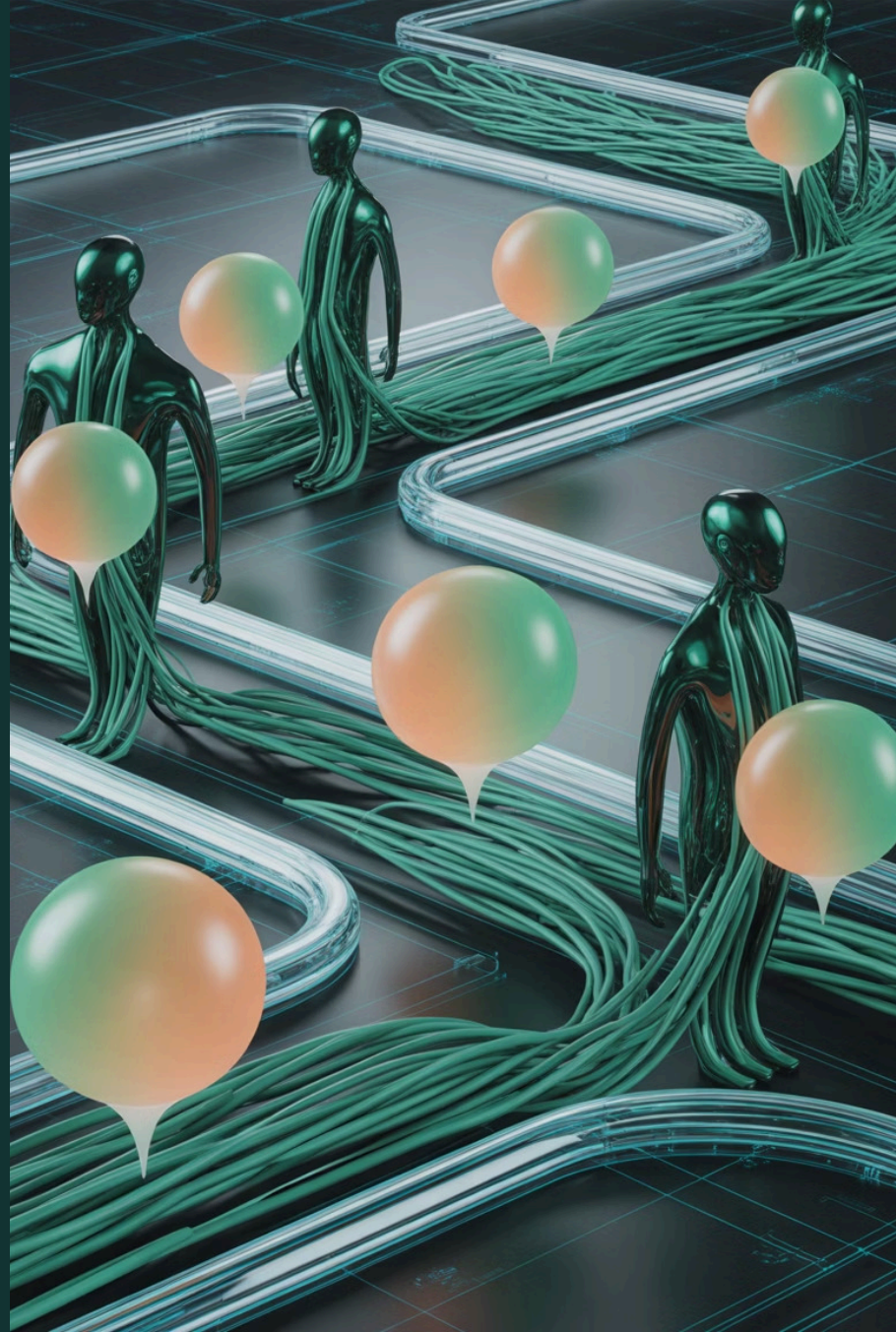
Types of Memory in Agentic AI

When to use which memory type – with real-world examples for AI professionals



Why Memory Matters

Retains Context

Agents maintain conversation flow

Enables Learning

Improves with each interaction

Powers Collaboration

Agents share knowledge effectively

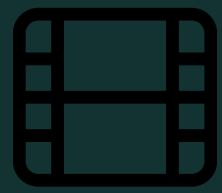
5 Key Memory Types

Short-Term

Working memory

Long-Term

Persistent storage



Episodic

Event-based memory

Semantic

Knowledge-based

Tool

Function access

Short-Term Memory

Multi-step conversations

Maintains immediate context

Tracks user requests

Coding assistant example

Discard after task

Session-level only

1. Short-Term (Working) Memory

- **Purpose**: Temporarily stores recent events, messages, or decisions.
- **Usage**: Helps agents stay coherent within a session or task.
- **Duration**: Session-level (discarded after task/session ends).

When to Use:

- For agents processing user instructions in a multi-step interaction.
- In tasks like code generation, multi-turn conversation, or summarization.

Example:
An AI coding assistant stores the last few user inputs and its own suggestions to maintain context during code completion.

Long-Term Memory

Persistent Learning

Remembers across multiple sessions

User Preferences

AI support agent personalizing responses

Vector DB Storage

Scales with growing knowledge

2. Long-Term Memory

- **Purpose**: Persistent storage of knowledge or experiences over time.
- **Usage**: Helps agents learn, evolve, and personalize behavior.
- **Duration**: Long-term, often stored in vector databases or knowledge graphs.

