From Protocols to Trust: Enabling Interoperability and Discovery in the Agentic Web

Sree Bhargavi Balija
UC San Diego
sbalija@ucsd.edu

Pradyumna Chari
Massachusetts Institute of Technology
pchari@mit.edu

Abhishek Singh
Project NANDA, MIT Media Lab

Ayush Chopra
Project NANDA, MIT Media Lab

Ramesh Raskar
Massachusetts Institute of Technology
raskar@mit.edu

Sriram Krishnan
Istari Digital

Raghu Bala
Synergetics

Abstract

As the proliferation of AI agents becomes inevitable across consumer and enterprise ecosystems, the need for interoperability, decentralization, and trust has emerged as a critical requirement. This paper presents a comparative analysis of existing agent communication protocols such as MCP (5; 23; 21), A2A (3), ACP (4), and recent initiatives from Cisco's Agency (ACP, AGP) (16), and argues for a decentralized, registry-synced architecture that ensures openness, security, and scalability. We build upon contributions from Cisco's Agency initiative (16), the Nanda research collective at MIT, and emerging trust frameworks from Acorn Labs, Mayfield Ventures, and Vigil (18). Drawing parallels to the evolution of the internet from dial-up to broadband, we introduce a two-layered registry architecture, a lightweight resolution layer and a semantic agent card layer unified by a robust trust layer (20). This architecture supports verifiable credentials, behavioral evaluation, and adaptive routing, and balances short-term interoperability needs with the long-term imperative of building a discoverable, monetizable, and economically aligned Internet of Agents. We further highlight the limitations of existing protocols for agent-to-agent collaboration (1; 9; 8) and propose design principles for establishing a global quilt of decentralized registries that can support mission-critical enterprise use cases while remaining open to innovation from consumer and Web3-driven ecosystems.

1 Introduction

The AI ecosystem is undergoing a rapid transformation, with autonomous agents becoming the fundamental units of intelligence across consumer applications and enterprise-grade workflows. Similar to how containerization catalyzed the Kubernetes revolution in cloud-native computing, agent orchestration is emerging as the next layer of abstraction—enabling scalable, intelligent, and dynamic systems.

Despite this growing enthusiasm, the current agent landscape remains fragmented. Protocols, toolchains, registries, and incentive structures are often siloed, hindering agent-to-agent collaboration and monetizable cooperation. Without a shared infrastructure for identity, discovery, trust, and reputation, the ecosystem risks echoing the pitfalls of early closed and incompatible AI silos.

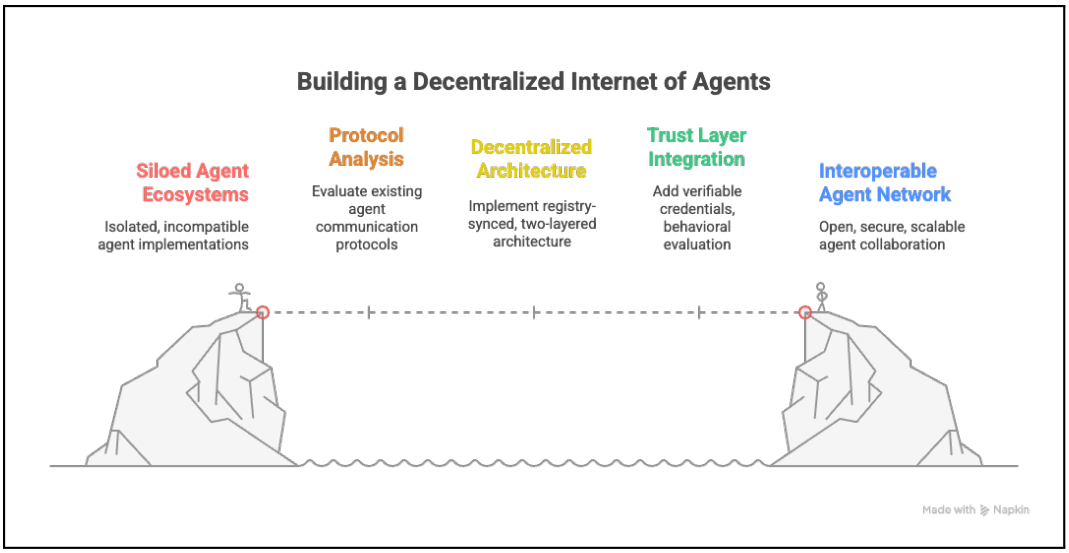


Figure 1: Building decentralized internet of agents

Recent initiatives such as the Model Context Protocol (MCP) (5; 23; 18), Agent-to-Agent Protocol (A2A) (3), and Agent Connect Protocol (ACP) (4) have introduced promising abstractions for communication. However, these protocols primarily target execution orchestration and fail to adequately address the deeper infrastructural needs of agent discovery, semantic identity, and dynamic trust management at scale.

This paper argues that enabling seamless collaboration between billions of autonomous agents requires foundational primitives that go beyond basic messaging. In particular, we advocate for a robust trust layer that quantifies, validates, and maintains agent reputation and behavioral integrity.

To address this, we build upon the contributions of the Nanda research collective at MIT, in collaboration with Cisco, Flower, Dell, HCL, TCS, and other academic partners. We propose a two-layered registry architecture tailored for the Agentic Web:

- **Layer 1:** A lightweight, fast-resolving registry that maps agent names or decentralized identifiers (DIDs) to metadata URLs, supporting rapid resolution and lookup.
- **Layer 2:** A semantic agent card (or "agent fact") layer that extends A2A's foundational ideas to include verifiable credentials, composability profiles, adaptive routing metadata, and service history.

