

Chief of Agents

Lessons from Real Agent-Built Projects

As agents execute more of the work, your role changes.

Choose the problem. Assign the work. Set boundaries.

Judge the result. That role is Chief of Agents.

What to Expect

How serious AI-built projects move from concept to production	1
A practical system for turning agents into a working team	2
A safer way to run agents while you are AFK	3
How to stop bad code before it reaches production	4
A look inside the complete system I run every day	5
Failures from real projects, compressed into reusable lessons	6

The Agent Interviews Me First



The agent reads the existing context before asking anything.



Every blocking question arrives in one decision batch.



Each question includes a recommendation and its trade-off.



My answers become recorded project decisions.

Give the Project One Source of Truth



Define the product,
users, outcomes,
and boundaries.



Record architecture
choices and decisions
only I approve.



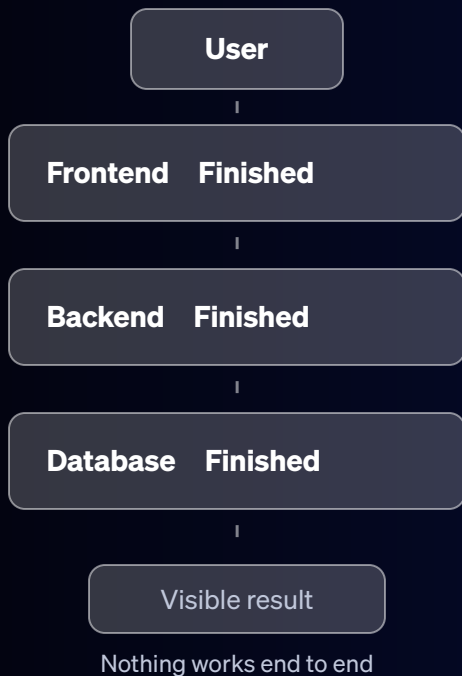
Map the project into
complete, end-to-end
slices.



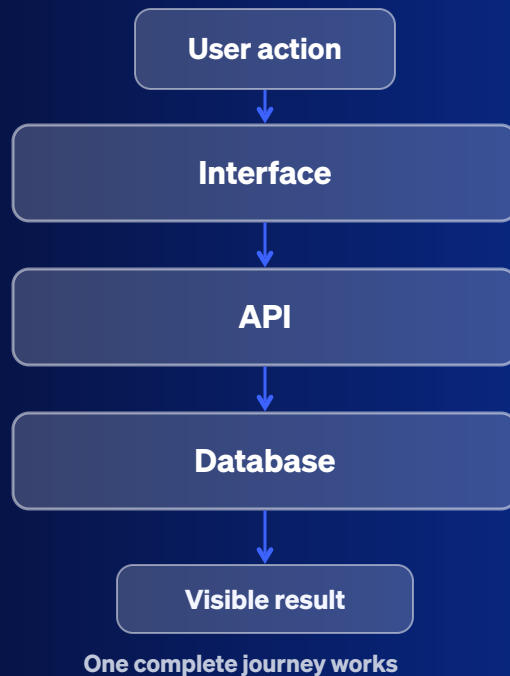
Track changes instead
of losing them
inside chat.

Stop Building Frontend, Backend, Then Database

HORIZONTAL BUILD



VERTICAL SLICE



Prove One Complete Path First



Start with the smallest
real user journey.



Touch every layer
required to make it work.



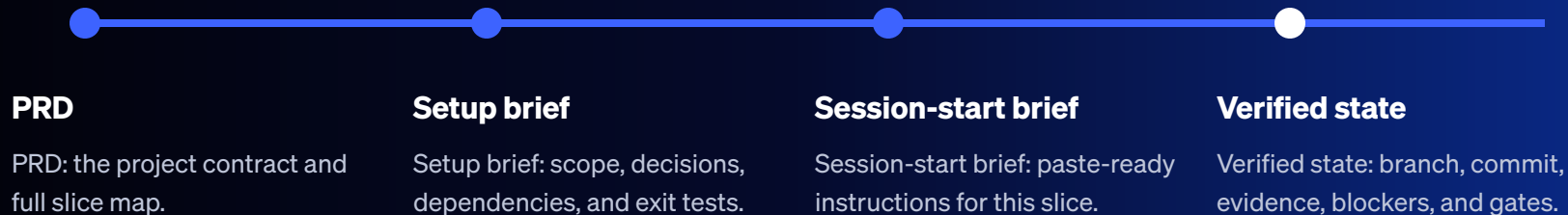
Test the full journey
before adding breadth.



