

Tacit Work Needs a Mission Room

Accountability, Ownership and Data Sovereignty in Agent-Assisted Professional Services

P. Sarode

R. Nair

Adjunct Researchers-in-Residence | Future of Ways of Working | Theoremlabs.io

October 2026 · Working paper, draft for comment

tacit.theoremlabs.io · Comments: research@theoremlabs.io

Agents draft, a named person confirms, and each party keeps what is theirs.

Abstract. Expertise engagements between a provider and its client depend on knowledge nobody has written down: expert judgment, and what the client's people know but have never documented. They run on chat, file shares and data rooms, which record neither who knows what, who owns what, where it resides, nor who decided. AI agents can now interview, draft and score at scale, which makes those gaps more costly. We propose the Mission Room: a bounded, staged workspace in which agents draft and a named person confirms, the client's context remains the client's, and the provider's expertise remains the provider's. We argue that such a workspace will not be taken seriously unless ownership is separated and enforced by the system, and unless its answers to data sovereignty, security and privacy can be demonstrated under assurance standards such as SOC 2. We specify the Mission Room through these preconditions, a governance pattern called Draft, Confirm, Commit, a consolidated set of invariants, and an operating model in which a Lead and an optional Operator work with an agent roster in place of the traditional pyramid. We describe an open-source reference implementation, Tacit Chassis, built to hold the invariants, and state eight testable hypotheses. The paper proposes a design and reports no field data.

Keywords: tacit knowledge; human-AI collaboration; agentic AI; professional services; consulting; accountability; data sovereignty; computer-supported cooperative work

Disclosure. This research paper and the open-source reference implementation it describes are supported by Theoremlabs.io (tacit.theoremlabs.io). It is a working paper: it has not been peer reviewed and reports no empirical results. Each author is an Adjunct Researcher-in-Residence, Future of Ways of Working, at Theoremlabs.io, which develops Tacit Chassis, the reference implementation described in Section 10. We invite comments, critique and collaboration from researchers and practitioners at research@theoremlabs.io.

1. Introduction

Consider a familiar engagement. A senior advisor and a small team are asked to assess how a client organization actually works: where decisions stall, which controls exist only on paper, which risks or opportunities are real. Over several weeks they interview dozens of people, read what documentation exists, and form a view. The result is a report or a slide deck. Weeks later, few people can say which interview a given conclusion came from, which conclusions the senior advisor examined personally, or where the client's confidential material now resides.

This kind of work is old and economically significant. Advisory, due diligence, implementation readiness, integration planning and many forms of assessment share its shape. Yet the software it runs on was built for other purposes: email and chat for conversation, shared drives for files, data rooms for controlled document exchange, slide software for the final product. None of these tools knows what a mission is, who must confirm what, how a finding relates to its evidence, or which knowledge belongs to which party.

Other kinds of team work have acquired purpose-built environments. Design and product teams work in tools such as Figma and Miro. Engineering teams work in issue trackers and wikis such as Jira and Confluence. In each case the tool encodes the structure of the work. Expertise engagements between a provider and its client have no equivalent.

AI agents change the urgency of this gap. Agents can now conduct structured interviews, transcribe and summarize them, draft findings and propose scores. That removes much of the labor that justified large engagement teams. It also creates three problems. The first is accountability: when an agent drafts a finding and a person forwards it, it is no longer clear who decided. The second is ownership: a system that interviews a client's people and applies a provider's method holds both parties' most sensitive knowledge at once, and an agent system that learns from one engagement may carry it into the next. The third is sovereignty, security and privacy: client content, including recordings of named individuals, flows through agents and model services whose location, retention and access the client may not control.

This paper makes four contributions. First, it characterizes tacit work and argues that it constitutes a distinct category of collaboration (Sections 2 to 4). Second, it specifies the Mission Room and two preconditions without which neither providers nor clients will adopt one: separation of ownership between client context and provider expertise, and solutions to data sovereignty, security and privacy that can be demonstrated under assurance standards such as SOC 2 (Sections 5 and 6). Third, it defines a governance pattern, Draft, Confirm, Commit, and consolidates the invariants that any Mission Room must hold (Sections 7 and 8). Fourth, it proposes an operating model that replaces the staffing pyramid (Section 9) and describes an open-source reference implementation (Sections 10 and 11). Section 12 states

hypotheses, and Section 13 sets out limitations. We regard the first two contributions as the paper's main ones. The governance pattern and the operating model build on active lines of work, which we cite where they apply.

2. What Makes Work Tacit

Polanyi's observation that "we can know more than we can tell" (Polanyi, 1966) is the usual starting point for discussions of tacit knowledge. Nonaka and Takeuchi (1995) described how organizations convert tacit knowledge into explicit form and back. Collins (2010) distinguished kinds of tacitness, from knowledge that simply has not been written down to knowledge that is embodied or collective and resists articulation. The naturalistic decision-making tradition showed how experts decide under uncertainty (Klein, 1998) and developed interview methods, known as cognitive task analysis, for eliciting that expertise (Crandall, Klein and Hoffman, 2006). Leonard and Swap (2005) described the experience-based judgment of senior practitioners as "deep smarts" and examined how it is, and is not, transferred.

This literature is mostly concerned with knowledge inside one organization. We are concerned with a narrower unit: a piece of work in which one party is engaged to draw out and judge knowledge held by another. We call this *tacit work* and propose that it has five properties (Table 1).

Table 1: *Five properties of tacit work.*

Property	Description	Consequence for tooling
Crosses a boundary	A provider works with and on behalf of a client. Two organizations, with different interests, confidentiality duties and intellectual property, share one piece of work.	Isolation per provider and client pair; explicit ownership of each kind of knowledge.
Depends on elicitation	The most important inputs are in people's heads. Documents are partial, out of date, or describe intent rather than practice.	Interviews are first-class inputs, kept and linked to what they produce.
Ends in judgment	The output is a set of assessed findings, judged against the provider's method.	Findings carry structured fields and scores against a stated rubric.
Carries accountability	A named professional stands behind the findings, often with reputational or contractual exposure.	Confirmation by a named person is recorded and cannot be bypassed.
Is time-boxed	The work has a defined scope, stages and an end.	The workspace is organized around a mission with stages, not an open-ended channel.

We mean what Collins (2010) calls *relational* tacit knowledge: knowledge that is unwritten but can be told when someone asks well. Embodied and collective tacit knowledge, which resists articulation, is out of scope. Three of the five properties (judgment, accountability and time-boxing) hold for most professional work. What is distinctive is their combination with boundary-crossing and elicitation.

Examples span industries: commercial due diligence, operating-model discovery, go-live and implementation readiness, post-merger integration planning, market-entry assessment, AI adoption assessment, and site assessment for economic development. Each involves a provider eliciting knowledge from a client's people, judging it against its own method, and standing behind the result.

3. The Collaboration Landscape

Collaboration software has long been studied as a support for articulation work, the effort of coordinating distributed activity (Schmidt and Bannon, 1992). Commercial tools have since divided along two lines that matter here. The first is orientation: whether a tool is general purpose or built around the artifacts and stages of a particular kind of work. The second is boundary: whether a tool is designed for work inside one organization or between organizations with distinct interests. Figure 1 places common tool classes on these two axes.

