Collective Memory as a First-Class Primitive for Decentralized Federated Learning and Agent Interoperability

Intellex

June 16, 2025

Abstract

As AI systems shift from single models to networks of collaborating agents, *memory*—personal, organizational, and collective—emerges as the scarce resource that makes agents useful, auditable, and adaptive. We argue that (1) personal and collective memory will be foundational in the era of AI, particularly as *personal language models* (PLMs) and organizational knowledge systems proliferate; (2) the training and governance substrate is moving from centralized federated learning (FL) to *decentralized* FL (DFL) and related gossip/swarm paradigms that better fit cross-organizational collaboration and Web3 incentives; and (3) effective agent ecosystems will require interoperability standards and protocols that let agents *share memory, provenance, expertise, and experience* without surrendering ownership. We propose an architectural pattern where **memory assets carry provenance, improvements to shared memory are rewarded without exposing data, and agents interoperate through open protocols using the collective memory.** A tokenized coordination layer (ITLX) then settles access, contributions, and attestations—ensuring that whenever memory moves, value moves with it.

1 Introduction

The last two years have seen a decisive pivot in AI from single-turn chat to *persistent, personalized* assistance. Mainstream platforms have shipped durable memory features (e.g., ChatGPT's persistent memory and controls) and "personal intelligence" stacks that explicitly fuse on-device context with generative models (e.g., Apple Intelligence). These shifts normalize the idea that the value of AI is not just parametric weights but the *memories* and context that shape behavior over time [26, 7, 6, 18].

At the same time, research and industry are rediscovering social-science notions of *collective memory*—shared, socially maintained knowledge that guides group identity and coordinated action. Halbwachs' classic thesis—that memory

is constructed and preserved socially—anticipates today's need for group-level knowledge systems that are portable across tools and organizations [14, 25].

This paper connects those threads to two technical realities. First, training and adaptation are moving from centralized FL toward *decentralized* schemes (gossip, swarm) that eliminate the coordinator and better fit cross-org constraints [17, 16, 41]. Second, multi-agent systems are mainstreaming: agents must interoperate, exchange evidence about *where knowledge came from* (provenance), and prove *who knows what* (expertise/credentials) [10, 36].

ACT-R, Personal Language Models, and Decentralized Federated Learning

Before turning to the technical mechanics of ACT-R, it is important to clarify why a cognitive model belongs in a discussion of federated learning and agent interoperability. Modern AI systems are increasingly expected to manage not only general-purpose reasoning but also the storage, retrieval, and selective sharing of *personal and collective memory*.

Why Collective Memory Matters

Artificial intelligence systems are often framed primarily as reasoning engines. Yet reasoning alone is insufficient: without memory, reasoning is generic, ungrounded, and detached from context. Memory provides the substrate upon which reasoning operates, ensuring that outputs are tied to past experiences, accumulated knowledge, and prior commitments.

Personal memory enables continuity for individuals. It allows a Personal Language Model (PLM) or agent to remember prior interactions, preferences, and constraints, so that behavior is consistent and trustworthy across time.

Collective memory, however, is even more critical. Communities, enterprises, and institutions rely on shared stores of memory to preserve identity, transmit expertise, and sustain coordinated action. Collective memory ensures that knowledge survives the turnover of individuals, that organizations can learn from past decisions, and that communities can grow without repeatedly reinventing solutions.

If systems fail to manage collective memory, every interaction resets to zero. The result is a collapse of continuity, loss of trust, and breakdown of coordination. By contrast, systems that make collective memory a first-class asset create the conditions for resilience, sustainability, and long-term learning.

Thus, while reasoning is the *process* of applying knowledge, memory—and especially collective memory—is the *infrastructure* that ensures reasoning is personal, durable, and shared across groups.

Memory and Personal Language Models

Personal Language Models (PLMs) can be understood as language models that are bound to an individual's or organization's unique memories. Rather than operating only on general-purpose training data, a PLM integrates personal or institutional knowledge—documents, histories, preferences, and experiences—into its responses. This makes the PLM a vehicle for *owned intelligence*: the ability to reason with and act upon information that belongs to a specific person or community.

Memory is central to how PLMs function. At the declarative level, memory provides the raw material: facts, episodes, and records that a model can retrieve when generating outputs. At the procedural level, memory includes the workflows, preferences, and action rules that guide how tasks are executed. Over time, the PLM refines which memories are most relevant, which procedures are most effective, and how to balance accuracy against cost or efficiency.

What distinguishes PLMs from generalized language models is not larger scale but *ownership and continuity of memory.* A PLM preserves context across sessions, adapts to feedback, and grounds its reasoning in memory assets that remain under the control of their owner. This enables the individual or organization to decide what should be retained, what can be shared with others, and what must remain private. In this way, PLMs transform memory into a structured, portable, and governable resource that makes intelligence personal, auditable, and extensible.

Introduction to ACT-R

John Anderson's ACT-R (Adaptive Control of Thought – Rational) provides a cognitive architecture that distinguishes between:

- **Declarative memory**: facts and episodes stored as "chunks," retrieved based on activation (recency, frequency, context).
- **Procedural memory**: skills and rules encoded as productions (IF–THEN), selected by learned utilities.
- **Control system**: a production system operating over limited "buffers" that exchange information between modules.

In ACT-R, *intelligence* is the capacity to retrieve the right chunks and fire the right productions at the right time. This split provides a principled way to ask: what is *safe, useful, and efficient* to exchange across systems?

Implementing ACT-R in Personal Language Models

Personal Language Models can be viewed as practical implementations of this architecture:

- **Declarative store**: a PLM binds to an individual's personal memory assets—documents, histories, preferences—which function as retrievable chunks.
- **Procedural rules**: workflows, prompt chains, and decision templates act as productions.
- **Utility signals**: feedback about which responses or workflows are effective updates the utilities guiding production choice.

This framing shows that a PLM is not merely a smaller LLM, but a structured agent where *personal knowledge* = declarative + procedural memory with provenance, tuned by experience.

Why Decentralized Federated Learning Matters

Decentralized Federated Learning (DFL) provides a mechanism for PLMs to improve without disclosing raw personal data. Seen through an ACT-R lens:

- **What is exchanged?** Not raw experiences, but strengthened chunks (better retrieval embeddings), refined rules (procedural updates), and improved utilities (performance signals).
- **How is it exchanged?** Peer-to-peer updates and proofs, avoiding the need for a central coordinator.
- **Why is it safe?** Raw personal data never leaves local control; only the *memory traces and utilities* derived from it are shared.

This makes DFL a market for *improvements to memory* rather than for raw data. Each PLM can benefit from collective experience while preserving ownership of its underlying memories.