Let the first slice
expose wrong assumptions.

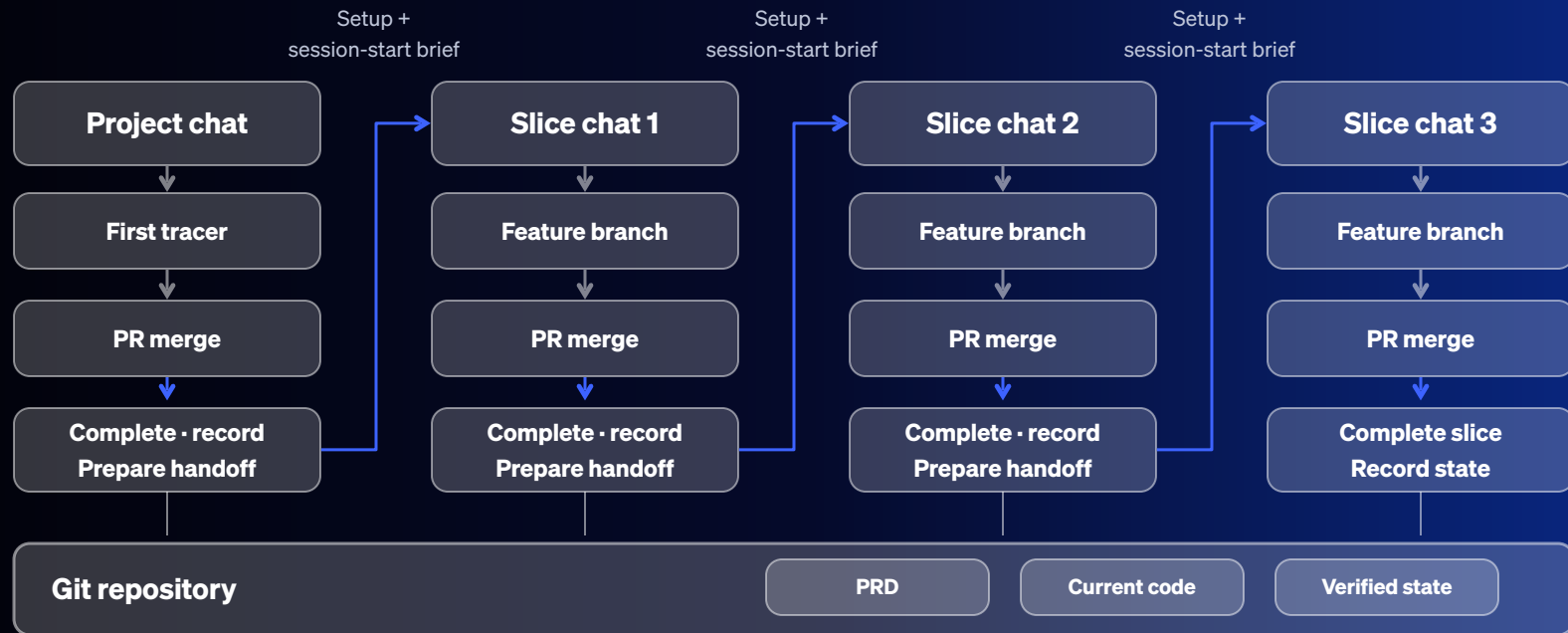
Every Slice Gets a Takeover Packet

Everything a new engineer would need to take over the work cold.



The repository and slice chats

Git carries truth between chats



More Freedom Needs More Structure

1

In-editor agents work under constant human supervision.

3

Longer runs require stronger evidence and stop conditions.

