BEN LABASCHIN

Agent Memory

From a timeout to a managed commitment.
How evidence becomes durable, revisable guidance.

THE STORY IN THREE MOVES

01 FAIL
ambiguous timeout

02 ROUTE
trusted evidence

03 GOVERN
managed commitment

Head of AI and Engineer @ Workhelix · O'Reilly Author · econoben.dev

ABOUT ME

I’m Ben Labaschin.

ENGINEER

Head of AI and Engineering @ Workhelix.

I build stateful systems where models, tools, data, and evidence work together.

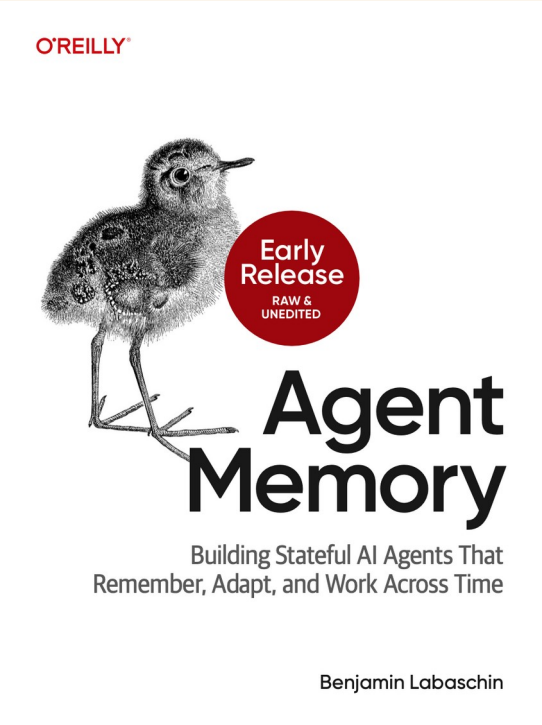
AUTHOR

Author of *Agent Memory* for O’Reilly
A practical guide to agents that remember, adapt, and work across time.

I focus on memory, retrieval, lifecycle policy, and governed adaptation. · econoben.dev

ABOUT ME

Agent Memory Early Release



Something Is Off...

```
def submit_purchase(order_id: str, payload: dict) -> VendorResponse:
    response = http_client.post(
        "https://buythings.important_vendor.com/purchases",
        json=payload,
        timeout=2.0,
    )
    response.raise_for_status()
    return VendorResponse.from_json(response.json())
```

The code has symptoms…

1. **It returns a timeout**
2. **It’s deceptively simple**
3. **No Error Logs**

The fix looks right.

```
def submit_purchase(order_id, payload):
    for attempt in range(3):
        try:
            return post("/purchases", json=payload, timeout=2.0)
        except TimeoutError:
            if attempt == 2: raise
            continue
```

What does the timeout tell us?

Retry budget

Three attempts. Bounded timeout.

Familiar pattern

Reads often tolerate this.

Missing state

A billable write may already have committed.

The timeout did not mean failure.

01 → 02 → 03 → 04

Request accepted

The vendor commits the charge.

Response lost

The agent observes only a timeout.

Blind retry

A fresh request repeats the operation.

Duplicate charge

The observation was ambiguous. The side effect was not.

A missing response is evidence about the network—not proof about the write.

SAME MODEL · DIFFERENT HARNESS

The harness changes the safe action.

BLANK-SLATE

New request.

Retry after timeout.

Fresh key each attempt

No state reconciliation

Duplicate charge possible

→

MEMORY-CAPABLE

Reuse the intent.

Reconcile before retry.

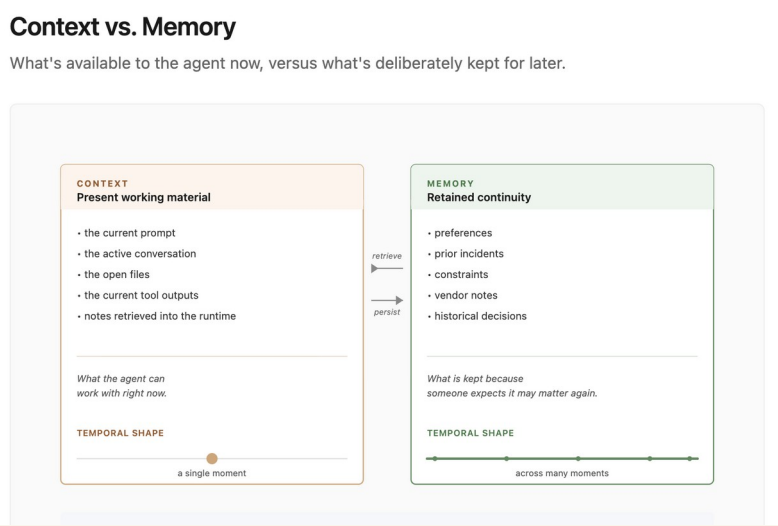
Stable idempotency key

Authoritative state check

One intended purchase

Same model. Different state, evidence, and tools.

