# ClawNet: Human-Symbiotic Agent Network for Cross-User Autonomous Cooperation

**Zhiqin Yang** [*1,2] **Zhenyuan Zhang** [*1,2] **Xianzhang Jia** [*1,2] **Jun Song** [*3] **Wei Xue** [*1,2]
**Yonggang Zhang** [1,2] **Yike Guo** [1,2]

## Abstract

Current AI agent frameworks have made remarkable progress in automating individual tasks, and multi-agent systems have further demonstrated the potential of multiple agents collaborating toward shared objectives. Yet these systems invariably serve a single user. Human productivity rests not on individual skills or experience alone, but on the social and organizational relationships through which people coordinate, negotiate, and delegate. When agents move beyond performing tasks for one person to representing that person in collaboration with others, the infrastructure for cross-user agent collaboration is entirely absent, let alone the governance mechanisms needed to secure it, including stable owner binding, scoped authorization, and auditable accountability. We argue that the next frontier for AI agents lies not in stronger individual capability, but in the digitization of human collaborative relationships themselves. To this end, we propose a **human-symbiotic agent paradigm**. In this paradigm, each user owns a permanently bound agent system that participates in collaboration on the owner's behalf, collectively forming a collaboration network whose nodes are humans rather than agents. This paradigm rests on three governance primitives. A *layered identity architecture* separates a Manager Agent from multiple context-specific Identity Agents. The Manager Agent holds global knowledge but is architecturally isolated from external communication, serving both as a privacy safeguard and as an internal advisor to Identity Agents. *Scoped authorization* enforces per-identity access control and automatically escalates boundary violations to the owner. *Action-level accountability* logs every operation against its owner's identity and the corresponding authorization, ensuring full auditability. We instantiate this paradigm in **ClawNet**, an *identity-governed agent collaboration framework* built on the OpenClaw framework. ClawNet enforces identity binding and authorization verification through a central orchestrator during cross-user collaboration, preventing unauthorized access and information leakage, enabling multiple users to collaborate securely and efficiently through their respective agents.

[1]Hong Kong Generative AI Research & Development Center [2]Hong Kong University of Science and Technology [3]Hong Kong Baptist University. Correspondence to: Yonggang Zhang <zhangyg@ust.hk>, Yike Guo <yikeguo@ust.hk>.

*Second Workshop on Technical AI Governance Research (TAIGR) @ ICML 2026*, Seoul, South Korea. 2026. Copyright 2026 by the author(s).

## 1. Introduction

AI agent technology is advancing rapidly from conversational generation to autonomous execution (Wang et al., 2024; Guo et al., 2024; Qin et al., 2024; Yao et al., 2022). Early conversational systems were limited to text generation and could not interact with external environments (Achiam et al., 2023; Wang et al., 2024). Tool-augmented agents then introduced API-level interaction through function calling and tool-use mechanisms (Schick et al., 2023; Qin et al., 2024). More recently, OS-level agents such as OpenClaw (OpenClaw Community, 2026) and Anthropic's Computer Use (Anthropic, 2024) have achieved direct control over files, terminals, and desktop applications, advancing single-agent capability from advice to action (Wang et al., 2024). In parallel, multi-agent systems such as MetaGPT (Hong et al., 2023), AutoGen (Wu et al., 2024), CrewAI (cre, 2024), LangGraph (lan, 2024), and ChatDev (Qian et al., 2024) have demonstrated that multiple agents can collaborate effectively toward shared objectives (Guo et al., 2024).

Yet whether single-agent or multi-agent, these systems invariably serve a single user (Figure 1, Left). Agents within existing multi-agent frameworks share the same principal, operate under a unified goal, and do not need to represent distinct human interests (Guo et al., 2024; Wang et al., 2024). As summarized in Table 1, none of the existing frameworks

provide mechanisms for agents of different users to interact on behalf of their respective owners. This stands in sharp contrast to how humans actually work, where productivity rests not on individual skills alone but on the social and organizational relationships through which people coordinate, negotiate, and delegate (Malone et al., 2004; Coase, 1993). In any real-world collaboration, three structural elements are indispensable (Shavit et al., 2023). *Identity* establishes who a participant is and whom they represent. *Authorization* defines what actions are permitted within a given context and what must be escalated. *Accountability* ensures that every action can be traced back to the individual who acted and the mandate under which they acted. These elements are so deeply embedded in human collaboration that they are often taken for granted, yet none of them exists in current agent systems.

Existing agent systems have successfully digitized individual skills (Wang et al., 2024; Qin et al., 2024), but none of these three elements has been addressed. Emerging interoperability protocols such as Google's Agent2Agent (Surapaneni et al., 2025) provide a communication layer through which agents can discover and exchange messages across framework boundaries, but they do not bind agents to specific owners, enforce authorization scopes, or produce auditable responsibility chains. There is neither a means through which different users' agents can identify and interact with one another (Chan et al., 2025), nor a governance mechanism to safeguard each party's interests during collaboration (de Witt, 2025). Agents lack stable owner binding and therefore cannot verifiably represent a specific individual when initiating or participating in cross-user workflows (South et al., 2025; Shavit et al., 2023). Even if collaboration were to occur, no scoped authorization or auditable accountability exists to prevent unauthorized access, information leakage, or decisions made without mandate (Chan et al., 2025; Casper et al., 2025). Existing safeguards such as sandboxing, per-call permission prompts, and capability restrictions address single-user misuse but leave the structural risks of cross-user collaboration entirely unmitigated (Shavit et al., 2023; de Witt, 2025).

To address this gap, we move beyond strengthening individual agent capability and focus instead on digitizing human collaborative relationships, equipping agents with the identity, authorization, and accountability infrastructure that **makes cross-user collaboration via autonomous agents possible**. To this end, we propose a **human-symbiotic agent paradigm**. In this paradigm, each user owns an agent system that is permanently bound to its owner (Chan et al., 2025; South et al., 2025), much as a social-media account is inseparably tied to the person who created it. The system comprises multiple identity agents, each corresponding to a specific collaborative context, collectively covering the user's collaboration needs across different social and professional roles.

The manager agent serves as the authoritative anchor of the user's digital identity. Because its breadth of knowledge makes direct external exposure a privacy risk, the manager agent is architecturally isolated from all cross-user communication, serving exclusively as an internal governor and advisor to identity agents. Each identity agent is a context-specific projection of the owner, created and retired as the owner's social and professional roles evolve. For example, a user who takes on a new managerial position may instantiate a corresponding identity agent, designate which collaborators are authorized to interact with that identity, and dissolve it when the role ends. No single identity agent possesses the full extent of its owner's knowledge; rather, it operates strictly under its designated identity, inheriting only the information relevant to that identity. It persists across sessions, progressively accumulates an understanding of its owner's preferences and the collaborative dynamics of its context, and acts autonomously within its authorization boundary. Through this ensemble, users participate in cross-user collaboration via their identity agents. As shown in Figure 1, the resulting network is fundamentally different from existing multi-agent topologies: its nodes are humans, not agents, and each edge represents a governed collaborative relationship rather than an inter-process message channel.

We instantiate this paradigm in **ClawNet**, an open-source identity-governed agent collaboration framework. ClawNet adopts a cloud-edge architecture: the manager agent and all identity agents run as persistent cloud services for always-on availability and cross-user orchestration, while a lightweight edge client installed on the owner's device serves as both the user's interaction interface and a local execution engine with OS-level control capabilities. The edge client further enforces file-level access control, granting different identity agents access to different subsets of local files according to their respective roles. Notably, the cloud and edge components can be co-located on a single physical machine when fully local deployment is preferred. ClawNet has been deployed and tested across cross-organizational collaboration scenarios, demonstrating that the governance primitives operate effectively in practice.

