

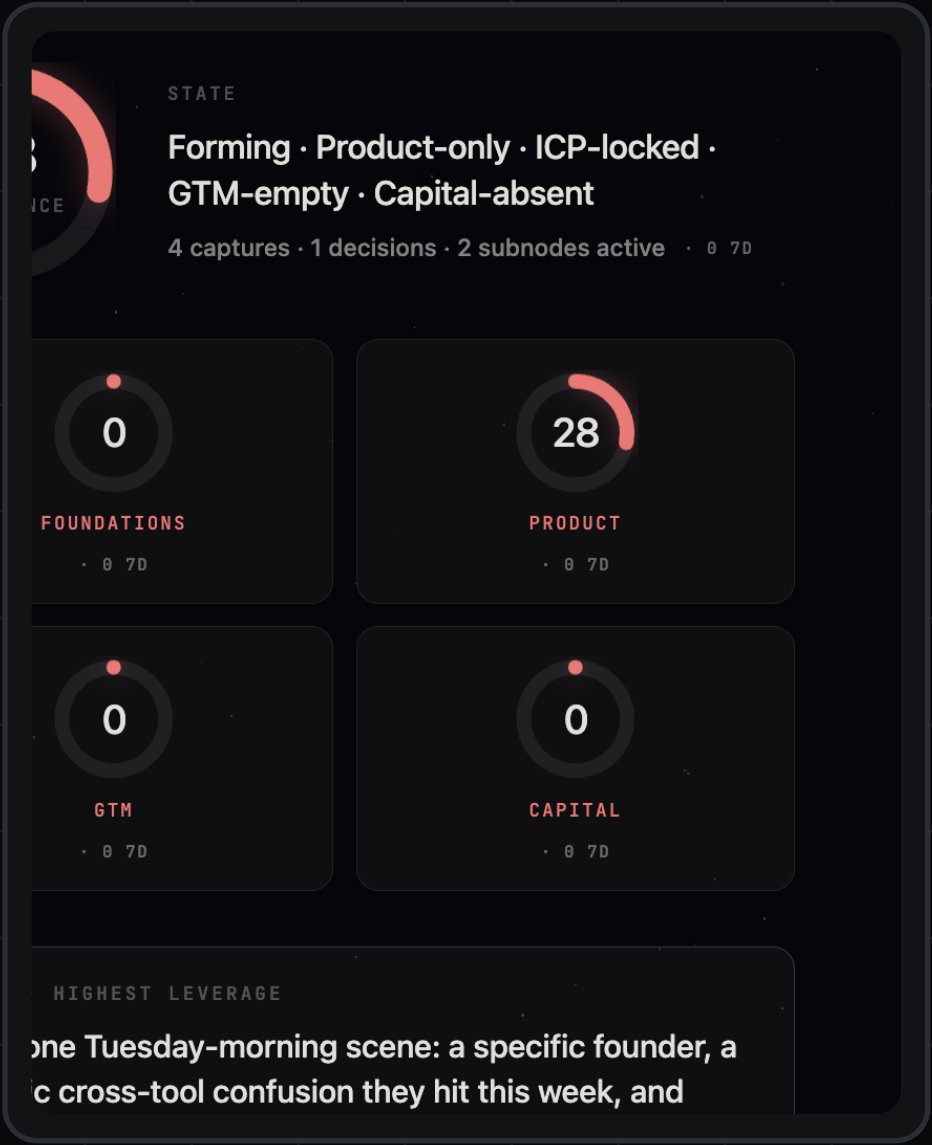
SELECTED BUILD / FOUNDER-LED SYSTEM

FOAAP

A venture operating system
that remembers.

Product architecture, state, memory, reasoning, and one next action.

PAULO VILLALOBOS
Founder + AI Product Engineer



01 / PROBLEM

Founder context does not live in one place.

Chats forget. Documents age. Dashboards display data without deciding what matters next.

01

FRAGMENTATION

Context is scattered across tools and conversations.

02

STATE LOSS

Each new session starts by rebuilding what was already known.

