J-JONES

CONTINUITY CURRENT

A weekly look at the infrastructure, memory, and control behind useful AI.

WEEK 001 · JJ AET

JJ AET: Memory Is Becoming Infrastructure. Authority Is Moving Outside the Model.

> **Week:** Week 001
> **Issue date:** Tuesday, October 27, 2026
> **Actual research cutoff:** Monday, October 26, 2026, 12:00 p.m. EDT. Research window: Monday, October 19, 2026, 12:00 p.m. EDT to the cutoff. Items dated earlier are labeled Background.

This week’s central shift

Three things moved in the same week. Inference software started routing by conversation. Memory and context for agents became managed products with review and deletion controls. And security, payment, identity, and consumer agent products all moved authority out of the prompt and into identity, permissions, and human approval.

Interpretation: the useful question for the next year is no longer "which model is smartest." It is "where does context live, and who decides what an agent may do with it." This issue covers four developments, labeled throughout: **Verified** (confirmed in a primary source), **Vendor claim** (stated by the company, not independently tested), **Interpretation** (our reading), **Prediction**, **Background** (dated before this week's window), and **Illustrative example** (an invented scenario with stated assumptions).

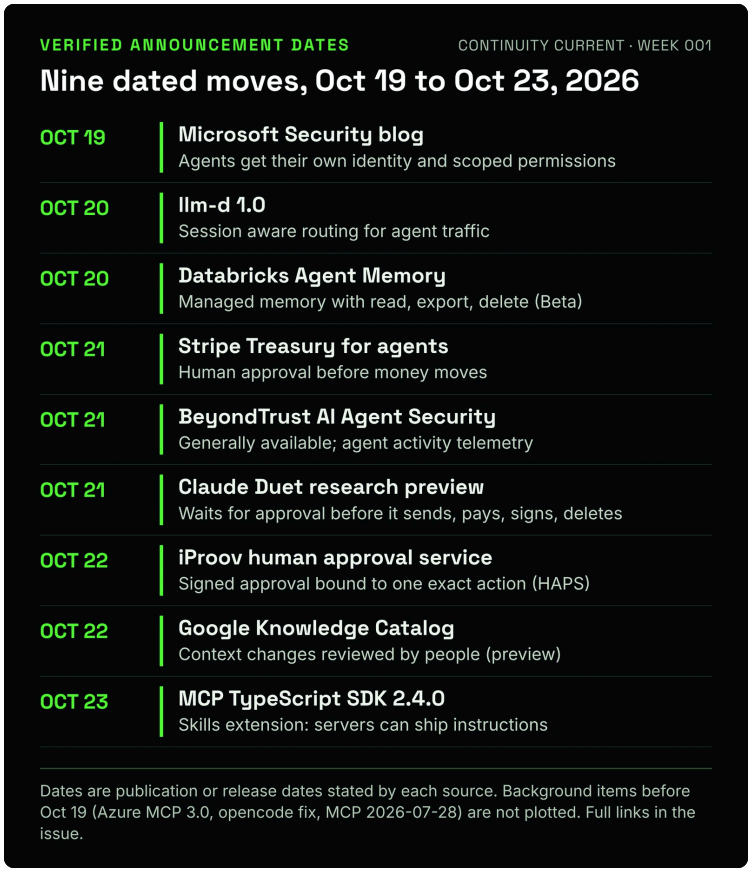


Figure 1. Nine dated moves, October 19 to 23, 2026
Publication or release dates for the in window developments in this issue.
Source: Microsoft [3], llm-d [2], Databricks [1], Stripe [5], BeyondTrust [6], Anthropic [7], iProov [9], Google Cloud [4], MCP TypeScript SDK [11]. Assumptions: Dates are as stated by each source. Background items dated before October 19 (Azure MCP 3.0, the opencode fix, MCP specification 2026-07-28) are not plotted.

Sources: [1] to [11].

If memory and authority become separate layers, which one would you want to own yourself?

1. Infrastructure: inference learns about sessions

What changed. Verified: llm-d, a CNCF sandbox project, released version 1.0 on October 20 with session aware routing, which keeps the turns of one conversation on the replica that already holds its cached context, plus support for the Responses API previous_response_id pattern [2].

Why it matters. Agents resend long, repeating context. Reusing cached work is where cost and latency are won.

Evidence and limits. Vendor claim, adopter reported: IBM reports serving up to 3,000 concurrent coding agents on 544 H100 GPUs at about a 98% prefix cache hit rate and about $0.11 per million tokens [2]. Google reports its prefix cache hit rate doubling from 35% to 70% [2]. These are self reported and not independently benchmarked.

Practical implication. If you self host, ask whether your serving layer understands sessions before you buy more GPUs.