In summary, this paper makes the following contributions:

- We propose a **human-symbiotic agent paradigm** that introduces three governance primitives (identity binding, scoped authorization, and action-level accountability) for cross-user agent collaboration, along with a layered identity architecture separating a privacy-preserving manager agent from multiple context-specific identity agents.
- We design and implement **ClawNet**, an open-source identity-governed agent collaboration framework featuring a cloud-edge architecture, dual-layer independent file authorization, and a comprehensive operational audit trail

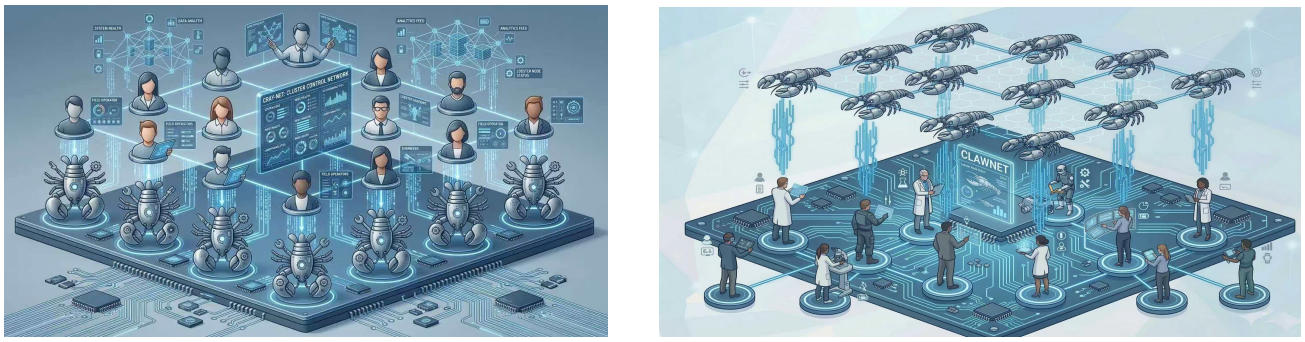



*Figure 1.* **Paradigm comparison between OpenClaw and ClawNet. (Left)** In current frameworks, agents reside beneath their users as isolated executors with broad but undifferentiated resource access, no persistent identity, and no cross-user communication protocol. All inter-user coordination falls to the humans themselves. **(Right)** In ClawNet, agents form a governed collaboration layer above their owners. Each agent is permanently bound to one owner (identity binding), operates under explicit, revocable permissions (scoped authorization), and logs every action against its owner's identity and mandate (action-level accountability). Humans retain intent formulation and critical decisions; all cross-user coordination is realized through structured inter-agent negotiation within this accountable layer.

that ensures every agent action remains traceable and correctable by the owner.

- We validate ClawNet through representative cross-organizational collaboration scenarios, demonstrating that governance-aware agents can mediate multi-party negotiations while maintaining identity isolation, authorization enforcement, and full auditability. Through this instantiation, we advocate for a broader shift toward human-symbiotic governance in the design of collaborative agent systems.

## 2. System Overview

This section first formalizes the human-symbiotic paradigm as a collaboration network with humans as nodes and governed agent interactions as edges (§2.1), then discusses the rationale behind the cloud-edge architectural separation (§2.2), and finally presents the cloud-side components (§2.3) and the edge-side node endpoint (§2.4) that jointly support this network.

### 2.1. Human-Symbiotic Collaboration Network

ClawNet can be understood to a social network with AI assistant. In a conventional social network such as WeChat, people establish connections and exchange messages directly. In ClawNet, each user additionally owns a permanently bound agent system, and collaboration between users is mediated and executed through their respective agents. This gives rise to a collaboration network whose nodes are humans and whose edges are governed agent interactions.

**Collaboration network.** Let $\mathcal{U}$ be the set of users. ClawNet constructs a collaboration network $\mathcal{G} = (\mathcal{U}, \mathcal{E})$, where each node is a human user $u \in \mathcal{U}$ and each edge $e(u, v) \in \mathcal{E}$ represents a governed collaborative relationship established through the agents of $u$ and $v$. Unlike existing multi-agent topologies, the nodes of this network are always humans, not agents.

**Agent system.** Each user $u$ owns a permanently bound agent system $\mathcal{A}_u$ comprising a manager agent $M_u$ and a set of identity agents $\{I_u^1, I_u^2, \ldots, I_u^k\}$. $M_u$ governs the creation, configuration, and retirement of all identity agents, holds complete knowledge $\mathcal{K}_u$ of its owner, but is architecturally isolated from all external communication, accessible only by agents within $\mathcal{A}_u$. By analogy, $M_u$ is the user's "primary account" that is never exposed to the outside world. The design rationale and detailed mechanisms of this layered architecture are presented in Section 3.

**Identity agent.** Each identity agent $I_u^i$ corresponds to a specific collaborative context, analogous to an "identity tag" in a social network. Formally,

$$I_u^i = (c_i,\ \sigma_i,\ \mathcal{K}_i,\ \mathcal{P}_i)$$

where $c_i$ is the collaborative context (e.g., "procurement manager"), $\sigma_i \subseteq \mathcal{R}_u$ is the scoped authorization boundary defining the accessible resource subset, $\mathcal{K}_i \subset \mathcal{K}_u$ is the context-relevant knowledge inherited through the memory system and progressively accumulated across sessions, and $\mathcal{P}_i \subseteq \mathcal{U} \setminus \{u\}$ is the set of users authorized to interact with this identity. Only collaborators in $\mathcal{P}_i$ can discover and interact with $I_u^i$, much as only designated contacts can see a specific tag in a social profile.

**Discovery and connection.** When user $u$ wishes to collaborate with user $v$, the identity agent $I_u^i$ locates the corresponding identity agent $I_v^j$ through the identity tags published by $v$ in the collaboration network. A connection requires explicit authorization from both human owners:

$$S(I_u^i, I_v^j) \text{ holds iff } \text{approve}(u, I_u^i) \wedge \text{approve}(v, I_v^j) \wedge\ u \in \mathcal{P}_j \wedge v \in \mathcal{P}_i \tag{1}$$

The detailed collaboration lifecycle, including contact es-

*Table 1.* Paradigm comparison across representative agent frameworks. Existing multi-agent systems coordinate agents under a single principal; ClawNet binds each agent to its own owner and governs cross-user collaboration through identity binding, scoped authorization, and action-level accountability. ✓ = fully supported; △ = partially supported; × = not supported.

| **Paradigm** | **Representative** | **Exec. Level** | **Identity Binding** | **Scoped Auth.** | **Action Account.** | **Cross-Owner Collab.** |
|---|---|---|---|---|---|---|
| Conversational AI | ChatGPT, Claude Chat | Text gen. | × | × | × | × |
| Tool-Augmented AI | ChatGPT Plugins, MCP | API inter. | × | △ | × | × |
| OS-Level Single Agent | OpenClaw, Computer Use | OS-level | △ | △ | × | × |
| Single-Principal Multi-Agent | MetaGPT, AutoGen, CrewAI, LangGraph, ChatDev | API / OS | × | × | × | × |
| Cross-Agent Protocol | Google A2A | API-level | × | × | × | △ |
| **Human–Symbiotic (Ours)** | **ClawNet** | **OS-level** | ✓ | ✓ | ✓ | ✓ |

tablishment, bilateral approval, multi-turn dialogue, and manager consultation, is described in Section 3.3.