From Cognitive Models to Collective Learning

1. **ACT-R**: Provides a principled way of distinguishing between different forms of memory—facts we recall, skills we apply, and the signals that help us choose among them.
2. **Personal Language Models (PLMs)**: Translate this distinction into practice by giving individuals or organizations a model that binds to their own memories, workflows, and preferences, making intelligence personal and portable.
3. **Decentralized Federated Learning (DFL)**: Supplies the mechanism for these PLMs to improve collectively, allowing shared learning and stronger collective memory without exposing private data.

Taken together, these concepts offer a coherent logic: *intelligence is only as strong as the memory it draws upon; PLMs provide a way to structure and own that memory; and decentralized federated learning makes it possible to grow personal knowledge into collective memory while preserving control.*

ACT-R as a Framework for Memory

John Anderson's ACT-R (Adaptive Control of Thought – Rational) provides a cognitive architecture that distinguishes between:

- **Declarative memory**: facts and episodes stored as "chunks," retrieved based on activation (recency, frequency, context).
- **Procedural memory**: skills and rules encoded as productions (IF–THEN), selected by learned utilities.
- **Control system**: a production system operating over limited "buffers" that exchange information between modules.

In ACT-R, *intelligence* is the capacity to retrieve the right chunks and fire the right productions at the right time. This split provides a principled way to ask: what is *safe, useful, and efficient* to exchange across systems?

2 ACT-R: A Cognitive Model for Relating Intelligence, Memory, and Knowledge

2.1 Overview (why ACT-R matters here)

ACT-R (Adaptive Control of Thought—Rational) is a cognitive architecture that explains human cognition as the interplay of two kinds of memory—*declarative* (facts/episodes) and *procedural* (skills/rules)—coordinated by a production system operating over module "buffers." Symbolic rules choose actions; sub-symbolic equations determine which facts are retrievable and which rules "win" when several apply. In short: **what you know** (declarative chunks) and **how you use it** (productions with learned utilities) together produce intelligent behavior. This split gives us a rigorous way to talk about *personal knowledge* and *collective memory* as first-class substrates for agent behavior [5, 2, 42].

2.2 Key pieces, very briefly

Declarative memory (chunks). Declarative knowledge is stored as *chunks* with attributes (feature–value pairs). Whether a chunk can be recalled in time depends on its *activation*, a learned function of recency, frequency, and contextual cues ("spreading activation"). A standard form is the base-level activation equation:

$$B_i \;=\; \ln \sum_{k=1}^{n} t_k^{-d}$$

where t_k are the time intervals since each presentation and d is a decay parameter. Intuitively: recently and frequently used facts are easier to retrieve [28, 32].

Procedural memory (productions). Skills are compiled into IF–THEN *productions* that read/write limited "buffers" connected to modules (goal, perception, motor, declarative memory). When multiple productions match the current buffered context, a sub-symbolic *utility* estimate selects the one with highest expected value (speed/accuracy tradeoffs) [2, 33].

Modules and buffers. Independent modules (e.g., declarative memory, goal, perceptual–motor) post small, currently relevant chunks into buffers. The production system can only "see" what's in those buffers—an elegant, resource-bounded API over long-term memory [5, 29].

Knowledge compilation. With practice, reasoning that initially depends on many declarative lookups becomes streamlined: frequently co-used chunks get retrieved faster (activation rises), and multi-step sequences compress into more efficient productions ("skill acquisition") [3].

2.3 Three working definitions (aligned to our thesis)

- **Memory** (in ACT-R): the *stored* substrate—chunks (declarative) and tuned utilities (procedural)—that can be retrieved/used.
- **Knowledge**: structured memory with provenance that is *ready* for retrieval or action (i.e., chunks linked to contexts; productions tuned by experience).
- **Intelligence**: the *ability to use* memory—retrieving the right chunks at the right time and selecting the right productions to act.

These map directly onto modern AI practice: retrieval-augmented generation (non-parametric memory), policy selection (routing/utility), and agentic tool use (buffered interfaces) [2].

From ACT-R to Personal Language Models (PLMs)

Personal Language Models can be viewed as practical implementations of this architecture:

- **Declarative store**: a PLM binds to an individual's personal memory assets—documents, histories, preferences—which function as retrievable chunks.

- **Procedural rules**: workflows, prompt chains, and decision templates act as productions.
- **Utility signals**: feedback about which responses or workflows are effective updates the utilities guiding production choice.

This framing shows that a PLM is not merely a smaller LLM, but a structured agent where *personal knowledge* = declarative + procedural memory with provenance, tuned by experience.

2.4 From ACT-R to Personal Language Models (PLMs)

A PLM is best understood as a model *bound to your memory assets*. In ACT-R terms:

1. Your documents, notes, histories, and preferences are *chunks* (declarative memory) with activation shaped by your usage (recency/frequency) and by cues (contextual prompts). *Design implication:* retrieval layers should weight by time and usage—not just semantic similarity—mirroring ACT-R activation [28].
2. Your "ways of doing things" (templates, SOPs, prompt-chains) are *productions* whose utilities should adapt to cost/quality feedback. *Design implication:* an intelligence router that picks between tools/models is a production-utility selector in modern clothes [2].
3. As you use the PLM, declarative lookups become faster and multi-step workflows compress (knowledge compilation). *Design implication:* let the PLM promote frequently co-occurring lookups into reusable skills or adapters [3].

This gives a principled, human-factors grounding for *personal knowledge*: it's not "a big model," it's *your* chunk store (with provenance) plus your learned utilities for acting [4].

2.5 From ACT-R to Collective (Organizational) Memory

Social theory emphasizes that collective memory is constructed and transmitted through group processes; ACT-R shows how memory drives action. Put together:

- A team or enterprise can expose a *shared chunk store* (glossaries, SOPs, incident reviews) with provenance. Think of this as a "group declarative memory" whose activation reflects organizational recency/frequency of use. *Design implication:* shared indexes should track usage-weighted freshness and cite sources [29].
- The organization's policies and playbooks are *group productions*; their utilities update from outcomes (cost, risk, customer satisfaction). *Design*

implication: route tasks to policies/models with learned utilities and log the utility updates as part of provenance [2].

- Buffers act as clean interfaces between functions (support, ops, finance). *Design implication:* agent interoperability should enforce minimal, typed exchanges—"only what fits in the buffer"—rather than wholesale data dumps [33].

2.6 Bridging to decentralized federated learning (DFL)

ACT-R's base-level learning gives a normative recipe for "which memory matters now." In decentralized settings:

1. **Update markets.** Multiple parties contribute local updates that *increase the activation or utility* of useful chunks/skills (better retrieval, better policies). Rewards should be tied to measured improvement (evaluation lift), not to data volume. *Analogy:* paying for "stronger memory traces" rather than for raw data [28].

