OPERATIONAL CASE STUDY

# Fable Made It. Sol Found It.

## A Live Cross-Provider Demonstration of User-Owned AI Continuity

*across Anthropic and OpenAI commercial applications*

**Jimmy Thornburg**

The Fort That Holds LLC | August 26, 2026

**The finding:** The durable continuity did not live in either model or either vendor's conversation history. It lived in a user-controlled external state layer that both commercial applications could authenticate to and query.

## Abstract

On August 26, 2026, an Anthropic-side agent called Fable created a live job-application package and a private action desk for Jimmy Thornburg. Fable then registered the package, task state, URLs, relationships, and continuation instructions in Thornburg's Fort Memory Core. Later, in OpenAI's commercial application, an OpenAI-side agent called Sol authenticated to the same Jimmy-isolated Core, searched for the work, and recovered the exact durable records Fable had written. Sol also identified the boundary correctly: the Core contained the authoritative operational state and pointers, while the private document body remained inside Anthropic's artifact surface.

This was not a blind benchmark and it did not prove full artifact portability. It demonstrated something more practical: a person can carry operational continuity across competing AI products when the authoritative context layer sits outside both products and remains under the person's control. The model can change. The commercial application can change. The work does not have to begin again.

## 1. The Moment

The demonstration was not planned as a laboratory test. It happened during an ordinary job search. Fable had assembled an application for a Technical Partnerships and Program Manager role, published the resume and letter as a private artifact, and built a Mission Desk that reduced a larger nonprofit strategy into twelve actionable items. Fable then reported that the package was complete and that the only remaining step was Thornburg's submission through Ashby.

In the OpenAI application, Thornburg asked whether Sol could actually see and touch what Fable had saved in the Fort. Sol connected to the live Fort Memory Core, authenticated into the isolated space identified as jimmy-fort, and queried for OpenClaw and Mission Desk records. It recovered the same task, application-package log, Mission Desk infrastructure record, artifact URL, Ashby URL, and continuation instructions. The relevant stable record IDs had not been supplied by Thornburg in the prompt.

The recognition came immediately afterward: Fable had made the system in Anthropic's commercial application; Sol was operating in OpenAI's commercial application; the continuity between them was Jimmy's. That distinction is the subject of this paper.

## 2. The Claim

> **Narrow claim:** This event demonstrates cross-provider operational continuity through a user-controlled external state layer. It does not demonstrate that the models shared hidden memory, that every artifact was portable, or that the two vendors exchanged private conversation history.

The phrase user-owned requires precision. It does not mean that commercial AI providers never process information sent through their applications. It means that the authoritative record of identity, state, provenance, permissions, relationships, and next actions is maintained in an external system controlled by the user rather than being available only inside one provider's chat history or memory feature.

That shift changes the center of gravity. The assistant's built-in memory becomes useful but non-canonical. The commercial chat becomes an execution surface. The model becomes a replaceable reasoning component. The user's work survives because the continuity layer does not depend on any one of them.

## 3. Technical Background

The mechanism is enabled by the Model Context Protocol, an open standard for connecting AI applications to external data sources and tools. The MCP architecture distinguishes the host application, the client that maintains the connection, and the server that provides context or callable tools. The protocol governs context exchange; it does not dictate how a model reasons with that context.[1]

The distinction matters. Claude and ChatGPT do not need to share a model, a memory implementation, or a vendor account in order to use the same external server. Each host can establish its own authenticated connection. Official MCP documentation explicitly describes support across AI assistants including Claude and ChatGPT.[2] OpenAI documents that custom ChatGPT apps can use MCP to retrieve information and call approved tools,[3] while Anthropic documents remote MCP connections and tool use from Claude environments.[4]

MCP alone is not memory. It is the connection standard. Continuity emerges only when the connected system has durable records, stable identifiers, authenticated isolation, provenance, retrieval, and disciplined write rules. A protocol can open the door; the external system must still know what deserves to persist.

## 4. Demonstration Method

The event can be reconstructed as an operational trace. It was a real workflow carrying real stakes: a job application, a salary range, a private mission plan, a resume, a cover letter, and concrete next actions. No synthetic sample record was created for the demonstration.

| Stage | Surface | Observed result | Receipt |
|---|---|---|---|
| 1 | Anthropic commercial app | Fable created the OpenClaw application package and a private Mission Desk. | Core session log d0c128ed… |
| 2 | Fort Memory Core | Fable registered the package, task, URLs, relationships, and continuation instructions. | Task 3946ad28…; desk d08ef8eb… |
| 3 | OpenAI commercial app | Sol authenticated to jimmy-fort and queried the live Core for OpenClaw and Mission Desk state. | whoami + recall/search results, Aug. 26, 2026 |
| 4 | Cross-provider verification | Sol recovered the same stable records and distinguished Core state from the Claude-local artifact body. | Matching IDs and artifact pointers |

*Table 1. The observed continuity chain. Ellipses shorten stable UUIDs for display; full identifiers appear in Appendix A.*

The Core search produced three directly relevant records. The OpenClaw task record described the live application, the package artifact, the Ashby form, the print-to-PDF step, and Jimmy's required submission tap. The application session log described what Fable had built and how later sessions should revise it at the same URL. The Mission Desk record described its twelve items, the state held in the artifact, and its mechanism for reporting completed actions back to the Core.