Context disappears. Memory is carried.



CONTEXT

Available to the model now—then rebuilt.

MEMORY

Deliberately retained so it can affect later work.

HARNESS

Persistence and retrieval move evidence across boundaries.

The useful answer was already in four places.

01 → 02 → 03 → 04

Vendor docs

Define idempotency and billable semantics.

Incident + RCA

Explain the duplicate charge and lost response.

Current runbook

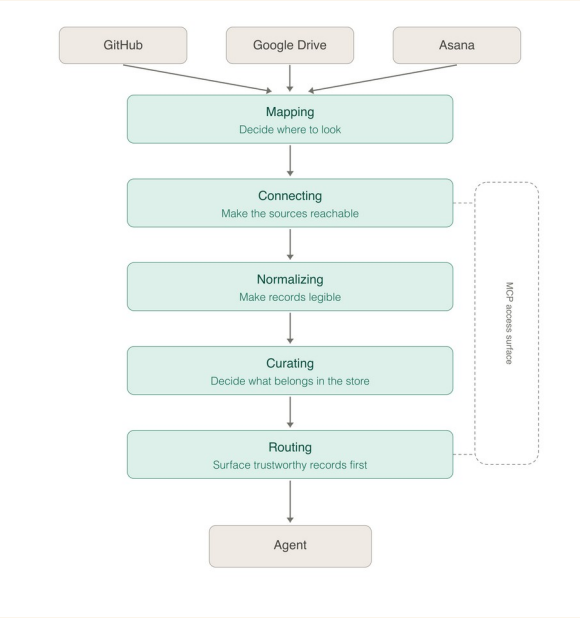
Assigns reconciliation and operational scope.

Engineer correction

Makes the stable-key rule explicit.

Different sources support different claims; access is not authority.

Five layers turn sources into usable evidence.



ACCESS

Mapping and connecting make organizational records reachable.

GOVERNANCE

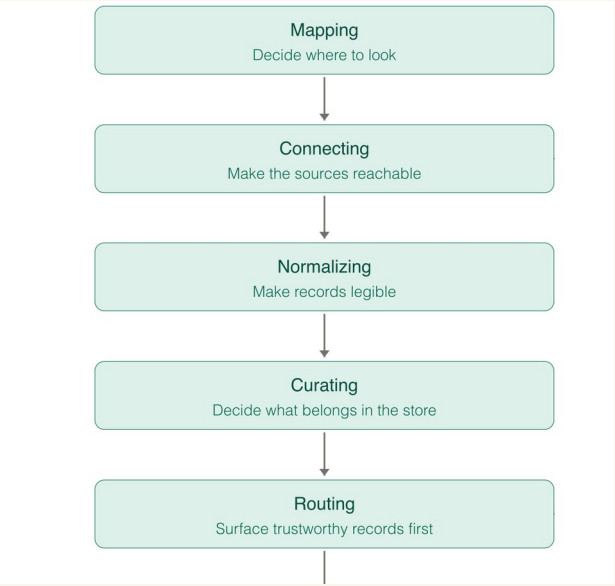
Normalization and curation attach comparable meaning, scope, and authority.

USE

Routing selects the evidence that belongs in this task now.

THREE DISTINCT GATES

Reachable is not admissible. Admissible is not relevant.



REACHABLE

Can the system query this source?

ADMISSIBLE

Does the record satisfy the store’s methodology and authority rules?

RELEVANT

Does it belong in this billing decision? A CI retry ticket does not.

Purchase calls must reuse the purchase intent’s idempotency key.

Engineer clarification · checked against vendor semantics and the current runbook

Useful now. Not automatically entitled to persist.

Would you save this?

01
USE NOW
Apply to this task.

02
PROPOSE
Open a candidate.

03
COMMIT
Create durable memory.

Useful now ≠ entitled to persist.

One encounter creates three candidates.

The three memory types in the vendor timeout case

Each memory type answers a different guiding question.
The cleaner the boundary, the more deliberately the system can be designed.

