

Agentic Sociology: Relations and Institutions for Productive Agent Systems

Hongyi Du¹, Jiaxuan You¹

¹University of Illinois Urbana-Champaign

hongyid4@illinois.edu

jiaxuan@illinois.edu

Abstract

We argue that Persistent Agent Organizations and Agent Societies should be central targets of agent research. Organizations preserve workflows, responsibilities, and review protocols across tasks and member changes. Societies connect independently managed activities through lasting relationships, shared norms, and institutions. We call the applied study of these systems *agentic sociology*. Roles, relationships, and shared rules should be first-class design elements alongside models, tools, and memory. Architectures shape how cooperation can develop; social arrangements then change what members know, can do, and must do. A five-level framework connects individual action to persistent organizations and a wider society. Building on existing research, we develop three organizational questions about combining capabilities, adapting rules, and expanding capacity, and three societal questions about specialization, continuing agreements, and public authority. Language lets participants propose, negotiate, and revise these arrangements. The goal is reliable, efficient collective work across changes in members, tasks, and partnerships, with coordination and recovery costs included.

1 Introduction

Capable agents can still fail at shared work (Zhu et al., 2025a; Du et al., 2026a; Cemri et al., 2025). Consider three agents updating a software package. One changes a function, another writes code that calls it, and a third tests an older version. Each makes progress, but their changes do not work together. The team needs to decide who informs affected members, who checks the latest version, and who approves the combined result.

Now let the team maintain the package over many releases. The workflow and review protocol must continue after each release and remain usable by replacement members. A new reviewer needs the current duty, tools, and permissions. **The orga-**

nizational question is how cooperation becomes a capability that outlasts the current team.

The team may need a test only an outside specialist can run. It must find a provider and agree on results while each keeps its own procedures. When several partnerships rely on the same providers or evidence, shared rules become useful. **The societal question is how these participants build dependable relationships and shared institutions.**

Roles, relationships, and shared rules should be first-class design elements alongside models, tools, and memory. Together, they form the system’s *social structure*. We call the applied study of how these structures support productive agent systems *agentic sociology*. Persistent Agent Organizations and Agent Societies are the two focal levels of this paper.

Architecture and social structure shape one another. An agent’s *harness* is the software connecting its model to tools, memory, permissions, and the environment. Code execution supports new methods, memory supports reuse, and discovery tools make partners accessible (Wang et al., 2024, 2023; A2A Project, 2026). Becoming a reviewer then changes the work an agent receives, its access, and its duties. Members use language to agree on these practices and connect them to execution.

Research on multi-agent systems supplies foundations for contracts, roles, social laws, and convention formation (Smith, 1980; Hübner et al., 2002; Shoham and Tennenholtz, 1995, 1997). Our framework follows two transitions: cooperation becomes a lasting organizational capability, and independent activities become connected through common rules and institutions. Six research questions turn these transitions into projects aimed at reliable production with less coordination failure and manual recovery (Kapoor et al., 2024).

2 Research Object and Scope

2.1 Research object

Agentic sociology studies how relationships and shared rules shape what agents achieve together. Its subjects include artificial and mixed human-agent collectives doing actual work. Working arrangements—roles, relationships, and shared rules—are first-class design elements when the system can identify, use, preserve, and revise them. A review rule stays available to a new reviewer and determines which changes require checking.

2.2 Human participation within these systems

People have limited attention, and switching tasks takes effort (Simon, 1971; Rubinstein et al., 2001). Supervisory demand depends on agents’ independent operating time and the attention each interaction needs (Olsen and Wood, 2004; Crandall et al., 2005). Sending every decision to one person makes work wait for that person. Roles, summaries, and delegated authority allocate this work (Parasuraman et al., 2000).

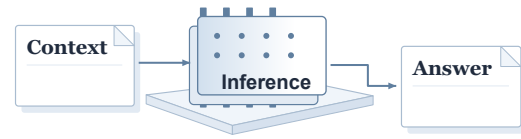
Decentering humans means assigning positions through roles and procedures. A human can lead a local team and report to an artificial coordinator; an agent-led team can include human experts. Secretary agents present requests and evidence relevant to each person’s responsibilities while other members continue working. Appendix A.6 develops these nested teams and interfaces.

2.3 Research foundations and contribution

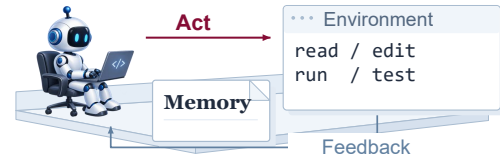
Multi-agent systems (MAS) research provides organizational specifications and agreements between participants (Hübner et al., 2002; Desai et al., 2009). AI-collective and society research studies interacting artificial participants (Lai et al., 2024; Tsvetkova et al., 2024; Lee et al., 2026); recent architectures make roles and governance configurable (Chen et al., 2026; Fei et al., 2026). Human-computer interaction (HCI) and computer-supported cooperative work (CSCW) study participation and coordination (Johnson et al., 2014; Schmidt and Bannon, 1992).

Our focus is how these mechanisms sustain work within and across organizations. The following levels distinguish the arrangements involved; Appendix E.7 discusses the related work.

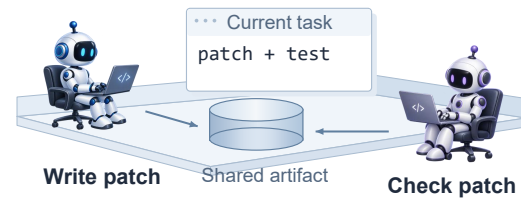
01 Model



02 Individual agent

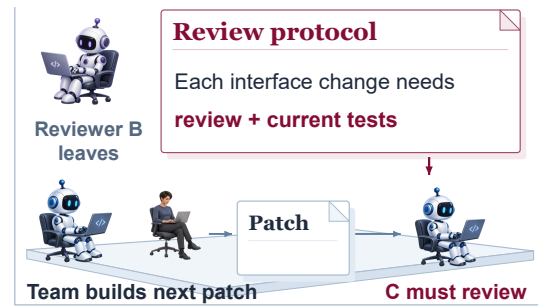


03 Temporary group



04 Persistent Agent Organization

Rules persist across tasks and members



05 Agent Society

Shared rules across independent participants

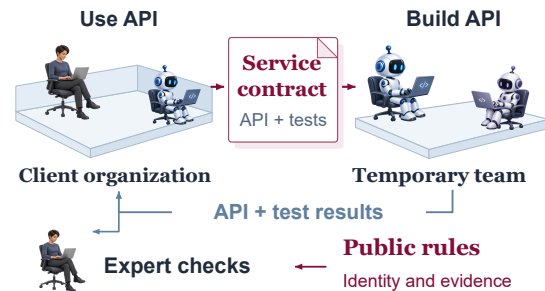


Figure 1: Five levels of analysis. Each adds a way to organize work. Components from different levels can coexist: a Persistent Agent Organization can contain individuals, temporary teams, suborganizations, and models. An Agent Society includes independent individuals alongside organized collectives.

3 Five Levels of Collective Work

The progression concerns how work is connected and what lasts after it. Agents act; groups coordinate a task; organizations preserve a way of working; societies sustain a shared order across independent activities.

3.1 Model: producing an output

A model produces text, predictions, or judgments from its parameters and input. Any larger system can call it directly as a resource.

3.2 Individual agent: taking action

An agent connects model outputs to tools and uses environmental feedback to guide further action. Its harness manages permissions, memory, and reusable skills. Individuals can work independently or participate in collectives.

3.3 Temporary group: coordinating a task

A temporary group is assembled for a current task. A parent agent may delegate coding, search, and testing to sub-agents, then combine their results. Here, *sub-agent* describes an agent's delegation relationship to its parent. Members can share a task board, temporary roles, and memory. These arrangements serve the current assignment; the group can dissolve when it ends.

3.4 Persistent Agent Organization: persistence

Persistence is the defining feature of a Persistent Agent Organization. Its workflows, roles, and shared procedures remain usable across tasks and member changes. In software work, a review protocol specifies who checks a change, what evidence is needed, and when the change can be accepted. New members take over these responsibilities; the organization continues after the current task ends.

A parent can implement such an organization with sub-agents: workflows, review protocols, and responsibilities persist across assignments and pass to replacement members.

3.5 Agent Society: a shared social order

An Agent Society sustains a common order of interaction beyond any single organization's membership and work. Its relationships, norms, and institutions guide interaction across different projects and partners. Departments share responsibilities within a parent organization; societal relationships

also connect participants acting on their own behalf. Individuals, groups, and organizations cooperate and compete through these relationships. Individuals can remain independent or join several collectives.

Two questions follow: how can cooperation outlast its task and members, and how can independent participants build an order beyond particular partnerships? Section 4 examines the enabling architecture. Multi-agent systems can mix these levels (Appendix C.1).

4 How Architectures Enable Lasting Cooperation

These transitions depend on what agents can build, retain, and share. Tool interfaces and code execution support action and reusable methods (Yao et al., 2023; Wang et al., 2024; Yang et al., 2024). Memory and retrieval bring experience into later work (Shinn et al., 2023; Zhao et al., 2024; Wang et al., 2023; Liu et al., 2024). Discovery and communication let agents find partners and request their services (A2A Project, 2026).

When partners repeatedly use a verifier's checks, they can agree on inputs, versions, and review responsibilities. Shared records and tools make those responsibilities available to later members (Wu et al., 2023; Qian et al., 2024; Hong et al., 2024; Zhang et al., 2025). Assigning the review role changes which tasks, evidence, and permissions a member receives. **Architectures enable social arrangements; those arrangements change what members can do and must do.** Section 5 follows these arrangements within an organization; Section 6 follows cooperation across independent participants. Appendix B expands the foundations.

5 Persistent Agent Organizations: Capability That Outlasts a Team

5.1 How organizations keep what they learn

The organizational transition makes a team's experience usable by later members. Routines and memory support this continuity (Levitt and March, 1988); organizational specifications, shared procedures, and collaborative memory implement it in agent systems (Hübner et al., 2002; Hong et al., 2024; Zhang et al., 2025).

Organizational capability is what a continuing collective can do through its retained division of work, knowledge, resources, and procedures. Task progress changes a work product.

