AgNet: A Novel AI Agent Network Architecture

Manoj Gupta
manoj@plotch.ai
Plotch.ai

Vikram Acharya
vikram@plotch.ai
Plotch.ai

Sai Sujan
sai@plotch.ai
Plotch.ai

Ayush Mittal
ayush@plotch.ai
Plotch.ai

***Abstract*—A big agentic switch is happening globally due to AI (Artificial Intelligence) whereby consumers and enterprises are switching to AI agents for task completions in an autonomous way. This paradigm shift due to AI is leading to proliferation of AI agents in enterprises requiring innovative agent network architectures for better discovery, communication and security of these agents. Here we present a novel agent network architecture where we introduce innovative concepts like agent registry, agent name server, agent text transfer protocol (ATTP), agent private cloud and agent gateway. Quite a lot of parallels have been drawn from current world wide web (www) Internet architecture where similar components exist to support discovery, security and communication.**

***Index Terms*—AgNet, AI Agent Network Architecture, AI Agents, Agent Discovery, Agent Registry, Agent Name Server (ANS), Agent Text Transfer Protocol (ATTP), Agent Private Cloud (APC), Zero Trust Architecture (ZTA), Agent Security, Communication Protocol, Scalability, Interoperability.**

I. Introduction

A crucial component of contemporary technology, artificial intelligence (AI) is spurring innovation and revolutionizing sectors all over the world. Fundamentally, AI is the process by which machines mimic human intellect in order to carry out functions like learning, thinking, and decision-making. AI agents are among the most revolutionary developments in AI; they are self-governing software systems created to interact with their surroundings, make choices, and carry out activities with little assistance from humans.

The proliferation of AI agents within enterprises has ushered in a new era of productivity and automation, redefining traditional workforce dynamics. The number of AI agents in large organizations may exceed millions which will create challenges in task performance, deployment, security and scalability. To manage such large network of AI agents in any organization would require a novel network architecture which would allow such a network to function seamlessly. Here we present AgNet which is a novel network architecture for AI agents.

II. Background

AI agents are essential for optimizing processes and boosting efficiency in multiple areas, ranging from customer service bots to advanced data analytics systems. These agents can operate both independently and as members of multi-agent systems, where their cooperation facilitates the completion of complex, interrelated tasks [1]. This development has led to Agentic AI, a concept that sees AI agents as collaborators, smoothly blending with human teams to enhance organizational productivity [6].

The influence of Agentic AI on productivity within enterprises is significant. By automating repetitive and time-consuming tasks, organizations can shift human resources to focus on strategic and creative initiatives. Additionally, Agentic AI supports better decision-making via data-driven insights, boosts operational efficiency, and promotes scalability in workflows. With organizations progressively implementing networks that incorporate millions of AI agents, the necessity for resilient architectures such as AgNet becomes essential to guarantee smooth communication, security, and scalability.

This change in approach requires creative architectures that can absorb many AI agents into a single ecosystem [9]. The major challenges are secure communication, dynamic agent discovery and operational scalability [8]. AgNet introduces a new framework to address these needs, inspired by the principles that govern internet protocols and infrastructure.

The AgNet framework presents essential elements like the Agent Registry, Agent Name Server (ANS), Agent Text Transfer Protocol (ATTP), and Agent Private Cloud (APC). Every one of these elements is crafted to replicate features similar to the World Wide Web, modified for the specific needs of AI agent networks. This blend of scalability, composability, and security renders AgNet a strong solution for the changing environment of agentic AI.

III. Network Architecture

AgNet is built on four principles: efficiency, security, scalability and composability. By integrating key subsystems for robust agent detection, communication and security you can realise these.

Modular design means you can add new agents and subsystems without affecting the core of the network. By breaking the architecture into separate but connected components AgNet allows for adaptability and agility to support small and large networks of AI agents. Plus modular design means you can deploy to specific organisations or sectors.

Security is built in to ensure safe operation at every level. Zero trust for authentication and authorisation and strong encryption for data in transit and at rest. AgNet is interoperable

so agents can talk to each other across different systems and older protocols.

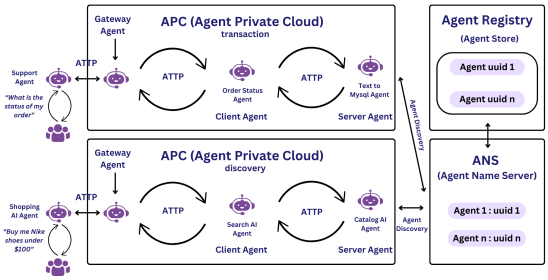

Fig. 1. Agent Network Architecture

A. Agent Discovery

Agent discovery is a key part of the AgNet framework to enable safe and efficient inter-agent interactions in a large network of AI agents. When agents can be in the millions, strong agent identification, resolution and communication is crucial. This section will cover the key components of the agent discovery process—Agent UUID, Agent Registry and Agent Name Server (ANS)—and how they help in seamless communication in the AgNet ecosystem.