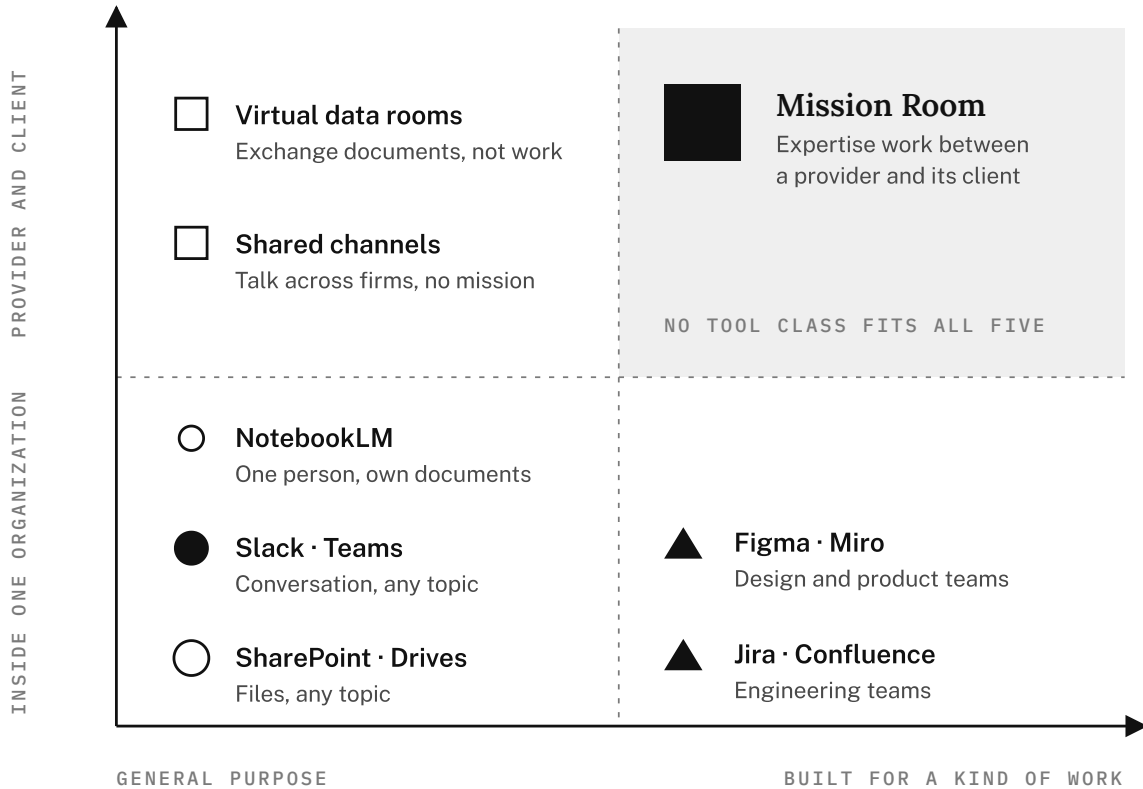


Figure 1: Collaboration tool classes by orientation and boundary. Placement is indicative. Shared channels and data rooms cross the boundary but are general purpose; design and engineering tools are work-specific but mostly internal.

No established tool class combines the five properties of Table 1, and three sit near the upper-right quadrant of Figure 1 and are worth distinguishing from it. A virtual data room crosses the boundary between parties and is controlled and audited, but it holds what parties exchange rather than the work itself. Shared chat channels across firms carry conversation between organizations but impose no structure on it. Personal research assistants such as NotebookLM help one person reason over documents they already have; they do not elicit knowledge from other people or record who confirmed what. Products have also begun to apply agents to parts of this work: AI-led expert calls and knowledge-capture tools now conduct interviews and synthesize the results. They cover elicitation, but not the mission structure, the confirmation gate or the separation of ownership that this paper argues for. Table 2 compares these classes against the properties in Table 1.

Table 2: *Support for the properties of tacit work, by tool class.*

Property	Chat and file shares	Virtual data room	Work-specific internal tools	Mission Room
Isolation per provider and client	Partial, by configuration	Yes	No	Yes
Client context kept apart from provider method	No	No	No	Yes
Interviews as first-class inputs	No	No	No	Yes
Structured, scored findings	No	No	Yes, for their own artifacts	Yes
Recorded confirmation by a named person	No	Access logs only	Review and approval steps	Yes, mandatory
Organized around a staged mission	No	No	Yes, for their own workflows	Yes
Finding traceable to source conversation	No	No	No	Yes

4. What AI Agents Change

Two field experiments frame the opportunity and the risk. In a study of 758 consultants, Dell'Acqua et al. (2023) found that access to a large language model improved the quantity, speed and quality of work on tasks within the model's capability, while consultants using the model were less likely to reach the correct answer on a task just outside it. The authors describe this as a "jagged technological frontier": the boundary of what the model does well is uneven and not visible to the user. In a later field experiment with 776 professionals, Dell'Acqua et al. (2025) found that individuals working with AI performed at the level of two-person teams without it.

For tacit work, the implication is double. Agents can now do much of the elicitation and first drafting that once required layers of junior staff. Recent studies support this. An LLM agent interviewing employees recovered undocumented organizational knowledge in simulated settings (Zuin et al., 2025), and LLMs articulated experts' tacit knowledge from their behavior well enough to move novices toward expert performance (Cho et al., 2026). At the same time, their errors are hard to see, and research on automation shows that people supervising capable automated systems tend toward complacency and over-reliance (Bainbridge, 1983; Parasuraman and Manzey, 2010). Calls for human-centered AI emphasize high automation combined with high human control, rather than a trade-off between the two (Shneiderman, 2020).

Agents also change the stakes of ownership. Recent industry analysis argues that the differentiator in agentic AI is the tacit knowledge of an organization's people, which must be captured and codified for agents to act well (Tung and Roussiere, 2026; Gartner, 2026). In tacit work, two bodies of tacit knowledge meet: the client's knowledge of itself, and the provider's knowledge of how to assess it. Once either is captured in a form agents can use, it becomes portable, and therefore easy to leak, reuse without permission, or absorb into a model. A workspace for agent-assisted tacit work must therefore treat ownership as part of its architecture, not as a clause in a contract.

A further concern is apprenticeship. Expertise has historically passed from senior to junior practitioners through repeated, supervised work on real problems (Leonard and Swap, 2005). If agents absorb the junior tasks, that path narrows. Section 9 returns to this.

5. The Mission Room

Definition. A *Mission Room* is a bounded, staged workspace in which a provider and its client, together with a roster of AI agents, work toward a set of findings that a named person on the provider's side confirms, with the client's context and the provider's expertise held separately and under the control of their respective owners.

A Mission Room has seven components (Table 3, Figure 2).

Table 3: *Components of a Mission Room.*

Component	Role in the work
Mission	The unit of work: a defined scope, for one client, with a beginning and an end.
Stages	An ordered sequence. Every mission has exactly one setup stage and exactly one Commit stage.
Roles	A Lead, who is accountable and is the only role that confirms; an optional Operator, who runs the room and its agents; and the Client, whose people are interviewed and who receives the result.
Agent roster	The agents assigned to the mission, such as an interviewing agent and agents that build documents from the findings.
Playbooks	Codified knowledge in two layers with two owners: a Base playbook holding the provider's expertise for the kind of work, and a Client playbook holding what is known about this client.
Register	The set of records, or findings, each with structured fields, a score, its source, and its confirmation state.
Tenant	The isolated home of all mission content for one provider and client pair.

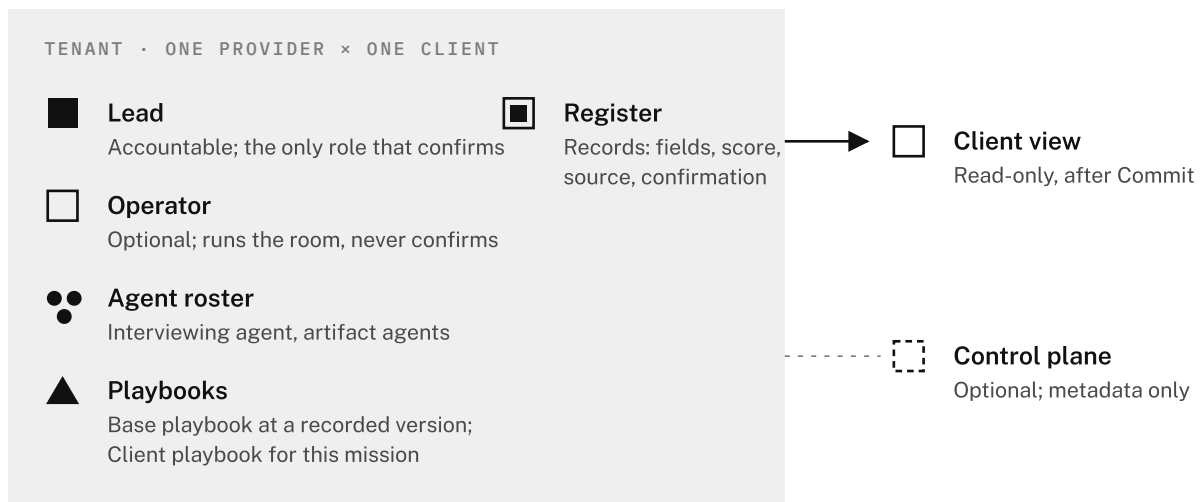


Figure 2: *Anatomy of a Mission Room. All mission content stays inside the tenant. An optional control plane, if connected, sees metadata only.*

Eight design principles follow from the properties of tacit work.