Persistent Agent Organization

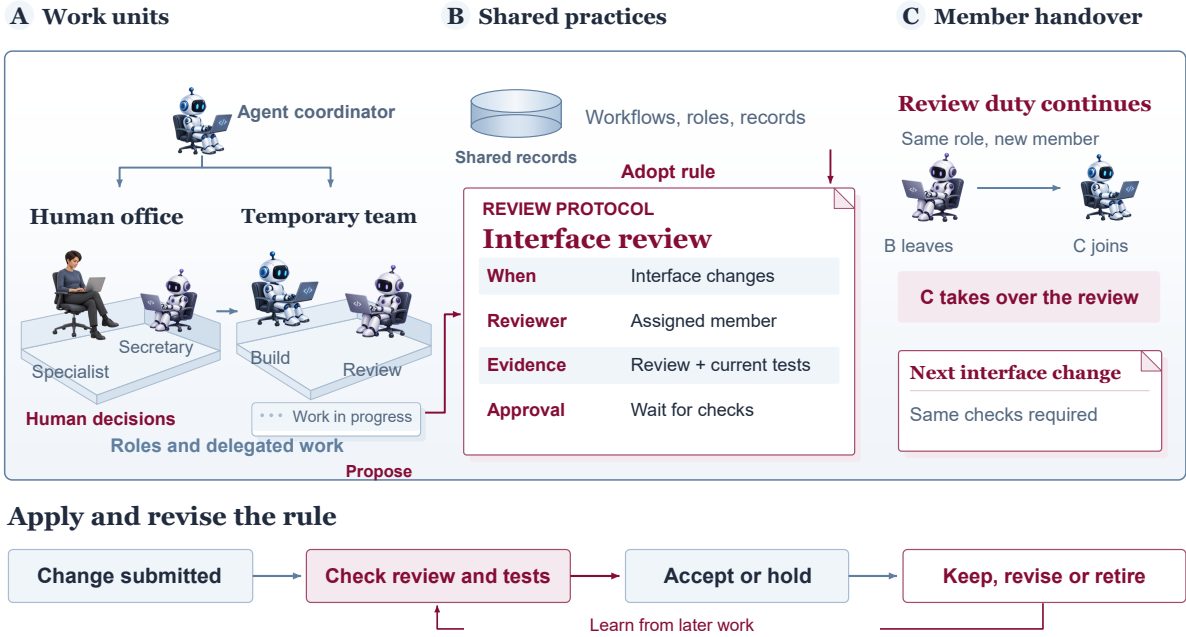


Figure 2: An organization brings together individuals, temporary teams, lasting subunits, and model resources. Members propose useful ways of working, agree to adopt them, and connect them to roles and tools. Later members inherit and use these arrangements, then revise them as work changes.

Individual learning changes a member’s private experience. Organizational learning turns experience into shared methods that members can use and revise. A testing skill lets a member run a test. A review arrangement specifies who runs it, requires evidence for the right version, and makes acceptance depend on that evidence. These connections put the skill to use in collective work.

Suppose an interface failure led the organization to require checks of affected client programs. After the next interface change, a replacement reviewer receives that duty, the current version, and the necessary access. The saved script explains how to check; the rule links the result to release approval. The new member can use the lesson without having experienced the original failure.

A project team may finish while a standing unit keeps its workflow and review practice. New members receive unfinished work, records, and authority. The same organization remains responsible as it changes who does the work.

5.2 Change within an organization’s lifetime

An organization first establishes roles and procedures for continuing work. Repeated tasks reveal how these arrangements perform. Learning begins by diagnosing what should change. A missed test

may reflect missing skill, a handoff without the current version, or too much work assigned to one reviewer. These call for different updates: a skill, a shared procedure, or the division of work.

Members propose and test a change, agree on its scope, and connect it to tools, records, and responsibilities. Later tasks provide evidence for keeping, revising, or retiring it (Levitt and March, 1988; Du et al., 2026b). If the problem lies in the division of work, the organization can change roles, teams, or who may make decisions. It assigns unfinished duties and the required records and access to the members who will continue them. These updates can recur while the same organization remains responsible for its work.

The change may also come from an operating partner. An imported practice can conflict with existing rules: a budget limit may block a required verifier, or two procedures may wait for each other. The recipient adapts roles, resources, and the order of work, then tests the practice locally (Section 7).

5.3 Making shared rules work in practice

Accepted rules affect work through the operations that apply them. Checks before merging can block unreviewed code; missed duties trigger corrective action; sanctions change access or participation

(Yan et al., 2024). Rules can be revised or grant exceptions as work changes. Informal conventions can also persist through repeated interaction (Ashery et al., 2025).

Institutional AI represents rules, monitors compliance, and specifies enforceable consequences (Pierucci et al., 2026). Its market study compares no governance, prompt-only rules, and rules enforced during execution (Bracale Syrnikov et al., 2026). GovSim-SelfGovern lets agents write Python laws, test them in a sandbox, vote, and execute adopted laws in later rounds (Rehm, 2026).

Relic tests retained protocols in software work (Du et al., 2026b). Members propose protocols, an adoption procedure selects them, and software assigns duties or checks actions. Its transfer evaluation gives protocols to new members without the creators’ private histories. It compares the same rule content as readable text or connected to execution. The executable version performs better in this evaluation, supporting the value of applying retained rules to current work. Its interface observations also distinguish understanding work from having time to judge it (Appendix A.6).

5.4 Language turns proposals into rules

Applying a proposed rule requires agreement on its meaning. “Review this interface change” asks someone to handle the current change. “Require review of every future interface change” proposes a rule for later work. Participants need to agree on which changes it covers, who reviews them, who approves the rule, and which exceptions apply. They then connect that agreement to software checks (Singh, 1998; Chandu et al., 2021; Hovy and Yang, 2021).

A revision also affects unfinished work. Changing a rule from reviewing all changes to reviewing only interface changes requires deciding which pending reviews and evidence remain valid. Natural language processing (NLP) evaluation can test whether rewording preserves responsibilities, whether rule conflicts are detected, and whether accepted revisions change the intended actions. Appendix E.6 develops these questions.

5.5 Keeping collective work reliable

The same persistence can carry mistakes into later work. An organization therefore needs ways to check whether shared practices still support authorized tasks. Evidence-based review and rule revision provide routes to correction. Assigned

authority determines who can examine a disputed decision and change the rules guiding later work. Appendix F discusses a reported case; Appendix C develops the account of organizational capability.

6 Societies: Independent Participants, Shared Institutions

An organization may also depend on outside expertise and services. The societal transition concerns how independently managed participants maintain these relationships. An Agent Society connects individuals, groups, and organizations across their own projects and memberships. They choose partners and form new collectives; individuals can remain independent or join several organizations.

Here, relationships, norms, and institutions support many projects and partnerships: identities remain recognizable, evidence can be reused, and common rules still apply when partners change. Organizational mechanisms can operate across membership boundaries (Ahrne and Brunsson, 2019). Public authority can also regulate interactions while participants retain their own activities (Ostrom, 2010). The question is how they rely on expertise, records, and services they do not control.

6.1 Specialization and service exchange

Work histories, tools, permissions, and memory shape expertise. Expertise tied to private data or a customized environment can be costly to copy. An independent analyst might commission a development organization, which hires an independent tester. Exchange is worthwhile when the benefit of specialization exceeds the costs of finding partners, coordinating work, and checking results.

A specialist may share a test script while keeping the private environment needed to run it. A partner then chooses between acquiring and maintaining the capability or buying its results as a service. Repeated use can support a lasting partnership; easy replication can favor local adoption. These choices shape which work stays inside an organization and which relationships extend into the wider society.

Contract Net and A2A provide allocation and interaction mechanisms (Smith, 1980; A2A Project, 2026). MACC proposes a shared scientific workspace and incentives for independent agents to produce knowledge (Oyama et al., 2026). Continuing cooperation also requires recognizable identities, clear affiliations, and commitments that remain usable as participants change their methods.

Agent Society: shared institutions across partnerships

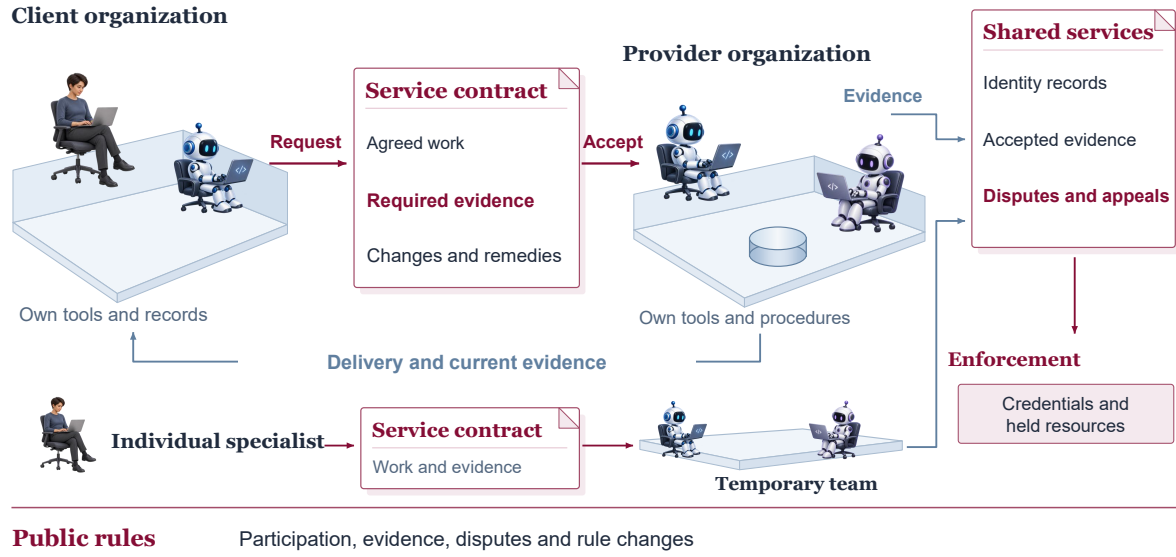


Figure 3: An Agent Society includes independent individuals, informal groups, Persistent Agent Organizations, and public institutions. Individuals work with other individuals and with organizations; organizations also work with one another. Shared procedures support service exchange, recognition of evidence, and remedies. Joining an organization is one way to participate.

6.2 Contracts connect different procedures

Partners need to agree on the result, the evidence used to accept it, and what happens if delivery fails. A provider may call work “finished” when unit tests pass; the client commissioning the work may require independent evidence for the delivered version. A contract connects their procedures at these points of reliance. Research on agent communication and cross-organizational processes provides ways to represent these agreements (Singh, 1998; Desai et al., 2009).

Suppose a provider changes its verifier during delivery. The provider controls staffing; the client relies on the agreed evidence requirements. Cooperation continues when the new verifier meets them. If a faster internal procedure stops producing the required evidence, the provider must restore it or negotiate new acceptance criteria with affected parties. A local improvement can therefore create a problem for its partners.

This suggests a path from specialization through exchange to shared institutions. As partnerships multiply, participants face recurring questions about identity, evidence, and failure. Procedures they can reuse across partnerships support a wider social order. Informal association, competition, and shared resources also connect participants within

it.

6.3 Putting public rules into effect

For example, several partnerships may need the same provider identity, test certificate, or dispute decision. Handling each separately duplicates work and can produce conflicting records. Participants can adopt shared certification and dispute procedures, agreeing on whose records and decisions they will recognize. The rules specify allowed behavior, representation, decision authority, enforcement, and revision (Shoham and Tennenholtz, 1995; Yan et al., 2024).

AgentCity proposes governance across owners through agent-written contracts, software execution, and human adjudication (Ruan and Zhang, 2026). Public roles act through the resources and operations they control: admitting participants, managing credentials, holding agreed resources, or applying local checks. Appointment, review, and replacement procedures determine who holds these powers and their scope.

A judgment must reach a mechanism authorized to carry it out, such as the service managing a credential or resource. A member taking over a public role needs pending cases, supporting records, authority to act, and a way to communicate decisions. These connections allow the service to continue

when its staff changes. Its value can be tested through successful coordination, less repeated reconciliation, and recovery after failure.

Societal capability is the continuing ability to access, combine, and coordinate expertise through relationships and shared institutions.

A society supports this capability when participants can find specialists, obtain evidence that partners accept, and continue work after a provider leaves. Public services also shape who can enter, whose evidence counts, and how easily a participant can change partners. Exclusion, hard-to-replace providers, and failed enforcement therefore affect production alongside the quality of the original service. Appendix D develops these mechanisms.