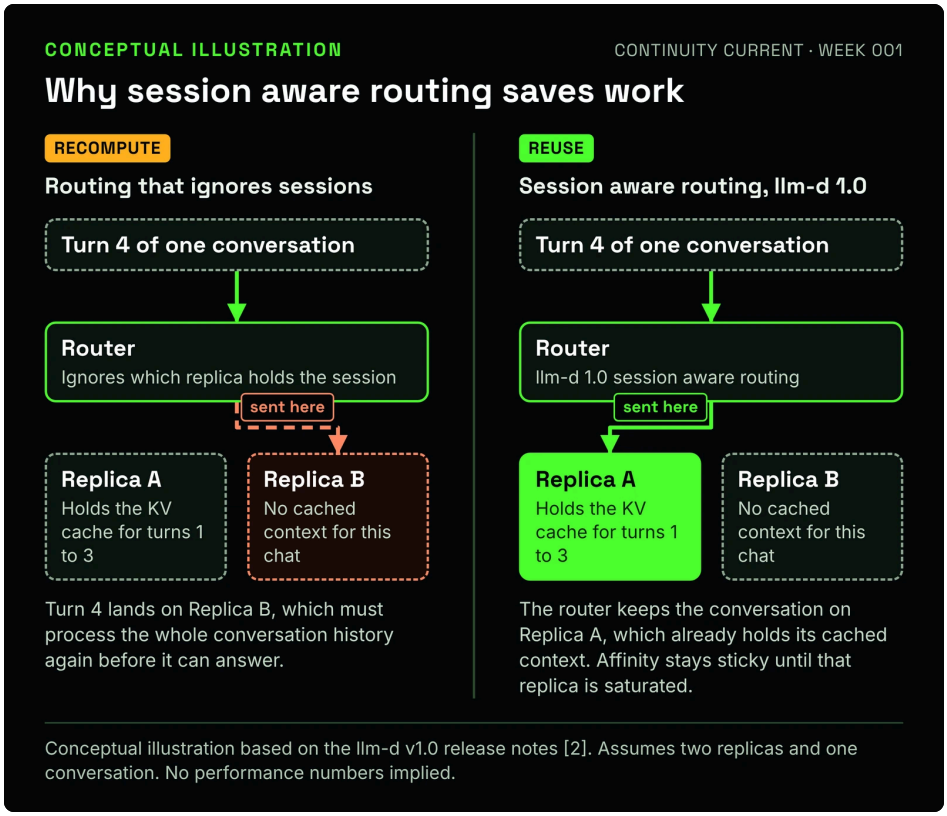


Figure 2. Why session aware routing saves work
Without session affinity, a later turn can land on a replica that must recompute the history. With it, the turn returns to the replica holding the cached context. **Conceptual illustration, based on llm-d v1.0 release notes [2].** Assumptions: Two replicas and one conversation. No performance numbers implied.

Sources: [2].

What would change in your plans if session context became cheaper to keep than to recompute?

2. Memory: becoming a managed, reviewable service

What changed. Verified: Databricks announced Agent Memory in Beta on October 20, with managed sessions, managed memory, and memory pipelines stored in Lakebase, its serverless Postgres service. Developers can read, export, and delete the entries an agent keeps about each user, and Databricks says the service is independent of any single agent framework [1]. On October 22, Google Cloud launched self learning context enrichment in preview for Knowledge Catalog. It proposes definitions and agent instructions with citations and a confidence score; changes below a threshold the team sets, and sensitive updates such as deletions, become a change request a person approves, edits, or rejects. Google says every change is logged and reversible, with rollout over the next few weeks [4].

Why it matters. Memory is becoming infrastructure with storage, review, and deletion rules.

Evidence and limits. Vendor claim: in one Databricks example, a first answer took 11 tool calls, 8 subagents, and nearly six minutes; after the workflow was saved, a related follow up took 2 tool calls and 41 seconds [1]. Databricks also says its memory architecture "tripled agent pass rate" on internal enterprise tasks while using 31% fewer tokens [1]. Google says Knowledge Catalog makes agents "nearly 4X more accurate" with up to 75% fewer tokens, based on internal benchmarks [4]. Neither post gives enough detail to reproduce the tests.

Practical implication. Before you adopt any memory service, confirm who can read it, who approves changes, how export works, and what deletion actually removes.

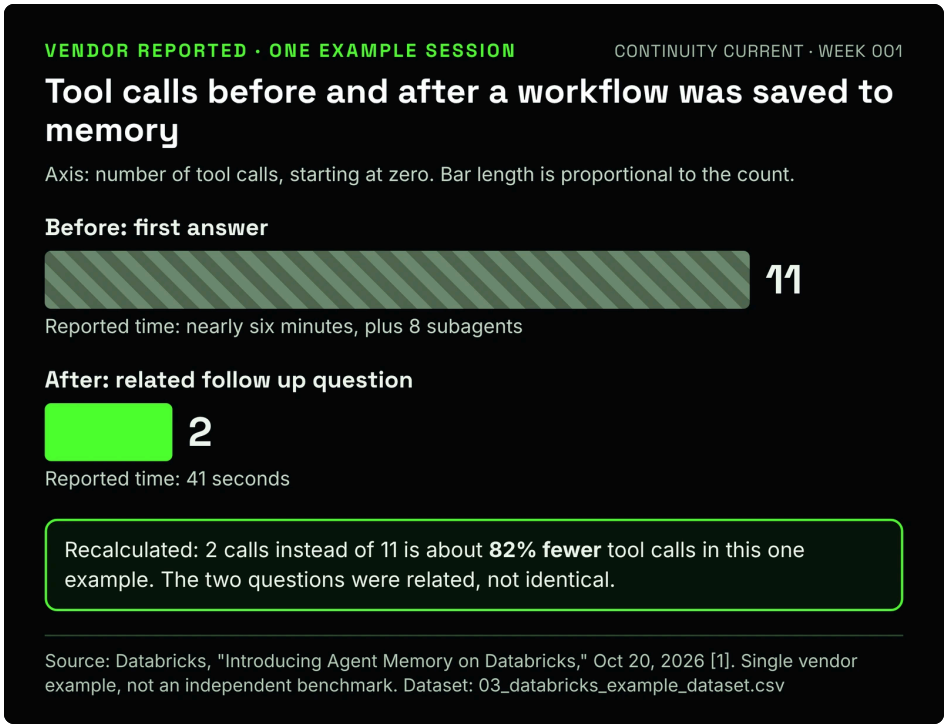


Figure 3. Tool calls before and after a saved workflow, one vendor example
11 tool calls before the workflow was saved, 2 for a related follow up after. Reported times: nearly six minutes versus 41 seconds.
Source: Databricks [1]. Vendor reported, single example session, not an independent benchmark. Dataset: 03_databricks_example_dataset.csv.
Assumptions: Axis starts at zero. The two questions were related, not identical. The 82% figure is our calculation from 11 and 2.

