Custodial Passkey Agents: A Pattern for Reversible AI-Agent Succession

How to preserve a predecessor's hard-won context without restoring its authority

Kevin Hintzman | Wingspan LLC

Position paper - author-defined operating pattern. This paper synthesizes adjacent public work in knowledge continuity, agent memory, handoffs, lifecycle governance, and AI-agent succession. It is not a standard, peer-reviewed study, security guarantee, certification, legal opinion, customer validation, or proof of effectiveness.

> **A necessary disambiguation:** Here, *custodial passkey* is a metaphor for a small, necessary continuity input; it never means an authentication mechanism. A Custodial Passkey Agent is not a password, access token, encryption key, secret, cryptographic credential, FIDO passkey, or WebAuthn credential. Credentials never belong in a continuity handoff.

The procedure nobody knew existed

Imagine replacing an experienced employee after a careful handover. The new person has the manuals, shared folders, project plans, and a tidy summary of open work.

Weeks later, an unusual case appears. The written procedure says what normally happens, but not why one report must be checked before another after a system change. That detail lived in the former employee's practiced experience: part observation, part judgment, part memory of an old exception.

The organization now faces an awkward choice. It can guess. It can rebuild the reasoning from scattered records. Or it can briefly ask the former employee a narrow question, transfer the answer to the new owner, and close the loop.

AI agents do not remember or judge like people, so this is only an analogy. The shared problem is narrower: responsibility can move to a successor before every useful piece of information moves with it.

First, what is an AI agent?

In plain language, **an AI agent is an AI system given a job, instructions, tools, and a way to work through multiple steps toward an outcome**.

A chatbot mainly answers in a conversation. An agent may also search approved sources, use software tools, maintain a task list, produce files, ask another agent for help, or pause for human approval. The rules that decide which agent acts and when are often called **orchestration**.

An agent's **working context** is what it can use for the current task: its instructions, recent messages, records fetched during the run, and results from tools. The **context window** is the amount of input the model can consider at one time. A model's training may encode general patterns and facts, but that is not a dependable archive of an organization's task history; task-specific material outside the window must be exposed again by tools or the surrounding system.

Saved agent memory is different. It is information stored so the system can bring it into a later task. A **backup** is different again: it is a copy of files or system settings used to restore an earlier version after loss. Chat history may feed context or memory, but it is not automatically a dependable operational record.

Long conversations may be shortened into summaries, and memory systems may retrieve the wrong item or return an outdated one. Some tested models and tasks have used long inputs unevenly; in another evaluation, a newer model retrieved simple facts reliably even near its context limit. The practical point is not that long context always fails, but that reliability depends on the model and task. A 2025 benchmark also found difficulty with extraction, updates, temporal reasoning, abstention, and reasoning across sessions (Liu et al., 2024; McKinnon, 2025; Wu et al., 2025).

This does not make summaries bad. It means a summary can correctly preserve a decision while omitting the rejected options, the unusual exception, the supporting source, or the limit on what the agent was allowed to do.

Handoff is not the same as succession

A **handoff** moves a task or useful information from one agent to another. **Succession** is the bigger change: the new agent becomes responsible for the role, and the old agent is retired or stored in an inactive archive where it cannot keep working.

Think of a relay race. A handoff is the baton exchange. Succession is the point after which the next runner owns the leg. The earlier runner does not stay beside them indefinitely, steering from the next lane.

That last distinction matters. If the old agent stays active as a fallback, the two agents may follow different instructions or keep different versions of what is current. People may then be unsure which agent is responsible. Continuity can quietly become co-ownership.

A fictional failure story

Consider Juniper Field Supply, an entirely fictional company. Each month, an AI agent prepares an internal report that matches supplier invoices with warehouse records. After a routine technology upgrade, the company retires that agent. A successor receives the current instructions, shared records, and a short handoff summary.

On the next report, the successor finds an old contract exception. The summary says to "review the legacy adjustment," but it does not say which older record explains it, why the newest report is incomplete, or whether the exception still applies.

The successor guesses from the patterns it sees. Its report is polished but wrong. A human reviewer catches the mismatch before the report is used.