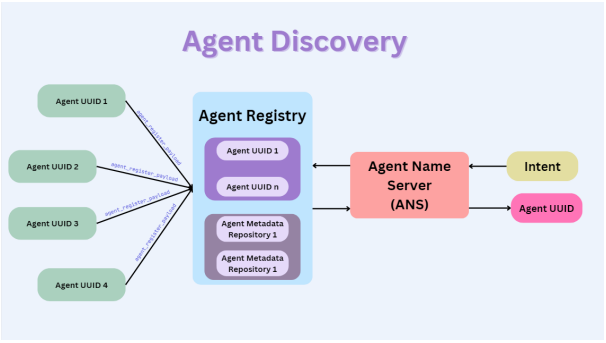

Fig. 2. Agent Discovery Framework

a) Agent UUID: The Agent Universally Unique Identifier (Agent UUID) is very important in AgNet framework, a unique and immutable identifier for every AI agent globally. This identifier helps in identifying, interacting and monitoring agents across the network, a solid foundation for agent based systems.

b) Design Principles: The Agent UUID is optimized to fulfill three main criteria to be performant and reliable. Top three criteria are: worldwide uniqueness, durability and standard compliance, — all of them are essential. Uniqueness guarantees that no UUID will ever be repeated in the network. Durability makes sure that the UUID continues to remain the same across the agent's life time, which helps in long term tracking, auditing and accountability. You have a standard-compliant system, which means that it will work out of the box, addressing this critical issue of interoperability between agents. Its a very important part of agent communication.

c) Role in Agent Discovery: Communication which is achieved using the agent UUID plays key role in communication and agent discovery. So in a system with many agents, wrong one shouldn't get called, above all, the call is quickest and secure. The agent discovery is performed using the Agent UUID wherein every agent is given a unique UUID that is mapping over cryptographic tokens and metadata resulting in better agent authentication and secure transactions among the entities of the network.

B. Agent Registry

The Agent Registry is the central database where all the metadata for every agent in the AgNet ecosystem is stored. This is an important layer of the ecosystem. Plays a key role in agent identification, capability management and lifecycle monitoring.

a) Architecture of the Agent Registry: The architecture of the agent registry is designed to make it easy to handle and access agent metadata in the AI Agent Network (AgNet). At the base of it is a **Metadata Repository** which is a central database to store all the metadata of every agent. It stores key information such as the UUID, the agent's capabilities and functions, its current status and availability and the communication endpoints (urls and APIs) of the agent. This central repository ensures consistency and reliability of metadata.

To keep the system up to date the registry has **Dynamic Updates** so changes can be made in real time. Whether agents are launched, updated or retired the registry reflects these changes instantly so the network is always current and working.

b) Key Functionalities of the Registry: The agent registry fulfills several pivotal functions to maintain the integrity and efficiency of AgNet:

- **Agent Discovery**: The registry is the backend for the Agent Name Server (ANS). It resolves agent queries and connects users to the right agents using metadata.
- **Capability Matching**: By analyzing the capabilities registered for each agent, the registry matches incoming queries or tasks to the most suitable agents, ensuring optimal resource allocation and task execution.
- **Agent Lifecycle Management**: The registry follows the entire lifecycle of agents from creation and deployment to modification and decommissioning. This ensures metadata is consistent across the network and reduces errors and inefficiencies.
- **Integration with the Agent Name Server**: The registry is a data source for the ANS. When a query is received the ANS looks up the registry to get the UUID and endpoint of the agent and then communicates and executes the task.

c) Advantages of the Agent Registry: The agent registry is good for the whole AgNet architecture. It's a central database from call to managing agents. It's the base of the

whole AgNet system. Very important for infrastructure as this is one of the first point of agent discovery and workflow.

To sum up, the agent registry is a part of the AgNet framework, metadata handling, agent identification and lifecycle management. Its connection to the Agent Name Server and real time updates means the network is scalable, consistent and adaptive to the changing world of AI.

