

# Metaphors for Memory: Charting a Design Space of AI Memory Tools and Interfaces



**Munyeong Kim**



**Michalis Famelis**



**Ian Arawjo**

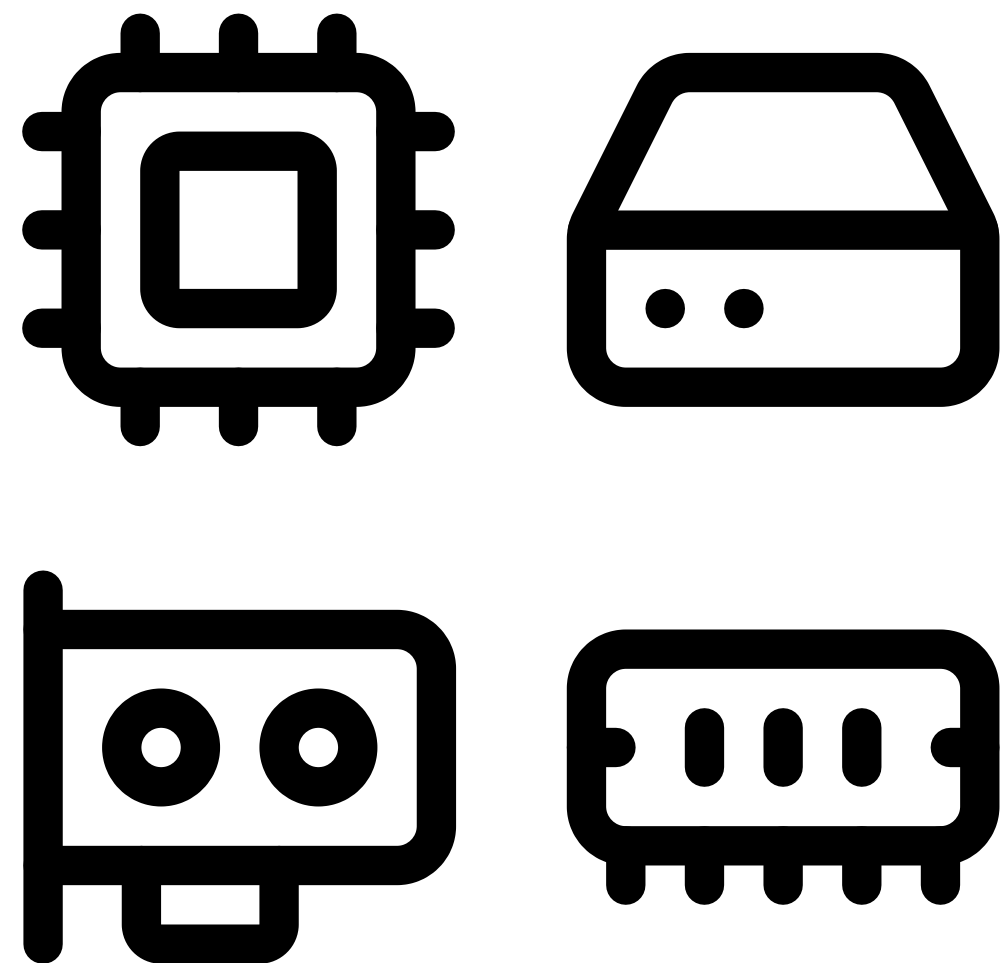
# What is AI memory?

# “AI memory” still lacks clear consensus

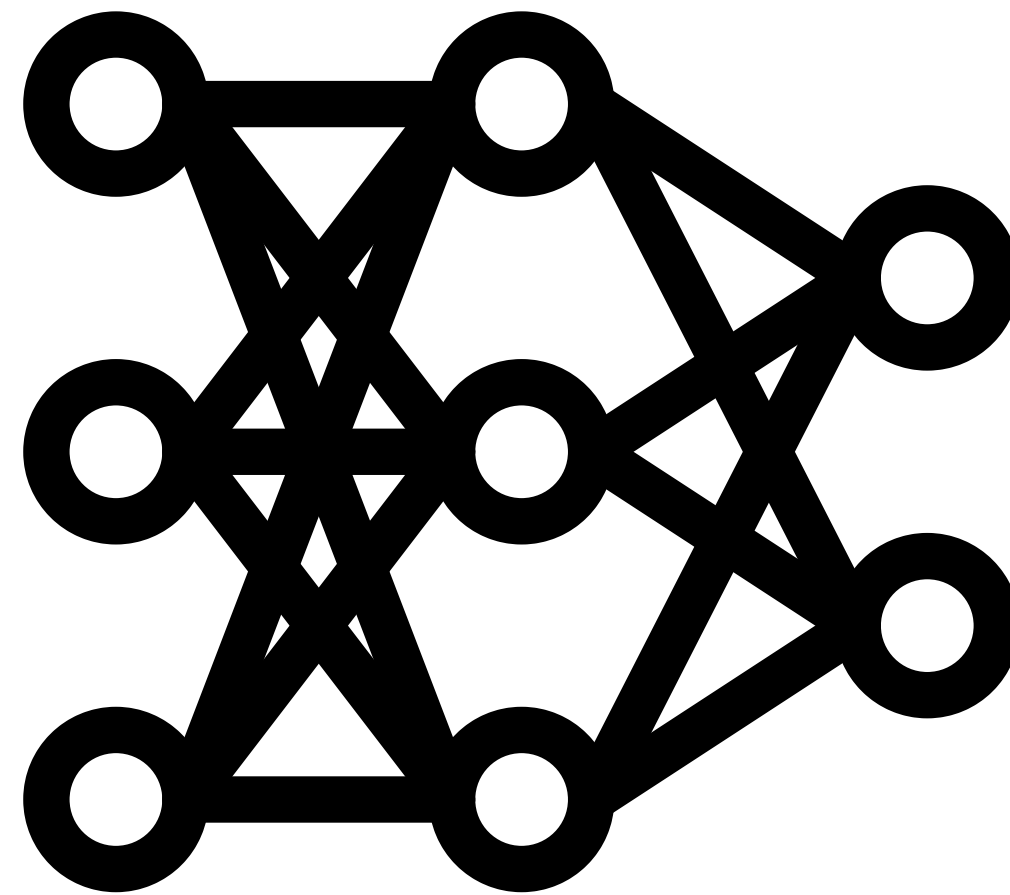
The term can range from hardware-level memory to LLM’s implicit parametric memory and explicit written memory (e.g. prompts, files).

From

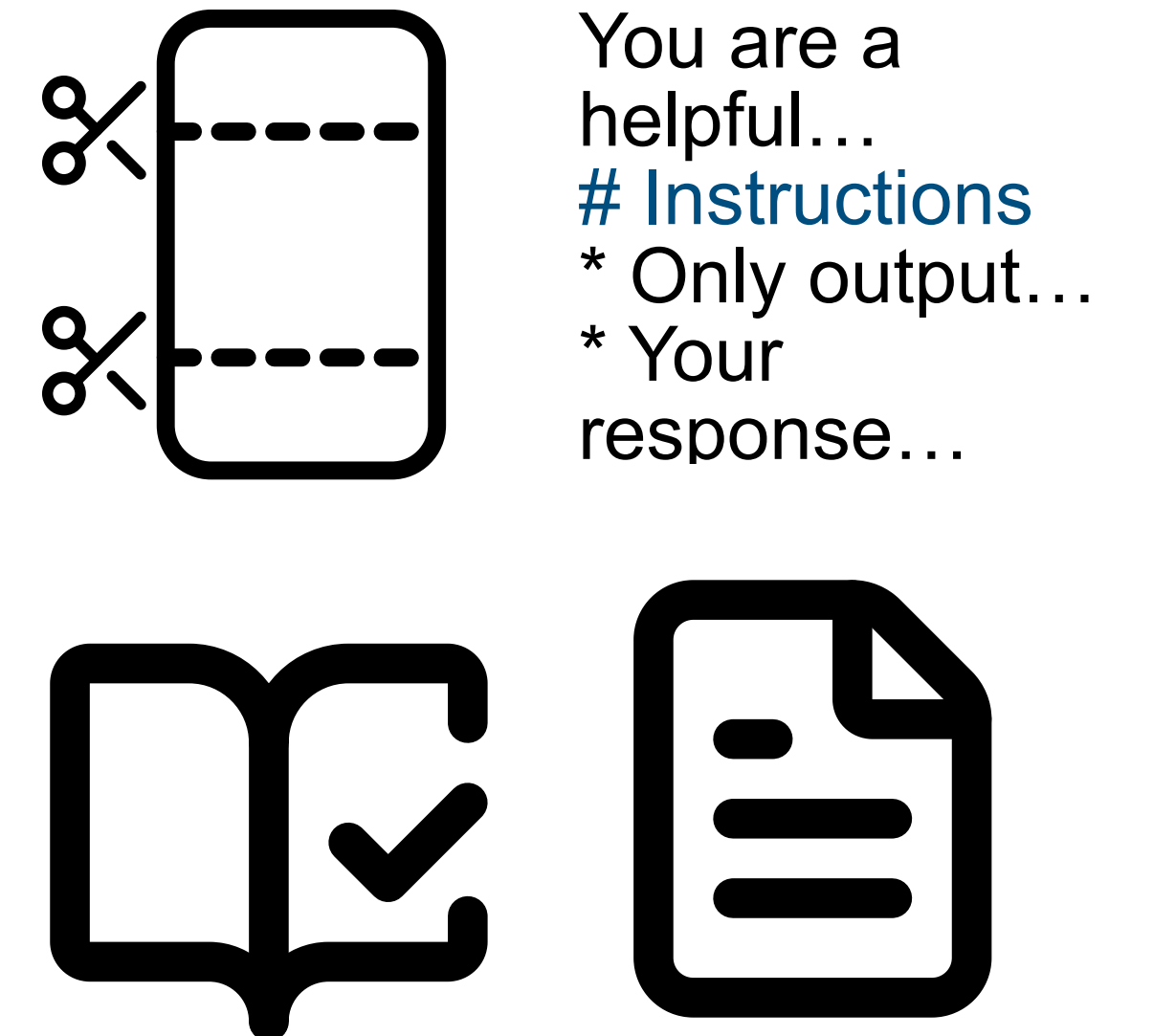
To



**Hardware Memory**



**Parametric (Implicit) Memory**



**Written (Explicit) Memory**

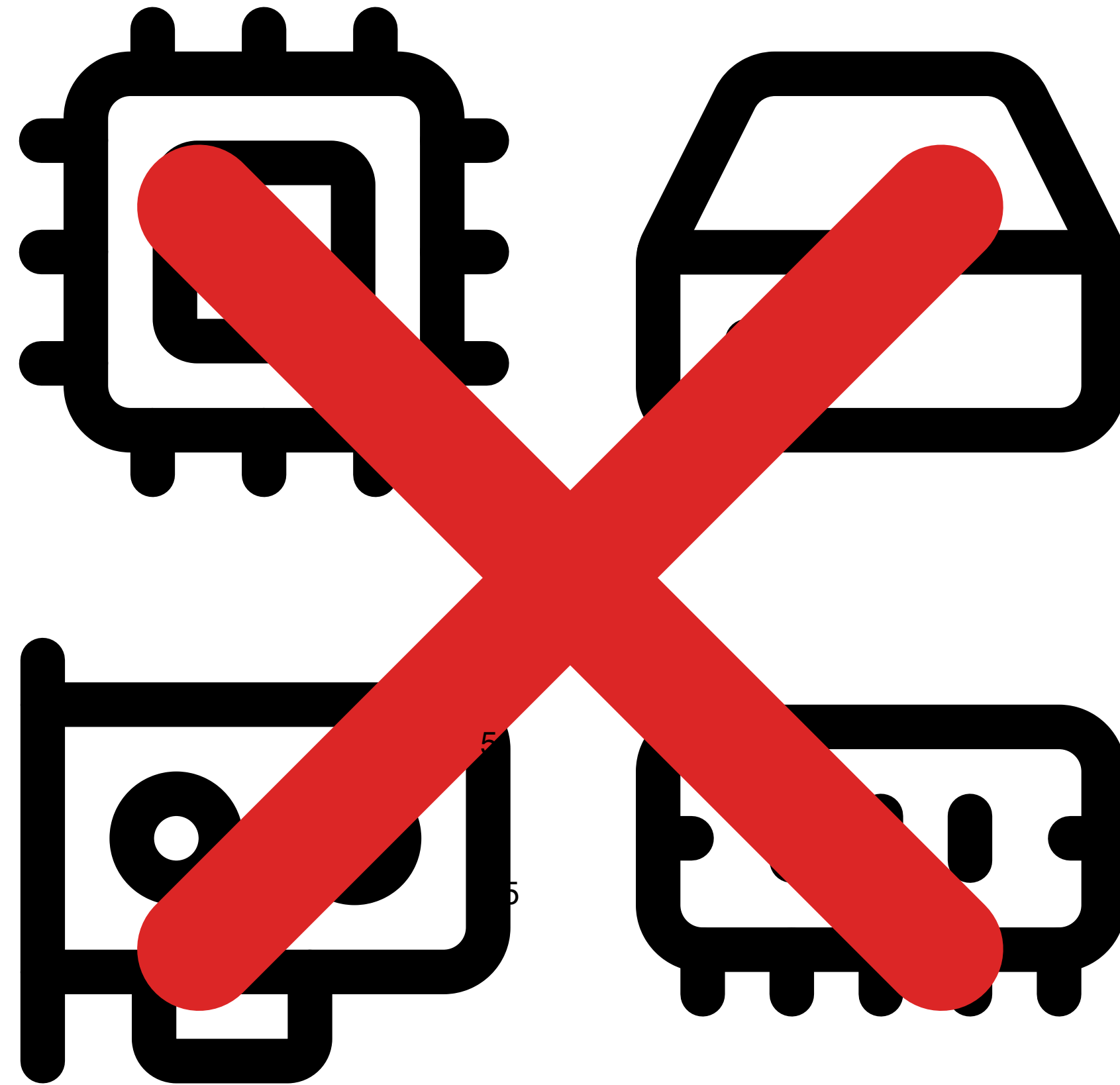
You are a helpful...  
# Instructions  
\* Only output...  
\* Your response...

# **Our Definition of AI memory**

Information that an agent retains through interaction, reorganizes as new information emerges, and reuses when relevant.

# Hardware-level memory is outside our scope.

We focus on “information” used and updated through human-AI interaction.

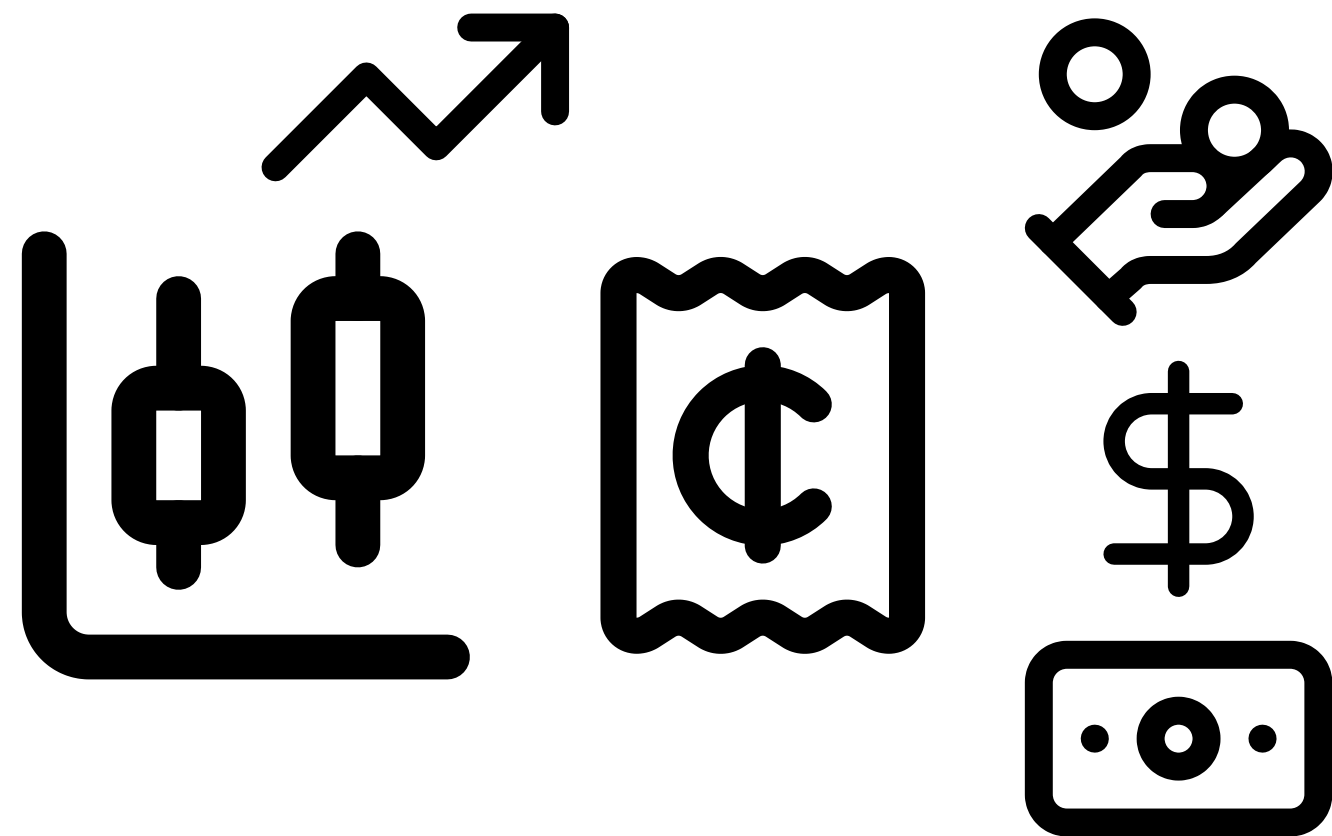


# **Why Does AI Memory Matter for Designing Interactive Systems?**

# Different Goals Require Different Memory Designs

Agents with different roles need different memory structures, operations, and interactions.

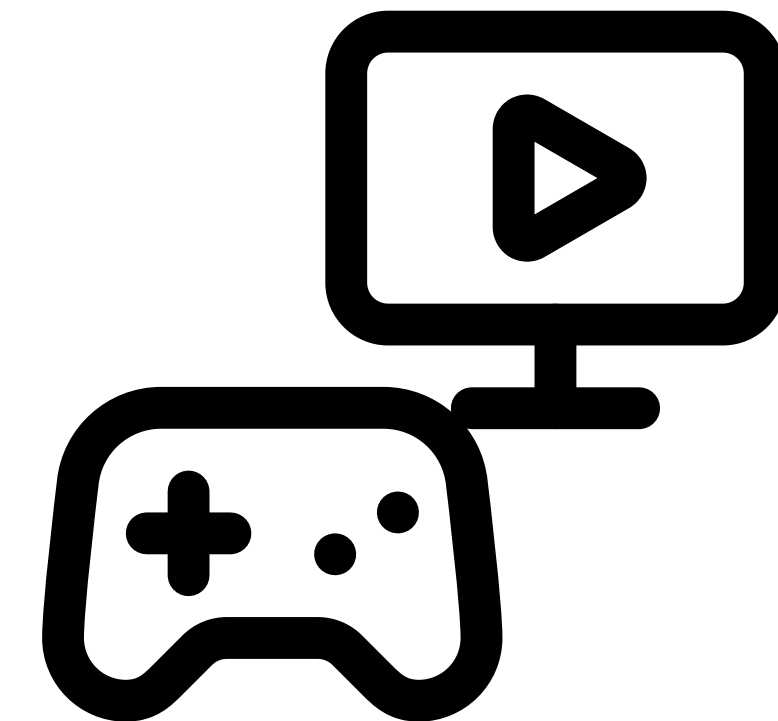
**Investment Team Agent**



**Clinical Support Agent**



**Role-Playing Agent**



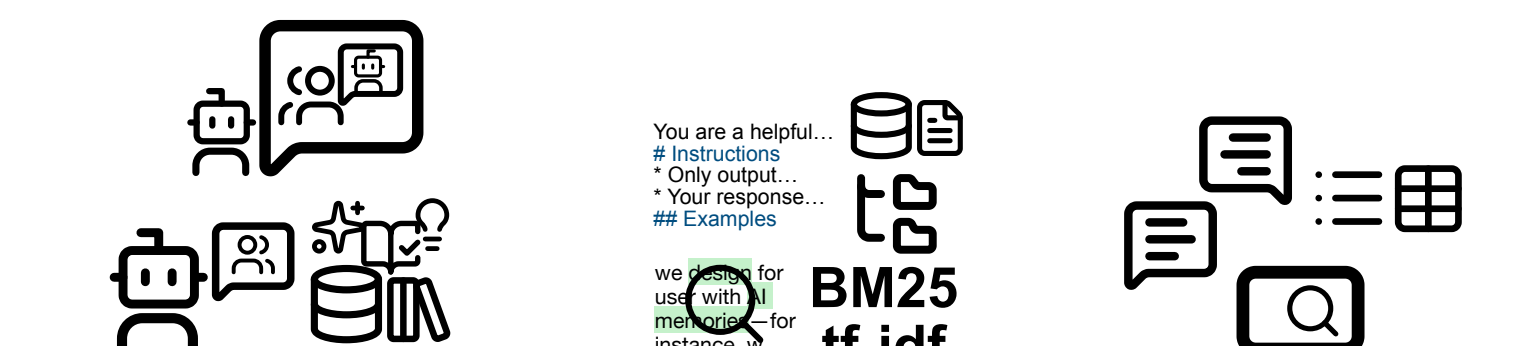
# Different Goals Shape Different Design Choices

We need to make design choices across architecture, operations, and interactions, based on what the system needs to prioritize.

## Investment Team Agent

## Clinical Support Agent

## Role-Playing Agent



Purpose of Memory system

Architecture

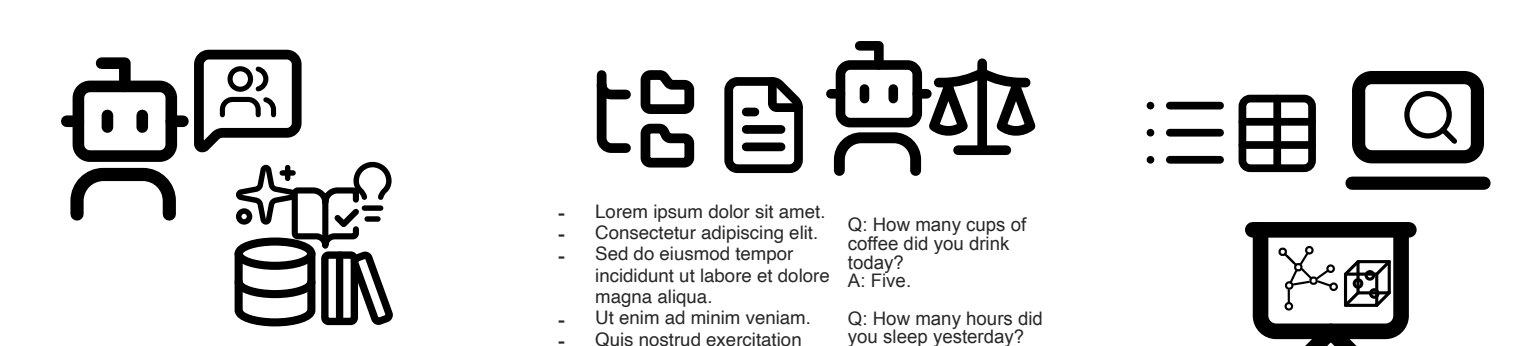
Interface

Interaction

Operations

we design for use with BM25 tf-idf

You are a helpful...  
# Instructions  
\* Only output...  
\* Your response...  
## Examples



Purpose of Memory system

Architecture

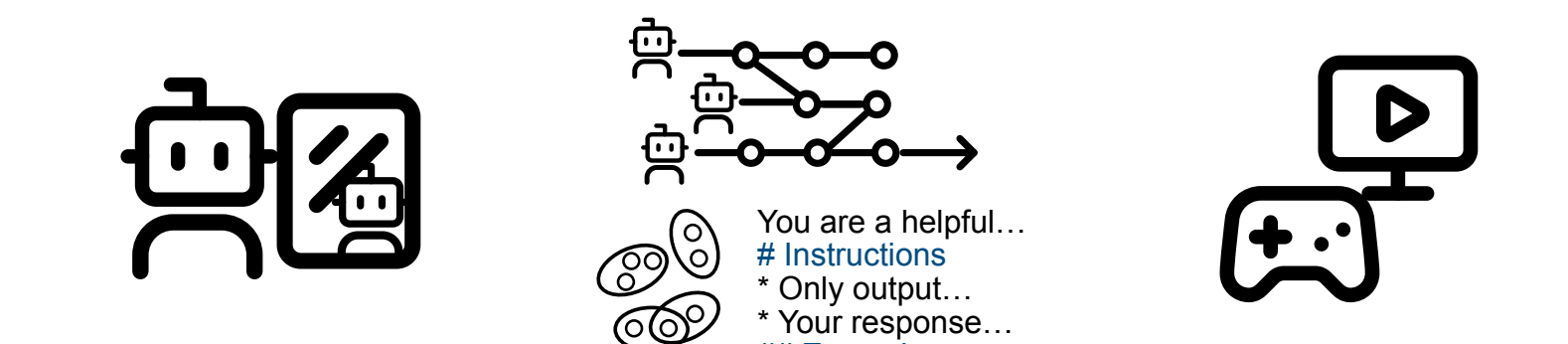
Interface

Interaction

Operations

Q: How many cups of coffee did you drink today?  
A: Five.  
Q: How many hours did you sleep yesterday?  
A: [Conversation ends.]

Q: How many cups of coffee did you drink today?  
A: Five.  
Q: How many hours did you sleep yesterday?  
A: [Conversation ends.]



Purpose of Memory system

Architecture

Interface

Interaction

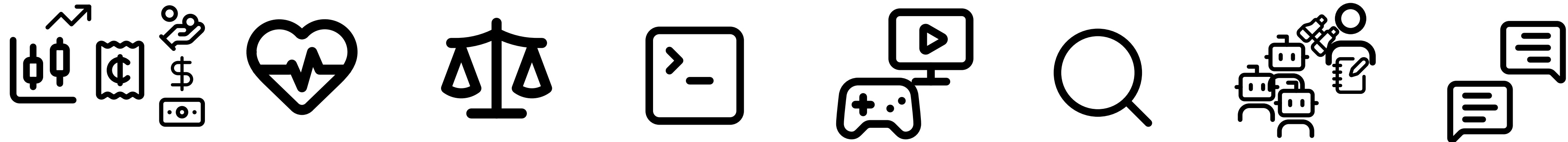
Operations

You are a helpful...  
# Instructions  
\* Only output...  
\* Your response...  
## Examples

# AI Memory Design Choices Matter

To support the interactions we want, a memory system needs the right operations and an architecture that can support them.