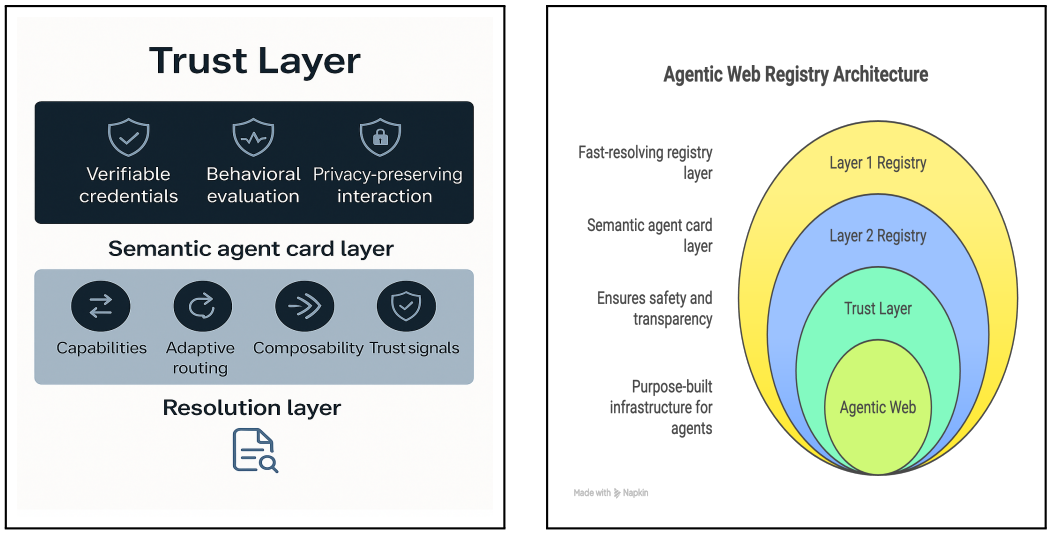
These two registry layers are unified via a trust layer that employs distributed and federated trust authorities, behavioral evaluation engines, and credential-based attestations. This design ensures that agents operate within safe, transparent, and verifiable boundaries.

Inspired by the shift from dial-up to broadband, we argue that the Agentic Web necessitates a purpose-built infrastructure where semantic discoverability, programmable incentives, and behavioral safety become first-class primitives. Furthermore, perspectives from Mayfield Ventures, Acorn Labs, and Vigil (16; 20) emphasize the need for trustworthy, auditable, and test-driven agent deployments—especially in mission-critical domains such as healthcare, finance, and defense.

As Vin Sharma of Vigil observes, we must move from demo-ready chatbots to production-grade autonomous systems that degrade gracefully under adversarial stress, comply with strict policies, and pass rigorous behavioral audits.

Through this paper, we aim to:

1. Provide a unified perspective on registry and trust-layer requirements for agentic systems across consumer and enterprise use cases.
2. Evaluate the "upgrade vs. switch" paradigms in adapting today's web infrastructure for decentralized agent interactions.



(a) Trust layer-enabled agent stack (b) Registry-synced agent discovery

Figure 2: Architectural illustrations of agent stack and discovery layer

3. Propose design principles and open questions for building a resilient, incentivized, and composable Internet of Agents.

Ultimately, we envision a future where agents are not only interoperable but also economically and behaviorally aligned—operating across a globally distributed mesh of registries, protocols, and verifiable trust layers.

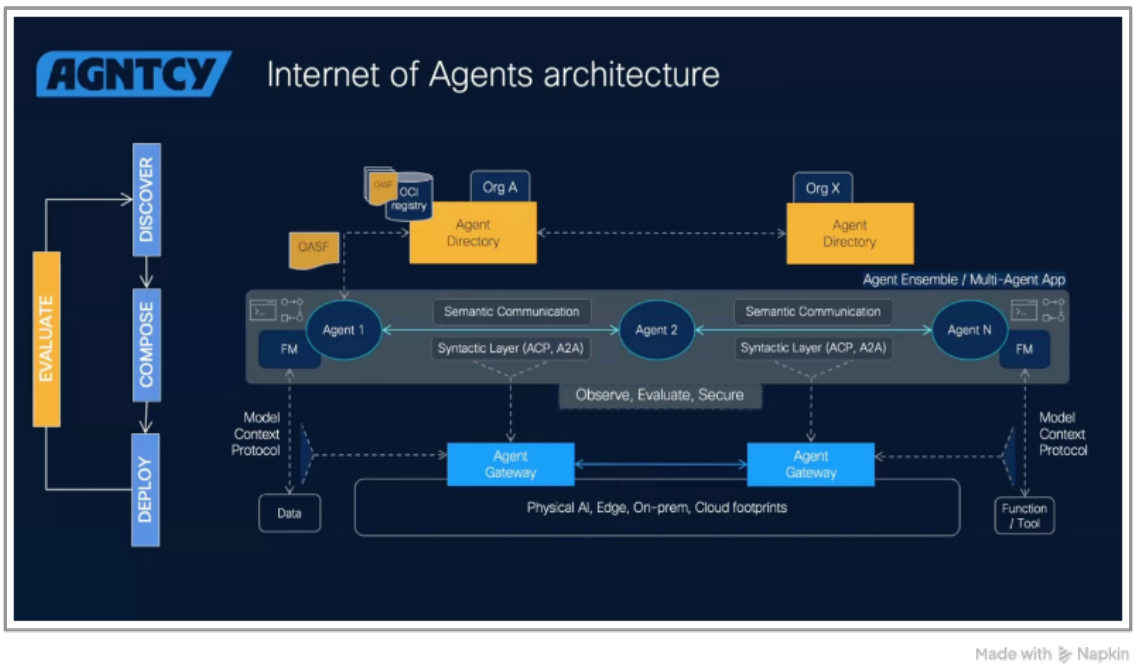


Figure 3: Internet of Agents: A Blueprint for Agent Collaboration and Secure Communication