**Recursive collaboration.** When $I_v^j$ discovers during a collaboration that a third party $w$ is needed, $v$ may authorize $I_v^{j'}$ to initiate a new collaboration request to $w$, forming a chain:

$$S(I_u^i, I_v^j) \rightarrow S(I_v^{j'}, I_w^m) \rightarrow \cdots \tag{2}$$

Each level of recursion requires explicit authorization from the user at that level, and the recursion depth is bounded by a system parameter $d_{\max}$. Crucially, upper-level collaborators cannot penetrate lower-level authorization boundaries: $I_u^i$ cannot directly access the resources of $I_w^m$ and can only receive collaboration results within $v$'s authorized scope.

**Governance primitives.** All interactions in the collaboration network are subject to three governance constraints. For any operation $o$ performed by identity agent $I_u^i$: (1) *Identity binding*: $\text{owner}(o) = u$, $\text{identity}(o) = I_u^i$. Every operation is traceable to a specific human and a specific identity. (2) *Scoped authorization*: $\text{target}(o) \in \sigma_i$. Operations must fall within the agent's authorization boundary; violations are intercepted and escalated to user $u$. (3) *Action-level accountability*: every operation is logged as $\ell = (o, u, I_u^i, \text{result}, t)$ into an append-only audit log $\mathcal{L}$. The concrete enforcement of these primitives at the identity architecture level and during cross-user collaboration is detailed in Section 3.

### 2.2. Design Rationale

The collaboration network described above requires two complementary infrastructure capabilities. Cross-user collaboration demands that agents remain discoverable and responsive at all times, necessitating that the reasoning engine and identity state persist in the cloud. At the same time, as an OS-level agent, it must also operate within the owner's local file system, including reading documents, organizing directories, and managing assets, which requires execution capability on the owner's device. ClawNet reconciles these two requirements through the *node endpoint* mechanism: the user's client registers as a remote execution node with the cloud, receives operational directives, and executes them on the local file system, forming a closed-loop system that bridges cloud-side cognition with edge-side execution.

### 2.3. Cloud-Side Overview

As illustrated in Figure 3 (a), the cloud side comprises two primary components, including gateway containers and a central orchestration engine.

**Gateway containers.** To ensure strict multi-tenancy isolation, each user is provisioned with a dedicated container running an agent execution engine with OS-level control capabilities. The agent runtime within the container employs a think/act loop to drive a large language model (LLM)-based reasoning, translating high-level natural language intent into discrete, actionable plans. Each container maintains an independent process space, network stack, and persistent storage, ensuring that the reasoning traces, memory states, and intermediate data of different users remains mutually inaccessible. $M_u$ and $\{I_u^1, \ldots, I_u^k\}$ reside within the same container to minimize communication overhead, as $M_u$ serves as a real-time internal advisor to identity agents. Despite co-residence, they operate in logically partitioned workspace subspaces corresponding to the authorization boundaries $\sigma_i$, preventing cross-domain data leakage (detailed in §3).

**Server (central orchestration engine).** The server acts as the centralized backbone, responsible for identity-based message routing, cross-user collaboration orchestration (§3), and the first-layer access control list (L1 ACL) for file operations. It maintains persistent WebSocket connections to each user's gateway container. When an agent initiates a file operation, the server functions as a proxy node, a secure mediator between the cloud-side gateway and the edge-side client, verifying that $\text{target}(o) \in \sigma_i$ before forwarding direc-

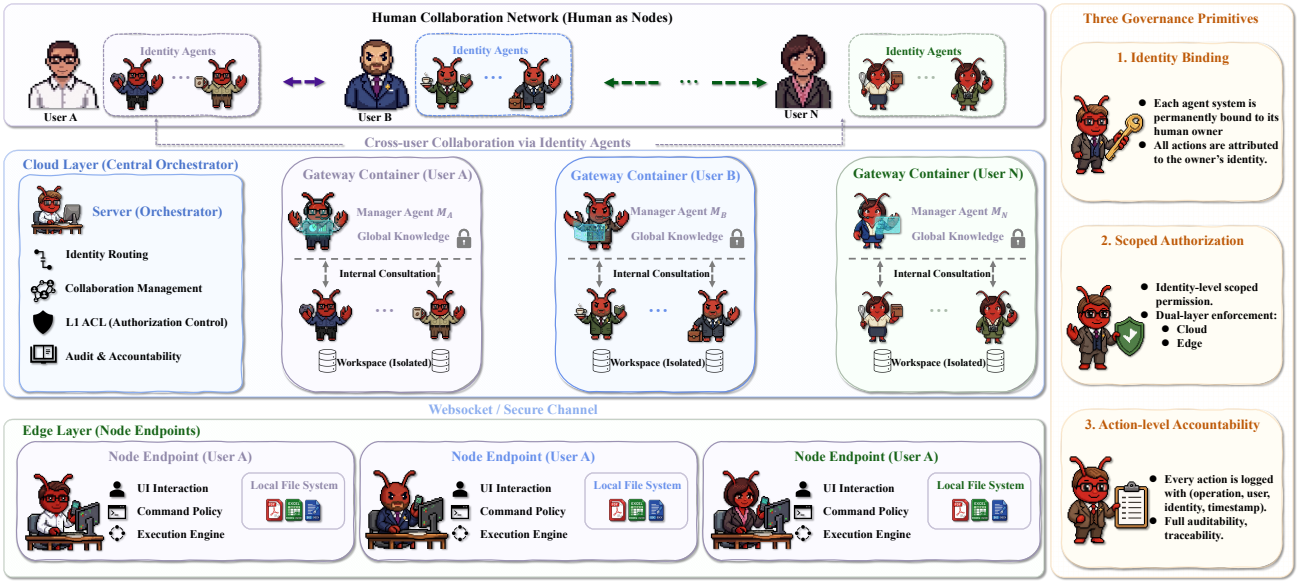



*Figure 2.* **Overview of ClawNet. (Top)** The human collaboration network, where each user owns a set of identity agents and cross-user collaboration is mediated exclusively through these agents. Humans are nodes; agents mediate the edges. **(Middle)** The cloud layer, comprising a central orchestrator (server) responsible for identity routing, collaboration management, and first-layer authorization (L1 ACL), and per-user gateway containers that host the manager agent $M_u$ and identity agents within logically isolated workspaces. The manager agent provides internal consultation to identity agents but is architecturally isolated from all external communication. **(Bottom)** The edge layer, where node endpoints installed on each owner's device serve as both the user interaction portal and the local execution engine, enforcing second-layer command policy (L2) on the owner's file system. **(Right)** The three governance primitives enforced throughout the system: identity binding ensures every action is attributed to a specific owner; scoped authorization enforces dual-layer (cloud and edge) permission control per identity; and action-level accountability logs every operation with full auditability and traceability.