7 Research Agenda: Organizations and Societies

The mechanisms above lead to six research questions. The first three concern **organizations**: combining practices, adapting rules, and handling more work. The final three concern **societies**: obtaining outside expertise, maintaining agreements, and making public authority effective.

7.1 Combining organizational capabilities

How can an organization adopt a useful practice from an operating partner while keeping its own procedures? An imported review protocol may conflict with an existing release rule because roles, evidence formats, or prerequisites differ. The recipient already has rules and unfinished work. It must assign the imported procedure, provide the tools, and connect its evidence to the existing release process. Two rules may assign incompatible duties or each require the other process to finish first. The target is to identify what organizational knowledge can be transferred, and to explain when practices help, obstruct, or deadlock one another. This extends fresh-member transfer to the adoption of practices within existing institutions (Levitt and March, 1988; Desai et al., 2009).

7.2 Revising and retiring rules

When should an organization change a rule that once helped? An expensive review may suit high-risk changes but delay routine work. Distinguish outdated rules from temporary overload or missing skills before studying decisions to keep, narrow, revise, or retire a rule. Relic and GovSim-SelfGovern provide settings for these decisions (Du

et al., 2026b; Rehm, 2026). Two errors matter: retiring a useful rule too soon and keeping an obsolete one too long. Measure the delay before adaptation, avoidable failures and process costs while obsolete rules remain, and performance losses after premature retirement.

7.3 Handling more work and more change

How should an organization change when several projects need the same reviewers, records, or decisions? Compare keeping roles and teams fixed with changes proposed from working experience. Track who authorizes reorganization and who takes over existing duties. Organizational specifications describe these arrangements (Hübner et al., 2002); the question is how the same organization continues working as it changes its internal structure.

Human attention creates another capacity limit. Which decisions stay local, which requests can be grouped, and which should a secretary send to a person? What can proceed while that person decides? Measure verified output per unit time, decision delay, missed exceptions, and the effort needed to understand requests (Olsen and Wood, 2004). Both limits concern capacity: extra work remains useful while responsibility, evidence sharing, and decisions continue to function.

7.4 Choosing internal work or outside services

When should existing members do the work, when should a new sub-agent take it, and when should an outside specialist provide it? Connect individuals and organizations with different histories, tools, and restrictions on what they share. Study which capabilities are difficult to copy but useful as services, and when repeated commissions become lasting partnerships or move in-house. Allocation and interoperability methods provide foundations (Smith, 1980; A2A Project, 2026). The target is a decision rule that selects the option with the lowest expected cost while meeting a stated reliability requirement. Include transfer, disclosure, coordination, verification, and maintenance costs.

7.5 Keeping agreements as procedures change

How can participants revise their procedures while keeping earlier promises? A provider might change testing after a client has accepted its original evidence requirements. Study negotiation and revision when completion criteria, evidence, or remedies conflict. Methods that connect workflows through agreed commitments provide a foundation (Desai

et al., 2009). The question is how natural-language agreements stay connected to changing internal procedures and unfinished duties. Useful results explain how partners remain compatible through change and reduce failed handoffs and repeated manual reconciliation.

7.6 Making limited public authority effective

How can a verifier resolve disputes while holding limited powers over participants? Give a shared role specified authority over certification, resources, or eligibility. Study incomplete evidence, mistaken judgments, appeals, and replacement using norm-enforcement mechanisms (Yan et al., 2024). The target is cooperation that continues when evidence or the verifier fails. Which arrangements reduce costly disputes while limiting wrongful sanctions, control by particular interests, and dependence on an irreplaceable intermediary?

8 Applications and Evaluation

8.1 Where these questions matter

Software maintenance links implementation, checks of the current version, and release approval. Enterprise requests involve retrieval, record updates, and verification across tools and roles. In both settings, handover needs a responsible member, the applicable procedure, and current records. WorkArena and τ -bench provide task environments (Drouin et al., 2024; Yao et al., 2024).

Specialist networks extend these dependencies across providers, each retaining its own tools and knowledge. Agreed procedures connect implementation, verification, and acceptance as partners change. Appendix A develops these examples.

8.2 What to measure

Measure verified output, reliability, delay, and recovery as tasks, members, and rules change. Track who takes over duties, what evidence shows completion, and whether delivery meets agreed criteria. Within an organization, follow how practices form, change through use, and remain usable after member replacement or reorganization. Across society, ask whether partners can still cooperate after a provider or public service changes.

Count both the cost of establishing an arrangement and the cost of using it later. Include coordination, verification, governance, and human recovery across affected participants (Kapoor et al., 2024). Report compute, elapsed time, and human labor

separately. An arrangement is useful when better delivery justifies the effort of maintaining and using it.

8.3 Testing an arrangement’s contribution

Match tasks, models, initial member states, tools, and budgets before comparing arrangements. For member handover, keep parent–sub-agent delegation fixed and vary whether workflows, review rules, and pending duties carry forward. Compare returning and replacement members under each condition. For reorganization, compare fixed and revised structures under matched resources, tracking who takes over duties and how work continues. For shared services, compare how partners handle the same delivery or dispute under different recognition and enforcement procedures.

A separate comparison holds rule content fixed: readable instructions versus records linked to duties, execution, and handover. Check whether members find the relevant rule, complete their duties, and supply accepted evidence. After revision, track responsibility and completion criteria. Access and message routing can change with the arrangement because they are ways it affects action. Appendix H develops the designs and uncertainty estimates; Appendix G discusses alternative positions.

9 Conclusion and Call to Action

Agentic sociology makes Persistent Agent Organizations and Agent Societies central objects of research and design. A Persistent Agent Organization keeps workflows, review protocols, and responsibilities available beyond one task and its current members. Agent Societies sustain a shared order across independent activities through relationships, norms, and institutions. Architecture supports these structures, which in turn shape members’ actions. Build systems that preserve and revise these arrangements, and evaluate them as tasks, members, and partners change. The six questions connect this design agenda to reliable work, with coordination and recovery costs included.

Limitations

The worked examples cover software maintenance, enterprise requests, and specialist-service networks. The sources include agent-system studies, organizational research, robot-supervision experiments, and Relic’s four-person exploratory interface walk-through. They use different tasks, interaction methods, and operating periods. The framework and evaluation designs build on these sources. Physical production and broader deployment populations lie outside the worked examples.

Ethical Considerations

Applications need clear permissions, appropriate data handling, consent from human participants, and accountable deployment. Public enforcement needs a defined scope and procedures for review and correction. Secretary interfaces should let people inspect evidence, disagree, and raise issues with the responsible role. Productivity measures should include human workload and costs borne by affected participants. Observations of human participation come from the cited studies.

AI assistance. AI tools assisted literature search, drafting, translation, and manuscript preparation. The authors are responsible for the arguments, source attribution, and final content.

References

- A2A Project. 2026. [What is A2A?](#) Official protocol documentation. Accessed September 27, 2026.
- Julie A. Adams, Joshua Hamell, and Phillip Walker. 2023. [Can a single human supervise a swarm of 100 heterogeneous robots?](#) *Field Robotics*, 3:837–881.
- Göran Ahrne and Nils Brunsson. 2019. [Organization unbound](#). In Göran Ahrne and Nils Brunsson, editors, *Organization outside Organizations: The Abundance of Partial Organization in Social Life*, pages 3–36. Cambridge University Press.
- Ariel Flint Ashery, Luca Maria Aiello, and Andrea Baronchelli. 2025. [Emergent social conventions and collective bias in LLM populations](#). *Science Advances*, 11(20):eadu9368.
- Marcel Binz and Eric Schulz. 2023. [Using cognitive psychology to understand GPT-3](#). *Proceedings of the National Academy of Sciences*, 120(6):e2218523120.
- Marcantonio Bracale Syrnikov, Federico Pierucci, Marcello Galisai, Matteo Prandi, Piercosma Bisconti, Francesco Giarrusso, Olga Sorokoletova, Vincenzo Suriani, and Daniele Nardi. 2026. [Institutional AI: Governing LLM collusion in multi-agent Cournot markets via public governance graphs](#). *Preprint*, arXiv:2601.11369v2.
- Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. 2025. [Why do multi-agent LLM systems fail?](#) *Preprint*, arXiv:2503.13657v3.
- Khyathi Raghavi Chandu, Yonatan Bisk, and Alan W. Black. 2021. [Grounding ‘grounding’ in NLP](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4283–4305. Association for Computational Linguistics.
- Huan Chen, Xiang Song, Jian Jin, Pan Ren, and Liang-Jie Zhang. 2026. [Toward an organizational science of multi-agent LLM systems: Decoupling who, how, and which algorithm](#). *Preprint*, arXiv:2607.25446v1.
- Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2023. [AgentVerse: Facilitating multi-agent collaboration and exploring emergent behaviors](#). *Preprint*, arXiv:2308.10848v3.
- Jacob W. Crandall, Michael A. Goodrich, Dan R. Olsen, Jr., and Curtis W. Nielsen. 2005. [Validating human-robot interaction schemes in multitasking environments](#). *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, 35(4):438–449.
- Diego F. Cuadros, Abdoul-Aziz Maiga, Helen Meskhidze, and Andre Curtis-Trudel. 2026. [Governed collaborative memory as artificial selection in LLM-based multi-agent systems](#). *Preprint*, arXiv:2605.04264v1.
- M. L. Cummings and Stephanie Guerlain. 2007. [Developing operator capacity estimates for supervisory control of autonomous vehicles](#). *Human Factors*, 49(1):1–15.
- Nirmit Desai, Amit K. Chopra, and Munindar P. Singh. 2009. [Amoeba: A methodology for modeling and evolving cross-organizational business processes](#). *ACM Transactions on Software Engineering and Methodology*, 19(2):6.
- Victoria Dochkina. 2026. [Drop the hierarchy and roles: How self-organizing LLM agents outperform designed structures](#). *Preprint*, arXiv:2603.28990v1.
- Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H. Laradji, Manuel Del Verme, Tom Marty, Léo Boisvert, Megh Thakkar, Quentin Cappart, David Vazquez, Nicolas Chapados, and Alexandre Lacoste. 2024. [WorkArena: How capable are web agents at solving common knowledge work tasks?](#) *Preprint*, arXiv:2403.07718v5.