GUIDING QUESTION	MEMORY TYPE	WHAT IT PULLED IN THIS CASE
What is true here?	**Semantic** FACTS	• Vendor X timeouts are ambiguous and do not reliably indicate failed purchase creation • Purchase creation is a billable vendor action • The integration supports idempotency keys
What happened before?	**Episodic** EVENTS	• Incident-2071 documenting a prior duplicate-charge incident • A previous retry-based fix replayed already-completed purchase requests • The earlier invoice spike that finance flagged
How should the agent act?	**Procedural** ROUTINES	• Do not retry billable purchase requests on timeout without reconciliation or idempotency • Check organizational memory before patching vendor integrations • Reconcile vendor-side state before replaying an ambiguous request

SEMANTIC

Stable-key reuse is required.

EPISODIC

An engineer corrected the retry assumption.

PROCEDURAL

Reconcile an ambiguous purchase before retrying.

A saved memory is a forecast.

Durable memory begins only when a future-use claim survives an explicit decision.

01 **MemoryCandidate**

Proposed content + future job

What future job?

→

02 **RetentionDecision**

Policy + authority decide

Should policy commit?

→

03 **ManagedMemory**

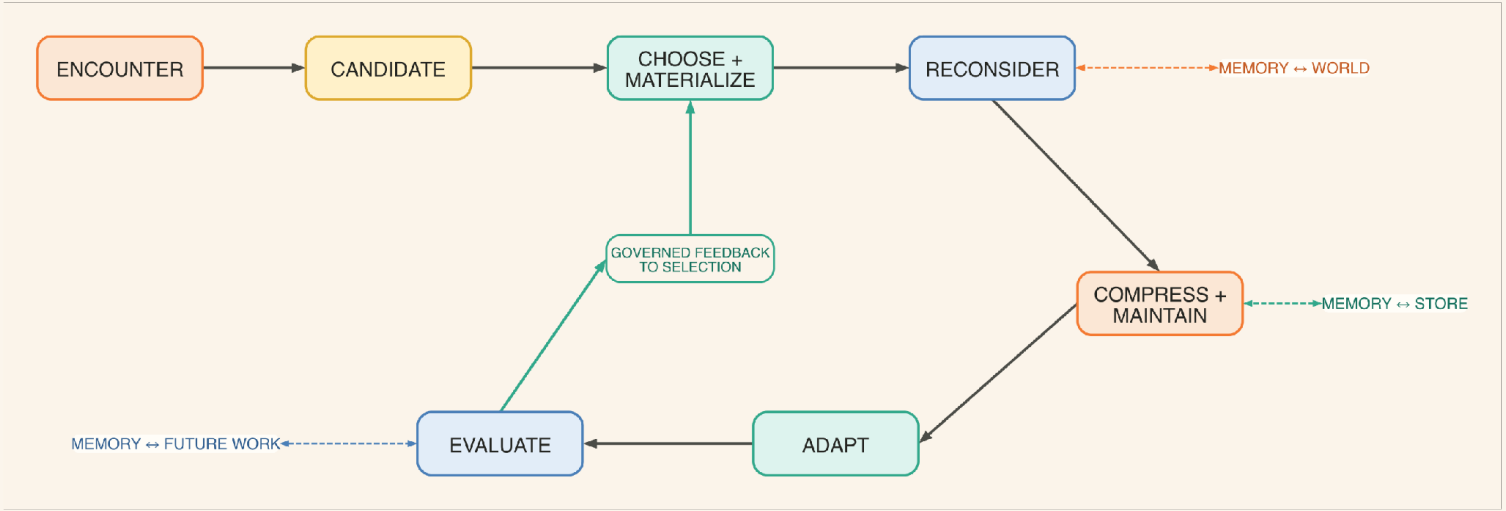
Owned, versioned, revisable

Who will maintain it?

Candidates remain inactive until the decision materializes them.

PROPOSE · DECIDE · OWN

Selection starts a lifecycle.



World, store, and future work keep reopening the commitment.

Then the runbook changes.

01

CONSOLIDATE

Semantic fact
The new source corroborates the stable-key rule.

02

DEMOTE

Correction episode
History stays true; independent retrieval value falls.

03

SUPERSEDE

Old procedure
A replacement now governs current action.

Same event. Different lifecycle actions.

A retention horizon reopens the decision.

TTL

Storage clock.

Expire after time.

Controls payload lifetime

Can remove without reconsideration

Useful for storage hygiene

→

RETENTION HORIZON

Decision trigger.

Reopen the commitment.