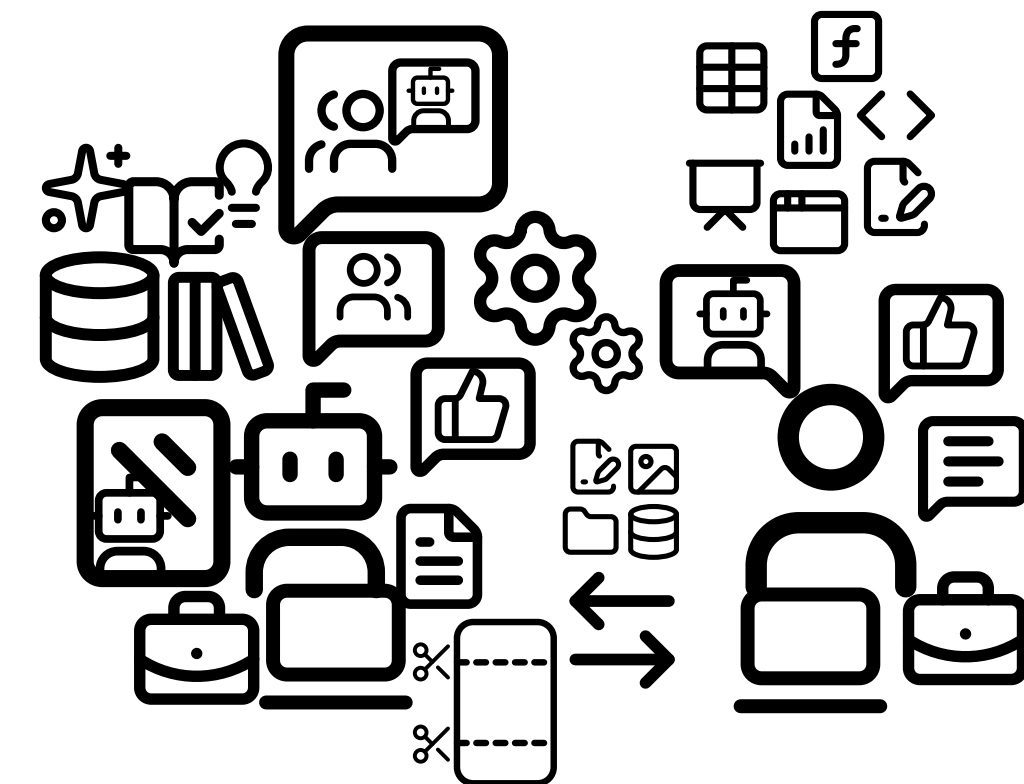
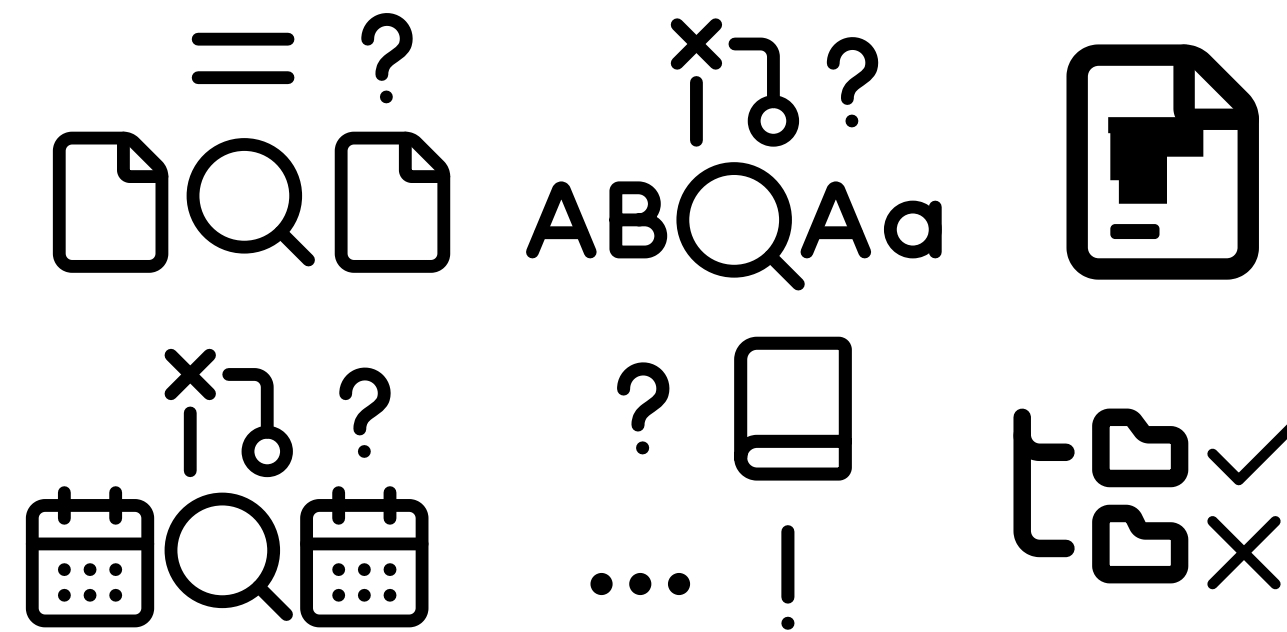
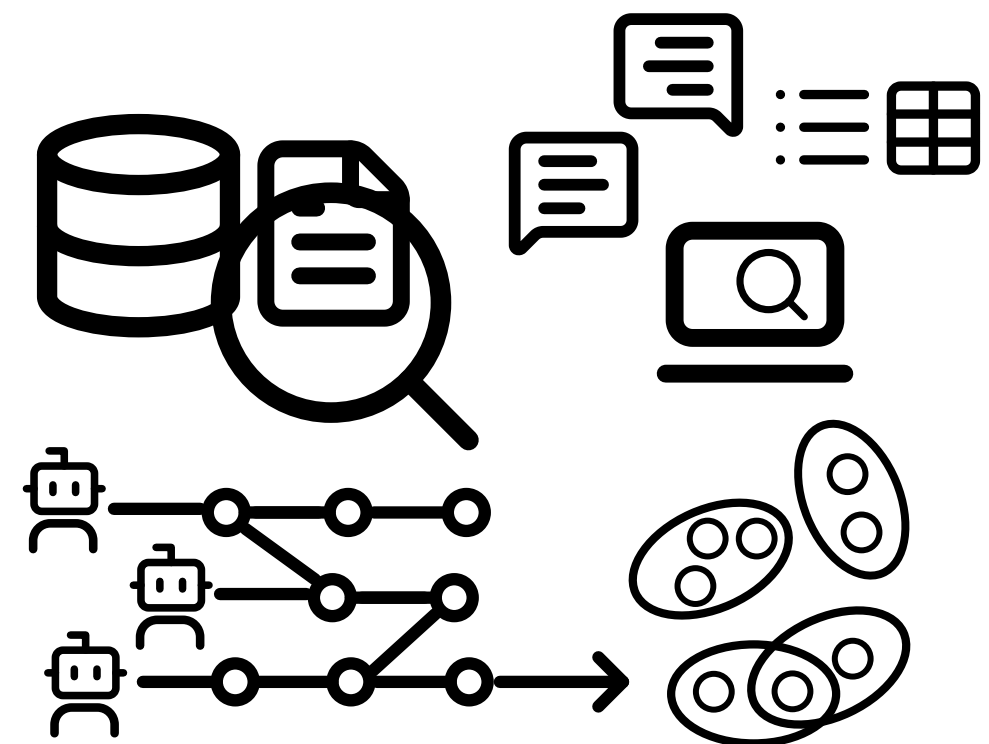
## Design goal of systems



Architecture

Operations

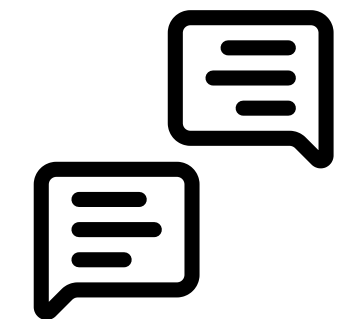
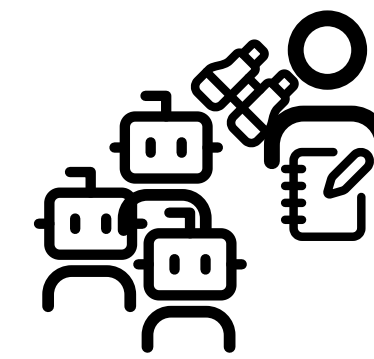
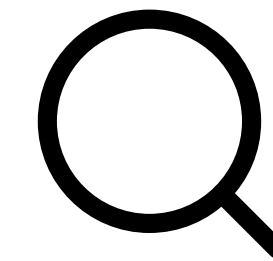
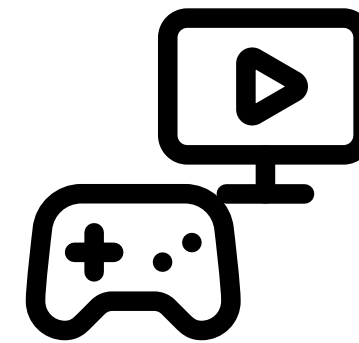
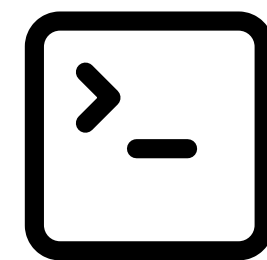
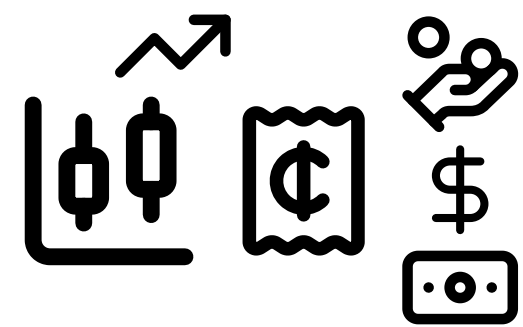
User interaction



# AI Memory Design Choices Matter

Misaligned designs may appear to work, but quietly affect user interaction.

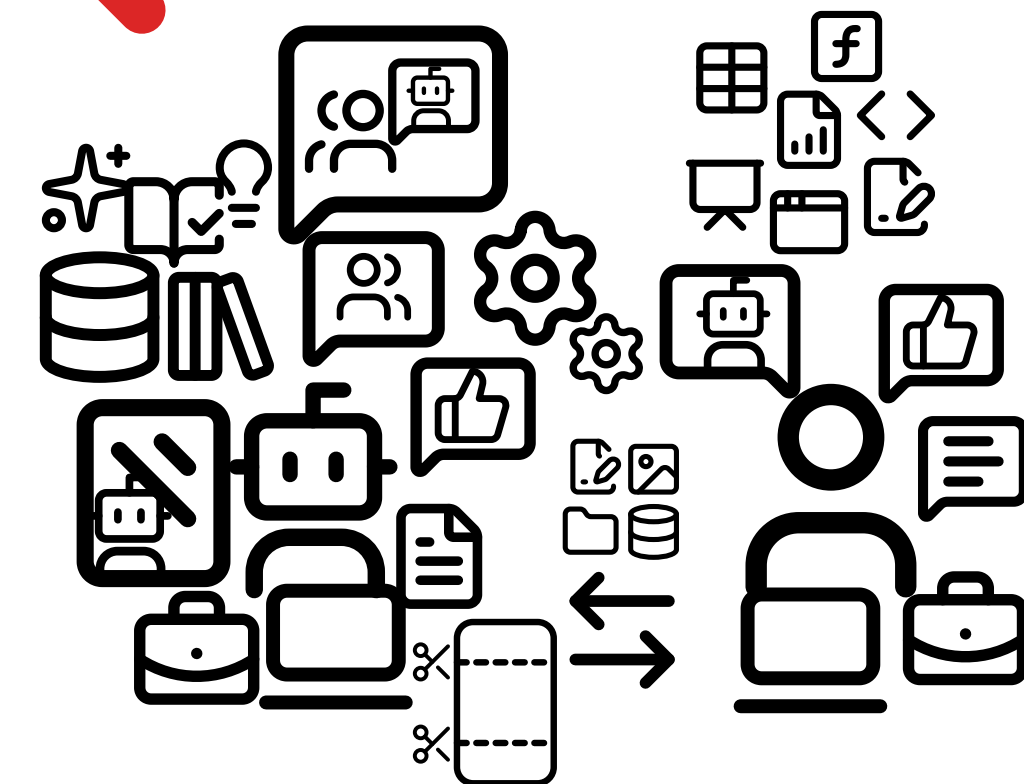
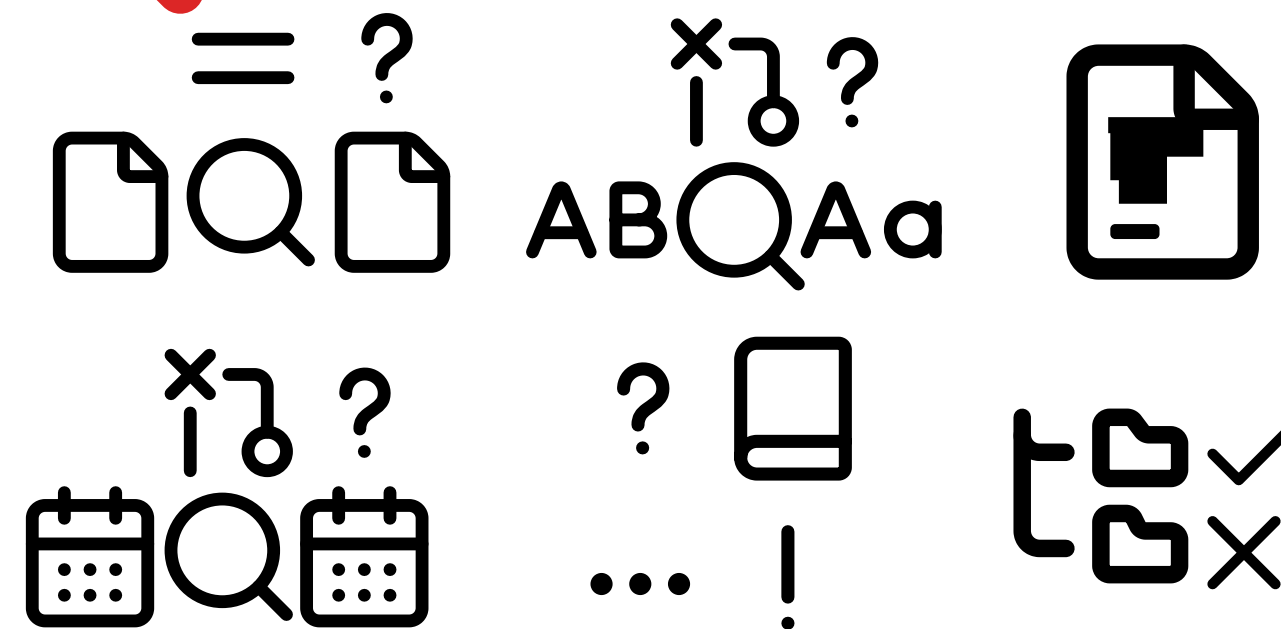
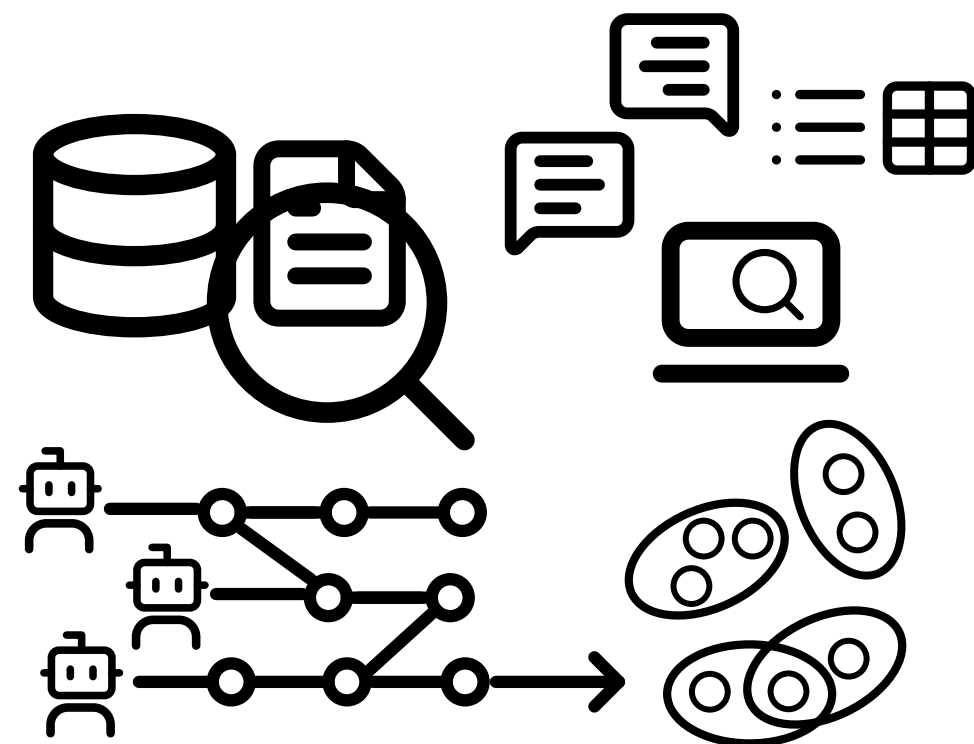
## Design goal of systems



Architecture

Operations

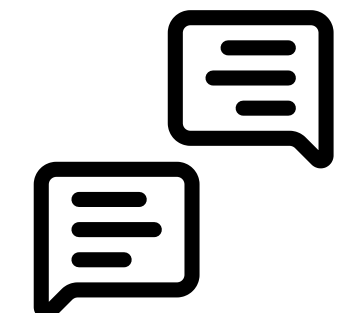
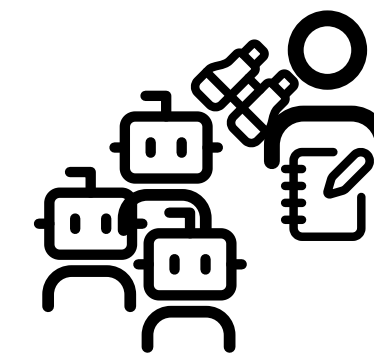
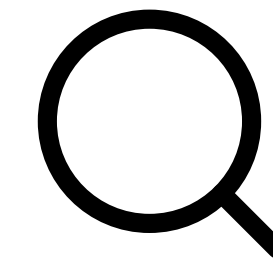
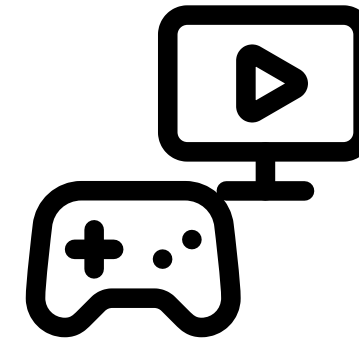
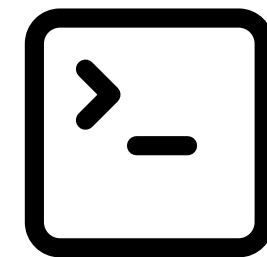
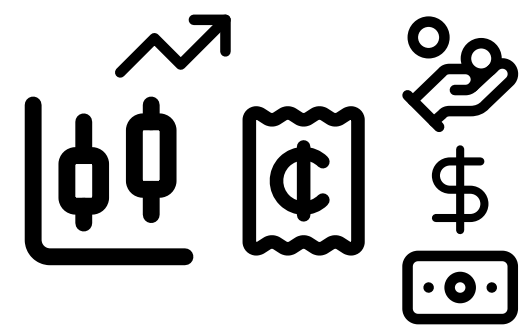
User interaction



# AI Memory Design Choices Matter

Misaligned designs may appear to work, but quietly affect user interaction.

## Design goal of systems



Architecture

Operations

User interaction

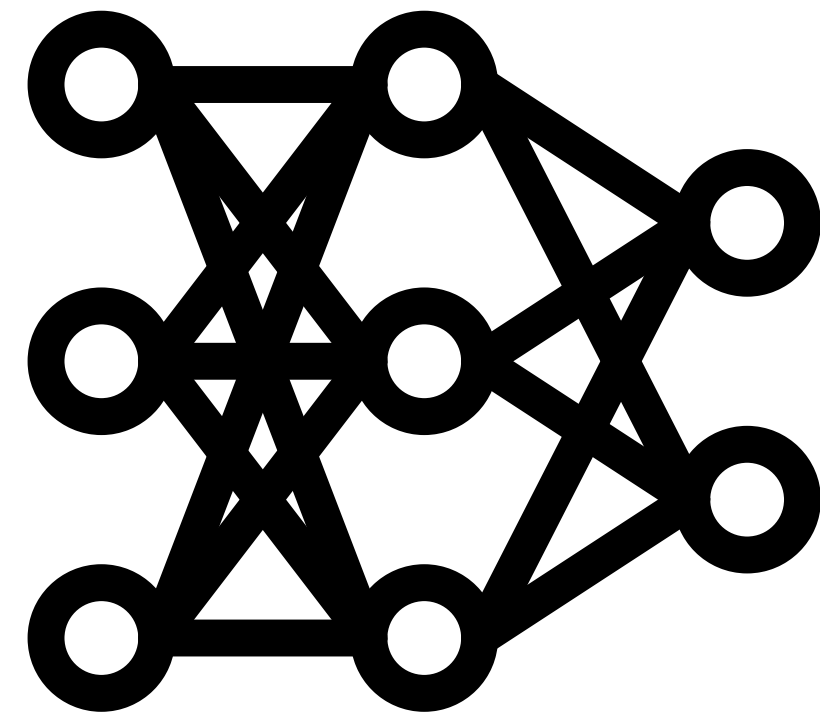


**E.g., users give up due to time and effort costs.**

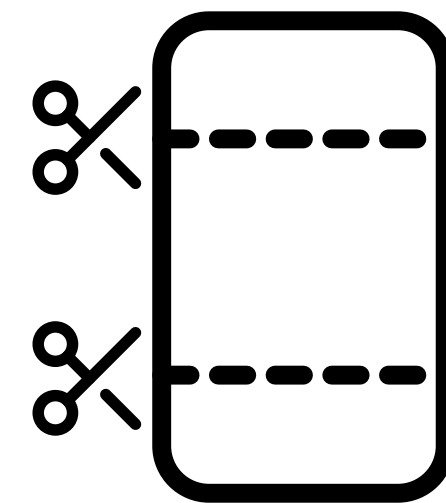
**What should we design in AI memory, and how?**

# Implicit and Explicit memory

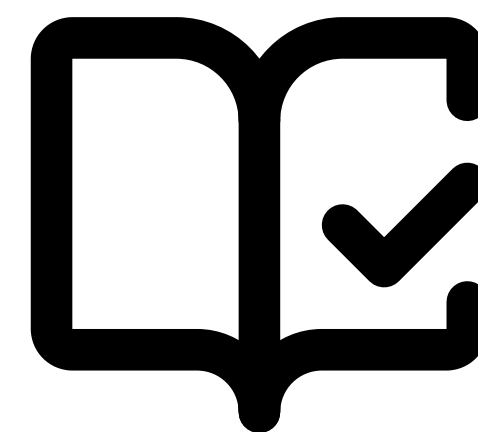
In LLM-based systems, memory can take implicit or explicit forms.



**Parametric (Implicit)  
Memory**



You are a helpful...  
# Instructions  
\* Only output...  
\* Your response...



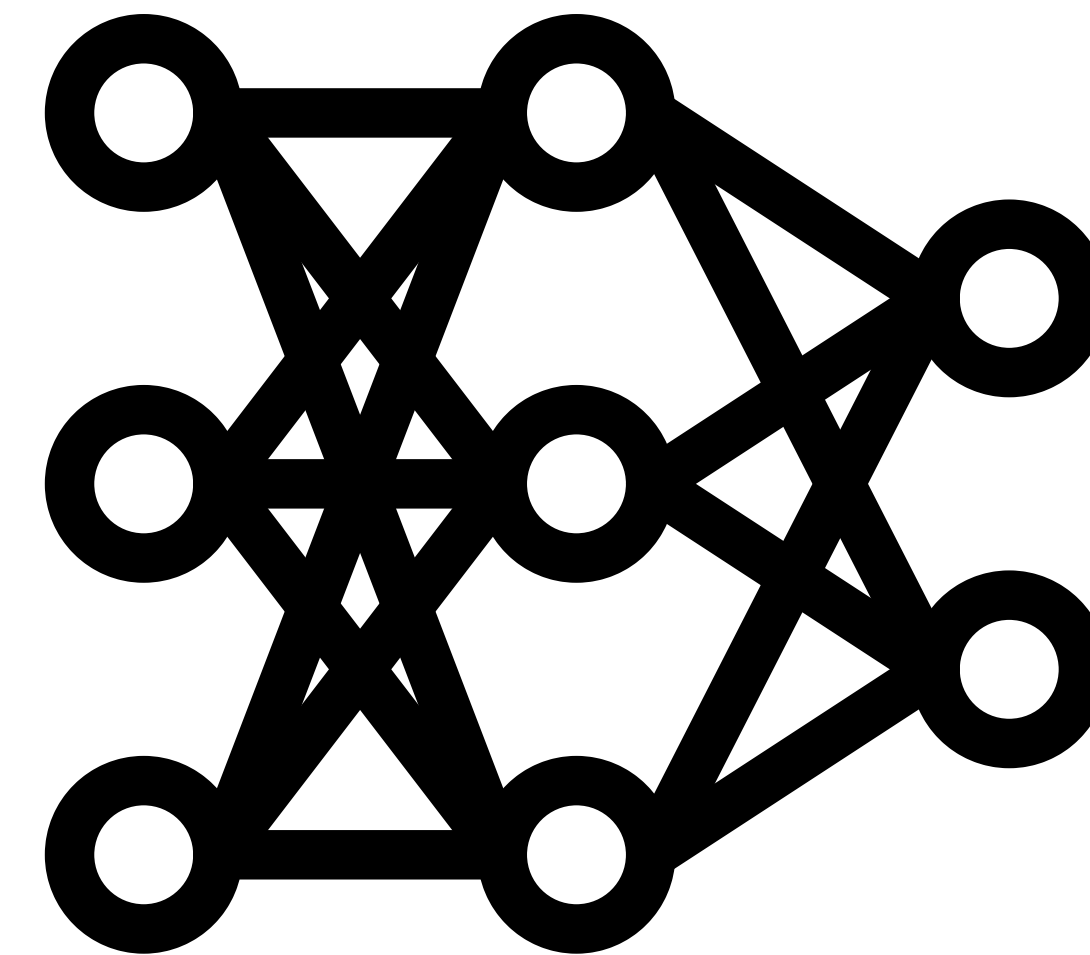
**Written (Explicit)  
Memory**

# Implicit and Explicit memory

In LLM-based systems, memory can take implicit or explicit forms.

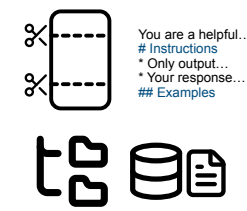
## **Implicit memory**

Knowledge implicitly encoded inside model parameters.



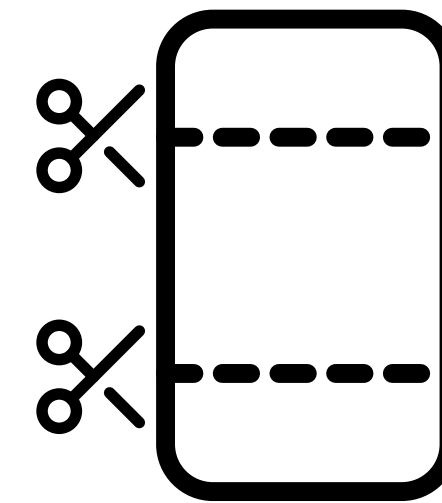
# Implicit and Explicit memory

In LLM-based systems, memory can take implicit or explicit forms.

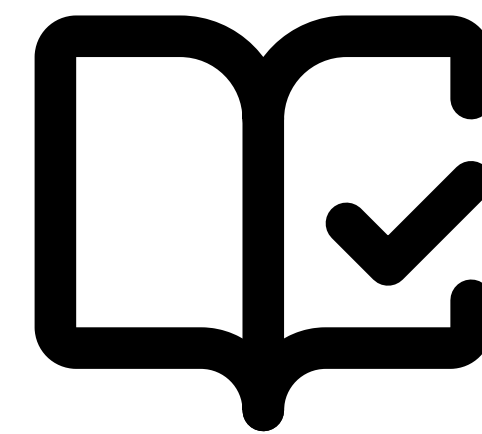


## Explicit memory

Information written outside the LLM, such as prompts, context windows, project files, and logs.