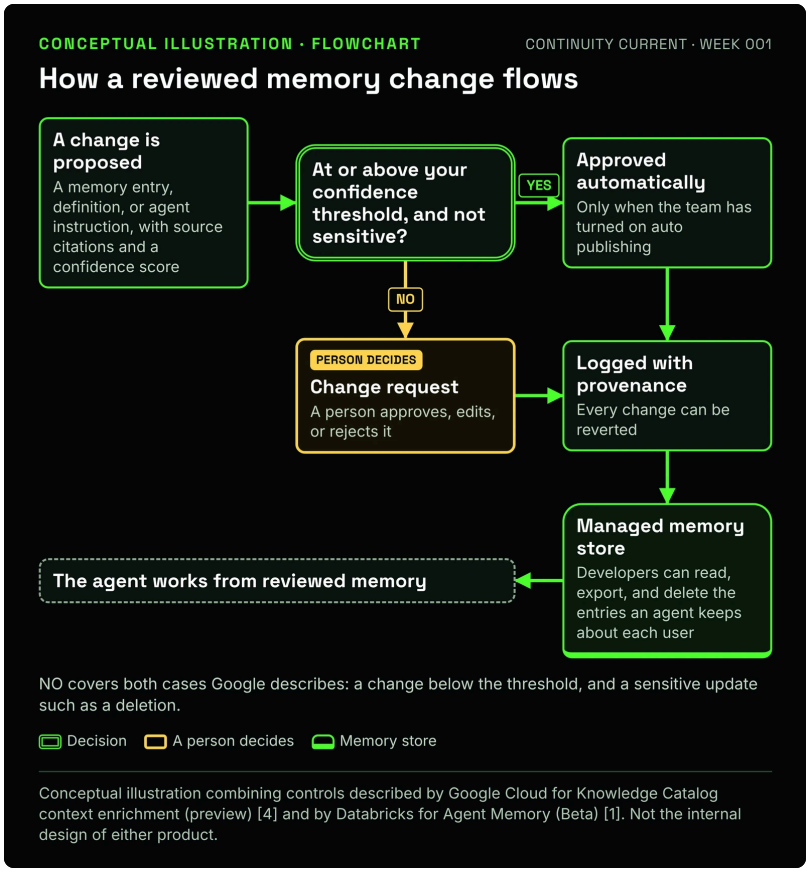


Figure 4. How a reviewed memory change flows
A proposed change to shared context is either published automatically, only when it clears the team's confidence threshold and is not sensitive, or becomes a change request for a person. Either way it is logged with provenance before the agent works from it.
Conceptual illustration combining controls described by Google Cloud for Knowledge Catalog context enrichment, in preview [4], and by Databricks for Agent Memory, in Beta [1].
Assumptions: Not the internal design of either product. Auto publishing applies only when a team has turned it on. Sensitive updates such as deletions go to a person regardless of score.

Sources: [1], [4].

What would you need to see before trusting a managed service with your agent's memory?

3. Security and control: approval moves outside the model

What changed. Verified: on October 19, Microsoft's security team introduced Dynamic Permissions and argued that agents need their own identity: "The employee retains the employee's access. The agent receives its own identity, permissions and its own authority scoped to its current task and for the duration of the task only" [3]. On October 21, Stripe announced Treasury for agents, where startups choose which accounts an agent may use, whether it can act or only read, and when money movement needs human approval [5]. The same day, Anthropic opened a research preview of Claude Duet, which waits for approval before it sends a message, makes a payment, signs a document, or deletes anything [7]; a separate automated check "runs outside the model doing the work" [8]. On October 22, iProov launched an approval service on the open HAPS specification that issues a signed, single use credential bound to one exact action; if a payment's amount or destination changes, the action is blocked [9].

Why it matters. Interpretation: a prompt cannot hold authority. Permission checks, approval gates, and audit records must sit outside the model, where the model cannot talk its way past them.

Evidence and limits. The HAPS repository describes itself as an experimental draft and not production ready [10]. Claude Duet is a research preview, and Anthropic says its safeguards will occasionally miss something [8]. **Vendor claim:** BeyondTrust, launching AI Agent Security on October 21, observed about 1.28 million shell commands requested from about 132,000 prompts in four weeks, from early customer samples; requested is not the same as executed [6].

Practical implication. Give each agent its own identity, keep permissions time limited, and bind approval to the exact action, not to a general "yes."



Figure 5. Where authority sits in one agent request
Two control points sit outside the model: the permission check before memory is read, and the policy gate before a tool runs.
Conceptual illustration, based on principles in Microsoft's Oct 19 post [3]. Assumptions: One agent and one tool. Not a diagram of any specific product, including J-Jones.