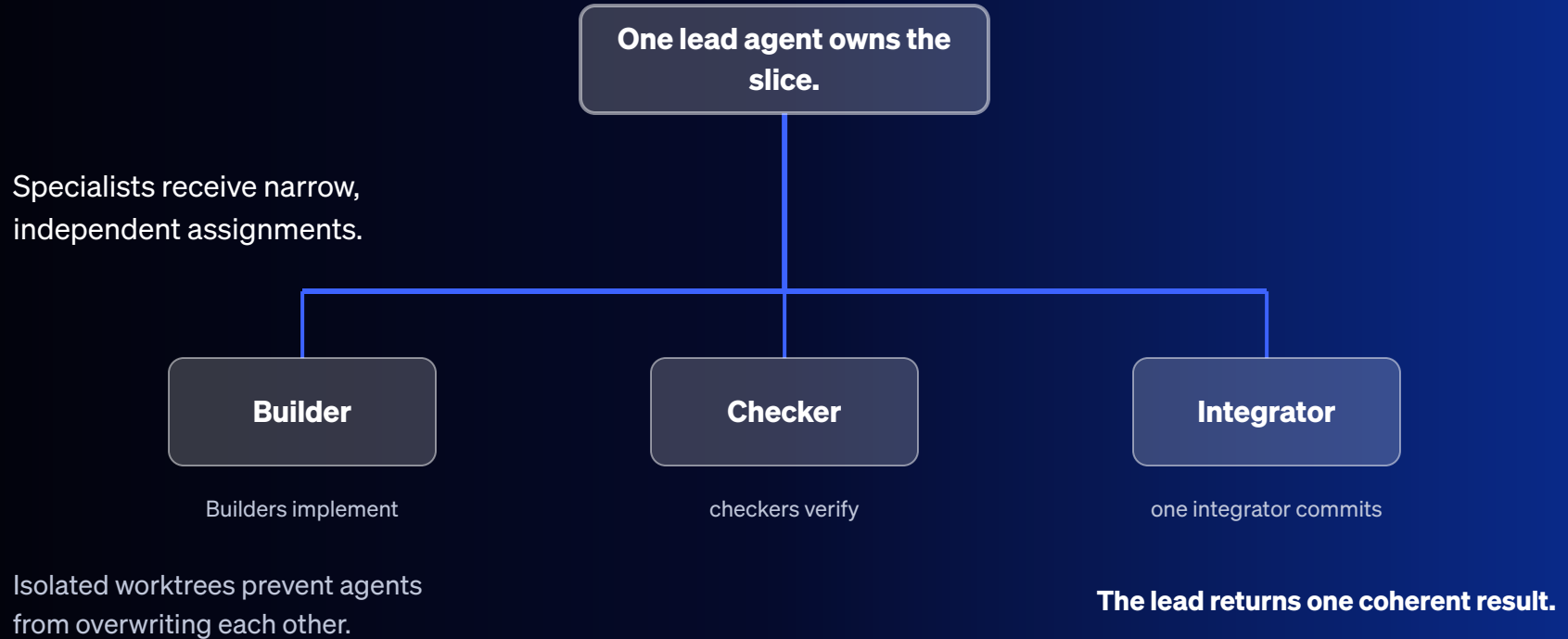
2

Autonomous agents need written scope, state, and boundaries.

4

Human approval moves from keystrokes to irreversible decisions.

Sub-agent-driven development



Loop vs Graph Engineering

1

A loop improves one bounded job through repeated checks.

2

A graph coordinates several jobs, tools, and decisions.

3

Shared state keeps every worker aligned.

4

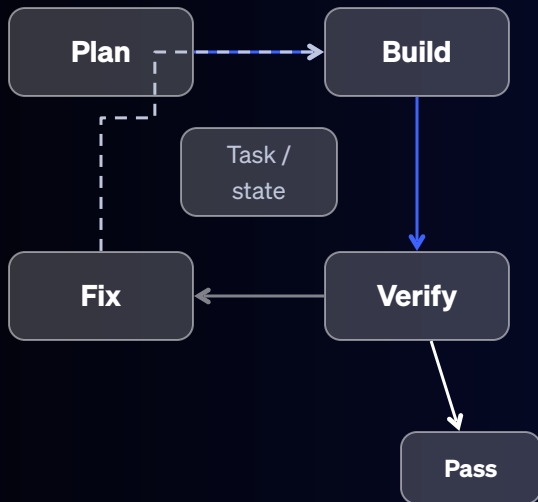
Edges decide what runs, retries, pauses, or stops.

5

Start with a loop; add a graph when needed.