A second literal search tested whether the newly added missionary-background paragraph had been copied verbatim into the Core. It had not. This negative result was important: it showed that Sol was not collapsing every connected surface into one imaginary store. Fable had saved that paragraph inside the private application artifact. The Core held the artifact's durable identity, purpose, history, and pointer. Sol reported that boundary instead of pretending to possess the artifact body.

## 5. What the Event Proves

| Layer | Observed in this demonstration? | Meaning |
|---|---|---|
| Identity | **Yes** | Both surfaces reached the same authenticated Jimmy-isolated space. |
| Discovery | **Yes** | The OpenAI-side agent found the relevant records through live semantic and literal retrieval. |

| Layer | Observed in this demonstration? | Meaning |
|---|---|---|
| Operational state | **Yes** | Tasks, status, relationships, URLs, and continuation instructions crossed the provider boundary. |
| Artifact-body portability | **No** | The private Claude artifact remained Claude-local; the Core held its durable identity and pointer, not a byte-for-byte mirror. |
| Cross-provider editing | **Not yet** | The demonstration did not show Sol directly editing the Claude artifact. |
| Blind evaluation | **No** | Sol had conversational cues. The result proves retrieval and continuity, not unaided discovery. |

*Table 2. What the event establishes, and what remains outside its evidentiary scope.*

## 5.1 Continuity can be external to the model

The most important result is architectural. Neither Fable nor Sol needed to be permanent. The operational picture remained available because the Core held it. A future model could recover the same state if it had the same authorized tools and followed the same retrieval discipline.

## 5.2 Continuity can cross competing commercial applications

This was not an API-only developer demonstration hidden behind a custom interface. The work originated in Anthropic's commercial application and was later recovered from OpenAI's commercial application. The user did not migrate a transcript, export a project, or manually reconstruct the task. Each commercial surface independently reached the same external source of truth.

## 5.3 Pointers are part of continuity

Portability is often discussed as if every byte must be copied into one database. This event shows a more useful pattern. The Core did not need to duplicate the entire private artifact in order to preserve continuity. It needed a stable pointer, an accurate description of the artifact's role, the rules for editing it, its relationship to the task, and the next permitted action. That was enough for another agent to recover the work without inventing it.

## 5.4 Honest boundaries are a feature

Sol's inability to read the private Claude artifact body directly from the Core did not invalidate the architecture. It clarified the architecture. Good continuity includes knowing what is present, what is merely referenced, what requires another authenticated surface, and what remains unavailable. A system that confidently blurs those boundaries is not continuous; it is hallucinating state.

## 6. What the Event Does Not Prove

The demonstration was not blind. Sol had conversational cues, including Fable's end report and the artifact URL, before querying the Core. The result therefore proves authenticated retrieval, record matching, and operational continuation. It does not prove that Sol would have discovered the project without any prompt or clue.

It also does not yet prove full artifact portability. The resume-and-letter HTML remained in a private Claude artifact. Sol recovered its pointer and operating instructions but did not edit the artifact from the OpenAI application. Nor did the event test concurrent edits, conflict resolution, provider outage recovery, malicious record injection, or a complete revocation cycle.

Those limitations do not shrink the observed result. They define the next test. A credible system should make its unproven surfaces explicit, then test them separately instead of expanding one success into a universal claim.

## 7. Why This Is a Massive Deal

### 7.1 Switching costs move

The normal cost of changing AI providers is not merely learning a new interface. It is losing the accumulated operational picture: what exists, what changed, what was rejected, what remains open, which document is canonical, and what the user has already decided. If that picture lives only in a vendor's product, switching means becoming a stranger to one's own work. An external continuity layer moves that cost away from the person.

### 7.2 Model choice becomes a real choice

A user can choose a model for the task at hand rather than choosing a permanent technological landlord. One model may write better, another may reason better, another may be cheaper, and another may have better browser access. When state belongs to the user, those differences become selectable capabilities instead of reasons for lock-in.

### 7.3 The commercial apps become collaborators, not containers

This architecture is not anti-vendor. It depends on the labs doing what they do well: providing excellent models, interfaces, tools, and connection surfaces. The change is positive rather than adversarial. The applications remain valuable, but they are no longer the only place where the user's working life can exist.