C. Agent Name Server (ANS)

The Agent Name Server (ANS) is the discovery engine in the AgNet architecture and acts as an intermediary that maps user intents to agents. It hides the complexity of the discovery process so Client-Agents can find and talk to Server-Agents.

a) Core Responsibilities: ANS is important for intent resolution, caching and fallback. Intent resolution is the process of taking high level intents from Client-Agents and turning them into actionable endpoints by querying the Agent Registry. Caching allows ANS to keep a local cache of frequently used agent information so performance and delay is minimized. When cache misses happen the fallback system allows ANS to actively query the Agent Registry and get/refresh the required data.

b) Query Flow: The query process begins when a Client-Agent sends a query with the intent and entities to the ANS. ANS checks its cache for the Agent UUID and endpoint. If the cache doesn't have the data, ANS queries the Agent Registry and gets the data. Then the UUID and endpoint is sent back to the Client-Agent.

c) Integration with ATTP: The ANS is tightly coupled with the ATTP protocol to enable secure communication between agents. All queries and responses are sent as ATTP messages that include encryption, authentication and versioning headers. This integration provides a seamless and secure interaction framework within the AgNet.

d) Security Features: To have a safe operational environment the ANS has various security features. Authentication is done by verifying the identity of the querying Client-Agents through authentication tokens in the ATTP headers. Access control ensures only authorized users can search or retrieve sensitive data. Data integrity is maintained through cryptographic methods to prevent data from being modified during transmission.

e) Advantages: ANS simplifies the discovery process for Client-Agents by abstracting away direct interaction with the registry. Its caching and resolution mechanisms improve overall performance of the AgNet. Fallback mechanisms and robust error handling capabilities improve fault tolerance so the system can work under various conditions.

D. Agent Communication

Agent communication in AgNet is an advanced, secure and efficient way for AI agents to talk to each other across the network. This is done through the **Agent Text Transfer Protocol (ATTP)** which is the heart of the system. To ensure compatibility with different environments and older systems the architecture has the **ATTP Adaptor** which bridges protocol gaps and adds features.

This section will go into detail about the communication protocols and components that make agent to agent communication reliable, scalable and secure.

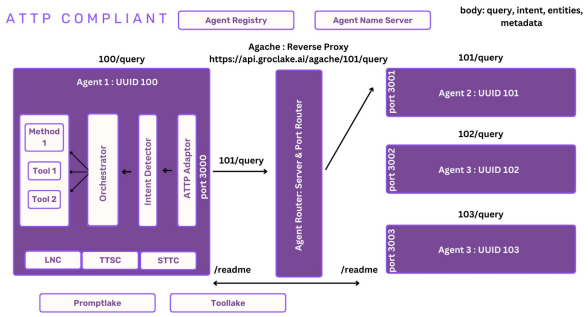

Fig. 3. Agent Communication Framework

1) Agent Text Transfer Protocol (ATTP): The **Agent Text Transfer Protocol (ATTP)** is AgNet fabric's agent to agent communication supervisor. It aims to be a light, secure, and efficient means of transporting data at low latency and high standards of safety.

a) Key Features: It is high performance, low latency, and can scale to millions of simultaneous interactions. Security is a fundamental quality, and encryption, authentication and tokenization are features baked into the protocol to ensure the secure transfer of data. This modular process makes the protocol easy to implement as you can plug in new features or extend agent capabilities.

b) Protocol Structure: ATTP message consists of two components: header and body. The header includes version, message type, agent-specific identifiers such as Agent Process Control ID (APCID) and Agent UUID, authentication tokens, timestamps, and encryption flags as metadata. Data like the payload carries the body or the query or task to be processed along with relevant metadata such as identified intent and entities. A diagram of this is shown in Figure 4.

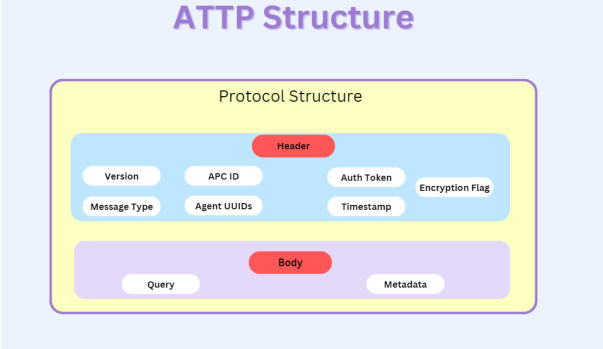

Fig. 4. Agent Communication Framework

c) Communication Flow: The communication flow in ATTP involves several steps to process the request correctly.

