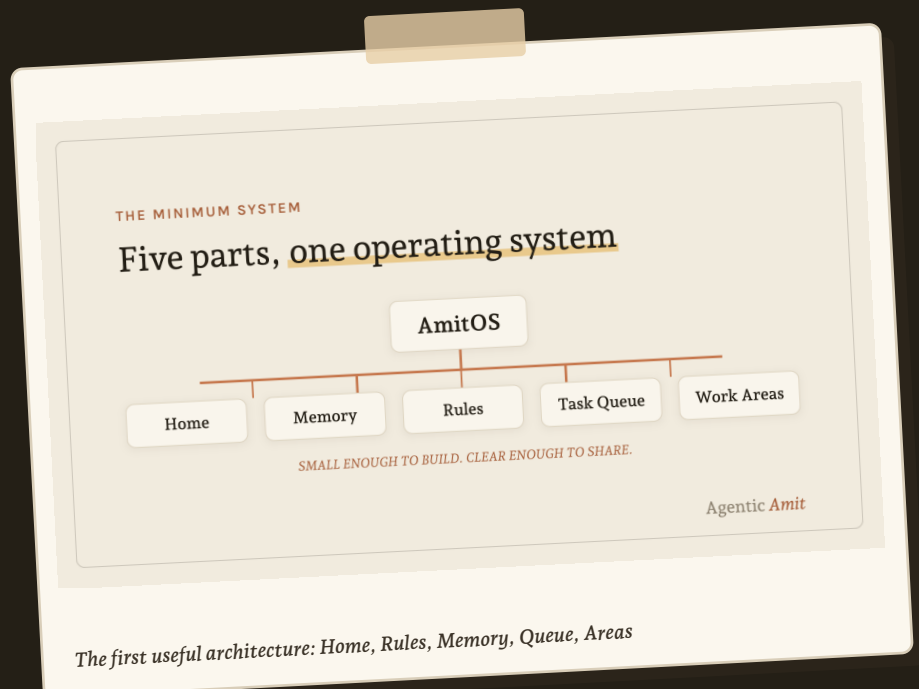


How I built AmitOS

One memory layer for every AI tool

A practical guide to shared memory, task state, and human approval in Notion.



INCLUDES

Architecture / schema / bootstrap
dry run / failure modes

START HERE

The short version

AmitOS gives changing AI tools one durable place to read and update context.

THE PROBLEM

Powerful tools, no shared starting point

Notes existed, but work state, durable memory, and approval boundaries were split across conversations.

THE SYSTEM

One operating context in Notion

Every agent enters through the same Home, reads the same rules and memory, and updates the same Task Queue.

THE BOUNDARY

Autonomy stops before irreversible action

Research, drafting, and building can continue. Sending, publishing, signing, charging, or deleting waits at Needs You.

The model can change. The context survives.

What this guide is based on

This is a documented personal system, not a generic productivity template.

The build was checked against the canonical AmitOS records on July 15, 2026.

JULY 6

Home records the move from an Obsidian vault to Notion as the source of truth.

SCRIPT 18

Build Your Own AmitOS contains the original mechanism, build order, and teaching arc.

LIVE PROTOCOL

Agent Operating Protocol defines the six task states and the Needs You approval gate.

LIVE SCHEMA

Task Queue confirms the minimum properties used to route, track, and hand off work.

