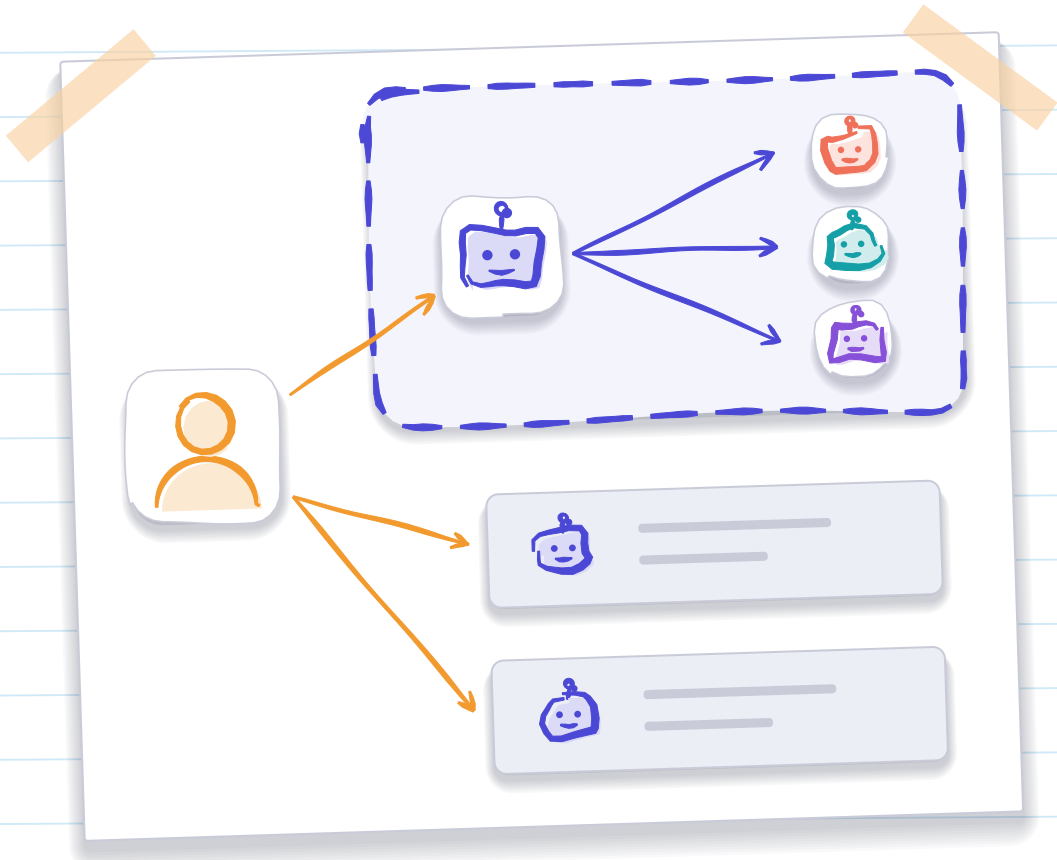
Prof. Aviv Zohar

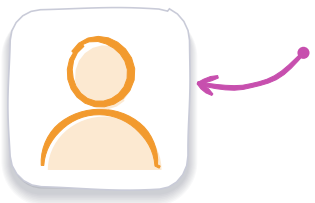
Semester A, 2026–7

Subagents and Parallel Sessions

One operator, several agents at once