When to Use:

- When agents need to recall previous tasks, user preferences, or outcomes.
- For personalization, expertise building, and adaptive planning.

Example:
A customer service agent remembers a user's product preferences and complaint history to offer more relevant support in future interactions

Episodic Memory

Specific Past Events

Timestamps and contexts

Medical Example

Diagnosis process replay

Decision Audit

Enables reflection and learning

3. Episodic Memory

- **Purpose**: Stores structured records of specific past episodes or interactions (who did what, when, and why).
- **Usage**: Helps agents reflect on and learn from past experiences.

When to Use:

- When agents must analyze or reference specific past decisions or conversations.
- Useful in simulations, agent training, or decision traceability.

Example:
An AI medical assistant stores a case history of patient diagnosis and prescribed actions to evaluate its decision quality later

Semantic Memory

General Knowledge

Facts about the world

Travel Planning

Landmark facts for itineraries

Embeddings + KB

Structured knowledge access

4. Semantic Memory

- **Purpose**: Encodes factual knowledge and general world understanding.
- **Usage**: Shared reference for language understanding and reasoning.

When to Use:

- For general knowledge access, like understanding that "Paris is the capital of France".
- Often implemented using embeddings + external knowledge bases.

Example:
An AI travel planner uses semantic memory to understand that Eiffel Tower is a tourist attraction in Paris.