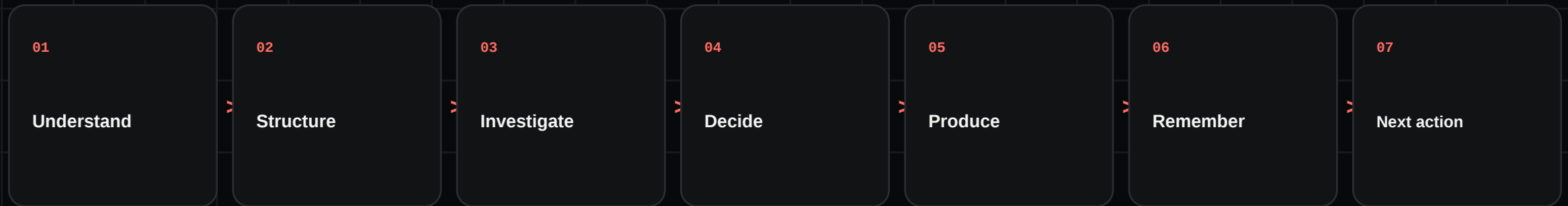
03

ACTION NOISE

Many options are visible; the highest-leverage move is not.

02 / OPERATING LOOP

The interface follows the work.



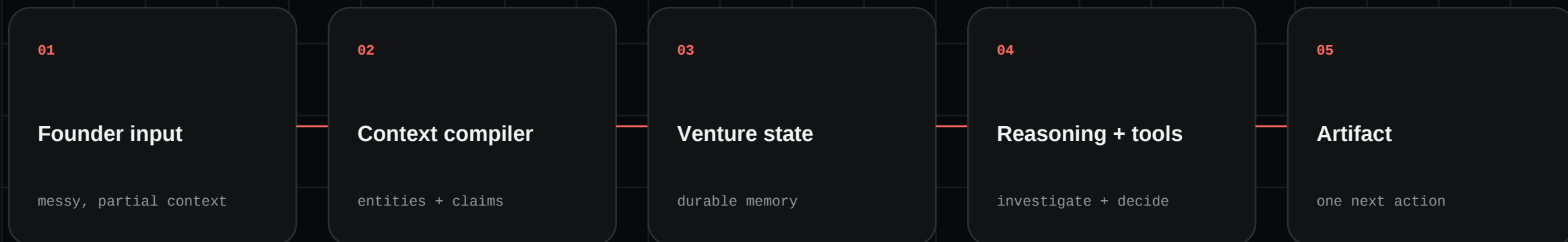
SYSTEM INVARIANT

Every meaningful input mutates state. Memory changes the next interaction.

Artifacts are consequences of decisions - not disconnected outputs.

03 / ARCHITECTURE

A simplified production path.



React + Vite

Supabase + Postgres

OpenAI + Anthropic

State-driven UI

● Built and deployed end-to-end by one founder

04 / PRODUCT EVIDENCE

Intake becomes structure.

The system pushes past a broad category and gathers the specific context it needs to act.

OBSERVED INTERACTION

Input: "I wanna build software for SMEs"

Response: treat that as a category, then narrow it into a venture with a defined user, problem, and operating context.

I'm intake. My job is to build the context of your venture — what it is, who it's for, how it goes to market, how it stays alive. I ask scoped questions across four domains: Foundations, Product, Go-to-Market, Capital. When context is built, we work inside those domains together and produce real artifacts.

ONE QUESTION

Let's start. What are you building?



i wanna build softwarte for SMEs

"Software for SMEs" is a category, not a venture.