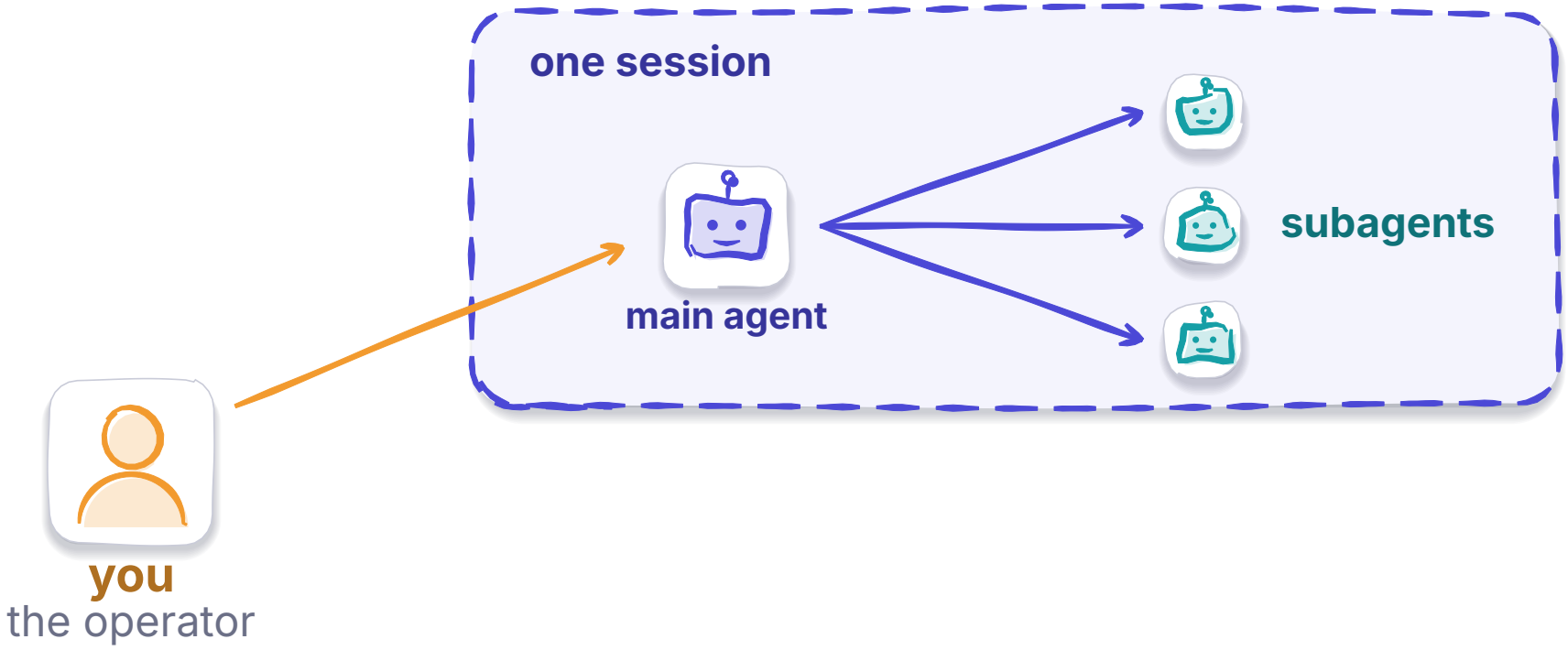
One operator, several agents



you
the operator

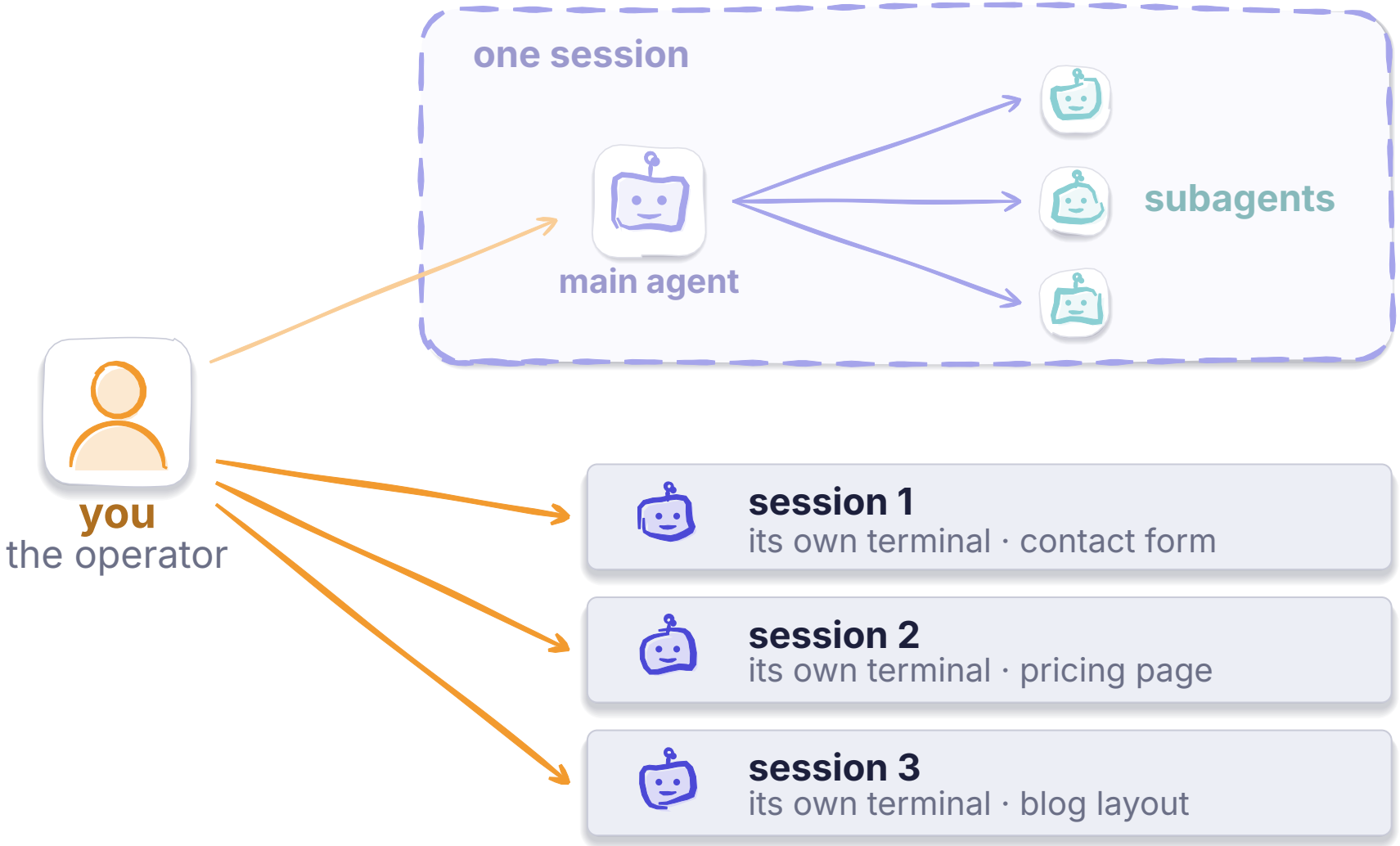
Every other participant is an agent: one you start, or one another agent starts.