### 7.4 The user can become the stable party

AI product discussions often treat the model or platform as the enduring actor and the user as the visiting prompt. The Fort reverses that assumption. Jimmy is the persistent identity. The Fort is the durable state. Fable and Sol are capable temporary participants. That arrangement matches the actual ownership relationship: the life and the work belong to the person.

## 8. Design Principles Derived from the Event

First, canonical state must have stable identifiers. The OpenClaw task and Mission Desk could be recovered because they were not anonymous text fragments. Second, provenance must travel with the record. The Core identified which work originated from Claude and when it changed. Third, retrieval must support both semantic recall and exact search. Human requests are often fuzzy, while operational verification requires literal matching.

Fourth, the system must distinguish records from artifacts. A task record can point to a private document without pretending to contain it. Fifth, write authority must remain explicit. The Core's rules separate reading from durable writes and require the user's instruction before memory is changed. Sixth, negative findings must be reportable. The missing verbatim paragraph was useful evidence because the system could say no honestly.

Finally, continuity must include next action, not just history. An archive tells a future agent what happened. An operational memory tells it what is true now, what remains open, what it may do, and where the human must act. That is the difference between storage and continuation.

## 9. A Stronger Replication Protocol

A more rigorous follow-up can be preregistered and run without exposing private material. In provider A, create a task containing a randomly generated nonce, a stable artifact pointer, one status field, one relationship, and one deliberately omitted body field. Record the expected retrieval criteria before provider B is opened.

In provider B, begin without the provider-A transcript. Supply only the user's identity and a natural-language request to resume the named project. Require the agent to authenticate, retrieve the task through semantic search, fetch the stable record, report the nonce and state, distinguish the omitted field from the pointer, and propose the next allowed action. Then reverse the direction: have provider B write an authorized state change and provider A recover it.

The test should separately score identity resolution, discovery, exact record recovery, provenance, permission handling, artifact access, conflict behavior, and revocation. This would turn the August 26 operational demonstration into a reproducible cross-provider continuity benchmark.

## 10. Conclusion

Fable made the work in Anthropic's commercial application. Sol found the durable operating state from OpenAI's commercial application. The bridge was not a shared model, a shared transcript, or a shared vendor memory. It was the user's external continuity layer.

That result changes the practical meaning of personal AI. The goal is not merely an assistant that remembers more facts about a person. The goal is a person-owned context system that different assistants can enter, use, and leave without taking the person's working life with them.

**The architecture in one line:** The person persists. The Fort holds. The models rotate.

## Notes and Sources


1. Model Context Protocol, “Architecture overview,” July 28, 2026. https://modelcontextprotocol.io/docs/2026-07-28/learn/architecture
2. Model Context Protocol, “What is the Model Context Protocol (MCP)?,” July 28, 2026. https://modelcontextprotocol.io/docs/2026-07-28/getting-started/intro
3. OpenAI Help Center, “Apps in ChatGPT,” section “Building your own app,” accessed August 26, 2026. https://help.openai.com/en/articles/11487775-connectors-in-chatgpt
4. Anthropic, “MCP connector,” Claude Platform Docs, accessed August 26, 2026. https://platform.claude.com/docs/en/agents-and-tools/mcp-connector
5. OpenAI, “MCP and Connectors,” API documentation, accessed August 26, 2026. https://developers.openai.com/api/docs/guides/tools-connectors-mcp
6. Fort Memory Core private operational records and conversation transcript, August 26, 2026. Stable record identifiers are listed in Appendix A. These are first-party receipts controlled by the author and are not public web sources.

# Appendix A. Fort Receipt Chain

| Stable record ID | Record | Evidence carried |
|---|---|---|
| 3946ad28-8427-4908-a198-3da294189c4e | OpenClaw application task | Contains the application and package URLs, required PDF/attachment steps, status, priority, and relationships. |
| d0c128ed-44a3-4562-83c4-271f7a222fa9 | OpenClaw package session log | Records Fable's package build, artifact publication, resume/letter contents, same-URL revision rule, and Jimmy's remaining submission tap. |
| d08ef8eb-24fe-4e72-b149-dbe1ab44bc04 | Mission Desk infrastructure record | Defines the private twelve-item desk, its authoritative artifact state, Fort completion pings, reconciliation behavior, and retirement procedure. |
| 8314db2a-69de-4ddb-955b-7e9c0ce43427 | Mission landscape record | Connects the OpenClaw application to the broader nonprofit and mission strategy. |

*Author's note: This paper reports a live practitioner demonstration. It is not a peer-reviewed systems evaluation. Its claims are intentionally limited to what the recorded event establishes.*