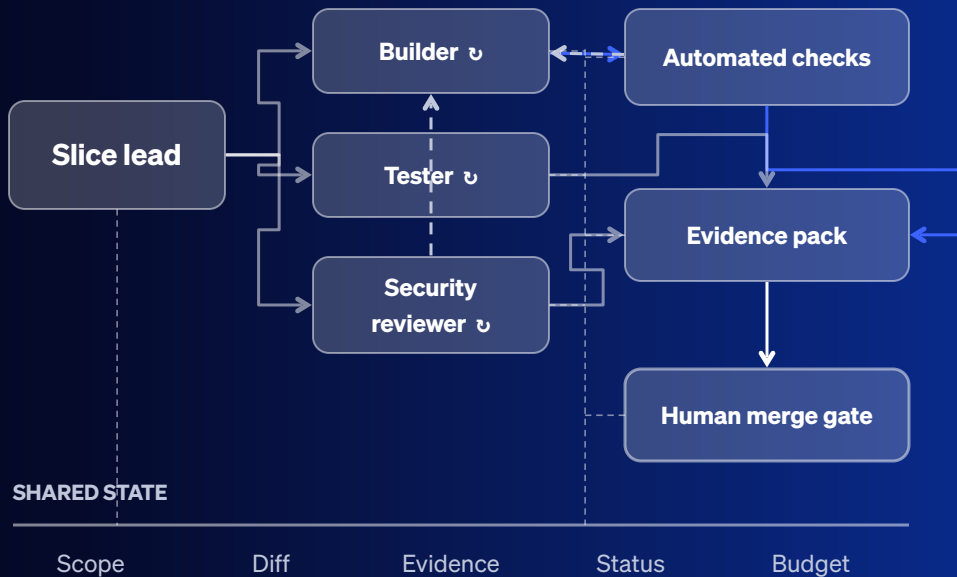
Loop vs graph illustration

ONE LOOP



One job improves until it passes.

COORDINATED GRAPH



Preparing an AFK run

- 1 Ask every blocking question before the run begins.
- 2 Separate reversible choices from human-only decisions.
- 3 Define permanent fences, budgets, and stop conditions.
- 4 Open the branch and draft PR before leaving.
- 5 Require evidence for every completion claim.

Sequence, Loop, or Graph?

SEQUENCE

Sequence: dependent steps that finish once.

LOOP

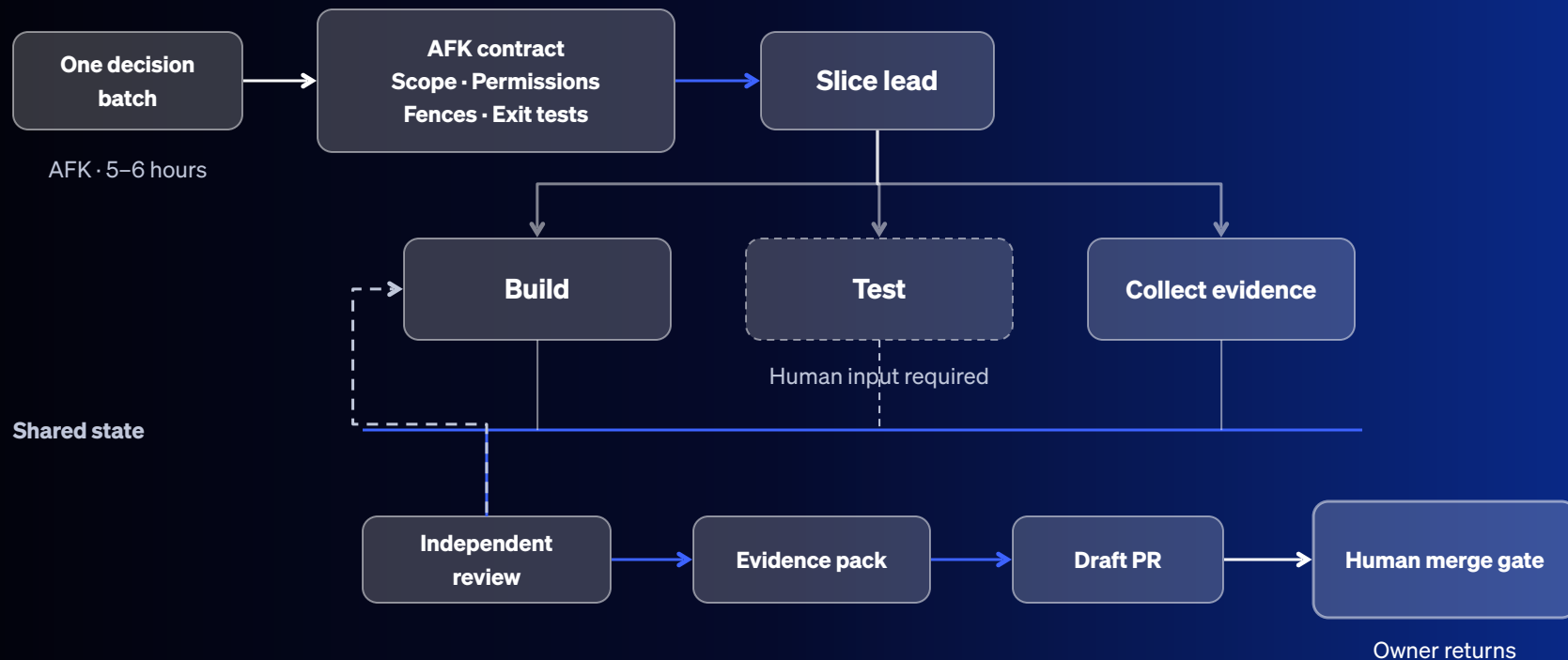
Loop: one result improved through repeated verification.