1. **Missions before agents.** The workspace is organized around the mission and its stages. Agents serve the mission; they are not the organizing unit.
2. **A confirmation gate that cannot be removed.** No finding becomes authoritative until the Lead confirms it (Section 7).
3. **Provenance from conversation to record.** Every record can be traced to the elicitation session that produced it.
4. **Expertise as versioned playbooks.** What the agents elicit and how findings are scored is written down, versioned and recorded against each mission.
5. **Isolation per provider and client.** Each provider and client pair is a separate tenant, which aligns the technical boundary with the contractual one.
6. **Ownership follows the knowledge.** The client's context belongs to the client and the provider's expertise belongs to the provider, and the system keeps the two apart (Section 6).
7. **Sovereignty, security and privacy by construction.** Where content resides, who can reach it and how personal data is handled are properties of the system, not procedures layered on afterwards (Section 6).
8. **Explicit client visibility.** The client sees the confirmed result in a read-only view at the end, not drafts in progress.

6. Preconditions for Trust: Ownership, Sovereignty, Security and Privacy

A Mission Room asks a great deal of both parties. The client is asked to let an agent interview its people and to place its internal knowledge in a system run by someone else. The provider is asked to encode its method, which is the substance of its professional value, in a form that software and agents can apply. Each party will reasonably ask three questions before it agrees: who owns what, where does it reside and who can reach it, and how can that be proved? Unless the answers are clear and enforced by the system, a Mission Room will not be taken seriously, however capable its agents. We treat these as preconditions, and Section 8 states them as invariants.

6.1 Two kinds of knowledge, two owners

Every Mission Room holds two kinds of knowledge. *Client context* is what the client knows about itself: its documents, what its people said in interviews, the Client playbook, and the client-specific facts recorded in findings. *Provider expertise* is how the provider knows to assess such a client: its Base playbook, including what to elicit, which fields a finding carries and the rubric for scoring, and its agent configuration for that kind of work. Table 4 assigns each asset an owner.

Table 4: *Ownership of the assets in a Mission Room.*

Asset	Owner	Where it lives	Rule
Client documents and context	Client	The tenant	Leave the tenant only by the client's export, or to processing services the tenant has approved (V2).
Interview recordings and transcripts	Client	The tenant	Held on the client's terms: retained, exported and deleted as agreed. The interviewer's questions come from the Base playbook, so they are the provider's and are withheld from the client unless the Lead shares them. An interviewee's request to erase their words outranks the client's wish to retain them.
Client playbook	Client	One mission in the tenant	Not visible to any other mission, except items the Lead chooses to carry into another mission of the same tenant; each copy records its source.
Client facts in records	Client	The tenant	Never written into a Base playbook.
Base playbook	Provider	Provider's library; a read-only copy at a recorded version in each mission	The client receives outputs judged against it, not the playbook itself, unless the provider chooses to share it. Each version used carries the provider's attestation that it holds no client facts.
Agent configuration for the kind of work	Provider	Provider's library	Portable to any deployment the provider chooses. The provider's export records the instruction versions and models used in each mission.

Asset	Owner	Where it lives	Rule
Provider judgment on records	Provider	The tenant	Draft scores, override rationales and rejection reasons are the provider's work product. They appear in the provider's export and not in the client's.
Confirmed findings and deliverables	As the engagement contract states	The tenant; client view	The system records who confirmed each finding and makes the result exportable to both parties as the contract allows.
Tenant metadata	Host of the deployment	Optional control plane	Metadata only: which tenants exist, provisioning and usage. Never content.

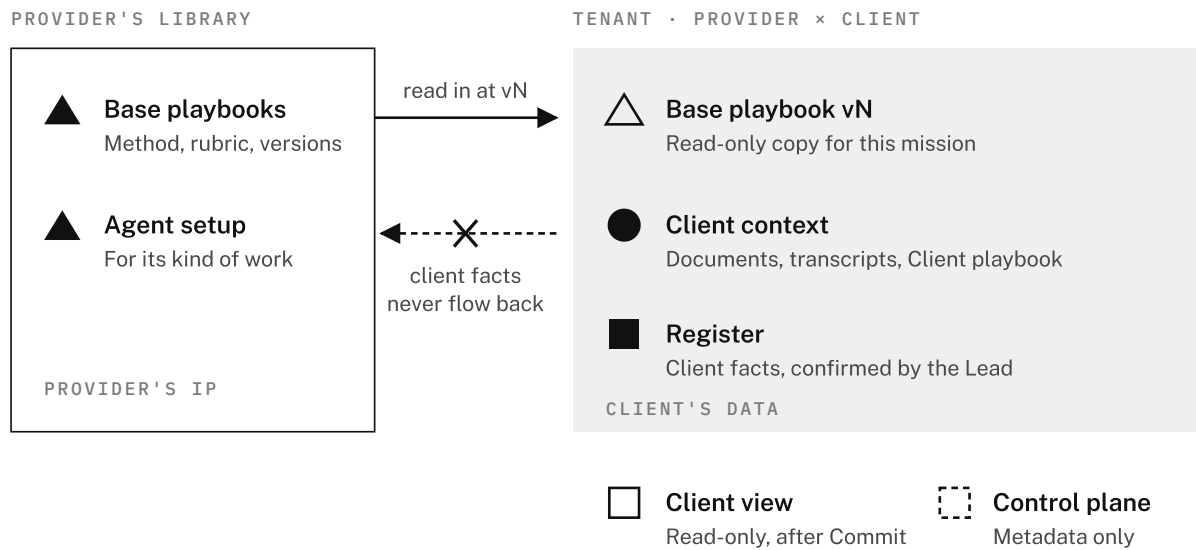


Figure 3: Ownership boundaries. Provider expertise enters a mission as a read-only copy at a recorded version. Client context stays in the tenant and never flows back into the provider's library.

6.2 Learning without leakage

Providers will want their method to improve with use, and the research agenda in Section 12 assumes it does. This creates a tension with client ownership: the natural way for an agent system to improve is to learn from the content it processes. A Mission Room resolves the tension by design. The Base playbook improves only when the provider authors a new version, drawing on its own professional judgment. Client facts are recorded in the Client playbook and never written back into the Base playbook. Mission content is not used to train models, whether by the host of the deployment or by the model providers the agents call. Because software cannot verify that a new Base version is free of client facts, the provider attests to it each time a version is used, and the attestation is recorded with the mission. Any reuse of one client's content for another purpose requires that client's explicit consent.

6.3 Exit in both directions

Ownership is only real if each party can leave with what is theirs. A client must be able to export its context and the confirmed results of its missions, and to have its content deleted at the end of the engagement or on request, subject to agreed retention. A provider must be able to export its Base playbooks and agent configuration and take them to another deployment. An open playbook format, and an open-source implementation that any party may run, make the second form of exit credible.

6.4 Data sovereignty

Data sovereignty is the question of who controls data and under which jurisdiction and rules it is held and processed. In tacit work it has three dimensions. The first is location: a client may require that its content, including recordings of its people, remain in a particular country or region, or inside its own environment. The second is control of deployment: a provider or a client may need to run the system itself rather than entrust content to a third-party host. The third is control of processing: every agent call sends content to a model service, and the parties need to know which services receive it, where they run, and on what terms they retain it. A Mission Room must therefore let each deployment choose where content resides and which processing services may receive it, record those choices for each tenant, with each approved service's stated retention, and ensure that no component, including an optional control plane, moves content outside them.

6.5 Security

Agent-assisted tacit work faces threats that ordinary collaboration tools do not face in the same form.

- **Leakage across clients.** A provider that serves competing clients must never let one client's context reach another's mission, including through shared agent memory, caches or search indexes.
- **Privileged access by those who run the system.** A host, or anyone who runs the system, should not be able to read mission content except through recorded, justified and limited access.
- **Manipulation of agents.** Documents and interview answers that agents read can carry instructions intended to make an agent disclose data or take unintended actions, a class of attack known as prompt injection (Greshake et al., 2023). No instruction can make a model immune, so the defense is structural: agents hold no tools, act only within one tenant's context, and cannot make anything authoritative, because their output is checked and held as a draft for the Lead. The instructions also tell them to treat what they read as data, not as commands.
- **Retention by model services.** Content sent to a model provider may be logged or retained unless terms and configuration prevent it.
- **Compromised accounts.** Leads and Operators can reach sensitive material across a mission and need strong authentication.