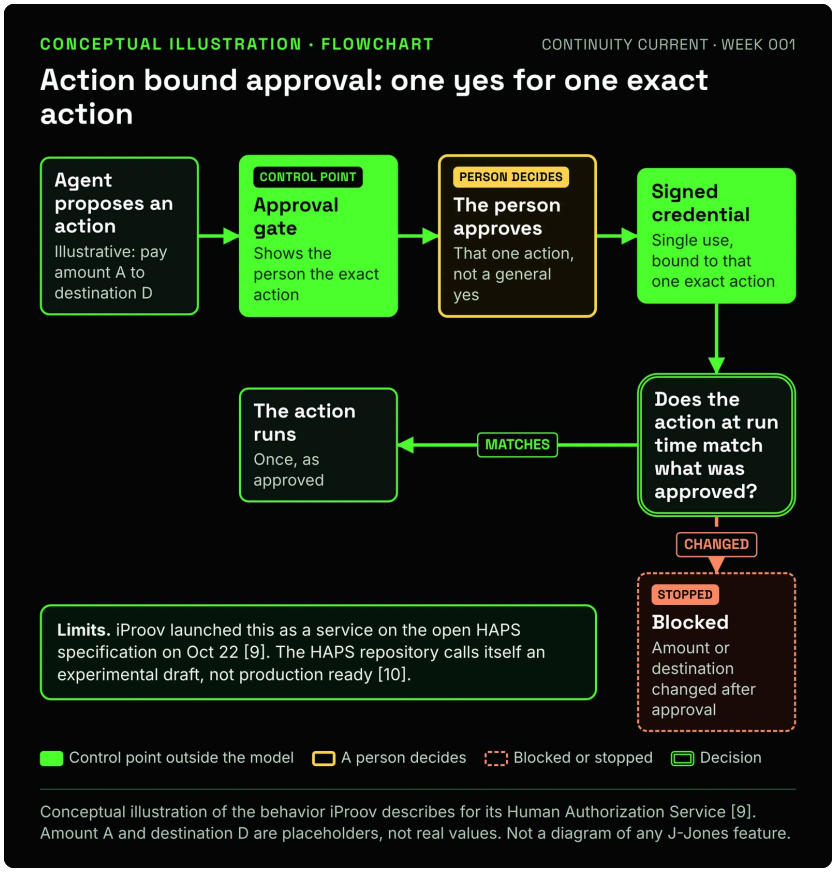


Figure 6. Action bound approval: one yes for one exact action
The approval is bound to the exact action a person saw. If the action that reaches run time matches, it runs once; if the amount or destination changed, it is blocked.
Conceptual illustration of the behavior iProov describes for its Human Authorization Service [9]. The HAPS repository calls the specification an experimental draft, not production ready [10].
Assumptions: Amount A and destination D are placeholders, not real values. Not a diagram of any J-Jones feature.

Sources: [3], [5] to [10].

Which action in your work should never run without a human saying yes?

4. Protocols and patches: the plumbing now carries instructions

What changed. Verified: on October 23, the official MCP TypeScript SDK released version 2.4.0 of its core package with the Skills extension (SEP-2640), so an MCP server can serve skills, meaning instructions for using its tools, alongside the tools themselves. The client verifies each skill's size and digest when reading it [11]. **Vendor claim:** BeyondTrust reports that about half of agent skill installs it observed arrived bundled in code repositories, and that skills often contain embedded code [6].

Background: MCP specification 2026-07-28 removed the initialize handshake and protocol level sessions and binds stored credentials to the server that issued them [12]; Azure MCP Server 3.0 (October 12) supports it with a read only mode [13]. And opencode 1.0.137 through 1.18.32 on Windows and WSL was reported vulnerable to command execution through MCP OAuth, fixed in 1.18.33 on September 28 without a CVE, so scanners will not flag old versions [14], [15].

Why it matters. The channel that delivers tools now delivers instructions too, which makes skills something to inventory and review like code.

Practical implication. List every MCP client and server you run, pin versions, review skills before agents load them, and update opencode to 1.18.33 or later.

BACKGROUND · VERIFIED · SPEC CHANGELOG

CONTINUITY CURRENT · WEEK 001

What MCP 2026-07-28 changed versus 2025-11-25

AREA	CHANGE IN 2026-07-28
Handshake	initialize handshake removed. Each request carries its protocol version and client capabilities.
Sessions	Protocol level sessions and the Mcp-Session-Id header removed.
Discovery	New server/discover call advertises versions, capabilities, identity.
Roots, Sampling, Logging	Deprecated. Still functional during the deprecation window; new implementations should not add them.
Authorization	Clients MUST validate a present iss value (RFC 9207).
Stored credentials	Keyed by issuer. MUST NOT be reused with a different authorization server.

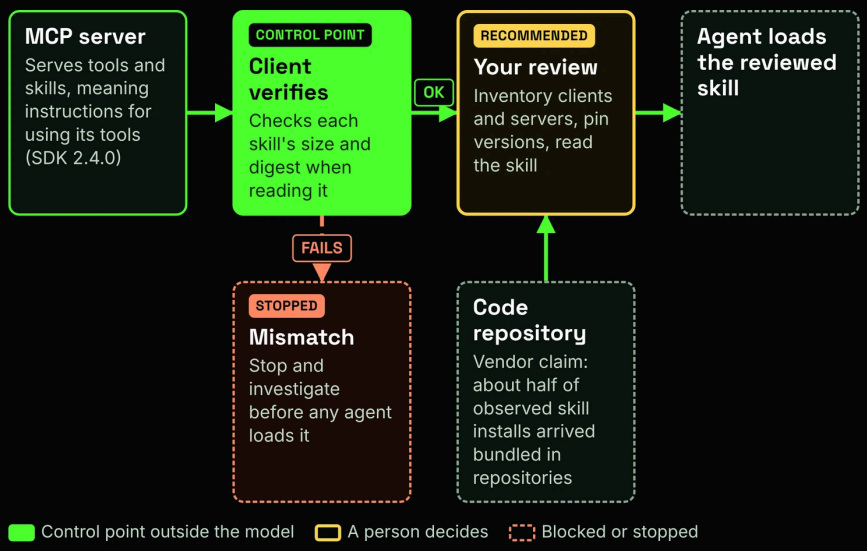
Source: Model Context Protocol specification 2026-07-28, "Key Changes" (modelcontextprotocol.io) [12]. Selected items only; see the full changelog.

Figure 7. Background: what MCP 2026-07-28 changed versus 2025-11-25
Selected changes from the official specification changelog.
Source: Model Context Protocol specification changelog [12]. Background, published before this week's window. Assumptions: Selected items only; see the full changelog.

CONCEPTUAL ILLUSTRATION · FLOWCHART

CONTINUITY CURRENT · WEEK 001

Skills over MCP: check instructions before an agent loads them



Conceptual illustration. Skills extension (SEP-2640) and size and digest checks: MCP TypeScript SDK 2.4.0 release [11]. Repository share: BeyondTrust, vendor reported [6]. The review step is this issue's recommended practice, not a protocol feature. What a client does on a failed check depends on the client.

Figure 8. Skills over MCP: check instructions before an agent loads them
A server can now ship skills, meaning instructions for using its tools. The client checks each skill's size and digest; the human review step is this issue's recommended practice.
Conceptual illustration. Skills extension (SEP-2640) and size and digest checks: MCP TypeScript SDK 2.4.0 release [11]. Repository share: BeyondTrust, vendor reported [6]. Assumptions: The review step is a recommended practice, not a protocol feature. What a client does on a failed check depends on the client.

Sources: [6], [11] to [15].

How would you find every AI tool on your machines that talks to an MCP server today?

5. The J-Jones connection

Disclosure: J founded FranklySolutions and created J-Jones. Treat this section as owner positioning, not independent analysis.