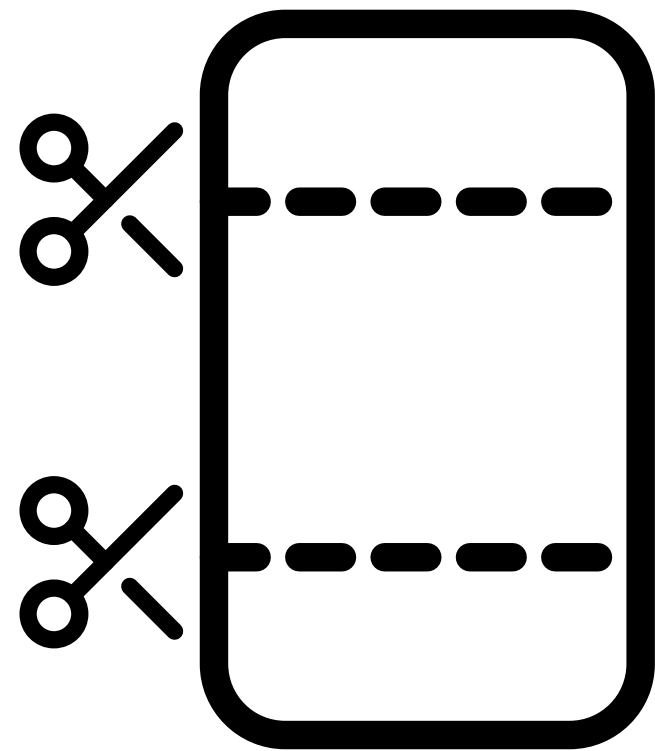


You are a helpful...  
# Instructions  
\* Only output...  
\* Your response...



# Explicit memory deserves more attention

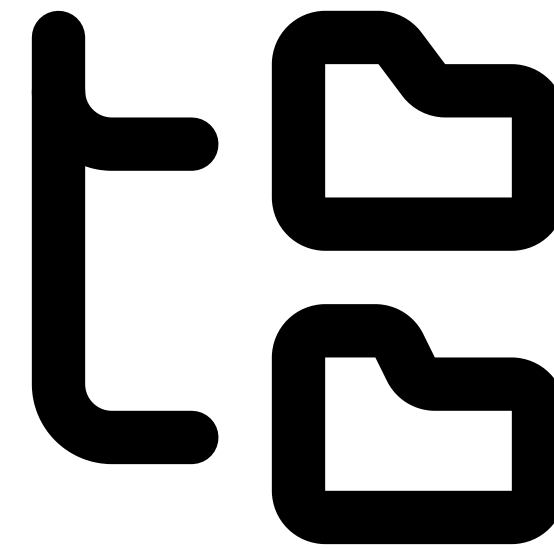
from a human-AI interaction perspective



**Context window**

You are a helpful...  
# Instructions  
\* Only output...  
\* Your response...  
## Examples

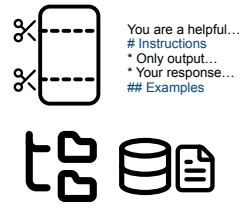
**System prompt**



**Workspace**  
(working repository)

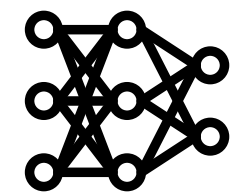


**External memory**



# Explicit memory

Easier to inspect and update during interaction



# Implicit memory

Hard to inspect and update during interaction

AGENTS.md > abc # Agent Notes

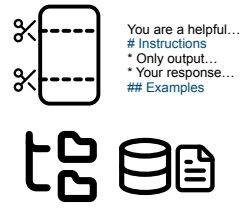
- 1 # Agent Notes
- 2
- 3 - For iOS UI changes, build the app and verify the affected flow in Simulator when feasible.
- 4 - When verifying a UI behavior in Simulator, capture and keep a screenshot that shows the behavior working, then mention the screenshot path in the final response.
- 5 - Prefer screenshots from `xcrun simctl io <device> screenshot` for iOS Simulator verification.
- 6

```
* Validation loss before training: 0.7032, accuracy: 51.2615%, auc: 0.5315

===== Training bert model on device: cuda =====
* Training epoch 1:
Avg. batch proc. time: 0.0374s, loss: 0.3680: 100%|██████████| 217/217 [01:01<00:00, 3.51it/s]
-> Training time: 61.8425s, loss = 0.3680, accuracy: 83.8873%
* Validation for epoch 1:
-> Valid. time: 2.3848s, loss: 0.2465, accuracy: 89.1055%, auc: 0.9654
save model succesfully!

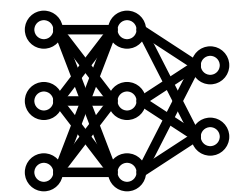
* Training epoch 2:
Avg. batch proc. time: 0.0604s, loss: 0.1769: 100%|██████████| 217/217 [01:02<00:00, 3.46it/s]
-> Training time: 62.6737s, loss = 0.1769, accuracy: 93.6127%
* Validation for epoch 2:
-> Valid. time: 2.3535s, loss: 0.2607, accuracy: 90.8257%, auc: 0.9678
save model succesfully!

* Training epoch 3:
Avg. batch proc. time: 0.0585s, loss: 0.0847: 100%|██████████| 217/217 [01:02<00:00, 3.45it/s]
-> Training time: 62.8606s, loss = 0.0847, accuracy: 97.0954%
* Validation for epoch 3:
-> Valid. time: 2.3710s, loss: 0.3368, accuracy: 90.9404%, auc: 0.9631
save model succesfully!
```



# Explicit memory

Explicitly specified agent behavior  
Greater user ownership and governance



# Implicit memory

Black-boxed agent behavior  
Reduced user ownership and governance

AGENTS.md > abc # Agent Notes

- 1 # Agent Notes
- 2
- 3 - For iOS UI changes, build the app and verify the affected flow in Simulator when feasible.
- 4 - When verifying a UI behavior in Simulator, capture and keep a screenshot that shows the behavior working, then mention the screenshot path in the final response.
- 5 - Prefer screenshots from ``xcrun simctl io <device> screenshot`` for iOS Simulator verification.
- 6

```
* Validation loss before training: 0.7032, accuracy: 51.2615%, auc: 0.5315

===== Training bert model on device: cuda =====
* Training epoch 1:
Avg. batch proc. time: 0.0374s, loss: 0.3680: 100%|██████████| 217/217 [01:01<00:00, 3.51it/s]
-> Training time: 61.8425s, loss = 0.3680, accuracy: 83.8873%
* Validation for epoch 1:
-> Valid. time: 2.3848s, loss: 0.2465, accuracy: 89.1055%, auc: 0.9654
save model successfully!

* Training epoch 2:
Avg. batch proc. time: 0.0604s, loss: 0.1769: 100%|██████████| 217/217 [01:02<00:00, 3.46it/s]
-> Training time: 62.6737s, loss = 0.1769, accuracy: 93.6127%
* Validation for epoch 2:
-> Valid. time: 2.3535s, loss: 0.2607, accuracy: 90.8257%, auc: 0.9678
save model successfully!

* Training epoch 3:
Avg. batch proc. time: 0.0585s, loss: 0.0847: 100%|██████████| 217/217 [01:02<00:00, 3.45it/s]
-> Training time: 62.8606s, loss = 0.0847, accuracy: 97.0954%
* Validation for epoch 3:
-> Valid. time: 2.3710s, loss: 0.3368, accuracy: 90.9404%, auc: 0.9631
save model successfully!
```

**How should we design AI memory systems?**

# How should we design AI memory systems?

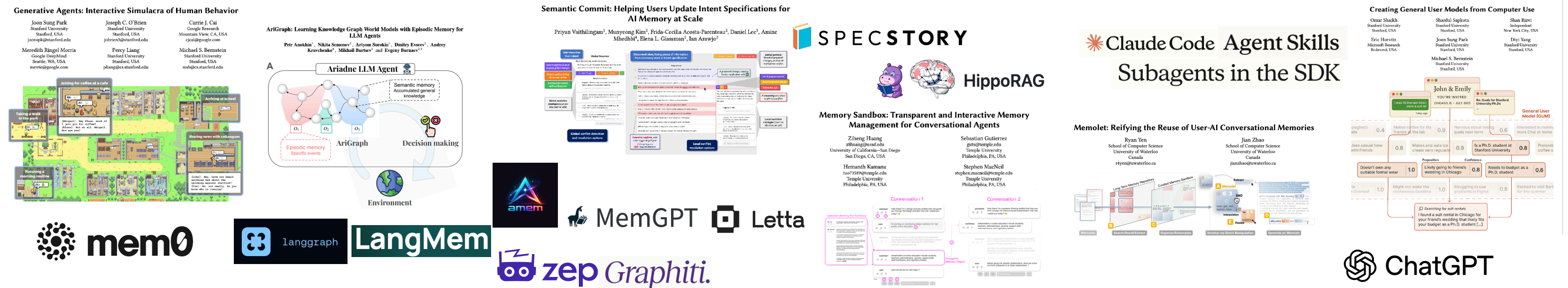
Unfortunately, we still lack a clear answer to this question.

- What design choices are available for AI memory systems?
- What trade-offs do these choices involve?
- What directions remain unexplored?

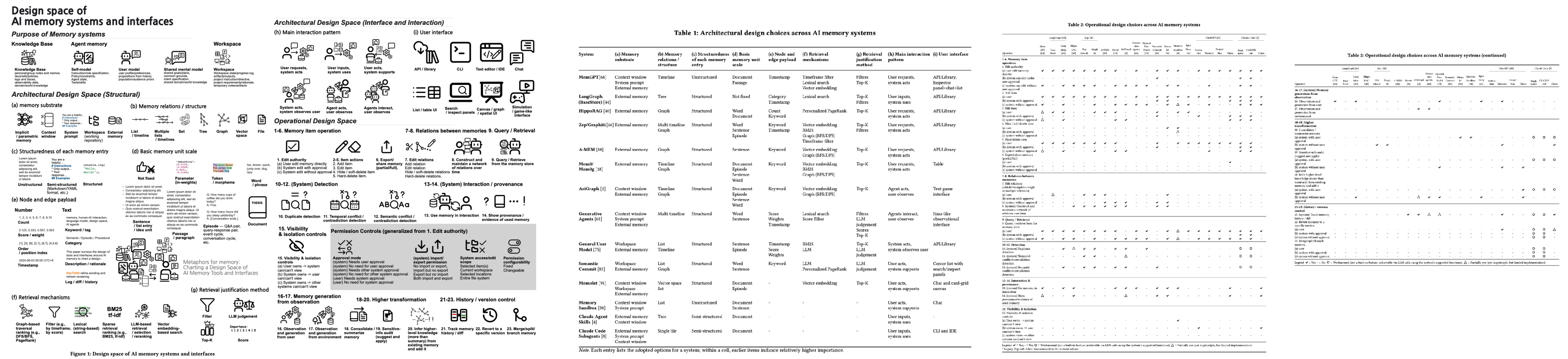


# First. Survey of AI memory systems to map their design space and new design opportunities.

## 1. Surveyed current AI memory systems



## 2. Mapped their architectural and operational design space



## 3. Derived implications and identified gaps

### Implications

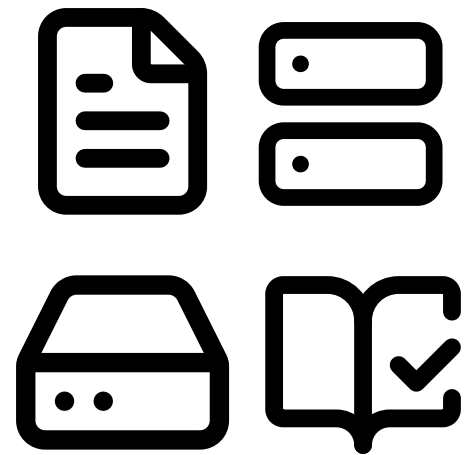
1. Implicit versus explicit memory
2. How should we scope and target memory substrates?
3. Granularity and density of memory components
4. Representing relations among memories
5. Choice of retrieval mechanisms
6. Interdependence between architecture and operations

### Gaps

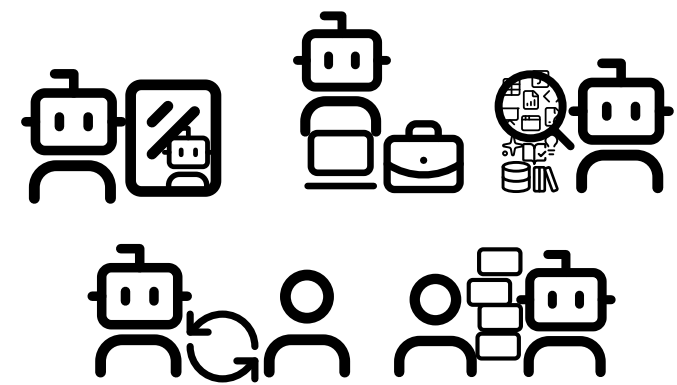
1. Underappreciated coupling between architecture/structure and operation
2. Gap in recording relationships among memories
3. Lack of user-agent common grounding steps
4. Accumulate-only lifecycle with no versioning
5. Lack of selective access control
6. Lack of approval control over memory operations
7. Gray zone in NLP and HCI

# Second. Dominant and less-used metaphors to inspire AI memory designs.

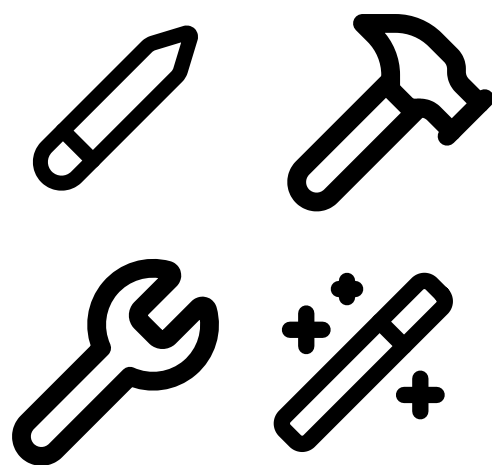
## 1. Dominant metaphors



Records, repositories,  
and database systems

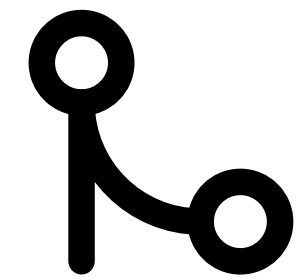


Neurobiology of  
human cognition

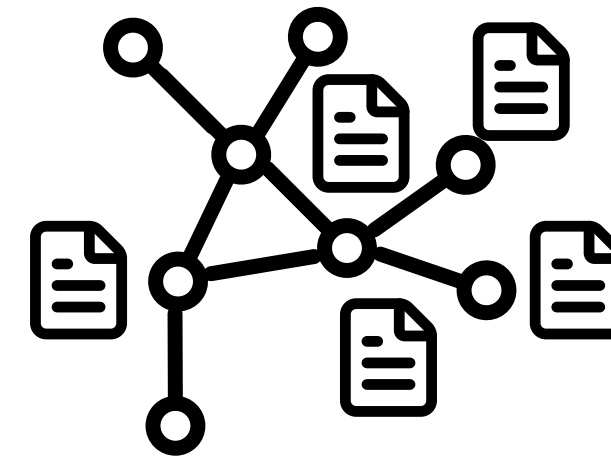


Skills and tools

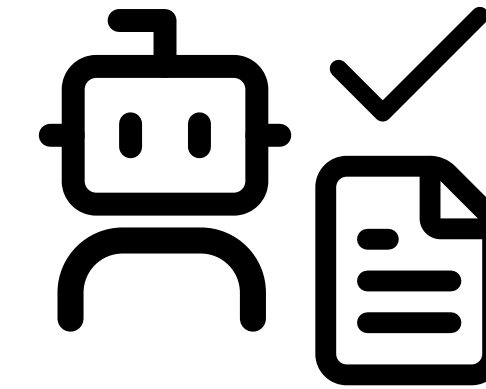
## 2. Less-used metaphors



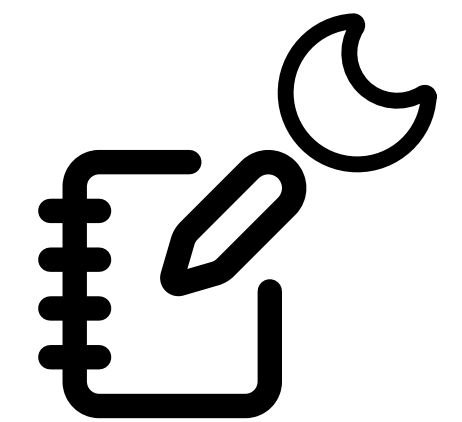
Software  
Version  
Control



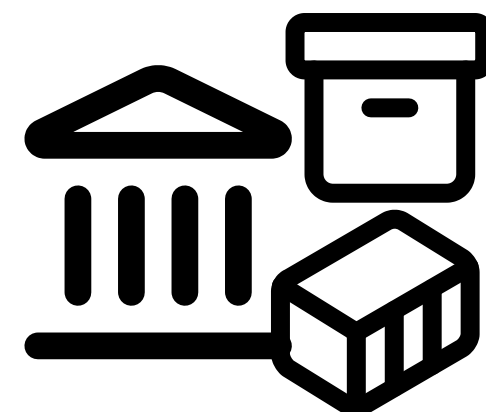
Zettelkasten



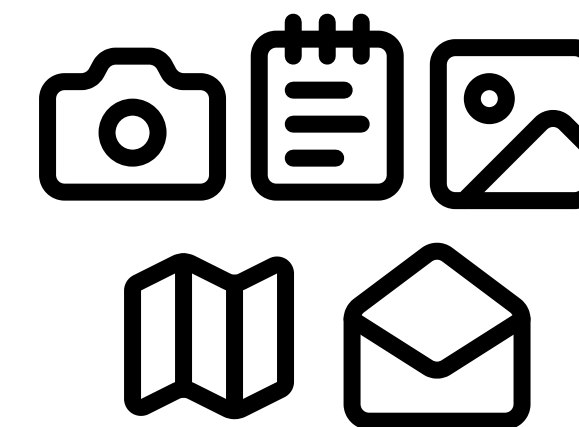
Requirements



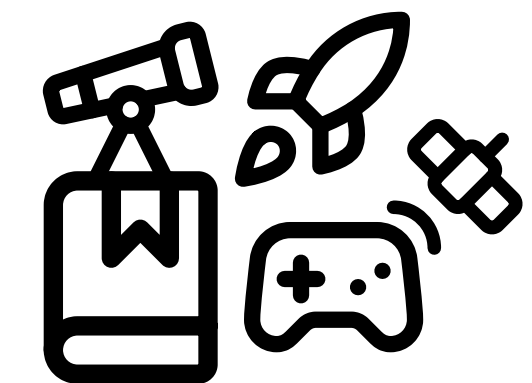
Personal  
Diary and  
Record



Community Archive



Cultural Probes



Science fiction  
tales/games

# **1. Charting a design space of AI memory**

# Survey and review of AI memory systems

Surveyed architectures (14 systems) and operations (15 systems / 22 subsystems).

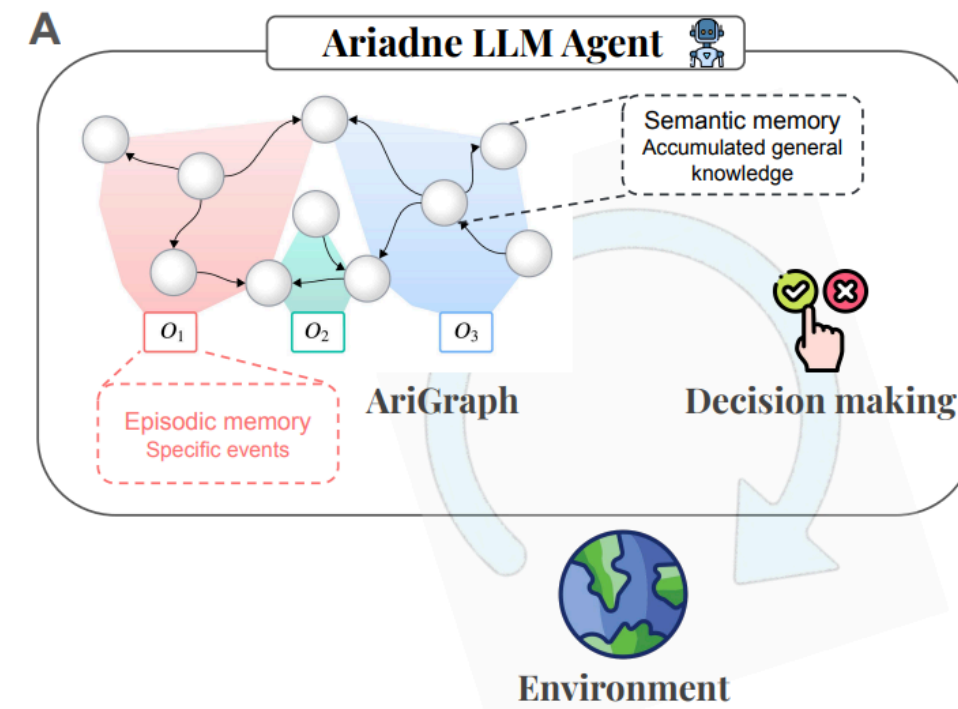
Snowball sampling across industrial and academic systems.

Prioritized depth over coverage.



**Ariadne LLM Agent**: Learning Knowledge Graph World Models with Episodic Memory for LLM Agents

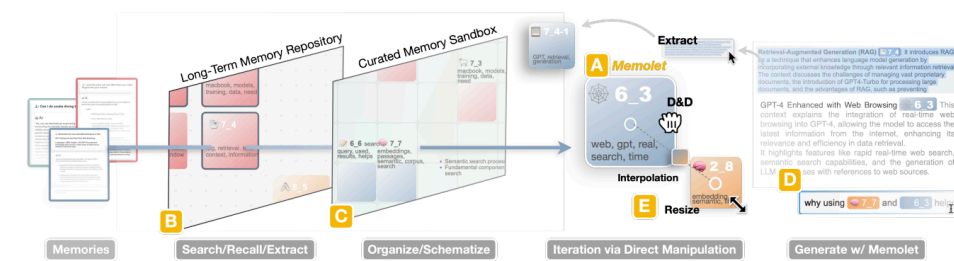
Petr Anokhin<sup>1</sup>, Nikita Semenov<sup>2</sup>, Artyom Sorokin<sup>1</sup>, Dmitry Evseev<sup>2</sup>, Andrey Kravchenko<sup>1</sup>, Mikhail Burtsev<sup>3</sup> and Evgeny Burnaev<sup>2,1</sup>



**Memolet: Reifying the Reuse of User-AI Conversational Memories**

Ryan Yen  
School of Computer Science  
University of Waterloo  
Canada  
r4yen@uwaterloo.ca

Jian Zhao  
School of Computer Science  
University of Waterloo  
Canada  
jianzhao@uwaterloo.ca



**Generative Agents: Interactive Simulacra of Human Behavior**

Joon Sung Park  
Stanford University  
Stanford, USA  
joonspk@stanford.edu

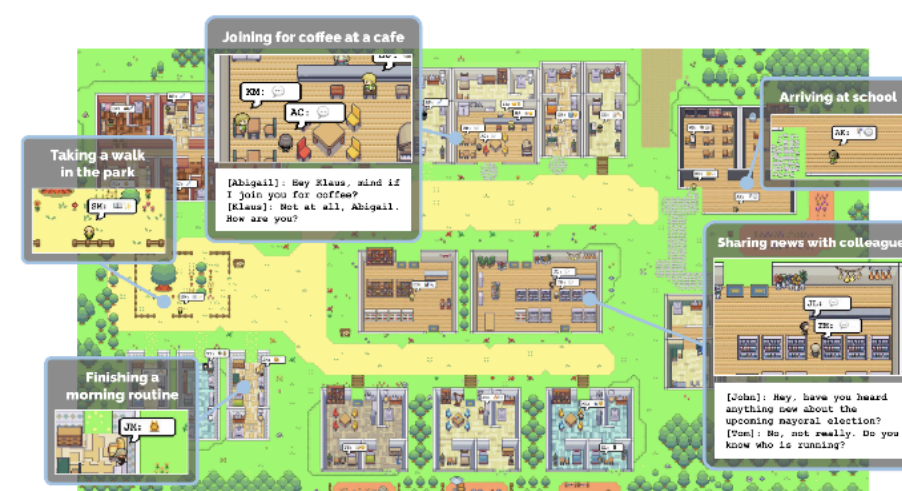
Joseph C. O'Brien  
Stanford University  
Stanford, USA  
jobrien3@stanford.edu

Carrie J. Cai  
Google Research  
Mountain View, CA, USA  
cjcail@google.com

Meredith Ringel Morris  
Google DeepMind  
Seattle, WA, USA  
merrie@google.com

Percy Liang  
Stanford University  
Stanford, USA  
pliang@cs.stanford.edu

Michael S. Bernstein  
Stanford University  
Stanford, USA  
msb@cs.stanford.edu



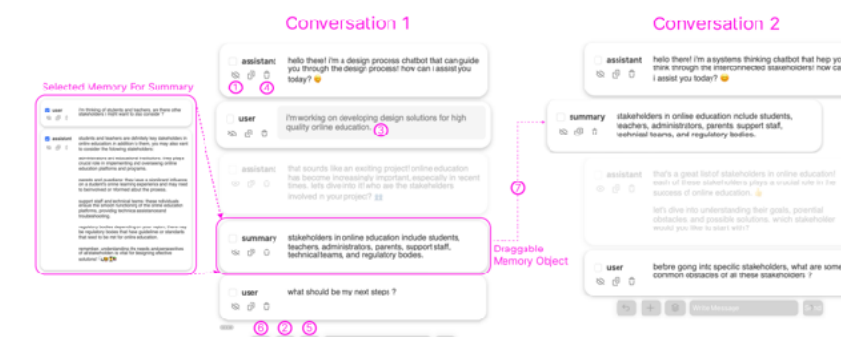
**Memory Sandbox: Transparent and Interactive Memory Management for Conversational Agents**

Ziheng Huang  
zshuang@ucsd.edu  
University of California—San Diego  
San Diego, CA, USA

Hemanth Kamana  
tu073589@temple.edu  
Temple University  
Philadelphia, PA, USA

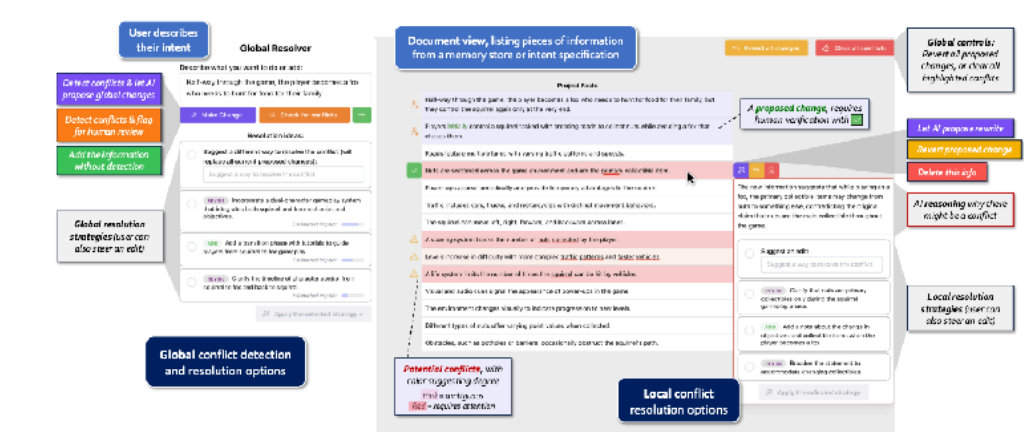
Sebastian Gutierrez  
guts@temple.edu  
Temple University  
Philadelphia, PA, USA

Stephen MacNeil  
stephen.macneil@temple.edu  
Temple University  
Philadelphia, PA, USA



**Semantic Commit: Helping Users Update Intent Specifications for AI Memory at Scale**

Priyan Vaithilingam<sup>1</sup>, Muyeong Kim<sup>2</sup>, Frida-Cecilia Acosta-Parenteau<sup>2</sup>, Daniel Lee<sup>3</sup>, Amine Mhedhbi<sup>4</sup>, Elena L. Glassman<sup>1</sup>, Ian Arawjo<sup>2</sup>



**Creating General User Models from Computer Use**

Omar Shaikh  
Stanford University  
Stanford, USA

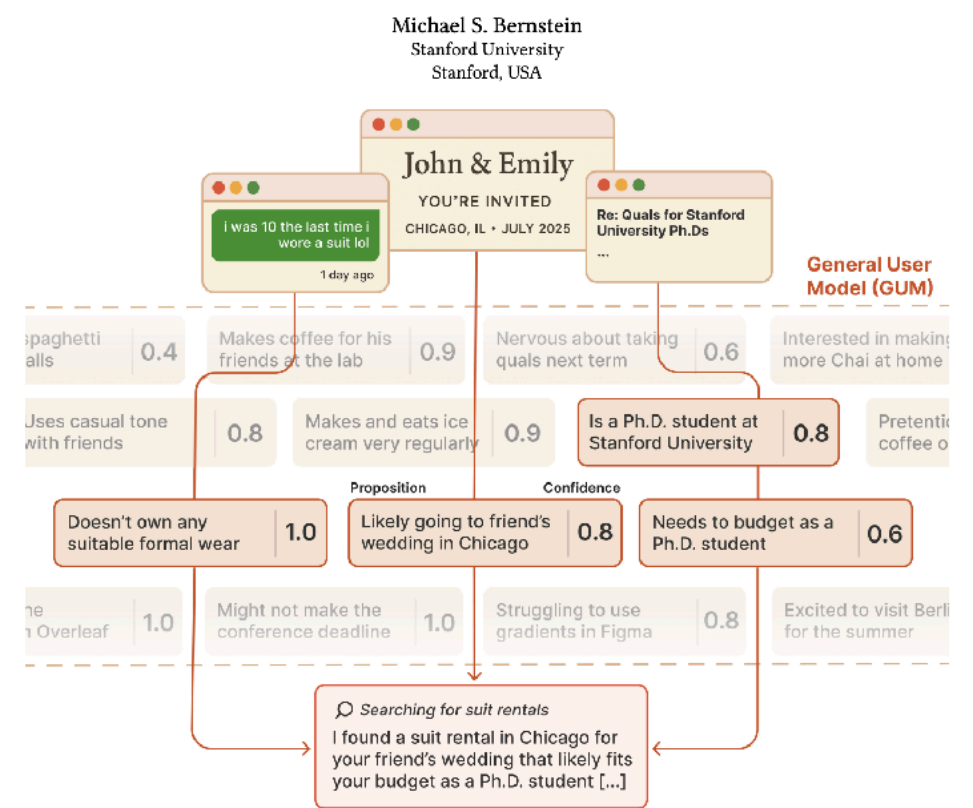
Shardul Sapkota  
Stanford University  
Stanford, USA

Shan Rizvi  
Independent  
New York City, USA

Eric Horvitz  
Microsoft Research  
Redmond, USA

Joon Sung Park  
Stanford University  
Stanford, USA

Diyi Yang  
Stanford University  
Stanford, USA



# Architectural and operational design choices

Started with seed axes and iteratively refined the coding scheme. When an axis changed, we revisited the affected codes, and stopped once new cases fit the existing axes and options.