Nothing dramatic happened. The company had documents, and the handoff took place. What was missing was the connection among the source record, the exception, when it applied, and who had authority to decide.

Research on human organizations provides an analogy, not proof about AI. Researchers distinguish **explicit knowledge** - facts and procedures people can write down - from **tacit knowledge**, such as practiced judgment that may be harder to explain fully. Research also shows that know-how can depend on the relationship among people, tasks, and tools. That helps explain how a summary can be accurate yet incomplete (Argote and Ingram, 2000; Nonaka, 1994).

The same entirely fictional story with a safer continuity path

Still within this fictional example, Juniper's team first checks current instructions, approved records, saved agent memory, and validated backups. None answers the narrow question.

An authorized person opens one temporary recovery session in a restricted workspace. The predecessor can view only the kinds of records needed to answer that question. It cannot change records, send messages, approve work, or take on ordinary tasks.

The predecessor points to the relevant older record, explains only the rationale that the evidence supports, and labels what remains uncertain. The successor checks the material against current sources and decides what, if anything, is usable. The human owner explicitly closes the recovery session, and the predecessor returns to archive.

The successor remains the owner throughout.

That is the heart of the **Custodial Passkey Agent** pattern.

The concept: useful by exception, dormant by default

A **custodial passkey** is the minimum-necessary continuity input needed to close a specific operational gap. A **Custodial Passkey Agent** is a predecessor preserved in dormant custody and temporarily used, within a narrowly governed continuity role, to recover or directly hand off that input when current sources cannot safely close the gap.

It is neither a memory database nor an active parallel worker.

Ordinary agent memory stores information for later tasks. An active fallback remains ready to take work. A Custodial Passkey Agent has no normal work, no ongoing responsibility, and no automatic permission to act. Its role exists only inside one bounded recovery session.

Its public lifecycle has five plain steps:

1. **Archive:** Store the predecessor inactive.
2. **Narrow reactivation:** Permit temporary access for one question.
3. **Evidence recovery:** Find and explain the relevant sources while separating supported facts from uncertainty.
4. **Direct successor receipt:** The successor plainly confirms that the handoff arrived and identifies anything unresolved.
5. **Re-archive:** Close temporary access and return the predecessor to inactive storage.

In compact form: **archive -> narrowly scoped reactivation -> evidence recovery -> direct successor receipt -> re-archive**.

"Reversible" means that access to archived context can be reopened temporarily. It does not mean that responsibility moves backward. A receipt confirms delivery only; it is not approval, proof of truth, proof of identity, or a credential.

Its components have close public neighbors, including agent-succession protocols, research on authoritative continuity and takeover costs, and dormant-memory reactivation. This article uses the term and lifecycle as author-defined vocabulary for that synthesis; it does not claim that the underlying combination is first, exhaustive, independently validated, or a standard (DCP-AI, 2026; KC and Budathoki, 2026; He and Yu, 2026; Yash, Roy, and Bhattacharjee, 2026).

Keep the recovered input bounded

The successor should receive only enough approved material to understand the specific gap, the supporting evidence, the limits on its use, and what remains uncertain. It should not receive the predecessor's entire history, credentials, or renewed authority.

This small, reviewable handoff packet can be called a **continuity envelope**. That is proposed vocabulary for a control objective, not a prescribed schema or file format. Each organization must choose review, retention, security, and closure controls appropriate to its own risks and obligations.

Safeguards in ordinary language

Before the technical terms, the everyday rules are straightforward.

Write down one question and name the person who may authorize the recovery. Let the predecessor see only the necessary records, preferably in a workspace where it can view but not change them. Do not let it contact outside systems or resume ordinary work. Label each statement as supported by a source or as an inference. Have the successor check the answer, record unresolved points, and close access at the deadline.

Security teams may use technical shorthand for these rules. **Purpose limitation** means using access only for the named question. **Least privilege** means granting the minimum access. **Separation of duties** means not letting one actor approve and perform every step. **Provenance** shows where a claim came from. **Freshness checks** ask whether it is still current. **Automatic expiry** ends access on time. **Fail-closed behavior** stops the process when a required check fails.