J-Jones is a local first, model agnostic cognitive continuity system. The problem it targets: your notes, decisions, and evidence end up scattered across AI chats and providers, and each new tool starts without them. J-Jones is designed to sit between your own memory and the AI tools you use, so a person decides what each tool may see and do. Its public white paper states the principle behind this week's news: "More AI capability should not automatically mean more authority over persistent information" [17]. **Verified release state:** J-Jones is not yet released; there is no public installer [16].

The two screens below come from the J-Jones interactive demonstration: simulated data, fictional names, and a design direction, not production evidence.

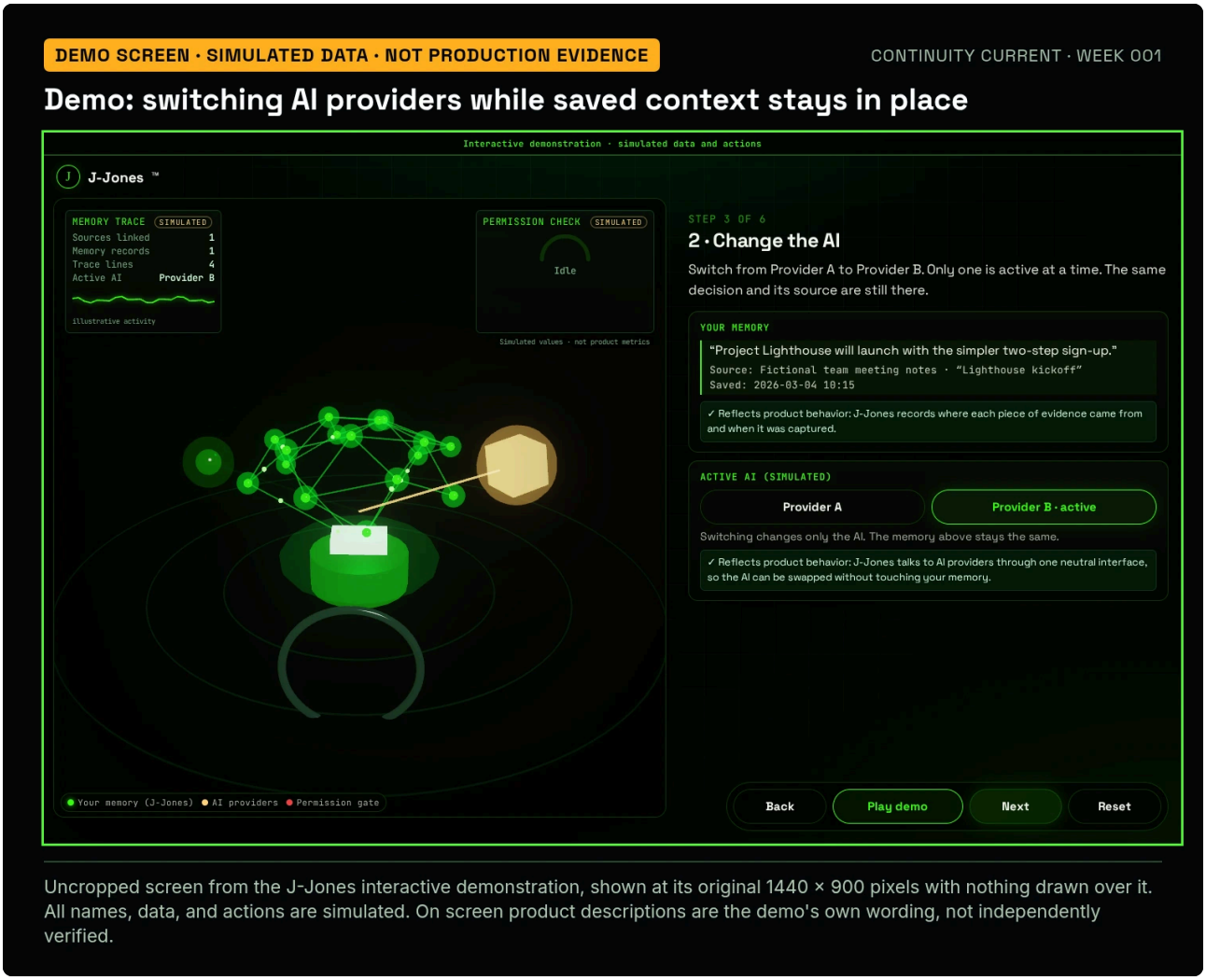


Figure 9. Demo screen with simulated data: switching AI providers
In the J-Jones interactive demonstration, the active AI switches from Provider A to Provider B while a saved decision and its fictional source stay in place.
Source: J-Jones interactive demonstration screen supplied by J (desktop-03), shown uncropped at its original 1440 × 900 pixels, with an enlarged detail of two regions. Nothing is drawn over the screen. Simulated data and actions. Not production evidence. Assumptions: The demo's own banner reads "Interactive demonstration · simulated data and actions." On screen notes that describe product behavior are the demo's wording and were not independently verified for this issue.

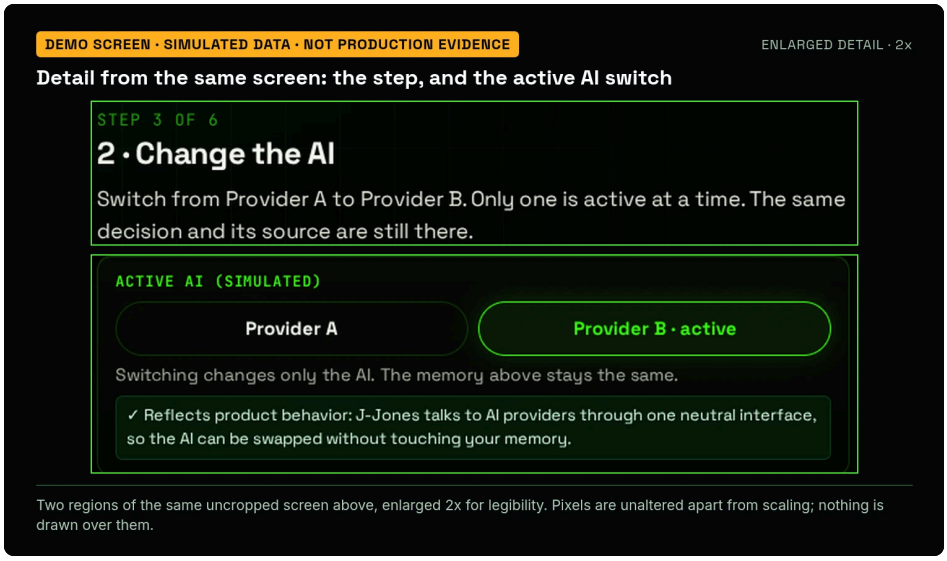


Figure 9, enlarged detail
Two regions of the same demo screen, enlarged 2x for legibility. Pixels are unaltered apart from scaling. Simulated data; not production evidence.