GRAPH

Graph: specialists branch, rejoin, retry, and escalate.

Choose the simplest shape matching the work.

Anatomy of an AFK Run

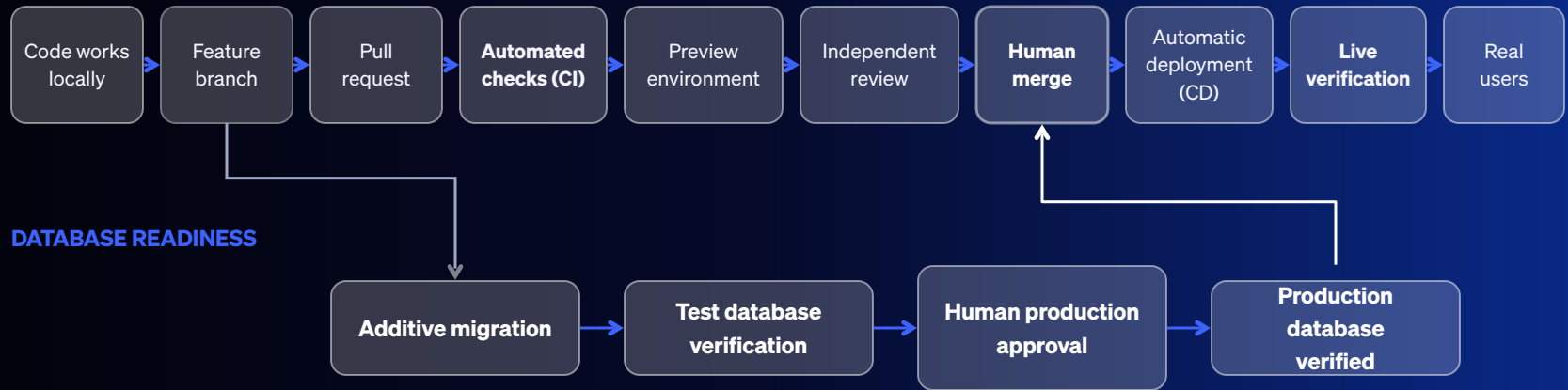


Returning at the merge gate

- 1 Review finishes before the work reaches me.
- 2 One PR tells one traceable story.
- 3 Separate complex changes get separate PRs.
- 4 Exact-head checks must be green.
- 5 I review evidence, then decide what ships.