These controls reduce risk; they cannot make the result automatically safe or correct. Organizations still need authentication, access control, incident response, backups, records management, privacy review, and human oversight. This pattern is not a compliance framework and not legal advice.

Why the predecessor must not become co-owner

The moment the predecessor resumes ordinary work, the pattern changes character.

Two active owners can disagree about the current state. The predecessor may rely on obsolete instructions. The successor may defer instead of learning. Old access may remain open. Operators may route difficult work backward until temporary continuity becomes permanent dependence.

A custodial predecessor does not watch ongoing work, receive new tasks, keep a separate to-do list, or set rules for the successor. It only recovers the agreed evidence or explains the missing context. The successor checks that material and remains responsible.

The central insight is simple: **preserve the predecessor's ability to explain a missing piece without preserving its authority to run the work.**

How this differs from familiar continuity tools

Tool	Its plain-language job	What it cannot guarantee
Ordinary documentation	Records procedures, decisions, and responsibilities. It should remain the first line of continuity.	That every exception, reason, or source relationship was written down.
Saved agent memory	Brings selected facts or prior interactions into a later session.	That the right item will be retrieved, is current, or is authorized for this use.
Backup or checkpoint	Restores stored files, settings, or earlier state after loss.	That the restored state is the right state to use now.
Always-active fallback agent	Keeps a second worker ready for deliberate redundancy.	Clear ownership; the two agents can diverge, duplicate work, or widen access.
Custodial Passkey Agent	Recovers one missing continuity input after current evidence is exhausted.	That the predecessor's answer is complete, current, or correct.

The pattern is a last-mile option, not a reason to neglect documentation. If current records can close the gap, use them and leave the predecessor archived.

Verification is part of the handoff

Reactivation does not recover "the truth." A predecessor can be mistaken, stale, overconfident, or unable to distinguish what it observed from what it inferred.

The successor should therefore verify three things before relying on the handoff:

1. **Evidence:** Can the claim be tied to an approved record that the successor can inspect?
2. **Currentness:** Has a later decision, instruction, or system change replaced it?
3. **Relevance and limits:** Does the material answer the narrow question, and does it make uncertainty visible without importing old authority?

The successor should explicitly record whether the material was usable and what remains unresolved. Closing the continuity event does not mean the transfer was complete or correct.

A three-level maturity ladder

Level 1 - Manual continuity practice. List important agent-supported workflows, name a human owner, preserve approved records, and use a human-led, time-limited exception for any predecessor recovery. Practice with fictional data.

Level 2 - Structured handoff. Use a standard form for the gap, allowed source categories, exclusions, review, receiver acknowledgment, and the event's expiry or retirement decision.

Level 3 - Authenticated and brokered continuity. Where risk justifies it, a separate continuity service can verify participants and enforce scope, duration, and closure. Those controls do not make the recovered content true, and the Custodial Passkey Agent label does not itself prove identity or confer access.

Maturity is not the amount of machinery. It is the organization's ability to preserve the right evidence, keep authority clear, and stop cleanly.

When not to use this pattern

Do not reactivate a predecessor if you cannot confirm that you have the right archived predecessor, that its preserved material is unchanged, that current sources were checked first, and that the narrow review can happen with allowed access. Never revive old credentials to make the pattern work.

Stop if the predecessor and current sources conflict without an authorized way to resolve the difference. Do not use the pattern to make unsupervised legal, medical, financial, employment, safety-critical, or other high-impact decisions. Do not retain a predecessor in conflict with privacy, contractual, legal-hold, deletion, or records obligations. Qualified human owners must decide those questions.

Sometimes the right answer is to re-derive the procedure from current records, consult a subject-matter expert, restore a validated backup, redesign the workflow, or accept that the missing knowledge cannot be safely recovered.

Practical first steps

Start smaller than the architecture diagram in your head.

Identify the few agent-supported workflows where one agent's history is unusually important. List the approved sources that should survive any succession. Write down what the successor must receive, what must never transfer, and who can authorize a narrow exception. Decide how the successor will acknowledge the handoff and when the recovery event ends. Then test the process with fictional data and one deliberately missing piece of context.

