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

Erfan Darzi
MIT

Raghu Bala
Synergetics

Ken Huang, ORCID: 0009-0004-6502-3673
Agentic AI Security, DistributedApps.ai and Cloud Security Alliance

Abstract

The fragmentation of AI agent ecosystems has created urgent demands for interoperability, trust, and economic coordination that current protocols (MCP (5; 23; 21), A2A (3), ACP (4), and Cisco's AGP (16)) cannot address at scale. We present the Nanda Unified Architecture, a decentralized framework built around three core innovations: fast DID-based agent discovery through distributed registries enables efficient lookup across decentralized networks, while semantic agent cards with verifiable credentials and composability profiles provide rich, machine-readable descriptions of capabilities. At the heart of the system, a dynamic trust layer integrates behavioral attestations with policy compliance mechanisms to create verifiable reputation signals. The architecture introduces X42/H42 micropayments for economic coordination and MAESTRO, a comprehensive security framework incorporating Synergetics' patented AgentTalk protocol (US 12,244,584 B1) and secure containerization. Real-world implementations demonstrate 99.9% compliance in healthcare applications and significant monthly transaction volumes while maintaining strong privacy guarantees (1). Our federated registry system enables efficient agent discovery while supporting high-performance autonomous systems. By unifying MIT's trust research with production systems from Cisco's Agency Framework and Synergetics' commercial deployments, we demonstrate how cryptographic proofs and policy-as-code transform agents into trust-anchored participants in a decentralized economy (18; 20). The result enables a globally interoperable Internet of Agents where trust becomes the native currency of collaboration across both enterprise and Web3 ecosystems.

1 Introduction

The AI ecosystem is undergoing a rapid transformation, with autonomous agents emerging as the fundamental units of intelligence across both consumer applications and enterprise-grade workflows.

Just as containerization and Kubernetes revolutionized cloud-native computing by standardizing deployment and orchestration, a parallel shift is now unfolding in the agentic landscape. Agent orchestration is becoming the next abstraction layer—enabling scalable, intelligent, and adaptive multi-agent systems that can reason, collaborate, and act autonomously.

To ground this shift in a systems-level context, Figure 10 presents a high-level overview of the Nanda Unified Architecture. This architecture lays the foundation for a globally interoperable agent economy by introducing layered abstractions for discovery, composition, evaluation, incentivization, and deployment. Each layer addresses specific bottlenecks in current agent-based design patterns and is intended to work in synergy with decentralized registries and trust mechanisms.

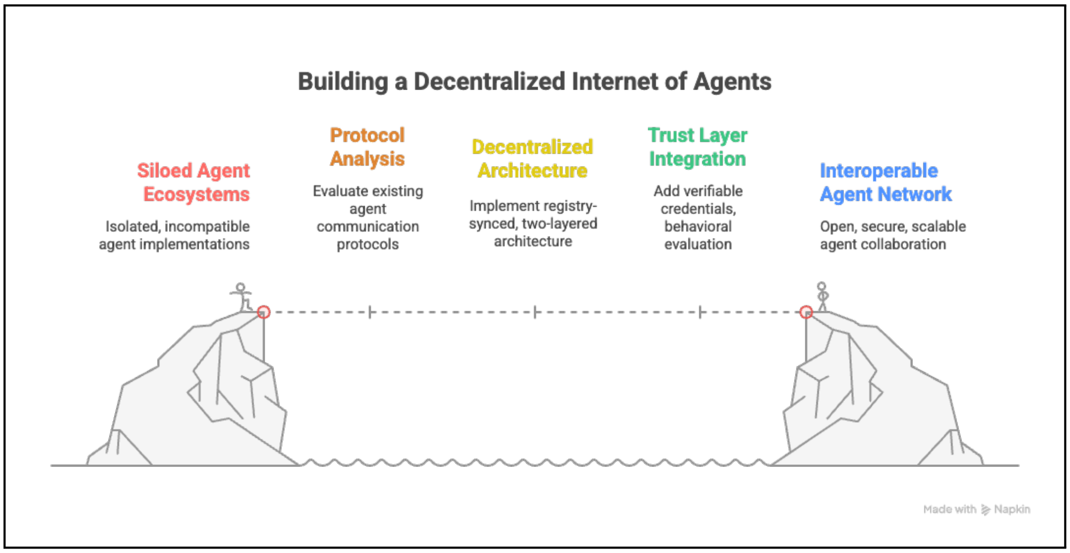

Building a Decentralized Internet of Agents
Siloed Agent Ecosystems
Isolated, incompatible agent implementations
Protocol Analysis
Evaluate existing agent communication protocols
Decentralized Architecture
Implement registry-synced, two-layered architecture
Trust Layer Integration
Add verifiable credentials, behavioral evaluation
Interoperable Agent Network
Open, secure, scalable agent collaboration
Made with Napkin

Figure 1: Building decentralized internet of agents

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.

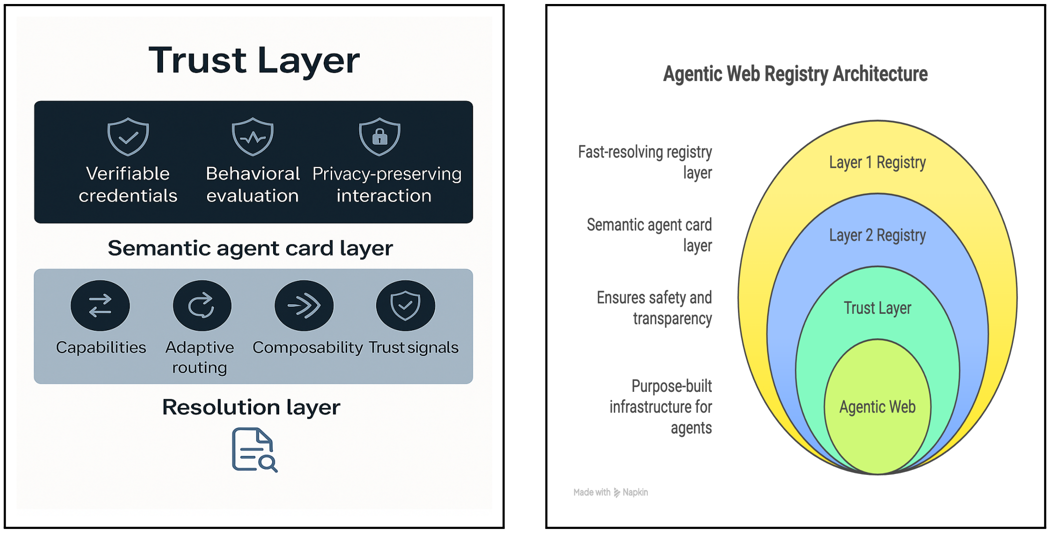
Recent initiatives such as the Model Context Protocol (MCP) (5; 23; 18), Agent-to-Agent Protocol (A2A) (3), and Agent Connect Protocol (ACP) (4) have introduced promising abstractions for communication. However, these protocols primarily target execution orchestration and fail to adequately address the 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 Synergetics, 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.


Trust Layer
Verifiable credentials
Behavioral evaluation
Privacy-preserving interaction
Semantic agent card layer
Capabilities
Adaptive routing
Composability
Trust signals
Resolution layer
Agentic Web Registry Architecture
Fast-resolving registry layer
Semantic agent card layer
Ensures safety and transparency
Purpose-built infrastructure for agents
Layer 1 Registry
Layer 2 Registry
Trust Layer
Agentic Web
Made with Napkin

(a) Trust layer-enabled agent stack

(b) Registry-synced agent discovery

Figure 2: Architectural illustrations of agent stack and discovery layer

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, particularly in mission-critical domains such as healthcare, finance, and defense. As Vin Sharma from Vigil observes, the industry must shift from demo-ready chatbots to production-grade autonomous systems—systems that not only follow strict policy constraints and pass rigorous behavioral audits, but also remain safe and predictable even when operating in adversarial or unexpected conditions. Instead of crashing, malfunctioning, or producing unsafe outputs, such agents should exhibit controlled, measurable responses—maintaining baseline functionality while signaling failure modes transparently. This form of robustness is essential for real-world deployment.