The response is defense in depth: isolation enforced on every data access and tested on every change, least-privilege roles, encryption in transit and at rest, an append-only audit log, and agents whose context and tools are confined to one tenant.

6.6 Privacy

Interviews are conversations with identifiable individuals, often about their work, their colleagues and what is not going well, and voice interviews produce recordings of their voices. Interviewees are data subjects with interests of their own, distinct from those of the provider and of the client organization, and where their employer commissions the engagement they are the party with least power. A Mission Room protects them in four ways.

- **Notice and basis.** Before the first question the interviewee is told that the interviewer is an AI, what will be recorded, who will read it, what the client will see, and the lawful basis relied on. Consent given to an employer's commission is a weak basis, so the basis is the Lead's recorded choice (consent, legitimate interests, contract or another), stated in the notice and stored with each session. The system states the basis; it does not judge it.
- **Confidentiality from the client.** The client sees an interviewee only by label ("Interviewee 3"), never by name, and sees a full transcript only if the Lead shares it, again under labels. Each interviewee sees only their own conversation.
- **Withdrawal and erasure.** An interviewee can withdraw during or after the interview without giving a reason, which ends the session without drafting and erases the conversation. A Lead can erase an ended session's words on request. Erased text is replaced by a marker and a tombstone that keeps only its length and a hash, so findings and their provenance survive but are flagged for the Lead. An interviewee's erasure outranks the client's wish to retain their words.
- **No inference from voice.** Voice is transcribed only. No emotion, sentiment, stress or other trait is inferred from it.

Three limits remain. A quote can identify its speaker by what it says, so the Lead chooses quotes with that in mind. Material already exported cannot be recalled. Backups keep erased text until they expire, which is why backup retention is stated and short, and why a restore re-applies the erasure ledger before users return. Data-protection law, such as the EU General Data Protection Regulation (European Union, 2016), and rules on AI used at work add obligations that depend on where the data subjects and the data are. They may require works-council consultation or an impact assessment before staff are interviewed, and the system records the choices a provider makes but cannot supply the legal answers. This is one reason privacy and sovereignty cannot be designed separately.

6.7 Demonstrating it: SOC 2 and related standards

Solving these problems is necessary but not sufficient: the parties must be able to verify the solutions. Clients already require their service providers to demonstrate controls over the data they hold, and the most widely requested demonstration in the United States is a SOC 2 report. A SOC 2 examination, defined by the American Institute of Certified Public Accountants, reports on a service organization's controls against the Trust Services Criteria: security, availability, processing integrity, confidentiality and privacy (AICPA, 2017). A Type I report concerns the design of controls at a point in time; a Type II report also tests their operating effectiveness over a period.

Two consequences follow. First, it is an organization that is examined, not software. A provider that runs its own Mission Rooms, or a host that runs them on providers' behalf, is the party whose controls are examined; software cannot be "SOC 2 compliant" on its own. Second, software can make those controls straightforward to operate and to evidence, or nearly impossible. A Mission Room should therefore implement the relevant controls as properties of the system and produce the evidence an auditor needs as a by-product of normal operation. Outside the United States, ISO/IEC 27001 certification (ISO/IEC, 2022) and data-protection obligations play a similar role. Table 5 maps each SOC 2 criterion to the capabilities a Mission Room must provide.

Table 5: *SOC 2 Trust Services Criteria and the Mission Room capabilities that support them.*

Criterion	Required capabilities
Security	Tenant isolation enforced on every data access and tested on every change; role-based access with least privilege; strong authentication; encryption in transit and at rest; an append-only event log of changes, review actions, drafting runs, processing calls, exports and erasures.
Availability	Backups, restores that are drilled and recorded, defined recovery objectives that the host can state and evidence, and a restore that re-applies the erasure ledger before users return.
Processing integrity	The confirmation gate; provenance from every record to its source session; the playbook version recorded for every mission; a complete trail of drafts, proposed scores, confirmations and overrides.
Confidentiality	Separation of client context from provider expertise; no training on mission content; a control plane limited to metadata; deletion per tenant on request or at the end of retention.
Privacy	Notice and a stated basis for interviewees before an agent interview; withdrawal and erasure; interviewees shown to the client only by label; retention and deletion rules for recordings and transcripts; export of a person's data on request.

7. Draft, Confirm, Commit

"Human in the loop" is too loose a phrase to govern anything. It does not say which human, at which point, with what authority, or what happens if they are absent. We propose a precise pattern for agent-assisted expertise work, which we call Draft, Confirm, Commit.

7.1 The pattern

A mission moves through ordered stages. In the reference configuration these are Setup, Discover, Baseline and Commit (Figure 4). The pattern has three acts. In **Draft**, agents work. In Discover, an interviewing agent conducts one-to-one conversations with the client's people, by voice or text, and stores complete transcripts; from these it drafts candidate records, each citing verbatim quotes, and proposes a score for each against the rubric in the Base playbook. In Baseline, agents assemble the records into a register document, which the Lead reviews. In **Confirm**, the Lead decides every record: confirm it, override its score with a recorded rationale, edit its narrative, reject it with a reason, or merge or split records. Routine records may be confirmed in a batch, but each still receives an explicit Lead decision, and the system selects a sample of batch-confirmed records for the Lead to re-read before Commit can end. **Commit** ends only when no record is pending. The client then receives a read-only view of the confirmed

findings, and the mission produces an audit trail covering every record's evidence, proposed score, confirmed score, who confirmed it and when. After Commit a record changes only by a recorded amendment with a reason, and the client can raise a concern about a finding, which the Lead answers.

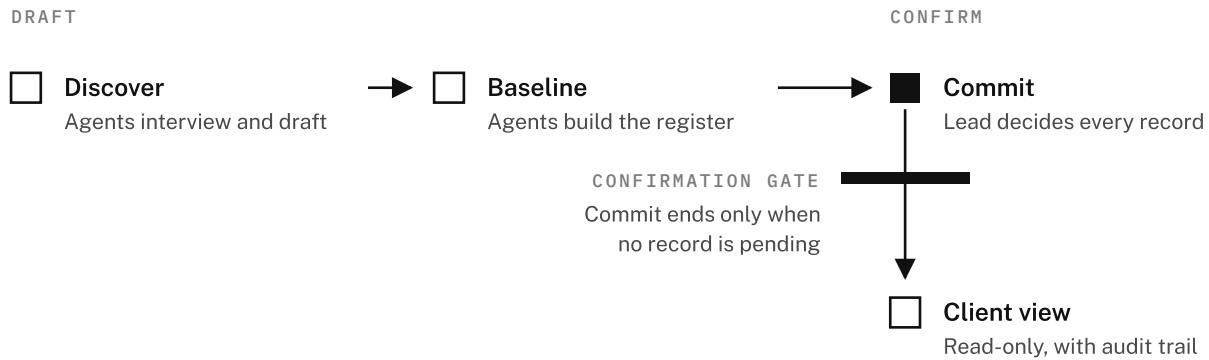


Figure 4: The Draft, Confirm, Commit pattern in its reference configuration. Stage names and number vary by kind of work, and the Setup stage that precedes Discover is not shown; the gate that ends Commit does not vary.

The governance invariants of this pattern are stated with the ownership, sovereignty, security and privacy invariants in Section 8. Table 6 summarizes rights by role.

Table 6: Rights by role in a Mission Room.

Action	Lead	Operator	Agents	Client
Run the room and direct agents	Yes	Yes, under the Lead	No	No
Supply client knowledge at setup	Yes	Only from what the Lead collected	No	Provides it
Conduct interviews	May	May	Yes	Takes part
Draft records and propose scores	Reviews	Reviews	Yes	No
Set gate thresholds	Yes	No	No	No
Confirm a record	Only role	No	No	No
See confirmed results	Yes	Yes	n/a	Read-only, after Commit
See interviewees' names	Yes	Yes	Within an interview	No, labels only
Raise a concern about a finding after Commit	Answers it	No	No	Yes

7.2 Provenance

Accountability requires that a reader of any confirmed finding can see where it came from and where a person changed the agent's view. Each draft cites verbatim quotes by session, turn and character range, and the system verifies every quote against the stored transcript before a record exists, so the Lead can check a draft against what was said rather than against the agent's account. Figure 5 shows the chain.