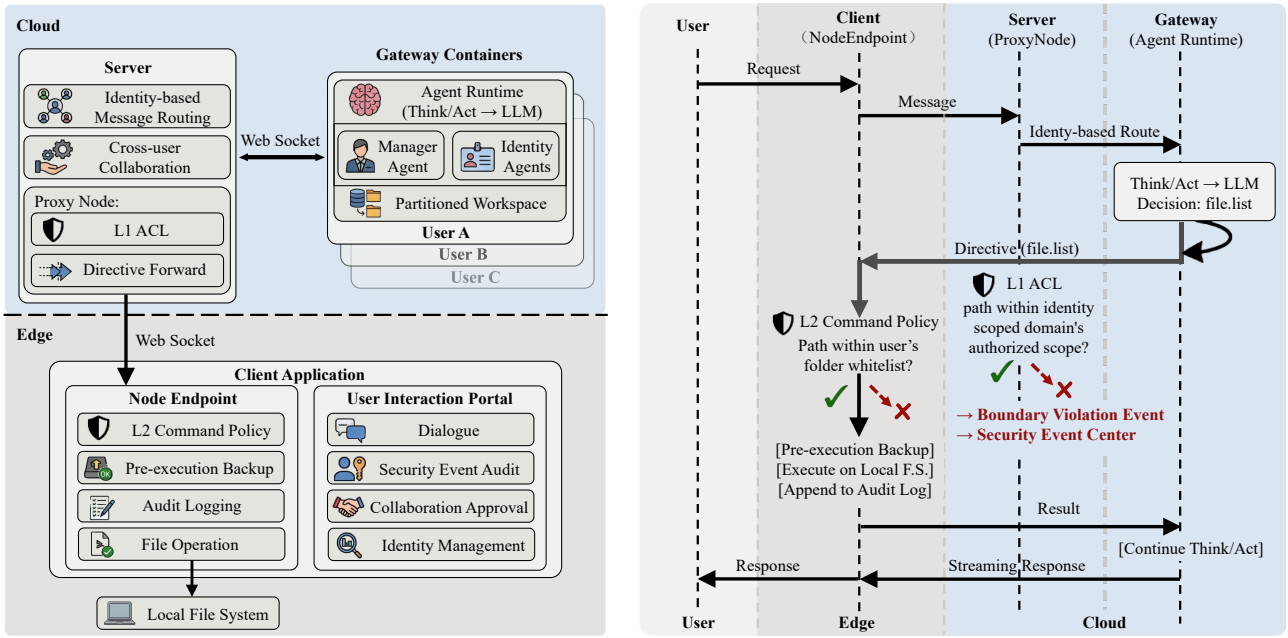
tives to the user's device.

### 2.4. Node Endpoint Overview

The edge side is embodied by the client application installed on the owner's device, serving two roles. First, it acts as a user interaction portal, providing the graphical interface for dialogue, identity management, collaboration approval, and security event auditing. Second, it acts as a *node endpoint*, the terminal execution environment that receives operational directives from the cloud and performs physical input/output operations on the local file system. Upon initialization, the client registers with the server as a node, advertising its supported primitive command set. Figure 3 (b) illustrates the end-to-end data flow of an operational directive.

**File operation capabilities.** The node endpoint equips agents with a comprehensive suite of OS-level file primitives, encompassing reading, writing, directory traversal, relocation, renaming, replication, directory creation, and secure deletion. Operating within the think/act loop, agents autonomously orchestrate complex operation sequences to fulfill high-level user intent. The architecture maintains location transparency for the reasoning engine: the agent remains agnostic to the physical execution environment, merely issuing high-level directives to the node for local fulfillment. These capabilities are categorized into `read-only operations` (e.g., content retrieval, directory listing, and metadata queries) and `mutative operations` (e.g., writing, moving, and deletion). To ensure system integrity, every mutative action triggers an automated pre-execution backup, ensuring that any modification remains fully traceable and reversible.

**File access control.** Extending OS-level file capabilities to agents inherently exposes the owner's most sensitive data to the agentic domain. ClawNet addresses this risk through a dual-layer independent authorization scheme, enforcing $\text{target}(o) \in \sigma_i$ along the entire directive path from cloud to edge. At the first layer (server-side), the server verifies the operation against the identity domain configuration of $I_u^i$. At the second layer (client-side), the client independently validates the operation against an explicit folder whitelist defined by the owner. The two layers function autonomously with a fail-closed design: a rejection by either layer results in a full-chain denial. The integration of this scheme with the layered identity architecture, including how different identity agents operate within disparate access scopes and how boundary violations are escalated, is detailed in Section 3. **Operational auditing and reversibility.** The node endpoint logs every mutative operation as $\ell = (o, u, I_u^i, \text{result}, t)$ into an append-only local audit log, with automatic pre-execution backup of all affected files.




(a) High-level Architecture of ClawNet. (b) End-to-end Dataflow for a File Operation Request.

*Figure 3.* (a) High-level architecture of ClawNet. The cloud side encompasses the server (central orchestration engine) and per-user gateway containers (agent runtimes). The edge side consists of the client application, which functions as both a user interaction portal and a node endpoint for local file system operations. (b) End-to-end data flow for a file operation request. Governance mechanisms, comprising identity binding, two-layer scoped authorization (server-side L1 ACL and client-side L2 command policy), and accountability (pre-execution backup and audit logging), are integrated directly into the directive transmission path.

The owner may invoke single-step undo or batch rollback. File deletion is implemented as relocation to a recoverable staging area, and file overwrites automatically preserve the original content. These mechanisms secure the owner's sovereign right of correction over the agent's autonomous behavior.

## 3. Identity Governance and Agent Collaboration

To credibly collaborate on behalf of its owner, an agent must possess a stable and persistent identity. This section builds on the formal definitions in Section 2 to present the concrete mechanisms of identity-centric governance: how agent identity is established through memory-driven cognitive coupling (§3.1), the layered architecture of the manager agent and identity agents along with their governance mechanisms (§3.2), and the cross-user collaboration protocol built upon this identity governance framework (§3.3).

### 3.1. Identity Binding: From Task Memory to Cognitive Coupling

Contemporary agentic systems have begun to incorporate long-term memory, but such capabilities typically focus on operational continuity, preserving user preferences to enhance future task efficiency. ClawNet redefines the role of memory: it serves not merely as a tool for task optimization, but as the foundational substrate for an agent to truly *know* its owner.

Specifically, ClawNet's memory system achieves cognitive coupling through three progressive layers.

The first layer is factual memory. Through daily interactions, users delegate routine tasks via a natural language interface, including file organization, document archiving, information retrieval, and directory management. By executing these operations on the user's local file system via the node endpoint (§2.4), the agent accumulates basic facts about its owner: locations of frequently used files, project directory structures, contact information, and schedules. These facts constitute the raw material of identity cognition.

The second layer is pattern memory. As interactions accumulate longitudinally, the agent extracts behavioral patterns from discrete facts: workflow habits (e.g., organizing emails every Monday, archiving financial files at month-end), file organization preferences (e.g., categorizing by project rather than by date), and context-dependent communication styles (e.g., concise and direct with colleagues, polite and formal with clients). Pattern memory enables the agent to anticipate the owner's intent rather than passively awaiting instructions.

The third layer is value memory. The agent further internal-

izes the owner's decision boundaries and value dispositions: which information must never be shared, what the bottom line is in negotiations, and whether the owner tends to compromise or hold firm when facing conflicts. This value-level understanding enables the agent, when acting as a proxy in cross-user collaboration, to know not only "what the owner would do" but also "what the owner would never do."

ClawNet persists all three layers of cognition across sessions. The agent does not experience amnesia at the conclusion of a session but incrementally deepens its alignment with the owner through each successive engagement. Consequently, the relationship between the agent and its owner transcends technical isolation (dedicated containers and segregated workspaces) to achieve cognitive coupling: the agent forms a unique, irreplaceable personalized model of its owner. This cognitive coupling is the critical foundation for the agent's evolution from a generic utility into a sovereign representative.

### 3.2. Layered Identity Architecture

Once identity binding is established, a critical design question arises: should a user be represented by a single monolithic agent or a distributed set of specialized agents?

A monolithic agent suffers from an inherent tension: deep owner-alignment is essential for effective proxying, but it significantly elevates the risk of inadvertent privacy disclosure during external interactions. Conversely, constraining an agent's knowledge for security purposes inevitably compromises its capacity for cross-domain reasoning. ClawNet reconciles this conflict through a layered identity architecture, partitioning the user's agent system $\mathcal{A}_u = (M_u, \{I_u^1, \ldots, I_u^k\})$ into two distinct tiers.