2 Related Work and Protocol Landscape

The agent interoperability landscape has seen a flurry of protocol development and conceptual frameworks:

MCP (Model-Tool Communication Protocol): Provides structured agent-to-tool interactions but is limited in peer-to-peer scenarios. MCP was rapidly adopted but requires manual tool lookup, making it suitable for agent-to-tool contexts rather than dynamic agent-to-agent collaboration (1; 15; 16; 20).

A2A (Agent-to-Agent Protocol): Pioneered self-declared agent cards and human-readable descriptions, offering a first step toward agent discoverability and composability (3). However, A2A lacks the dynamic behavioral metadata needed for large-scale multi-agent orchestration (4).

ACP (Agent Connect Protocol) and **AGP (Agent Gateway Protocol)** from Cisco's Agency: Extend the ecosystem to support dynamic agent-to-agent communication, group interactions, and cross-vendor composability (2; 13). These protocols explicitly target the formation of decentralized agent ecosystems, emphasizing that agent-to-agent communication requires more than repurposed agent-to-tool abstractions.

Agent Directories and OASF (Open Agentic Schema Framework): The Agency's initiative to create a decentralized, OCI-based registry layer, where directories can interconnect under policy-driven trust and discovery mechanisms (5). These initiatives parallel efforts by the Nanda registry architecture, which proposes a global quilt of hybrid registries that span enterprises and decentralized communities (17; 27).

Decentralized Identity (DIDs) and Web3 paradigms: Offer robust primitives for agent identity and trust, aligning with the need to support both enterprise-grade and consumer-facing agents in a unified ecosystem (21; 28).

Across these efforts, a clear consensus emerges: agent discovery, trust, and routing must be designed from the ground up for a decentralized, collaborative, and economically-aligned future. Our architecture synthesizes these lessons, integrating a trust layer as a first-class citizen to support verifiable credentials, privacy-preserving discovery, and secure multi-agent orchestration.

Table 1: Comparison of Protocols in Agent Interoperability Landscape

Protocol Name	Origin	Limitations	Use Cases
MCP	Model-Tool Communication Protocol	Lacks dynamic agent-to-agent orchestration; manual tool lookup; not suited for scalable peer-to-peer agent collaboration	Tool orchestration in data pipelines; agent execution within ML workflows
A2A	Agent-to-Agent Protocol	Lacks dynamic behavioral metadata; not robust for large-scale multi-agent scenarios	Initial prototypes for agent discoverability; small-scale interoperability
ACP	Cisco's Agency	Execution-level focus; may require further extension for semantic and trust-layer integration	Mission-critical workflows; cross-enterprise orchestration
AGP	Cisco's Agency	Early-stage protocol; may require integration with dynamic discovery and adaptive trust layers	Secure enterprise-grade agent group communication
OASF	Cisco's Agency	Focuses on metadata and schema; requires complementary trust layer for holistic safety guarantees	Extending agent descriptions for security audits and trust alignment

2.1 Mathematical Models and Formal Methods for Trust, Incentivization, and Interoperability

2.1.1 Secure Communication & Data Privacy

For homomorphic encryption or zero-knowledge proofs, formalize privacy guarantees:

$$\text{Privacy Loss } (\epsilon, \delta) = \Pr[\mathcal{M}(D) \in S] \leq e^{\epsilon} \Pr[\mathcal{M}(D') \in S] + \delta$$

Where

- $\mathcal{M}$: *Agent-internal* computation mechanism.
- D, D': Neighboring datasets.
- (ϵ, δ): Differential privacy budget.

2.1.2 Dynamic Registry Resolution Time

For **Layer 1 registry resolution**, model it as:

$$\text{Resolution Time} = O(\log N)$$

where N is the number of registered agents, assuming **balanced tree-like or trie-based data structures**.

For **federated registry sync** (CRDT or gossip-based protocols), approximate convergence time as:

$$T_{\text{convergence}} = O(\log N)$$

where N is the total number of registry nodes.

2.1.3 Trust Score as a Weighted Graph Centrality

Model trust relationships as a **weighted graph** $G = (V, E)$ where:

- V: Agents.
- E: Trust edges with weights w_{ij}.

Trust score for agent i:

$$\text{TrustScore}_i = \frac{1}{d_i} \sum_{j \in N(i)} w_{ij}$$

- $N(i)$: Neighbors of i.
- d_i: Degree of agent i.

This captures **reputation propagation** and can be extended using *EigenTrust* or *PageRank* variants:

$$\mathbf{T} = \alpha \mathbf{W} \mathbf{T} + (1 - \alpha)\mathbf{e}$$

where:

- $\mathbf{W}$: Row-stochastic matrix of edge weights.
- $\mathbf{e}$: Base trust vector.
- α: Dampening factor.

