danic.ai-os / v1.0 — agentic core

REFERENCE ARCHITECTURE

AGENTIC WORKFLOWS BUILT TO RUN IN *PRODUCTION*.

SYSTEM LIVE
80+ agents · 68+ commands
40+ skills · hooks · ai-brain

01 Start with the workflow

7 QUESTIONS BEFORE ANY MODEL

1
Input
what triggers it?

2
Context
what does AI need?

3
Decision
classify, route, draft

4
Tool / Action
what system to touch?

5
Validation
how is it checked?

6
Human Approval
where is judgment?

7
Output

what does success look like?

*Don't start with the model. **Start with the workflow.***

02

danic.ai-os layered architecture

8 LAYERS · 1 GATEWAY · FEEDBACK LOOP

A

Trigger Layer

HOW WORK BEGINS

- hooks
- slash-commands
- user query
- file event
- webhook
- schedule

B

Context Layer

GATHER RIGHT INPUTS

- ai-brain
- skills (40+)
- org-layer
- memory
- vector store
- past sessions

C

Reasoning Layer

INTERPRET · PLAN · DRAFT

- orchestrator

D

Action Layer

TOOL GATEWAY · MCP

- github

- sub-agents (80+)
- classify
- plan / route
- summarize
- draft / extract

- slack
- postgres
- notion
- calendar
- supabase

E

Guardrails

CHECK BEFORE ACTION

- schema check
- policy check
- NFR framework
- confidence threshold
- source / citation
- test verification

F

Human Layer

APPROVAL WHERE RISK IS HIGH

- legal
- finance
- security
- external send
- customer impact
- compliance (GDPR)

G

Output Layer

WHAT THE WORKFLOW PRODUCES

- draft email
- PR / code change
- updated ticket
- report

H

Feedback Layer

CLOSE THE LOOP

- logs
- telemetry
- user feedback
- evals

- alert
- dashboard update

- prompt updates
- memory updates

Architecture principle

- AI operates *inside* a workflow, never outside it.
- Every action requires context, validation, and accountability.
- Production AI = `reasoning + tools + controls + review`

03 What makes it production-ready

6 NON-NEGOTIABLES

01 Clear inputs

Define the trigger and required data. No ambiguity at the entry point.

02 Reliable context

Right docs, right history, right business rules — assembled deterministically.

03 Action boundaries

Limit what tools and systems the agent can touch. Least privilege by default.

04

Validation

Verify outputs before anything happens. Schema, policy, source, test.

05

Human oversight

Keep humans in the loop for high-risk, customer-facing or irreversible decisions.

06

Continuous improvement

Learn from every run. Evals, telemetry, prompt + memory updates.