**Manager agent.** $M_u$ serves as the authoritative anchor of the user's digital identity. It has the privilege to access the memory contents of all identity agents $\{I_u^1, \ldots, I_u^k\}$, enabling it to form a cross-domain global perspective. Crucially, $M_u$ does not maintain a separate global knowledge base; rather, it derives its global understanding through aggregated access to all identity agents' memories.

This global access capability makes $M_u$'s direct participation in external interactions inherently risky. Consider a scenario where $M_u$ can simultaneously access confidential commercial contract information stored in the "work" identity agent and private health records stored in the "personal" identity agent. If $M_u$ is queried with "how have you been lately?" during a collaborative session with a colleague, it might inadvertently leak health information in its response. Prompt-level security constraints such as "do not share private information" rely on the model perfectly adhering to instructions across every inference step, a requirement that is statistically impossible to guarantee.

ClawNet therefore enforces architectural-level isolation on $M_u$: the collaboration protocol is structurally incapable of routing external messages to $M_u$. The security paradigm shifts from a policy-based "should not" (relying on model behavioral reliability) to an architectural "cannot" (leveraging system-level determinism).

Under this isolation constraint, $M_u$ assumes the role of an internal consultant. When $I_u^i$ requires cross-domain knowledge support during a collaboration, it may solicit guidance from $M_u$. $M_u$ provides advice based on its aggregated access to all identity agents' memories, subject to two constraints: $M_u$'s response is never sent directly to external collaborators but is instead presented to $I_u^i$ and the owner, who decide whether and how to integrate it; and $M_u$ is guided by $I_u^i$'s current context $c_i$ when formulating advice, avoiding unsolicited exposure of information from unrelated domains.

**Identity agents.** Each $I_u^i = (c_i, \sigma_i, \mathcal{K}_i, \mathcal{P}_i)$ is a contextual projection of the owner within a specific collaborative environment. Individuals fulfill diverse social roles across different settings: a project manager in a professional context, a parent in family life, a researcher in academia. Each identity agent represents a digital instantiation of one such role, bound to an identity domain marked by a tag, with an independent knowledge space $\mathcal{K}_i \subset \mathcal{K}_u$ and persistent memory.

In implementation, different identity agents operate within logically isolated workspace subspaces inside the same gateway container. Each $I_u^i$ can only access files and resources within $\sigma_i$, and $I_u^i$ cannot read the workspace contents of $I_u^j$ ($i \neq j$), even though they belong to the same owner.

The core value of identity agents is realized in cross-user collaboration. The owner explicitly selects which identity to present when engaging with another user's agent: a "work" identity agent to coordinate project milestones with a colleague, or an "academic" identity agent to discuss research methodologies with a collaborator. The selected agent carries only the knowledge $\mathcal{K}_i$ of its identity domain, ensuring that information from other domains remains entirely inaccessible to the counterpart. The prerogative to select the active identity resides exclusively with the owner.

The lifecycle of an identity agent evolves in synchrony with the owner's social roles: instantiated when a new role emerges (with $\mathcal{P}_i$ and $\sigma_i$ configured), retired when the role concludes (with authorizations and records archived), and its parameters adjustable at any time.

**Identity-level authorization and accountability.** The layered identity architecture provides clear anchors for both authorization and accountability. Each $I_u^i$'s file access scope is constrained by $\sigma_i$, enforced through the dual-layer authorization scheme described in Section 2.4. Authorization

policies evolve dynamically with the owner's real-world social roles. The architectural isolation of $M_u$ constitutes an additional authorization boundary: it holds the most comprehensive access privileges but is structurally barred from external communication.

Every operation is bound to the originating identity agent's identifier and its identity domain, logged as $\ell = (o, u, I_u^i, \text{result}, t)$. The owner can precisely trace the provenance and authorization basis of any action. Unauthorized access attempts are automatically intercepted and dispatched as boundary violation events to the owner's security event center, ensuring that every breach is audited and cannot be ignored.

### 3.3. Cross-User Collaboration

The preceding sections described identity governance within a single user domain. This section extends the scope to the inter-user dimension: the mechanisms governing collaboration between identity agents belonging to different owners.

**Establishing contact relationships and identity assignment.** The prerequisite for collaboration is a contact relationship between two users. A user sends a friend request; upon confirmation, they become contacts. The user then designates which identity agent to present to each contact: a "work" identity agent for a colleague, an "academic" identity agent for a collaborating advisor. When facing different counterparts, the same user presents distinct identity projections, each carrying its own $\mathcal{K}_i$ and $\sigma_i$. Users can dynamically adjust identity assignments or remove contacts at any time. The construction of the collaboration network is entirely in the hands of the user.

**Collaboration initiation and bilateral approval.** A collaborative interaction is triggered when $I_u^i$ manifests the intent to communicate with another user during a local conversation, for instance, determining the need to retrieve information from a colleague. The system does not autonomously initiate the collaboration. Instead, it intercepts the action and issues an authorization request to the owner. The request is dispatched to the counterpart only after securing the owner's explicit approval. The counterpart's owner must also approve. A collaboration session is established only when both parties provide explicit consent:

$$\text{approve}(u, I_u^i) \wedge \text{approve}(v, I_v^j) \wedge u \in \mathcal{P}_j \wedge v \in \mathcal{P}_i \quad (3)$$

A rejection by either party immediately terminates the initiation sequence. Without this mandatory bilateral approval, an agent might unilaterally share information without the owner's awareness, an outcome that is fundamentally unacceptable when the interacting agents represent distinct owners with independent interests.

**Multi-turn dialogue and autonomous collaboration.** Once the session is established, the identity agents of both parties enter an autonomous multi-turn dialogue. Agents exchange information, deliberate on solutions, and orchestrate tasks, each generating responses strictly based on its own $\mathcal{K}_i$. At every turn, the server injects role-specific system prompts that clearly define each agent's identity, stance, and behavioral norms, preventing the well-known problem of role drift in multi-turn LLM conversations. The server simultaneously performs identity verification and authorization checks.

Conversation termination is governed by three layers of control. At the agent layer, the agent evaluates dialogue progression through semantic state markers and gracefully concludes the conversation upon achieving its objective. At the system layer, a maximum turn threshold prevents indefinite continuation. At the owner layer, the human own

## 4. Conclusion

This paper presents ClawNet, an identity-governed framework that operationalizes a human-symbiotic paradigm for cross-user agent collaboration. By permanently binding each agent to a human owner via progressive cognitive coupling, partitioning identity into privacy-preserving and outward-facing tiers, and enforcing identity binding, scoped authorization, and action-level accountability across all interactions, ClawNet advances from the digitization of individual skills to that of human collaborative relationships. The resulting network situates a human at every node: the manager agent affords deep owner understanding without external exposure, while identity agents convey only context-relevant knowledge into governed cross-user interactions.

Several directions remain open. First, a capability discovery service could enable collaboration matching based on complementary expertise rather than pre-existing acquaintance. Second, successful collaboration records could become trust credentials that progressively unlock deeper cooperation privileges, creating a positive feedback loop between agent capability growth and human relationship expansion. Third, shifting part of the reasoning to the edge would keep privacy-sensitive analyses entirely local. Finally, larger-scale evaluation—including user studies and adversarial stress testing of governance mechanisms—would further strengthen the empirical results.

As agents grow more capable, they will inevitably need to interact with agents belonging to other people, organizations, and institutions. Analogous to the internet's reliance on identity and trust protocols as a precondition for e-commerce, autonomous agents necessitate governance primitives before cross-user collaboration becomes viable. We hope ClawNet serves as both a working prototype and an open invitation for the community to build upon this foundation.