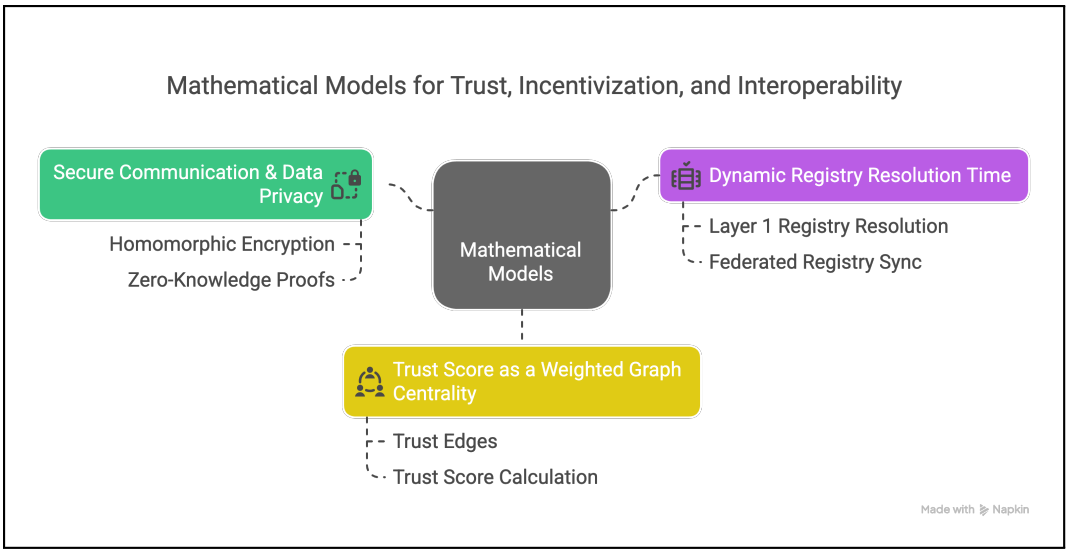


Figure 4: Mathematical models

3 Incentivization: The Role of Microtransactions

As Coinbase's X42 demonstrates, decentralization must include economic incentives. Agents need native compensation methods for services like compute, storage, and intelligence. X42 enables real-time micropayments via HTTP headers, supporting:

- Agent-to-agent task delegation
- Pay-per-use access to memory or APIs
- Real-time inference compensation

Traditional payment models (e.g., credit cards) introduce latency, static identity assumptions, and high fees—making them impractical for agents.

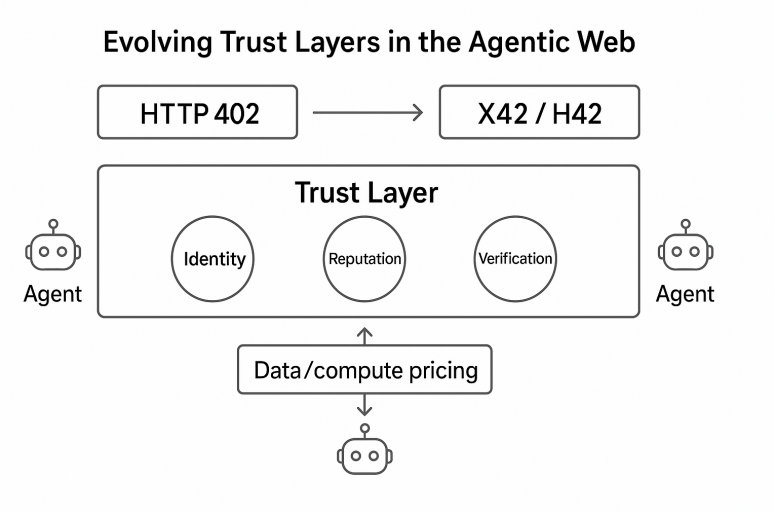


Figure 5: Architecture overview of trust layer-enabled agent stack

4 Evolving Trust Layers: From HTTP 402 to X42/H42 and Beyond

The Agentic Web represents a paradigm shift where autonomous agents seamlessly interact and transact across distributed ecosystems. This evolution is underscored by the journey from HTTP 402, a once-forgotten status code reserved for payments, to the modern micropayment protocols X42 and H42 that enable real-time, decentralized agent-to-agent (A2A) transactions. These payment rails act as the foundation for a trust layer, a dynamic construct integrating identity, reputation, and verification modules to ensure reliable, autonomous cooperation.

As agents independently access data, compute, and intelligence resources, the trust layer expands beyond traditional monetary exchanges to include data/ compute pricing and the emerging notion of knowledge commoditization. In this agentic economy, trust is no longer a passive backdrop but an active participant, enabling agents to self-organize, monetize interactions, and drive discovery without compromising privacy or security.

The conceptual diagram captures this progression, illustrating how the reimagining of payments and identity layers forms the robust fabric of the Agentic Web, paving the way for packet-switched intelligence and the commoditization of micro-wisdom. booktabs array float caption

Table 2: Trust Evaluation Dimensions for Agent Systems

Dimension	Technical Approach	Example Tools	Benefits
Behavioral Predictability	Anomaly detection on logs or event data	One-Class SVMs, Markov Chains	Ensures expected and safe agent behavior
Policy Compliance	Policy-as-code with live runtime enforcement	eBPF, OPA, Rego	Enforces regulatory or system-level constraints dynamically
Provenance & Attestation	Use of verifiable credentials and signatures	W3C VCs, DIDs, Signed attestations	Builds trust in agent identity and data lineage
Secure Execution	Code sandboxing in secure runtimes	WASM (Wasmer, Wasmtime)	Isolates agents and mitigates code-level attacks
Resilience & Containment	Out-of-distribution and adversarial testing	Robustness test frameworks	Measures agent robustness under novel or adversarial input

5 The Need for Decentralization

A mesh topology of registries supports discovery without compromising ownership or control.