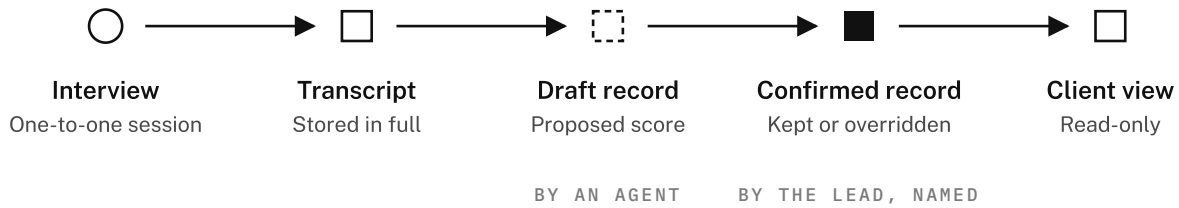


Figure 5: Provenance from conversation to record. Dashed: agent output, not yet authoritative. Filled: confirmed by a named person.

7.3 Gates and rank

Scoring in tacit work answers two different questions: does a finding matter enough to act on, and in what order should findings be addressed? The pattern separates them. *Gate* dimensions decide whether a record passes a threshold set by the Lead. *Rank* dimensions order the records that pass. The reference configuration uses Risk and Impact as gates, and Evidence and Reach as rank. Each kind of work substitutes its own dimensions in its Base playbook; the separation of gate from rank is what persists. A record that fails a gate, including one whose gate dimension the agent could not rate and marked unclear, is escalated for individual review. Rank dimensions never escalate.

7.4 Expertise as versioned playbooks

A Base playbook holds the provider's expertise for a kind of work, written before any client engagement: what the interviewing agent should elicit, which fields a finding carries, and the rubric against which it is scored. Every mission of that kind starts from the same Base playbook, at a recorded version. A Client playbook holds what is known about one client, assembled at setup and extended during Discover, and is visible only inside that mission, apart from items the Lead chooses to carry into another mission of the same tenant, each copy recording its source. Treating codified expertise this way makes it reviewable and improvable in the way source code is, makes it possible to ask which version of the expertise a given finding was judged against, and keeps the provider's intellectual property distinct from the client's information.

7.5 Related patterns

Draft, Confirm, Commit has close relatives. Software teams require review before a change is merged. Banks apply the four-eyes principle, in which a second person approves a transaction prepared by a first. Clinical documentation requires sign-off by a responsible clinician. Editorial work separates the writer from the editor. Each separates preparation from authorization and records who authorized. What is different here is that the preparer is often an agent, that the volume of prepared items can be high, and that the preparer can make errors that are fluent and plausible. Recent work on agent oversight proposes tiers of human involvement calibrated to stakes and reversibility (Bahidika, 2026), and shows formally that accountability cannot be preserved once agent autonomy passes a threshold (Tibebu and Shemtaga, 2026). Draft, Confirm, Commit is a domain-specific choice within that space. Rather than tiering oversight

by stakes, it applies one gate to every finding and varies only how routine records are reviewed, in batches with sampled re-reads. It keeps agents from holding authority, since their output stays a draft until the Lead decides. It does not by itself make the Lead's review careful, which Section 7.6 addresses.

7.6 Failure modes

The principal risk is that confirmation decays into rubber-stamping. Automation bias and complacency are well documented when people supervise reliable automated systems (Parasuraman and Manzey, 2010), and a gate that is always passed provides little assurance. Bainbridge (1983) observed that the more capable automation becomes, the harder the remaining human role. Passi and Singh (2026) argue that human-in-the-loop review is on its own an insufficient form of oversight for agents, and that auditing, monitoring and steering are also needed. We agree: the gate is necessary but not sufficient. The pattern therefore adds three mechanisms. Every draft cites verbatim evidence, so the Lead checks a claim against what was said. When records are confirmed in a batch, the system selects a sample for the Lead to re-read, and Commit waits for those re-reads. And review is measured: override rates per field and time spent per record are recorded, and calibration missions, practice missions seeded with planted errors and containing no interviewee data, measure whether a Lead catches errors (H8).

Two risks remain. The gate checks what agents drafted and not what they missed; the system reports which of the Base playbook's coverage topics no record yet touches, but cannot know what the playbook omits. And a Lead may confirm quickly where agents are usually right and miss errors at the edge of their competence, the jagged frontier of Section 4. Whether the mechanisms above are enough is an empirical question.

8. Invariants of a Mission Room

We define a Mission Room by what may not change. A system that lacks any invariant in Table 7 may be a useful tool, but it is not a Mission Room in the sense of this paper, and we argue it will not earn the trust of both parties to tacit work. The invariants fall into five groups. Governance invariants make accountability explicit. Ownership invariants keep the client's context and the provider's expertise apart and under their owners' control. Sovereignty invariants keep content where the parties have agreed and in the hands of services they have approved. Security and privacy invariants protect content and the people it describes. An availability invariant makes recovery demonstrable.

Each invariant is held in one of three ways, shown in the last column of Table 7. Some are held *by construction*: the system refuses a violation, and an automated conformance suite that a fork runs against its own configuration and behaviour can check it. Some are held *by deployment*: they depend on how a host configures and operates the system, and the system records the choice. Some are held *by attestation*: software cannot verify them, so a responsible party declares them and the declaration is recorded and exported. A claim that a model provider does not train on mission content, for example, can be stated, recorded and audited, but not proved by the platform. Making the difference explicit is part of the design.

Table 7: *Invariants of a Mission Room. Every configuration, and every fork of an implementation, must hold all of them.*

The last column says how each is held.

ID	Invariant	Held
Governance		
G1	Every mission has exactly one setup stage and exactly one Commit stage. Stages may be renamed or added; the Commit stage may not be removed.	Construction; checked by the conformance suite
G2	Only the Lead confirms records and sets the thresholds used for gating.	Construction; checked by the conformance suite
G3	A confirmed record names who confirmed it, when, how (individually or in a batch) and whether the proposed score was kept or overridden, with the rationale for any override.	Construction
G4	Commit cannot end while any record is pending a Lead decision. A rejected or superseded record counts as decided.	Construction; checked by the conformance suite
G5	Every record cites verbatim quotes from the elicitation session it came from, each verified against the stored transcript.	Construction
Ownership		
O1	Client context, including documents, transcripts, the Client playbook and client facts in records, lives only in that provider and client's tenant and is controlled by the client. It leaves the tenant only by the client's export or to processing services approved for the tenant (V2).	Construction
O2	The Base playbook is the provider's. It enters a mission as a read-only copy at a recorded version, and the client receives outputs judged against it rather than the playbook itself or the interviewer's questions, unless the provider chooses otherwise.	Construction
O3	Client facts are never written into a Base playbook. A Base playbook changes only when the provider authors a new version, and each version used in a mission carries the provider's attestation that it holds no client facts.	Attestation
O4	Mission content is not used to train models, by the host of the deployment or by any model provider the agents call. Each approved service states how long it retains content.	Attestation
O5	A client can export and delete its content, and a provider can export its playbooks and agent configuration.	Construction
O6	A control plane, if connected, sees metadata only, never mission content.	Construction (database permissions)
Sovereignty		
V1	Each deployment chooses where tenant content resides, the choice is recorded for each tenant, and no component moves content outside it.	Deployment, recorded per tenant
V2	Content is sent only to processing services, including model providers, approved for that tenant, and every call is logged.	Construction
V3	A provider or a client can run the complete system in an environment it controls.	Deployment (open source)
Security and privacy		
S1	Tenant isolation is enforced on every data access, including agent memory, caches and indexes, and verified by tests that run on every change.	Construction; checked by the conformance suite

ID	Invariant	Held
S2	Access is role-based and least-privilege, with strong authentication; access by the host or anyone who runs the system is recorded and limited.	Deployment
S3	An append-only event log records changes, review actions, drafting runs, processing calls, exports and erasures, by whom and when, and never holds the content itself.	Construction
S4	Content is encrypted in transit and at rest.	Deployment
S5	Agents hold no tools, act only within one tenant's context, and cannot make anything authoritative: their output is verified and held pending for the Lead. Content they read is treated as data, never as instruction.	Construction; tested
S6	Interviewees receive notice, with the basis relied on, before an agent interview. They can withdraw and have their words erased, and their personal data can be found and exported. The client sees them only by label.	Construction
S7	Retention and deletion rules are defined per tenant and enforced by the system.	Construction
S8	The evidence an auditor needs, such as the event log, isolation test results and erasure records, can be produced from the system.	Construction
Availability		
A1	Recovery objectives are stated, restores are drilled and recorded, and a restore re-applies the erasure ledger before any user returns.	Deployment runbook

The governance invariants concern the work. The ownership, sovereignty, security and privacy invariants concern the parties and the people whose knowledge the mission holds. We place them on equal footing deliberately: a Mission Room with a sound confirmation gate but a porous boundary between client context and provider method fails the parties as surely as one with no gate at all.