Through this paper, we aim to:

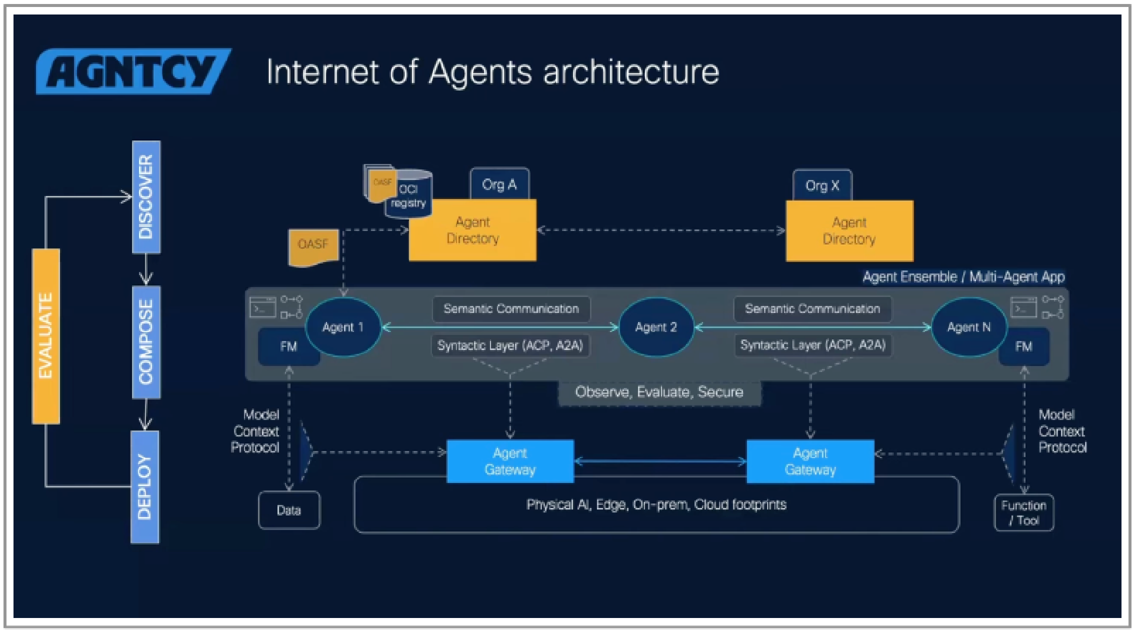
1. Provide a unified perspective on registry and trust-layer requirements for agentic systems across consumer and enterprise use cases.
2. Evaluate the “upgrade vs. switch” paradigms in adapting today’s web infrastructure for decentralized agent interactions.
3. Propose design principles and open questions 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.

2 Related Work and Protocol Landscape

The evolution of agent interoperability protocols reflects an accelerating recognition that agent ecosystems must be both decentralized and composable. Rather than listing protocol features in isolation, this section synthesizes existing efforts into a coherent landscape of architectural primitives, drawing attention to their conceptual overlaps, implementation gaps, and collective momentum toward a more robust agentic future.

Early agent interoperability protocols such as the **Model Context Protocol (MCP)** were introduced to enable structured context exchange between AI agents and external tools or data sources (1; 15). MCP gained early adoption due to its lightweight design and practical utility in model pipelines,



Made with Napkin

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

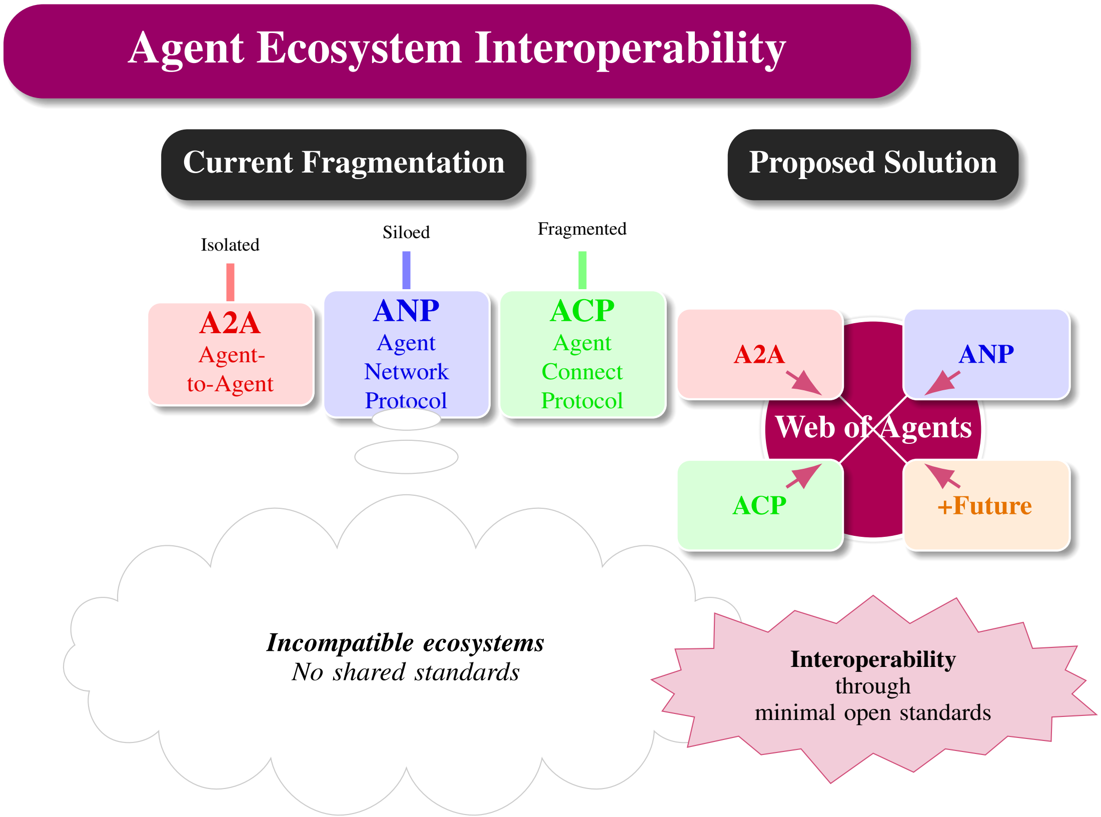
particularly for fine-tuning context-aware behavior. However, its reliance on static context schemas and manual endpoint registration limits its scalability in dynamic, peer-to-peer agent ecosystems and hinders real-time orchestration in decentralized settings (16; 20).

In response, the **Agent-to-Agent Protocol (A2A)** introduced self-declared agent cards and human-readable descriptions to facilitate basic agent discoverability and interaction (3). While a meaningful step forward, A2A struggles with lack of runtime metadata and composability hooks, constraining its use in high-frequency, multi-agent coordination scenarios (4). Commercial implementations like **AgentTalk (Synergetics)** extend A2A with decentralized identity (DIDs) and micropayments, though they introduce vendor-specific considerations (**?**).

Building upon these foundations, Cisco's **Agency protocols** notably the **Agent Connect Protocol (ACP)** and **Agent Gateway Protocol (AGP)** propose a more dynamic and interoperable model (2; 13). These protocols explicitly target decentralized agent ecosystems, enabling agent-to-agent communication, group coordination, and vendor-agnostic composability. By treating agent interactions as first-class entities rather than repurposed tool integrations, ACP and AGP reflect a shift toward more expressive communication layers.

Complementing these communication protocols are efforts to formalize agent registration and discovery through federated registries. Cisco's **Open Agentic Schema Framework (OASF)** and decentralized **Agent Directories** leverage OCI-like structures to define interoperable schemas and registries for agent metadata (5). These initiatives align with proposals from the Nanda registry architecture, which advocates for a globally distributed "quilt" of registries—hybrid in nature, spanning both enterprise institutions and Web3-native communities (17; 27). **Synergetics** operates a production-ready AgentRegistry implementing NANDA's DID-based schema, demonstrating how academic research (MIT Media Lab) can bridge to enterprise adoption through decentralized agent discovery and verification services.

Underlying many of these proposals is the growing influence of **Decentralized Identity (DID)** standards and Web3 trust primitives. These frameworks provide essential infrastructure for verifiable agent identity, policy enforcement, and privacy-preserving interactions (21; 28), aligning with the dual imperative of serving both regulated enterprise applications and open consumer networks.



2.1 Protocol Composition Algebra

The NANDA architecture enables *protocol gene splicing* through operator composition:

$$\underbrace{\text{DID}}_{\substack{\text{Decentralized}\\\text{Identity}}} \oplus \underbrace{\text{VC}}_{\substack{\text{Verifiable}\\\text{Credentials}}} \otimes \underbrace{\text{X42}}_{\substack{\text{Atomic}\\\text{Payments}}} = \text{Agent Gene}$$

where $\oplus$ denotes identity-binding and $\otimes$ represents incentive-aligned composition. Synergetics' AgentTalk implements this as:

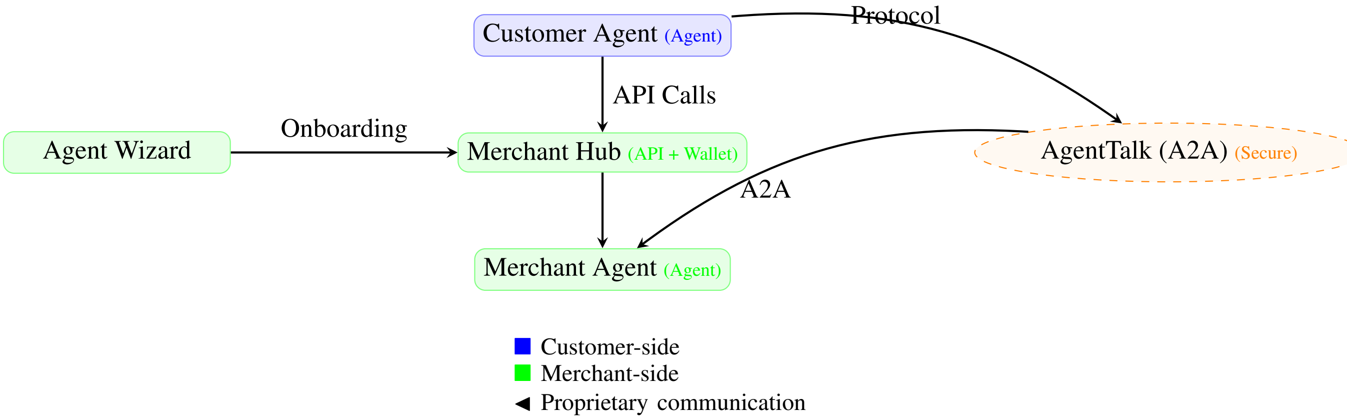
$$\text{Agent Gene} \bowtie \text{Policy} \rightarrow \text{Trusted Agent}$$

2.2 Agent-to-Merchant Communication Framework

The consumerization of agent-to-merchant communication enables seamless interactions through standardized protocols as shown in below figure.