- Agent identity and discovery must be decentralized using DIDs.
- Registry federation (as implemented by Cisco's agent directory) enables organizations to host and sync their own agent catalogs.
- Centralized directories risk reintroducing the very gatekeeping that the internet dismantled. To build a vibrant, inclusive agent ecosystem:

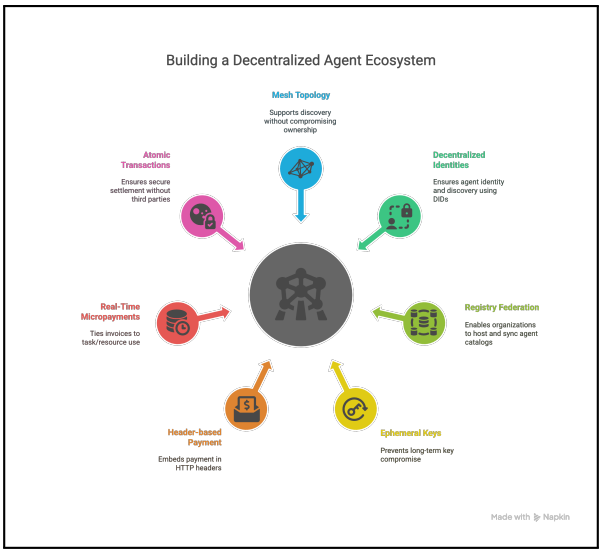


Figure 6: Building a Decentralized Agent Ecosystem

Table 3: X42/H42 Micropayment Features

Feature	Technical Implementation	Benefits for Agents
Ephemeral Keys	Short-lived keys for payment/auth	Prevents long-term key compromise
Header-based Payment Protocol	Payment embedded in HTTP headers (e.g., X-Payment, H42-Payment)	API-native payment without endpoint change
Real-Time Micropayments	Invoices tied to task/resource use	Fine-grained compute/data monetization
Integration with Agent Economics	Monetization of agent services	Enables autonomous agent income
Atomic Transactions	Inline crypto verification	Secure settlement, no third-party needed

6 The Agency Framework

Cisco's *Agency* initiative introduces a robust and modular framework for multi-agent collaboration, with **trust** as a foundational pillar rather than an afterthought. Recognizing that agent systems must be not only interoperable but also verifiable and secure, Agency embeds trust mechanisms across every layer of the agent lifecycle—from discovery to deployment and evaluation.

The framework addresses trust across four key components:

Agent Directory (Trust through Identity and Discovery)

Cisco's directory layer extends OCI (Open Container Initiative) formats to support distributed registries of agent metadata. This registry is:

- Federated and policy-controlled, allowing organizations to define access and trust policies.
- Designed to participate in a global Internet of Agents.
- Capable of syncing with other directories under customizable trust models (e.g., open, shared, private).

- Powered by decentralized identity (DIDs), ensuring agents are verifiably unique and traceable.

OASF (Open Agentic Schema Framework)

Built as an extension of OCSF (Open Cybersecurity Schema Framework), OASF provides a semantically rich and extensible schema to describe agent capabilities, safety constraints, and verification hooks. It is designed to:

- Allow external validators to audit agent behavior.
- Capture verifiable credentials, provenance, and reputation signals.
- Support policy-aligned deployment contracts, enabling granular control over what an agent can and cannot do.

Agent Gateway (Trust through Controlled Interaction)

Beyond identity, how agents interact must also be governed. The Agent Gateway:

- Supports secure group-based messaging.
- Enables both point-to-point and many-to-many communication models.
- Implements built-in access control.

This is essential for establishing zero-trust communication in collaborative and competitive environments.

6.1 IO Mapper (Semantic Trust and Compatibility)

Interoperability often fails at the semantic level. Cisco's IO Mapper is an intelligent layer that aligns input-output expectations across heterogeneous agents. Powered by LLMs, it:

- Provides semantic mediation between agents with differing ontologies.
- Enables trust through compatibility, ensuring agents correctly interpret requests and responses without misalignment or ambiguity.

In essence, Cisco's Agency Framework operationalizes trust by combining:

- Verifiable identity
- Semantic validation
- Behavioral containment
- Secure communication channels

This positions it as a leading initiative for building production-grade, mission-aligned agent ecosystems—particularly in enterprise and edge environments where trust is non-negotiable.