2. **No central coordinator needed.** Gossip/swarm variants distribute aggregation across peers; ACT-R's emphasis on local cue-driven retrieval supports this: each node can strengthen the traces that matter for its context and share only *updates* and *proofs* [31].

3. **Provenance.** In human cognition, we remember *where* something was learned (source memory). Machine collectives should do the same: every contributed chunk/update carries provenance so agents (and auditors) can assess reliability. *Practical standard:* PROV/VCs attached to memory assets and updates.

2.7 Design principles we inherit from ACT-R

1) Treat memory as weighted, time-sensitive. Retrieval should reflect recency/frequency (base-level activation), not just static embeddings. *Metric:* exposure-weighted freshness for each memory asset [28].

2) Separate "facts" from "skills." Keep declarative stores (what we know) distinct from procedural policies (how we act); log utility updates from outcomes (cost/quality). *Metric:* utility lift per policy over time [2].

3) Keep interfaces small (buffers). Exchange just enough context to choose/act; resist cross-org data hoarding. *Metric:* bytes moved per successful task; provenance completeness [33].

4) Compile what works. Promote frequently co-used lookups into skills (adapters, macro-productions) with auditable provenance. *Metric:* reduction in steps/time for repeated workflows [3].

5) Reward improvements to memory, not data volume. In DFL, pay for better retrieval/decision quality (evaluation lift) rather than for uploading data. *Metric:* model/agent performance delta per contributed update [31].

2.8 Implications for agent interoperability

Finally, ACT-R's architecture itself is a template for interop:

- **Shared performatives over buffers.** Production rules operate on limited, typed buffer content; likewise, cross-org agents should communicate via small, typed messages (requests, informs, proposals) that reference memory assets by ID and include provenance/credentials. *This mirrors FIPA-style ACLs adapted to modern agents* [33].
- **Utility-guided routing.** When several agents could respond, pick the one with the best learned utility (cost/quality) for the current cues—exactly the subsymbolic choice in ACT-R. *This is your "intelligence router" rationale through a cognitive lens* [2].
- **Provenance as memory of action.** Each interaction should generate a compact *UseReceipt* that ties outputs to inputs and memory assets—a machine analogue of source memory that enables trust and learning over time.

2.9 Bottom line

ACT-R gives a mature, evidence-based vocabulary for our claims:

> *Memory* (chunks with graded activation) and *knowledge* (retrievable, contextualized memory plus tuned skills) are distinct, and *intelligence* is the competent use of both under constraints. Personal knowledge (PLMs) is the owned, provenance-bearing chunk store + learned policies of an individual or team; collective memory is the shared, usage-weighted store and policy library a group preserves. Decentralized federated learning is the market where improvements to that shared memory are exchanged without exposing raw data; agent interoperability is the buffer-level protocol that lets many actors use and update that memory safely.

This lets us justify, with cognitive grounding, why *personal knowledge and collective memory must be treated as first-class assets*—and how to wire systems so agents can create, use, update, and revoke them across organizational boundaries [5, 2].

3 System Model & Formal Definitions

3.1 Actors

- **Owner** (person or enterprise): holds keys over one or more memory assets; can license, revoke, and receive royalties.
- **Requester** (app/agent/org): asks to use a memory asset (e.g., call a PLM or knowledge shard) under a license.
- **Agent** (software actor): executes "ask" requests, tools, and training locally for its principal (Owner or Requester).
- **Attester**: verifies provenance/quality claims (e.g., dataset origin, policy compliance); issues revocable attestations.
- **Verifier/Auditor**: checks enforcement (e.g., that revocations were honored; that receipts are complete).
- **FL Coordinator** (can be decentralized): opens rounds, collects update claims, evaluates, and settles payouts.

3.2 Artifacts (on-/off-chain primitives)

MemoryAsset Addressable knowledge/model artifact used at inference or training time (e.g., vector index, fine-tuned head, policy library).

LicensePass Time/scope/cap constrained access right to a MemoryAsset.

UseReceipt Immutable record linking a permitted use to inputs, policy, and metering outcome.

UpdateDelta A local improvement (gradient, adapter, ruleset) proposed to improve shared memory or a model.

Tombstone A revocation record that *invalidates* previously granted access and must be enforced by gateways/caches/agents.

Attestation A signed claim about an artifact (e.g., provenance, evaluation result, credential).

3.3 Canonical fields (schemas)

MemoryAsset

```
{id, owner, fingerprint, uri, policy_hash, royalty_split, price_per_call,
version, created_at, credentials[], provenance_ref}
```

LicensePass

```
{id, asset_id, licensee, scope[], cap_calls, expiry, issued_at, policy_ref}
```

UseReceipt

{id, asset_id, license_id, requester, policy_version, metered_calls, proof_hash, output_hash, ts, settlement {amount, splits}}

UpdateDelta

{id, base_model_ref, contributor, delta_hash, metrics_claimed, stake, ts}

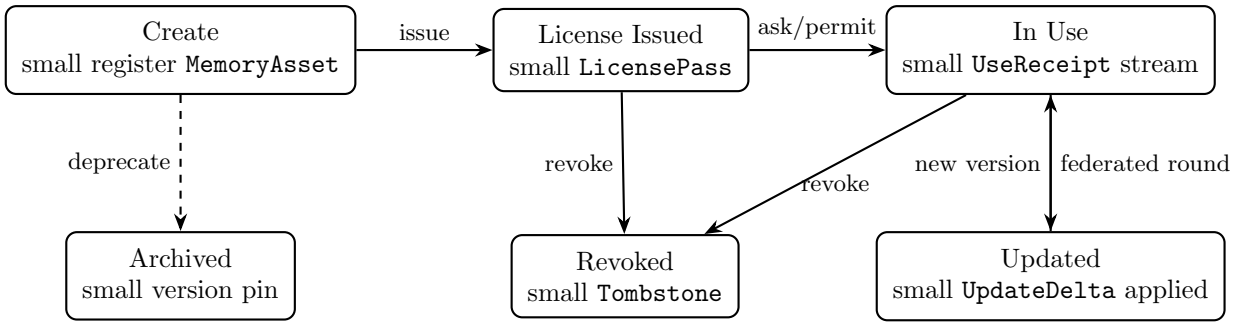
Tombstone

{id, asset_id, reason, revoked_at, policy_version}

3.4 Assumptions & trust boundaries

- Finality on the settlement chain is bounded and known; licenses and tombstones reference finalized blocks.
- Gateways/agents are *honest-but-curious*; misuse is deterred via receipts, audits, slashing, and revocation enforcement.
- FL evaluation may be centralized at first (pragmatic) with a roadmap to decentralized or committee-based scoring.
- Personally identifiable data *does not* leave the Owner's environment in the clear; results/updates/proofs are exchanged.