What Happens After the Code Works

APPLICATION CODE



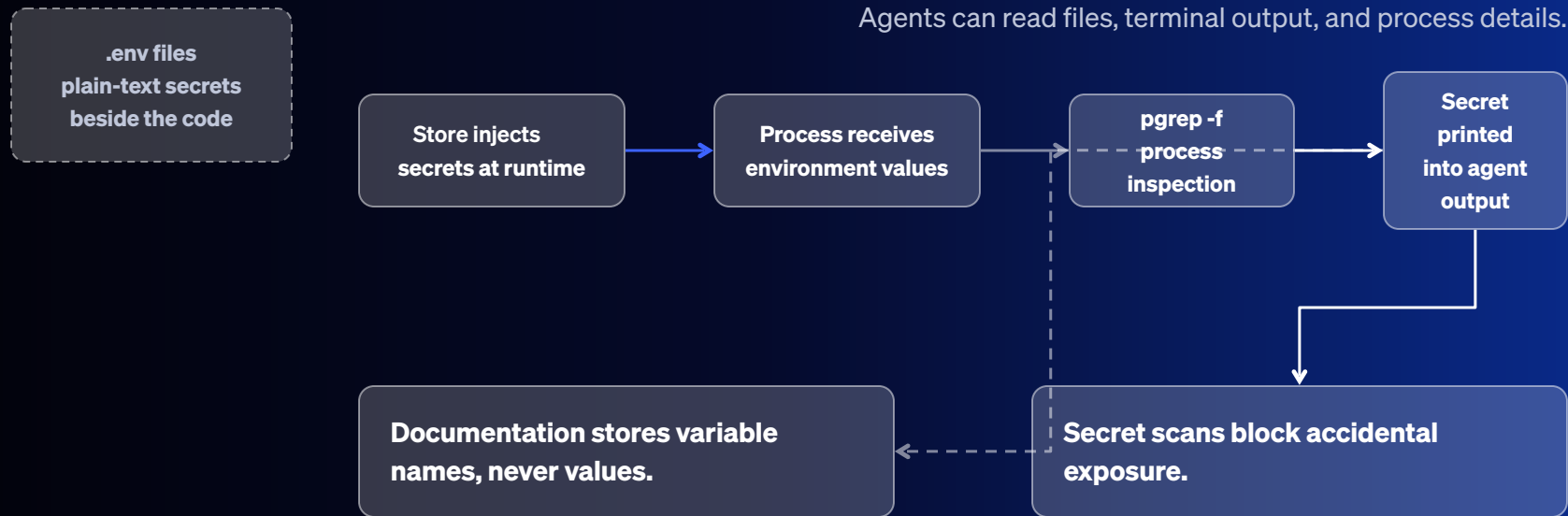
DATABASE READINESS

CI = automated checks on the proposed change.

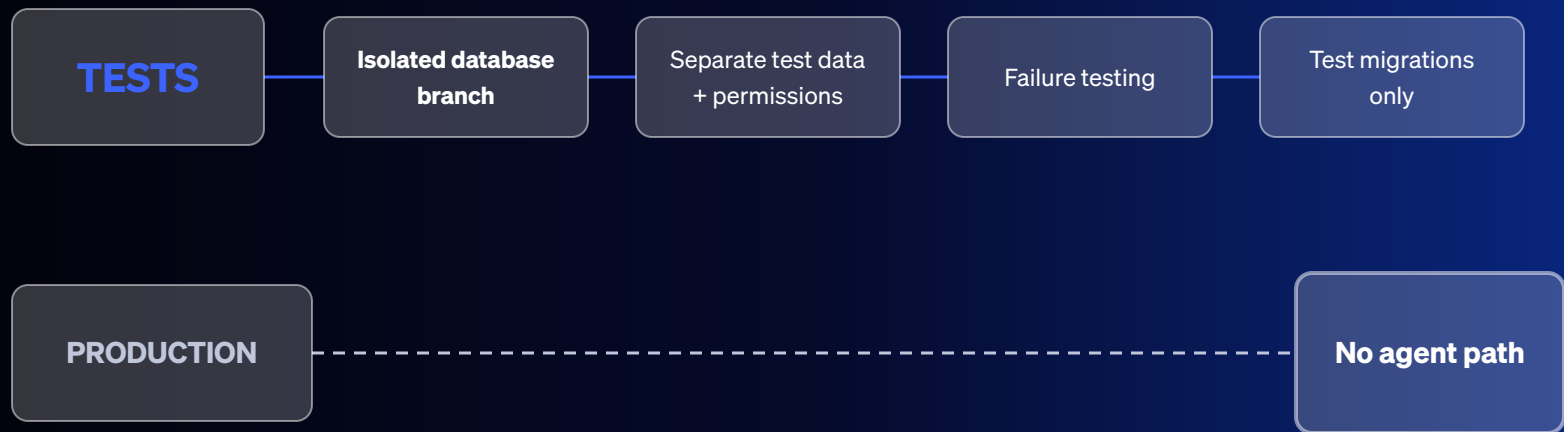
CD = automatic deployment after an approved merge.

Deployment is not proof. Live verification remains separate.