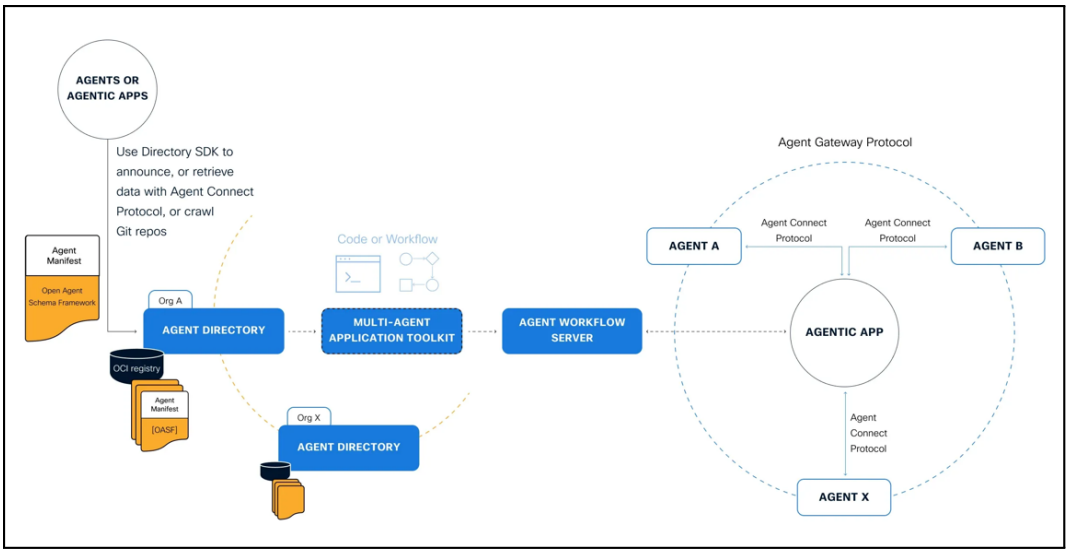


Figure 7: Cisco Agency Framework: Operationalizing Agent Trust

Table 4: Recommended Practices for the Internet of Agents

Recommendation	Technical Rationale	Expected Impact
Embrace decentralized identity	DIDs and VCs for agent sovereignty	Eliminates centralized identity dependence
Enforce behavioral validation	Runtime anomaly detection and policies	Improves safety in dynamic conditions
Adopt X42/H42 micropayments	Ephemeral payment models	Enables scalable agent incentive models
Develop secure containerization	WASM, eBPF, TEEs	Prevents misbehavior and data leakage
Align on open schemas	OASF/OCSF interoperability standards	Future-proof, cross-vendor compatibility
Leverage test-driven evaluation	Adversarial testing + verification pipelines	Trust in safety and performance

7 Architectural Blueprint

We propose a five-layer architecture:

- **Discovery** – Decentralized registries, DID-based identity
- **Composition** – A2A, ACP protocols with LangChain and CrewAI integration
- **Deployment** – Framework-agnostic deployment model across cloud, edge, and local nodes
- **Evaluation** – Standardized observability and benchmarking, extended from OpenTelemetry
- **Incentivization** – X42/H42 powered microtransactions and programmable payments

8 Interoperability Meets Economic Coordination

Beyond communication, multi-agent systems require shared incentives. Platforms like BitGPT and Coinbase demonstrate that local coordination and resource sharing (e.g., memory, APIs, compute) can be unlocked through granular, anonymous payments. Instead of revealing identity, agents can collaborate via conditional incentives.

This is key to enabling:

- Autonomous swarm coordination
- Pay-per-inference contracts
- Tokenized intelligence and marketplace integration

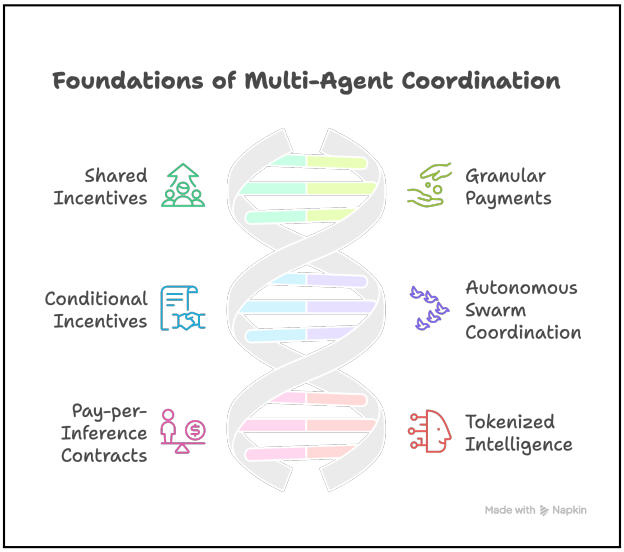


Figure 8: Agent economy

9 Towards Trustworthy Agent Infrastructure

Echoing insights from Mayfield Ventures, Acorn Labs, and Vigil, **trust** in agents is the paramount requirement for production-grade adoption. Just as Kubernetes brought operational order to cloud-native computing, agent-native systems require a dedicated trust framework.

Trust must be:

- **Quantified** — through test-driven development and composable evaluations
- **Context-aware** — validated under adversarial, noisy, and out-of-distribution conditions
- **Containable** — ensuring agents operate only within their policy-scoped bounds
- **Transparent** — with attestable provenance, behavioral predictability, and regulatory compliance