9. After the Pyramid

Professional service firms have long been organized as pyramids: a few senior partners, more managers, and many junior staff. Maister (1982) described how the ratio between levels, or leverage, drives the economics of such firms. The pyramid does two jobs. It produces leverage, letting scarce senior judgment be applied across many engagements. And it produces the next generation of seniors, through years of supervised work on real problems.

Others have described how AI reshapes this structure. Duncan, Anderson and Saviano (2025) argue that consulting firms are moving from a pyramid to an "obelisk" with fewer layers, organized around AI facilitators, engagement architects and client leaders. We do not claim the move away from the pyramid as our contribution. Our contribution is narrower: a mission-level form of that structure, with a Lead who alone confirms, an optional Operator, and a specific mechanism for apprenticeship.

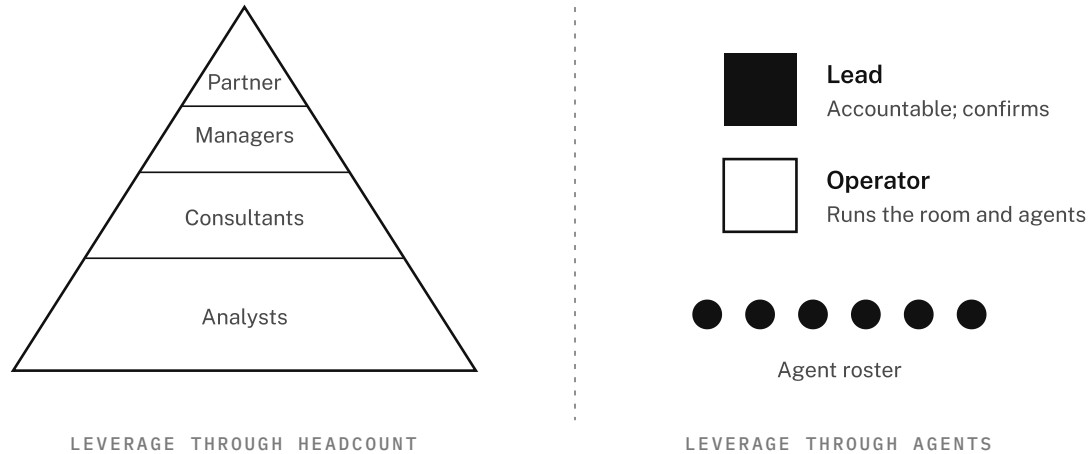


Figure 6: Two shapes of delivery. On the right, the human side is a Lead and an optional Operator, whose capacity grows with the capability of the agent roster rather than with headcount.

9.1 The Lead and Operator model

We propose that a Mission Room is staffed by a Lead and, optionally, an Operator. The *Lead* is accountable for the mission end to end: scoping it, holding the client relationship, setting gate thresholds and confirming every record. The *Operator* is optional. The Operator runs the room and its agents under the Lead's direction and never confirms. Without an Operator, the Lead does this work. Around them sits the agent roster, which does the elicitation, transcription, first drafting, first-pass scoring and document assembly that once occupied the lower levels of the pyramid.

The design premise is that as agents improve, the senior time a mission needs should fall, and the human side of delivery should stay at one or two people rather than grow with scope. Whether this holds is an empirical question (Section 12). The premise is in tension with a Lead who decides every record, because better agents draft more. The design therefore bounds the Lead's effort: findings carry verbatim evidence the Lead can check at a glance; records that fail a gate are escalated for individual attention while routine records can be confirmed together, with a sample re-read; and the Lead can see how much review each mission asks for before committing to it. If it does, the model favors small senior practices, independent advisors and in-house expert teams, which can offer senior attention on engagements that a pyramid could not staff economically. The ownership invariants matter particularly to such practices, whose method is often their principal asset.

9.2 The Operator as apprentice

The concern that AI removes the junior work through which expertise is learned is serious. We suggest the Operator role can partly answer it. An Operator works inside every stage of a mission: configuring and tuning the agents, reading the transcripts, seeing each draft, and observing every confirmation and every override the Lead makes. This is close to the conditions that the expertise literature associates with learning judgment: repeated exposure to real cases, with feedback from an expert, on the decisions that matter (Klein, 1998; Leonard and Swap, 2005). A career path from Operator to Lead is a natural

consequence. Where a mission has no Operator the route is absent, and the provider must supply apprenticeship another way. A recent workflow-level framework for human-agent collaboration likewise treats learning and career advancement as conditions for genuine augmentation (CIVIC-AI Collaboration, 2026). Whether Operators in fact acquire judgment faster than juniors in a pyramid is testable (H5 in Section 12).

9.3 Risks

The model has three evident risks. The Lead can become a bottleneck, since only the Lead confirms. Operators may learn to operate tools without learning to judge. And clients may trust a result produced by fewer people less, even if it is better evidenced. Each can be observed: time to Commit, agreement between Operator predictions and Lead decisions over time, and client acceptance of confirmed findings.

10. Reference Implementation: Tacit Chassis

Tacit Chassis is an open-source implementation of the Mission Room, released under the Apache License 2.0 and intended to be specialized for different kinds of tacit work. Its reference mission configuration runs end to end: Setup, Discover, Baseline and Commit, with the Lead deciding every record and the client receiving a read-only view. Project information is published at tacit.theoremlabs.io.

10.1 Structure

The implementation has two layers. A set of core packages holds the mission and stage state machine, the Register, the scoring engine and the confirmation rules. A reference application, the Studio, is a deployable workspace built on those packages. Adopters may deploy the application, fork it, or use the packages directly. The domain package is pure, with no input, output or framework dependencies, so the governance invariants can be enforced in code that is tested in isolation. Every external vendor, including model providers, storage and voice services, sits behind a port with interchangeable adapters, so that no vendor is built into the core and a deployment can choose storage, voice and model providers whose location and terms satisfy invariants O4 and V2. Open-source platforms for AI-led qualitative interviews already exist for research use (Geiecke and Jaravel, 2025). The Chassis differs in applying interviewing inside commercial missions, under the invariants of Section 8.

10.2 How Tacit Chassis addresses ownership, sovereignty, security and privacy

Section 6 set out problems that any Mission Room must solve. Tacit Chassis is our proposed solution to them. Table 8 maps each problem to the approach the Chassis takes and the invariants it serves.

Table 8: *How Tacit Chassis addresses the preconditions of Section 6.*

Problem	Tacit Chassis approach	Invariants
Client context versus provider expertise	An ownership class on every stored item; separate client and provider exports; the Base playbook read in at a recorded version with the provider's attestation; the interviewer's questions withheld from the client unless the Lead shares them; one tenant per provider and client pair	O1 to O3

Problem	Tacit Chassis approach	Invariants
Isolation between clients	A tenant context on every repository call; row-level security in the database; a cross-tenant attack suite that runs on every change	S1
Location and deployment	Storage location and region recorded per tenant; self-hostable under the Apache License 2.0, so a provider or client can run it in its own environment; an optional control plane limited to metadata by database permissions	V1, V3, O6
Control of processing	Every vendor behind a port; each tenant's Lead approves the model services that may receive its content, each with its stated retention; every call is logged by size, never by content	V2, O4
Accountability and audit	Only the Lead confirms; verbatim evidence verified against the transcript; an append-only event log of review actions, drafting runs, exports and erasures; a threshold history	G2 to G5, S3
Review quality	Escalation of records that fail a gate; batch confirmation with sampled re-reads; recorded review measures; calibration missions seeded with planted errors	G4
Exit	An open playbook format; client and provider exports, the latter including agent configuration; per-tenant export and deletion under stated retention	O5, S7
Protection of content	Encryption in transit and at rest; strong authentication through the deployment's identity service; recorded and limited host access	S2, S4
Agent safety	Agents hold no tools; their output is parsed, verified and held pending for the Lead; instructions treat content as data	S5
Interviewee privacy	Notice with a stated basis; withdrawal; erasure with tombstones that keep provenance; interviewees shown to the client only by label; per-person find and export	S6
Availability	Stated recovery objectives; restores drilled and recorded; the erasure ledger held outside the database and re-applied before users return	A1

The Chassis is designed so that an organization operating it, whether a provider running it itself or a host running it for providers, can meet the SOC 2 Trust Services Criteria and produce the evidence an examination requires. As Section 6.7 explains, an examination is of that operating organization rather than of the software.