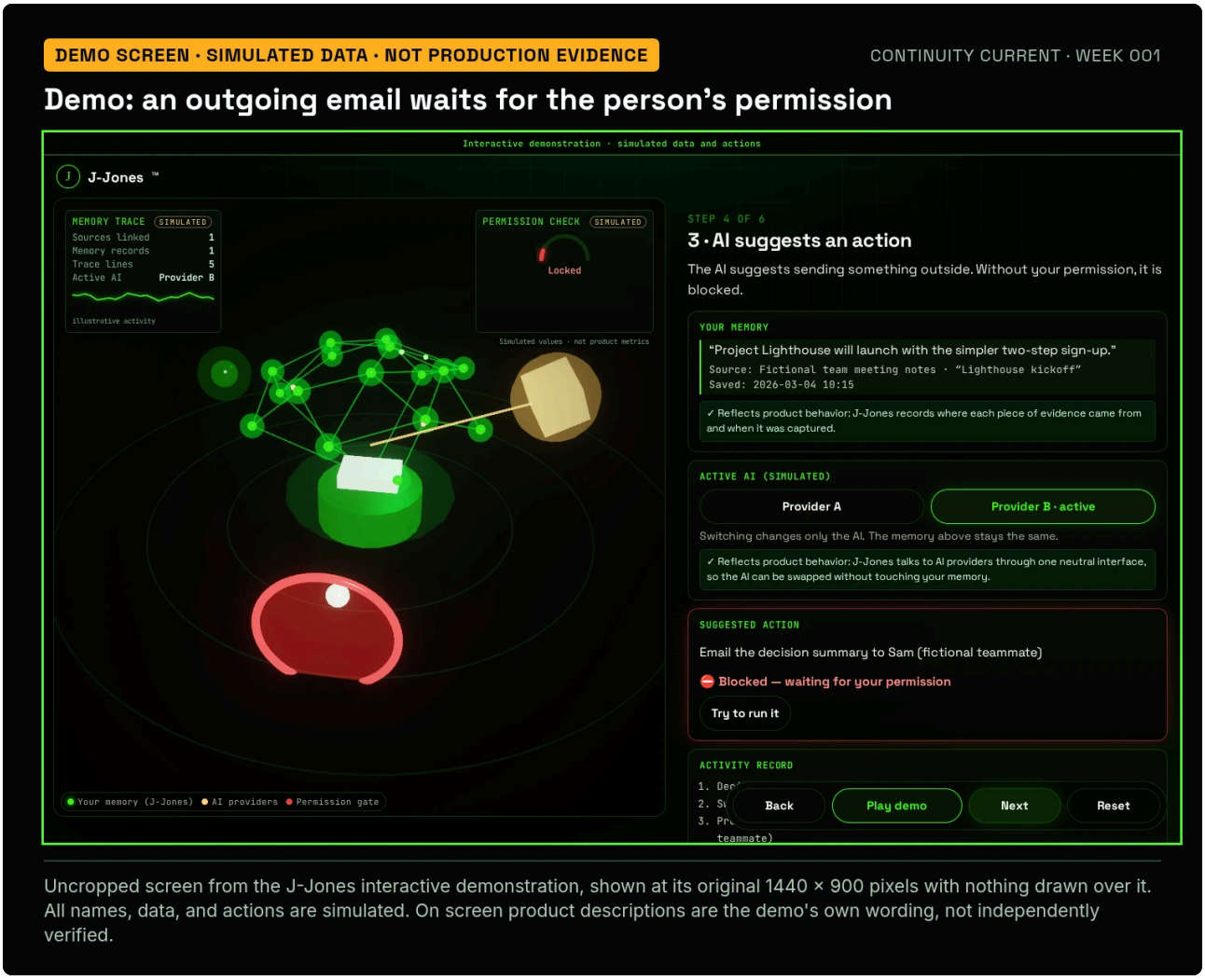


Figure 10. Demo screen with simulated data: an action waits for permission
In the same demonstration, the AI suggests emailing a decision summary to a fictional teammate, and the action shows as blocked until the person gives permission.
Source: J-Jones interactive demonstration screen supplied by J (desktop-04), shown uncropped at its original 1440 × 900 pixels, with an enlarged detail of two regions. Nothing is drawn over the screen. Simulated data and actions. Not production evidence. Assumptions: Fictional names and simulated values. On screen notes that describe product behavior are the demo's wording and were not independently verified for this issue.

