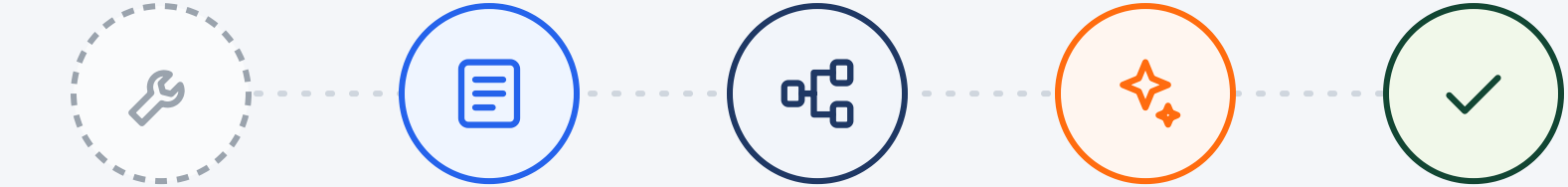
AGENTIC AI IN BUSINESS · WORKSHOP II



A page of your own, with n8n and Claude Code behind it

Morning 2 of 3 · first we talk, then you build your page, then a look at this afternoon

Frank Huettner · 16 November 2026

Today: first we talk, then you build

1–3

Laptops closed

- System architecture
- Setting up Claude Code
- Product architecture

4

Your tasks

laptops open, with the workbook: a page that asks your mailbox, a better search flow, your team's rules