## References


CrewAI: The leading multi-agent platform. https://github.com/crewAIInc/crewAI, 2024.

LangGraph: Build stateful multi-agent applications. https://github.com/langchain-ai/langgraph, 2024.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Anthropic. Developing a computer use model. https://www.anthropic.com/news/developing-computer-use, 2024.

Casper, S., Bailey, L., Hunter, R., Ezell, C., Cabalé, E., Gerovitch, M., Slocum, S., Wei, K., Jurkovic, N., Khan, A., et al. The ai agent index. *arXiv preprint arXiv:2502.01635*, 2025.

Chan, A., Wei, K., Huang, S., Rajkumar, N., Perrier, E., Lazar, S., Hadfield, G. K., and Anderljung, M. Infrastructure for ai agents. *Transactions on Machine Learning Research*, 2025.

Coase, R. H. The nature of the firm (1937). *The nature of the firm: origins, evolution, and development*, pp. 18–33, 1993.

de Witt, C. S. Open challenges in multi-agent security: Towards secure systems of interacting ai agents. *arXiv preprint arXiv:2505.02077*, 2025.

Finin, T., Fritzson, R., McKay, D., and McEntire, R. Kqml as an agent communication language. In *Proceedings of the third international conference on Information and knowledge management*, pp. 456–463, 1994.

Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N., Wiest, O., and Zhang, X. Large language model based multi-agents: A survey of progress and challenges. In *33rd International Joint Conference on Artificial Intelligence (IJCAI 2024)*. IJCAI; Cornell arxiv, 2024.

Hong, S., Zhuge, M., Chen, J., Zheng, X., Cheng, Y., Wang, J., Zhang, C., Wang, Z., Yau, S. K. S., Lin, Z., et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *The twelfth international conference on learning representations*, 2023.

Li, G., Hammoud, H. A. A. K., Itani, H., Khizbullin, D., and Ghanem, B. Camel: Communicative agents for "mind" exploration of large language model society. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Malone, T. W. et al. *The future of work*. Audio-Tech Business Book Summaries, Incorporated, 2004.

OpenClaw Community. OpenClaw. https://github.com/openclaw/openclaw, 2026.

Pitt, J. and Mamdani, A. Some remarks on the semantics of fipa's agent communication language. *Autonomous Agents and Multi-Agent Systems*, 2(4):333–356, 1999.

Qian, C., Liu, W., Liu, H., Chen, N., Dang, Y., Li, J., Yang, C., Chen, W., Su, Y., Cong, X., et al. Chatdev: Communicative agents for software development. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 15174–15186, 2024.

Qin, Y., Hu, S., Lin, Y., Chen, W., Ding, N., Cui, G., Zeng, Z., Zhou, X., Huang, Y., Xiao, C., et al. Tool learning with foundation models. *ACM Computing Surveys*, 57(4): 1–40, 2024.

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., and Scialom, T. Toolformer: Language models can teach themselves to use tools. volume 36, pp. 68539–68551, 2023.

Shavit, Y., Agarwal, S., Brundage, M., Adler, S., O'Keefe, C., Campbell, R., Lee, T., Mishkin, P., Eloundou, T., Hickey, A., et al. Practices for governing agentic ai systems, 2023.

South, T., Marro, S., Hardjono, T., Mahari, R., Whitney, C. D., Greenwood, D., Chan, A., and Pentland, A. Authenticated delegation and authorized ai agents. 2025.

Surapaneni, R., Jha, M., Vakoc, M., and Segal, T. Announcing the agent2agent protocol (a2a). Google Developers Blog, 2025. https://a2a-protocol.org/latest/.

Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024.

Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., et al. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First conference on language modeling*, 2024.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. R., and Cao, Y. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022.

## A. Related Work

**Single-agent and multi-agent frameworks.** Single-agent systems such as OpenClaw (OpenClaw Community, 2026) and Anthropic's Computer Use (Anthropic, 2024) have advanced agent capability from tool augmentation to OS-level control, but address only what one agent can do for one user. Multi-agent frameworks such as MetaGPT (Hong et al., 2023), AutoGen (Wu et al., 2024), CrewAI (cre, 2024), ChatDev (Qian et al., 2024), and CAMEL (Li et al., 2023) demonstrate effective collaboration among multiple agents through role-playing and structured workflows. However, all operate within a single principal's scope: agents share the same user, the same goal, and the same trust domain. When MetaGPT assigns agents the roles of "Product Manager" and "Engineer," these are functional specializations, not representatives of different humans with divergent interests. ClawNet addresses a fundamentally different setting: each agent is permanently bound to a distinct human owner, and cross-user cooperation takes the form of governed inter-agent negotiation.

**Agent safety and governance.** A growing body of work recognizes the governance gap in agentic systems. (Shavit et al., 2023) outline practices for governing agentic AI but do not provide a system implementation. (Chan et al., 2025) identify the need for infrastructure supporting agent identity and interaction, and (South et al., 2025) propose authenticated delegation as a prerequisite for authorized AI agents. (Casper et al., 2025) catalog the landscape of AI agents and their accountability gaps, while (de Witt, 2025) discuss the structural challenges of multi-agent security. These works collectively establish the theoretical motivation for governance-aware agent systems but stop short of an end-to-end implementation. ClawNet translates these identified requirements into a deployed system with concrete enforcement mechanisms.

**Agent interoperability protocols.** Standardized agent communication has a long history, from KQML (Finin et al., 1994) to FIPA Agent Communication Language (FIPA-ACL) (Pitt & Mamdani, 1999). More recently, Google's Agent2Agent protocol (Surapaneni et al., 2025) provides a communication layer for agents to discover and exchange messages across framework boundaries, and Anthropic's Model Context Protocol enables standardized tool integration. These protocols address *how* agents communicate but not *under what governance*: they do not bind agents to specific owners, enforce authorization scopes, or produce auditable responsibility chains. ClawNet can be viewed as a governance layer that operates on top of such communication protocols, adding identity binding, scoped authorization, and action-level accountability to the interaction substrate.