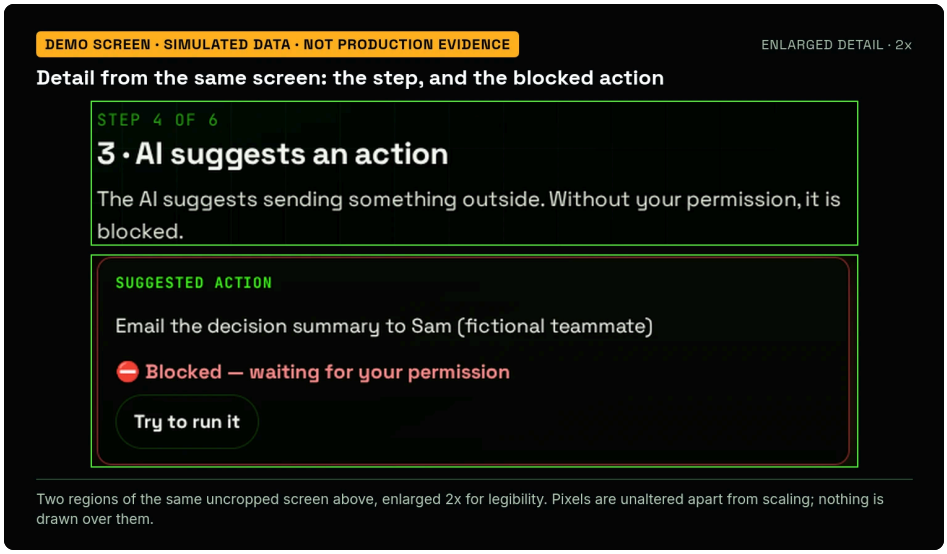


Figure 10, enlarged detail
Two regions of the same demo screen, enlarged 2x for legibility. Pixels are unaltered apart from scaling. Simulated data; not production evidence.

Illustrative example: a consultant moves from one model provider to another. Assumptions: one person, three client projects, a switch made for price or capability. If context lives with one provider, it is restated by hand. If it lives in a user controlled layer, the new model can pick it up while the person approves what is kept. This is a design goal, not a measured result.

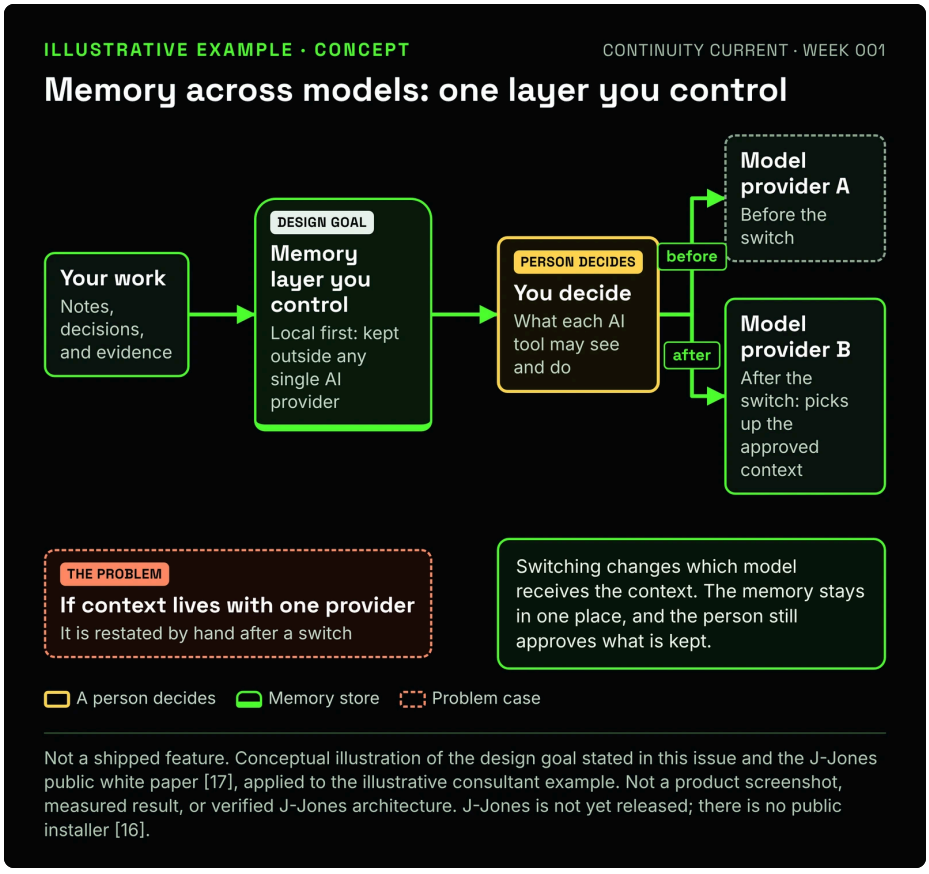


Figure 11. Memory across models: one layer you control (concept)
The illustrative consultant example above as a diagram: switching providers changes which model receives the context, while the memory stays in one place and the person approves what is kept.
Illustrative example and conceptual illustration of the design goal in the J-Jones public white paper [17]. Not a shipped feature, product screenshot, measured result, or verified J-Jones architecture. Assumptions: One person, a switch made for price or capability. J-Jones is not yet released; there is no public installer [16].

Sources: [16], [17]. To talk through continuity for your own work, book a consultation at https://calendly.com/franklysolutions/consultation.

What would you lose today if you had to change AI providers tomorrow?

6. Practical example: one memory permission decision

Illustrative example: a small accounting firm's agent drafts a client email and wants to save "Client prefers quarterly invoices" to long term memory, then send the email. Assumptions: one agent, one client, a firm policy that external messages need approval.

1. The memory write is low impact and inside the task, so it is allowed and logged.
2. Sending the email leaves the firm, so the policy gate pauses it.
3. A staff member reviews the exact message and recipient, then approves. The audit record keeps who approved and when.
4. If the email had quoted text from an untrusted attachment, the gate would pause it for review.