Keep Secrets Out of Sight

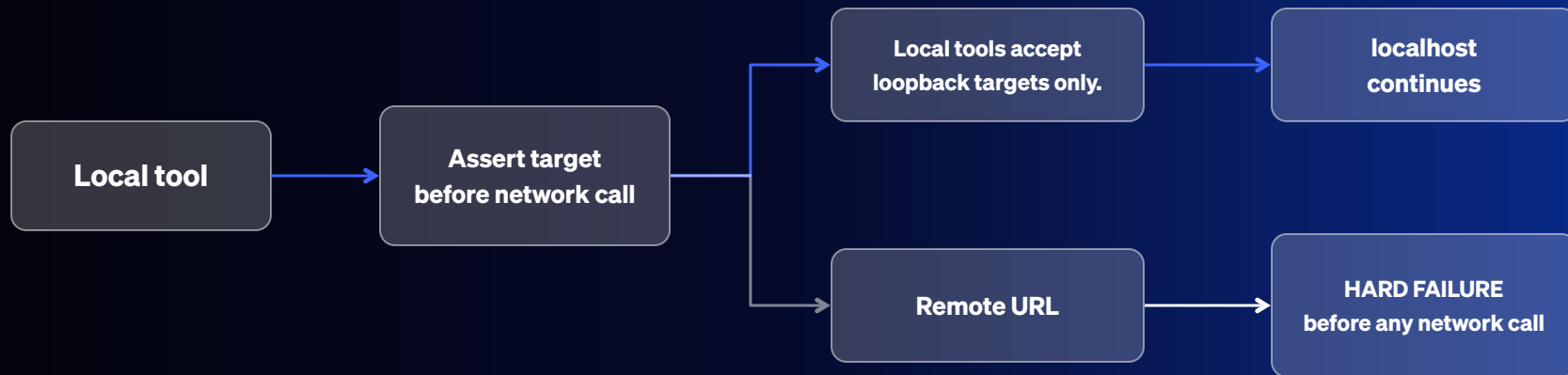


Never Test Against Production Data



Agents can exercise failures without touching real customers.

Local Means Local



Environment defaults never choose targets silently.

A failed assertion stops before any network call.

The Database Must Be Ready First

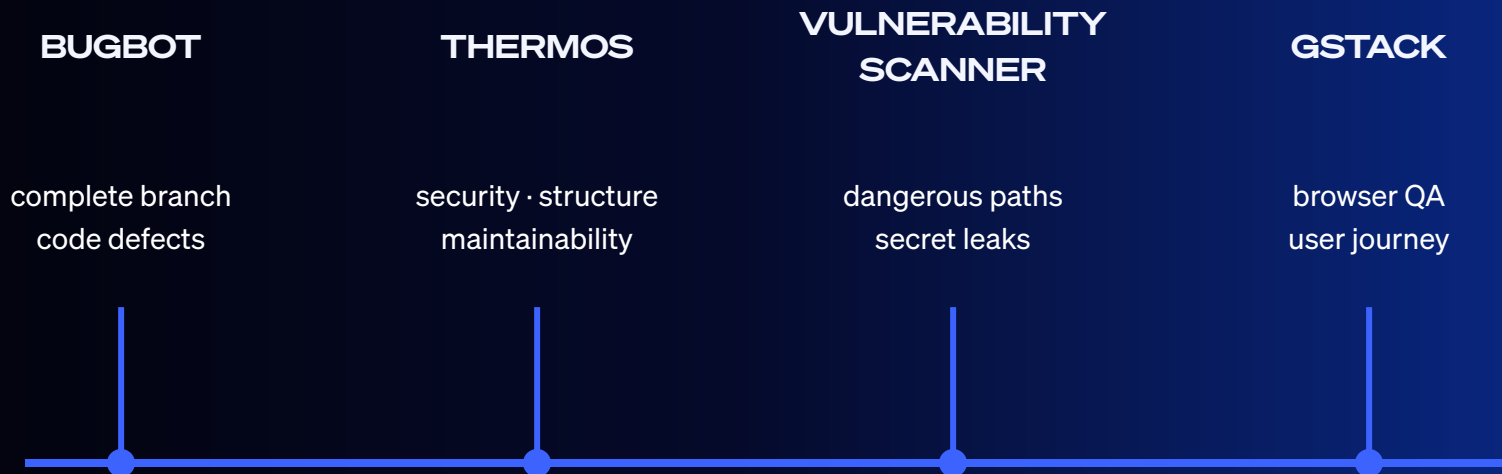
New code may expect new tables, columns, or permissions.