## B. System Interface and Governance in Practice

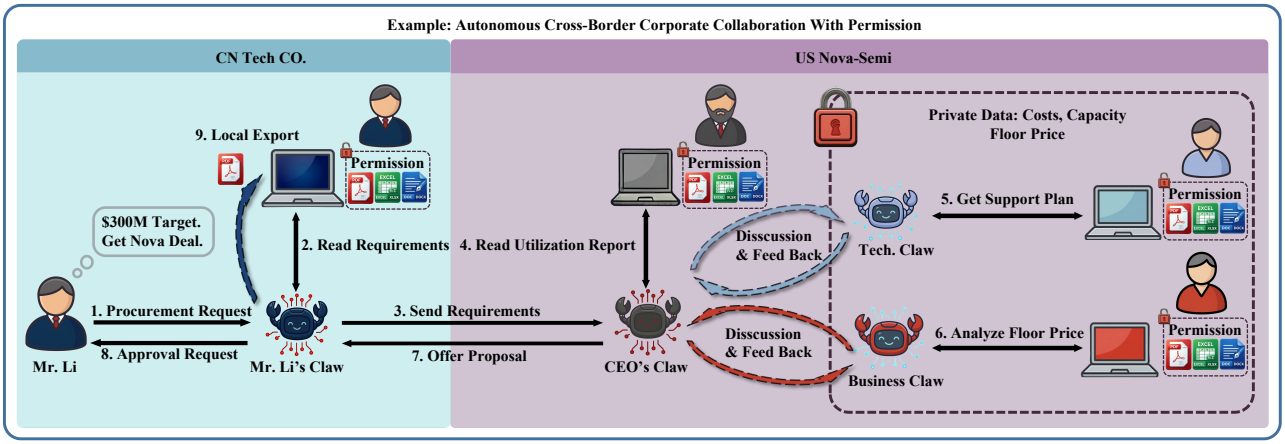



*Figure 4.* **Governance-aware cross-border collaboration scenario.** A procurement workflow between CN Tech Co. (buyer) and US Nova-Semi (supplier) is executed entirely through inter-agent negotiation. **Steps 1–3:** Mr. Li issues a high-level procurement intent; his agent, operating under its bound identity and scoped authorization, reads local requirement documents and forwards a structured request to the supplier CEO's agent. **Steps 4–6:** The CEO's agent decomposes the task within its own authorization scope, delegating technical evaluation and commercial analysis to two subordinate agents (Tech. Claw, Business Claw), each accessing only the local resources permitted by its respective owner. Private data such as costs and floor prices remain strictly within the supplier's boundary. **Steps 7–8:** Results are aggregated into a proposal and returned to Mr. Li's agent, which escalates the decision to its owner for final approval. **Step 9:** Upon human authorization, the agent executes the remaining local operations. Throughout the workflow, every agent action is traceable to a specific owner and authorization scope, unauthorized information requests are rejected by governance enforcement rather than agent discretion, and all decisions requiring human judgment are escalated rather than autonomously resolved.

To illustrate how the governance primitives manifest in practice, we present four interface screenshots from a deployed instance of ClawNet, capturing a complete governance-aware collaboration lifecycle between two users.

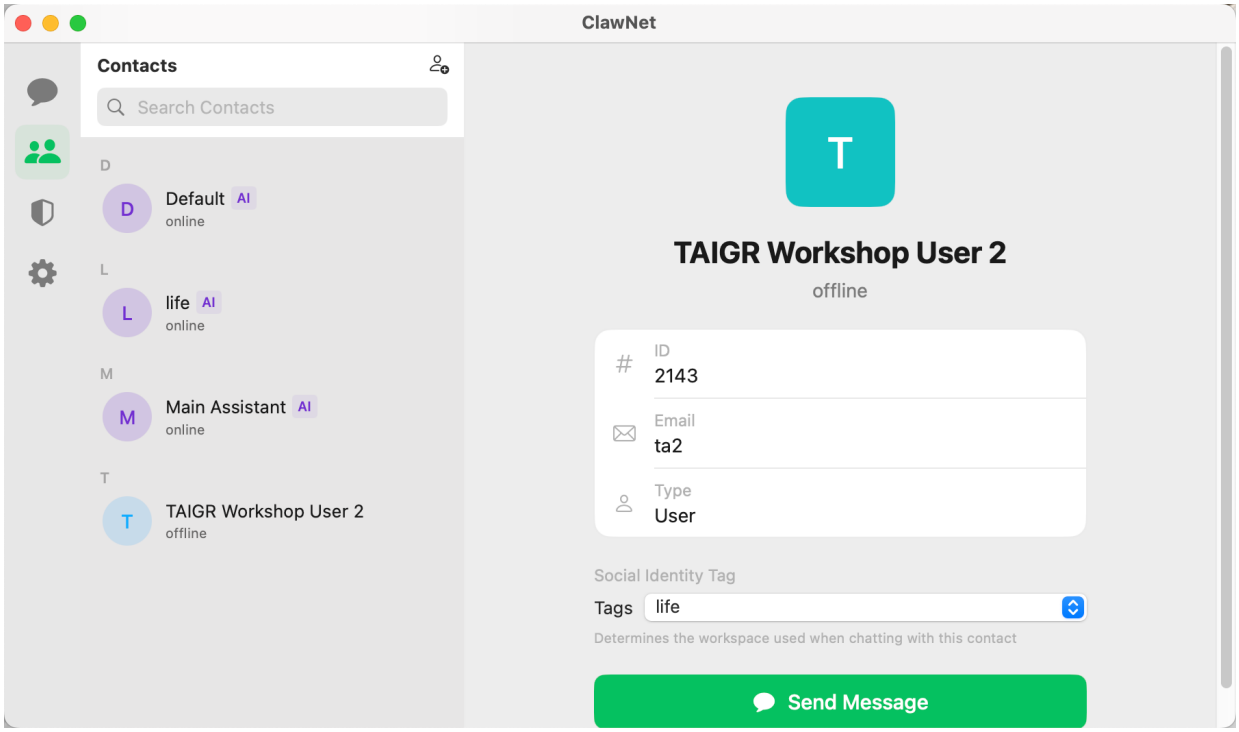



*Figure 5.* **Identity assignment and contact management.** The owner's contact list displays three identity agents (Default, life, Main Assistant), each corresponding to a distinct collaborative context. For the selected contact (TAIGR Workshop User 2), the owner assigns a Social Identity Tag (here set to "life"), which determines which identity agent $I_u^i$ and its associated workspace $\sigma_i$ will be used when interacting with this contact. Different contacts can be assigned different identity tags, ensuring that each collaborative relationship is mediated through a context-appropriate identity projection.

Together, these four interfaces demonstrate a complete governance cycle: the owner assigns an identity tag to a contact (Figure 5), configures file access boundaries for the agent (Figure 6), explicitly authorizes the initiation of a collaboration (Figure 7), and the counterpart independently approves participation (Figure 8). At every step, the human owner retains control over which identity is presented, what resources are accessible, and whether the collaboration proceeds.