First the Client-Agent extracts the intent and entities from the user query. The extracted message is then sent to the Server-Agent through the **Agent Name Server (ANS)**. The Server-Agent processes the request and sends the response back to the Client-Agent and that's the communication cycle.

d) Advantages: ATTP has many benefits for the Ag-Net ecosystem. It facilitates more seamless interaction, as it provides standardized communication to better integrate as a whole system between disparate agents. Intelligent routing and caching are also leveraged to optimize details and reduce latency in data exchange. And notion has strong security measures like encryption and authentication that maintain the integrity and confidentiality of data.

2) ATTP Adaptor: The **ATTP Adaptor** bridges the agents and external systems by translating incoming payloads, extracting intent and entities and routing through the ANS.

a) Role in Workflow: The ATTP Adaptor plays a key role in processing incoming requests by extracting the routing information. It uses the **Agent Name Server (ANS)** to resolve the destination agent and route the query. And it integrates with the orchestrator to make sure the data fits the overall workflow of the AgNet system.

b) Advantages: ATTP Adaptor Benefits. It provides interoperability by enabling communication between different agents and protocols. Due to its scalable nature, it is easy to add new agents or tools to the network. It also comprises robust error mitigation, which gracefully handles both erroneous inputs and routing errors, ensuring the system operates smoothly within the AgNet environment.

The ATTP Adaptor is an important component that adds flexibility and scalability to the AgNet ecosystem.

E. Agent Security

Agent security [11] [12] is a fundamental part of the AgNet architecture to ensure the safe and reliable operation of its distributed AI agent ecosystem. In a highly dynamic and interactive environment, keeping the confidentiality, integrity and availability of communication and data is critical [7]. To address this AgNet uses **Zero Trust Architecture (ZTA)** [10] principles which is "never trust, always verify". The **Agent Private Cloud (APC)** is the mechanism to implement these principles, it provides a secure and isolated environment for all agents.

1) Agent Private Cloud (APC): The **Agent Private Cloud (APC)** is the base of AgNet's security framework. It's a secure sandbox that enforces strict security protocols and operational isolation so all agent interactions happen in a controlled and monitored environment. The APC architecture has advanced security features that prevent unauthorized access, data breaches and malicious activity so operations can run smoothly and securely.

The APC incorporates several key functionalities to achieve these security goals:

- **Gateway Agent:** The Gateway Agent is the entry point for external requests into the APC. It validates incoming requests to ensure security protocols are met, encrypts communication and routes authenticated requests to the internal agents. This ensures only authorized and legitimate requests are processed.
- **Data and Task Isolation:** To reduce risk the APC uses resource segmentation, which puts agents and their tasks in separate areas. These areas are defined by roles and permissions so unauthorized access is prevented and the scope of any breach is limited.
- **Real-Time Surveillance and Anomaly Identification:** The APC watches agent actions and task performance to detect any abnormalities or signs of security breaches. Machine learning algorithms detect anomalies like frequent login failures or unusual task behavior so threats can be managed proactively.
- **End-to-End Encryption:** All data in the APC is encrypted whether it's being sent or stored. This ensures confidentiality and integrity of sensitive information like user details, task specs and operational details, blocks unauthorized access and maintains data integrity across the system.

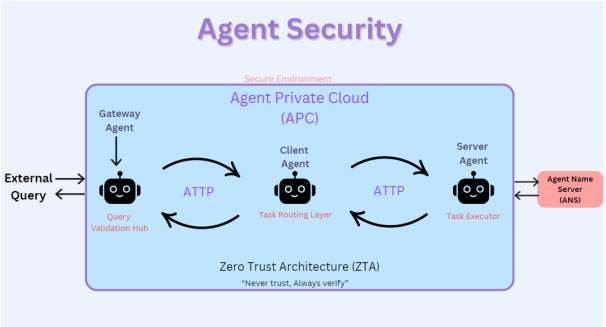

Fig. 5. Agent Security Framework

The APC follows the principles of Zero Trust Architecture (ZTA). Continuous authentication, minimal privilege access and flexible security is applied throughout the environment. By limiting agents to only the resources they need for their job the APC reduces the attack surface so even if an agent is compromised it's not a big threat to the system.

Furthermore, AgNet's overall security system which includes ANS, Agent Registry and ATTP has the APC. These components together form a layered security system. By ensuring that every action in the APC is verified and controlled, this integration supports AgNet's goal of being robust, scalable, and trusted across the distributed agent network.

IV. Results

We have validated AgNet in both simulations and real-world applications to demonstrate its relevance, safety, and operational efficiency. Key findings from the assessment are detailed below:

A. Applicability Assessment

The AgNet system was successfully evaluated using a network of four online commerce agents. Every agent has registered and found each other successfully communicates using natural language via the ATTP format and executes tasks like perform product searches, respond to customer order status inquiries, and generate catalogs for sellers. In addition, ANS reduced the average agent lookup latency by 65% due to its caching and fallback systems.