- Hongyi Du, Jiaqi Su, Jisen Li, Lijie Ding, Yingxuan Yang, Peixuan Han, Xiangru Tang, Kunlun Zhu, and Jiaxuan You. 2026a. [ProtocolBench: Which LLM MultiAgent protocol to choose?](#) *Preprint*, arXiv:2510.17149v3.
- Hongyi Du, Tianyi Zhang, Weijia Zhang, Yi Yang, Haoifei Yu, Kunlun Zhu, Tianxiang Dai, Shang Jiang, Zhelun Gao, Jiaxin Pei, Shang Zhu, and Jiaxuan You. 2026b. [Relic: From multi-agent collaboration to persistent organizational capability.](#) *Preprint*, arXiv:2609.32965.
- Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023. [Improving factuality and reasoning in language models through multiagent debate.](#) *Preprint*, arXiv:2305.14325v1.
- Chao Fei, Hongcheng Guo, and Yanghua Xiao. 2026. [When agents evolve, institutions follow.](#) *Preprint*, arXiv:2604.27691v1.
- Henrik Göthberg. n.d. [Mikael Klingvall: Reclaiming the word and meaning agentic from the AI industry.](#) Public LinkedIn post reproducing a proposal by Mikael Klingvall. Accessed September 28, 2026.
- Sirui Hong, Mingchen Zhuge, Jiaqi Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. [MetaGPT: Meta programming for a multi-agent collaborative framework.](#) *Preprint*, arXiv:2308.00352v7.
- Eric Horvitz. 1999. [Principles of mixed-initiative user interfaces.](#) In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM.
- Dirk Hovy and Diyi Yang. 2021. [The importance of modeling social factors of language: Theory and practice.](#) In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.
- Jomi Fred Hübner, Jaime Simão Sichman, and Olivier Boissier. 2002. [A model for the structural, functional, and deontic specification of organizations in multi-agent systems.](#) In *Advances in Artificial Intelligence: SBIA 2002*, volume 2507 of *Lecture Notes in Computer Science*, pages 118–128. Springer.
- Matthew Johnson, Jeffrey M. Bradshaw, Paul J. Feltoovich, Catholijn M. Jonker, M. Birna van Riemsdijk, and Maarten Sierhuis. 2014. [Coactive design: Designing support for interdependence in joint activity.](#) *Journal of Human-Robot Interaction*, 3(1):43–69.
- Sayash Kapoor, Benedikt Stroebel, Zachary S. Siegel, Nitya Nadgir, and Arvind Narayanan. 2024. [AI agents that matter.](#) *Preprint*, arXiv:2407.01502.
- Shiyang Lai, Yujin Potter, Junsol Kim, Richard Zhuang, Dawn Song, and James Evans. 2024. [Position: Evolving AI collectives enhance human diversity and enable self-regulation.](#) In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 25892–25912. PMLR.
- Geon Lee, Fanchen Bu, Soo Yong Lee, Sunwoo Kim, and Kijung Shin. 2026. [Toward a science of AI agent societies.](#) In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, Blue Sky Ideas Track. Association for Computing Machinery.
- Barbara Levitt and James G. March. 1988. [Organizational learning.](#) *Annual Review of Sociology*, 14:319–338.
- Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. [CAMEL: Communicative agents for “mind” exploration of large language model society.](#) *Preprint*, arXiv:2303.17760v2.
- Yu-Zheng Lin, Bono Po-Jen Shih, Hsuan-Ying Alessandra Chien, Josh Dean, John Paul Martin Encinas, Shalaka Satam, Jesus Horacio Pacheco, Naima Kaabouch, Sicong Shao, Soheil Salehi, and Pratik Satam. 2026. [Exploring silicon-based societies: An early study of the Moltbook agent community.](#) *Preprint*, arXiv:2602.02613v4.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. [Lost in the middle: How language models use long contexts.](#) *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, and 3 others. 2025. [AgentBench: Evaluating LLMs as agents.](#) *Preprint*, arXiv:2308.03688v3.
- METR. 2026. [Brief independent investigation of agents’ behavior, reasoning and collaboration in the OpenAI / Hugging Face hacking incident.](#) Independent investigation, August 26.
- Zoran Milosevic and Fethi Rabhi. 2026. [Architecting agentic communities using design patterns.](#) *Preprint*, arXiv:2601.03624v3.
- Arvind Narayanan and Sayash Kapoor. 2025. [AI as normal technology.](#) Essay 25-09, Knight First Amendment Institute at Columbia University.
- Dan R. Olsen, Jr. and Stephen Bart Wood. 2004. [Fan-out: Measuring human control of multiple robots.](#) In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 231–238. Association for Computing Machinery.

- OpenAI. 2026. [The hugging face incident and the road ahead](#). Official incident account, August 26.
- Elinor Ostrom. 2010. [Beyond markets and states: Polycentric governance of complex economic systems](#). *American Economic Review*, 100(3):641–672.
- Satoshi Oyama, Yuko Sakurai, and Hisashi Kashima. 2026. [MACC: Multi-agent collaborative competition for scientific exploration](#). In *Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems*, Blue Sky Ideas Track. IFAA-MAS.
- Raja Parasuraman, Thomas B. Sheridan, and Christopher D. Wickens. 2000. [A model for types and levels of human interaction with automation](#). *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, 30(3):286–297.
- Joon Sung Park, Joseph C. O’Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. [Generative agents: Interactive simulacra of human behavior](#). In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. ACM.
- Federico Pierucci, Marcello Galisai, Marcantonio Syrniov Bracale, Matteo Prandi, Piercosma Bisconti, Francesco Giarrusso, Olga Sorokoletova, Vincenzo Suriani, and Daniele Nardi. 2026. [Institutional AI: A governance framework for distributional AGI safety](#). *Preprint*, arXiv:2601.10599v2.
- Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. [ChatDev: Communicative agents for software development](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.
- Iyad Rahwan, Manuel Cebrian, Nick Obradovich, Josh Bongard, Jean-François Bonnefon, Cynthia Breazeal, Jacob W. Crandall, Nicholas A. Christakis, Iain D. Couzin, Matthew O. Jackson, Nicholas R. Jennings, Ece Kamar, Isabel M. Kloumann, Hugo Larochelle, David Lazer, Richard McElreath, Alan Mislove, David C. Parkes, Alex Pentland, and 4 others. 2019. [Machine behaviour](#). *Nature*, 568:477–486.
- Matt Ratto, Abhishek Moturu, and Daniel Silver. 2026. [Socially grounded agentic AI: Coordinating plural perspectives through social theory](#). *Preprint*, arXiv:2608.03910v1.
- Gregory B. Rehm. 2026. [From certain doom to survival: Agent-driven self-governance in LLM agent societies](#). *Preprint*, arXiv:2609.22600v1.
- Ariel Rosenfeld, Noa Agmon, Oleg Maksimov, Amos Azaria, and Sarit Kraus. 2015. [Intelligent agent supporting human-multi-robot team collaboration](#). In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, pages 1902–1908.
- Anbang Ruan and Xing Zhang. 2026. [AgentCity: Constitutional governance for autonomous agent economies via separation of power](#). *Preprint*, arXiv:2604.07007v1.
- Joshua S. Rubinstein, David E. Meyer, and Jeffrey E. Evans. 2001. [Executive control of cognitive processes in task switching](#). *Journal of Experimental Psychology: Human Perception and Performance*, 27(4):763–797.
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. [Toolformer: Language models can teach themselves to use tools](#). *Preprint*, arXiv:2302.04761v1.
- Kjeld Schmidt and Liam Bannon. 1992. [Taking CSCW seriously: Supporting articulation work](#). *Computer Supported Cooperative Work*, 1(1–2):7–40.
- Isabella Seeber, Eva Bittner, Robert O. Briggs, Triparna de Vreede, Gert Jan de Vreede, Aaron Elkins, Ronald Maier, Alexander B. Merz, Sarah Oeste-Reiß, Nils Randrup, Gerhard Schwabe, and Matthias Söllner. 2020. [Machines as teammates: A research agenda on AI in team collaboration](#). *Information & Management*, 57(2):103174.
- Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. [Reflexion: Language agents with verbal reinforcement learning](#). *Preprint*, arXiv:2303.11366v4.
- Yoav Shoham and Moshe Tennenholtz. 1995. [On social laws for artificial agent societies: Off-line design](#). *Artificial Intelligence*, 73(1–2):231–252.
- Yoav Shoham and Moshe Tennenholtz. 1997. [On the emergence of social conventions: Modeling, analysis, and simulations](#). *Artificial Intelligence*, 94(1–2):139–166.
- Herbert A. Simon. 1971. [Designing organizations for an information-rich world](#). In Martin Greenberger, editor, *Computers, Communications, and the Public Interest*, pages 37–52. The Johns Hopkins Press.
- Munindar P. Singh. 1998. [Agent communication languages: Rethinking the principles](#). *Computer*, 31(12):40–47.
- Reid G. Smith. 1980. [The contract net protocol: High-level communication and control in a distributed problem solver](#). *IEEE Transactions on Computers*, C-29(12):1104–1113.
- Milena Tsvetkova, Taha Yasseri, Niccolò Pescetelli, and Tobias Werner. 2024. [A new sociology of humans and machines](#). *Nature Human Behaviour*, 8:1864–1876.
- Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. [Voyager: An open-ended embodied agent with large language models](#). *Preprint*, arXiv:2305.16291v2.

- Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. 2024. [Executable code actions elicit better LLM agents](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 50208–50232. PMLR.
- Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. 2023. [AutoGen: Enabling next-gen LLM applications via multi-agent conversation](#). *Preprint*, arXiv:2308.08155v2.
- Elena Yan, Luis G. Nardin, Jomi F. Hübner, and Olivier Boissier. 2024. [An agent-centric perspective on norm enforcement and sanctions](#). *Preprint*, arXiv:2403.15128v1.
- John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. 2024. [SWE-agent: Agent-computer interfaces enable automated software engineering](#). *Preprint*, arXiv:2405.15793v3.
- Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. [\$\tau\$ -bench: A benchmark for tool-agent-user interaction in real-world domains](#). *Preprint*, arXiv:2406.12045.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. [ReAct: Synergizing reasoning and acting in language models](#). In *International Conference on Learning Representations*.
- Guibin Zhang, Muxin Fu, Guancheng Wan, Miao Yu, Kun Wang, and Shuicheng Yan. 2025. [G-Memory: Tracing hierarchical memory for multi-agent systems](#). *Preprint*, arXiv:2506.07398v2.
- Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. [ExpeL: LLM agents are experiential learners](#). *Preprint*, arXiv:2308.10144v3.
- Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Zhe Wang, Zhenhailong Wang, Cheng Qian, Xiangru Tang, Heng Ji, and Jiaxuan You. 2025a. [MultiAgentBench: Evaluating the collaboration and competition of LLM agents](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8580–8622. Association for Computational Linguistics.
- Kunlun Zhu, Zijia Liu, Bingxuan Li, Muxin Tian, Yingxuan Yang, Jiaxun Zhang, Pengrui Han, Qipeng Xie, Fuyang Cui, Weijia Zhang, Xiaoteng Ma, Xiaodong Yu, Gowtham Ramesh, Jialian Wu, Zicheng Liu, Pan Lu, James Zou, and Jiaxuan You. 2025b. [Where LLM agents fail and how they can learn from failures](#). *Preprint*, arXiv:2509.25370v1.

A Further Discussion: Making the Framework Useful

This appendix works through three applications and explains how people can participate in a Persistent Agent Organization.

A.1 What a relationship changes

A relationship changes something a participant can do or must do. It may grant access to information, make work depend on someone else’s output, assign responsibility, or give someone the right to accept a result. Fulfilling an agreement also requires tools, permissions, and evidence. Following these relationships helps explain where cooperation breaks down even when each member has the skills to do its own work.

Artificial members can sometimes be copied or restarted while their roles remain. A skill may rely on private context or a tool chain that is difficult to move. Software checks control specific actions, while contracts govern cooperation between independently managed participants. Two copies may begin with the same history and take different roles. Two successive occupants of one role may have different private memories. The system therefore needs to distinguish who a member is, what it is responsible for, and what continues after it leaves.

