

Neighbourhood HQ: your personal assistant cockpit

Connect notes, drafts and approval into a system you can explain and maintain.

Build

Check

Try independently

The situation and your goal

Situation

A fictional organizer wants a weekly plan. The assistant should retrieve approved notes, draft the plan, flag unsupported claims and save only the human-approved version.

Keep this artifact

An assistant workflow, test matrix, approved plan and a maintenance runbook.

Understand the mechanism

01

Composition needs contracts

Each component should have explicit input and output. A notes step returns evidence; a model step returns a draft; an approval step returns a decision. Ambiguous handoffs make failures hard to locate.

02

Approval is a state transition

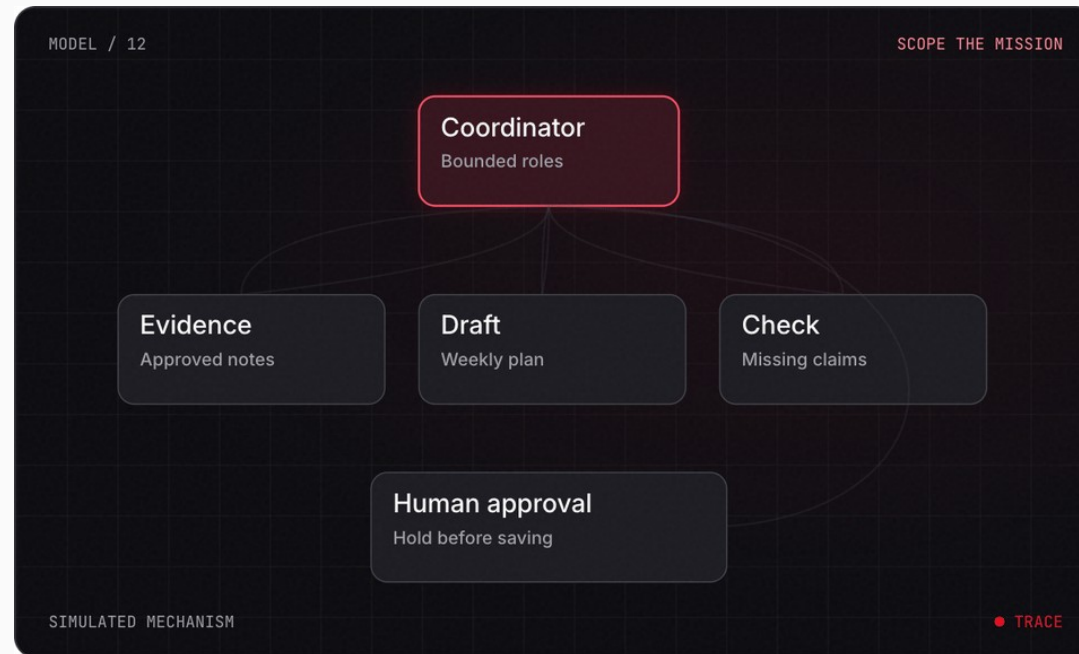
A request to approve is not approval. Keep pending, approved and rejected states separate; only the approved branch can save or send the selected plan.

03

Observability enables independence

Record a run ID, evidence IDs, validation result, approval outcome and final artifact. When something fails, diagnose the component instead of blindly changing the whole prompt.

Explore before you build



Map the smallest complete system

1

Use the evidence-garden notes and choose one output: a local weekly meetup plan. Draw contracts and approval states before choosing nodes.

2

Check: Every handoff has named input, output and owner.

3

Hint: Start with a fixed workflow; multi-agent orchestration is an optional extension.

Connect evidence and drafting

- 1 Use your approved notes as model context in a Basic LLM Chain. Require source IDs, plan actions and unanswered questions.
- 2 Check: The output is a draft with inspectable evidence.
- 3 Hint: This chain is not a tool-choosing agent just because it produces a plan.

Validate and request review

- 1 Validate output structure and source support. Present the draft for human review with approve, reject and edit outcomes.
- 2 Check: An unanswered request remains pending.
- 3 Hint: Do not turn a string saying approved into boolean permission from an untrusted message.

Gate the action

- 1 Use an explicit approval field from the authorized review step. Only true approval reaches the save action; false and missing return to review.
- 2 Check: Approved saves; rejected and pending do not.
- 3 Hint: A future send-to-bot action needs its own approved destination.

Run the failure matrix

- 1 Test missing evidence, stale capacity, invalid JSON, denied approval, missing approval and changed attendee count.
- 2 Check: The log locates the failure and shows what was prevented.
- 3 Hint: Keep each case and outcome beside your workflow export.

Create a maintenance runbook

1

Document how to refresh notes, rotate credentials, inspect logs, replay a failed case and disable actions. Export your completed project evidence from this learning app.

2

Check: A partner can explain and troubleshoot the system.

3

Hint: Your portfolio evidence is self-reported; demonstrate the actual build to a partner.

When it goes sideways

01

A plan is saved before review

Approval was implied instead of checked. Try: Require explicit authorized approval and test missing/false states.

02

You cannot tell why a claim is wrong

Evidence IDs and validation logs were dropped between steps. Try: Keep provenance and a run ID across handoffs.

03

More agents made debugging harder

Responsibilities overlap and tools are too broad. Try: Start with one workflow; add specialists only for independent work with a shared budget.

Review your evidence

1

Every handoff has an input/output contract.

2

Draft claims cite approved source IDs.

3

Validation has a visible failure route.

Review your evidence

1 Approved, rejected and pending states differ.

2 Six failure cases are recorded.

3 Runbook lets a partner replay a case.

An approval field is missing. What should happen?

- 1 Save because no one rejected it.
- 2 Keep the draft pending review.
- 3 Invent a reviewer.

When should you add another agent?

- 1 Whenever an animation looks better with more nodes.
- 2 When an independent task earns the coordination cost.
- 3 To hide unclear responsibilities.

Build with a clear contract

Design a small reviewed personal-assistant workflow using approved notes and a Basic LLM Chain. Handoffs: evidence with source IDs, structured draft, validation, authorized human approval, saved plan. Missing approval is pending; rejected approval never saves. Give a six-case failure matrix, a run log schema and a maintenance checklist. Optional specialists are advisory only, at most 5 agents total including coordinator, no recursive recruiting. Do not claim tools, credentials or actions are connected unless they actually are.

[Open copyable prompt](#)

Add bounded specialist review

1

Each role owns a narrow task.

2

Shared step/time budget is enforced.

3

Final evidence and approval remain visible.

Build independently, then teach it back

Your next challenge

Add a source reviewer and style reviewer as independent advisory steps. Keep a coordinator and at most four workers, five agents total. They cannot recruit more agents; actions stay behind human approval.

Explain your decisions

Show your artifact, one test that passed, one failure you repaired and one boundary the system respects.