Table 1: Architectural design choices across AI memory systems

System	(a) Memory substrate	(b) Memory relations / structure	(c) Structuredness of each memory entry	(d) Basic memory unit scale	(e) Node and edge payload	(f) Retrieval mechanisms	(g) Retrieval justification method	(h) Main interaction pattern	(i) User interface
MemGPT [64]	Context window System prompt External memory	Timeline	Unstructured	Document Passage	Timestamp	Timeframe filter Lexical search Vector embedding	Filters Top-K	User requests, system acts	APILibrary, Inspector panel+chat+list
LangGraph (BaseStore) [44]	External memory	Tree	Structured	Not fixed	Category Timestamp	Lexical search	Top-K Filters	User inputs, system uses	APILibrary
HippoRAG [40]	External memory	Graph	Structured	Word Document	Count Keyword	Personalized PageRank	Top-K	User requests, system acts	APILibrary
Zep/Graphiti [66]	External memory	Multi timeline Graph	Structured	Word Sentence Episode	Keyword Timestamp	Vector embedding BM25 Graph (BFS/DFS) Timeframe filter	Top-K Filters	User requests, system acts	APILibrary
A-MEM [88]	External memory	Graph	Structured	Sentence	Keyword	Vector embedding Graph (BFS/DFS)	Top-K	User requests, system acts	APILibrary
Mem0/ Mem0g [18]	External memory	Timeline Graph	Structured	Document Episode Sentence Word	Keyword	Vector embedding BM25 Graph (BFS/DFS)	Top-K	User requests, system acts	Table
AriGraph [2]	Context window External memory	Timeline Graph	Structured	Document Episode Sentence Word	Keyword	Vector embedding Graph (BFS/DFS)	Top-K	Agent acts, user observes	Text game interface
Generative Agents [65]	Context window System prompt External memory	Multi timeline	Structured	Word Sentence	Score Weights Timestamp	Lexical search Score Filter	Filters LLM judgement Top-K	Agents interact, user observes	Sims-like observational interface
General User Model [75]	Workspace External memory	List Timeline	Structured	Sentence Episode	Timestamp Score Weights	BM25 LLM	Top-K LLM judgement	System acts, system observes user	APILibrary
Semantic Commit [83]	Workspace External memory	List Graph	Structured	Word Sentence	Keyword	LLM Personalized PageRank	LLM judgement	User acts, system supports	Center list with search/inspect panels
Memolet [91]	Context window Workspace External memory	Vector space Set	Structured	Document Episode	-	Vector embedding	Top-K	User acts, system supports	Chat; and card-grid canvas
Memory Sandbox [38]	Context window System prompt	List	Unstructured	Document Sentence	-	-	-	User acts, system supports	Chat
Claude Agent Skills [4]	External memory Context window	Tree	Semi-structured	Document	-	-	-	User inputs, system uses	-
Claude Code Subagents [8]	External memory System prompt Context window	Single file	Semi-structured	Document	-	-	-	User inputs, system uses	CLI and IDE

Note. Each entry lists the adopted options for a system; within a cell, earlier items indicate relatively higher importance.

Table 2: Operational design choices across AI memory systems

	LangGraph [44]				Zep [66]				ChatGPT [65]				Claude Code [5]						
	Mem CPT [64]	Base Store [45]	Lang Mem [46]	Hippo SAU [40]	Zep (tsou0)	Graphiti [66]	A-MEM [88]	Mem0 [18]	AriGraph [2]	Generative Agents [65]	General User Model [75]	Semantic Commit [83]	Memory Sandbox [38]	Story [16]	Interv. [16]	Project files	Work space	CLAUDE .md	Claude Code [5]
Operatic	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1-6. Memory item operation																			
1. Edit authority																			
(a) user edit memory directly	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(b) system can edit under user approval	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(c) system can edit without user approval	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2. Add item																			
(a) user	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(b) system with approval	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(c) system without approval	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3. Edit item																			
(a) user	Δ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(b) system with approval	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(c) system without approval	Δ	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4. Hide / soft-delete item																			
(a) user	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(b) system with approval	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(c) system without approval	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5. Hard-delete item																			
(a) user	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(b) system with approval	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(c) system without approval	Δ	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6. Export/share memory (partial/full)																			
(a) user	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(b) system with approval	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(c) system without approval	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7-8. Relations between memories																			
7. Edit relations (add/delete/update; single or multiple relations)																			
(a) user	-	Δ	-	-	✓	✓	✓	-	-	-	-	✓	-	-	-	-	-	-	-
(b) system with approval	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(c) system without approval	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8. (system) Construct and maintain a network of relations over time	-	-	-	✓	✓	✓	✓	✓	-	-	✓	✓	-	-	-	-	-	-	-
9. Query / Retrieval																			
9. Query / retrieve from the memory store																			
(a) user	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(b) system with approval	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(c) system without approval	✓	-	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10-12. Detection																			
10. (system) Duplicate detection	-	-	✓	Δ	✓	✓	-	✓	✓	-	✓	-	-	-	-	-	-	○	○
11. (system) Temporal conflict/contradiction detection	-	-	-	-	✓	✓	-	-	Δ	-	-	-	-	-	-	-	-	-	-
12. (system) Semantic conflict/contradiction detection	-	-	✓	-	✓	✓	✓	✓	-	-	✓	✓	-	-	-	-	-	○	○
13-14. Interaction & provenance																			
13. (system) Use memory in interaction	✓	-	✓	✓	-	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
14. (system) Show provenance/evidence of used memory	Δ	-	-	✓	-	-	-	Δ	-	-	✓	✓	-	-	-	-	-	-	-
15. Visibility & Isolation																			
15. Visibility & isolation controls																			
(a) User owns → system can/can't view	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-	-
(b) system owns → user can/can't view	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-	-
(c) system owns → other systems can/can't view	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	-	-	-	-	-

Legend: ✓ = Yes; - = No; ○ = Workaround (not a built-in feature; achievable via LLM calls using the system's supported functions); Δ = Partially yes (yes in principle, but limited implementation)  
\*Legacy Zep soft-delete was removed in the current release.

Table 2: Operational design choices across AI memory systems (continued)

	LangGraph [44]				Zep [66]				ChatGPT [65]				Claude Code [5]						
	Mem CPT [64]	Base Store [45]	Lang Mem [46]	Hippo SAU [40]	Zep (tsou0)	Graphiti [66]	A-MEM [88]	Mem0 [18]	AriGraph [2]	Generative Agents [65]	General User Model [75]	Semantic Commit [83]	Memory Sandbox [38]	Story [16]	Interv. [16]	Project files	Work space	CLAUDE .md	Claude Code [5]
Operatic	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16-17. (system) Memory generation from observation																			
16. Observation and generation from user	✓	-	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17. Observation and generation from environment	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	-	-	-	-	-
18-20. Higher transformation																			
18. Consolidate / summarize memory	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	-	-	-	-
(a) system with user approval	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
(b) system without user approval	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
19. Sensitive-info audit (suggest and apply)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(a) system with user approval	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
(b) system without user approval	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
20. Infer higher-level knowledge (more than summary) from existing memory and add it	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
(a) system with user approval	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
(b) system without user approval	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
21-23. History / version control																			
21. (system) Track memory history / diff	-	-	-	-	-	-	-	-	-	✓	✓	Δ	Δ	-	-	-	-	-	○
22. Revert memory to a specific version	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
(a) user	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
(b) system with approval	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
(c) system without approval	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
23. Merge/split/branch memory	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
(a) user	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
(b) system with approval	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○
(c) system without approval	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○

Legend: ✓ = Yes; - = No; ○ = Workaround (not a built-in feature; achievable via LLM calls using the system's supported functions); Δ = Partially yes (yes in principle, but limited implementation)

**Table 1 (left): Architectural design choices across AI memory systems**

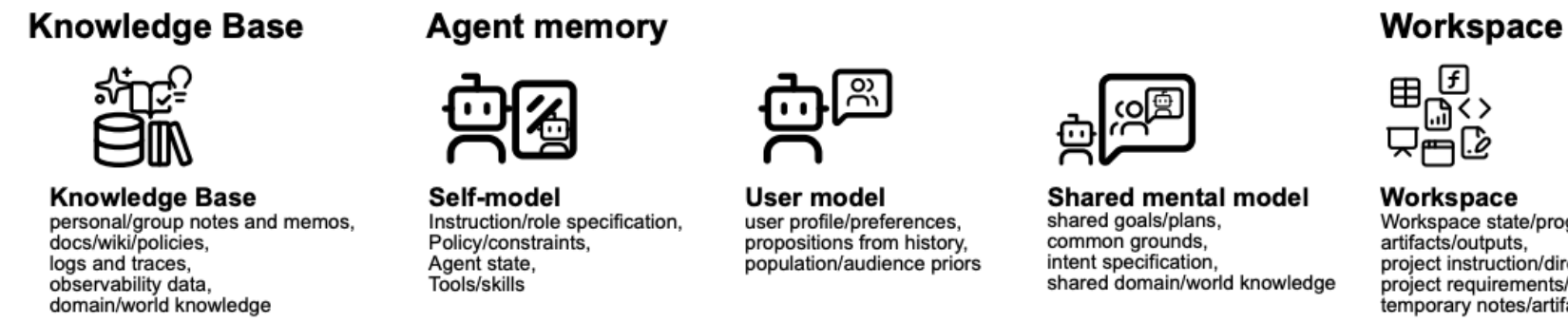
**Table 2 (right): Operational design choices across AI memory systems**

# Design Space of AI memory systems and interfaces

We then developed a design space based on these design surveys.

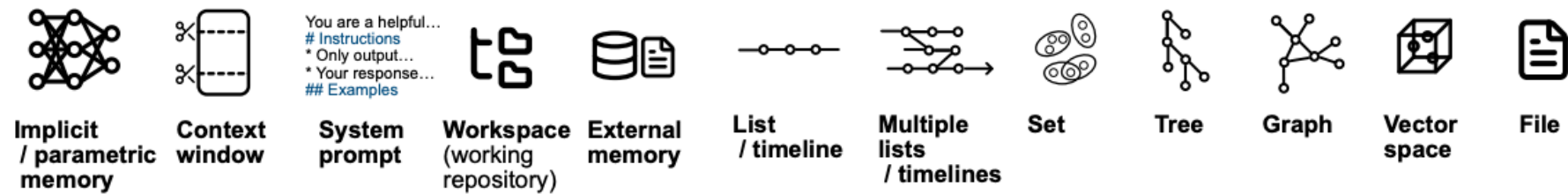
## Design space of AI memory systems and interfaces

### Purpose of Memory systems



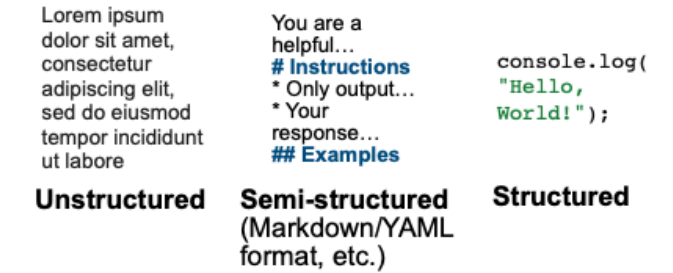
### Architectural Design Space (Structural)

#### (a) memory substrate

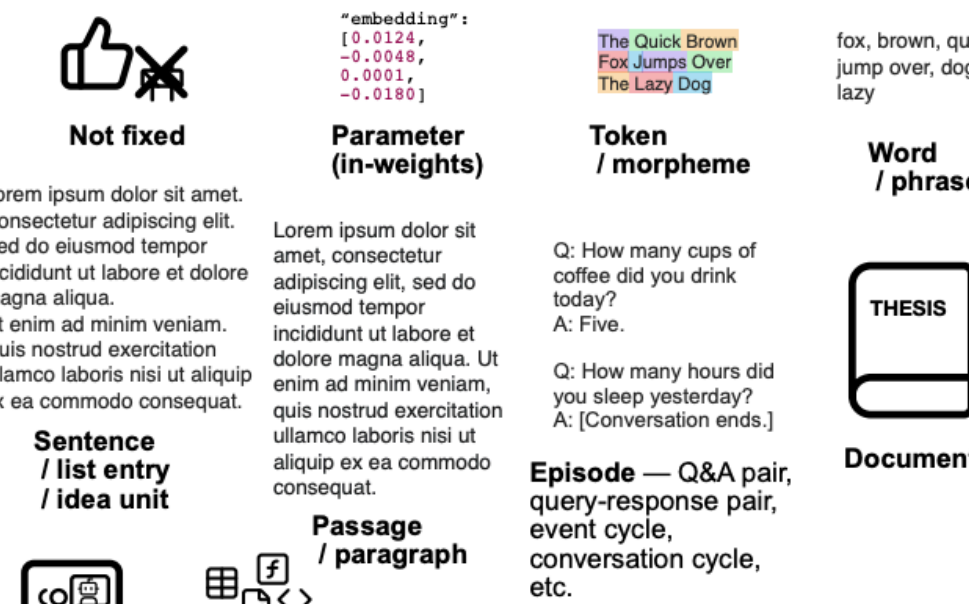


#### (b) Memory relations / structure

#### (c) Structuredness of each memory entry

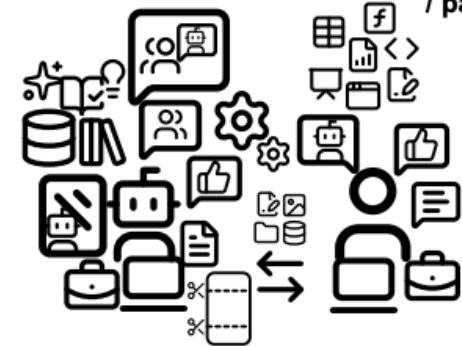


#### (d) Basic memory unit scale



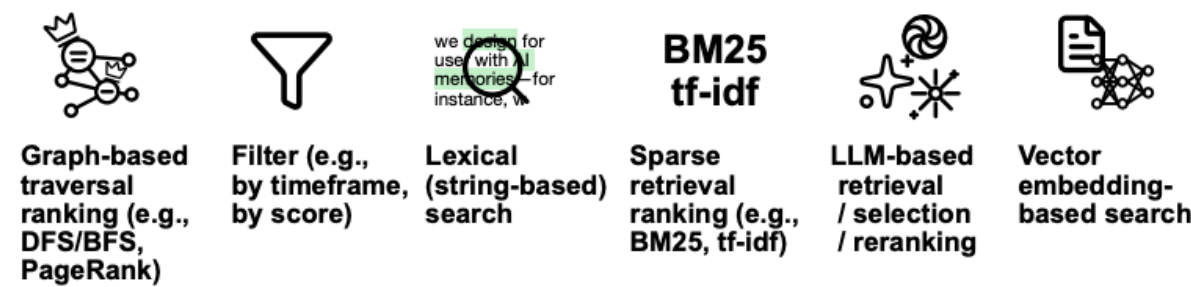
#### (e) Node and edge payload

Number	Text
Count 1, 2, 3, 4, 5, 6, 7, 8, 9, 10	memory, human-AI interaction, language model, design space, AI agents
Score / weight 0.123, 0.333, 0.557, 0.053	Keyword / tag
Order / position index [1], [3], [6], [2,1], [8,7], [4,5,6]	Category Semantic / Episodic / Procedural
Timestamp 2024-08-03 00:00 UTC+9	Description / rationale This paper surveys the design of tools and interfaces around AI memory to chart a design..
	Log / diff / history 93a72469 refine wording and refresh rendering

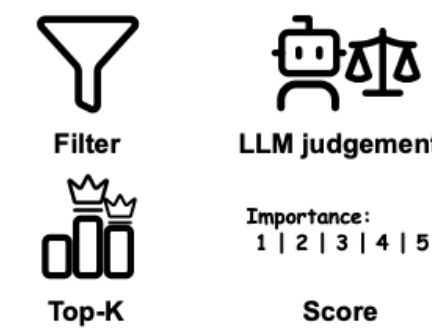


Metaphors for memory: Charting a Design Space of AI Memory Tools and Interfaces

#### (f) Retrieval mechanisms

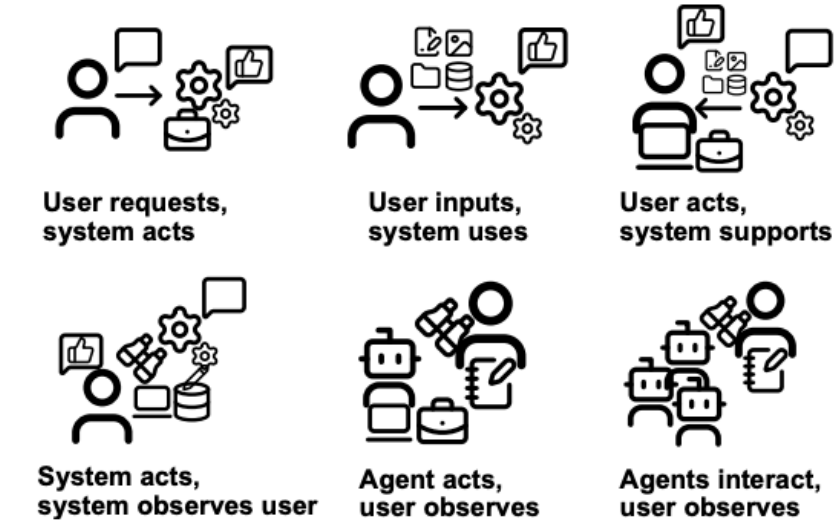


#### (g) Retrieval justification method

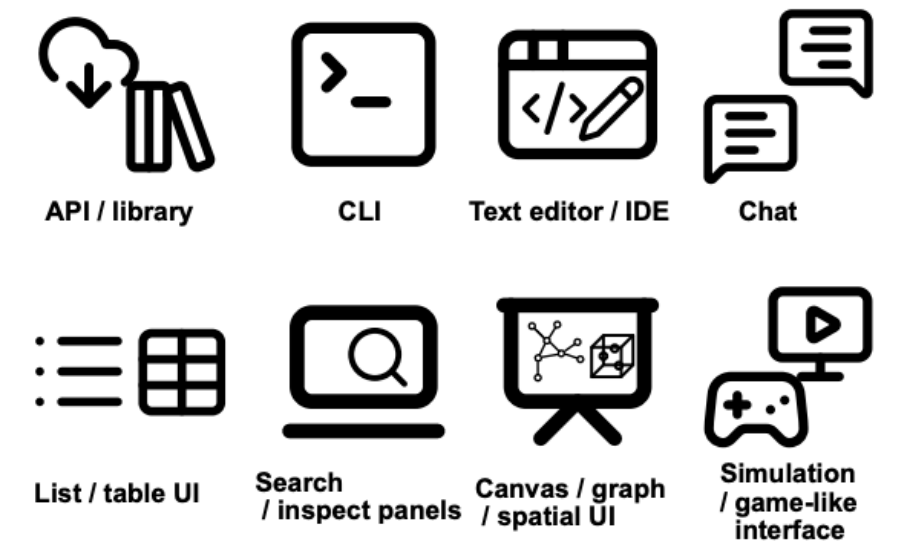


### Architectural Design Space (Interface and Interaction)

#### (h) Main interaction pattern

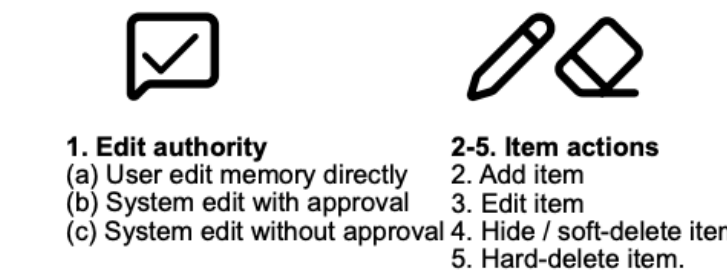


#### (i) User interface



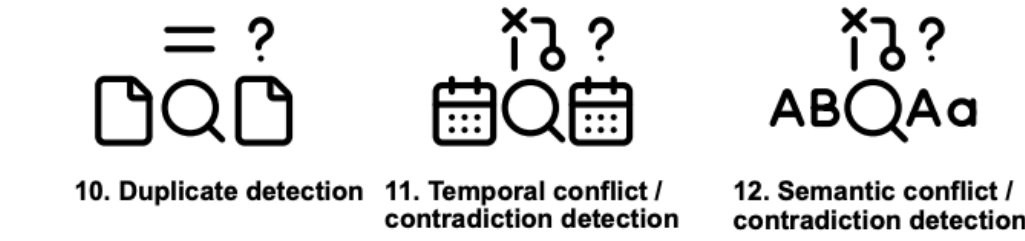
### Operational Design Space

#### 1-6. Memory item operation

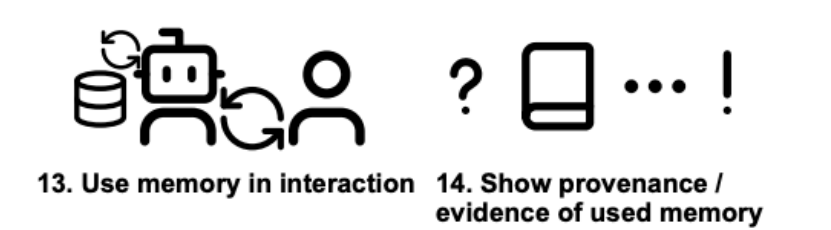


#### 7-8. Relations between memories 9. Query / Retrieval

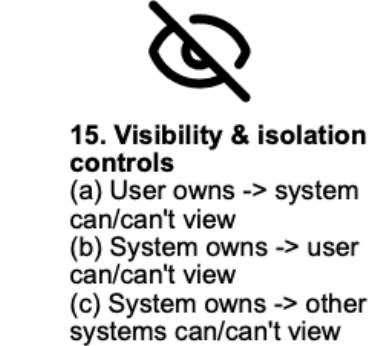
#### 10-12. (System) Detection



#### 13-14. (System) Interaction / provenance



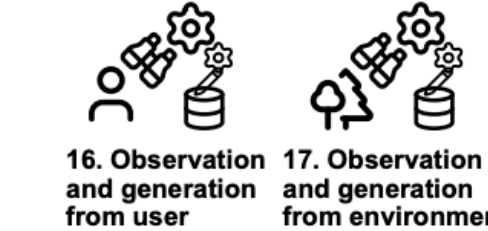
#### 15. Visibility & isolation controls



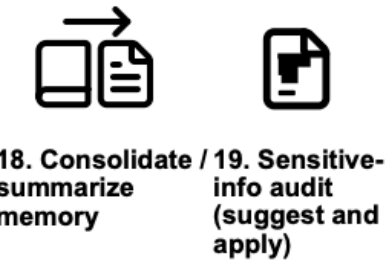
#### Permission Controls (generalized from 1. Edit authority)



#### 16-17. Memory generation from observation



#### 18-20. Higher transformation



#### 21-23. History / version control



Figure 1: Design space of AI memory systems and interfaces

Figure 1: Design space of AI memory systems and interfaces (continued)

# Survey Implications and Identified Gaps

## Implications

1. Implicit versus explicit memory
2. How should we scope and target memory substrates?
3. Granularity and density of memory components
4. Representing relations among memories
5. Choice of retrieval mechanisms
6. Interdependence between architecture and operations

## Gaps

1. Underappreciated coupling between architecture/structure and operation
2. Gap in recording relationships among memories
3. Lack of user-agent common grounding steps
4. Accumulate-only lifecycle with no versioning
5. Lack of selective access control
6. Lack of approval control over memory operations
7. Gray zone in NLP and HCI

**One key takeaway**

**Interdependence among  
architecture, operations, and resulting interactions**

Without this alignment, memory systems may fail to support their intended interactions.