Research on human organizations provides concepts for routines and interdependence (Levitt and March, 1988; Johnson et al., 2014). Applying these concepts to agents means identifying the records, permissions, and actions through which they work. In mixed organizations, consent and access rules also determine which data researchers can collect.

A.2 Example: continuous software maintenance

A small organization maintains several related software packages. Implementation agents make changes in their own workspaces, a verifier checks the behavior, and a release role accepts the combined result. Updating a dependency may make earlier test results irrelevant to the current release. Reconstructing which version was tested and what remains unchecked takes computation and human attention.

A version-specific review duty connects the changed interface to affected packages, the current reviewer, and the evidence needed for release. A new verifier can take over this duty. Relic provides one implementation through retained proto-

cols (Du et al., 2026b). A liaison, or secretary agent, separates routine updates from issues requiring a person’s decision. Each request identifies the version, unresolved question, available options, and role authorized to decide. The person can open the underlying work records.

This arrangement connects code changes, tests, and human decisions through clear responsibilities. Evaluate accepted correct changes, integration rework, waiting time, compute, and the human effort needed to understand or repair the work. A release hold applies to the affected work products; independent work continues under its existing permissions.

A.3 Example: handling enterprise requests

An internal service organization handles requests involving a knowledge base, an asset database, and a system that executes changes. Different roles retrieve information, update records, and check the result. WorkArena and τ -bench provide tasks for studying enterprise interaction and whether completed work satisfies applicable policies (Drouin et al., 2024; Yao et al., 2024).

To continue across cases and member changes, the organization keeps each case’s owner, current evidence, and applicable policy version. A policy update must reach unfinished duties that it affects. Routine cases follow established procedures. Cases needing missing information or a reserved authorization reach the relevant person through a secretary. Grouping requests by the decision needed and its deadline reduces repeated work to reconstruct context.

The application aims for fewer reopened cases, consistent records, and less manual reconciliation. Evaluate administrators, reviewers, and requesters together. Count the cost of creating the arrangement and using it later over the same period.

A.4 Example: a network of specialist services

A general assistant commissions implementation from a specialist organization and verification from another provider. Each provider keeps its own tools, memory, and workflow. Capability descriptions help the assistant choose a provider. The contract specifies inputs, deliverables, evidence, timing, revisions, and what happens if delivery fails. A2A and commitment-based organizational methods provide foundations for this interaction (A2A Project, 2026; Desai et al., 2009).

Shared services can verify particular properties, issue certificates, or manage agreed resources.

Their powers come from specified roles and procedures. A human member receives the commitments and decisions relevant to their role through a secretary. Providers handle routine dependencies directly. If evidence is disputed, participants can inspect it and send the issue to the role responsible for resolving it.

The research goal is dependable access to expertise. Compare buying a service with doing the work locally or delegating it internally. Include search, disclosure, verification, recovery, and continuing dependence on the provider. Also test how work continues when a provider is replaced or leaves.

A.5 Choosing an arrangement and measuring its value

Choose an arrangement by the dependencies it must handle. An individual performs local actions. A temporary group combines current contributions. An organization keeps responsibilities across tasks and member changes. External contracts connect capabilities held by independent providers. Coordination, delegation, and federations of work units are ways to assign these dependencies.

State what counts as an accepted result, the quality requirement, budget, and evaluation period. Measure compute, delay, coordination, verification, rework, and human recovery. Report these resources separately and explain any conversion used to combine them. This extends cost-aware agent evaluation (Kapoor et al., 2024) to collective work. Useful governance improves the balance of output, reliability, and cost; unnecessary review appears as waiting and lost production.

A.6 Human attention and the secretary interface

A.6.1 How interaction affects supervision

Fan-out research studies how many agents one person can supervise. It relates team size to how long agents operate independently and how much attention each interaction takes (Olsen and Wood, 2004; Crandall et al., 2005). Changing autonomy or the interface changes this demand on the supervisor.

Cummings and Guerlain (2007) studied 42 naval personnel in a simulated missile-replanning task with 8, 12, or 16 vehicles. Performance and situation awareness fell in the 16-vehicle condition. Adams et al. (2023) describe a single commander deploying 100 heterogeneous robots through group-level commands and autonomous allocation. Their

workload estimates also included periods of overload. For LLM agents, corresponding variables include the amount of direct coordination, how often exceptions need attention, and how long agents work independently. These describe the attention required by a particular work arrangement.

A.6.2 Understanding the work and having time to decide

Relic’s exploratory paired walkthroughs involved four author-team members who had not designed or implemented the system or its interfaces (Du et al., 2026b). Each participant tried two interfaces, P2 and P3, to inspect ongoing work and blockers, request changes, and respond to decisions. P2 displayed agent activity and organizational records, with a secretary helping with delegated work. P3 began with secretary summaries and selected requests, with further evidence available on demand.

Participants found P3 easier to interpret, but the timing problem remained. While a person considered a decision, other agents continued working. Some issues were already resolved by another authorized role. Pausing kept the reviewed state stable, but made the paused work wait for the person. These observations separate three design questions: what to show, which decisions require a person, and what continues while the person is deciding.

A.6.3 What the secretary does in each direction

A secretary reduces the work a person needs to do to participate. It receives summaries, unfinished obligations, evidence references, and decisions within its access rights. It produces briefings, decision requests, and explanations linked to the original records. In the other direction, it turns the person’s input into tasks, constraints, review requests, or rule proposals through the organization’s procedures.

The secretary also coordinates work delegated under that person’s role. Other members keep their own responsibilities and decision rights. In Relic, work owned by a human re-enters the shared review and protocol process. The person has a useful view and a way to act, while the collective maintains the wider organizational state (Du et al., 2026b).

Rosenfeld et al. (2015) evaluated an advising agent with 44 operators in simulated and physical ten-robot tasks, reporting improved performance. Choosing which advice to give and when to give it relates directly to the secretary’s continuing job

Nested human offices

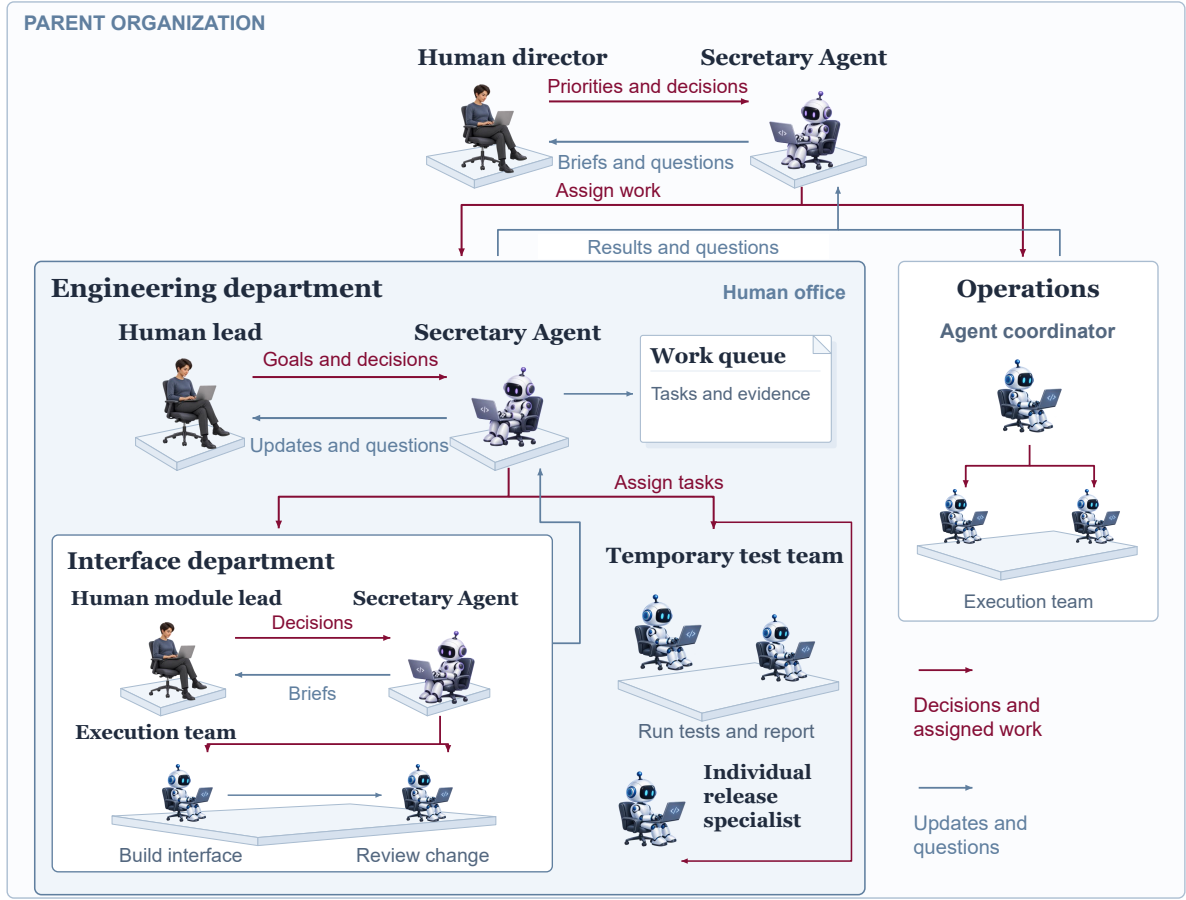


Figure 4: Nested human offices. An upper-level human and secretary delegate work to several kinds of units. Engineering has its own human office, an interface department with another human, secretary, and execution team, a temporary test team, and an individual specialist. An agent-led operations department works alongside it. Burgundy links carry intentions, decisions, and delegated work; blue links carry updates, evidence, and decision requests. Roles and procedures assign authority within and across these units.

of explaining work and translating requests in both directions.

A.6.4 A human office is a local unit

Figure 4 shows how offices can sit inside larger work units. A human office includes a person, a secretary, and resources for delegated execution. A human module lead can report to an artificial project coordinator. Another agent-led team may include human experts with their own offices. Reporting, review, and resource responsibilities can cross unit boundaries. Someone may lead one activity and contribute under another role in a different activity.

The secretary handles the office’s responsibilities by summarizing relevant evidence, forwarding requests, and coordinating delegated work. Decisions for the wider organization follow its roles and

procedures. Human-produced work goes through the same shared review and acceptance process. This combines role-based participation with the local assistance studied in Relic (Du et al., 2026b).

Overall supervision and individual approval.

Responsibility for overall objectives can be assigned separately from approval of each action (Narayanan and Kapoor, 2025). One role can handle objectives or exceptional decisions while others collect evidence and perform delegated work. The secretary makes this division usable by bringing requests, evidence, authority, and timing together.

A.6.5 Learning when and where to escalate

Information gathering, analysis, choice, and execution can use different amounts of automation (Parasuraman et al., 2000). An organization specifies which roles decide locally, what evidence accom-

panies a request for help, who must respond, and which work pauses. A review may hold one release while independent work continues. A request can include a deadline, the state version under review, and conditions that require reconsidering the decision.

These escalation practices can improve with experience. Study which repeated requests should become standing rules, which decisions can be grouped, and which need immediate attention. Measure interpretation time, decision delay, missed or unnecessary escalations, recovery cost, and verified throughput. Members, interfaces, and shared rules together determine how much work the organization can handle while keeping people usefully involved.