Tool Memory

API Selection

Choosing right integration

DevOps Example

Workflow automation

Usage Patterns

Tool effectiveness history

Function Access

How to invoke tools

5. Tool Memory

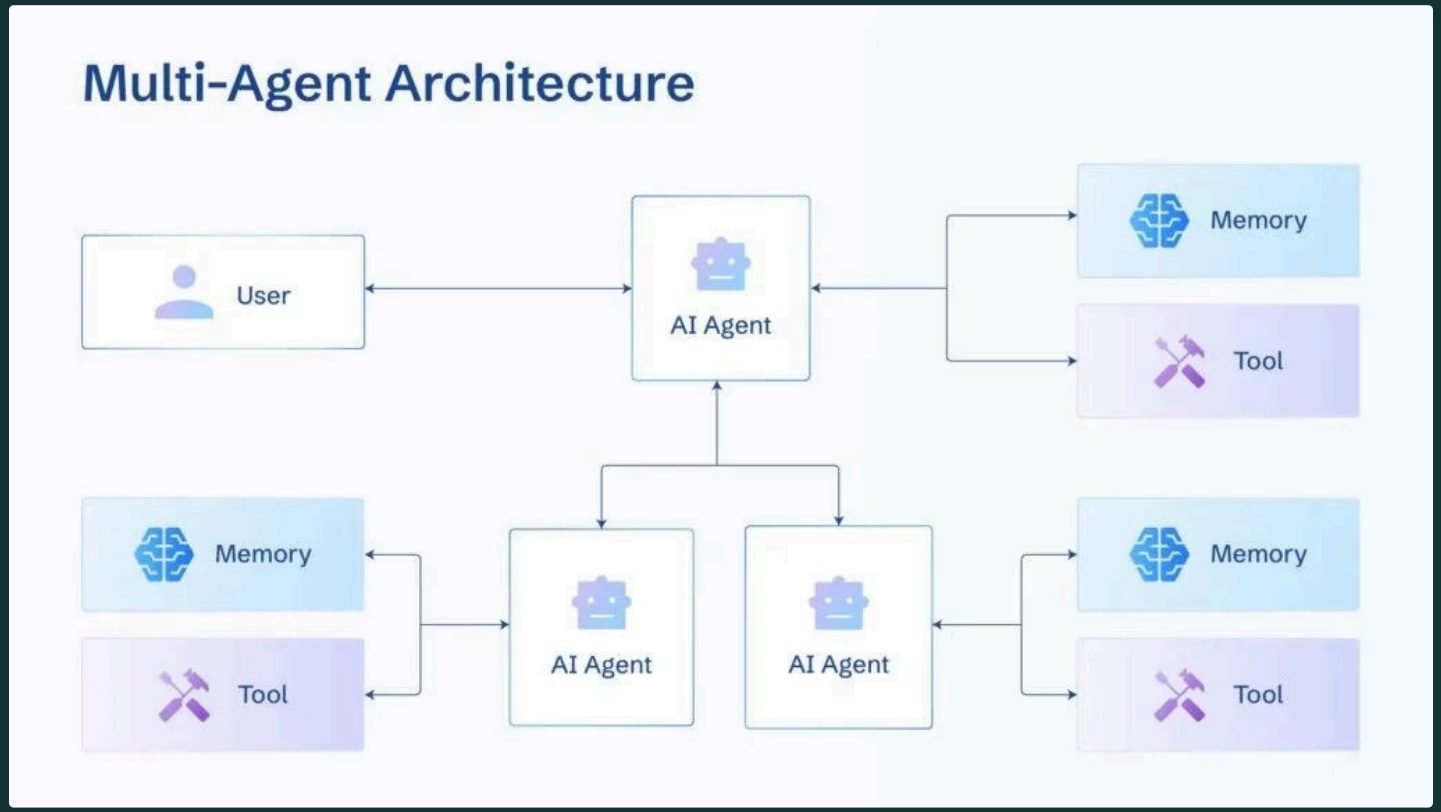
- **Purpose**: Stores metadata about tools, functions, or APIs an agent can invoke.
- **Usage**: Enables agents to reason about and select appropriate tools.

When to Use:

- In environments where agents can choose from multiple APIs or tools.
- Helps maintain performance and flexibility.

Example:
An AI developer agent remembers how to call GitHub, Docker, or Slack APIs during a DevOps task

👫 Dedicated vs Common Memory



👫 Dedicated vs Common Memory

✳️ Dedicated Memory (Per-Agent)

Each agent has its **own memory space**, which includes:

- Short-term session memory
- Long-term personal experiences
- Specialization-relevant knowledge

Purpose:

- Enables autonomy
- Supports task-specific learning
- Ensures agents don't leak sensitive or irrelevant info to others

Example:
An HR assistant agent remembers only employee-related data, while a finance assistant retains budgets and forecasts.