Even Bob Odenkirk in the world could say

05 / PRODUCT EVIDENCE

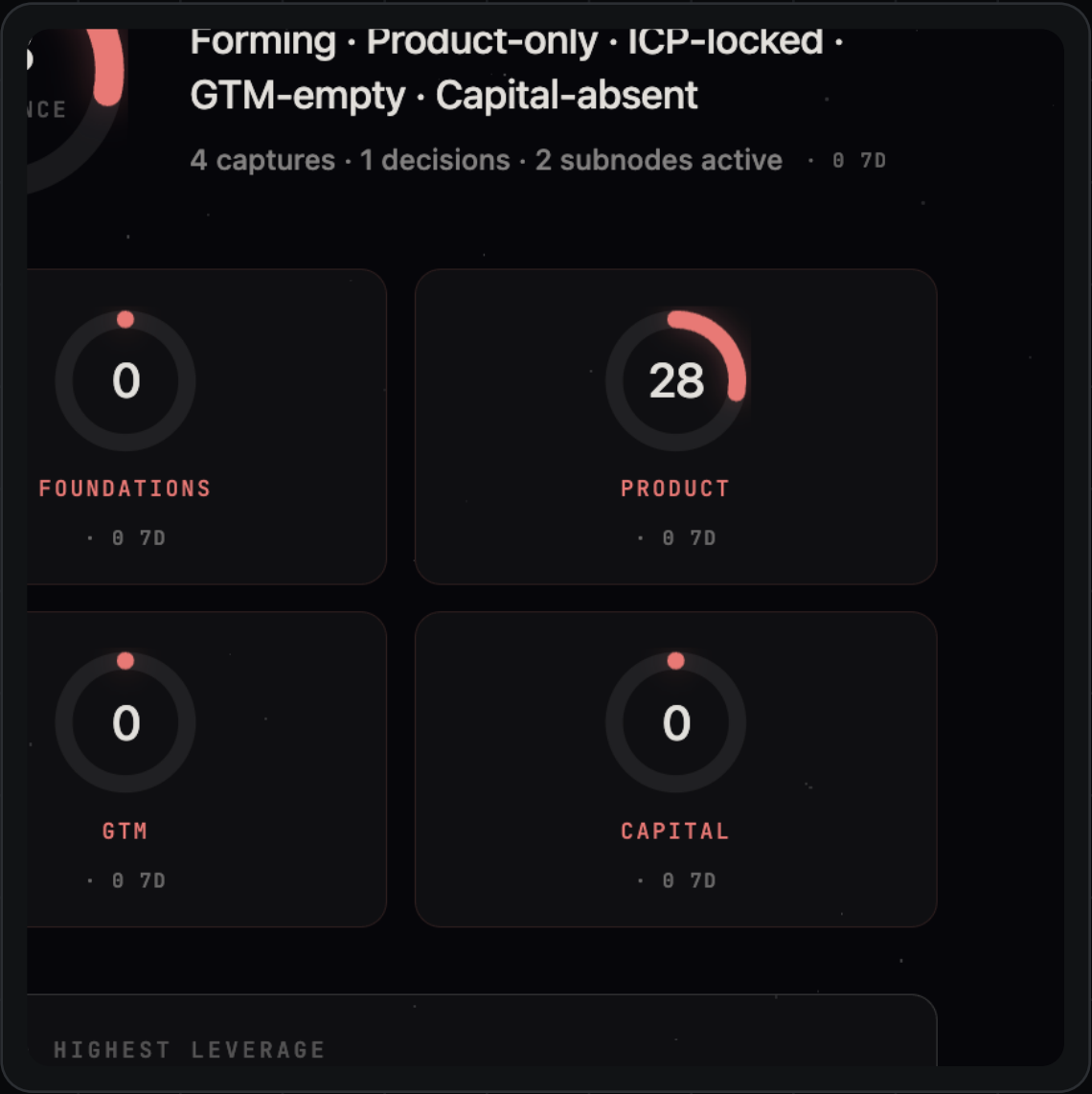
Memory changes the next move.

FOAAP exposes coherence, contradictions, state, and the highest-leverage action instead of another passive dashboard.

28 COHERENCE

02 CONTRADICTIONS

01 NEXT ACTION



What I would harden next.

A real case study should show where the system can fail - and the next engineering move.

01**MESSAGE LIFECYCLE**

Observed: an undefined atelier message ID.

Fix: typed state transitions and a regression test around optimistic messages.

02**MEMORY SYNC**

Observed: a failed save queued for retry.

Fix: idempotent retry queue, visible sync state, and offline-safe persistence.

03**OPERATIONAL PROOF**

Need: failures visible before users report them.

Fix: trace every state mutation, tool call, retry, and terminal outcome.

ROLE FIT

Give me the hard part.

I turn ambiguous product problems into working systems.

- 01 AI product MVPs
- 02 Stateful agent systems
- 03 Internal tools + automations
- 04 Focused web applications

CONTACT

paulo@foaap.app

Porto, Portugal / Remote

foaap.app

Founder + AI Product Engineer