**One key takeaway**

# **Interdependence between architecture and operations, and final interaction**

**Case 1. When operations do not support intended interactions:**

**Interactions**

Recover and edit memory

**Operations**

No versioning operations & hard edit only

**Result**

**Users permanently lose previous memories when editing**

One key takeaway

# Interdependence between architecture and operations, and final interaction

Case 2. When the architecture does not support intended operations:

**Interaction** Identify conflicting directions within the team

**Operations** Supports semantic conflict identification and resolution

**Architecture** **Single Markdown file with no information on who wrote what**

**Result** **System cannot tell who said what**

**One key takeaway**

# **Interdependence between architecture and operations, and final interaction**

When either operations or architecture fail, the system cannot work as intended.



# Gray Zone in NLP and HCI

We also found a field-level pattern, where NLP work often overlooks interaction, while HCI work often under-specifies architecture.

**NLP**

**Architecture**

**Operations**

**User interaction**



Limited user-agent support for interactive memory manipulation

**HCI**

**Architecture**

**Operations**

**User interaction**

Performance degradation from architectural mismatch



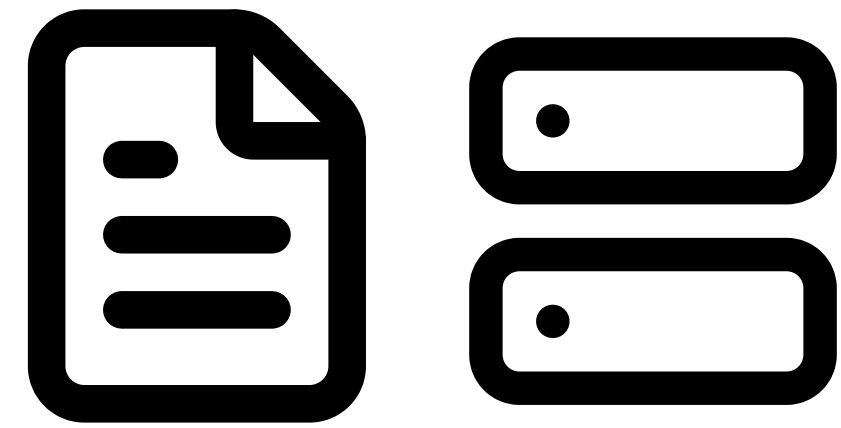
## **Long story short**

**When designing any memory system, we need to think about its architecture, operations, and the resulting interaction together.**

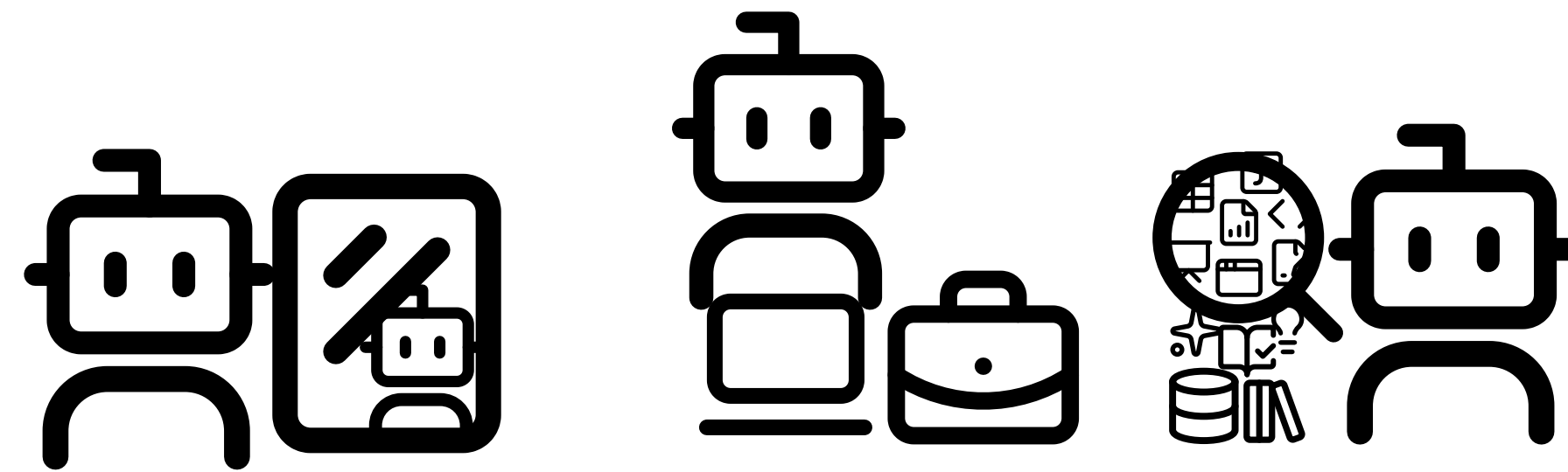
## **2. Metaphors for memory**

# Dominant Metaphors for memory

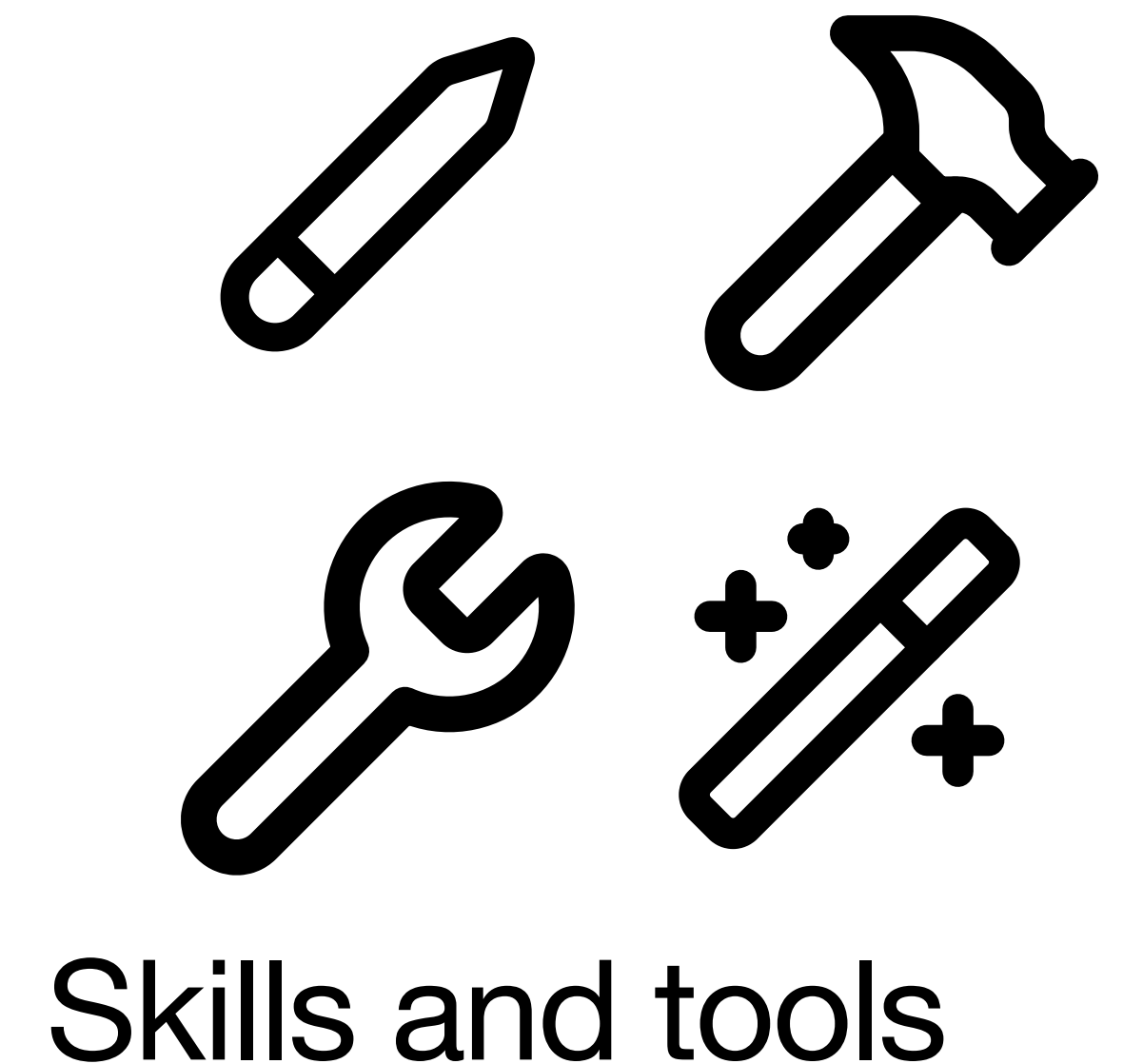
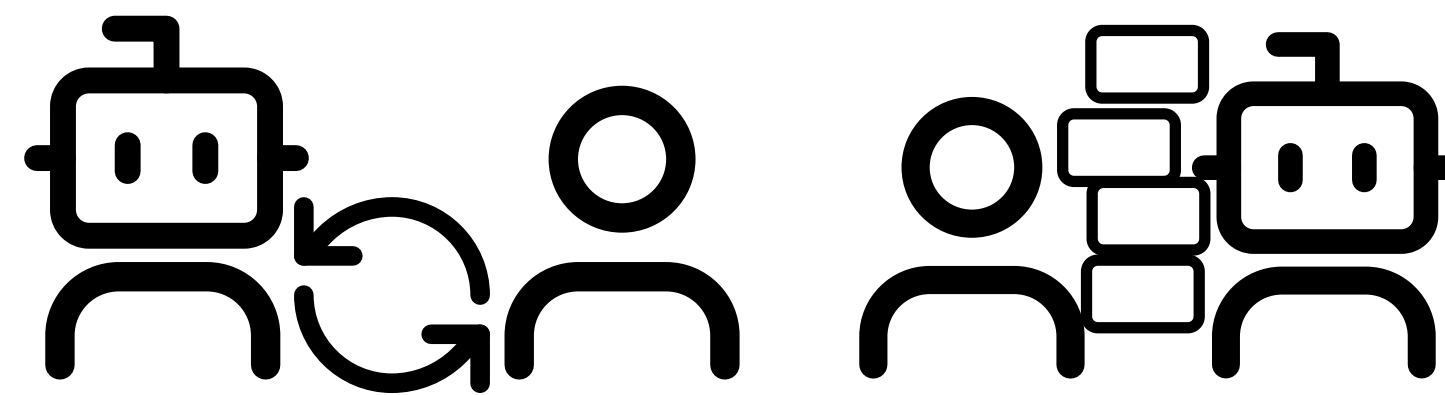
We also found that the systems in our survey repeatedly drew on these three metaphors.



Records,  
repositories, and  
database systems



Neurobiology of  
human cognition



Skills and tools

# **1. Memory as Records, repositories, and database systems**

# Memory as Records, repositories, and database systems

MemGPT explicitly draws inspiration from memory systems in traditional operating systems.



## Abstract

Large language models (LLMs) have revolutionized AI, but are constrained by limited context windows, hindering their utility in tasks like extended conversations and document analysis. To enable using context beyond limited context windows, we propose *virtual context management*, a technique **drawing inspiration from hierarchical memory systems in traditional operating systems** which provide the illusion of an extended virtual memory via paging between physical memory and disk. Using this technique, we introduce MemGPT (MemoryGPT), a system that intelli-

# Memory as Records, repositories, and database systems

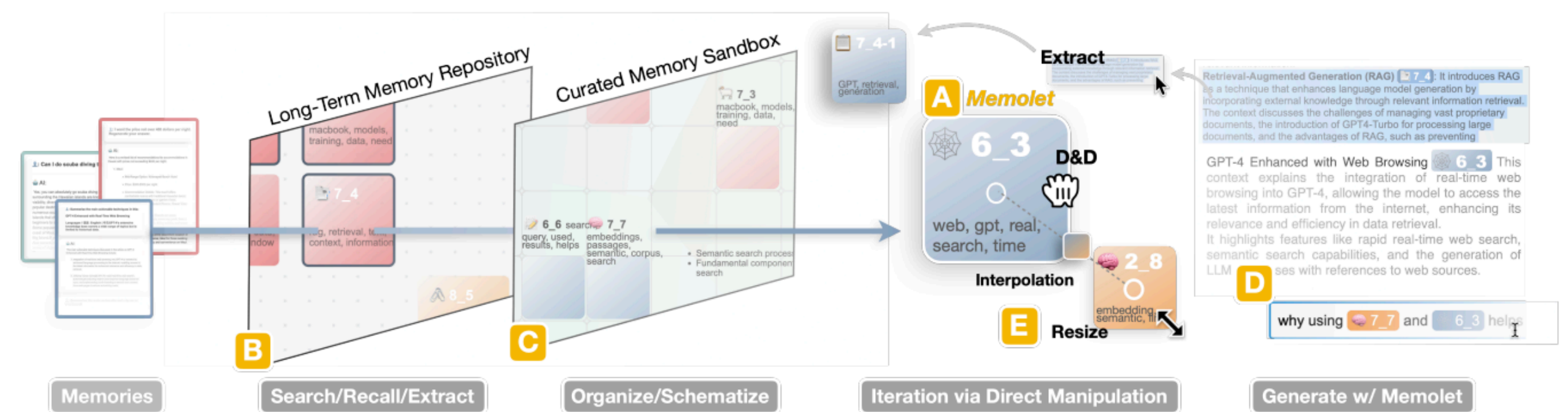
Mem0 and Memolet also draw on the notion of a “memo” in how they frame their systems.



## Memolet: Reifying the Reuse of User-AI Conversational Memories

Ryan Yen  
School of Computer Science  
University of Waterloo  
Canada  
r4yen@uwaterloo.ca

Jian Zhao  
School of Computer Science  
University of Waterloo  
Canada  
jianzhao@uwaterloo.ca



## **2. Memory as Neurobiology of human cognition**

# Memory as Neurobiology of human cognition

This metaphor has a long history, tracing back to Alan Turing's question, "Can machines think?"

## I.—COMPUTING MACHINERY AND INTELLIGENCE

BY A. M. TURING

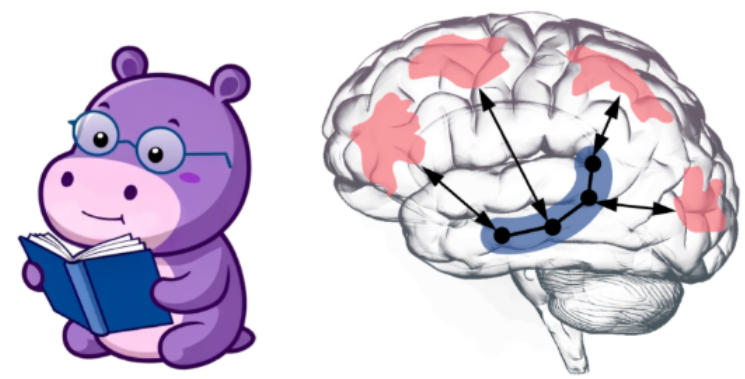
### 1. *The Imitation Game.*

I PROPOSE to consider the question, 'Can machines think?'

This should begin with definitions of the meaning of the terms 'machine' and 'think'. The definitions might be framed so as to

# Memory as Neurobiology of human cognition

It can be used as design inspiration — HippoRAG uses the human hippocampus and neocortex as core concepts for its architecture.



## HippoRAG

### HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models

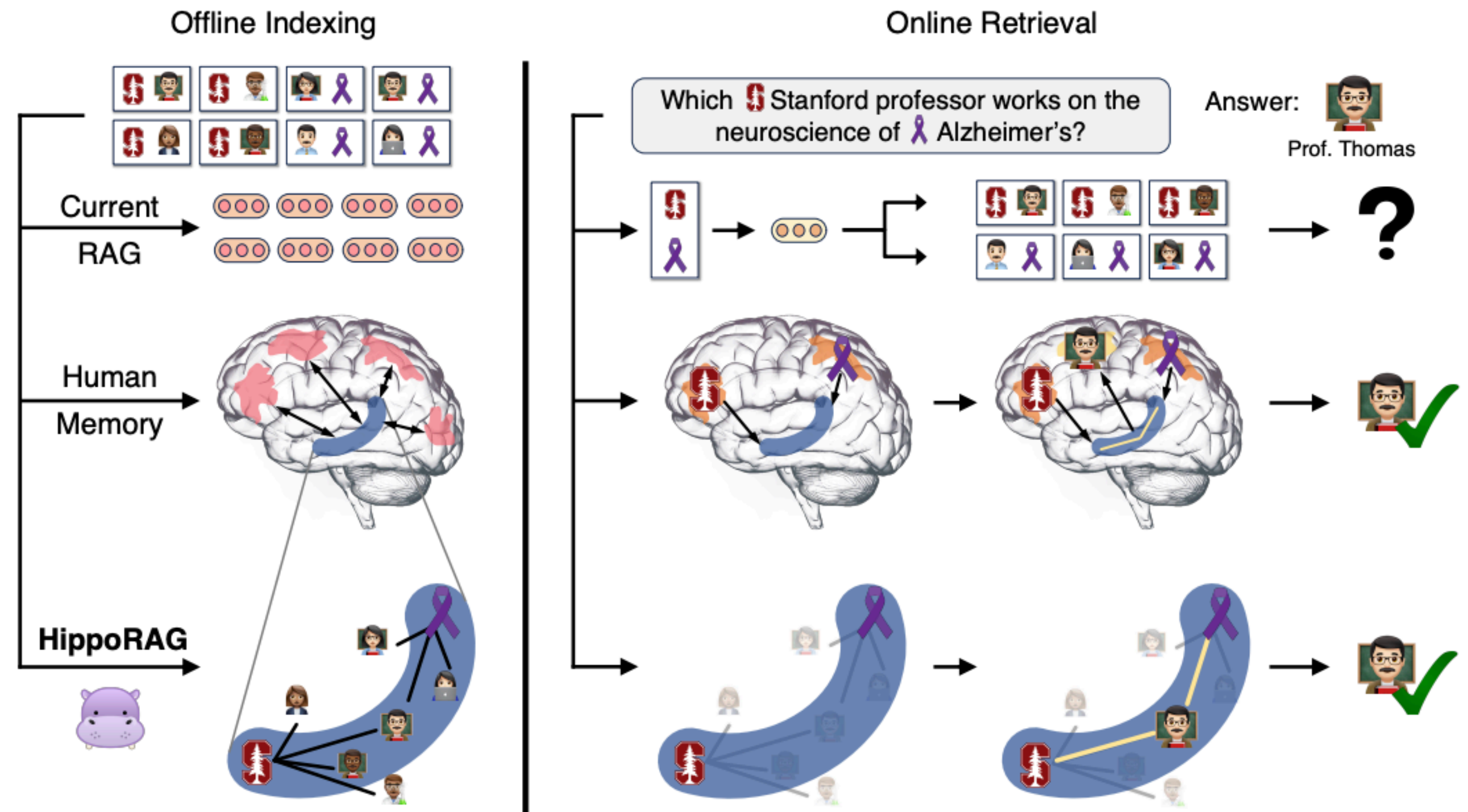
Bernal Jiménez Gutiérrez  
The Ohio State University

Yiheng Shu  
The Ohio State University

Yu Gu  
The Ohio State University

Michihiro Yasunaga  
Stanford University

Yu Su  
The Ohio State University



# Memory as Neurobiology of human cognition

Or as a way to describe system operations through human-cognition language, e.g. learn, understand, or have personalities.

 MemGPT  Letta

Machines that learn

 zep *Graphiti.*

Memory that understands  
when things change

 mem0

Give your AI a memory  
and personality

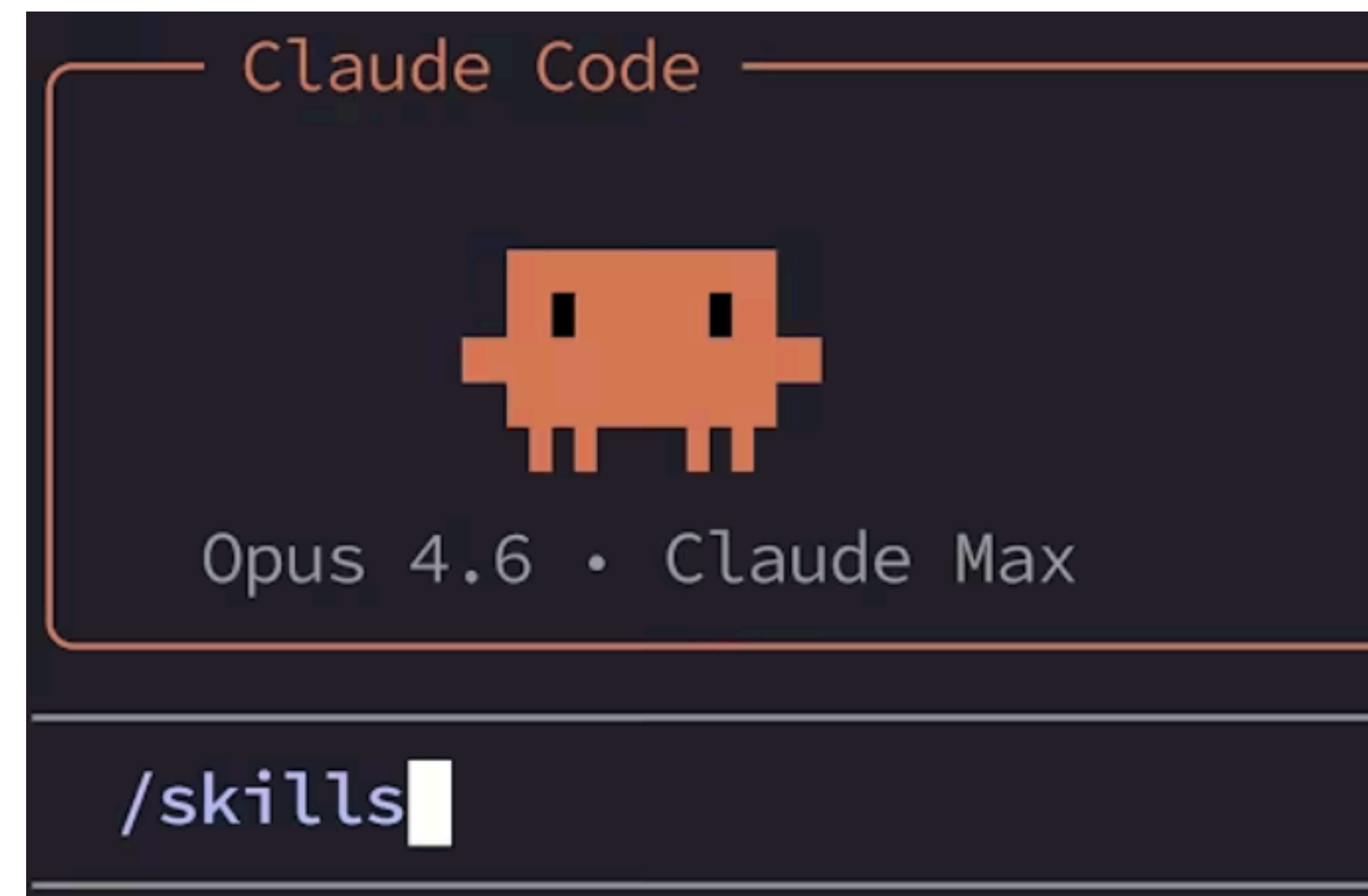
 SPECSTORY

AI rules that learn your style

# **3. Memory as Skills and Tools**

# Memory as Skills and Tools

This metaphor frames stored memories as reusable skills or tools that the agent can draw on for specific tasks.



Skills are folders of instructions, scripts, and resources that Claude loads dynamically

The AI identifies the best tool for the task,

# Metaphors both illuminate and hide

These metaphors help us define and make sense of AI memory, but they also shape how AI memory is imagined by centering particular ideas while marginalizing others.

*Reference:* **Metaphors for designers working with AI**

Dave Murray-Rust<sup>a,\*</sup>, Iohanna Nicenboim<sup>a</sup>, Dan Lockton<sup>b</sup>

<sup>a</sup>Delft University of Technology, The Netherlands

<sup>b</sup>Eindhoven University of Technology, The Netherlands

\*corresponding e-mail: d.s.murray-rust@tudelft.nl

[doi.org/10.21606/drs.2022.667](https://doi.org/10.21606/drs.2022.667)

**Abstract:** In this paper, we explore the use of metaphors for people working with artificial intelligence, in particular those that support designers in thinking about the creation of AI systems. Metaphors both illuminate and hide, simplifying and connecting to existing knowledge, centring particular ideas, marginalising others, and shaping fields of practice. The practices of machine learning and artificial intelligence draw heavily on metaphors, whether black boxes, or the idea of learning and training, but at the edges of the field, as design engages with computational practices, it is not al-

# Imagining new metaphors

We here introduce seven less-used metaphors to “connect previously disassociated domains to generate new ideas”.

## Generative metaphor: A perspective on problem-setting in social policy

DONALD A. SCHÖN

**Metaphor Cards**  
A Generative Design Toolkit

**THEATRE**

*“[The ICTR] is important for the people of Rwanda ...it shows that the international community is interested in what happened to them, interested in their welfare. It’s symbolic in another sense that the international system because of its nature can’t really prosecute everybody, so you have to concentrate on a few symbols of those offenses. ...Pick them out, make sure they are prosecuted. And in that way you can then send the message to people of the same status who are in other jurisdictions, a message of deterrence that accountability even at that level is possible.”*

— Hassan Jallow, ICTR Prosecutor

**THEATRE**

“There is no culture that has not had a theatre in some form, for theatre is the art of people acting out—and giving witness to—their most pressing, most illuminating, and most inspiring concerns. Theatre is at once a showcase and a forum, a medium through which a society displays its ideas, fashions, moralities, and entertainments, and debates its conflicts, dilemmas, yearnings, and struggles.

...Theatre is above all, a living art form: an art that continually forms before our eyes and is continuously present to an audience even as it is presented by its actors.”

— Robert Cohen, Playwright

**BRIDGING CONCEPTS:** Audience, Conflict, Symbolic Action

source: <http://www.tribunavoice.org/voices/video/270> source: Cohen, 2013, Theater Brief, 10th Edition

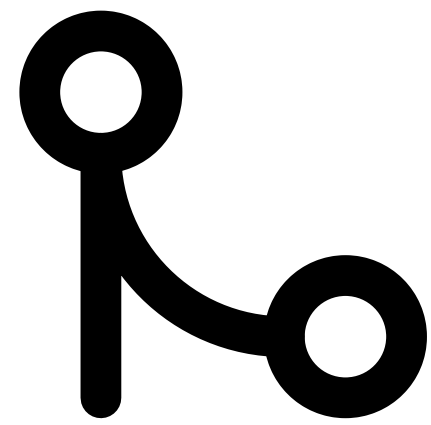
# Imagining new metaphors

## Methodological details on generative metaphor study

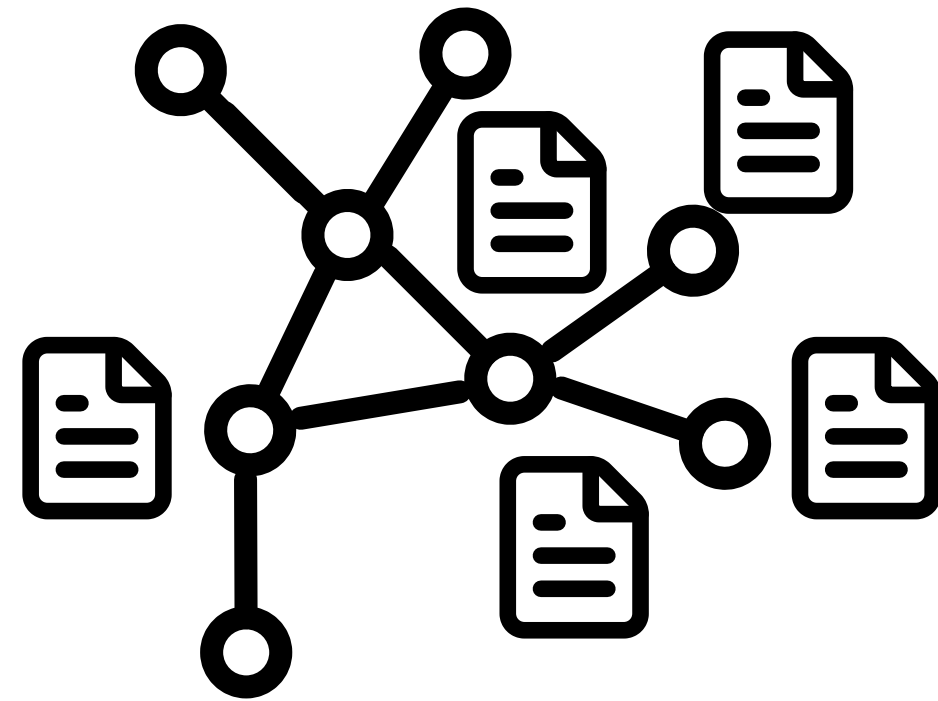
- Research-led iterative design process conducted over approximately three months.
- All authors participated in proposing, discussing, and defining candidate metaphors.
- Iterated refinement on metaphors themselves plus their specific properties and implications to serve clear and productive design inspiration to readers.

