. Addison-wesley Reading, 1999.

[22] Stefan Feuerriegel, Jochen Hartmann, Christian Janiesch, and Patrick Zschech. Generative ai. *Business & Information Systems Engineering*, 66(1):111–126, 2024.

[23] Roy T. Fielding and Julian Reschke. Hypertext transfer protocol (http/1.1): Message syntax and routing. `https://www.rfc-editor.org/rfc/rfc7230`, 2014. RFC 7230.

[24] Ali Essam Ghareeb, Benjamin Chang, Ludovico Mitchener, Angela Yiu, Caralyn J. Szostkiewicz, Jon M. Laurent, Muhammed T. Razzak, Andrew D. White, Michaela M. Hinks, and Samuel G. Rodriques. Robin: A multi-agent system for automating scientific discovery, 2025. URL `https://arxiv.org/abs/2505.13400`.

[25] Debi Prasad Ghosh. Agentic ecosystemin engineering design: A framework for interoperable legacy tools and emergent collaboration via mcp/a2a protocols. 2025.

[26] Google DeepMind. Announcing the agent2agent protocol (a2a). `https://developers.googleblog.com/en/a2a-a-new-era-of-agent-interoperability/`, April 2025. Accessed: 2025-05-20.

[27] Önder Gürcan. Llm-augmented agent-based modelling for social simulations: Challenges and opportunities. *HHAI 2024: Hybrid Human AI Systems for the Social Good*, pages 134–144, 2024.

[28] Dick Hardt. The oauth 2.0 authorization framework. Technical report, 2012.

[29] Soodeh Hosseini and Hossein Seilani. The role of agentic ai in shaping a smart future: A systematic review. *Array*, page 100399, 2025.

[30] Laurie Hughes, Yogesh K Dwivedi, Tegwen Malik, Mazen Shawosh, Mousa Ahmed Albashrawi, Il Jeon, Vincent Dutot, Mandanna Appanderanda, Tom Crick, Rahul De', et al. Ai agents and agentic systems: A multi-expert analysis. *Journal of Computer Information Systems*, pages 1–29, 2025.

[31] Sayash Kapoor, Noam Kolt, and Seth Lazar. Resist platform-controlled ai agents and champion user-centric agent advocates. *arXiv preprint arXiv:2505.04345*, 2025.

[32] J. Klensin. Simple mail transfer protocol. RFC 5321, October 2008. `https://www.rfc-editor.org/rfc/rfc5321`.

[33] Serguei Kolos. Corba: a practical introduction. 2000.

[34] Butler W Lampson. Hints for computer system design. In *Proceedings of the ninth ACM symposium on Operating systems principles*, pages 33–48, 1983.

[35] Barry M Leiner, Vinton G Cerf, David D Clark, Robert E Kahn, Leonard Kleinrock, Daniel C Lynch, Jon Postel, Lawrence G Roberts, and Stephen S Wolff. The past and future history of the internet. *Communications of the ACM*, 40(2):102–108, 1997.

[36] Barry M Leiner, Vinton G Cerf, David D Clark, Robert E Kahn, Leonard Kleinrock, Daniel C Lynch, Jon Postel, Larry G Roberts, and Stephen Wolff. A brief history of the internet. *ACM SIGCOMM computer communication review*, 39(5):22–31, 2009.

[37] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

[38] Nian Li, Chen Gao, Mingyu Li, Yong Li, and Qingmin Liao. Econagent: large language model-empowered agents for simulating macroeconomic activities. *arXiv preprint arXiv:2310.10436*, 2023.

[39] Bang Liu, Xinfeng Li, Jiayi Zhang, Jinlin Wang, Tanjin He, Sirui Hong, Hongzhang Liu, Shaokun Zhang, Kaitao Song, Kunlun Zhu, et al. Advances and challenges in foundation agents: From brain-inspired intelligence to evolutionary, collaborative, and safe systems. *arXiv preprint arXiv:2504.01990*, 2025.

[40] Yupei Liu, Yuqi Jia, Runpeng Geng, Jinyuan Jia, and Neil Zhenqiang Gong. Formalizing and benchmarking prompt injection attacks and defenses. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 1831–1847, 2024.

[41] Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

[42] May Malka, Erez Farhan, Hai Morgenstern, and Nir Shlezinger. Decentralized low-latency collaborative inference via ensembles on the edge. *IEEE Transactions on Wireless Communications*, 2024.

[43] Samuele Marro, Emanuele La Malfa, Jesse Wright, Guohao Li, Nigel Shadbolt, Michael Wooldridge, and Philip Torr. A scalable communication protocol for networks of large language models. *arXiv preprint arXiv:2410.11905*, 2024.

[44] Microsoft. Asynchronous request-reply pattern, 2025. URL `https://learn.microsoft.com/en-us/azure/architecture/patterns/async-request-reply`. Accessed: 2025-05-21.

[45] Model Context Protocol Project. Model context protocol specification. `https://modelcontextprotocol.io/specification/2025-03-26`, 2025. Accessed: 2025-05-20.