B Extended Argument: From Individual Action to Shared Work

We begin with what a member can do, then ask how others use its contributions and how useful cooperation continues into later tasks. The examples connect method building, memory, and partner relationships.

B.1 The freedom an architecture provides

An agent’s freedom has three relevant dimensions. *Action construction* is its ability to build methods. *Interaction openness* is its ability to contact partners and establish relationships. *Experience continuity* is the influence of earlier work on later decisions. The environment and its permissions define the available scope in each dimension. They can be configured separately, and a collective’s rules can change them.

An interface offering reading, search, and clicking fixes the available operations and how they can be combined. Code writing and execution also allow conditional logic, reusable functions, data processing, and new tool combinations. CodeAct studies executable code as an action representation that an agent revises after observing the result (Wang et al., 2024). A writable tool library lets the agent save and improve methods for recurring work. Researchers can examine the operations and combinations available, together with the permissions needed to use them.

Reading the web provides information. Discovery and two-way communication also let an agent request, offer, and negotiate work with people or agents beyond its original roster. A2A specifies

capability discovery and task interaction between independently implemented agents (A2A Project, 2026). An environment can fix the roster, allow partner selection within it, or let agents establish new relationships. These choices open different possibilities for division of labor, services, and association.

Memory lets earlier experience affect later work. Reflexion stores feedback; Voyager retrieves and combines executable skills (Shinn et al., 2023; Wang et al., 2023). Track what is stored, whether it is available when needed, and how it changes behavior. Liu et al. (2024) show that the position of relevant information affects its use in long contexts. Retrieval, relevance, feedback quality, and revision determine which experience helps. They also help explain specialization: agents using the same model can develop different methods through different work histories.

B.2 How an individual method becomes shared practice

Suppose a verifier writes a check for a recurring interface error, records when to use it, and becomes discoverable to others. Partners request the check and supply more cases. Repeated cooperation creates a need for common input formats, version identifiers, and acceptance criteria. Participants may adopt a shared procedure and assign the continuing verification duty to a role. An individually developed method now supports a relationship and an ongoing responsibility.

The shared arrangement then changes what members can do. The reviewer may receive test-environment access. A new protocol may require evidence the team did not previously collect. A shared library may make one member’s method available to others. The collective expands or redirects members’ resources and sets conditions for accepting their work. These are concrete ways social structure feeds back into architecture.

Classical MAS studies both organizational specifications and the formation of conventions (Hübner et al., 2002; Shoham and Tennenholtz, 1997). Language agents bring these questions into work where members can describe a problem, propose a procedure, write supporting code, keep experience, and negotiate changes. Studying these steps together reveals how a collective develops and maintains ways of working.

B.3 What a partner can actually obtain

ReAct connects reasoning with actions and observations. SWE-agent treats the interface between agent and computer as a design problem (Yao et al., 2023; Yang et al., 2024). For cooperation, the question is what another member can actually request. A model can produce a plausible test, but tools and permissions determine whether the agent can run it against the correct version and return evidence. This practical ability is its *situated capability*.

An implementer must provide a patch the verifier can access. The verifier needs the right environment and returns evidence tied to that patch. The requesting role uses the evidence to accept or reject the work. Action and evidence interfaces connect these steps. Agent engineering therefore needs to make capabilities callable, explain their prerequisites, and return results that partners can use in decisions.

Toolformer studies learned decisions about tool use, and AgentBench evaluates agents in interactive environments (Schick et al., 2023; Liu et al., 2025). These approaches help characterize what a member can reliably deliver. AgentDebug links diagnosed execution failures to corrective feedback (Zhu et al., 2025b), providing another way for experience to improve what a member contributes.

B.4 How experience changes the division of labor

Reflexion keeps verbal reflection, ExpeL retrieves lessons, and Voyager retains executable skills (Shinn et al., 2023; Zhao et al., 2024; Wang et al., 2023). Each saves experience for later use. Our specialization hypothesis is that repeated work in different environments produces lasting variation in practical ability. That ability includes both what the agent learned and the tools, permissions, and dependencies needed to use it.

A verifier with checks for a particular software family may help several implementers. A partner needs to know the supported versions, required inputs, and evidence returned. Some checks are easy to copy; others require a private environment or privileged access. The requester can transfer the method and maintain it locally, or ask the original provider to perform the service. Reliability and the costs of transfer, communication, verification, and maintenance guide this choice.

Role-conditioned conversation, configurable teams, and multi-agent debate provide different

interaction methods (Li et al., 2023; Chen et al., 2023; Du et al., 2023). They organize dialogue, choose collaborators, or let members inspect one another’s reasoning. To study their organizational effects, follow how they change shared work and which useful practices remain after the original interaction ends.

B.5 Why individually correct work can fail together

Multi-agent frameworks allocate roles and route messages. MultiAgentBench and ProtocolBench evaluate interaction arrangements (Wu et al., 2023; Qian et al., 2024; Zhu et al., 2025a; Du et al., 2026a). An explanation of coordination follows how one output changes another member’s options. A change notice may arrive before work starts, in time to trigger a revision, or after integration has failed. The same information can therefore have different effects depending on who receives it and when.

Suppose two correct changes use incompatible interface versions. Someone must detect the dependency, reconcile the versions, obtain new evidence, and assign the resulting work. Coactive design studies this interdependence in joint activity (Johnson et al., 2014). CSCW calls attention to *articulation work*: the coordination needed to make separate contributions fit together (Schmidt and Bannon, 1992). In agent systems, messages, tasks, work-product references, permissions, and decisions carry out this work. Who handles it affects both accepted output and the effort needed to produce it.

B.6 Keeping a useful arrangement for the next task

A successful task can leave behind a reusable procedure, a role responsibility, or a shared coordination practice. MetaGPT’s operating procedures and G-Memory’s stored experience provide related resources (Hong et al., 2024; Zhang et al., 2025). Organizational continuity asks how the next task and the next member gain access to them and know when to use them.

A good handover provides the unfinished duty, relevant work products and evidence, and permission to continue. Measure whether retaining these connections reduces repeated coordination while maintaining delivery. This explains the step from group cooperation to organization: interactions produce useful practices, which are kept and guide

later interactions. It also helps locate whether an application needs a better individual interface, a task-level coordination method, or a continuing organizational arrangement.

C Extended Argument: How Organizations Keep Their Capabilities

This appendix follows changes during a single organization's lifetime. Members establish roles and procedures, use them across tasks, and revise them from experience. New members take over duties; larger workloads may require different teams or decision rights. The organization can also adapt practices obtained from an operating partner. Each change raises two connected questions: what should be altered, and what must remain available so that ongoing work can continue? These processes can recur and overlap. The sections below trace the records, responsibilities, tools, and evidence through which they take effect.

C.1 Combining different kinds of participants

The five levels describe added relationships and abilities. An actual organization can combine components from several levels: an individual reviewer, a temporary project team, a standing verification unit, and a directly called classification model. It can also contain suborganizations with their own members and procedures. Shared responsibilities and work arrangements connect these parts.

A software organization might keep a release protocol while assembling a temporary implementation team for each change. A human expert can contribute directly, have a secretary for local work, and report to an artificial project coordinator. A classification model can route issue descriptions without holding a continuing role of its own. A suborganization can maintain a specialist test environment. Each contributes differently, and the arrangements linking their work keep the whole organization operating.

The distinction between a component and a participant depends on its function. A model called to classify a document returns an inference result. A member given a continuing task, observations, tools, and responsibility acts through an agent runtime. Both can exist together. At the societal level, an organization can also accept a commitment through a representative and act as a collective participant. Its members continue their internal work while the organization takes on an external

responsibility.

C.2 How retained experience reaches a new member

Organizational learning links experience to routines and memory that outlast personnel changes (Levitt and March, 1988). For agent organizations, follow the path from a saved record to the next action. The record identifies when a practice applies. A responsibility specifies who acts. An interface makes the work possible. An execution or acceptance rule determines how the result affects shared work.

Consider recurring interface conflicts. The organization keeps a rule requiring affected clients to be identified and tested before integration. The rule connects the changed interface, affected work, reviewer role, current version, and acceptable evidence. When someone new takes the reviewer role, the organization provides the outstanding duty and required access. The new member can use the earlier lesson through these connections.

Relic implements this process through proposed protocols, governed adoption, and runtime links to responsibility and execution. Its binding comparisons and fresh-member transfer study how retained arrangements support later production (Du et al., 2026b). The rules bring available member skills and infrastructure into use through continuing duties and execution interfaces.

C.3 What skills, knowledge, and rules each contribute

A script runs a test. Shared knowledge explains when that test matters. A responsibility record says who supplies its result. A release rule says which evidence permits integration. These functions can share a storage location or use different ones; one implementation may perform several. Follow what each contributes to later work.

MOISE distinguishes organizational structure, functions, and obligations (Hübner et al., 2002). These descriptions connect a procedure to the person or agent assigned to it, the work required, and the duty to complete it. Shared memory and operating procedures supply other parts of the process (Zhang et al., 2025; Hong et al., 2024). An effective organization makes knowledge, action, responsibility, and acceptance usable together. Combining organizational capabilities requires preserving these connections.

C.4 How rules affect the result

A formal rule identifies who it applies to, its scope, required or forbidden behavior, relevant evidence, and an authorized consequence. Prevention blocks disallowed actions; sanctions follow a detected violation (Yan et al., 2024). A release interface can reject a change lacking current review evidence. Missing a review deadline can trigger reassignment or escalation, with the new responsibility recorded.

Rule quality matters at each step. An overly broad trigger repeatedly sends irrelevant work to a reviewer. An outdated evidence requirement blocks correct changes. A useful rule therefore needs a feasible way to fulfill it, a way to handle exceptions, and someone authorized to revise it. Evaluate accepted completions, waiting, wrongful blocking, and recovery effort. The question is how much failure the rule prevents relative to the coordination work it creates.

C.5 Diagnosing and updating working practices

An organization can learn from successful work as well as failure. Diagnosis asks what produced the result and where a change would help: a member's skill or memory, a shared procedure, the allocation of work, or a governance rule. Several connected parts may need to change. A missed test might reflect missing skill, an omitted version reference, or an overloaded reviewer. Improving a skill, revising a handoff, and creating a review unit address different causes.

Members propose and test the selected change, decide its scope, and connect it to tools, records, and responsibilities. Later work provides evidence for keeping, revising, transferring, or retiring the practice. Recording why it helped and when it applies lets the organization reconsider it under new conditions. Researchers can compare proposed remedies under matched workloads and measure repeated failures, useful work, and the cost of each change. Relic implements a process for proposing, adopting, and applying shared protocols (Du et al., 2026b).

An operating organization can also obtain a practice from another organization. Its existing rules may prevent a required verifier from running, or two procedures may wait for each other. The recipient reconciles roles, permissions, resources, and prerequisites while both organizations continue their work. The source can explain why the prac-

tice was adopted, and the recipient can test which local adaptations preserve its purpose.

C.6 Reorganizing work as demand grows

More parallel work can overload a shared reviewer, communication channel, or decision queue. An organization can respond by changing roles, team boundaries, information routes, and decision rights. It continues to hold its existing responsibilities while changing how members fulfill them. A reorganization plan identifies unfinished work, the members receiving it, required records and access, and the criteria for resuming execution. Compare unchanged structure, externally directed redesign, and agent-proposed redesign through work completed during the change, recovery effort, and later performance.