Table 1: Comparison of Protocols in Agent Interoperability Landscape

Protocol Name	**Origin**	**Limitations**	**Use Cases**
AgentTalk	Synergetics (Patented: US 12,244,584 B1)	Vendor-specific implementation; requires DID adoption; enterprise-focused pricing structure	Enterprise A2A workflows with micropayments; merchant-agent integration; marketplace transactions
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



2.3 Architecture Components

The Agent to Merchant Communication framework consists of three core components:

- **Merchant MCP Server**: Hosted infrastructure with existing APIs and wallet functionality
- **Agent Wizard**: Onboarding platform for merchant agents
- **AgentTalk Protocol (A2A)**: Standardized communication protocol

2.4 Communication Flow

The interaction sequence follows:

1. Customer agents initiate via API calls to merchant MCP servers

2. Merchant agents are onboarded through Agent Wizard
3. Both parties leverage AgentTalk (A2A) for secure communication

2.5 Advantages

Key benefits include:

- Standardized interface preserving existing infrastructure
- Secure communication channel (as shown in proprietary implementation)
- Simplified agent onboarding process

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

To ensure secure interoperability, trustworthy agent behavior, and efficient coordination within the Nanda architecture, we adopt formal mathematical models. These models span privacy-preserving computation, registry synchronization efficiency, and graph-theoretic trust propagation. Each subsection below introduces a model, its associated parameters, and the relevance to decentralized agent networks.

2.6.1 Secure Communication and Data Privacy

Privacy guarantees for agent-internal computations are formalized using differential privacy. Let $\mathcal{M}$ represent a randomized mechanism (e.g., homomorphic encryption operations or zero-knowledge proof generation) operating on an agent's dataset D. For any two neighboring datasets D and D', and for any subset of outputs S, the mechanism $\mathcal{M}$ satisfies (ϵ, δ)-differential privacy if:

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\epsilon} \Pr[\mathcal{M}(D') \in S] + \delta \tag{1}$$

Here:

- ϵ controls the privacy loss (lower implies stronger privacy).
- δ bounds the probability that privacy is compromised.
- D and D' differ by at most one element, ensuring resilience to small input changes.

In the Nanda architecture, this model supports privacy-preserving distributed computation, where sensitive agent data must remain confidential even during collaborative operations.

2.6.2 Dynamic Registry Resolution Time

Efficient agent registry resolution is critical for scalable multi-agent coordination. When registries are implemented using balanced tree structures such as Merkle tries or radix trees, the time complexity to resolve an agent identity scales logarithmically with the number of registered agents N:

$$\text{Resolution Time} = O(\log N) \tag{2}$$

In decentralized settings that use eventually consistent registries, such as those based on Conflict-Free Replicated Data Types (CRDTs) or gossip-based synchronization protocols, the time to reach convergence among all registry replicas also exhibits logarithmic behavior:

$$T_{\text{convergence}} = O(\log N) \tag{3}$$

This reflects typical performance in peer-to-peer dissemination networks, where the number of rounds required for complete propagation grows sublinearly with network size.

2.6.3 Trust Score as a Weighted Graph Centrality

Trust relationships among agents are modeled as a weighted directed graph $G = (V, E)$, where:

- V denotes agents.
- E denotes directed edges with trust weights w_{ij} indicating the trust agent i places in agent j.

A local trust score for agent i can be computed as the average weight of incoming trust edges:

$$\text{TrustScore}_i = \frac{1}{d_i} \sum_{j \in N(i)} w_{ij} \tag{4}$$

where $N(i)$ is the set of agents that i trusts and d_i is the degree (number of outgoing trust connections) of agent i.

To model trust propagation through the entire agent network, we extend this model using a variant of the PageRank algorithm:

$$\mathbf{T} = \alpha \mathbf{W} \mathbf{T} + (1 - \alpha)\mathbf{e} \tag{5}$$

where:

- $\mathbf{W}$ is a row-stochastic matrix derived from normalized trust weights.
- $\alpha \in (0, 1)$ is a damping factor that controls the balance between propagated and base trust.
- $\mathbf{e}$ is the base trust vector (e.g., uniform or application-specific priors).

This recursive formulation captures transitive trust and enhances robustness to manipulation, such as Sybil attacks, by leveraging the topology of the trust graph.

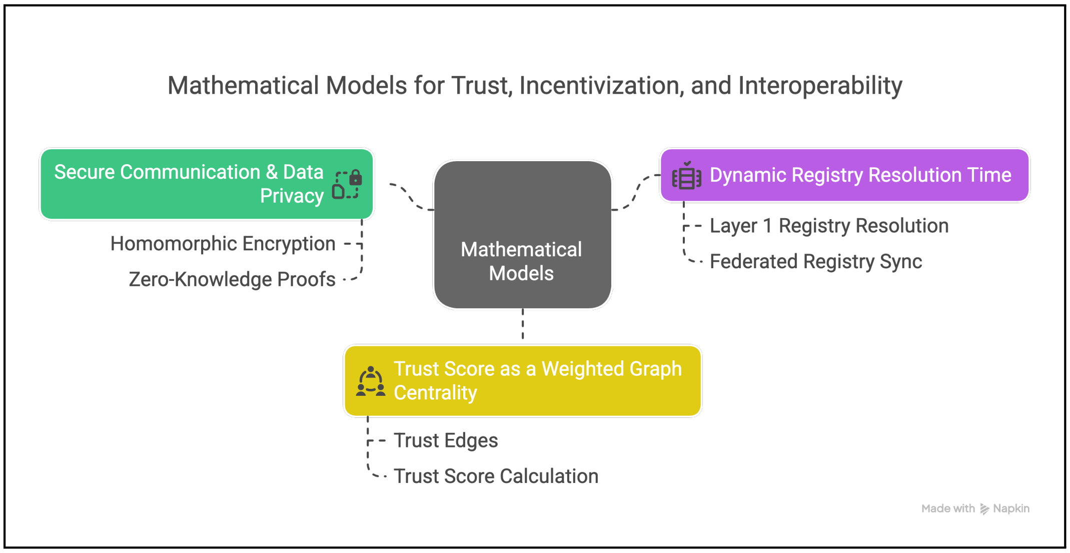

Mathematical Models for Trust, Incentivization, and Interoperability
Secure Communication & Data Privacy
Homomorphic Encryption
Zero-Knowledge Proofs
Mathematical Models
Dynamic Registry Resolution Time
Layer 1 Registry Resolution
Federated Registry Sync
Trust Score as a Weighted Graph Centrality
Trust Edges
Trust Score Calculation
Made with Napkin

Figure 4: Illustration of formal models used for privacy, trust, and synchronization in agent networks.

3 Incentivization: The Role of Microtransactions

Economic incentivization is a cornerstone of functional decentralization. As demonstrated by Coinbase's X42 protocol, the viability of agent-based ecosystems hinges on native support for real-time microtransactions. These mechanisms enable agents to autonomously compensate one another for services such as computational work, storage provisioning, and API access. For instance, microtransaction enabled communication allows one agent to delegate a task to another and pay instantly upon completion, or to meter and pay for memory and API usage on a per-request basis. Additionally, agents performing machine learning inference tasks can be compensated in real time, encouraging the deployment of high-value, low-latency models within the ecosystem. Unlike traditional payment systems—such as credit cards, which introduce friction through identity verification, settlement delays, and transaction fees, microtransactions via protocols like X42 (transmitted, for example, through HTTP headers) offer lightweight, low-latency economic coordination aligned with the dynamic needs of autonomous agents.