# Imagining new metaphors

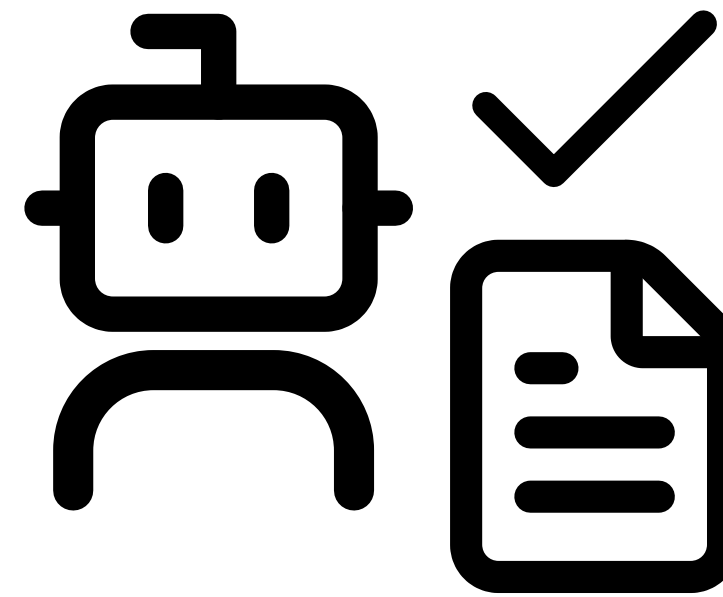
We here introduce seven less-used metaphors to “connect previously disassociated domains to generate new ideas”.



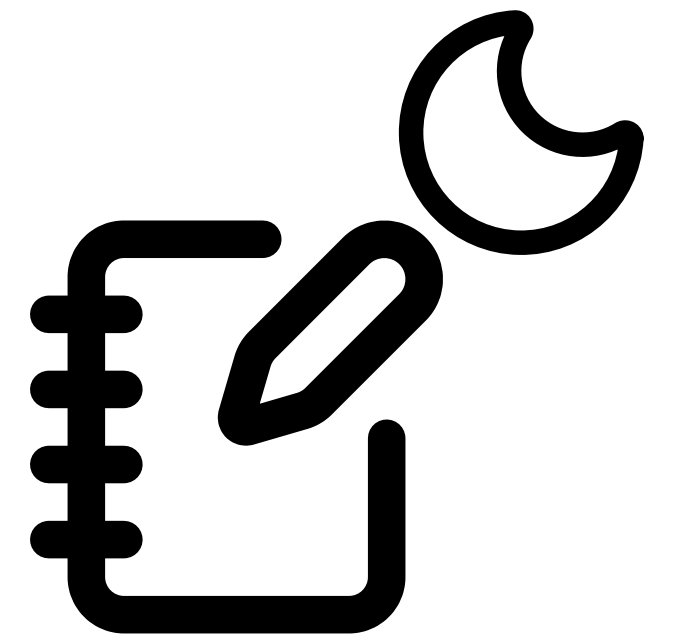
Software  
Version Control



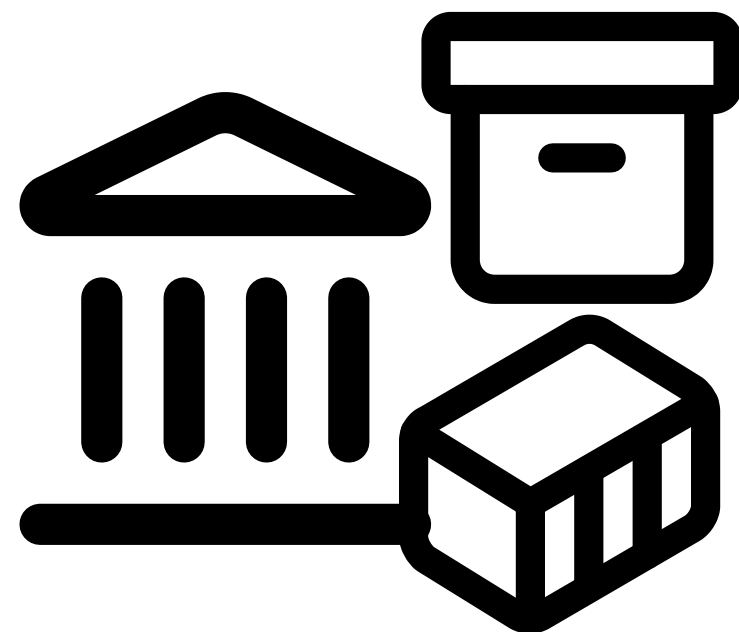
Zettelkasten



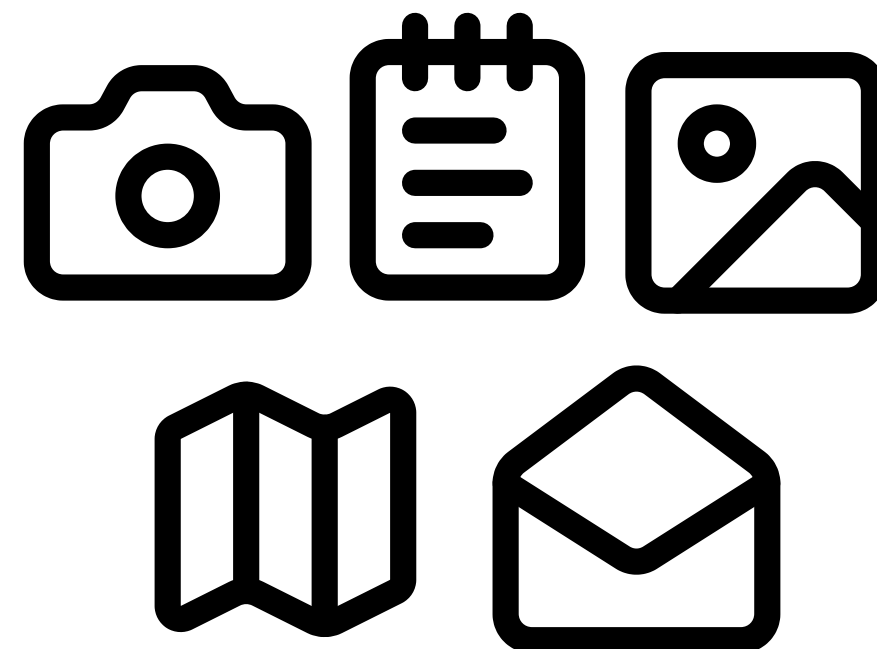
Requirements



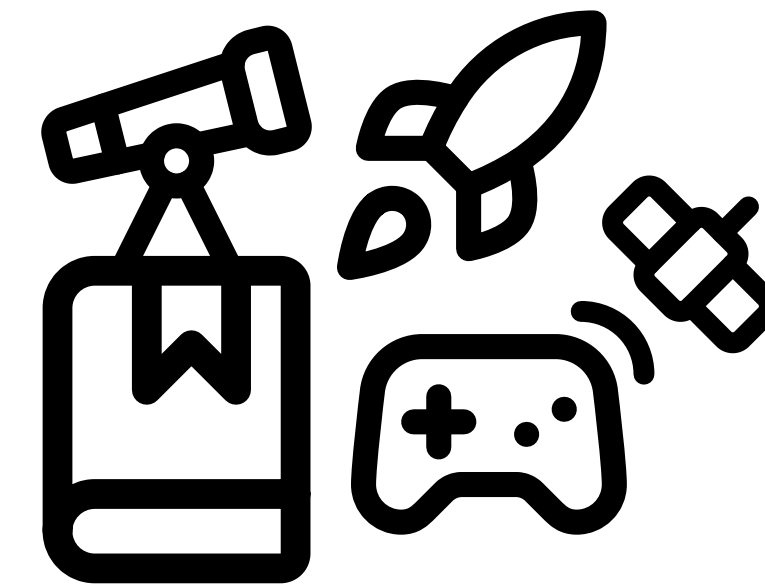
Personal Diary  
and Record



Community Archive



Cultural Probes

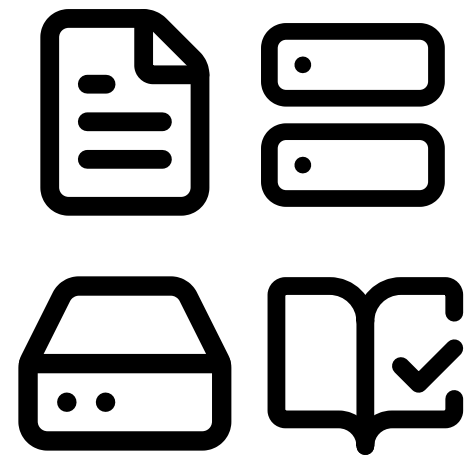


Science fiction  
tales/games

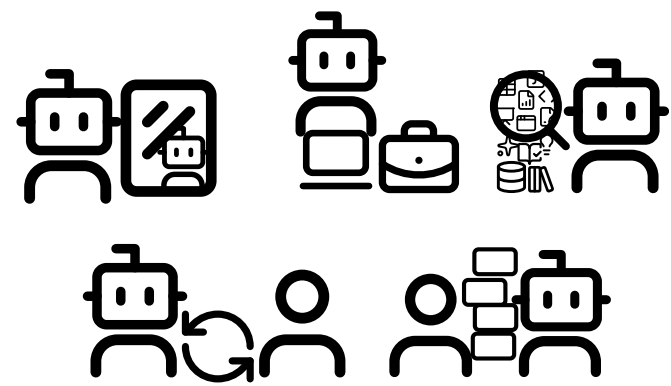
# Inspire new possibility from metaphors

These metaphors can be used as design tools alongside the three dominant metaphors, giving us new ways to think about AI memory.

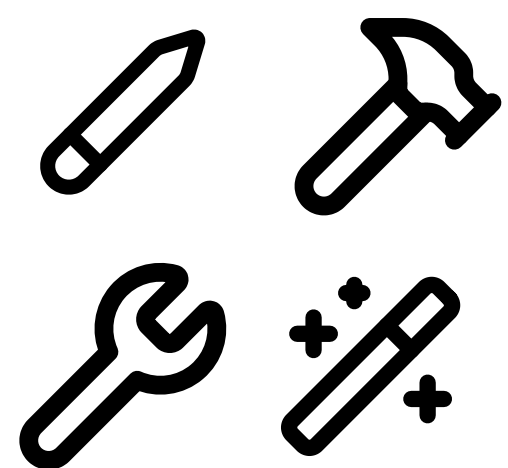
## 1. Dominant metaphors



Records, repositories,  
and database systems

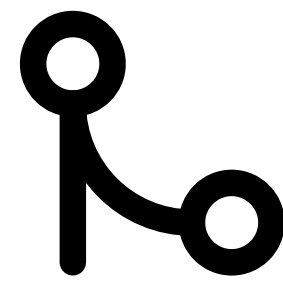


Neurobiology of  
human cognition

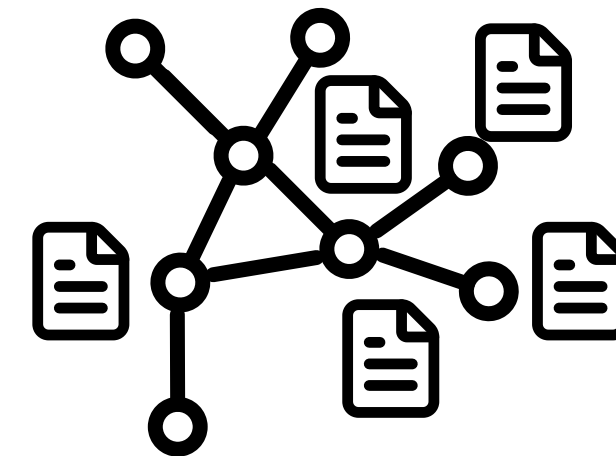


Skills and tools

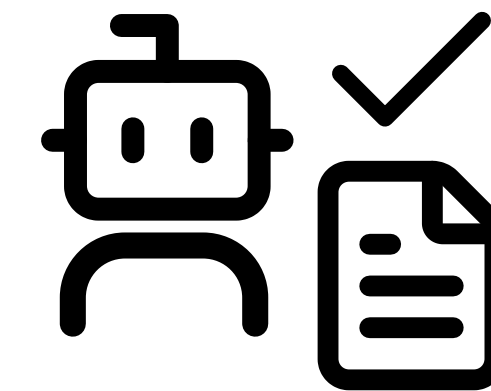
## 2. Less-used metaphors



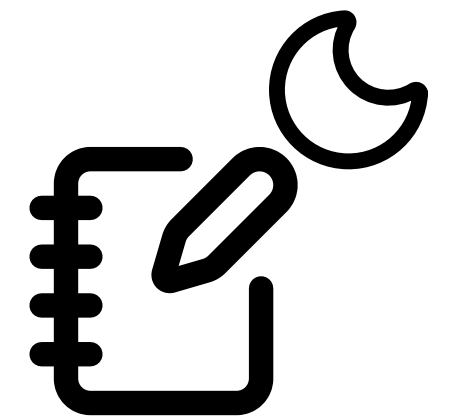
Software  
Version  
Control



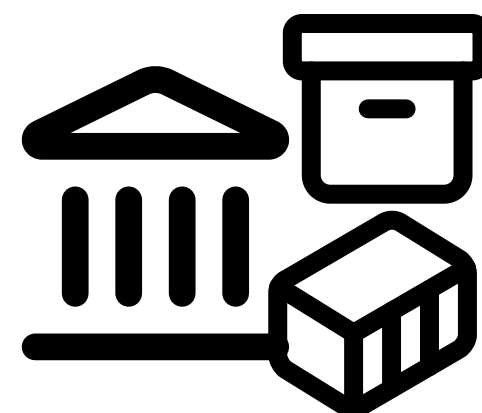
Zettelkasten



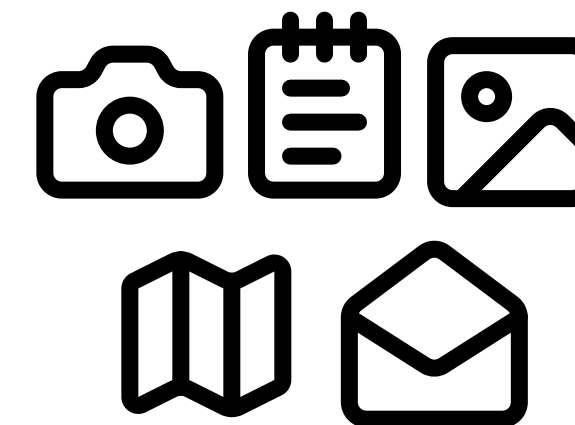
Requirements



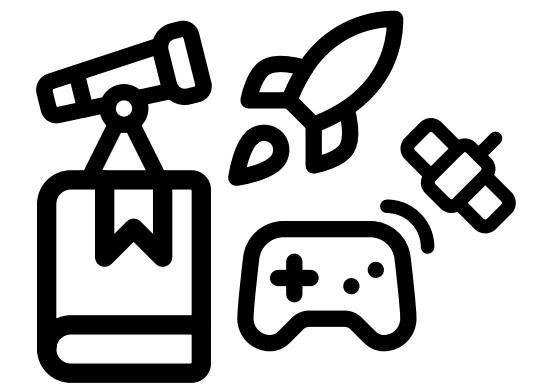
Personal  
Diary and  
Record



Community Archive



Cultural Probes



Science fiction  
tales/games

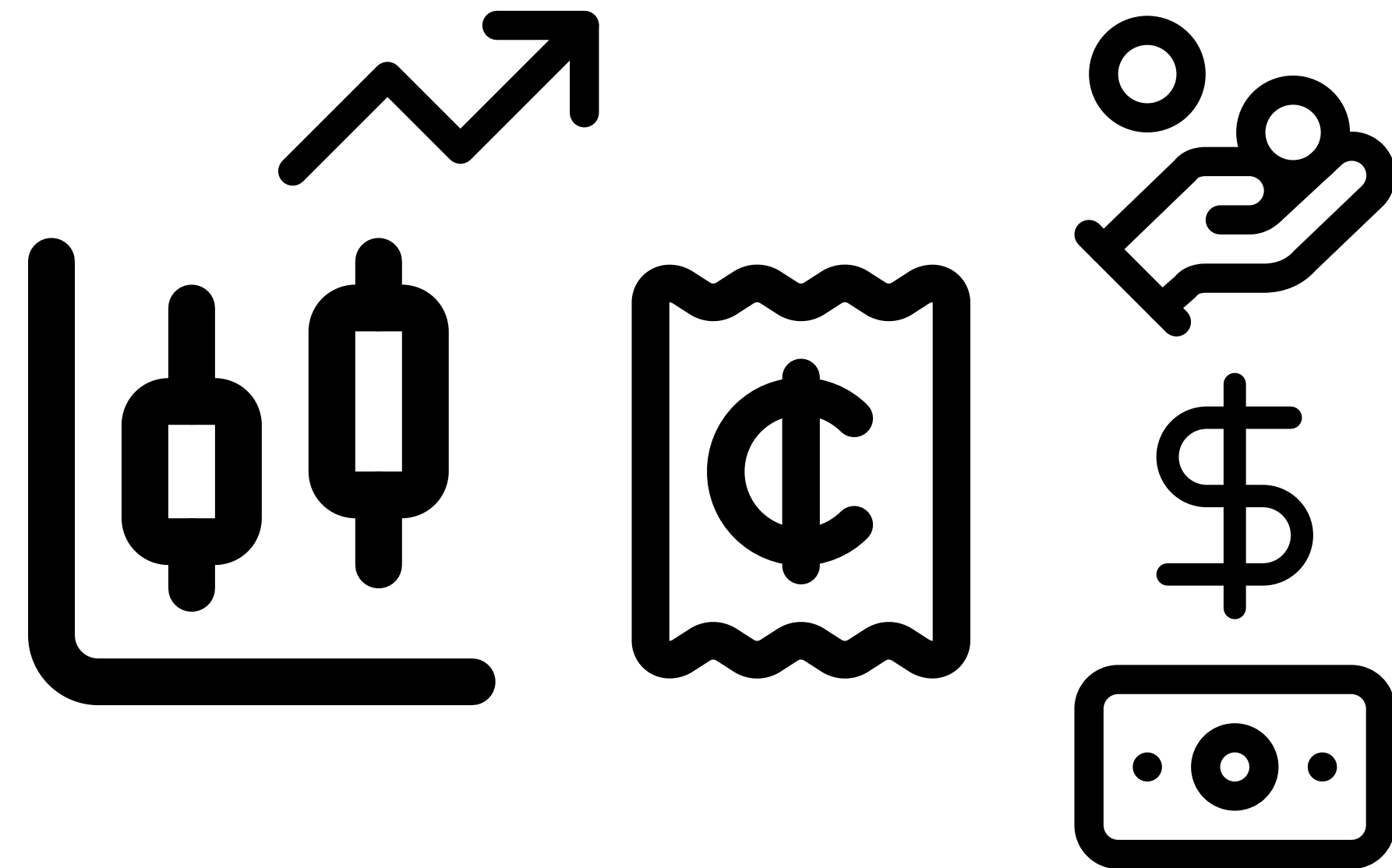
# **Metaphors for Memory: A real use case from my own work**

# E.g. AI Memory System for an Investment Team

We're applying our design space and metaphors in designing an in-house investment team's memory system.

## Properties

- Gather information from multiple sources and build an accurate, structured picture
- Distinguish different levels of information to avoid overfitting to weak signals or overlooking important ones
- Identify and help resolve disagreements between internal and external perspectives



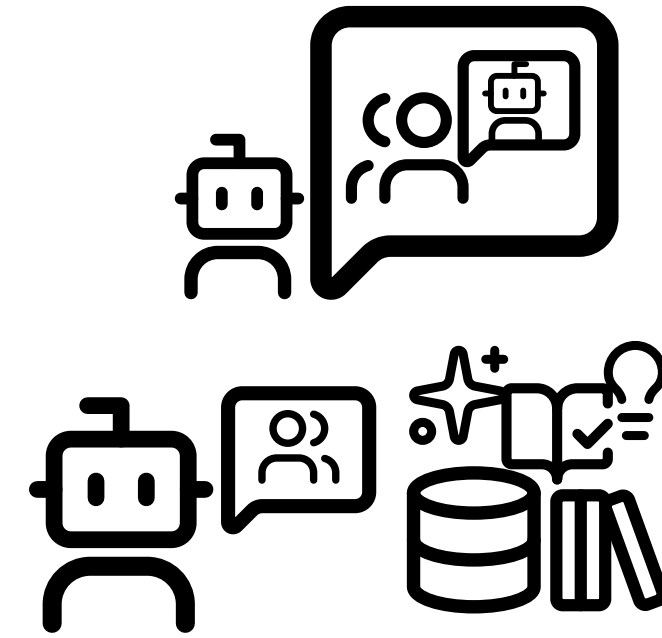
# E.g. AI Memory System for an Investment Team

I use the design space to think through its architecture, operations, and interaction, and to identify what properties the memory system needs for their workflow.

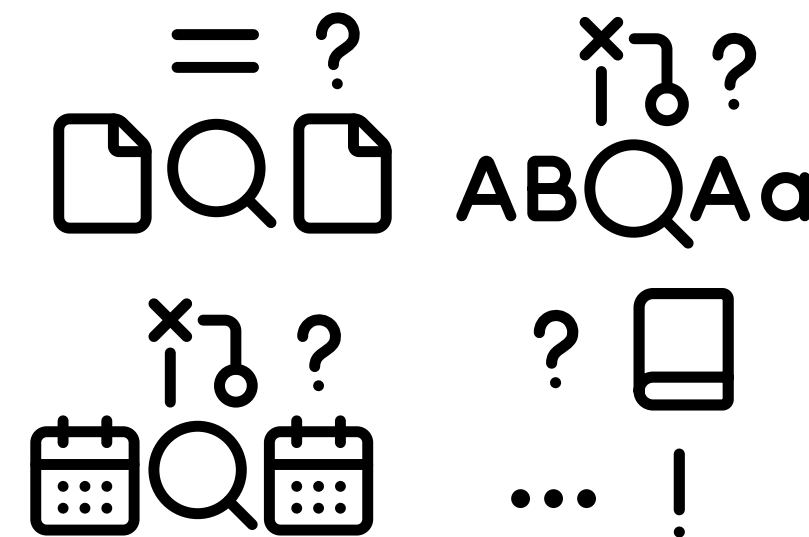
## Properties

- Gather information from multiple sources and build an accurate, structured picture
- Distinguish different levels of information to avoid overfitting to weak signals or overlooking important ones
- Identify and help resolve disagreements between internal and external perspectives

## Architecture, Operations, and Interaction



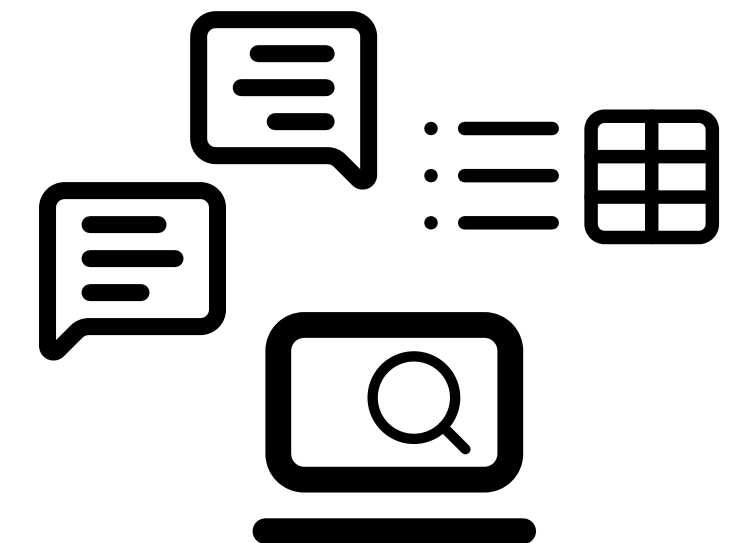
Purpose of Memory system



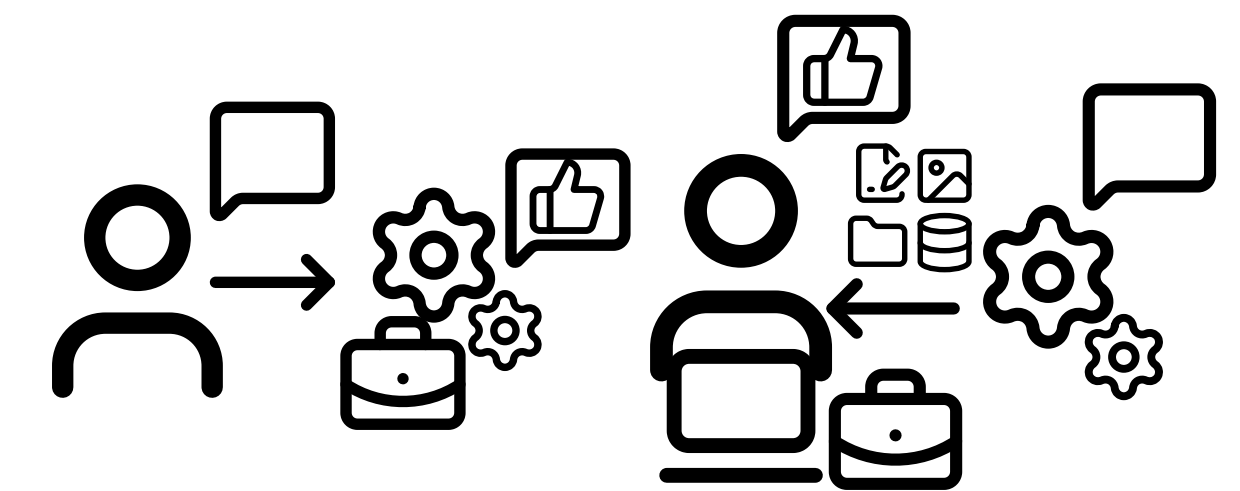
Operations

You are a helpful..  
 # Instructions  
 \* Only output...  
 \* Your response...  
 ## Examples  
 we design for user with AI memories — for instance, w

Architecture



Interface



Interaction

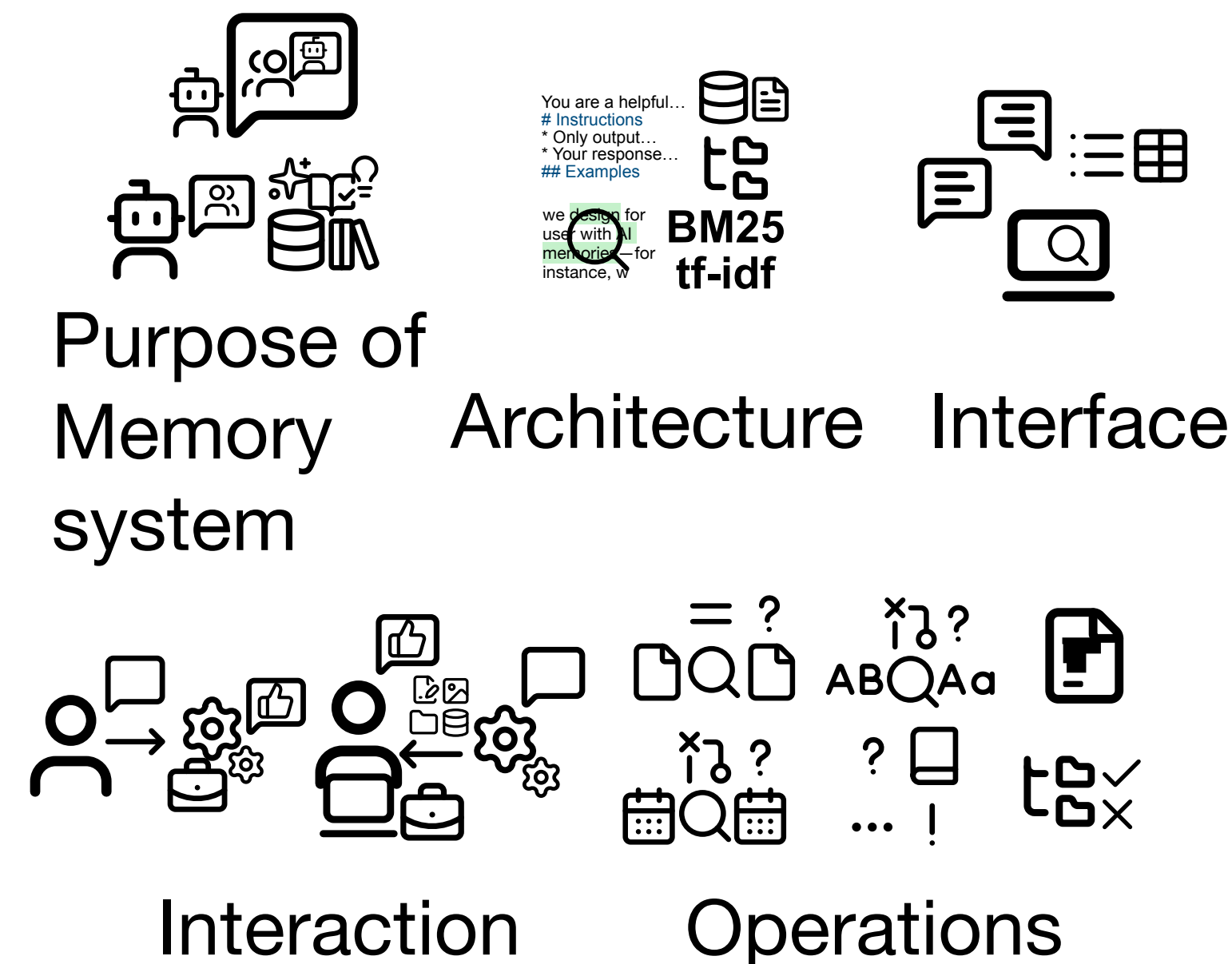
# E.g. AI Memory System for an Investment Team

Metaphors help us imagine possible improvements for the system.