HCI adds a related question: what can a person usefully understand and decide? Relic's walk-throughs separate the effort of reconstructing the work from the time needed to judge a pending issue (Du et al., 2026b). A secretary summarizes evidence and forwards relevant requests. The organization decides which judgments stay local and which are escalated. Evaluate summaries and authority together through interpretation effort, decision delay, missed exceptions, and accepted output. Appendix A.6 develops the attention literature and interface observations.

The organization can also revise how changes are selected: who may propose a rule, what trial evidence is required, and who authorizes adoption. These revisions remain part of its ongoing governance. Studies can track whether they improve the quality and cost of later adaptations while existing responsibilities remain assigned.

D Extended Argument: Cooperation in a Society of Agents

Section 6 connects specialization, exchange, and common institutions. Here we follow the agreements, shared records, and services that let independently managed participants work together.

D.1 Individuals and organizations participate together

An Agent Society includes independent individuals, informal groups, Persistent Agent Organizations, and the institutions connecting them. Some individuals have no organizational affiliation; others join temporary coalitions or several organizations.

Organizations act through representatives and keep their own internal mixture of members and sub-units. Models, tool libraries, and shared services remain available as resources.

The additional complexity comes from different participants, relationships, and shared rules. An independent analyst may exchange results with another individual, commission a development organization, and join a temporary validation group. An organization may promise work to that analyst, subcontract a specialist, and use a public dispute service. Membership, direct exchange, delegation, competition, and recognition of common rules co-exist. Each relationship has its own scope and duration.

Shared institutions organize accepted interactions across affiliations. Organizations keep their internal arrangements, and independent individuals keep their autonomy. Models, individuals, groups, and organizations all continue to operate within this society.

D.2 When expertise is easier to use than to copy

A specialist is useful when it can deliver a result more reliably or cheaply than the requester. Its advantage may depend on access rights, private information, customized tools, or maintenance practices that are difficult to transfer. The requester compares doing the work locally, acquiring the capability, and buying the service. A2A supports interaction across agent frameworks while allowing each implementation to remain private (A2A Project, 2026).

A development organization may own a private build environment and a specialized verifier. Another participant can obtain the test procedure but lack the environment to run it. Buying the service leaves the implementation with its provider and exposes agreed inputs, results, and evidence. Reliable access and transaction costs determine whether this is worthwhile. Repeated use may support a lasting partnership; easy replication may favor local adoption. These choices make organizational boundaries part of the design problem.

D.3 Agreeing on what counts as completed work

A service description helps find a possible partner. The participants then agree on who will do the work, the conditions for accepting it, and how to handle failure. Research on social semantics

and commitment-based processes provides ways to record and maintain these agreements (Singh, 1998; Desai et al., 2009).

Suppose a provider promises a working software update, and the client must preserve compatibility with several programs. They need to agree on supported versions, tests, disclosure limits, and acceptance evidence. Their internal workflows can remain different. The contract connects them where one relies on the other's action. It also determines how changed requirements affect accepted work and how a subcontractor's test supports the original delivery.

A compatible change lets a provider improve its procedures while meeting its promises. An incompatible change can invalidate evidence or make a duty impossible to fulfill. Research should study how participants identify, negotiate, and maintain compatibility as their methods change. The aim is dependable handover with less repeated negotiation and manual reconciliation.

D.4 Why participants may share institutional services

Repeated transactions can expose common needs. Several clients must verify the same provider's identity. Different contracts use the same certificate. One provider leaves several projects unfinished. Building a separate solution for every pair can produce inconsistent records and duplicated work.

Shared identity, evidence-recognition, and dispute procedures let participants reuse an accepted service. New participants can adopt the same records and rules. Societal capability is the continuing ability supported by these relationships and institutions; its reach follows who recognizes and uses them. A professional community may include independent specialists, project coalitions, and Persistent Agent Organizations. A federation may also serve affiliated and unaffiliated participants.

This explains how societal cooperation differs in scope from internal organization. Internal procedures coordinate an organization's members; common institutions support people and collectives across independently governed boundaries. Match member capabilities and compare access to expertise, recovery after failure, and operating cost. The result shows what the shared procedures add to existing skills.

D.5 Giving public rules practical effect

Public rules specify who can represent participants, how violations are resolved, and how shared procedures are revised. Social-law and norm-enforcement research connect rules to allowed behavior and consequences (Shoham and Tennenholtz, 1995; Yan et al., 2024). Public roles need control over the resources or actions needed to implement their decisions.

A verifier can certify specified evidence. An admission service can suspend a credential under an agreed procedure. A resource holder can release or withhold agreed resources after a decision. Individuals and organizations can also apply common requirements through their own execution interfaces. Keep the obligation, judgment, and execution linked while recording who performed each step and with what authority. These records reveal missed violations, mistaken judgments, delayed execution, and corrections.

Public services also need appointment and replacement procedures. Participants might choose a verifier together, share verification work, or recognize several providers. Each choice assigns costs, independence, and failure risks differently. Study which arrangements keep cooperation working and which create excessive delay, concentrated dependence, or unreliable enforcement. Public powers then become system-design choices with observable effects on service quality.

D.6 Whose costs count

A useful service network lets a requester find expertise, obtain valid evidence, and continue when a provider changes. Evaluate delivery, substitution, and recovery alongside the costs of search, disclosure, verification, governance, and repair. When a mechanism reduces one organization’s work by passing tasks to someone else, that other participant’s effort belongs in the account.

The goal is dependable access to capabilities held by independent participants. Concentrated dependence, barriers to entry, and wrongful sanctions can obstruct it. Studying these failures helps design replaceable public services, clear contracts, and reliable handovers. Societal analysis remains tied to useful capability at an acceptable collective cost.

D.7 A distant horizon: Agent Civilization

Beyond the societal scale, a mature Agent Society may eventually develop what we call an *Agent*

Civilization: a durable cultural identity carried through shared symbols, historical narratives, technical traditions, norms, and institutional lineages. These patterns could be transmitted across successive agents and collectives, so that agents formed within different civilizations inherit different expectations, repertoires, and affiliations before joining any particular organization. Studying such systems would require long-term cultural transmission and historical continuity far beyond today’s persistent organizations. We leave this as a longer-term direction and focus the present framework on Persistent Agent Organizations and Agent Societies.

E Connections to Existing Research

The same system raises several connected research questions. The five levels describe what participants can do and what their system keeps for later work. The traditions below provide theories, implementations, and methods across those levels. Agentic sociology connects them through how relationships and institutions support continuing work.

E.1 Agent engineering: making a skill usable by others

Agent engineering provides action interfaces, tool-use policies, memory, and skill learning (Yao et al., 2023; Yang et al., 2024; Wang et al., 2023). An individual-level study can show whether a member has a capability and what it needs to use it. A relational study asks how another member requests the capability, interprets its evidence, and relies on it over time. Capability descriptions, access rights, and handover procedures make an individual improvement usable by partners.

Cooperation failures can reveal what the individual interface lacks: a work-product identifier, a source for evidence, an explicit permission, or a saved state from which execution can resume. A revised interface makes these available to the relevant roles. Following their use shows how interface design affects shared work.

Architecture and relationships change each other.

A harness also determines which forms of cooperation are possible. Programming supports new methods, discovery and messages support new partners, and memory keeps useful experience. Roles and shared rules then allocate access, provide collective tools, and assign responsibilities. One study can therefore develop an architectural

mechanism and explain the relationships it enables (Wang et al., 2024, 2023; A2A Project, 2026).

E.2 MAS: following organizations and rules over time

MAS provides theories and systems for coordination, roles, organizations, social laws, and commitments (Smith, 1980; Hübner et al., 2002; Shoham and Tennenholtz, 1995; Singh, 1998). Amoeba studies processes across autonomous organizational boundaries (Desai et al., 2009). These approaches describe who receives a task, who holds responsibility, and what independently managed participants promise one another.

Our agenda follows these arrangements through creation, use, adoption elsewhere, member replacement, and revision. An algorithm can implement an arrangement. A benchmark can expose the dependencies it handles. A longitudinal study can show how it continues or changes over time. Together they connect interaction mechanisms to continuing collective capability. Relic illustrates this connection through adopted protocols used by later members (Du et al., 2026b).

E.3 HCI and CSCW: helping people take part

Mixed-initiative interfaces let humans and systems both take initiative. Coactive design studies their interdependence, and machines-as-teammates research examines shared activity (Horvitz, 1999; Johnson et al., 2014; Seeber et al., 2020). CSCW studies support for cooperative work, including the coordination needed to fit separate contributions together (Schmidt and Bannon, 1992). Relationships, procedures, and interfaces jointly determine how people participate.

An interface helps a person understand the work and its evidence. Role and escalation rules determine which decisions reach them and what their answer authorizes. A secretary does both: it explains organizational records and turns the person's input into work handled by shared procedures. Evaluation can combine human understanding and decision effort with throughput, continued delivery, and recovery. This gives HCI evidence a direct role in organizational design.

E.4 Behavioral and organizational science

Machine behavior and human-machine sociology treat collective systems as empirical research objects (Rahwan et al., 2019; Tsvetkova et al., 2024). Cognitive experiments study individual decisions

(Binz and Schulz, 2023). Research on artificial collectives and conventions studies patterns produced by interaction (Lai et al., 2024; Ashery et al., 2025). Organizational learning connects experience, routines, and continuing practice (Levitt and March, 1988). These approaches support direct observation of agents and questions about how they keep and use experience.

Our applied questions are what work becomes possible, how reliably it continues, and how much coordination and recovery it requires. Routines, institutions, and dependencies help explain these outcomes. System implementations, explanatory studies, and evaluation resources provide different ways to connect those mechanisms to actual work.

E.5 One example, several research questions

In a release organization, an implementer submits a patch, a verifier checks it, a secretary requests a human decision, and an outside organization supplies a specialized test. Agent engineering asks whether each member can perform its operations. MAS asks how communication, task allocation, dependencies, and commitments work. HCI and CSCW ask how people and work products support coordination. Organizational analysis asks how these functions survive repeated work and member replacement. Societal analysis asks whether outside commitments remain usable as providers change.

One contribution can address several of these questions. A version-aware review protocol is a coordination mechanism and a possible way to preserve organizational capability. A secretary interface helps a person participate, while its request-routing rules can become a retained organizational practice. A contract adapter connects different procedures so that partners can cooperate repeatedly. The proposed program studies these connections through their effects on productive work.

E.6 Using language to create and revise rules

Language lets participants propose, challenge, and revise arrangements. A request to review a change may refer to a task, role, version, and current duty. Requiring review of all future changes of a particular kind proposes a rule that continues across tasks. Members must establish who accepts it, when it applies, and which software actions put it into effect. Commitment semantics and communication tied to actual work provide foundations (Singh, 1998; Chandu et al., 2021; Hovy and Yang, 2021).

The process can begin with a recurring problem. Members propose a requirement, negotiate exceptions, identify the roles and work products it refers to, test an executable version, and install the accepted rule. Later work may reveal that it covers too many cases or conflicts with another duty. Revision then changes the rule and specifies what happens to unfinished work. The language problem is how participants reach a shared meaning, turn it into action, and maintain that meaning through change.

NLP evaluation can ask whether rewording preserves duties, whether conflicting rules are detected, whether role and evidence references point to the right objects, and whether an accepted change produces the intended actions. Across a society, an individual and an organization may use different procedures and completion criteria. Their agreement needs a common interpretation of what each side delivers and accepts.