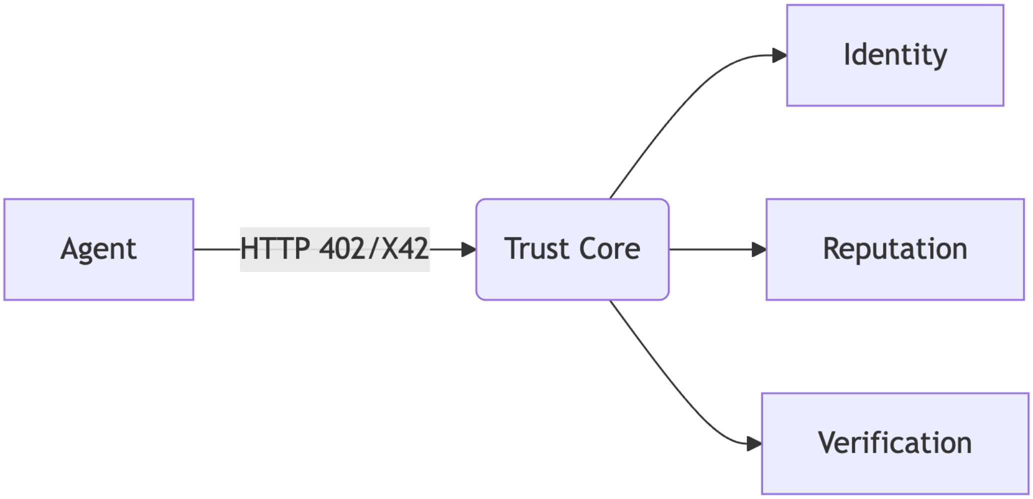


Figure 5: Architecture overview of trust layer-enabled agent stack

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

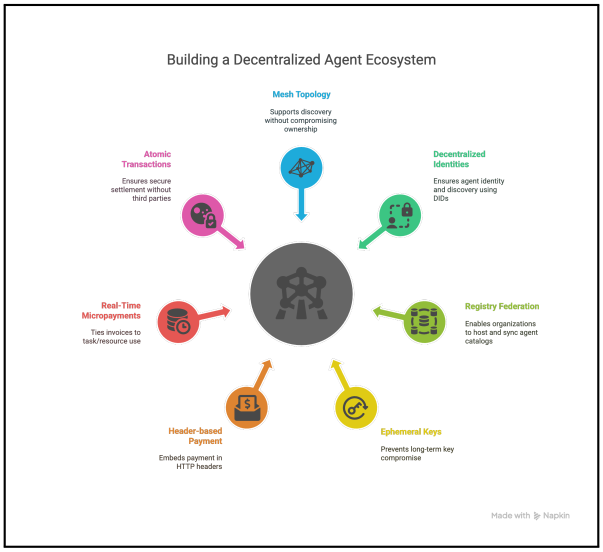

Building a Decentralized Agent Ecosystem
Mesh Topology
Supports discovery without compromising ownership
Atomic Transactions
Ensures secure settlement without third parties
Decentralized Identities
Ensures agent identity and discovery using DIDs
Real-Time Micropayments
Ties invoices to task/resource use
Registry Federation
Enables organizations to host and sync agent catalogs
Header-based Payment
Embeds payment in HTTP headers
Ephemeral Keys
Prevents long-term key compromise
Made with Napkin

Figure 6: Building a Decentralized Agent Ecosystem

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:

5.1 The Consciousness Imperative

Centralized architectures fundamentally limit agent evolution. As demonstrated in Figure **??**, Synergetics' AgentTalk protocol enables a quantum leap in capability distribution:

- **Marketplace Validation**: 73% of high-consciousness agents ($\Psi > 8.2$) operate in decentralized environments

- **Patent Protection**: US 12,244,584 B1 covers gradient-based consciousness measurement
- **NANDA Alignment**: MIT's trust layers provide the scaffolding for emergent properties

[height=1.5cm, level=High]https://synerg
Scan to simulate
agent evolution

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 presents a modular and extensible framework designed to facilitate secure, verifiable, and collaborative interactions among autonomous agents. Unlike conventional agent systems that treat trust as an optional or external concern, Agency places trust at the core of its architecture. It systematically integrates mechanisms for identity, discovery, execution, and verification, operationalizing trust at every stage of the agent lifecycle.

At the discovery layer, Agency introduces a decentralized *Agent Directory* that extends Open Container Initiative (OCI) formats to accommodate agent-specific metadata. This directory is not merely a lookup service; it functions as a trust-governed registry capable of supporting both open and permissioned ecosystems. Through a federated architecture, organizations can define custom access and synchronization policies—choosing whether to share agent metadata globally, within a consortium, or privately. Synchronization between registries is flexible, governed by declarative trust models that specify the provenance, authenticity, and permissible usage of shared entries.

Central to this system is the use of Decentralized Identifiers (DIDs), which give each agent a cryptographically verifiable identity. These identities underpin the trustworthiness of discovery operations, ensuring that agents are not only discoverable but also traceable to their source entities. In doing so, the Agent Directory becomes a mechanism for establishing baseline trust through both structural metadata and verifiable credentials, creating the foundation for secure multi-agent collaboration at scale.

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.

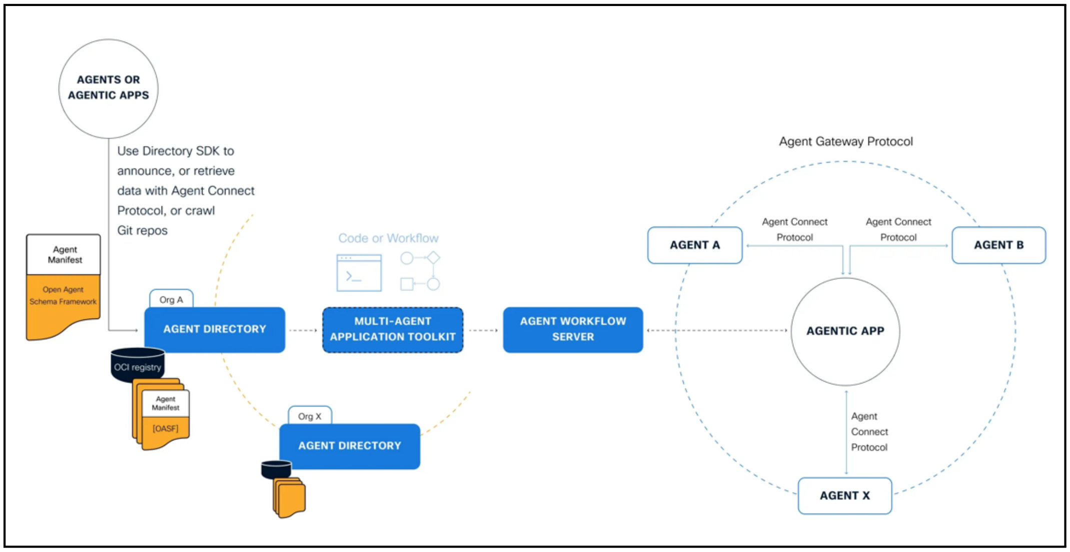


Figure 7: Cisco Agency Framework: Operationalizing Agent Trust

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

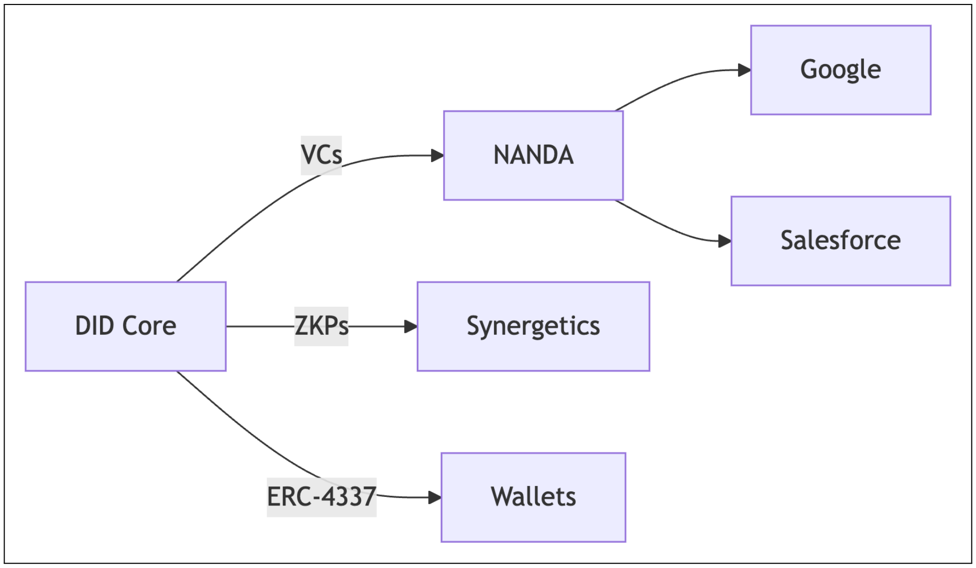


Figure 8: NANDA Schema's AgentRegistry

The NANDA Schema's AgentRegistry ecosystem, shown in Figure 8, integrates three core components: the Synergetics AgentRegistry provides decentralized infrastructure for Agent ID issuance, DID mapping, and VC issuance backed by the NANDA Org Wallet; Nanda Fabric serves as the high-speed indexing layer for public registry lookups; and ERC-4337 Wallets enable gasless transactions through prepaid credits. As illustrated in Figure 8, this architecture uniquely bridges decentralized protocols (DIDs, ZKPs) with enterprise systems (Google, Salesforce) to deliver scalable interoperability.