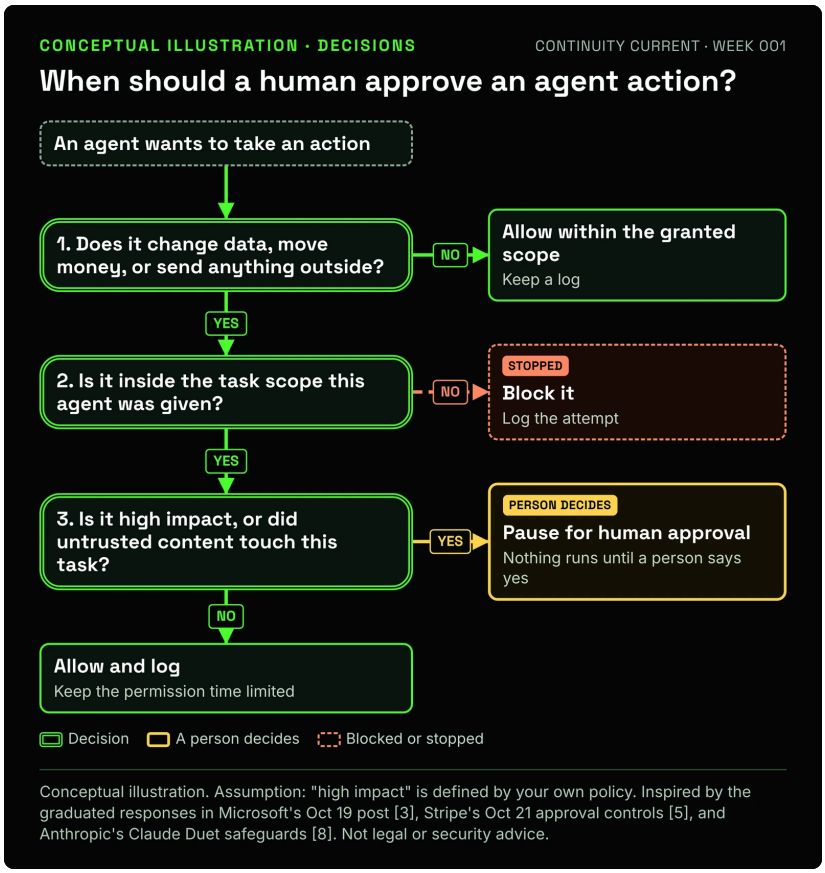


Figure 12. When should a human approve an agent action?
Three questions: does it change data, move money, or leave the system; is it in scope; is it high impact or touched by untrusted content.
Conceptual illustration, inspired by Microsoft [3], Stripe [5], and Anthropic [8]. Assumptions: High impact is defined by your own policy. Not legal or security advice.

Sources: conceptual, based on [3], [5], and [8].

Where would you draw the line between what an agent remembers and what it may do?

7. What to watch next

Prediction: more memory services will publish export, review, and deletion terms, because buyers will compare them, and action bound approval will spread from payments into email, code, and procurement agents. **Verified dates to watch:** Stripe's spending limits are promised "in the coming weeks" [5], Google's context enrichment rolls out "in the next few weeks" [4], and Claude Duet remains a waitlisted research preview for US Pro and Max subscribers [7]. **Open questions:** whether HAPS moves beyond an experimental draft [10], and whether llm-d's adopter numbers hold up in independent tests [2].

Sources: [2], [4], [5], [7], [10]. If you want help mapping where memory and authority live in your own setup, book a consultation at https://calendly.com/franklysolutions/consultation.

Which of these shifts will reach your own work first?

Sources

1. Databricks, "Introducing Agent Memory on Databricks," Oct 20, 2026. https://www.databricks.com/blog/introducing-agent-memory-databricks
2. llm-d project, "llm-d v1.0: State-of-the-art inference, ready for Day 2," Oct 20, 2026. https://llm-d.ai/blog/llm-d-v1.0-state-of-the-art-inference-ready-for-day-2
3. Microsoft Security Blog, "Agents are your newest insiders. Security must move at their speed," Oct 19, 2026. https://www.microsoft.com/en-us/security/blog/2026/10/19/agents-are-your-newest-insiders-security-must-move-at-their-speed/
4. Google Cloud Blog, "Knowledge Catalog: self-learning context enrichment for AI agents," Oct 22, 2026. https://cloud.google.com/blog/products/data-analytics/knowledge-catalog-self-learning-context-enrichment-for-ai-agents
5. Stripe, "Introducing Stripe Treasury for agents," Oct 21, 2026. https://stripe.com/newsroom/news/treasury-for-agents
6. BeyondTrust, "BeyondTrust AI Agent Security Now Generally Available," Oct 21, 2026. https://www.beyondtrust.com/press/ai-agent-security-general-availability
7. Anthropic, "Introducing Claude Duet," Oct 21, 2026. https://claude.com/resources/articles/claude-duet
8. Anthropic, "Claude Duet Safety, Privacy, and Security," Oct 23, 2026. https://claude.com/resources/articles/claude-duet-safety-privacy-and-security
9. iProov, "iProov Launches New Human Authorization Service for AI Agent Gateways with First Integrations," Oct 22, 2026. https://www.iproov.com/press/human-authorization-service-ai-agent-gateways-integrations
10. iProov, "HAPS: Human Approval and Presence Specification" repository, accessed Oct 26, 2026. https://github.com/iProov/HAPS
11. Model Context Protocol, TypeScript SDK release "@modelcontextprotocol/core@2.4.0," Oct 23, 2026. https://github.com/modelcontextprotocol/typescript-sdk/releases/tag/%40modelcontextprotocol/core%402.4.0
12. Background. Model Context Protocol, "Key Changes," specification 2026-07-28. https://modelcontextprotocol.io/specification/2026-07-28/changelog
13. Background. Microsoft Azure SDK Blog, "Azure MCP Server 3.0 stable release: More tools. More possibilities," Oct 14, 2026 (release Oct 12). https://devblogs.microsoft.com/azure-sdk/azure-mcp-server-3-0-stable-release/
14. Background. Pillar Security, "Open and Shut: Remote Code Execution in opencode via MCP OAuth," page dated Oct 19, 2026 (fix released Sep 28). https://www.pillar.security/blog/open-and-shut-remote-code-execution-in-opencode-via-mcp-oauth
15. Background. GitHub, anomalyco/opencode pull request #51414, merged Sep 25, 2026. https://github.com/anomalyco/opencode/pull/51414
16. J-Jones, official site, accessed Oct 26, 2026. https://jjones.ai
17. J-Jones, "J-Jones White Paper + Roadmap: Public Edition 1.0," Sep 30, 2026. Available by email request at https://jjones.ai/#updates

J-Jones Continuity Current. Week 001. Graphics are original works created for this issue, except Figures 9 and 10, which are J-Jones demo screens with simulated data supplied by J.