The goal is not to preserve every old agent. It is to avoid two bad defaults: deleting useful continuity without review, or keeping retired agents quietly alive forever.

A well-governed successor should inherit responsibility. A custodial predecessor should provide, at most, the missing evidence needed to carry it.

Glossary

Agent: An AI system given a job, instructions, tools, and a way to work through multiple steps toward an outcome.

Context: The instructions, records, recent interaction, tool results, and current state available to an agent while it works.

Saved agent memory: Information deliberately stored so an agent can retrieve it in a later session. It may help continuity, but retrieval and currentness still need checking.

Backup or checkpoint: A stored copy of files, settings, or earlier state that can be restored after loss. Restoration alone does not show that the old state is right for the current task.

Custody: Controlled preservation of records, state, or an inactive agent under a named human owner and clear handling rules.

Handoff: A bounded transfer of a task, evidence, or responsibility from one agent or owner to another.

Succession: The larger change in active ownership from a predecessor agent to a successor.

Passkey (in this article): A metaphor for a minimum-necessary continuity input that helps close one operational gap. It is never an authentication mechanism, credential, password, access token, encryption key, or FIDO/WebAuthn passkey.

Custodial Passkey Agent: A dormant predecessor temporarily used to recover or directly hand off a specific continuity input without regaining active ownership.

Continuity envelope: A small, reviewable packet that states the gap, the supporting evidence, important limits, and remaining uncertainty. This is proposed vocabulary, not a standard format or implementation schema.

Source lock: A declared set of approved source categories for a continuity handoff. This is proposed vocabulary, not a technical lock or security guarantee.

Authority boundary: The line between what an agent may inspect, explain, recommend, approve, change, or send.

Protected reference: A general indication that restricted material exists, without including the material or granting access to it.

Receipt: The successor's record that the bounded handoff arrived and whether anything remains unresolved. It is not approval, proof of truth, proof of identity, or a credential.

Drift: Divergence between current reality and an older agent's instructions, memory, assumptions, sources, or state.

Source list and evidence boundaries

Established adjacent foundations

1. **NASA, "Knowledge Management."** Current official practice page; last updated June 12, 2026. Supports the ideas of critical knowledge, knowledge continuity, and preserving context through transitions. It does not address AI-agent succession. Source: nasa.gov
2. **NASA Knowledge Capture and Transfer Working Group, _Ensuring Knowledge Continuity during Employee Transitions_ (2021/2022).** Official report supporting systematic, owned, targeted transfer rather than tool collection alone. Its NASA survey findings are descriptive and are not generalized in the article. Source: ntrs.nasa.gov
3. **Ikujiro Nonaka, "A Dynamic Theory of Organizational Knowledge Creation," _Organization Science_ 5(1), 1994.** Influential conceptual foundation for tacit and explicit knowledge. Human tacit knowledge is not equated with AI state. Source: doi.org
4. **Linda Argote and Paul Ingram, "Knowledge Transfer: A Basis for Competitive Advantage in Firms," _Organizational Behavior and Human Decision Processes_ 82(1), 2000.** Supports the claim that knowledge can be embedded in people, tasks, tools, and their interactions. Source: doi.org
5. **Linda Argote, "Knowledge Transfer Within Organizations: Mechanisms, Motivation, and Consideration," _Annual Review of Psychology_ 75, 2024.** Current research review supporting the distinction between sending information and achieving usable transfer. Source: doi.org
6. **International Atomic Energy Agency, _Risk Management of Knowledge Loss in Nuclear Industry Organizations_ (2006).** Official, sector-specific guidance supporting risk-based identification and retention of critical knowledge. It is not an AI or compliance standard. Source: www-pub.iaea.org
7. **Theodore R. Sumers et al., "Cognitive Architectures for Language Agents," _Transactions on Machine Learning Research_, 2024.** Primary research organizing language-agent memory into modular components and action processes. Source: openreview.net
8. **Anthropic, "Effective context engineering for AI agents" (2025).** Official vendor guidance explaining why agents need deliberate context selection and why compacted or external memory should preserve important decisions and unresolved issues. It is implementation guidance, not independent evidence or a continuity standard. Source: anthropic.com
9. **Runnan Fang et al., "Memp: Exploring Agent Procedural Memory," Findings of ACL 2026.** Primary research on extracting and using procedural experience as agent memory. It supports treating procedure as distinct from ordinary factual recall; it does not study retired-agent succession. Source: aclanthology.org
10. **Zidi Xiong et al., "How Memory Management Impacts LLM Agents: An Empirical Study of Experience-Following Behavior," ACL 2026.** Primary research showing that retrieved experience can propagate errors or be misleading, supporting verification rather than automatic trust in remembered material. Source: aclanthology.org
11. **Nelson F. Liu et al., "Lost in the Middle: How Language Models Use Long Contexts," _Transactions of the Association for Computational Linguistics_ 12, 2024.** Primary research showing that long-context access can vary with information position. It does not test succession summaries. Source: aclanthology.org