Outcome: Your own page asks your mailbox

5

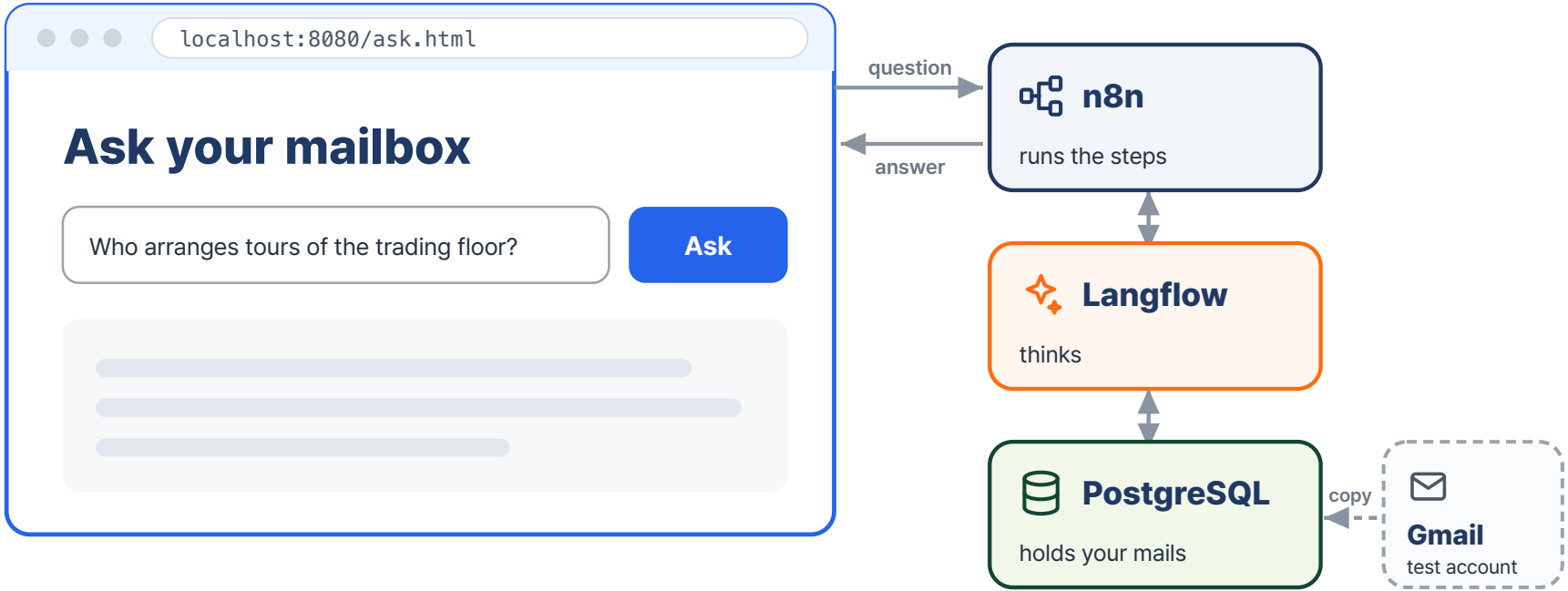
This afternoon

the voluntary tutorial

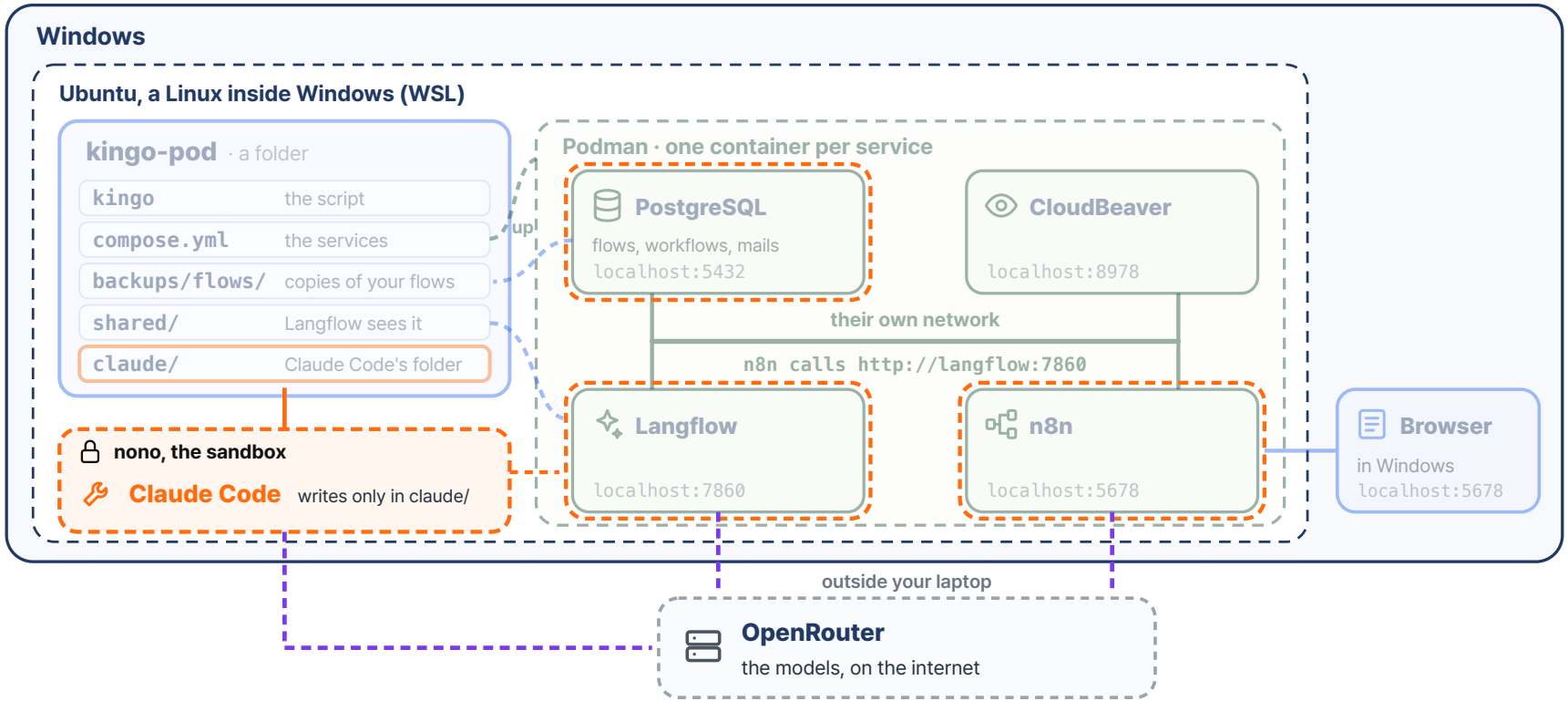
Outcome: n8n automation in your team's project

While we talk, nothing to type: the workbook has every step.

Today we build this: a web page to ask your mailbox



nono: Claude Code reaches your tools, not your laptop



Tell Claude Code your team's rules: one file, read at every start

Why: every teammate's Claude Code builds the same way, and nobody has to say it again in every session.

you send
a request
with the six principles, from the workbook

→

asks you, one principle at a time
Claude Code
writes your team's rules

→

you check them
claude/AGENTS.md
read at every start

Your names
for workflows, nodes, tables

Example: `Blue – Intake by Mail`

Your contracts
what goes in, what comes out

Example: Triage returns `action`

Where a person decides
which step waits for an OK

Example: a draft waits for your OK

What Claude Code remembers

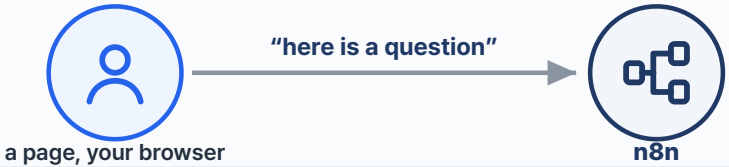
- it writes down what it learns, in `claude/memory/`
- you can read it, correct it, delete it

Another project: `./kingo claude --in ~/projects/thesis`

- Your team's `AGENTS.md` stays in kingo-pod: Claude Code does not read it when working in a different folder.
- Instead, it reads that folder's own `~/projects/thesis/CLAUDE.md` or `~/projects/thesis/AGENTS.md`; kingo does not put files there.

Webhook or HTTP Request: who calls whom?