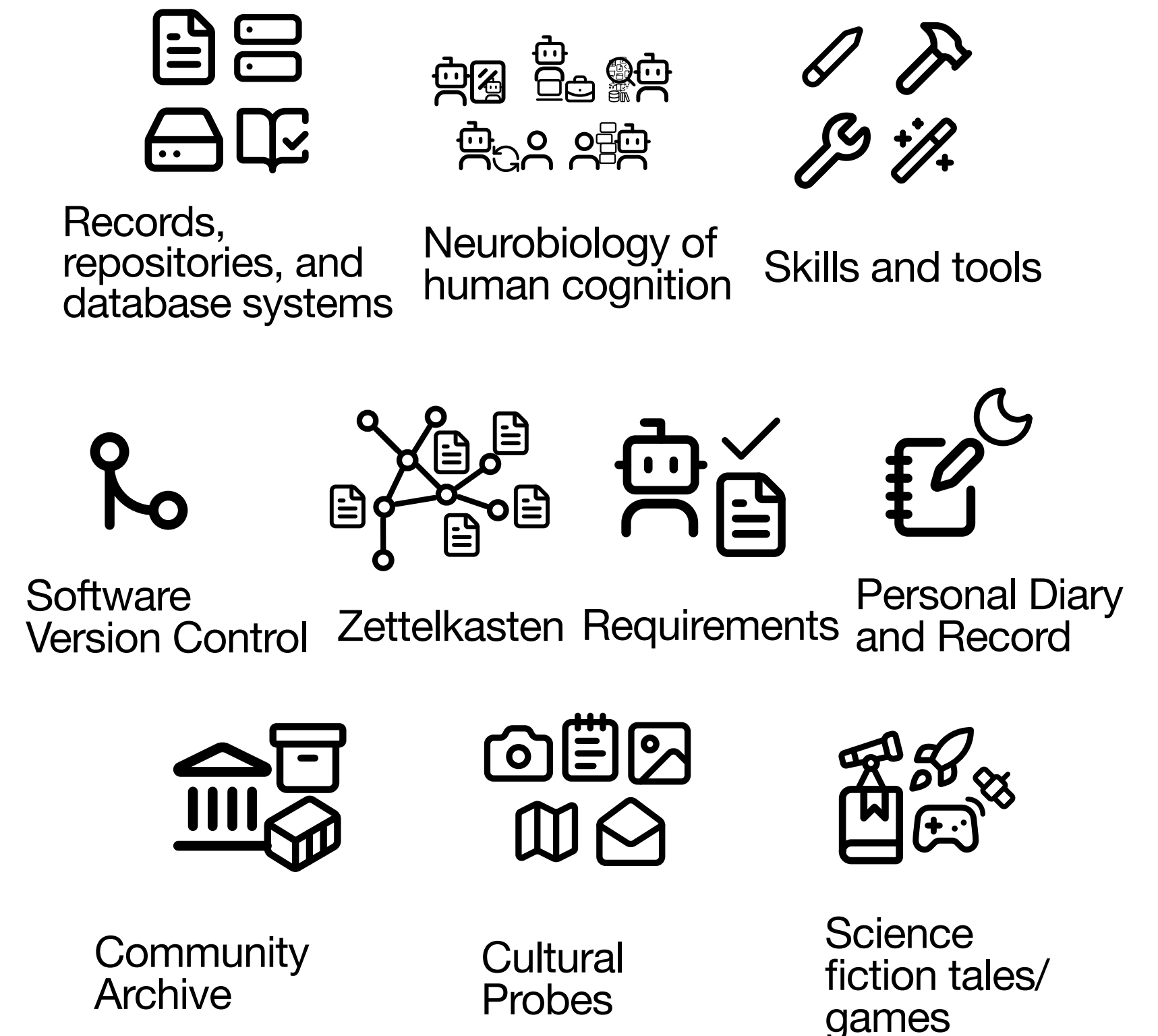
## Properties

- Gather information from multiple sources and build an accurate, structured picture
- Distinguish different levels of information to avoid overfitting to weak signals or overlooking important ones
- Identify and help resolve disagreements between internal and external perspectives

## Architecture, Operations, and Interaction

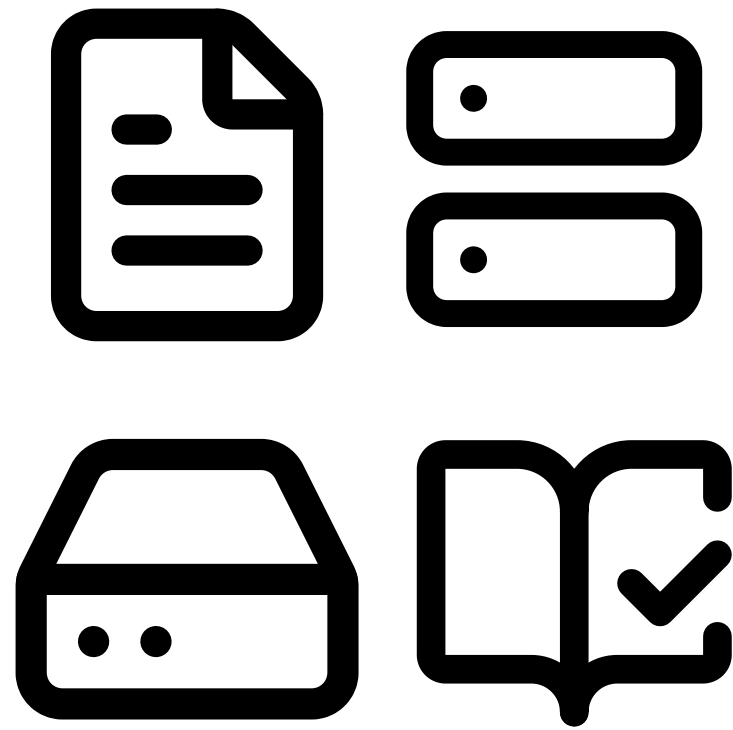


## Dominant and Less-used Metaphors



# E.g. AI Memory System for an Investment Team

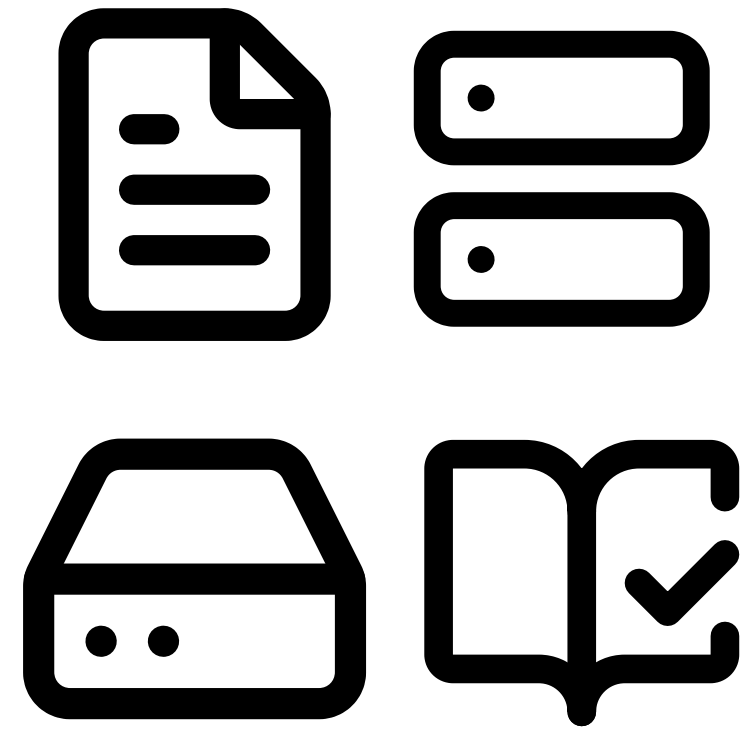
## 1. Dominant metaphors



Records, repositories,  
and database systems

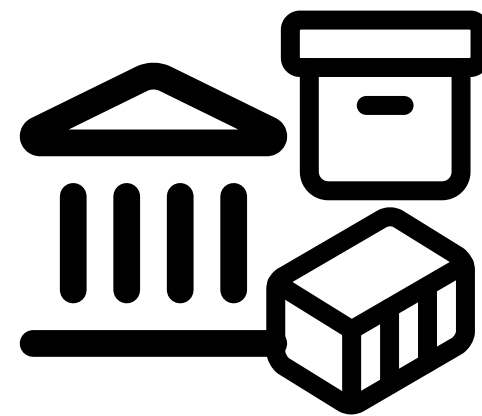
# E.g. AI Memory System for an Investment Team

## 1. Dominant metaphors



Records, repositories,  
and database systems

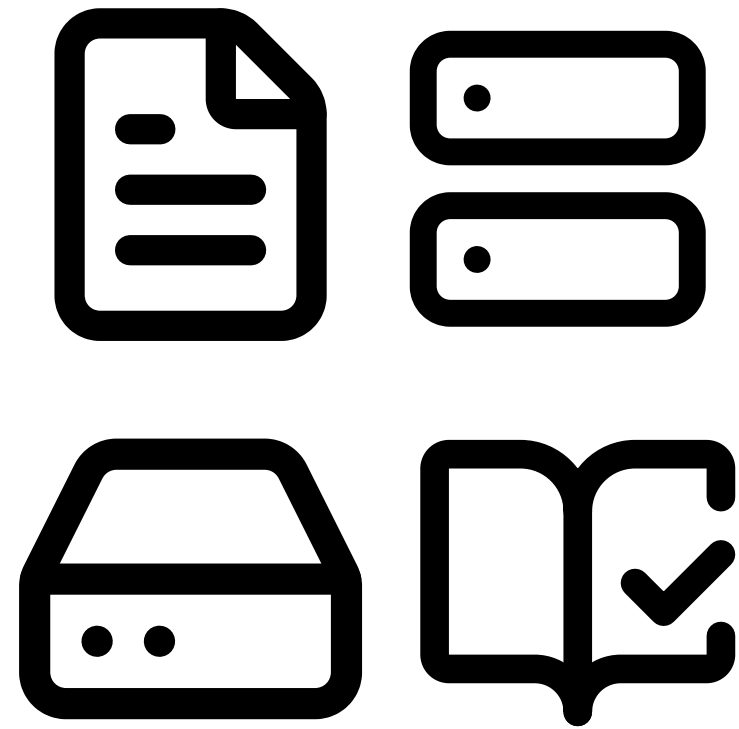
## 2. Less-used metaphors



Community Archive

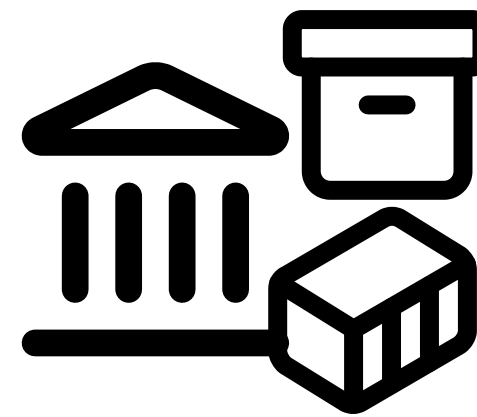
# E.g. AI Memory System for an Investment Team

## 1. Dominant metaphors



Records,  
repositories, and  
database systems

## 2. Less-used metaphors

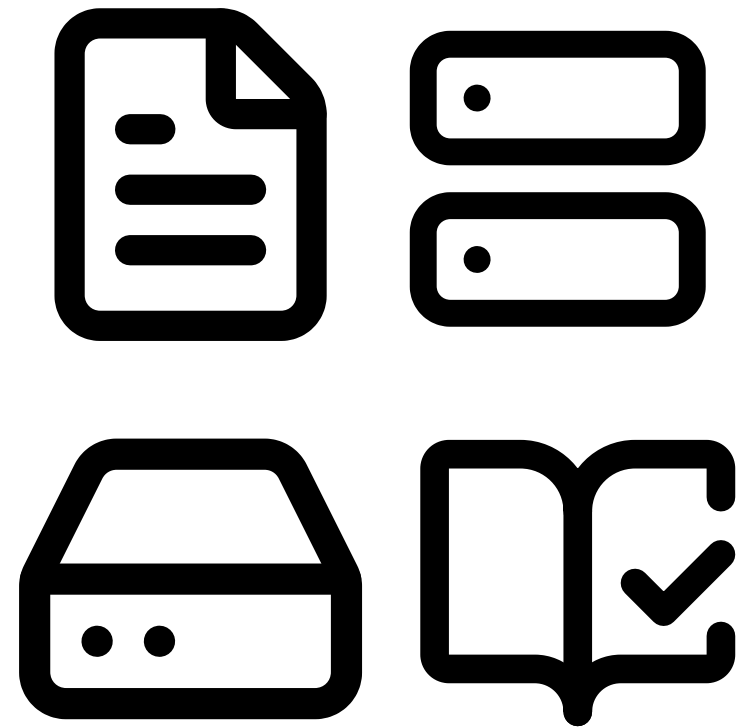


Community Archive

- Transmission and re-access/  
reconstruction
- Voluntary participation and production
- Cross-referencing and knowledge  
networks

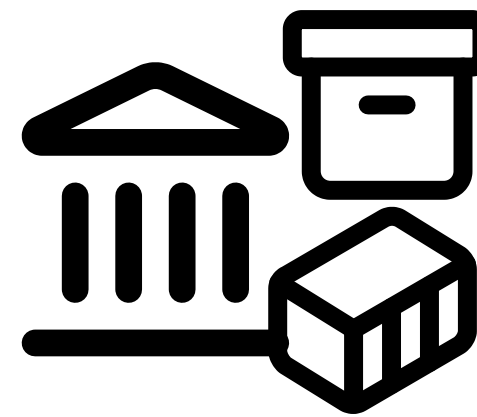
# E.g. AI Memory System for an Investment Team

## 1. Dominant metaphors



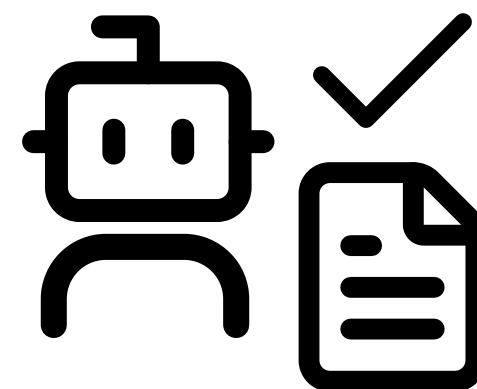
Records,  
repositories, and  
database systems

## 2. Less-used metaphors



Community Archive

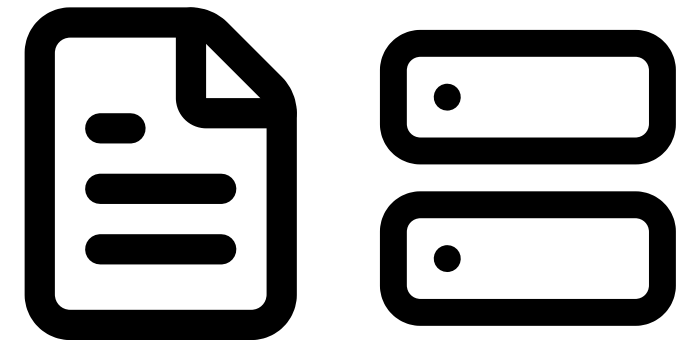
- Transmission and re-access/reconstruction
- Voluntary participation and production
- Cross-referencing and knowledge networks



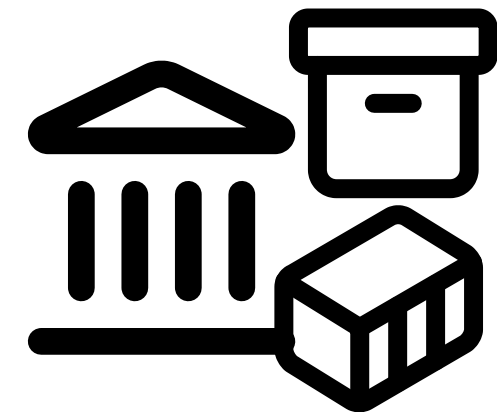
Requirements

- Shared grounding and interpretability
- Reducing redundancy and ambiguity
- Verifiability
- Normative reference
- Conflict-free
- Providing rationale

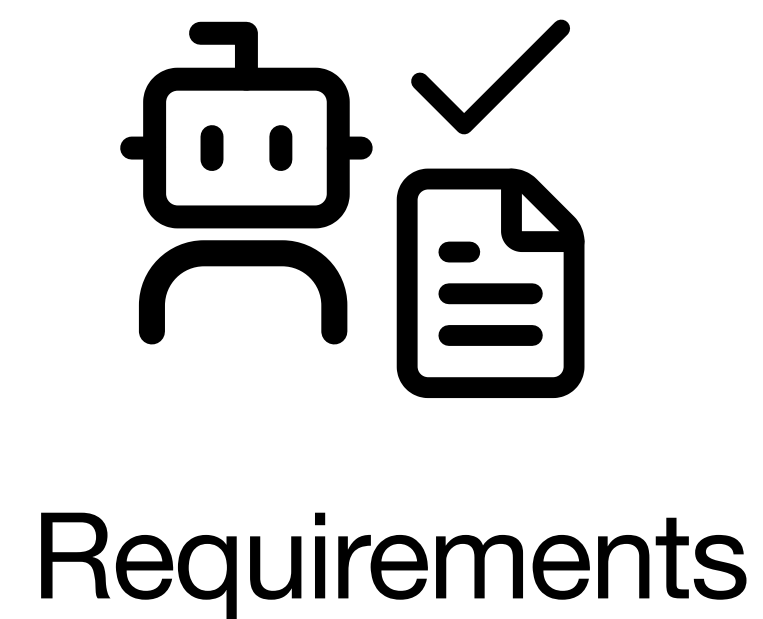
# E.g. AI Memory System for an Investment Team



Records,  
repositories, and  
database systems



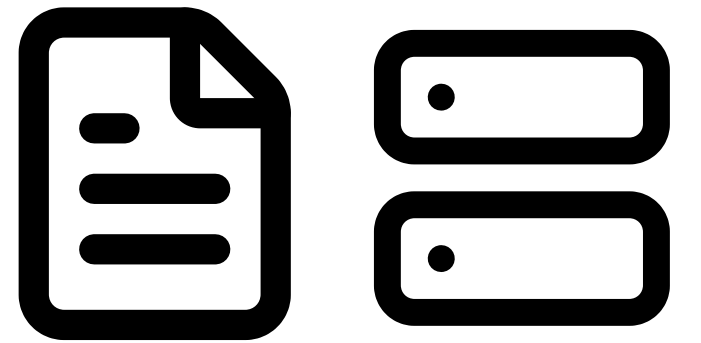
Community Archive



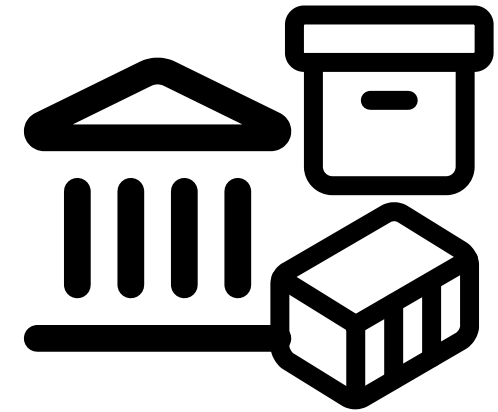
Requirements

*There may be different requirements  
and opinions within the team...*

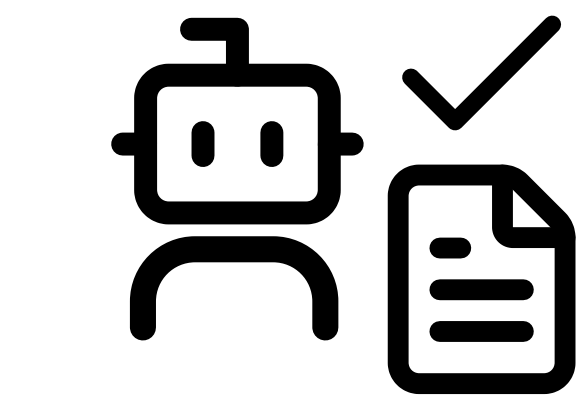
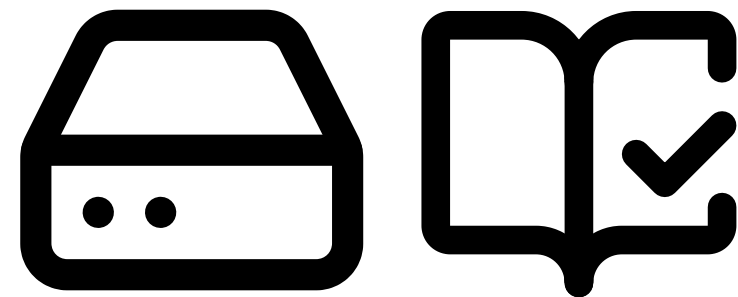
# E.g. AI Memory System for an Investment Team



Records,  
repositories, and  
database systems



Community Archive

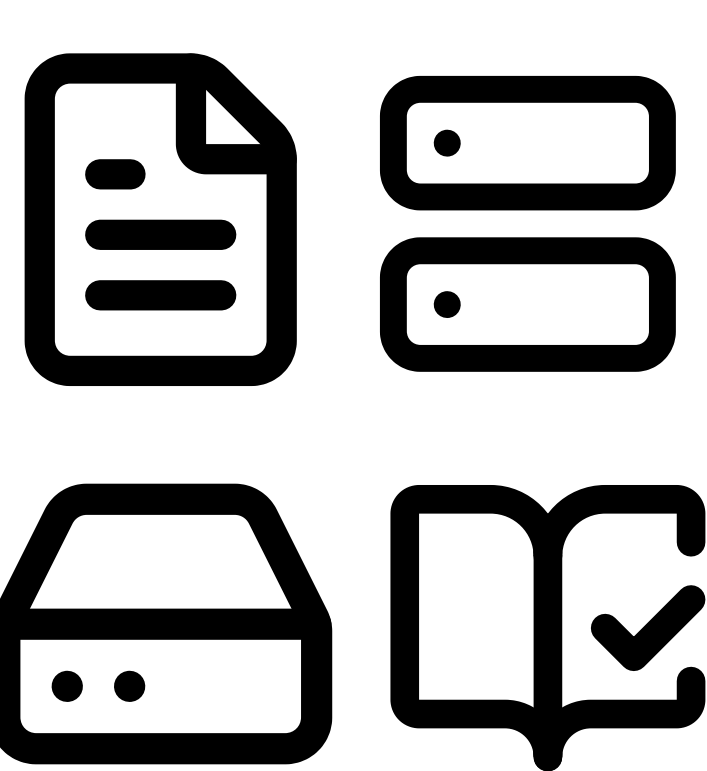


Requirements

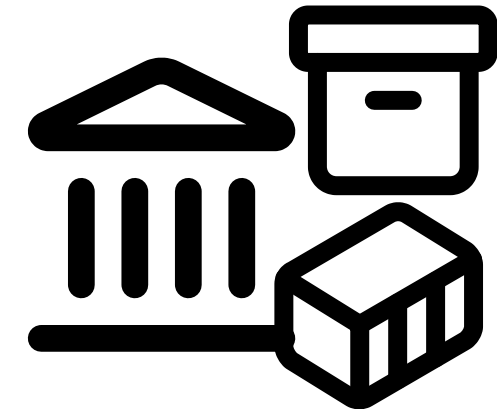
*There may be different requirements  
and opinions within the team...*

*What if we implement **operations** that  
help identify and resolve these  
conflicts? With records.*

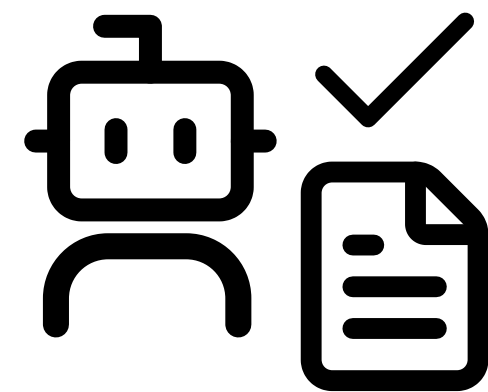
# E.g. AI Memory System for an Investment Team