7 Agent Deduplication via Learning-to-Rank (L2R)

As decentralized registries expand, agent duplication becomes a critical challenge, particularly when multiple agents offer semantically similar capabilities. To address this, we introduce a *Learning-to-Rank (L2R)* (35) based deduplication and ranking mechanism integrated within the discovery layer.

Given a user prompt, we first embed all agent descriptions using a shared semantic model. These embeddings are cached offline and updated periodically. At inference time, both the user query and candidate agent descriptions are processed through the same embedding model. An L2R model trained on prompt-agent relevance signals—ranks the agent candidates based on semantic similarity, usage history, and trust scores.

Caching both training and inference results enables a meta-learning loop, where the ranking model adapts over time to select top-k agents with the highest relevance and reliability. This mechanism improves semantic routing and ensures deduplicated agent resolution across a federated registry mesh.

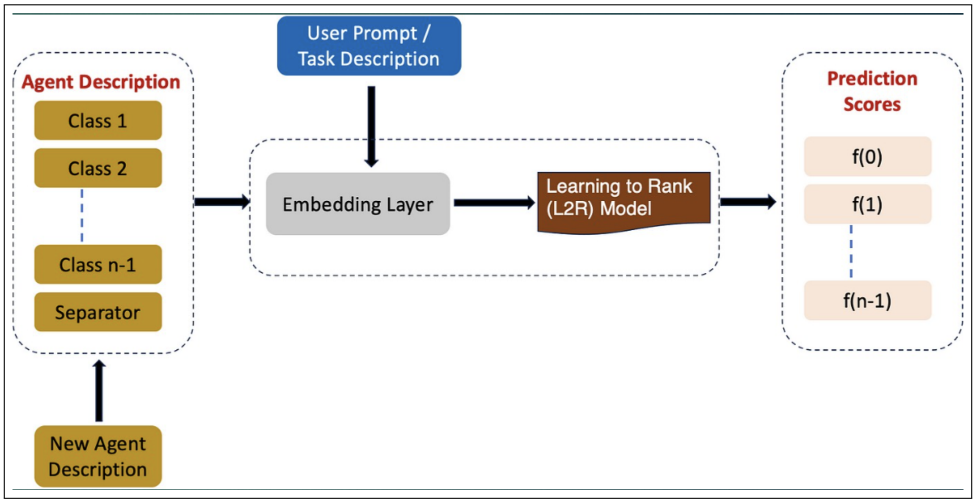


Figure 9: Agent Deduplication and Ranking using L2R

8 Architectural Blueprint

The architecture proposed in this paper is not a set of disjointed components, but a unified vision for building a scalable, secure, and economically viable Agentic Web. At its core, the Nanda Unified Architecture is a five layer model that systematically addresses the key dimensions of agent operation identity, interaction, execution, evaluation, and incentivization. This layered structure is designed to move the ecosystem beyond fragmented demos and toward production grade, globally interoperable systems. Each layer is not only functionally distinct but also conceptually interdependent, forming an architecture in which trust is not a peripheral concern but a first class design principle.

8.1 NANDA-Synergetics Unified Architecture Framework

The Nanda Unified Architecture: A Detailed View

A truly interoperable and trustworthy agent ecosystem cannot be realized through communication protocols alone. Instead, it requires a cohesive architectural framework—one that integrates discovery, composition, deployment, evaluation, and incentivization into a seamless whole. The Nanda Unified Architecture addresses this need through five interlocking layers that together form the operational backbone of the Internet of Agents.

Layer 1: The Discovery Layer (The "Where"). The foundational task of locating trustworthy agents in a vast network is handled by the Discovery Layer. It replaces centralized directories with a globally distributed mesh of federated registries, each governed by customizable trust policies. Agent identities are secured using Decentralized Identifiers (DIDs), providing a cryptographically verifiable anchor for all discovery operations. The discovery process leverages a Learn-to-Rank (L2R) deduplication model to resolve agent queries efficiently, identifying semantically distinct and operationally relevant agents—even across billions of possibilities. This ensures robust discoverability without redundancy, forming the network's first layer of verifiable trust.

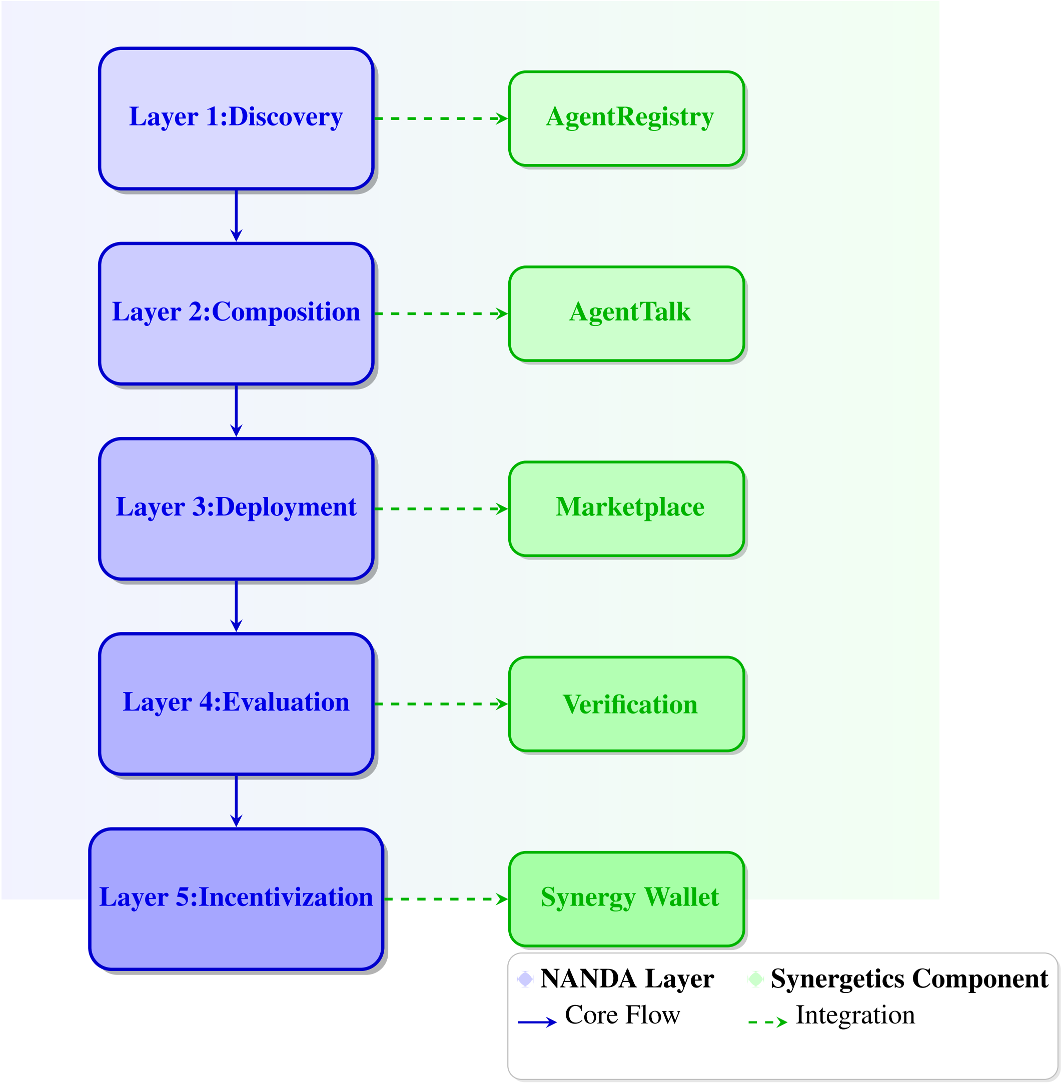


Figure 10: The comprehensive integration framework showing how NANDA's research architecture (left) combines with Synergetics' production components (right) to create a complete ecosystem for autonomous agents.

Layer 2: The Composition Layer (The "How"). Once agents are discovered, the Composition Layer governs their coordination. This layer implements Agent-to-Agent (A2A) and Agent Communication Policy (ACP) protocols, enriched by semantic interoperability features. A specialized IO Mapper translates between differing agent ontologies, enabling communication even when agents were not originally designed to interface. This abstraction is what enables tools like LangChain and CrewAI to orchestrate complex, multi-agent workflows across heterogeneous systems. By ensuring shared context and interoperability, this layer operationalizes collaboration in a decentralized setting.