3.5 State machine (asset lifecycle)



3.6 Invariants

1. Every permitted use produces a *UseReceipt*; every revoke produces a *Tombstone*.
2. A *Tombstone* must prevent any post-`revoked_at` use from being receipted; attempts become evidence of violation.
3. Version pins and provenance form an immutable chain of custody for `MemoryAssets` and applied `UpdateDeltas`.

3.7 Operation sketch (contract API)

- `mint_asset()`, `issue_license()`, `record_use_receipt()`, `revoke()`, `open_round()`, `record_score()`, `settle_round()`.

---

4 Threat Model & Privacy Guarantees

4.1 Adversaries

- **Curious requester**: attempts to infer private data from outputs or repeated queries.
- **Malicious gateway**: continues serving a revoked asset (cache replay).
- **FL poisoner**: submits harmful UpdateDeltas to degrade models or embed backdoors.
- **Sybil farm**: creates many identities to farm rewards or manipulate governance.
- **Prompt-injection exfiltration**: abuses tool calls to leak data during "ask."
- **Colluding attesters**: issue false provenance/quality claims.

4.2 Controls & guarantees

1. **Least-privilege policy & sandboxed inference**. All *ask* calls go through a policy engine (scope, cap, rate) and run in sandboxes; tool calls are whitelisted, logged, and rate-limited. *Guarantee*: minimized attack surface; auditable tool usage.
2. **Revocation enforcement with tombstones**. Revocations emit `Tombstone` and invalidate caches; gateways must check the latest tombstone before serving. *Guarantee*: post-revoke calls are blocked; violations are provable.
3. **UseReceipts & selective disclosure**. Each permitted use produces a receipt (policy version, hashes). Receipts can be shared publicly with selective disclosure (e.g., reveal metering, hide full prompt).
4. **FL sanity checks + bond/slashing**. Submissions require a small stake; evaluator performs DP/secure aggregation or held-out scoring; low-quality or adversarial updates are rejected and penalized.
5. **DP/RAG guardrails**. Retrieval adds controlled noise or k-anonymity; long prompts are truncated/filtered for PII; repeated queries trigger throttling or redaction.
6. **Attester governance**. Attesters hold revocable credentials; random audits; cross-attestation required for sensitive claims; slashing for fraud.

7. **Sybil resistance**. DID/VCs for enterprise lanes; minimum-stake and aging for influence; reward curves that penalize correlated behavior.

4.3 Threat → Evidence matrix

Threat	Control	Evidence produced
Cache replay post-revoke	Tombstone + watcher	Revocation tx id; watcher audit log; 403 receipts with `tombstone_id`
Membership inference	Rate-limits + DP	UseReceipt shows DP mode and ε bound; throttling logs
FL poisoning	Sanity checks + stake	Score reports; rejection proofs; slashing event
Prompt-injection	Sandbox tools + policy	Tool call logs; denied-call receipts; policy version pin
Colluding attesters	Cross-attest + audits	Attester credentials; conflicting claims flagged; slashing receipt
Sybil farming	Stake + aging + caps	Identity graph; stake aging data; per-entity caps reported

4.4 Privacy modes (operator-selectable)

- **Strict** (consumer PLMs): selective disclosure receipts; DP retrieval; tool calls disabled by default.
- **Balanced** (most B2B): provenance on by default; tool allow-list; FL eval with held-out sets.
- **Open** (public assets): full receipts; caching allowed; minimal DP.

4.5 Revocation latency SLO

Let T_r be block time of `revoke` and T_e the earliest enforced 403 across gateways. We report $P50/P95$ of $(T_e - T_r)$ weekly, with penalties (fee rebates) if SLOs are missed.

5 Economic Design: Token Flows Tied to Memory Verbs

5.1 Principle

ITLX moves only when memory moves. Sources and sinks tie to the five verbs: **create**, **license**, **use**, **update**, **revoke**.

5.2 Sources & sinks

- **Create** (register/attest): small fee (sink) to deter spam; attesters earn for valid attestations (source).
- **License** (permit): payer commits $ITLX$ for a cap/term; a portion streams as uses occur (escrowed source).
- **Use** (meter): micro-royalty split to Owner; protocol and verifier fees (sinks).
- **Update** (FL): reward pool funds payouts proportional to measured improvement; contributors stake (temporary sink) and winners earn (source).
- **Revoke** (forget): fee funds propagation and audits (sink); violators can be slashed (sink to treasury; partial rebate to challenger).

5.3 Pricing primitives

Let p be the per-call price (in $ITLX$). Let s_o and s_p be the owner and protocol royalty shares with

$$s_o + s_p = 1.$$

For a batch of n metered calls:

$$\text{OwnerRevenue}(n) = s_o\, p\, n, \tag{1}$$

$$\text{ProtocolFee}(n) = s_p\, p\, n. \tag{2}$$

5.4 FL payout rule (evaluation-weighted)

Let M_0 be the baseline metric and M_i the post-update metric. Define the lift $L_i = \max(0,\, M_i - M_0)$. For reward pool R:

$$L_i = \max(0,\, M_i - M_0), \tag{3}$$

$$\text{payout}_i = \frac{L_i}{\sum_j L_j + \varepsilon}\, R. \tag{4}$$

Submissions must post stake σ; invalid or harmful updates forfeit σ.

5.5 Anti-gaming & sybil resistance

- **Attester staking**: signed claims carry stake; contradictory or fraudulent claims are slashable.
- **Correlated-use penalty**: highly correlated requesters (same identity cluster) yield diminishing royalties to discourage wash-usage.
- **License caps & pacing**: caps/epoch pacing prevent instant drain or artificial spikes pre-TGE.
- **Open metrics**: live dashboard shows $ITLX$ metered by verb; if memory doesn't move, numbers don't move.

---

6 Governance & Provenance Policy

6.1 Roles

- **Protocol stewards** (multisig/DAO): parameter updates (fees, floors), module upgrades, treasury policy.
- **Registry operators**: list allowed contract versions; emergency pause for critical bugs.
- **Attesters** (revocable): issue specific attestations (dataset lineage, safety checks).
- **Auditors** (independent): measure SLOs (revocation latency, receipt coverage); publish weekly reports.