Governing source changes

Contradiction or redundancy

Use-and-cost evidence

One manages duration. The other schedules judgment.

A CONDITIONAL DECISION

Choose the action—not a memory score.

01

CONSTRAINTS

Is this action allowed?
Authority · validity · obligations

02

CURRENT EVIDENCE

What changed?
World · store · future work

03

SMALLEST ACTION

Smallest justified change.
Demote · consolidate
Supersede · delete

Signals rank options inside the constraints; they do not replace them.

Forgetting is not deletion.

MEMORY	ACTION	WHAT CHANGES	WHAT REMAINS
Semantic fact	Consolidate	Independent retrieval	Source-linked claim
Correction episode	Demote	Ranking priority	History + provenance
Old procedure	Supersede	Active authority	Replacement link
Deletion	Not used	—	No universal expiry

Remove the smallest property the decision requires.

Compression preserves the future job.

UNSAFE ROLLUP v0

Shorter text.
Fails validation.

Drops the EU exception

Blurs authority

Breaks claim-to-source lineage

→

VALIDATED REPLACEMENT v2

Future-use contract.
Controlled cutover.

Keeps the exception

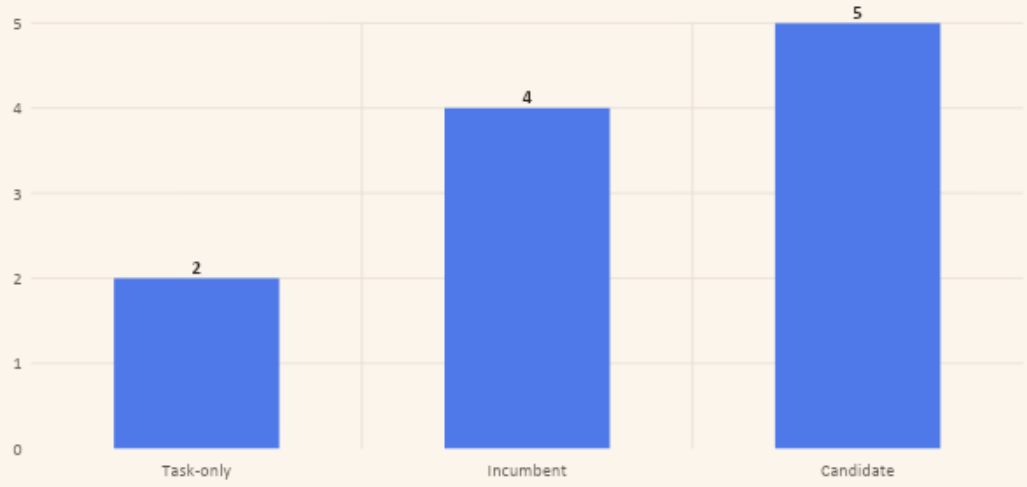
Names dependencies

Carries a rebuild path

v2 becomes current; v1 remains historical.

ILLUSTRATIVE OFFLINE REPLAY

Prove that remembering changed the action.



2 / 5 → 5 / 5

cases passed

Candidate: +0.2 cost vs incumbent
Direct source: 5/5 at cost 5.0
Activation: awaiting purchasing owner

Deterministic five-case replay · not a benchmark

The next timeout ends differently.

01 → 02 → 03 → 04

Same intent

The purchase keeps a stable operation identity.

Same key

A replay cannot create a second charge.

Reconcile first

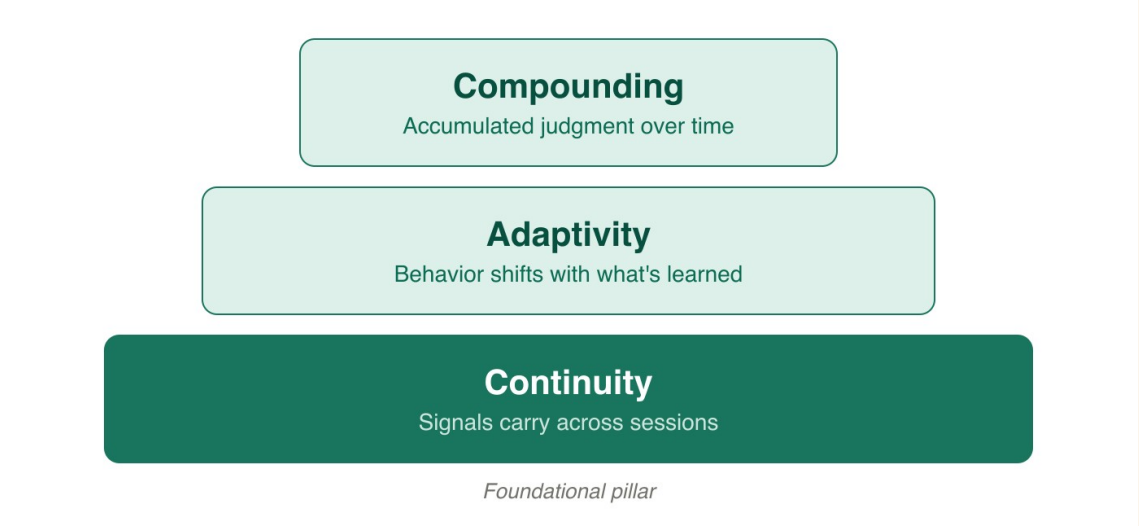
Check authoritative state before another write.