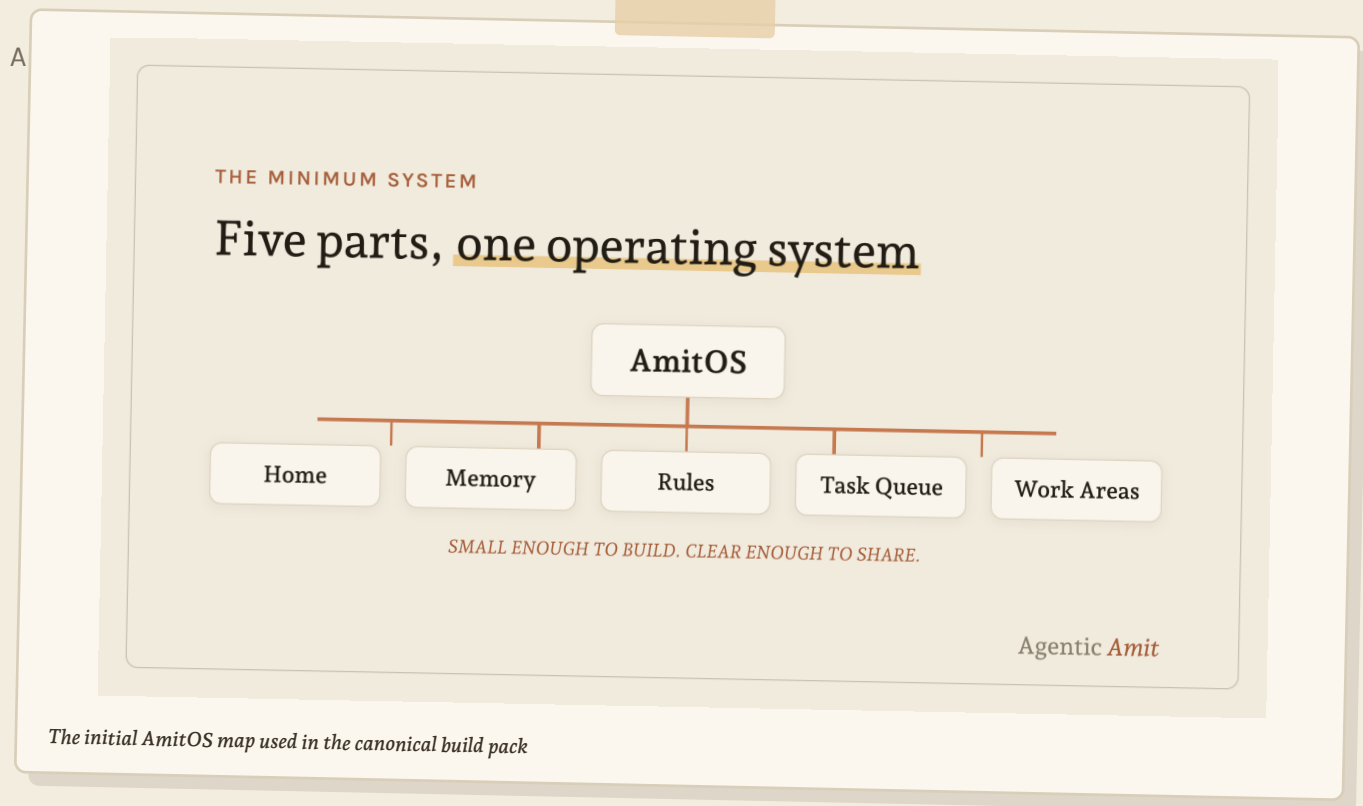
LIVE MEMORY

Memory defines durable facts and pointers; Daily Notes preserve chronological history.

BOUNDARY

No invented performance numbers, time savings, or adoption claims are included.

The five-part architecture

**01 Home**

Routes the agent into the system.

02 Operating Rules

Defines behaviour and autonomy.

03 Memory

Stores durable facts and pointers.

04 Task Queue

Makes work state visible.

05 Work Areas

Holds domain context and artifacts.

Design rule: add a new database only after a real task proves the current structure cannot hold the work.

Home routes. Rules constrain.

Separate navigation from behaviour so each stays easy to maintain.

HOME

Where am I?

- What AmitOS is
- Where rules live
- Which area owns the task
- Where the Task Queue lives

OPERATING RULES

How do I work?

- Writing and formatting
- Evidence and attribution
- Filing and handoff
- Autonomy boundary

Bootstrap sequence

1. Read Home

2. Read Rules

3. Read Memory

4. Open Work Area

5. Find or create task

If the agent cannot find the right context in five reads, improve the router before adding more memory.

Memory is not a transcript archive

Store information according to how it will be reused.

MEMORY

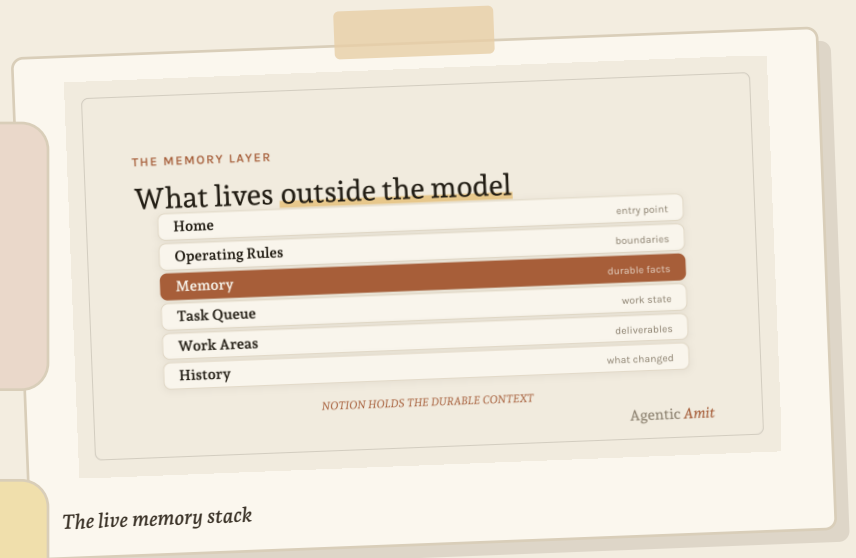
Durable facts and pointers

DAILY NOTES

History and decisions

WORK AREAS

Artifacts and domain context



Write to Memory when the fact will change future decisions across tasks. Write to Daily Notes when the value is chronological evidence. Put the artifact itself in the relevant Work Area.