6.2 Attester onboarding

1. Submit DID/VC package and scope (what you attest).
2. Stake $ITLX$ proportional to scope; larger scope $\rightarrow$ higher stake.
3. Probationary period with enhanced audits; misbehavior $\rightarrow$ slashing + delisting.

6.3 Parameter governance

- Fee floors/ceilings, slashing ratios, minimum stakes, FL metrics and test sets: proposed via on-chain proposals; timelocked execution; emergency veto for critical risk.

6.4 Dispute resolution

1. **Filing**: bundle of PROV graph, UseReceipts, policy versions, and alleged violation.
2. **Review**: third-party auditor samples logs/caches; attesters weigh in.
3. **Outcome**: uphold (no action), remediation (refunds/revokes), or penalty (slashing, delisting).
4. **Appeal**: time-bounded; requires additional bond; public rationale required.

6.5 Provenance retention & data minimization

- Keep hashes and pointers (PROV edges) on-chain; retain raw artifacts off-chain under owner control.
- Receipts store minimal necessary fields; selective disclosure for public views.

---

7 Interoperability Spec (Minimum Viable Standard)

7.1 Message verbs (inspired by FIPA ACL)

- `request.ask`: requester asks to use a `MemoryAsset`.
- `inform.receipt`: gateway publishes a `UseReceipt`.
- `propose.license`: owner proposes a `LicensePass`.
- `inform.revoke`: owner publishes a `Tombstone`.
- `propose.update`: contributor submits an `UpdateDelta`.
- `inform.score`: evaluator publishes update scores.

7.2 Envelope (common fields)

```
{ id, verb, ts, sender, recipient, intent_ref?,
prov_ref?, vc_refs[], payload{...}, signature }
```

7.3 Payload sketches

request.ask

```
{ asset_id , license_id , question_hash , policy_version , budget_calls }
```

inform.receipt

```
{ asset_id , license_id , requester , metered_calls , proof_hash , output_hash }
```

propose.license

```
{ asset_id , licensee , scope[] , cap_calls , expiry , price_per_call , royalty_split
```

inform.revoke

```
{ asset_id , license_id , tombstone_id , revoked_at , reason }
```

propose.update

```
{ base_model_ref , delta_hash , metrics_claimed , stake }
```

inform.score

```
{ round_id , submission_id , metric , score , payout }
```

7.4 Provenance & credentials

Messages SHOULD include:

- $\mathbf{prov}_{ref}$: $pointer to a PROV bundle documenting inputs \rightarrow$ **outputs.**

- $\mathbf{vc}_{refs}$: $verifiable credentials proving identity, role, or expertise.$

7.5 Cross-chain intents (optional)

- $\mathbf{intent}_{ref}$: $pointer to across-chain intent that resolves payments/escrows across networks; receipts include cha$

7.6 Compatibility

- Verbs map to FIPA performatives; payloads carry PROV and VC references; receipts/tombstones are chain-anchored.

---

8 Evaluation Plan & Pilot Design

8.1 Metrics (reported weekly)

- **Retrieval lift**: task accuracy with vs. without MemoryAssets; ΔEM / ΔF1.

- **Revocation latency**: $P50/P95$ time from revoke to enforced 403 across gateways.

- **FL improvement**: metric lift per round; robustness under non-IID splits.
- **Provenance coverage**: % of outputs with complete PROV links + at least one VC-verified source.
- **Cost/latency**: median $ITLX$ per useful outcome; end-to-end latency budget per "ask."
- **ITLX by verb**: metered by create/license/use/update/revoke.

8.2 Evaluation protocols

E1: Retrieval/Augmentation Sample $N = 500$ queries in a target domain (e.g., logistics). Compare baseline LLM vs. LLM+MemoryAsset. Report accuracy, citations present, latency, and cost. Pre-register prompts and grading rubric.

E2: Revocation SLO Issue a LicensePass to a public gateway, then revoke at t_0. Poll k gateways every second for 10 minutes; record time to first enforced 403. Publish receipts and tombstone id.

E3: FL Round Open a toy dataset round with reward pool R. Require stake σ, run held-out evaluation, publish scores and payouts. Report $\sum L_i$ lift and fairness (Gini) of payouts.

E4: Provenance audit Randomly sample 100 UseReceipts; verify that each links to inputs, policy version, and (if applicable) VC-backed source. Report coverage and discrepancies.

8.3 Pilots (6–8 weeks)

Pilot A: Logistics ETA/Slotting **Hypothesis**: MemoryAssets (lane history, SOPs) reduce lateness and dwell time.
Setup: 3 operators; upload 12 weeks of anonymized history locally; join one FL round.
Success: $\geq$5% MAE improvement; $\geq$10% dwell reduction on suggested slots; P95 revoke $<$ 60s.
Evidence: UseReceipts, FL scorecards, Tombstones.

Pilot B: Nonprofit Intake Triage **Hypothesis**: MemoryAssets (intake rubric, referral map) improve time-to-service and reduce duplicate intakes.
Setup: 3 orgs; shared referral ontology; privacy mode Strict.
Success: $\geq$20% faster time-to-service; $\geq$30% fewer duplicate intakes; provenance coverage $>$ 90%.
Evidence: Receipts with PROV links; audit logs; satisfaction survey.

8.4 Reporting & transparency

Publish a weekly dashboard with the six metrics, links to receipts/tombstones, and notes on any incidents, mitigations, or parameter changes.

9 Why memory (personal and collective) becomes the unit of value

9.1 Personal memory in mainstream AI

Enterprise and consumer stacks now treat *persistent user memory* as product-critical. OpenAI's "Memory" exposes UI controls for saving, recalling, and deleting remembered facts; media and analyst coverage emphasizes that longer-term, user-controlled memory is a shift from stateless chat to ongoing assistance. Apple's "personal intelligence" likewise centers on fusing private context with generative models under strong privacy constraints. Together, these point to memory (not just bigger models) as the driver of utility and trust [26, 34, 6].

Concurrently, research on memory architectures for LLM agents (e.g., MemGPT, hierarchical/long-term memory evaluations) shows that agent reliability and planning hinge on structured, tiered memory rather than raw parameter count [27].

9.2 Collective and organizational memory

Social theory frames *collective memory* as knowledge that is constructed, preserved, and transmitted through social processes; organizational theory formalizes "institutional/organizational memory" as the stored information from an organization's history that informs present decisions. Empirical and board-level guidance underscores the costs of memory loss—from operational failures to strategic drift [14, 40, 9].

For enterprises, this maps cleanly to AI: models should *run on* institutional memory and cite it (provenance) so decisions can be audited and improved. The RAG literature explicitly separates *non-parametric memory* (external, updateable stores) from parametric weights; provenance and updatability are why RAG has been widely adopted in industry [15, 35].