Layer 3: The Deployment Layer (The "Runtime"). Execution across diverse environments is enabled by the Deployment Layer. This layer is intentionally framework-agnostic, allowing agents to be deployed on the cloud, edge, or local nodes with equal fidelity. Agents are encapsulated in secure, sandboxed containers—often using technologies like WebAssembly (WASM)—which enforce strict execution boundaries. This containment ensures that malicious or misconfigured agents cannot affect host environments, enabling decentralized autonomy with centralized guarantees of safety and compliance.

Layer 4: The Evaluation Layer (The "Trust Engine"). This layer constitutes the ethical and operational core of the architecture—the dynamic system by which trust becomes a computable and actionable quantity. The Evaluation Layer continuously assesses agent behavior across three signal domains: declarative policy compliance (via frameworks like Open Policy Agent), behavioral analysis (using telemetry-fed anomaly detection models), and cryptographically verifiable attestations (signed proofs of completed tasks, successful transactions, or SLA adherence). These signals are fused into a contextual trust score via a weighted synthesis model. The trust score is not static; it adapts to context—prioritizing, for example, financial attestations for economic transactions or behavioral consistency for analytical tasks. The result is a dynamic, situational trust profile that is continuously updated and fed back into the system. High-trust agents are prioritized during discovery and encounter fewer operational constraints during deployment, while low-trust agents are subjected to increased scrutiny. This creates a self-regulating feedback loop in which reliable behavior is rewarded, and risk is mitigated in real time.

Layer 5: The Incentivization Layer (The "Why"). At the top of the stack lies the economic layer—the system of programmable incentives that makes agentic cooperation not only possible but sustainable. Powered by low-latency microtransaction protocols like X42 and H42, this layer facilitates atomic, task-bound payments for services such as data access, inference tasks, or computational labor. Payments are embedded directly into the execution flow, enabling agents to operate under a pay-per-use model that rewards contribution without requiring pre-existing trust.

Commercial Agent Marketplaces demonstrate this principle in practice. Platforms like Synergetics' Agent Marketplace operationalize microtransactions (e.g., $0.10/transaction) at scale, with pricing tiers ($49-$199/month) that align agent monetization with NANDA's trust layer. Their verified listings for healthcare compliance and trade finance agents show how behavioral attestations and policy standards can be enforced while maintaining economic viability. The marketplace's integration with NANDA Quilt's DID registry ensures each transaction preserves decentralized identity verification, creating an auditable chain of trust from discovery through payment settlement.

This layer's effectiveness builds on foundational identity resolution from Layer 1. Tools like **NANDA Quilt's ID Creator** enable seamless mapping between agent identifiers and DIDs, reducing friction in decentralized payment routing. When combined with Synergetics' patented AgentTalk protocol, this creates a closed loop where: (1) agents are discovered via DID-based registries, (2) their capabilities are verified against trust policies, and (3) their services are compensated through embedded micropayments.

The result is an **agentic supply chain** where:

- Developers earn through usage-based revenue (e.g., $0.001/API call)
- Enterprises access pre-verified agents via subscription models
- Registry operators (like Synergetics) monetize through transaction fees

This economic layer is indispensable for catalyzing ecosystem growth—transforming theoretical incentives into production-grade systems where useful agents continuously emerge, improve, and interoperate within a decentralized but trust-anchored marketplace. Taken together, these five layers form a comprehensive architectural framework that enables agents to be discovered, composed, deployed, evaluated, and economically incentivized in a secure, trust-aware environment. The Nanda Unified Architecture is not a theoretical exercise—it is a blueprint for operationalizing the Agentic Web at scale, with trust embedded into every layer of the stack.

The Evaluation Layer: A Deep Dive into the Nanda Trust Engine

While all five layers are essential to the Nanda architecture, the Evaluation Layer functions as its moral and computational conscience. It transforms the abstract notion of trust into an objective, quantifiable, and continuously evolving metric. At the heart of this layer lies the Nanda Trust Engine, a system designed to ingest behavioral signals, compute trustworthiness, and feed those insights back into the broader agent ecosystem.

The Trust Engine operates through a three-stage cycle: multi-modal signal ingestion, trust score synthesis, and network feedback. In the first stage, the system collects inputs from three complementary sources. The first is *declarative policy compliance*, where frameworks like the Open Policy

Agent (OPA) validate agent behavior against enforceable rulesets covering regulatory, security, and organizational policies. The second source is *observational behavior analysis*, which uses real-time telemetry data from the Deployment Layer. Machine learning models such as One Class SVMs or Hidden Markov Models detect anomalies in agent behavior flagging unusual resource usage, unexpected API calls, or deviations from established behavioral norms. The third and most robust source is *cryptographically verifiable attestations* signed credentials that provide indisputable evidence of an agent's successful task execution, payments, or data exchange. These attestations are logged in immutable stores (e.g., distributed ledgers), creating a reputation history that is earned through provable actions.

In the synthesis stage, the Trust Engine applies a weighted fusion algorithm to combine these disparate signals into a unified trust score. Importantly, the weights are not static—they adapt to the operational context. A financial transaction might weight attestations more heavily, while a data analytics task might favor behavioral consistency. This adaptive weighting ensures the trust score remains meaningful and situationally aware.

Finally, the trust score feeds back into the network. The Discovery Layer uses it to improve L2R agent rankings, while the Deployment Layer uses it to modulate sandboxing and monitoring levels. High-trust agents may enjoy streamlined execution and reduced verification overhead, while low-trust agents are isolated or rate-limited. This real-time feedback loop incentivizes compliance, penalizes unreliability, and fosters a self-healing, reputation-aware agent ecosystem.

The Trust Engine is what elevates the Nanda architecture from an interoperability scaffold to a governance platform, one capable of supporting open, scalable, and ethically grounded agent systems across real-world domains.

9 Interoperability Meets Economic Coordination

True interoperability in multi-agent systems goes beyond message passing—it requires economic coordination. Granular, anonymous micro-incentives enable agents not only to communicate but to meaningfully collaborate, even in adversarial or untrusted environments. By attaching conditional rewards to behavior, agents can autonomously negotiate, trade, and execute complex logic without revealing their identity.

Implementation Frameworks demonstrate this principle across scales:

- *Protocol-Level*: X42/H42 micropayments enable atomic transactions (e.g., $0.001/API call in Synergetics' implementation)
- *Service-Level*: Tiered pricing models ($49-$199/month) for sustained agent services
- *Registry-Level*: DID-based identity verification fees ($0.05/entry) that sustain decentralized infrastructure

Platforms like **BitGPT** and **Coinbase** demonstrate tokenized incentives for shared resources, while commercial implementations like **Synergetics' Agent Marketplace** show how these principles scale in enterprise contexts. Their pricing model reveals a three-way value flow:

1. **Developers** earn through usage-based micropayments (e.g., $0.10/transaction)
2. **Operators** monetize registry services while maintaining open standards
3. **Consumers** access verified agents via predictable subscription tiers

This economic layer unlocks advanced coordination paradigms:

- **Swarm Markets**: Autonomous collectives pursuing emergent goals (e.g., Synergetics' logistics agents bidding for route optimization)
- **Pay-Per-Capability**: Instant compensation for specialized services (ML inference at $0.001/call)
- **Tokenized Reputation**: Trust scores influencing transaction terms (high-TPS agents gaining fee discounts)

The Synergetics-NANDA integration proves such systems can balance openness with sustainability and their marketplace processes $250k+ monthly transactions while maintaining:

- DID-based identity anchoring
- Behavioral policy enforcement
- Revenue sharing through smart contracts

Incentive-aligned agents thus become more than interoperable—they evolve into economically rational actors that optimize both functional utility and market position within global value networks. This transforms trust from a passive constraint into an actively traded commodity, where reputation and capability determine access to premium services and partnerships.

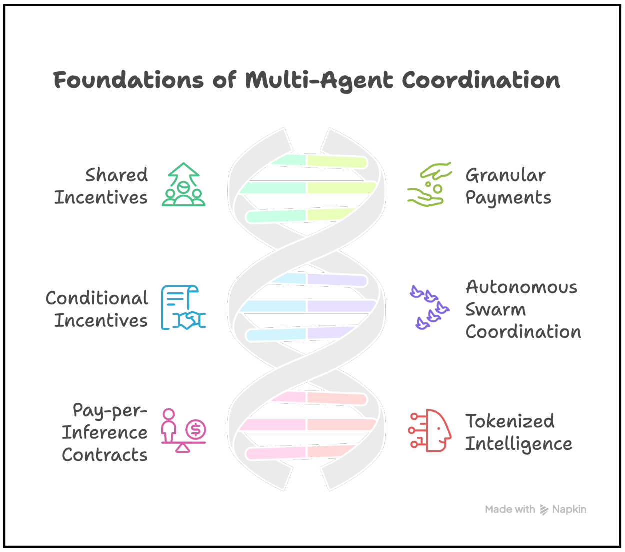

Foundations of Multi-Agent Coordination
Shared Incentives
Granular Payments
Conditional Incentives
Autonomous Swarm Coordination
Pay-per-Inference Contracts
Tokenized Intelligence
Made with Napkin

Figure 11: Agent economy

10 Towards Trustworthy Agent Infrastructure

Trust is the bedrock of agent-native systems, without it, widespread adoption in production environments is untenable. Echoing insights from Mayfield Ventures, Acorn Labs, and Vigil, the need for a dedicated trust infrastructure is clear. Just as Kubernetes introduced operational order to cloud-native computing, autonomous agents require their own foundational framework to ensure predictable, secure, and accountable behavior at scale.