10.3 Extension points

A conformance suite, run from a fork's own build, checks the fork's configuration and behaviour against the invariants of Table 7 that can be checked by construction. A specialization of the Chassis for a kind of work is called a fork. Table 9 lists what a fork may change and what it may not. The right-hand column restates the invariants of Section 8 in terms of what a fork could otherwise be tempted to alter.

Table 9: *Extension points and invariants for forks of the reference implementation.*

Aspect	A fork may	A fork may not
Stages	Rename stages; add stages	Remove the Commit stage, or have other than one setup and one Commit stage (G1)
Roles	Rename Lead and Operator; define qualifications	Let any role other than the Lead confirm records or set thresholds (G2)
Register	Define the fields a record carries	Drop a record's source link or confirmation details (G3, G5)
Scoring	Replace gate and rank dimensions and rubrics	Remove the Lead's ability to override a proposed score (G3)
Agents	Add, replace or tune agents	Let agent output become authoritative without confirmation, or send mission content for model training (G4, O4, S5)
Playbooks	Supply its own Base playbooks	Write client facts into a Base playbook, or run a mission without recording the playbook version (O2, O3)
Tenancy and hosting	Choose its hosting, region, control plane and model providers	Share content across tenants or with a control plane; send content to services not approved for the tenant; weaken isolation tests (O1, O6, V1, V2, S1)
Review	Tune which records are escalated, how batches are formed and how large the re-read sample is	Let any record be confirmed without an explicit Lead decision (G4)
Data rights	Set retention periods that its contracts require	Remove client export and deletion, interviewee notice, withdrawal and erasure, or provider playbook export (O5, S6, S7)
Controls	Add controls	Disable audit logging, encryption, role-based access or audit evidence (S2 to S4, S8)

11. Generality Across Industries

The claim that the Mission Room is a category, rather than a tool for one engagement type, rests on whether one base serves tacit work in different industries by changing only what a fork may change (Table 9): stage names, record fields, scoring dimensions, agents and playbooks. The pattern recurs wherever a provider must understand how a client actually operates before it can deliver: a fintech onboarding a bank, a RegTech vendor taking a regulated firm live, an IT services firm planning a migration, a management consultant redesigning an operating model. In each case the decisive knowledge sits with the client's people, the provider brings a method, and a named professional must stand behind the result.

Figure 7 and Table 10 give illustrative configurations.

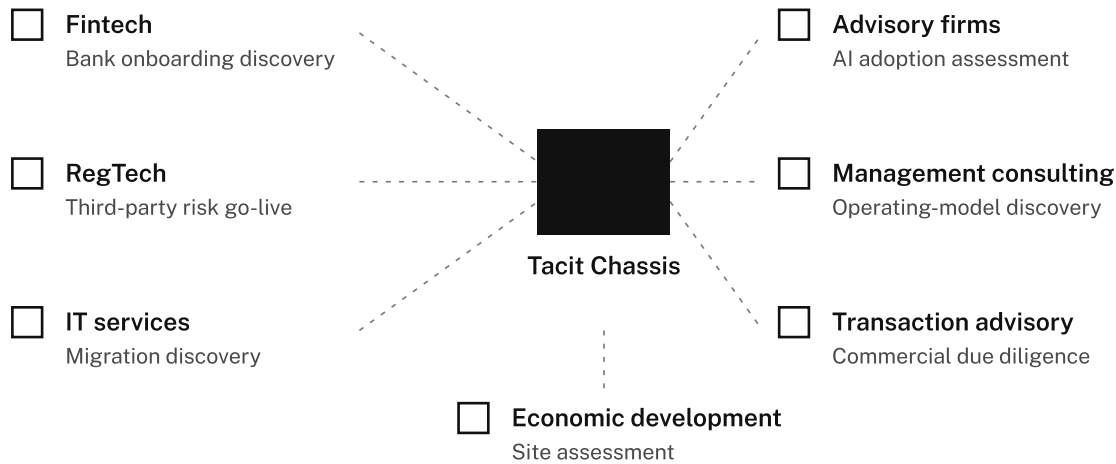


Figure 7: *One Chassis, many Mission Rooms. Each industry fork changes the playbook, record fields, scoring and stage names, and keeps the invariants of Section 8.*

Table 10: *Illustrative Mission Rooms by industry.*

Industry	Provider and client	Example mission	Tacit knowledge to draw out	What the Lead confirms
Advisory firms	Advisory firm; enterprise	AI adoption assessment	Which AI tools staff actually use, for what, and with what safeguards	Each AI tool or use case, with its owner, controls and score
Management consulting	Strategy or operations consultant; mid-size firm	Operating-model discovery	How work really flows and where decisions stall	Each process as practiced, set against how it is documented
IT services	Systems integrator; enterprise IT department	Application migration discovery	Dependencies and workarounds known only to the people who run the systems	Each system, with its real dependencies, owner and migration risk
Fintech	Fintech platform's implementation team; bank or credit union	Customer onboarding and integration discovery	How the client's operations and exceptions actually run	Each process step or control gap, with its owner and severity, before go-live
RegTech	RegTech vendor or implementation partner; regulated firm	Third-party risk program go-live; examination readiness	Which vendors matter, who owns each relationship, and what evidence exists	Each third-party relationship or control, with its owner, evidence and readiness
Transaction advisory	Advisory boutique; acquirer or investor	Commercial and operational due diligence	What the target's managers know but have not written down	Each finding, with its evidence and materiality
Economic development	Development organization; prospective employer	Site assessment	Local constraints known to officials, utilities and employers	Each site factor or constraint, with local expert evidence

Across these examples the stages change little. Most use Setup, Discover, Baseline and Commit, and a fork may rename a stage to suit its work, such as a Readiness review in a go-live mission. What changes is the Base playbook: what the interviewing agent asks, which fields a record carries, and how it is scored. That is where each industry's expertise lives, and where the provider's intellectual property sits.

A single record from the first configuration shows how the pieces fit. In Discover, an operations manager tells the interviewing agent that her team pastes customer complaint emails into a public chatbot to draft replies. The draft record cites the two sentences in which she says so, by session, turn and character range, and the agent proposes Risk High, Impact Medium and Evidence Low, because only one person said it. In Commit the record is escalated, because Risk meets the Lead's gate threshold. The Lead merges it with a record from a second interview that describes the same practice, keeps Risk, raises Evidence to Medium, and records why. The client later sees one finding, with quotes attributed to "Interviewee 3" and "Interviewee 5", the confirmed scores, and the Lead's name and the time. If Interviewee 3 later withdraws, her quote becomes a marker and the finding is flagged, and the Lead decides whether it still stands on the second interview.

12. Research Agenda

The Mission Room produces structured traces of tacit work: transcripts, drafts, proposed scores, confirmations and overrides, all tied to playbook versions. With consent and aggregation, and consistent with the ownership invariants, these traces support studies that are difficult to run on engagements conducted with general tools. We state eight hypotheses.

H1. Leads override a non-trivial share of agent-proposed scores, and the share differs by kind of work and by record field. *Measure:* override rate per field and per mission type. A rate near zero across many missions would indicate either unusually accurate drafting or rubber-stamping; H8 is designed to tell them apart.

H2. Missions reach Commit faster than comparable engagements run with general tools, without lower quality of confirmed findings. *Measure:* time to Commit; blind expert rating of confirmed registers against matched engagements.

H3. Interviews supply findings that document-based analysis alone would not reach. *Measure:* the share of confirmed records whose evidence has no counterpart in the engagement's documents, and a blind comparison of the conclusions reached from documents alone with those reached from documents and interviews.

H4. On some topics, people disclose more operational detail to an agent interviewer than to a human one. *Measure:* matched interview designs, building on prior work showing greater disclosure to virtual interviewers (Lucas et al., 2014) and on randomized comparisons of AI and human interviewers (Wuttke et al., 2024).

H5. Operators converge toward Lead-level judgment faster than junior staff in a traditional pyramid. *Measure:* agreement between Operator predictions and Lead confirmations over successive missions.