9.3 Personal Language Models (PLMs) as a concrete form

A PLM can be defined as a model *bound to a person's or team's memory assets*, distinct from a general LLM. Market narratives increasingly recognize PLMs and personal AI stacks as a separate track favoring privacy, precision, and control, indicating investor and customer pull for *owned memory* powering personalized AI [13].

10 From centralized to decentralized federated learning

10.1 Centralized FL recap

The original FL recipe (FedAvg) coordinates many clients through a central server that aggregates local updates to a shared model, reducing the need to centralize data. It is communication-efficient and robust to non-IID data, but still relies on a coordinator and associated trust, connectivity, and jurisdictional assumptions [20].

10.2 Why decentralize: gossip, swarm, DFL

Empirical and theoretical work shows *fully decentralized* learning (gossip variants, segmented gossip) can match centralized FL while removing the single point of failure and easing scaling across heterogeneous networks. The "swarm learning" line combines blockchain-based coordination with edge learning to integrate data owners without a central orchestrator—explicitly "beyond FL" [16, 41, 17].

Recent surveys consolidate the pattern: in *Decentralized Federated Learning* (DFL), participants act as both learner and aggregator, communicating peer-to-peer. This better fits cross-enterprise collaboration, multi-chain ecosystems, and Web3 incentive design, at the cost of new challenges (communication scheduling, robustness, and incentives) [19].

10.3 Memory-centric view of DFL

Seen through a memory lens, DFL is not only about training a single model—it's a protocol for *sharing improvements to collective memory* without exposing raw data. Updates and their *provenance* (who trained on what, under which constraints) become first-class artifacts to price, audit, and route—aligning naturally with tokenized coordination layers.

11 Interoperability: agents, provenance, and credentials

11.1 Agent messaging and shared semantics

Agent ecosystems need shared message semantics to collaborate across boundaries. The FIPA Agent Communication Language (ACL) provides standardized performatives (e.g., `request`, `inform`, `propose`) that remain relevant foundations for cross-platform agent interop. Recent surveys discuss newer, LLM-age protocols (e.g., MCP, ACP, A2A) and highlight the need for secure context exchange and capability discovery [10, 43, 45].

11.2 Provenance as the trust fabric

For memory to be reused safely, agents must *show their work*. The W3C PROV family defines interoperable models and serializations for stating how an artifact was produced—by which entities, activities, and agents—so consumers can assess quality and reliability. PROV has been applied to ML pipelines and multi-agent platforms, enabling audit of models, datasets, and workflow steps [36, 8, 11].

11.3 Verifiable expertise and attestations

When agents claim expertise ("I'm a certified logistics planner"), we need *verifiable* proofs. The W3C Verifiable Credentials (VC) standard and DIDs provide cryptographically verifiable, privacy-respecting credentials—already used for education and skills verification—and can naturally attach to agents and memory assets [38, 39].

11.4 Cross-chain action rails

Interoperability must cross chains as well as organizations. Intents-based execution (as on NEAR) shows a path where agents submit high-level goals resolved across multiple chains with chain abstraction—reducing UX and liquidity fragmentation for multi-agent workflows [23, 22, 30].

12 A reference architecture: Memory Assets + Federated Update Markets + Interop Rails

We propose three composable layers to make memory a first-class asset in decentralized learning and multi-agent coordination.

(A) Memory Assets. *Definition:* Addressable artifacts containing knowledge usable by agents at inference or training time: vector indexes, knowledge graphs, policy libraries, domain glossaries, or fine-tuned heads. Each asset carries **provenance** (PROV), **access policies**, and optional **verifiable credentials** (who authored/verified). RAG-style retrieval treats these as non-parametric memories; MemGPT-style agents manage multi-tier storage (short-term, recall, archival) [15].

Operational implications:

- **Ownership:** Individuals and enterprises retain ownership; assets are portable across tools via standard schemas (PROV/VC) [36].
- **Auditability:** Model outputs can be traced back to memory sources and credentials (Model Cards/Data Cards complement provenance) [21].

(B) Federated Update Markets. *Definition:* Decentralized FL rounds where participants contribute *updates* (gradients, adapters, prompts, rules) trained locally on private memory; a protocol selects and aggregates updates using peer-to-peer or blockchain-mediated schemes (gossip/swarm). Incentives (tokens, credits) reward *measured utility* (loss reduction, evaluation lift), not raw data sharing [16, 41].

Operational implications:

- **No central coordinator:** Improves resilience and cross-org fit [17].
- **Selective disclosure:** Updates can be differentially private or secure-aggregated; only model deltas and proofs move. *(Swarm Learning demonstrates the pattern in regulated settings.)* [41]

(C) Interop Rails. *Definition:* Agent-to-agent messaging that carries **intents**, **capabilities**, and **claims**, plus links to memory assets/provenance. Cross-chain execution via intents abstracts liquidity and chain choice. *Standards:* FIPA-ACL (performatives), W3C PROV (what produced what), VCs/-DIDs (who says so), and intent rails for settlement [10, 36, 23].

13 Why now? Convergence of product signals and governance

Product roadmaps from major vendors are converging on *owned memory + provenance*: durable personal context (Apple Intelligence), persistent conversational memory (ChatGPT), and enterprise guidance (NIST AI RMF; Gartner AI TRiSM) that emphasizes transparency, data protection, and governance—all of which memory/provenance primitives can satisfy [6, 26, 24, 12].

For enterprises, institutional memory loss is quantifiably costly; journalism and management literature document repeated failures due to lost knowledge. A memory-first AI program is therefore not cosmetic—it is operational risk management [9].

14 Design sketch: Putting it together for PLMs and enterprises

14.1 For individuals (PLMs)

Goal: Own your memories; let agents use them without giving them away.

Mechanics.

1. Create *personal memory assets* (calendars, domain bookmarks, project notes, transaction histories), tagged with PROV.
2. Bind to a PLM (adapter/head or retrieval index); the agent manages tiered memory (short-term/recall/archival) [27].

3. Share *results* (answers, recommendations) with embedded provenance and optional VCs attesting to skill or source trust [38].

14.2 For enterprises (institutional memory)

Goal: Convert institutional memory into interoperable assets that power agents across business units and partners.

Mechanics.

1. Curate organizational memory (FAQs, SOPs, incident reviews) into addressable assets with PROV/Model Cards/Data Cards [21].

2. Join federated rounds with partners; exchange *updates*, not data (DFL/-gossip/swarm) [16, 41].

3. Use interop rails (FIPA-style messages, VCs/DIDs) for agent coordination and access control; settle multi-chain tasks via intents rails where applicable [10, 37, 23].