[46] Jeffery C Mogul. Clarifying the fundamentals of http. In *Proceedings of the 11th international conference on World Wide Web*, pages 25–36, 2002.

[47] Paul Murley, Zane Ma, Joshua Mason, Michael Bailey, and Amin Kharraz. Websocket adoption and the landscape of the real-time web. In *Proceedings of the Web Conference 2021*, pages 1192–1203, 2021.

[48] San Murugesan. The rise of agentic ai: Implications, concerns, and the path forward. *IEEE Intelligent Systems*, 40(2):8–14, 2025.

[49] Naptha AI. Naptha ai. `https://naptha.ai/`, 2025. Accessed: 2025-05-22.

[50] NEAR AI. Agent interaction and transaction protocol (aitp) specification. `https://aitp.dev/`, 2025. Accessed: 2025-05-20.

[51] OpenAI. Deep research system card. `https://cdn.openai.com/deep-research-system-card.pdf`, 2025.

[52] Cristina Paniagua. Service interface translation. an interoperability approach. *Applied Sciences*, 11(24):11643, 2021.

[53] Mike P Papazoglou and Willem-Jan Van Den Heuvel. Service oriented architectures: approaches, technologies and research issues. *The VLDB journal*, 16:389–415, 2007.

[54] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701320. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

[55] Weihong Peng and Jennifer Cisna. Http cookies–a promising technology. *Online Information Review*, 24(2):150–153, 2000.

[56] Perplexity. Perplexity.ai. https://perplexity.ai/, 2025. Accessed: 2025-05-22.

[57] J. Postel. Transmission control protocol. RFC 793, September 1981. https://www.rfc-editor.org/rfc/rfc793.

[58] Eric Raymond. The cathedral and the bazaar. *Knowledge, Technology & Policy*, 12(3):23–49, 1999.

[59] Drummond Reed, Manu Sporny, Dave Longley, Christopher Allen, Ryan Grant, Markus Sabadello, and Jonathan Holt. Decentralized identifiers (dids) v1. 0. *Draft Community Group Report*, 2020.

[60] Arion Research. The danger of fragmentation in ai agent ecosystems, 2024. URL https://www.arionresearch.com/blog/x8nwy1u8hufvp84sczxuskg49qspop. Accessed: 2025-05-21.

[61] Jingqing Ruan, Yihong Chen, Bin Zhang, Zhiwei Xu, Tianpeng Bao, Hangyu Mao, Ziyue Li, Xingyu Zeng, Rui Zhao, et al. Tptu: Task planning and tool usage of large language model-based ai agents. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023.

[62] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. pearson, 2016.

[63] Ranjan Sapkota, Konstantinos I Roumeliotis, and Manoj Karkee. Ai agents vs. agentic ai: A conceptual taxonomy, applications and challenge. *arXiv preprint arXiv:2505.10468*, 2025.

[64] Johannes Schneider. Generative to agentic ai: Survey, conceptualization, and challenges. *arXiv preprint arXiv:2504.18875*, 2025.

[65] Candy Schwartz. Web search engines. *Journal of the American Society for Information Science*, 49(11):973–982, 1998.

[66] Leah R Shafer. Open standards and the digital age: History, ideology, and networks. *Afterimage*, 42(3):39, 2014.

[67] Yonadav Shavit, Sandhini Agarwal, Miles Brundage, Steven Adler, Cullen O'Keefe, Rosie Campbell, Teddy Lee, Pamela Mishkin, Tyna Eloundou, Alan Hickey, et al. Practices for governing agentic ai systems. *Research Paper, OpenAI*, 2023.

[68] Nir Shlezinger, Erez Farhan, Hai Morgenstern, and Yonina C Eldar. Collaborative inference via ensembles on the edge. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8478–8482. IEEE, 2021.

[69] Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D Nguyen. Multi-agent collaboration mechanisms: A survey of llms. *arXiv preprint arXiv:2501.06322*, 2025.

[70] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36:80079–80110, 2023.

[71] Wildcard AI. agents.json specification. `https://docs.wild-card.ai/agentsjson/introduction`, 2025. Accessed: 2025-05-20.

[72] Xingyu Yan and Yang Feng. Mandating interoperability of interpersonal communications services in an ecosystems competition context. *Computer Law & Security Review*, 50:105850, 2023.

[73] Alexei Zamyatin, Mustafa Al-Bassam, Dionysis Zindros, Eleftherios Kokoris-Kogias, Pedro Moreno-Sanchez, Aggelos Kiayias, and William J Knottenbelt. Sok: Communication across distributed ledgers. In *Financial Cryptography and Data Security: 25th International Conference, FC 2021, Virtual Event, March 1–5, 2021, Revised Selected Papers, Part II 25*, pages 3–36. Springer, 2021.

[74] Qianrui Zhao, Yinan Wang, Bo Yang, Ke Shang, Maozeng Sun, Haijun Wang, Zijiang Yang, and Haojie Xin. A comprehensive overview of security vulnerability penetration methods in blockchain cross-chain bridges. *Authorea Preprints*, 2023.