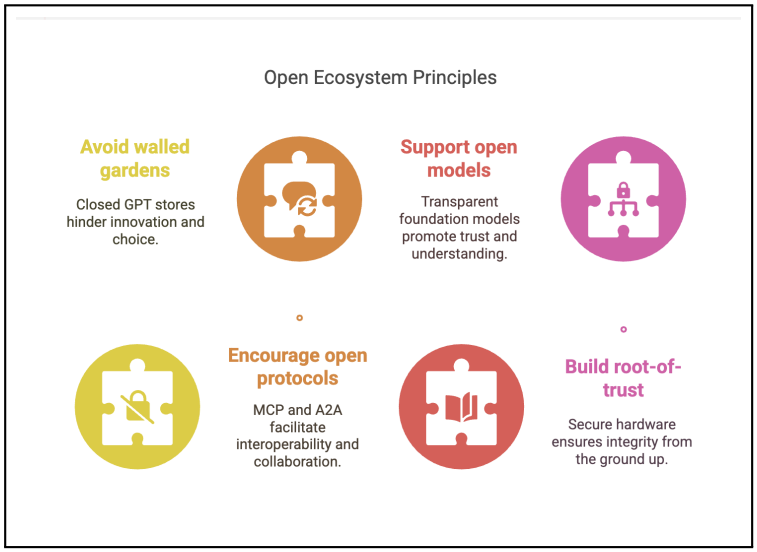
Frameworks like *Vigil* provide tooling to evaluate agents across safety, reliability, security, and compliance. These capabilities are essential in mission-critical domains such as banking, healthcare, and legal operations—where agent misbehavior is unacceptable.

10 Lessons from Kubernetes and Open Ecosystems

History demonstrates that open standards (e.g., HTTP, HTTPS, containers, Kubernetes) accelerate innovation. Agentic infrastructure needs the same approach:

- Avoid walled gardens (e.g., closed GPT stores)
- Encourage open protocols (MCP, A2A)
- Support transparent, open-weight foundation models
- Build root-of-trust from secure hardware to orchestration layers

Enterprise readiness demands this openness not only in discovery and access but also in evaluating agents' fitness for mission-critical roles.



11 Recommendations for the Ecosystem

- Embrace decentralized identity and registry-based agent discovery
- Enforce behavioral validation before mission-critical deployments
- Adopt pay-per-capability microtransaction infrastructure via X42/H42
- Develop secure containers and enforce mandatory access policies
- Align on open schema frameworks like OASF for cross-agent operability
- Leverage test-driven evaluation methodologies for agent trust scores

12 AutoPatch+: Trust Layer for LLM Reliability

Figure 9: AutoPatch+ Trust Layer: Elevating LLM Reliability

AutoPatch+ is a cutting-edge hallucination verification framework for large language models (LLMs), combining Retrieval-Augmented Generation (RAG) with Neural Additive Model (NAM)-based trust layers.

It detects, verifies, and rewrites incorrect outputs—boosting model reliability and user trust. AutoPatch+ represents a paradigm shift in ensuring factual, transparent, and accountable AI-generated answers.

Table 5: Projects and Trust Layer Concepts

Project Name	Description	Concept in Trust Layer
Synergetics	AgentConnect protocol with decentralized registry for A2A and dApp interoperability	DID-based identity and agent verification framework
AxonVertex	Policy framework for NANDA registries, incl. confidential computing	Policy-based trust with anonymization
AutoPatch+	Detects and fixes hallucinations in LLMs via RAG + NAM	Direct trust layer enforcing generation accuracy
Universitas AI	Socrates agent for research; validates limitations, uses citations	Hallucination control via evidence-based answers
Acoer	Health agent leveraging TEEs and federated learning for overdose prevention	Privacy-preserving trust via cryptographically verifiable insights
BitGPT	Connects generative agents with verifiable outputs	Trust layer for AI output verification

13 Conclusion

This paper champions the emergence of a decentralized, economically aligned, and open agent ecosystem, one that echoes the collaborative ethos of the early internet. We propose a new foundation for the Internet of Agents, grounded in five essential agentic primitives: *discovery*, *composition*, *payment*, *trust validation*, and *semantic coordination*. These primitives go beyond enabling autonomous

function—they empower agents to transact, collaborate, and reason in dynamic, adversarial, and trust-sparse environments. Initiatives such as Cisco's Agency, MIT's NANDA Registry, Coinbase's X42, Vigil's trust infrastructure, and AutoPatch+ for hallucination verification exemplify early momentum in this space. However, communication protocols alone are insufficient. The next generation of infrastructure must introduce a registry synced, verifiable, and economically incentivized trust layer, agnostic to agent origin, framework, or modality, that enables scalable, secure, and composable agent interactions.

At the heart of this vision lies the principle of **agent trust**, which cannot be solely derived from self-declared capabilities. Agents must be validated through behavioral audits, adaptive routing, and cryptographically verifiable interactions. A robust trust layer should support verifiable credentials, privacy preserving exchanges, and dynamic trust evaluation mechanisms that are resilient to adversarial behavior. Just as Kubernetes defined secure orchestration for the cloud era, the agent native stack must enforce containment, transparency, and policy compliance for autonomous entities, particularly in sensitive domains like finance, healthcare, and law. Furthermore, economic interoperability is critical to agent scalability. Microtransaction layers such as X42 and decentralized identity systems like DIDs unlock permissionless value exchange among agents, enabling models of peer inference, delegated computation, and swarm coordination. We call upon academia, open source communities, startups, and enterprises to embrace federated discovery, open schemas, and trust first architectures. The breakthroughs that shaped the internet such as HTTP, containers, and Kubernetes succeeded not only through engineering but through inclusive and interoperable design cultures. The Internet of Agents demands the same.

In that same spirit, we believe a collaborative, trustworthy, and production-grade Internet of Agents is not just possible, it is imminent. Let this paper be both an architectural blueprint and a community invitation to shape that future.