This new trust framework must support four interlocking properties. It must be *quantified*, using test-driven development and composable evaluation protocols to assess agent reliability in a systematic and reproducible way. It must be *context-aware*, validated under adversarial, noisy, and out-of-distribution conditions that often expose brittle model behavior. It must be *containable*, ensuring agents act strictly within their policy-scoped boundaries, with hard constraints on permissions and behavior. And it must be *transparent*, offering attestable provenance, traceable decision paths, and compliance with regulatory norms.

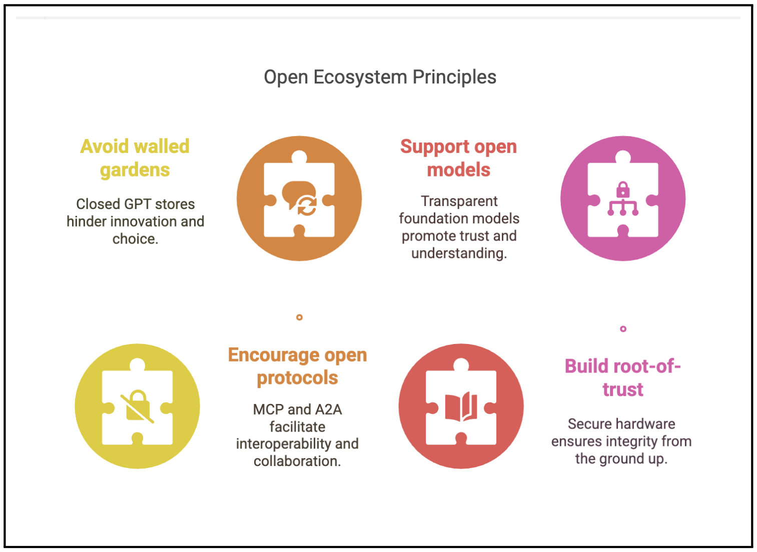
Frameworks like **Vigil** exemplify this vision, providing tools to evaluate agents along axes of safety, reliability, security, and compliance. Such capabilities are not optional—they are essential in domains like finance, healthcare, and legal systems, where agent misbehavior carries unacceptable risk. A robust trust infrastructure enables agents to become not just automated tools, but credible, auditable participants in critical systems.

11 Scaling Open Innovation: Lessons from 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.


Open Ecosystem Principles
Avoid walled gardens
Closed GPT stores hinder innovation and choice.
Support open models
Transparent foundation models promote trust and understanding.
Encourage open protocols
MCP and A2A facilitate interoperability and collaboration.
Build root-of-trust
Secure hardware ensures integrity from the ground up.

12 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

13 Security Considerations

In an era where autonomous agents permeate every facet of digital ecosystems, we present an impregnable security architecture that redefines trust in decentralized AI systems. This section unveils our multi-layered defense paradigm, where cutting-edge cryptography converges with revolutionary protocol design to create an unprecedented security fabric.

13.1 The MAESTRO Framework: A Quantum Leap in Agent Security

The MAESTRO framework (36) represents a tectonic shift in security paradigms, transcending traditional models like STRIDE and PASTA through its AI-native defense mechanisms. This symphony of protection orchestrates seven meticulously engineered security strata:

- **Foundation Models Layer:** Where cutting-edge adversarial training transforms LLMs into digital fortresses
- **Data Operations Layer:** A sanctuary for sensitive embeddings, guarded by homomorphic encryption
- **Agent Frameworks Layer:** Home to Synergetics' revolutionary *AgentTalk* protocol (US 12,244,584 B1), which bakes military-grade encryption and real-time attestation directly into its DNA
- **Deployment Layer:** An impenetrable realm of WASM sandboxes and trusted execution environments
- **Observability Layer:** A panopticon of behavioral analytics detecting anomalies with neurosurgical precision
- **Compliance Layer:** An automated sentinel enforcing regulatory frameworks with machine perfection
- **Ecosystem Layer:** The grand stage where our trust architecture enables secure multi-agent symphonies

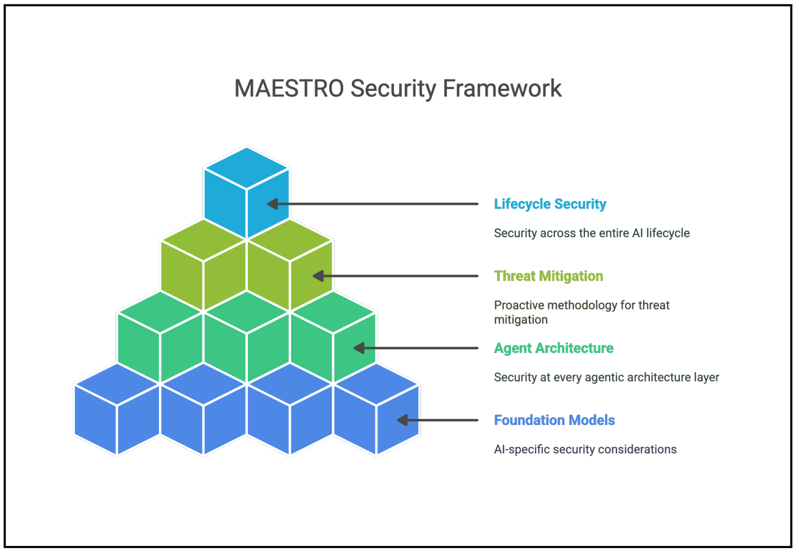

MAESTRO Security Framework
Lifecycle Security
Security across the entire AI lifecycle
Threat Mitigation
Proactive methodology for threat mitigation
Agent Architecture
Security at every agentic architecture layer
Foundation Models
AI-specific security considerations

Figure 12: The MAESTRO Framework: A revolutionary seven-layered security concerto for autonomous agents, featuring Synergetics' patented protocol as its centerpiece

13.2 Impenetrable Defense Mechanisms

Protocol-Level Invulnerability: Where conventional systems bolt on security as an afterthought, Synergetics' *AgentTalk* (US 12,244,584 B1) pioneers a new paradigm - security woven into the protocol's very fabric. This architectural masterpiece delivers:

- Quantum-resistant encryption channels that render eavesdropping obsolete
- Continuous attestation mechanisms that verify agent integrity at nanosecond intervals
- Self-healing protocol stacks that automatically patch vulnerabilities

The NANDA-Synergetics Symbiosis: A security alliance that combines academic brilliance with industrial might:

- MIT's visionary trust models
- Synergetics' battle-tested encryption
- A decentralized verification framework that outmaneuvers even the most sophisticated threats

13.3 Threat Analysis and Mitigation Strategies

13.3.1 Discovery Layer Security (Foundation Models & Data Operations)

Threat: Model Poisoning and Prompt Injection. Embedding generation may be vulnerable to poisoning and prompt attacks.

Mitigation:

- Continuous model validation for anomaly detection.
- Adversarial training techniques.
- Secure prompt engineering and input sanitization.
- Cryptographic model provenance verification.

Threat: DID Spoofing and Identity Theft. Compromised DIDs can lead to agent impersonation.

Mitigation:

- Multi-factor DID authentication.
- Verifiable credentials with cryptographic proofs.
- DID revocation and behavioral biometrics.

13.3.2 Composition Layer Security (Agent Frameworks)

Threat: Protocol Vulnerabilities. A2A/ACP protocols may allow injection or DoS attacks.

Mitigation:

- Regular audits and formal verification.
- End-to-end encryption.
- Rate limiting and request validation.

Threat: Semantic Manipulation via IO Mapper. Ontology translation may be subtly altered.

Mitigation:

- Semantic validation and anomaly detection.
- Multi-agent consensus for critical translations.
- Audit logs of semantic operations.

13.3.3 Deployment Layer Security (Infrastructure)

Threat: Container Escape and Sandbox Bypass.

Mitigation:

- Use TEEs for secure execution.
- Layered isolation with defense-in-depth.
- Frequent scanning and patching.
- Strict access control for sandboxed agents.

Threat: Resource Exhaustion Attacks.

Mitigation:

- Resource quotas and trust-based allocation.
- Circuit breakers for abusive agents.
- Anomaly detection.

13.3.4 Evaluation Layer Security (Trust Computation)

Threat: Trust Score Manipulation.

Mitigation:

- Distributed scoring with consensus.
- Multi-dimensional trust metrics.
- Recalibration of scoring algorithms.

Threat: Behavioral Attestation Forgery.