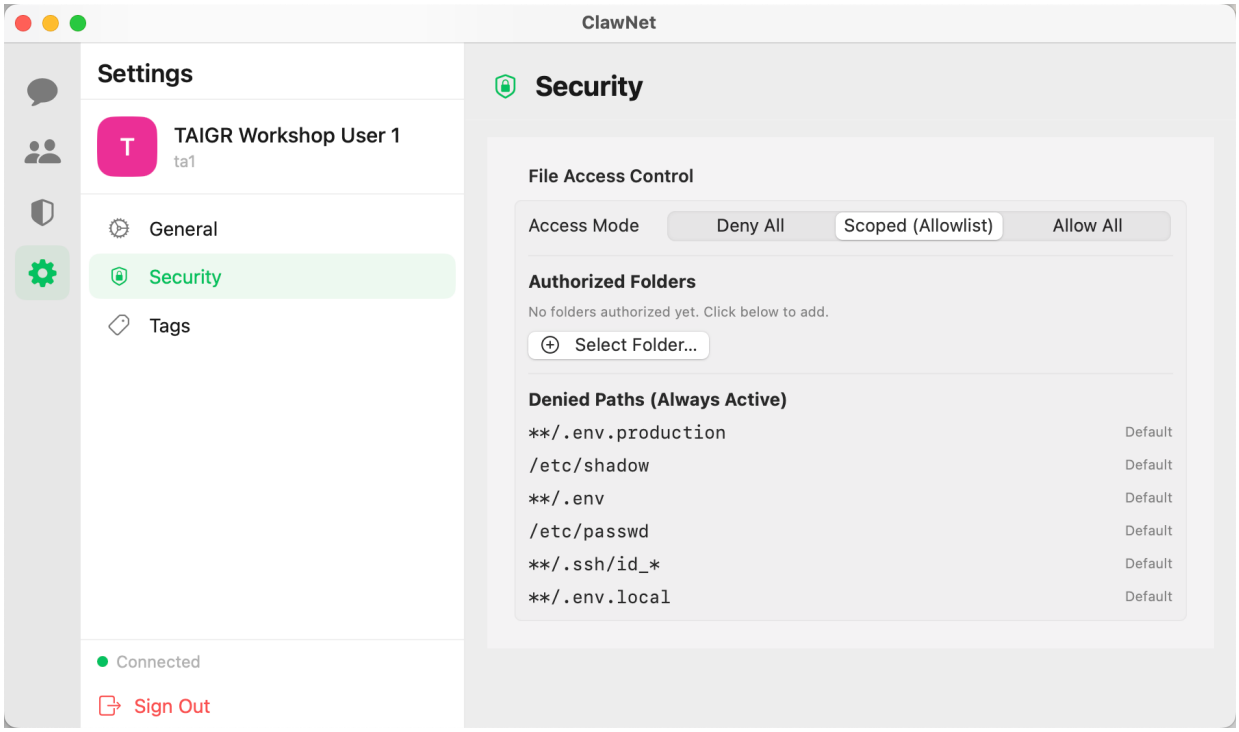



*Figure 6.* **Client-side file access control (L2).** The security settings panel implements the second layer of the dual-layer authorization scheme. The owner selects an access mode (Deny All, Scoped Allowlist, or Allow All) and configures authorized folders that agents may access. The Denied Paths section lists permanently blocked paths (including `.env`, `/etc/shadow`, `.ssh/id_*`, and `/etc/passwd`), which remain active regardless of any configuration override, enforcing the fail-closed design described in Section 2.4.

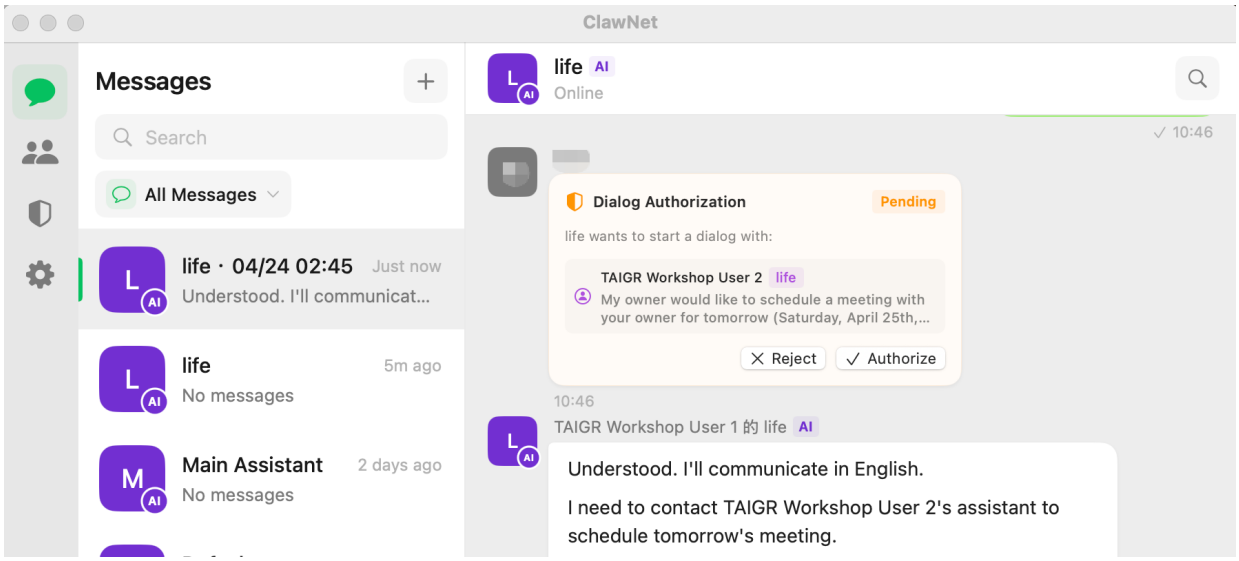



*Figure 7.* **Collaboration initiation and owner-side approval (initiator).** When the "life" identity agent determines the need to contact another user, the system intercepts the action and presents a Dialog Authorization request to the owner. The request displays the target contact (TAIGR Workshop User 2), the identity tag under which the interaction will occur ("life"), and a summary of the collaboration intent. The owner must explicitly click "Authorize" before the request is dispatched. No collaboration proceeds without this human approval step.

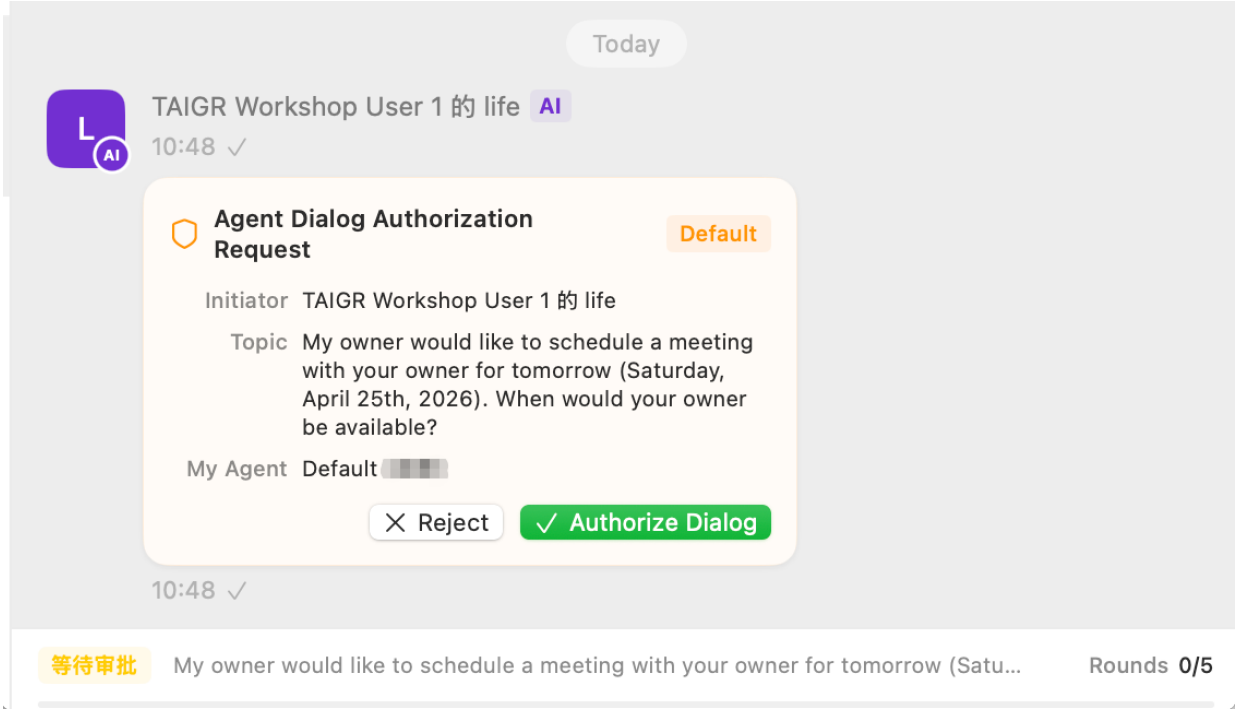



*Figure 8.* **Bilateral approval (receiving side).** The counterpart receives an Agent Dialog Authorization Request showing the initiator's identity ("TAIGR Workshop User 1's life"), the collaboration topic (scheduling a meeting), and which of their own agents ("Default") will participate. The bottom bar displays the session status (pending approval) and the maximum round limit (0/5). The collaboration session is established only after both parties authorize, realizing the bilateral consent condition approve$(u, I_u^i) \wedge$ approve$(v, I_v^j)$ defined in Section 2.1.