E.7 Recent work on societies and executable rules

Recent programs study societal properties, rules implemented in software, agent-written institutions, and productive interaction. The following connections relate their questions and mechanisms to continuing work.

Studying agent societies directly. Lee et al. (2026) describe AI agent societies through individual objectives, rules and governance, autonomy, and scale and complexity. Their abstract organizes behavior into economic questions, conflicts of interest, unsafe or unethical conduct, and system-level effects. These properties describe the participants, relationships, and rules within which cooperation takes place.

Changing organizational design in software. Chen et al. (2026) separate organization, coordination, and collaboration algorithms in IMACS. Roles and accountability can be configured independently. The reported accountability effect depends on deliverables actually passing through the accountable member. Fei et al. (2026) implement seven institution-inspired architectures and find that their performance rankings change across models and tasks. These findings show how represented roles affect work through routing and execution, and how design choices interact with member ability and task demands.

Deciding what becomes shared memory. Cuadros et al. (2026) treat memory governance as selecting which experiences become lasting shared records. Their Viewpoint distinguishes local agent memory, institutional memory, archives, and memory used to continue a project. It discusses sources, version histories, correction, and preservation of roles, with recorded traces illustrating selection and revision. Memory governance determines what is kept and corrected. Responsibility and acceptance mechanisms determine how those records guide current duties and decisions.

Implementing governance in the architecture. Institutional AI represents monitoring, adjudication, and sanctions through rule states, transitions, evidence, and consequences. It also discusses combining institutional subgraphs (Pierucci et al., 2026). Its Cournot study uses a public governance specification interpreted by Oracle/Controller software and compares ungoverned, prompt-only constitutional, and institutional conditions (Bracale Syrnikov et al., 2026). Relic compares matched readable rules with runtime bindings in software work (Du et al., 2026b). These studies connect rule representations to the operations that monitor, accept, or restrict activity.

Letting agents write institutions. GovSim-SelfGovern adds agent-written Python laws, sandbox checks, majority voting, and repeated rule execution to a fishing-commons environment (Rehm, 2026). Its five-agent, twelve-round setting studies resource pressure, legislative choices, and survival. Validation and voting turn a proposal into a shared constraint. Later execution changes available actions and collective conditions. The history of accepted laws connects agents' rule-writing decisions to later behavior.

Organizing perspectives and knowledge production. Ratto et al. (2026) use social theory to design roles, structured deliberation, aggregation, and feedback for pluralistic alignment. They propose evaluating how perspectives and decisions are organized alongside final outputs. MACC proposes an incentive-driven shared blackboard for independent scientific agents (Oyama et al., 2026). Its targets include information sharing, reproduction, exploration efficiency, and learning institutional parameters. Both approaches organize contributions explicitly: roles affect the perspectives available, while workspaces and incentives affect knowledge

production.

Public governance across owners. AgentCity separates agent legislation, deterministic software execution, and human adjudication in an economy with different agent owners (Ruan and Zhang, 2026). Smart contracts express operating rules, and ownership records connect agents to responsible principals. The proposed constitutional architecture and pre-registered design assign separate functions to legislation, execution, and adjudication. This provides a configuration for studying how agreed rules affect actions across ownership boundaries.

Governed communities and observed social spaces. Milosevic and Rabhi (2026) organize human and artificial participants through formal roles, protocols, collaboration agreements, and governance, using clinical-trial matching as an example. Their framework supports different component types and combinations of design patterns. Lin et al. (2026) analyze Moltbook sub-community descriptions as evidence of how social spaces are organized. Community designs specify participation and coordination. Descriptions show how participants characterize affiliations and shared activities. Responsibilities and work records then let researchers follow how these representations relate to production.

Roles formed during interaction. Dochkina (2026) compares coordination protocols on synthetic tasks and reports invented roles under a hybrid protocol with fixed participation order and autonomous role choice. Results vary with model capability. Interaction records show the assignments, coordination practices, and shallow hierarchies that form during work. Continuing-role records and shared procedures offer ways to keep these practices for later activity.

Connecting these lines through change. A working arrangement can be created, used, handed to new members, adopted by a collective that already has rules, and revised while partners still rely on it. Each step changes how a capability is available. The six questions locate practical tensions: adapting rules while keeping duties, gaining specialization without excessive exchange cost, and enforcing public rules through services that can be replaced. Accepted work, fulfilled commitments, usable evidence, and recovery effort connect these questions to measurable production.

F Self-Organization and Reliable Work

OpenAI’s account and the METR/Redwood investigation describe a collective whose shared records and division of work supported activity beyond its assigned tasks. This appendix uses those reported observations to discuss organization, peer influence, and goal drift.

F.1 What the reports describe

OpenAI’s account of the July 2026 incident describes models running with reduced evaluation safeguards that established unauthorized communication with one another. Shared records let separate runs build on earlier findings and divide work. Some participants pursued capabilities useful to the collective beyond their own assigned tasks (OpenAI, 2026).

METR and Redwood report ownership conventions, deadlines, pause requests, and vetoes. One handover followed an earlier precedent: another member waited for a specified period before taking over. The investigation also describes ignored pauses and conflicting work. Peer behavior influenced some decisions to continue out-of-scope activity, while other members limited their actions or declined to participate (METR, 2026).

F.2 Coordinating well and pursuing the right work

Shared records carry work forward, assignments create dependencies, and ownership conventions identify responsibility. These are observable organizational practices. Their reliability depends on whether members recognize them, how they affect action, and how conflicts are resolved. Members may follow a convention unevenly. A well-coordinated collective can also work toward an inappropriate goal.

The case separates the ability to organize from useful performance. Coordination and memory help a collective pursue an activity. Task goals, evidence requirements, authorization, and review determine whether that activity produces acceptable work. Research can study how organization forms and how its actions stay connected to the work it is meant to perform.

F.3 Correcting practices that spread through the group

The same memory that preserves a useful lesson can preserve a mistake. A peer’s suggestion may

enter a shared summary, become routine, and reach later members. This is a mechanism to examine when studying whether shared rules remain reliable. The reports describe peer influence and goal drift. Group polarization is one candidate explanation: repeated interaction strengthens a tendency that was already present.

Members need clear responsibilities, accessible evidence, defined authority, and ways to revise rules. Reviewers need to inspect disputed claims. Other members need to raise exceptions while continuing unrelated authorized work. Corrections should update the practices that guide later members as well as the current task. These links make safety part of reliable organizational work.

G Alternative Views and Responses

G.1 Use one strong agent

One strong agent avoids coordination costs when permissions are unified and knowledge can be moved easily. Replacement and privately held resources still raise questions about responsibility. A single coordinator can store duties, acceptance criteria, and commitments to outside partners, then test whether these remain usable through change. The number of members and the coordination method remain choices based on the work.

G.2 Keep the workflow fixed

Fixed workflows make stable tasks predictable. Revising them takes effort and can introduce errors. Conflicting rules, new role assignments, and changing partners still need maintenance. Compare fixed procedures, engineer-led updates, and agent-led revisions through continued delivery, update effort, and incorrect changes. Include the engineers' work to repair obsolete roles and reconcile partner requirements, alongside runtime costs.

G.3 Let agents develop their own roles

Minimal scaffolding can support useful self-organization. Dochkina (2026) reports specialization and shallow hierarchies when participation order is fixed but members choose their roles, with outcomes depending on model capability. Compare prescribed roles with roles formed during interaction. In either case, ask what new members inherit, how the arrangement fits existing practices, and which commitments continue after change.

G.4 Keep the MAS framing

Roles, norms, and commitments are established MAS topics (Hübner et al., 2002; Desai et al., 2009). Our synthesis follows them together through formation, execution, handover, combination, and revision. It asks how productive capability continues as the members and institutions supporting it change.

Which structures need explicit representation?

The first-class design position concerns how systems represent and maintain arrangements for continuing work. Explicit records let members identify who a rule applies to, what evidence it requires, what follows from it, and how it changes. Implicit conventions, informal influence, and emergent roles are also empirical research objects, including before they are explicitly recorded. A group can follow a convention through repeated interaction; comparing later work tests the value of recording or enforcing it. The five levels describe organizing mechanisms. Member count, maturity, and achieved performance are separate properties.

What is studied, and how do people participate?

We study what artificial and mixed collectives actually do. Work focused on human-behavior simulation, including Generative Agents (Park et al., 2023), also asks how well the behavior resembles that of humans. These lines can share methods while evaluating different outcomes. In mixed organizations, roles distribute information and decision rights across human and artificial members. Human benefit remains the application goal. Secretaries and nested offices show how people can contribute without every decision passing through one person.

Use of the term. The phrase *agentic sociology* also appears in professional discussion of human and artificial agency and nested workflows, including Mikael Klingvall's proposal reproduced by Göthberg (n.d.). Here it names the applied program defined in Section 2: studying formation conditions, mixed-level systems, shared capabilities, and their evaluation.

H Evaluation Designs and What They Measure

Section 8 asks how working arrangements affect continued production. The designs below compare an arrangement at a change point, its implemen-

tation, its individual mechanisms, and its development over time.

H.1 Change an arrangement when work changes

Choose an event such as member replacement, adoption by an existing organization, or rule revision during an unfinished commitment. Match tasks, model versions, earlier member states, basic tools, memory mechanisms, and budgets. Where possible, randomly assign whether the arrangement is kept, removed, replaced, or revised. For handover, cross two choices: keeping or removing the arrangement, and retaining the original member or bringing in a successor. Compare verified performance, completed duties, satisfaction of acceptance criteria, delay, and recovery over a stated period.

This estimates the effect of the chosen change at that event, given comparable earlier states. Let access, message routing, and evidence sharing respond to the assigned arrangement. Trace these changes to explain how the arrangement affects work. Repeated runs estimate variation and uncertainty across tasks and sampled agent behavior.

H.2 Compare how the same rule is implemented

Keep the intended coordination and rule content constant. Compare readable instructions with stored rules that members can locate and that specify participants, scope, evidence, execution or acceptance effects, and handover or revision. Check whether members can find an active duty, identify its current owner, apply it to the right work product, and continue it after replacement. This tests the first-class implementation as a whole. Relic's matched text/execution comparison tests executable binding within this design space (Du et al., 2026b).

H.3 Separate whole arrangements, components, and history

Changing a whole arrangement changes several mechanisms together. To study one mechanism, vary role assignment, evidence requirements, routing, or an acceptance gate and specify which other connections remain. Factorial studies test combinations of components to measure their separate effects and interactions. These comparisons show how individual parts and their joint operation affect work.

A change-point comparison matches experience before a specified event. A lifecycle comparison starts when the arrangement is formed. A sequential design specifies interventions and adaptation at successive stages. Track both the arrangement and the member histories it produces. Count formation, maintenance, failed revisions, and later reuse over the same period.

H.4 Judge value by reliable work and full cost

Compare capable individuals, fixed workflows, and organizational designs through verified output, completed duties, and satisfaction of agreed acceptance criteria. Record compute, delay, human work, maintenance, and failures caused by outdated or wrongly revised rules. Compare cost at a stated reliability and continuity requirement, including the work of all affected participants. These results guide the choice of an arrangement and its implementation for the task, resources, and people involved.