👫 Dedicated vs Common Memory

🌐 Common Memory (Shared Across Agents)

A **shared memory pool** accessible by multiple agents:

- Contains global context (project goals, shared timelines)
- Stores collaboration history
- Includes shared tools or documents

Purpose:

- Enables multi-agent collaboration
- Ensures synchronized understanding
- Reduces redundant memory across agents

Example:
In an Agentic AI-powered product design system, design agents, research agents, and QA agents access a common memory of product specs and deadlines

Dedicated vs Shared Memory

Dedicated Memory

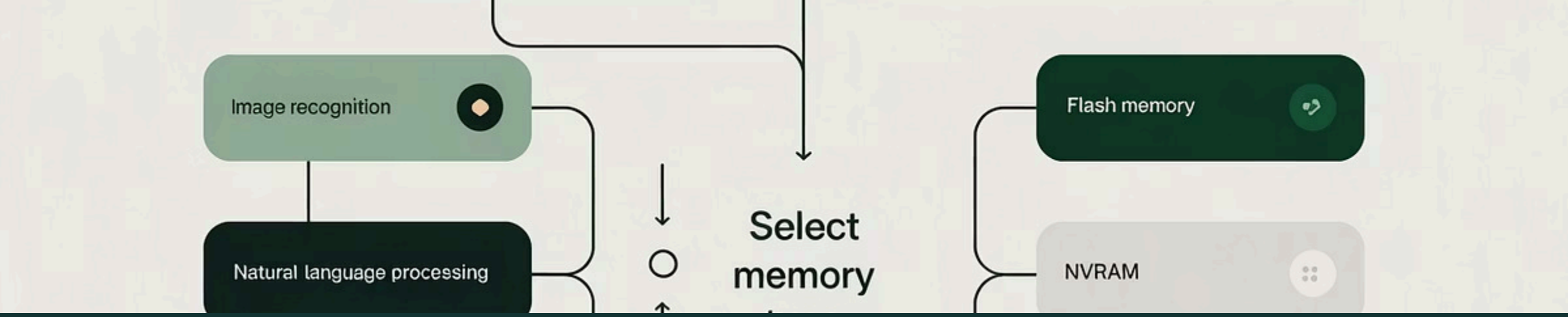
- Private to each agent
- Supports autonomy
- Personal task storage

Shared Memory

- Common knowledge pool
- Enables collaboration
- Goals and timelines

💡 Choosing the Right Memory: Quick Guide

Task Scenario	Memory Type	Dedicated or Shared
Multi-turn chat or UI agent	Short-Term	Dedicated
Personalized recommendation	Long-Term	Dedicated
Knowledge-based Q&A	Semantic	Shared
Simulation replay or strategy reflection	Episodic	Dedicated or Shared
Team project coordination	Common Context Memory	Shared
Tool selection in multi-agent workflows	Tool Memory	Shared



Continution...

Task	Memory Type	Scope
Chat	Short-Term	Dedicated
Personalization	Long-Term	Dedicated
General Q&A	Semantic	Shared
Reflection	Episodic	Mixed
Team Projects	Shared Context	Shared

Design Memory Intelligently

Task Flow

Use short-term for conversation

Personalization

Use long-term for preferences

Teamwork

Use shared for collaboration

Learning

Mix episodic + semantic

Memory Architecture Patterns

Tiered Architecture

Cache recent, archive old

Vector Storage

Similarity search capabilities

Relevance Filtering

Only store what matters

🛠️ Implementation Tools & Tech

Popular tools to implement memory in Agentic AI systems:

- **Vector Databases**: Pinecone, Weaviate, FAISS (for long-term, semantic)
- **Graph Databases**: Neo4j, for episodic relationships
- **LangChain / LlamaIndex**: For memory abstraction layers
- **Redis / In-memory stores**: For short-term or session-based memory
- **JSON/NoSQL stores**: For personal or agent-specific memory

Hope this helps! 🙂

Thank You!

Manikant Kandkuri 🙂

https://www.linkedin.com/in/manikantkandukuri/ 🙂