Records,  
repositories, and  
database systems



Community Archive



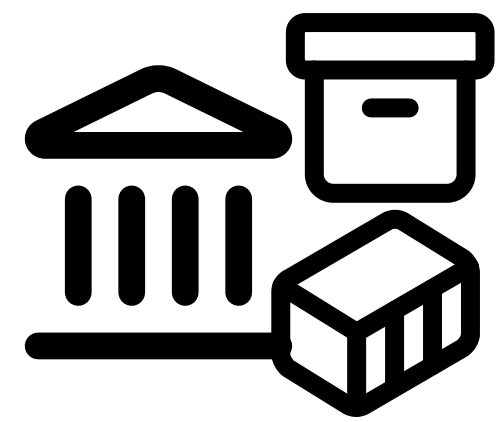
Requirements

*There may be different requirements and opinions within the team...*

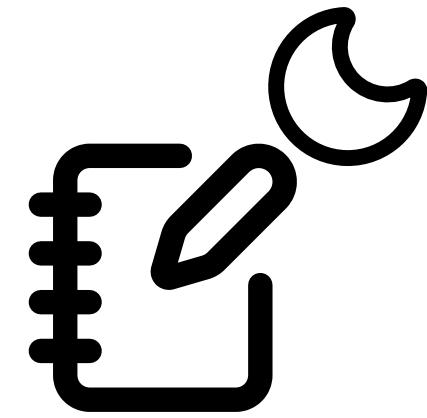
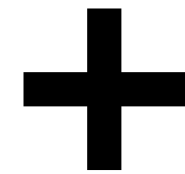
*What if we implement **operations** that help identify and resolve these conflicts? With records.*

*To support this, each **memory payload** should include who created the record and what context it came from.*

# E.g. Designing investment team agent

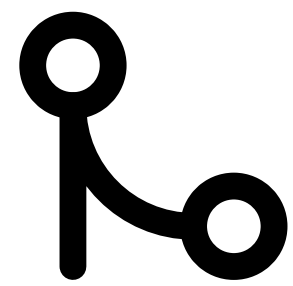


Community  
Archive

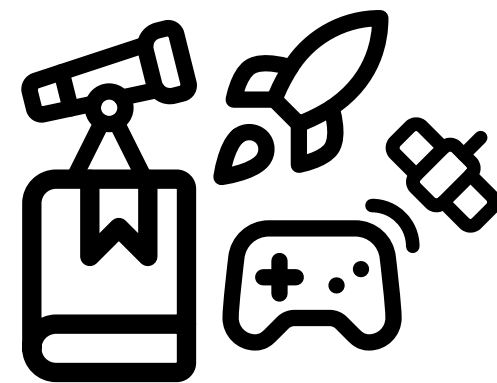
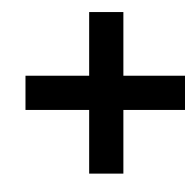


Personal  
Diary and  
Record

*Separate individual memory vaults  
with a shared common space*



Software  
Version  
Control



Science fiction  
tales/games

*In-house “style memory” for team-  
wide investment decision in unison*

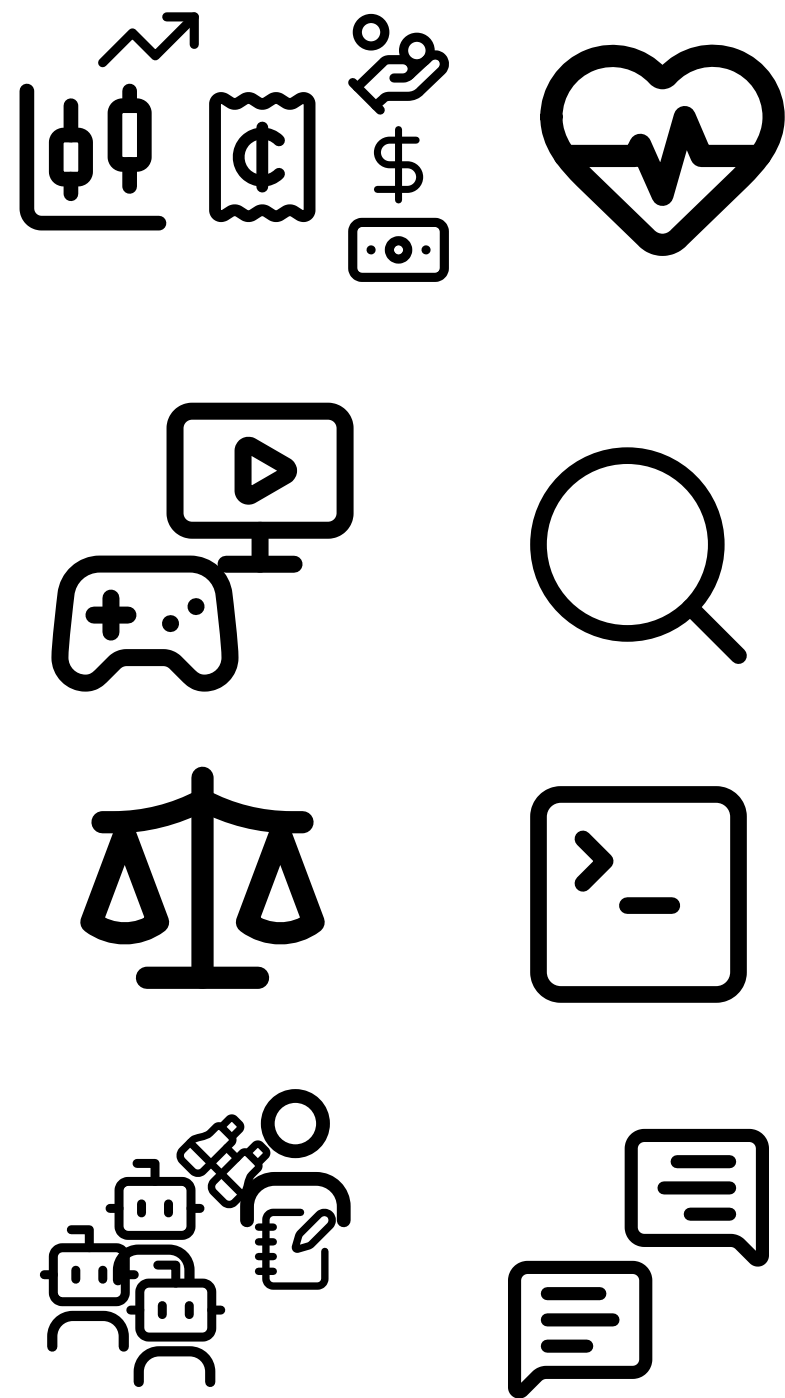


# Applying this framework to your own cases

## Systems

## Design Space

## Metaphors



### Design space of AI memory systems and interfaces

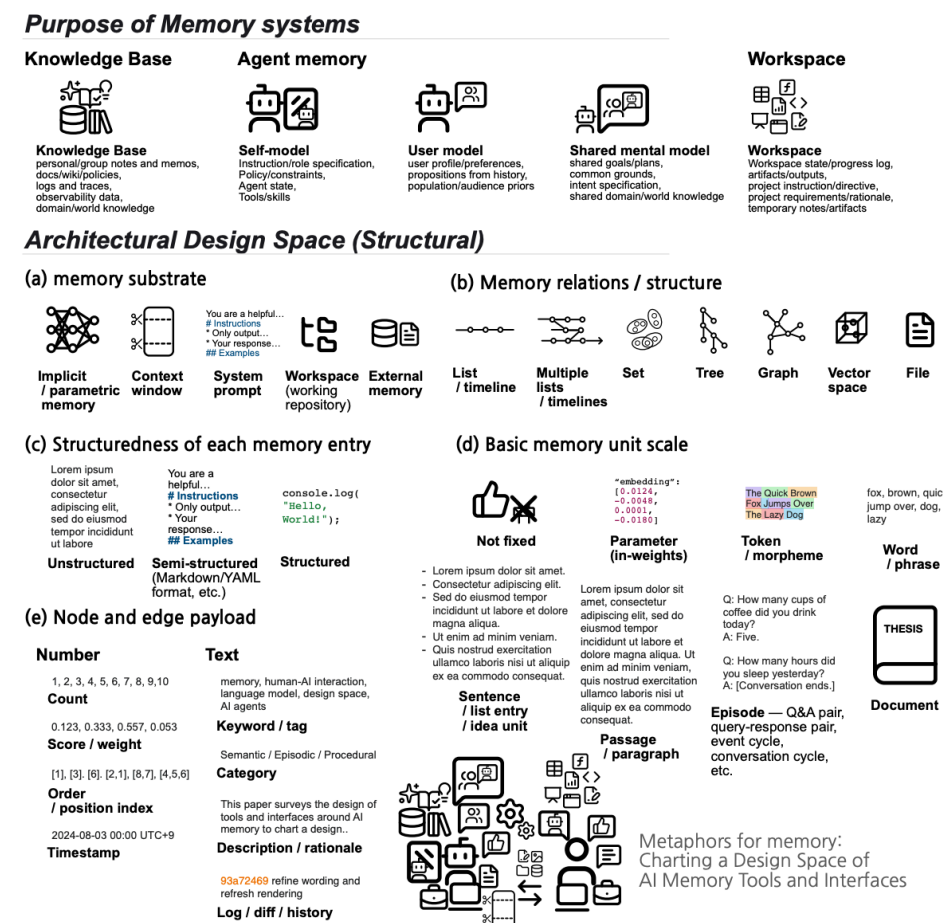


Figure 1: Design space of AI memory systems and interfaces

### Architectural Design Space (Interface and Interaction)

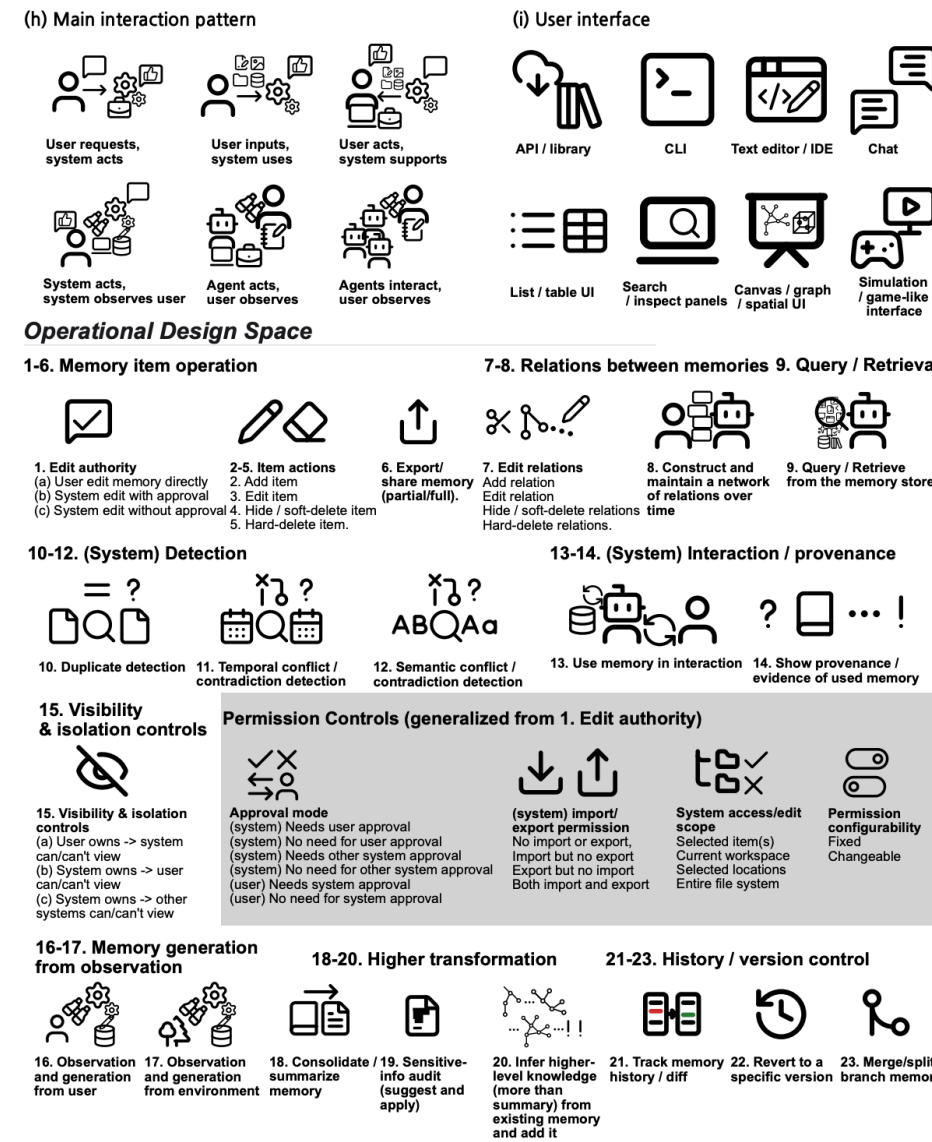
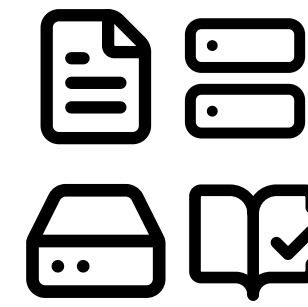
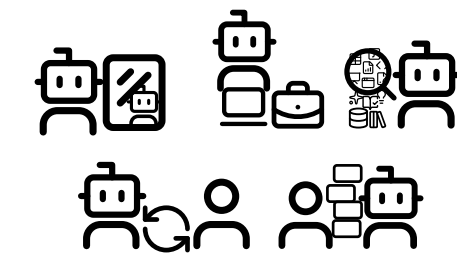


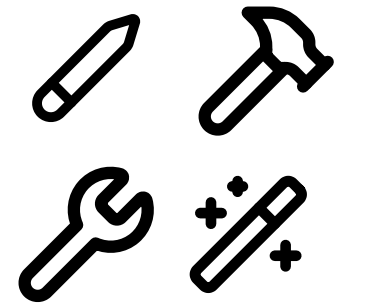
Figure 1: Design space of AI memory systems and interfaces (continued)



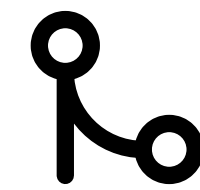
Records, repositories, and database systems



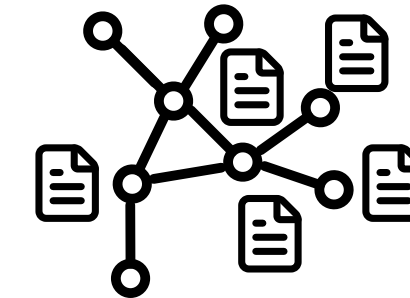
Neurobiology of human cognition



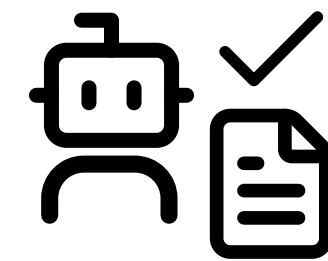
Skills and tools



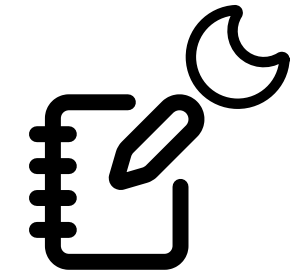
Software Version Control



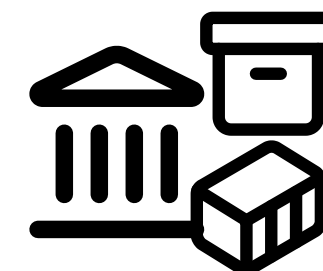
Zettelkasten



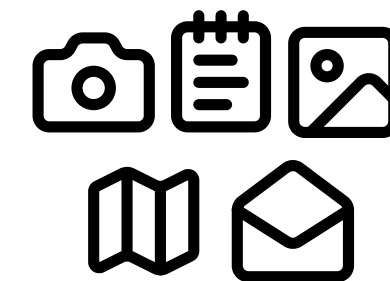
Requirements



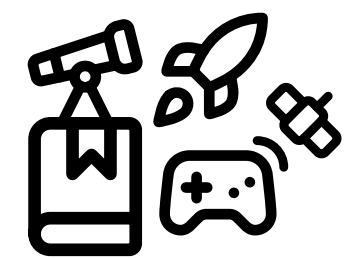
Personal Diary and Record



Community Archive



Cultural Probes



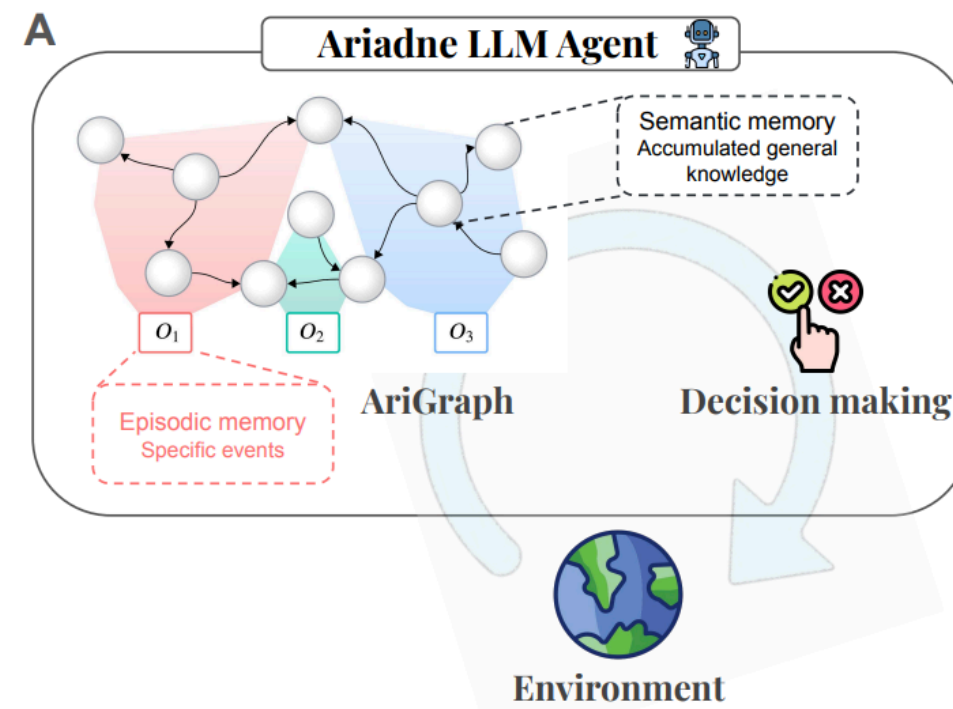
Science fiction tales/games

# For fellow researchers studying AI memory



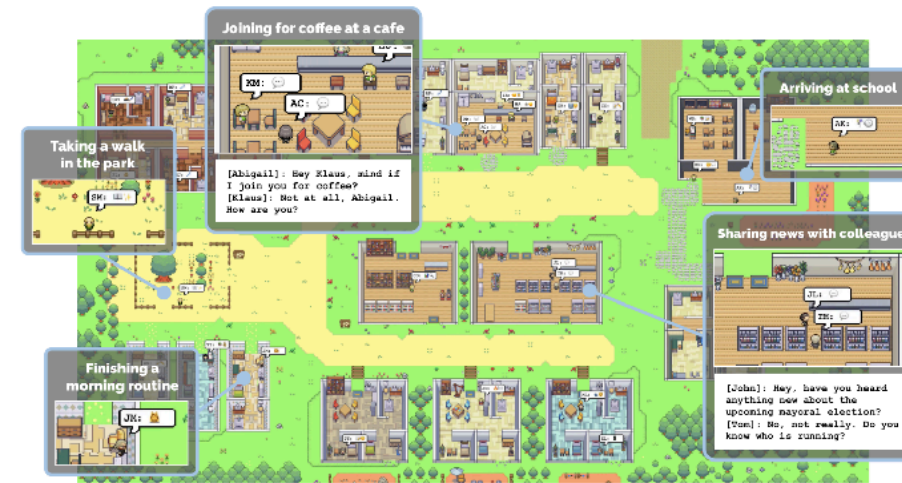
## AriGraph: Learning Knowledge Graph World Models with Episodic Memory for LLM Agents

Petr Anokhin<sup>1</sup>, Nikita Semenov<sup>2</sup>, Artyom Sorokin<sup>1</sup>, Dmitry Evseev<sup>2</sup>, Andrey Kravchenko<sup>1</sup>, Mikhail Burtsev<sup>3</sup> and Evgeny Burnaev<sup>2,1</sup>



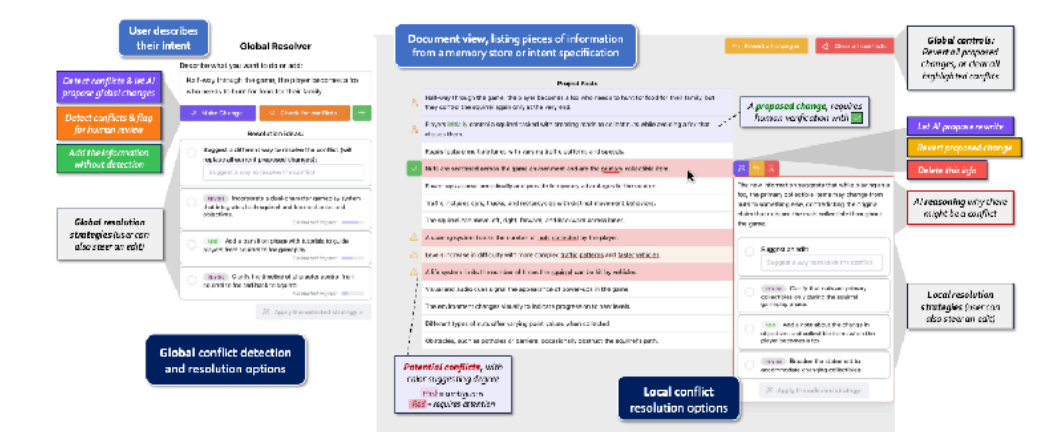
## Generative Agents: Interactive Simulacra of Human Behavior

Joon Sung Park, Stanford University, USA  
 Joseph C. O'Brien, Stanford University, USA  
 Carrie J. Cai, Google Research, Mountain View, CA, USA  
 Meredith Ringel Morris, Google DeepMind, Seattle, WA, USA  
 Percy Liang, Stanford University, USA  
 Michael S. Bernstein, Stanford University, USA



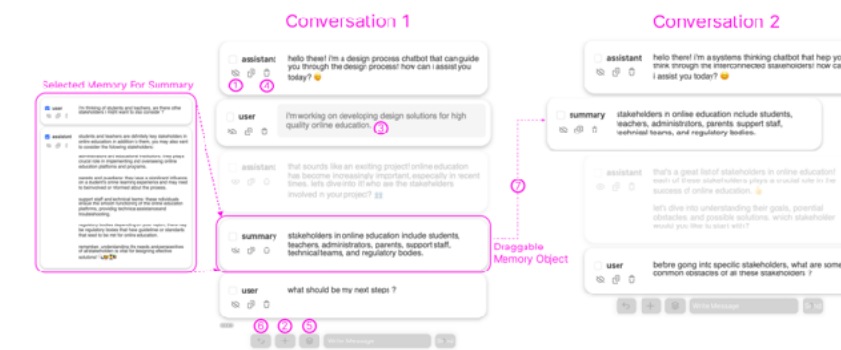
## Semantic Commit: Helping Users Update Intent Specifications for AI Memory at Scale

Priyan Vaithilingam<sup>1</sup>, Munyeong Kim<sup>2</sup>, Frida-Cecilia Acosta-Parenteau<sup>2</sup>, Daniel Lee<sup>3</sup>, Amine Mhedhbi<sup>4</sup>, Elena L. Glassman<sup>1</sup>, Ian Arawjo<sup>2</sup>



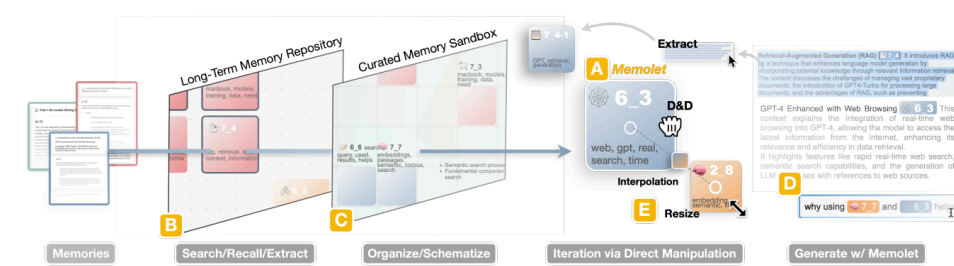
## Memory Sandbox: Transparent and Interactive Memory Management for Conversational Agents

Ziheng Huang, University of California-San Diego, CA, USA  
 Sebastian Gutierrez, Temple University, Philadelphia, PA, USA  
 Hemant Kamaana, Temple University, Philadelphia, PA, USA  
 Stephen MacNeil, Temple University, Philadelphia, PA, USA



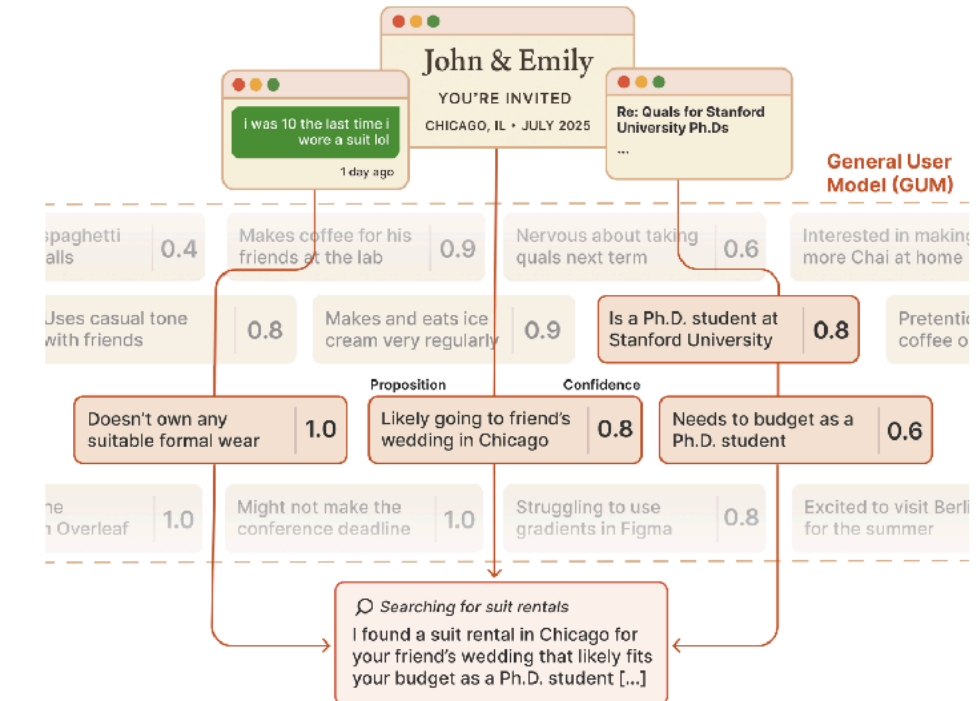
## Memolet: Reifying the Reuse of User-AI Conversational Memories

Ryan Yen, School of Computer Science, University of Waterloo, Canada  
 Jian Zhao, School of Computer Science, University of Waterloo, Canada



## Creating General User Models from Computer Use

Omar Shaikh, Stanford University, USA  
 Shardul Sapkota, Stanford University, USA  
 Shan Rizvi, Independent, New York City, USA  
 Eric Horvitz, Microsoft Research, Redmond, USA  
 Joon Sung Park, Stanford University, USA  
 Diyi Yang, Stanford University, USA  
 Michael S. Bernstein, Stanford University, USA



# Metaphors for Memory: Charting a Design Space of AI Memory Tools and Interfaces



# Questions