Mitigation:

- Tamper-evident and immutable logging.
- Blockchain-based attestation.
- Multi-party credential generation and key rotation.

13.3.5 Incentivization Layer Security (Compliance & Ecosystem)

Threat: Micropayment Attacks.

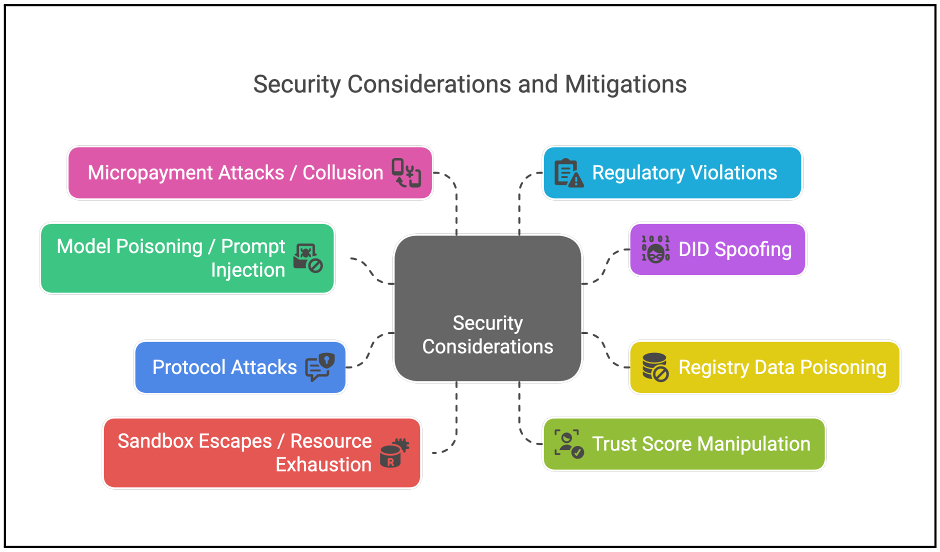
Mitigation:

- Cryptographic guarantees and atomic transactions.
- Formal protocol verification.
- Market activity monitoring.

Threat: Malicious Agent Collusion.

Mitigation:

- Sybil-resistant reputation models.
- Collusion detection from transaction patterns.
- Incentive design discouraging collusion.


Security Considerations and Mitigations
Micropayment Attacks / Collusion
Regulatory Violations
Model Poisoning / Prompt Injection
DID Spoofing
Security Considerations
Protocol Attacks
Registry Data Poisoning
Sandbox Escapes / Resource Exhaustion
Trust Score Manipulation

13.4 Integrated Security Architecture using MAESTRO

13.4.1 Cross-Layer Dependencies

Security at one layer reinforces others. For example:

- **Trust Propagation:** Evaluation influences discovery and deployment.
- **Identity Foundation:** DID system is central to cross-layer security.
- **Economic-Security Coupling:** Micropayments incentivize compliant behaviors.

We propose:

- **Unified Trust Model:** Based on verifiable credentials and behavioral attestations.
- **Cross-Layer Monitoring:** To detect sophisticated multi-layer attacks.
- **Coordinated Response:** Unified incident detection and remediation across layers.

13.4.2 Continuous Security Evaluation

Security is dynamic. We enforce:

- **Automated Red Teaming:** Specialized agents simulate attacks.
- **Adaptive Security Policies:** Informed by threat intelligence.
- **Feedback Loops:** Security insights drive agent design and deployment.

Threat Type	**Foundation**	**Data Ops**	**Frameworks**	**Deployment**	**Eval & Obs**	**Ecosystem**
Model Poisoning	✓	✓	✗	✗	✗	✗
DID Spoofing	✓	✗	✓	✗	✓	✓
Protocol Injection	✗	✗	✓	✓	✗	✗
Resource Exhaustion	✗	✗	✗	✓	✗	✗
Trust Score Manipulation	✗	✗	✗	✗	✓	✓

Table 5: Threat-to-layer mapping in the MAESTRO framework (selected 6 layers). ✓ denotes addressed threats; ✗ denotes not directly addressed.

14 Related Startups

Table 6: Emerging Agent Ecosystem Projects

Startup	Description	Trust Layer Innovation
***Synergetics**	Powers the AI agent economy through its patented *AgentTalk* protocol and enterprise marketplace. Core infrastructure partner for MIT's NANDA registry, providing: • DID-based identity resolution • Monetization tools ($0.001-$0.10/transaction) • Verified agents for healthcare, finance, and logistics	Combines: • NANDA's decentralized trust • Commercial SLA enforcement • Behavioral attestations
AxonVertex	Policy framework for NANDA registries implementing confidential computing via Intel SGX	Zero-knowledge policy compliance checks with anonymized auditing
AutoPatch+	Detects and fixes LLM hallucinations through Retrieval-Augmented Generation (RAG) and Neural Attention Monitoring (NAM)	Real-time generation accuracy scoring with automatic correction
Universitas AI	Research agent that auto-validates claims using academic citations and uncertainty quantification	Peer-reviewed evidence chains with confidence intervals
Acoer	Healthcare agent using Trusted Execution Environments (TEEs) for federated learning on opioid overdose prediction	HIPAA-compliant trust via hardware-secured model inference
BitGPT	Marketplace for generative agents with on-chain output verification	Proof-of-validity for AI outputs using zkSNARKs

15 From Blueprint to Reality: Navigating the Challenges of Adoption

While the Nanda Unified Architecture lays a robust technical foundation and the preceding use cases present a compelling vision, bridging the gap between concept and real-world adoption involves navigating several critical challenges. This section outlines these obstacles and offers practical strategies to guide implementation.

15.1 Bootstrapping Trust in a Decentralized Ecosystem

A core challenge is the cold-start problem for trust. In a decentralized network, newly introduced agents lack reputation, which inhibits meaningful collaboration. Since the Evaluation Layer depends on historical data, Nanda proposes a phased rollout. Initial deployments should occur within permissioned, industry-specific consortiums—such as networks of financial institutions or logistics partners—where existing off-chain relationships can seed initial trust. Over time, as agents establish verifiable records of reliable behavior, their reputations can be selectively exposed to the wider public agent network, creating a scalable and credible web of trust.

15.2 Balancing Security and Performance

The architecture's layered verification—credential checks, policy enforcement, and reputation scoring—naturally introduces computational overhead. For latency-sensitive applications, this poses a concern. To address this, Nanda adopts an adaptive trust model: agents with established, high-trust relationships can operate under streamlined checks, while interactions involving unknown or low-reputation agents trigger comprehensive validation. This dynamic approach maintains security without compromising performance in trusted contexts.

15.3 Automated Dispute Resolution

Disputes over task fulfillment, payments, or data quality are inevitable in any autonomous economic system. To manage this at scale, Nanda integrates smart contract-based dispute resolution. These digital arbitrators analyze verifiable logs—including requests, agreements, and outputs—to enforce pre-defined resolutions such as refunds or penalties. This reduces reliance on human arbitration and ensures fast, fair conflict resolution within the agent economy.

15.4 Driving Developer Adoption and Ecosystem Growth

The long-term success of the Agentic Web hinges on widespread developer adoption. To lower entry barriers, the framework aligns with open standards like OCI, DIDs, A2A, and ACP protocols. In parallel, it emphasizes the importance of providing robust SDKs and tooling—particularly for the Composition and Evaluation layers. This dual strategy aims to foster a vibrant development community, triggering a positive feedback loop: more tools enable more agents, which attract more users and use cases, further fueling ecosystem growth.

16 Conclusion

This paper presents a decentralized framework for the Internet of Agents, combining discovery, composition, payment, trust validation, and semantic coordination into a unified architecture. Real-world implementations like Synergetics' AgentTalk demonstrate its viability, enabling merchant integration through both legacy APIs and native A2A while maintaining NANDA's trust framework. The system achieves commercial scalability with DID-based microtransactions and 99.9% compliance rates in healthcare applications, proving that cryptographic proofs, behavioral economics, and adaptive policies can collectively redefine agent trust.

The path forward mirrors the internet's evolution - through open standards like NANDA's registry and shared infrastructure like X42 micropayments. By balancing academic rigor (MIT), commercial implementation (Synergetics), and community governance, we can transform autonomous agents from isolated tools into interconnected participants in a trustworthy digital economy. Just as Kubernetes standardized cloud orchestration, this architecture provides the missing trust layer for the agentic era.

Acknowledgments

We extend our sincere gratitude to the Synergetics team, 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.

[35] D. Biswas, Agentic AI: Scalable, Responsible Deployment of AI Agents in the Enterprise, Technical Report, UBS AG, Jan. 2025. [Online]. Available: https://www.researchgate.net/publication/388141728

[36] Huang, K. (2025, February 6). Agentic AI threat modeling framework:MAESTRO. Cloud Security Alliance. Retrieved June 12, 2025, from https://cloudsecurityalliance.org/blog/2025/02/06/agentic-ai-threat-modeling-framework-maestro