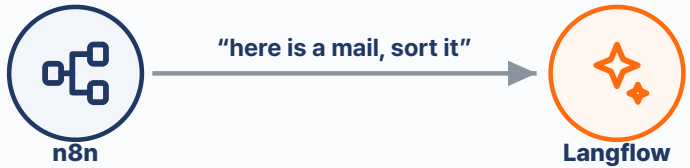
Webhook: someone calls n8n



Example: WS2.4. Your browser calls n8n for the page of drafts (Approvals Page), then with your OK, from a draft's form.

HTTP Request: n8n calls someone

Example: WS2.3 - Triage New Mail. n8n calls the Langflow flow Triage and reads its answer.

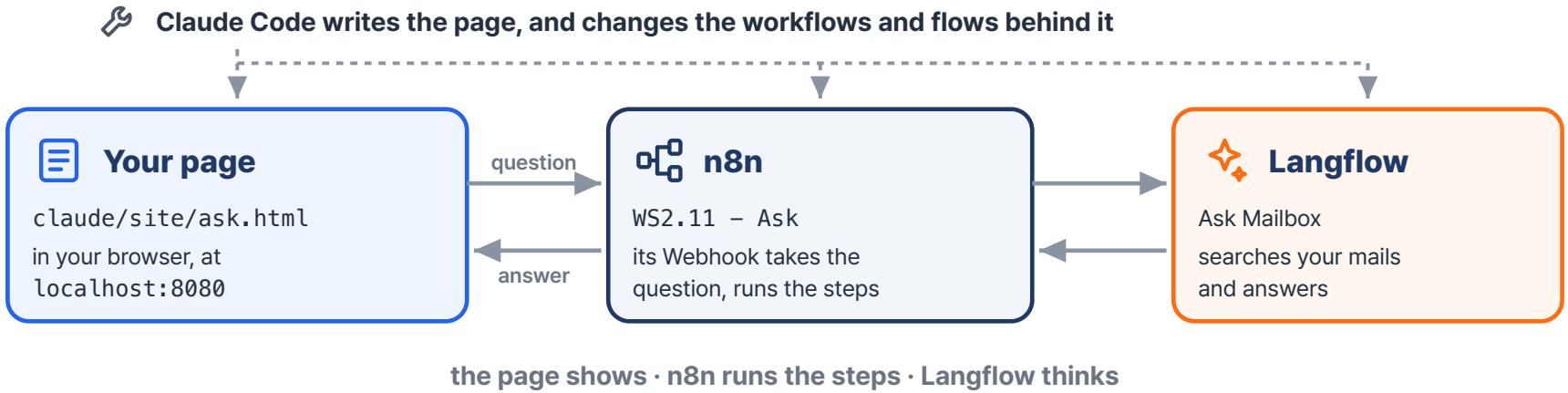


Both ways, an answer comes back. Only the direction differs: who starts the call.

Why a Webhook, and not HTTP?
A Webhook is HTTP: the address where n8n listens. HTTP Request is the node with which n8n calls out.

Why not the other way round?
Langflow's own webhook answers only “started”, never with the answer. And if Langflow called n8n, the model would start actions.

Claude Code can easily write webpages



- n8n works behind the page. The page knows only one address. The key for Langflow and all the steps stay in n8n.
- The page is one file in Claude Code's folder. A tiny web server hands it to your browser.
- Claude Code checks how the page looks (`look`), and can click through it (`clickthrough`).
- Claude Code can also change the flows: it changes a copy, you test it and switch it on.

Now you build: open your laptop and the workbook

Blocks 1–3

Build a web page that asks your mailbox, and improve the search flow

1. a simple webhook in n8n
2. Ask in n8n, and Claude Code writes your page
3. Claude Code improves the search flow

Outcome: your own page asks your mailbox

Block 4

now, or this afternoon

Write your team's rules, together with Claude Code

1. Claude Code asks you, one principle at a time
2. you answer and check
3. it writes them into `claude/AGENTS.md`

Outcome: Claude Code names your rules

Open the workbook: `ws2_s2_workbook.html` in your download. With short quizzes along the way.

Something does not work? Ask Claude Code what you see, or raise your hand.

This afternoon: we integrate n8n automation into your project



- The automation takes the routine off people: reading the request, looking things up, checking the rules.
- Your users work on the page. n8n works behind it.

Ideas for your project: what comes in, what waits, who decides

A · KSMC

- **comes in:** a mail asking for a quote
- **waits:** a draft quote, priced from your price tiers
- **decides:** a sales engineer; Approve sends the quote

`WS2.12 – KSMC – Quote Intake`, `… – Quote Queue`

B · Kgoda

- **comes in:** a trip request on your page
- **waits:** trips outside the travel policy
- **decides:** the traveller's manager

`WS2.12 – Kgoda – Trip Request`, `… – Trip Queue`

C · K-Motor

- **comes in:** a supplier's bid by mail
- **waits:** every bid, with the rules it breaks
- **decides:** the buyer

`WS2.12 – K–Motor – Bid Intake`, `… – Bid Queue`

Your team's page is in your download: `ws2_s2_team_….html`. You set up the templates, let Claude Code explain them, and change them for your team. The tutorial is voluntary.