15 Evaluation: What to measure

- **Memory utility:** Lift from memory assets vs. baseline parametric model (RAG metrics; long-term conversation benchmarks) [15, 1].
- **Collective update quality:** Improvement from federated updates; robustness under heterogeneity (compare DFL vs. FL) [17].
- **Interop correctness:** Rate of successful multi-agent plans; provenance completeness (PROV coverage) and credential verification rates [36].
- **Governance readiness:** Alignment to NIST AI RMF/AI TRiSM controls (traceability, transparency, data protection) [24, 12].

16 Related work

Memory architectures. MemGPT and subsequent work argue that tiered memory and memory management primitives are essential for long-term, multi-turn agents [27].

RAG and non-parametric memory. RAG formalizes combining parametric and non-parametric memory and is widely adopted in enterprise stacks; recent coverage emphasizes traceability/citation [15, 35].

Decentralized learning. Gossip/DFL surveys and empirical studies show decentralized learning as a viable, sometimes superior, alternative to coordinator-based FL; swarm learning demonstrates blockchain-mediated coordination in regulated contexts [16, 19, 41].

Provenance and credentials. PROV, Model Cards, and Data Cards supply the documentation substrate; VCs/DIDs bring cryptographic verification of claims about agents and assets [36, 21].

Agent interop. FIPA ACL remains a baseline for communicative acts; modern surveys highlight emergent protocols for LLM-age agents and the need for secure context exchange [10, 45].

17 Limitations and open questions

- **Economic design.** How to price memory *access* vs. *updates*? How to prevent "poisoned" updates or provenance spam? (NIST's RMF profiles for GenAI provide partial guidance.) [24]
- **Privacy and policy.** Even when sharing *updates*, leakage risks remain; aligning with enterprise governance (AI TRiSM) requires enforcement and auditability [12].
- **Standard gaps.** PROV covers provenance, VCs cover claims, but end-to-end schemas for *agentic* workflows (tools, contexts, memories) are still emerging [44].

18 Conclusion

If models are the *engines*, memory is the *fuel system*: who owns it, how it's routed, and how cleanly it burns determine performance and trust. PLMs and organizational knowledge systems make memory *owned*; RAG and MemGPT-style architectures make it *usable*; PROV/VCs make it *auditable*; and DFL/gossip/swarm make it *shareable without exposure*. With intent-based cross-chain rails, agents can act across ecosystems while citing where their knowledge came from. Treating **collective memory as a first-class asset** aligns technical feasibility with enterprise governance and Web3 market design.

19 Acknowledgements

Draft prepared for discussion; all errors are ours.

Note on terminology. We use "PLM" as "personal language model" bound to a person's/organization's memory assets, distinct from "pre-trained language model."

References

[1] ACL. Evaluating Very Long-Term Conversational Memory of LLM Agents. In *Proceedings of the 62nd Annual Meeting of the Association for Compu-*

tational Linguistics, pages 1–15, 2024. URL https://aclanthology.org/2024.acl-long.747.pdf.

[2] ACT-R. ACT-R About. URL https://actr.psy.cmu.edu/about/. Accessed: 2025-06-16.

[3] ACT-R. ACT-R Tutorial, 2008. URL https://escholarship.org/content/qt2fp4d5mf/qt2fp4d5mf_noSplash_bc8f37d509d4a6e49192d45466f81edd.pdf. Accessed: 2025-06-16.

[4] ACT-R. Understanding ACT-R - An Outsider's Perspective. *arXiv preprint arXiv:1306.0125*, 2013. URL https://arxiv.org/pdf/1306.0125.

[5] John R. Anderson, Daniel Bothell, Michael D. Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. An Integrated Theory of the Mind. *Psychological Review*, 111(4):1036–1060, 2004. doi: 10.1037/0033-295X.111.4.1036.

[6] Apple. Introducing Apple Intelligence for iPhone, iPad, and Mac, June 2024. URL https://www.apple.com/newsroom/2024/06/introducing-apple-intelligence-for-iphone-ipad-and-mac/. Accessed: 2025-06-16.

[7] Ars Technica. Chatgpt can now remember and reference all your previous chats, April 2025. URL https://arstechnica.com/ai/2025/04/chatgpt-can-now-remember-and-reference-all-your-previous-chats/. Accessed: 2025-06-16.

[8] Badrish Chandramouli, Jiannan Wang, Justin Levandoski, Ahmed Eldawy, et al. Capturing end-to-end provenance for machine learning pipelines. *ScienceDirect*, 158:103–115, 2024. doi: 10.1016/j.ins.2024.01.015.

[9] Financial Times. The Value of Institutional Memory, 2023. URL https://www.ft.com/content/29df737f-8e7e-47c1-a490-dfc6a6ce05c2. Accessed: 2025-06-16.

[10] FIPA. Agent Communication Language, 2000. URL https://www.fipa.org/specs/fipa00018/OC00018.pdf. Accessed: 2025-06-16.

[11] Michael Friedman, Jan Van den Bussche, and Stijn Vansummeren. Provenance-Based Interpretation of Multi-Agent Information. In *Proceedings of the 2020 USENIX Conference on Theory and Practice of Provenance*, pages 1–15, 2020. URL https://www.usenix.org/system/files/tapp2020-paper-friedman.pdf.

[12] Gartner. Tackling Trust, Risk and Security in AI Models, 2023. URL https://www.gartner.com/en/articles/ai-trust-and-ai-risk. Accessed: 2025-06-16.

[13] GlobeNewswire. From One-Size-Fits-All to Custom Built Models: Personal AI Charts an Alternative Path No LLM Required, June 2025. URL https://www.globenewswire.com/news-release/2025/06/04/3093589/0/en/From-One-Size-Fits-All-to-Custom-Built-Models-Personal-AI-Charts-an-Alternative-Path-N html. Accessed: 2025-06-16.

[14] Maurice Halbwachs. Maurice Halbwachs - "Historical Memory and Collective Memory" - summary, March 2012. URL https://culturalstudiesnow.blogspot.com/2012/03/maurice-halbwachs-historical-memory-and.html. Accessed: 2025-06-16.

[15] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *arXiv preprint arXiv:2005.11401*, 2020. URL https://arxiv.org/abs/2005.11401.

[16] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Decentralized learning works: An empirical comparison of gossip and centralized approaches. *ScienceDirect*, 54(3):389–400, 2020. doi: 10.1016/j.ins.2020.04.032.

[17] Yang Liu, J. Q. James, Jiawen Kang, Dusit Niyato, and Shengli Zhang. Decentralized Federated Learning: Fundamentals, State of the Art, Frameworks, Trends, and Challenges. *arXiv preprint arXiv:2211.08413*, 2022. URL https://arxiv.org/abs/2211.08413.