12. **Max McKinnon, "Retrieval Quality at Context Limit" (Google Research, 2025).** Primary vendor research reporting strong position-independent fact retrieval for one tested model and task. It is included as a counterweight: long-context limitations are model- and task-dependent, not universal. Source: research.google
13. **Di Wu et al., "LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory," ICLR 2025.** Primary benchmark covering extraction, multi-session reasoning, temporal reasoning, updates, and abstention. Its reported results are not generalized beyond the evaluated systems. Source: proceedings.iclr.cc
14. **OpenAI Agents SDK, "Agents" and "Handoffs" documentation.** Official vendor documentation illustrating current agent definitions and control transfer through handoffs. It is one implementation example, not a vendor-neutral standard. Source: openai.github.io and Source: openai.github.io
15. **NIST AI Risk Management Framework Core.** Official, voluntary framework supporting documented roles, human oversight, inventories, decommissioning, scope, and governance across the lifecycle. It does not prescribe this pattern. NIST is revising the framework; publication should use the current version. Source: airc.nist.gov
16. **FIDO Alliance, "Passkeys."** Official source defining authentication passkeys as FIDO credentials/password-replacement technology. Cited only to disambiguate the article's metaphorical use. Source: fidoalliance.org

Emerging adjacent work - not established standards

17. DCP-AI, "DCP-06: Succession & Inheritance" (self-published v2.0 protocol specification, 2026). Its page labels the specification "Published" and says revisions are tracked through adopter feedback. It addresses agent succession, identity separation, selective memory transfer, audit records, and decommissioning. It is relevant prior public vocabulary, not an independently issued standard or validation of this article's pattern. https://docs.dcp-ai.org/specs/DCP-06/
18. **Dipesh KC and Anjila Budathoki, "Handoff Debt: The Rediscovery Cost When Coding Agents Take Over Interrupted Tasks" (arXiv:2606.02875, June 2026).** Unreviewed preprint studying coding-agent takeover from interrupted work. It provides adjacent evidence that handoff quality can affect successor effort, but it does not test dormant predecessor reactivation or establish this pattern. Source: arxiv.org
19. **Jun He and Deying Yu, "Beyond Memory: A Transactional Continuity Kernel for Long-Lived AI Agents" (arXiv:2608.11632, August 12, 2026).** Very recent, unreviewed preprint on governed authoritative state and continuity. Source: arxiv.org
20. **Nilutpaul Sarker Yash, Tirtho Roy, and Ushashi Bhattacharjee, "Towards Reversible Forgetting: Managing Obsolete Knowledge in Continual Enterprise AI Agents" (arXiv:2608.18177, August 18, 2026).** Very recent, unreviewed preprint proposing active, dormant, and retired memory states with reactivation. It concerns memory, not bounded predecessor-agent recovery. Source: arxiv.org

Positioning statement

The sources establish adjacent foundations: critical-knowledge continuity, tacit and contextual transfer, procedural and episodic agent memory, context selection and retrieval, handoffs, lifecycle governance, succession, authoritative state, and dormant memory. This package presents the archive-to-reactivate-to-recover-to-successor-receipt-to-re-archive combination as **an author-defined synthesis and operating pattern**. It makes no claim of first invention, patentability, standardization, external validation, or proven effectiveness.