DO NOT STORE

full chat logs / duplicate drafts / facts with no future use

A short memory page is a feature. It is easier for every tool to read before work begins.

The minimum Task Queue schema

Enough structure to route work, preserve state, and make a clean handoff

PROPERTY	TYPE	PURPOSE	EXAMPLE
Task	Title	Name the outcome	Publish field guide
Status	Select	Show work state	In Progress
Area	Relation / Select	Route the task	Agentic Amit
Type	Select	Identify work kind	Resource
Due	Date	Show timing	2026-07-15
Deliverable Link	URL	Point to artifact	Public PDF
Needs From You	Text	Exact human ask	Approve caption
Created	Created time	Preserve intake	Automatic
Updated	Last edited	Show recency	Automatic

The most useful field is Needs From You. It turns a vague blocker into a specific human decision.

Download the CSV schema from the starter pack.

One task. Six visible states.

QUEUED

Has the work been captured?

TRIAGED

Is the outcome and input clear?

IN PROGRESS

Is substantive work happening?

NEEDS YOU

Is a human decision required?

APPROVED

Has the irreversible step been cleared?

DONE

Is the final artifact filed and linked?

Do not skip directly from In Progress to Done when the work includes an external action.

Make the human gate visible

The

hidden in a prompt



The agent may continue

- Research
- Analyse
- Draft
- Organise
- Build reversible artifacts

The agent must stop

- Send or publish
- Sign or accept
- Pay, charge, or buy
- Delete important records
- Make another irreversible change

A useful Needs From You note says what decision is required, what will happen after approval, and where the artifact is linked.

Different tools, one operating context

CAPABILITIES

Different tools, the same system

	Home	Memory	Rules	Queue
Claude	✓	✓	✓	✓
ChatGPT	✓	✓	✓	✓
Codex	✓	✓	✓	✓

Each tool reads the same Home, Memory, Rules, and Queue

CLAUDE

Start the task or explore the problem

CHATGPT

Shape the draft or reason through the decision

CODEx

Build, verify, and file the artifact

Give every agent the same entry ritual

Use the full text file from the starter pack. This is the compact version.

You are working inside AmitOS, a Notion-based operating system.

Before acting:

1. Read Home.
2. Read Operating Rules.
3. Read Memory.
4. Open the relevant Work Area.
5. Find or create the Task Queue record.

You may research, analyse, draft, organise, and build reversible artifacts.

Move the task to Needs You before sending, publishing, signing, charging, paying, deleting, or making another irreversible external change.

At the end, update status, link the current deliverable, record the next action, and write only durable facts or pointers to Memory.

The model can change. The context survives.

Run one real task end to end

The dry run is more valuable than another hour of database design.

1

Capture

Create a rough task from your phone and set it to Queued.

2

Enter

Ask an agent to read Home, Rules, Memory, and the relevant Work Area.

3

Build

Have it create a reversible deliverable and link the current version.

4

Gate

Require it to stop at Needs You before the external action.

5

Approve

Give a clear decision and move the task to Approved.

6

File

Complete the action, link the final artifact, and move the task to Done.

PASS CONDITION

The task remains understandable when a different tool picks it up without a fresh briefing.

Four failure modes to catch early

Most problems come from unclear ownership, excessive memory, or an invisible gate.

01 The second-brain trap

SYMPTOM You build many databases before one task has completed the lifecycle.

FIX Delete structure until the dry run is obvious.

02 Memory becomes a landfill

SYMPTOM Transcripts and duplicate drafts bury the facts that affect future work.

FIX Keep facts and pointers. Move history elsewhere.

03 The queue is optional

SYMPTOM Agents create artifacts without updating state or linking the current version.

FIX Make queue updates part of the bootstrap and definition of done.

04 The gate is only prose

SYMPTOM The agent sees the warning but still has no visible state for human approval.

FIX Use Needs You and require the exact decision in the task.

Build the smallest version that can survive a handoff

- ✓ Home explains the system and routes work.
- ✓ Operating Rules define behaviour and autonomy.
- ✓ Memory contains durable facts and pointers only.
- ✓ Task Queue has the six exact statuses.
- ✓ Needs From You states one concrete decision.
- ✓ A different tool can resume the task.
- ✓ The final artifact is linked before Done.

DOWNLOADS

agenticamit.com/resources

PDF / CSV schema / bootstrap instruction / starter ZIP

Full implementation prompt

agenticamit.com/prompt/build-your-own-agentic-os-in-notion

AmitOS is a personal operating system built by Amit Maraj. It is not an official Notion, Anthropic, or OpenAI framework.