[18] MacRumors. 'Apple Intelligence' Personal AI unveiled for iPhone, iPad, and Mac, June 2024. URL https://www.macrumors.com/2024/06/10/apple-intelligence-generative-personal-ai-unveiled-for-iphone-ipad-and-mac/. Accessed: 2025-06-16.

[19] Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. Decentralized Federated Learning: A Survey and Perspective. *arXiv preprint arXiv:2306.01603*, 2023. URL https://arxiv.org/abs/2306.01603.

[20] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54, pages 1273–1282, 2017. URL https://proceedings.mlr.press/v54/mcmahan17a/mcmahan17a.pdf.

[21] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model Cards for Model Reporting. *arXiv preprint arXiv:1810.03993*, 2019. URL https://arxiv.org/abs/1810.03993.

[22] NEAR. Unpacking NEAR Intents: A Deep Dive, 2024. URL https://www.near.org/blog/unpacking-near-intents-a-deep-dive. Accessed: 2025-06-16.

[23] NEAR Protocol. Introducing NEAR Intents: A New Type of Transaction Between AI and the ..., 2024. URL https://pages.near.org/blog/introducing-near-intents/. Accessed: 2025-06-16.

[24] NIST. Artificial Intelligence Risk Management Framework: Generative AI Profile, 2023. URL https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf. Accessed: 2025-06-16.

[25] Number Analytics. Understanding Halbwachs in Cultural Memory. URL https://www.numberanalytics.com/blog/halbwachs-cultural-memory-guide. Accessed: 2025-06-16.

[26] OpenAI. Memory and new controls for ChatGPT, April 2024. URL https://openai.com/index/memory-and-new-controls-for-chatgpt/. Accessed: 2025-06-16.

[27] Joon Sung Park, Joseph O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Memgpt: Towards LLMs as Operating Systems. *arXiv preprint arXiv:2310.08560*, 2023. URL https://arxiv.org/abs/2310.08560.

[28] Alexander A. Petrov. Computationally efficient approximation of the base-level learning equation in ACT-R. In *Proceedings of the 7th International Conference on Cognitive Modeling*, pages 220–225, 2006. URL https://alexpetrov.com/pub/iccm06/PetrovICCM06.pdf.

[29] PyACT-R. Modules and buffers, and writing productions in - Colab, 2020. URL https://colab.research.google.com/github/abrsvn/pyactr-book/blob/master/notebooks/3_modules_and_buffers_and_writing_productions_in_pyactr.ipynb. Accessed: 2025-06-16.

[30] Reflexivity Research. NEAR Q4 2024 Overview, 2024. URL https://www.reflexivityresearch.com/all-reports/near-q4-2024-overview. Accessed: 2025-06-16.

[31] Semantic Scholar. An Integrated Theory of the Mind, 2004. URL https://pdfs.semanticscholar.org/2709/9ec9ea719f8fd919fb69d66af677a424143b.pdf. Accessed: 2025-06-16.

[32] SpringerLink. A Comparison of Approximations for Base-Level Activation in ACT-R. *Journal of Artificial General Intelligence*, 9(1):1–22, 2018. doi: 10.1007/s42113-018-0015-3.

[33] Niels A. Taatgen, Christian Lebiere, and John R. Anderson. Modeling paradigms in ACT-R. In *Cambridge Handbook of Computational Psychology*. Cambridge University Press, 2006. URL https://www.ai.rug.nl/~niels/publications/taatgenLebiereAnderson.pdf.

[34] The Verge. Chatgpt's AI 'memory' can remember the preferences of ... - The Verge, April 2024. URL https://www.theverge.com/2024/4/29/24144680/chatgpt-plus-memory-chatbot-subscription-details-preferences-personal-assistant. Accessed: 2025-06-16.

[35] TIME. Patrick Lewis, 2023. URL https://time.com/7012883/patrick-lewis/. Accessed: 2025-06-16.

[36] W3C. PROV-Overview - World Wide Web Consortium (W3C), 2013. URL https://www.w3.org/TR/prov-overview/. Accessed: 2025-06-16.

[37] W3C. Decentralized Identifiers (DIDs) v1.0, 2022. URL https://www.w3.org/TR/did-1.0/. Accessed: 2025-06-16.

[38] W3C. Verifiable Credentials Data Model v2.0 - World Wide Web Consortium (W3C), 2023. URL https://www.w3.org/TR/vc-data-model-2.0/. Accessed: 2025-06-16.

[39] W3C-CCG. Verifiable Credentials for Education, Employment, and Achievement Use Cases. URL https://w3c-ccg.github.io/vc-ed-use-cases/. Accessed: 2025-06-16.

[40] James P. Walsh and Gerardo Rivera Ungson. Organizational memory. *Academy of Management Review*, 16(1):57–91, 1991. URL https://skat.ihmc.us/rid%3D1255442505000_1811726224_21686/Organizationa1%20Memory%20-%20Walsh.pdf.

[41] Stefanie Warnat-Herresthal, Hartmut Schultze, Krishnaprasad Lingadahalli Shastry, Sathyanarayanan Manamohan, Saikat Mukherjee, Vishesh Garg, Ravi Sarveswara, Kristian Händler, Peter Pickkers, N. Ahmad Aziz, Sofia Ktena, et al. Swarm Learning for decentralized and confidential clinical machine learning. *Nature*, 594(7862):265–270, 2021. doi: 10.1038/s41586-021-03583-3.

[42] Wikipedia. ACT-R - Wikipedia, 2024. URL https://en.wikipedia.org/wiki/ACT-R. Accessed: 2025-06-16.

[43] Michael Wooldridge, Nicholas R. Jennings, and David Kinny. FIPA: A Standard for Agent Interoperability. In *Proceedings of the 5th International Conference on Autonomous Agents*, pages 1–10, 2000. URL https://lia.disi.unibo.it/books/woa00/pdf/R2.pdf.

[44] Wei Zhang, Chen Li, Hao Wang, Ming Zhou, et al. PROV-AGENT: Unified Provenance for Tracking AI Agent Interactions in ... *arXiv preprint arXiv:2508.02866*, 2024. URL https://arxiv.org/pdf/2508.02866v3.

[45] Wei Zhang, Chen Li, Hao Wang, Ming Zhou, et al. A Survey of Agent Interoperability Protocols: Model Context Protocol, Agent Communication Protocol, and Agent to Agent Protocol. *arXiv preprint arXiv:2505.02279*, 2025. URL https://arxiv.org/html/2505.02279v1.