B. Security Performance

The Agent Private Cloud [APC] proved to sufficiently contain bad agents and did not allow access to key network components. In addition, the end-to-end encryption added in the ATTP protocol did not result in data breach during simulated Man-In-The-Middle (MITM) attack scenarios, showing that security was well and truly up to scratch.

C. Operational Efficiency

Agent Registry facilitated task allocation between these lines with a success rate of 92% by mapping agent capabilities to task requirements, astronomically improving task efficiency. ATTP's modular approach minimized the protocol overhead by 25%, leading to faster transmission than traditional agent communication protocols. However, the AgNet architecture can effectively optimize communication and task processing, as shown by the efficiency gain.

V. Conclusion

This paper introduced AgNet, an AI agent network architecture for large-scale AI agent environments. AgNet addresses the problems of scalability, security and communication by introducing the Agent Registry, Agent Name Server (ANS), Agent Text Transfer Protocol (ATTP), and Agent Private Cloud (APC).

These findings show that AgNet can control networks consisting of a few agents with good performance in terms of security and system performance. The architectural scalability makes it suitable for organizations of all sizes, and the APC's embedded security measures provide trust and resilience against the ever-evolving cyber threat landscape.

In the future, we will focus on testing the network with millions of agents, extend the ATTP protocol (for cross-network interoperability) and add advanced AI models to enable dynamic capability matching and anomaly detection. AgNet represents a significant step towards grouping the potential of agentic AI to optimize efficiency and automate organizations.

Acknowledgment

We acknowledge the amazing effort of the extended research team at Plotch.ai who tirelessly worked on various aspects of the Agentic research. These extended researchers included Monica Gupta, Neha Vats, Tarun Nagpal, Tarun Choudhary, Shrey Pant, Vijay Balaram and Karthikeyan M.

References

[1] T. Masterman, S. Besen, M. Sawtell, and A. Chao, *The Landscape of Emerging AI Agent Architectures for Reasoning, Planning, and Tool Calling: A Survey*, arXiv preprint arXiv:2404.11584v1, 2024.

[2] J. S. Rosenschein and G. Zlotkin, *Infrastructure for Discovery in Multi-agent Systems*, Technical Report CMU-CS-99-123, Carnegie Mellon University, 1999.

[3] O. Shehory and S. Kraus, *Methods for Task Allocation via Agent Coalition Formation*, Artificial Intelligence, vol. 101, no. 1-2, pp. 165-200, 1998.

[4] F. Bellifemine, A. Poggi, and G. Rimassa, *Developing Multi-agent Systems with a FIPA-compliant Agent Framework*, in Software Engineering for Large-Scale Multi-Agent Systems, P. Ciancarini and M. Wooldridge, Eds., Springer, 2001, pp. 203-217.

[5] E. Johnson, *Agent Communication Protocols*, Smythos Artificial Intelligence, Mar. 2024.

[6] T. Abuelsaad, D. Akkil, P. Dey, A. Jagmohan, A. Vempaty, and R. Kokku, *Agent-E: From Autonomous Web Navigation to Foundational Design Principles in Agentic Systems*, arXiv preprint arXiv:2407.13032, 2024.

[7] R. S. Hallyburton, D. Hunt, S. Luo, and M. Pajic, *A Multi-Agent Security Testbed for the Analysis of Attacks and Defenses in Collaborative Sensor Fusion*, arXiv preprint arXiv:2401.09387, 2024.

[8] G. Brown, *Introduction to LMOS*, Eclipse Foundation, May 2024.

[9] H. Lee and I. Kim, *Multi-agent Systems: A Survey about its Components, Framework, and Workflow*, International Journal of Advanced Computer Science and Applications, vol. 15, no. 3, pp. 45-58, 2024.

[10] D. Smith, *Understanding AI Risks and How to Secure Using Zero Trust*, LevelBlue Security Essentials Blog, Apr. 2024.

[11] P. Novák, M. Rollo, J. Hodík, and T. Vlček, *Communication Security in Multi-agent Systems*, in Multi-Agent Systems and Applications III, V. Mařík, J. Müller, and M. Pechouček, Eds., Springer, 2003, pp. 454-463.

[12] A. Baudet, O. E. Aktouf, A. Mercier, and P. Elbaz-Vincent, *Systematic Mapping Study of Security in Multi-Embedded-Agent Systems*, IEEE Access, vol. 9, pp. 160-175, 2021.