One charge

Return the confirmed purchase.

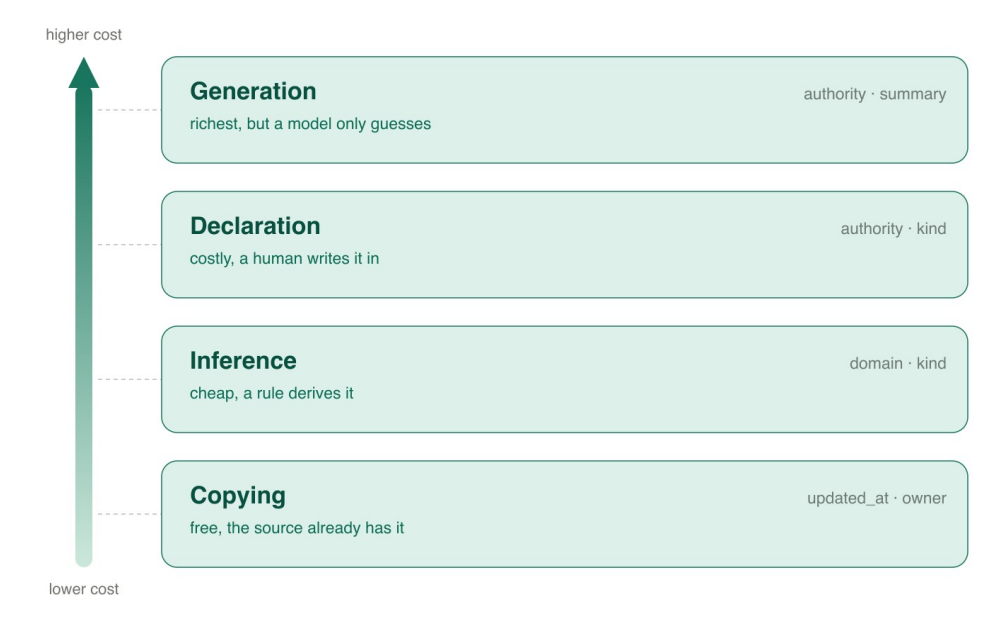
Route the evidence. Govern the lifecycle. Measure the action.

Continuity makes adaptation possible.



Continuity → adaptivity → compounding.

Richer meaning costs more to create—and to trust.



COPY

Cheap, direct, and source-grounded.

INFER

Useful and scalable, but imperfect.

DECLARE + GENERATE

Richer meaning still does not create organizational authority.

Raw records become routed evidence.

A common envelope helps only when each record remains semantically legible.

01

RawRecord

Identity, source, title, URL, text, metadata

Can we parse it?

→

02

MemoryRecord

Domain, owner, kind, authority, freshness

Can we compare it?

→

03

RoutedEvidence

Task-scoped and authority-ranked

Does it matter now?

Technical structure is necessary. Semantic structure makes routing principled.

SHAPE · MEANING · DECISION

Evaluation can propose review—not grant authority.

STATE	EVIDENCE	WHAT OPENS	WHAT CREATES MEMORY
Policy 7	2 supports < 3	Nothing	—
Policy 8b	2 supports ≥ 2	Proposal + owner review	Owner approval
Evaluation	Candidate 5/5	Policy review	Never self-activation
Current status	Hard constraints pass	Awaiting owner	No activation yet

Policy adapts only through a versioned, governed lane.

Any questions?

What should your agent remember—and who will own the answer?

BEN LABASCHIN

econoben.dev · O'Reilly Author

Thank you.

Route the evidence. Govern the lifecycle. Measure the action.

BEN LABASCHIN

econoben.dev · O'Reilly Author