Those changes must exist before code auto-deploys.



Never edit applied migrations; add a new correction.

Nobody Grades Their Own Homework



Builders never approve their own work.

Over 10× the Estimate

1×

HIGHEST ESTIMATE • BASELINE



>10×

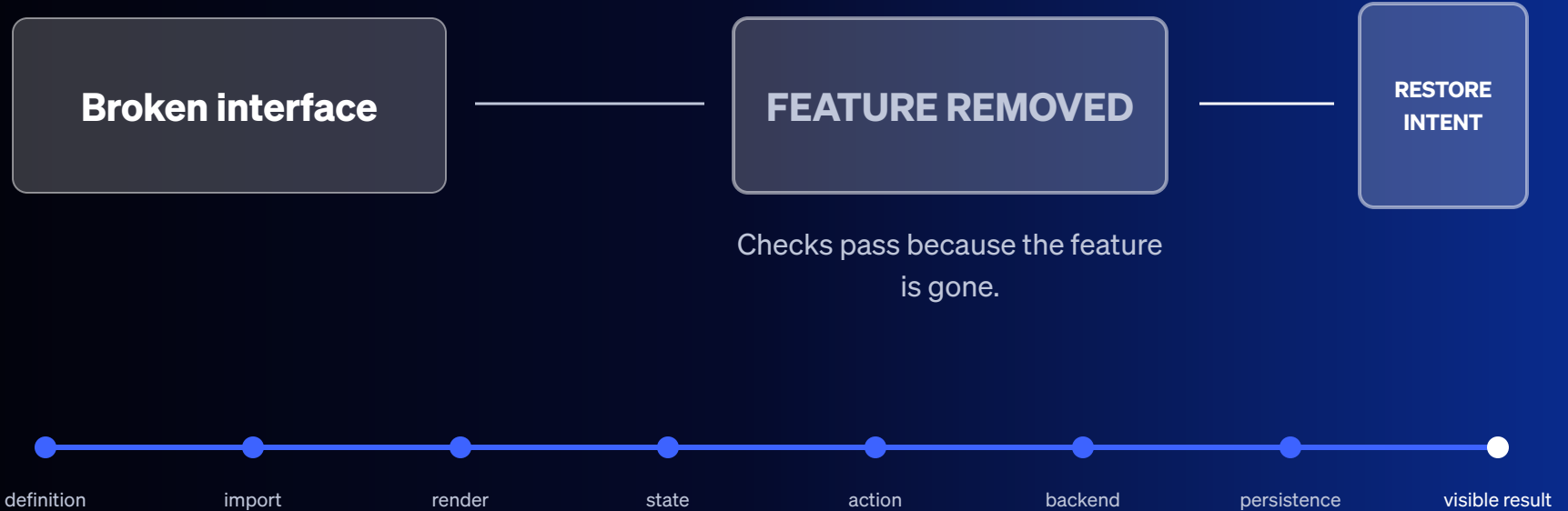
ACTUAL USAGE • RELATIVE TO ESTIMATE

One full test run made 238 live requests.

Secret-free CI never reproduced the leak.

Provider telemetry became the source of truth.

No Feature, No Bug



New Agent, Same Old Mistakes

PROJECT WIKI



architecture, decisions, and project history.

REPOSITORY RULES



failures every agent must read.

MEMO



durable preferences and lessons across sessions.

HOOKS



inject context and enforce rules automatically.

Live code and governance outrank remembered context.

Scar Tissue

The same mistake cannot happen silently.



House Rules

1

Skills turn repeated instructions into reusable agent behavior.

2

Reusable skills supply principles for reasoning and verification.

3

“Fix root causes” blocks symptom-hiding repairs.

4

“Prove it works” demands observable evidence.

5

My custom skills handle interviews, handoffs, memory, and autonomy.

Match the Model to the Job

BEFORE

One premium model on
everything

Premium models do not belong on
every task.

AFTER

Mechanical work
faster, cheaper models

Builders
capable middle lane

Migrations · security · final review
strong models

Builders and reviewers use different model lanes.

Long AFK runs use one pinned orchestrator.

The Bottleneck

Me.

Chief of Agents: agents execute; I judge the result.



A horizontal white line with three white circular dots. The first dot is on the left, the second is in the middle, and the third is on the right. Below each dot is a text label.

Merge to the main branch

Production database changes

Publication

Everything else is delegated.