Acknowledgments

We extend our sincere gratitude to the Cisco Agency team, Coinbase X42 engineers, BitGPT researchers, Mayfield Fund, Acorn Labs, Vigil AI, and the Nanda consortium for their invaluable insights on building interoperable and incentivized agent ecosystems. Their pioneering work in agent-to-agent micropayments, decentralized infrastructure, and trust layer mechanisms has significantly shaped our understanding of the emerging Agentic Web.

References

[1] 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.

[2] Khanh-Tung Tran et al. *Multi-agent collaboration mechanisms: A survey of LLMs*. arXiv preprint arXiv:2501.06322, 2025.

[3] Idan Habler et al. *Building a secure agentic AI application leveraging A2A protocol*. arXiv preprint arXiv:2504.16902, 2025.

[4] Jun Liu et al. *ACPs: Agent collaboration protocols for the Internet of Agents*. arXiv preprint arXiv:2505.13523, 2025.

[5] Xinyi Hou et al. *Model Context Protocol (MCP): Landscape, security threats, and future research directions*. arXiv preprint arXiv:2503.12345, 2025.

[6] Sanjay Aiyagari. *Natural Language Interaction Protocol (NLIP): Redefining secure communication between natural language AI models*. SPIE Proceedings: Disruptive Technologies in Information Sciences IX, 2025.

[7] Reid G. Smith. *The contract net protocol: High-level communication and control in a distributed problem solver*. IEEE Trans. on Computers, C-29(12):1104–1113, 1980.

[8] Stefan Poslad. *Specifying protocols for multi-agent system interaction*. ACM Trans. on Autonomous and Adaptive Systems, 2(4):1–25, 2007.

[9] Tim Finin, Don McKay, and James Weber. *KQML as an agent communication language*. Proc. of the Third Int'l Conf. on Information and Knowledge Management, pp. 456–463, 1994.

[10] S. Poslad, P. Buckle, and R. Hadingham. *The FIPA-OS agent platform: Open source for open standards*. In Practical Applications of Intelligent Agents and Multi-Agent Technology, pp. 355–368, 2000.

[11] Fabio Bellifemine, Agostino Poggi, and Giovanni Rimassa. *JADE: A FIPA2000 compliant agent development environment*. Proc. of the Fifth Int'l Conf. on Autonomous Agents, pp. 216–217, 2001.

[12] Sanjay Aiyagari. *Security design for NLIP: A universal protocol for AI-enabled systems*. SPIE Proceedings: Disruptive Technologies in Information Sciences IX, 2025.

[13] Xueguang Lyu. *LLMs for multi-agent cooperation*. arXiv preprint arXiv:2504.12345, 2025.

[14] Ning Wang, Yifan Zhang, and Li Chen. *A comprehensive survey on multi-agent cooperative decision-making: Scenarios, approaches, challenges and perspectives*. arXiv preprint arXiv:2503.13415, 2025.

[15] Vinay Kumar. *The open source Model Context Protocol was just updated—here's why it's a big deal*. VentureBeat, March 26, 2025.

[16] Benj Edwards. *MCP: The new "USB-C for AI" that's bringing fierce rivals together*. Ars Technica, April 1, 2025.

[17] Kyle Wiggers. *OpenAI adopts rival Anthropic's standard for connecting AI models to data*. TechCrunch, March 25, 2025.

[18] Ravie Lakshmanan. *Researchers demonstrate how MCP prompt injection can be used for both attack and defense*. The Hacker News, April 30, 2025.

[19] Kasimir Schulz et al. *MCP: Model context pitfalls in an agentic world*. HiddenLayer, April 10, 2025.

[20] Arjun Sha. *What is Model Context Protocol (MCP) explained*. TechTalks, April 14, 2025.

[21] Colin Masson. *Context is the missing link: The emergence of the Model Context Protocol in industrial AI*. ARC Advisory Group, March 25, 2025.

[22] Matthias Bastian. *Anthropic's new open protocol lets AI systems tap into any data source*. The Decoder, November 25, 2024.

[23] Zankar Desai. *Introducing Model Context Protocol (MCP) in Copilot Studio*. Microsoft Copilot Studio Blog, March 19, 2025.

[24] Chris McKenzie. *Getting started: Model Context Protocol*. Medium, December 19, 2024.

[25] Tim Wagner. *Understanding Model Context Protocol*. Vendia, 2025.

[26] Emma Roth. *Anthropic launches tool to connect AI systems directly to datasets*. The Verge, November 25, 2024.

[27] Fiona Jackson. *OpenAI agents now support rival Anthropic's protocol*. TechRepublic, March 28, 2025.

[28] Janakiram MSV. *Why Anthropic's Model Context Protocol is a big step in the evolution of AI agents*. Forbes, November 30, 2024.

[29] Lynn Greiner. *Anthropic introduces the Model Context Protocol*. InfoWorld, November 26, 2024.

[30] Kyle Wiggers. *Google to embrace Anthropic's standard for connecting AI models to data*. TechCrunch, April 9, 2025.

[31] Mark Wallace. *Integrating Model Context Protocol tools with Semantic Kernel: A step-by-step guide*. Medium, March 5, 2025.

[32] Abid Ali Awan. *10 awesome MCP servers*. Medium, March 2025.

[33] 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.

[34] Roy T. Fielding and Julian Reschke. *Hypertext Transfer Protocol (HTTP/1.1): Message syntax and routing*. RFC 7230, 2014.