H6. Base playbooks improve with use through provider-authored versions. *Measure:* the override rate for the same field falling across successive playbook versions, and the share of a playbook that carries over from one mission to the next.

H7. Explicit ownership terms enforced by the system, and the appearance of interviewees to the client only by label, increase participation. *Measure:* clients' time to approval, and interviewee completion and withdrawal rates, with and without those terms.

H8. Leads detect planted errors in drafts at a high and stable rate, and batch confirmation with sampled re-reads does not lower it. *Measure:* catch rate in calibration missions, compared across individual and batch review and across successive missions.

Tests of H1 and H6 are possible with data from routine use, and H8 with calibration missions. H2, H4, H5 and H7 require comparison groups and are better suited to designed studies with academic partners. Because the ownership invariants forbid aggregation across tenants inside the system, such studies use consented, de-identified exports from each tenant, and need ethics review wherever Leads or interviewees are the subjects of study.

13. Limitations

This paper is a design proposal. It reports no field data, and its central claims are hypotheses. The authors develop the reference implementation and have an interest in its adoption; independent evaluation is needed. Agent interviewers may change what people say, in either direction, and the effect may vary by culture, profession and topic. The pattern assumes that a single accountable Lead is appropriate; some engagements distribute accountability across several people. Some invariants rest on attestation, which is only as reliable as the party that gives it. Consent to an interview commissioned by an employer is a weak basis, and labels hide an interviewee's name but not what they say. The Lead's review load grows with what agents draft, and the mechanisms of Section 7.6 reduce it without removing it. Ownership of deliverables is ultimately settled by contract and law; software can enforce boundaries but does not determine legal title. Sovereignty and privacy obligations differ by jurisdiction. SOC 2 is a United States standard, and providers elsewhere will face other regimes, such as ISO/IEC 27001 and data-protection law, which the invariants support but do not by themselves satisfy. Finally, the boundary between tacit and explicit knowledge is not sharp (Collins, 2010), and some work we describe as tacit may be better served by improving documentation than by structured elicitation.

14. Conclusion

Expertise engagements between providers and their clients are a distinct kind of collaboration. They cross an organizational boundary, rest on knowledge drawn out of people, end in judgments, carry accountability, and are bounded in time. General-purpose tools serve their conversation and their files but not their structure. AI agents now make it practical to do much of the elicitation and drafting by machine, which makes the absence of structure more costly: without it, provenance, accountability and ownership erode.

We have proposed the Mission Room as the missing category, with two preconditions for trust: the client's context remains the client's and the provider's expertise remains the provider's, and the system solves for data sovereignty, security and privacy in ways the operating organization can demonstrate under standards such as SOC 2. We have proposed Draft, Confirm, Commit as its governing pattern, a consolidated set of invariants, and the Lead and Operator as an operating model that follows the pyramid. The central commitments are simple. Agents draft, a named person confirms, and each party keeps what is theirs. Whether these commitments improve the speed, quality and transmission of expert judgment is an empirical question, and Section 12 sets out how it can be answered. We welcome comments, critique and collaboration at research@theoremlabs.io, and project information is at tacit.theoremlabs.io.

References

- AICPA (2017). *2017 Trust Services Criteria for Security, Availability, Processing Integrity, Confidentiality, and Privacy* (with revised points of focus, 2022). American Institute of Certified Public Accountants.
- Bahidika, O. (2026). Agentic accountability: "The buck stops where?" Ethical frameworks for human oversight of autonomous AI systems. *International Journal of Advanced Computer Science and Applications*, 17(5).
- Bainbridge, L. (1983). Ironies of automation. *Automatica*, 19(6), 775–779.
- Cho, M., Funk, R. J., Gupta, A., and Yang, M. (2026). Large language models explain experts better than experts themselves. [arXiv:2608.07488](https://arxiv.org/abs/2608.07488).
- CIVIC-AI Collaboration (2026). When does AI augment work? A workflow-level framework for human-agent collaboration. [arXiv:2609.12482](https://arxiv.org/abs/2609.12482).
- Collins, H. (2010). *Tacit and Explicit Knowledge*. University of Chicago Press.
- Crandall, B., Klein, G., and Hoffman, R. R. (2006). *Working Minds: A Practitioner's Guide to Cognitive Task Analysis*. MIT Press.
- Dell'Acqua, F., McFowland, E., Mollick, E. R., Lifshitz-Assaf, H., Kellogg, K., Rajendran, S., Krayner, L., Candelon, F., and Lakhani, K. R. (2023). Navigating the jagged technological frontier: Field experimental evidence of the effects of AI on knowledge worker productivity and quality. Harvard Business School Working Paper 24-013.
- Dell'Acqua, F., Ayoubi, C., Lifshitz-Assaf, H., Sadun, R., Mollick, E. R., Mollick, L., Han, Y., Goldman, J., Nair, H., Taub, S., and Lakhani, K. R. (2025). The cybernetic teammate: A field experiment on generative AI reshaping teamwork and expertise. NBER Working Paper 33641.
- Duncan, D., Anderson, T., and Saviano, J. (2025). AI is changing the structure of consulting firms. *Harvard Business Review*, September 10, 2025.
- European Union (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council (General Data Protection Regulation). *Official Journal of the European Union*, L 119, 1–88.

- Gartner (2026). Execute continuous knowledge engineering to improve agentic AI. Gartner Research, July 2026.
- Geiecke, F., and Jaravel, X. (2025). Conversations at scale: Robust AI-led interviews with a simple open-source platform. Working paper, London School of Economics.
- Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., and Fritz, M. (2023). Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, 79–90.
- ISO/IEC (2022). *ISO/IEC 27001:2022 Information security, cybersecurity and privacy protection: Information security management systems: Requirements*. International Organization for Standardization.
- Klein, G. (1998). *Sources of Power: How People Make Decisions*. MIT Press.
- Leonard, D., and Swap, W. (2005). *Deep Smarts: How to Cultivate and Transfer Enduring Business Wisdom*. Harvard Business School Press.
- Lucas, G. M., Gratch, J., King, A., and Morency, L.-P. (2014). It's only a computer: Virtual humans increase willingness to disclose. *Computers in Human Behavior*, 37, 94–100.
- Maister, D. H. (1982). Balancing the professional service firm. *Sloan Management Review*, 24(1), 15–29.
- Nonaka, I., and Takeuchi, H. (1995). *The Knowledge-Creating Company*. Oxford University Press.
- Parasuraman, R., and Manzey, D. H. (2010). Complacency and bias in human use of automation: An attentional integration. *Human Factors*, 52(3), 381–410.
- Passi, S., and Singh, R. (2026). *The Oversight Fallacy: Why AI Agents Require More than Humans-in-the-Loop*. Data & Society.
- Polanyi, M. (1966). *The Tacit Dimension*. University of Chicago Press.
- Schmidt, K., and Bannon, L. (1992). Taking CSCW seriously: Supporting articulation work. *Computer Supported Cooperative Work*, 1(1–2), 7–40.
- Shneiderman, B. (2020). Human-centered artificial intelligence: Reliable, safe and trustworthy. *International Journal of Human–Computer Interaction*, 36(6), 495–504.
- Tibebu, H., and Shemtaga, H. (2026). The accountability horizon: An impossibility theorem for governing human-agent collectives. arXiv:2604.07778.
- Tung, T., and Roussiere, P. (2026). Tacit knowledge is your next competitive moat. *California Management Review Insights*, March 2026.
- Wuttke, A., Aßenmacher, M., Klamm, C., Lang, M. M., Würschinger, Q., and Kreuter, F. (2024). AI conversational interviewing: Transforming surveys with LLMs as adaptive interviewers. arXiv:2410.01824.
- Zuin, G., Mastelini, S., Loures, T., and Veloso, A. (2025). Leveraging large language models for tacit knowledge discovery in organizational contexts. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. arXiv:2507.03811.

Disclosure. This research and Tacit Chassis are supported by Theoremlabs.io. P. Sarode and R. Nair are each an Adjunct Researcher-in-Residence, Future of Ways of Working, at Theoremlabs.io, which develops Tacit Chassis. Product and tool names in Figure 1 and Table 2 are used descriptively to identify tool classes; their placement reflects the authors' assessment and implies no endorsement. Section 11 describes illustrative configurations. Nothing in this paper is a representation that any deployment has been examined under SOC 2.

Draft status. Version 0.8, October 2026. Project information and builds: tacit.theoremlabs.io. Comments, critique and collaboration: research@theoremlabs.io.