One operator, several agents



way one
you talk to the main agent

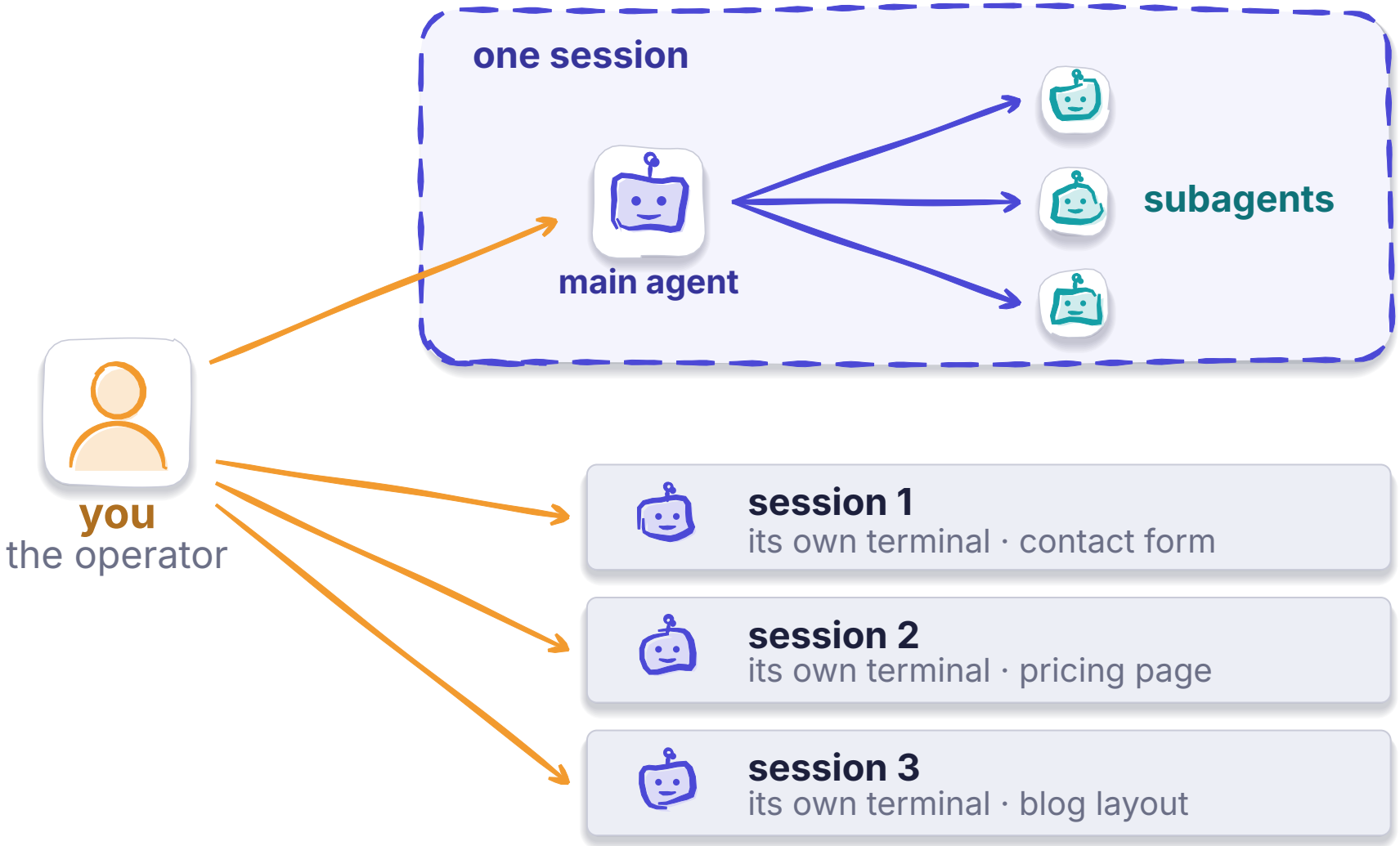
One operator, several agents



way one
you talk to the main agent

way two
you talk to each one

One operator, several agents



way one
you talk to the main agent

delegate a task

way two
you talk to each one

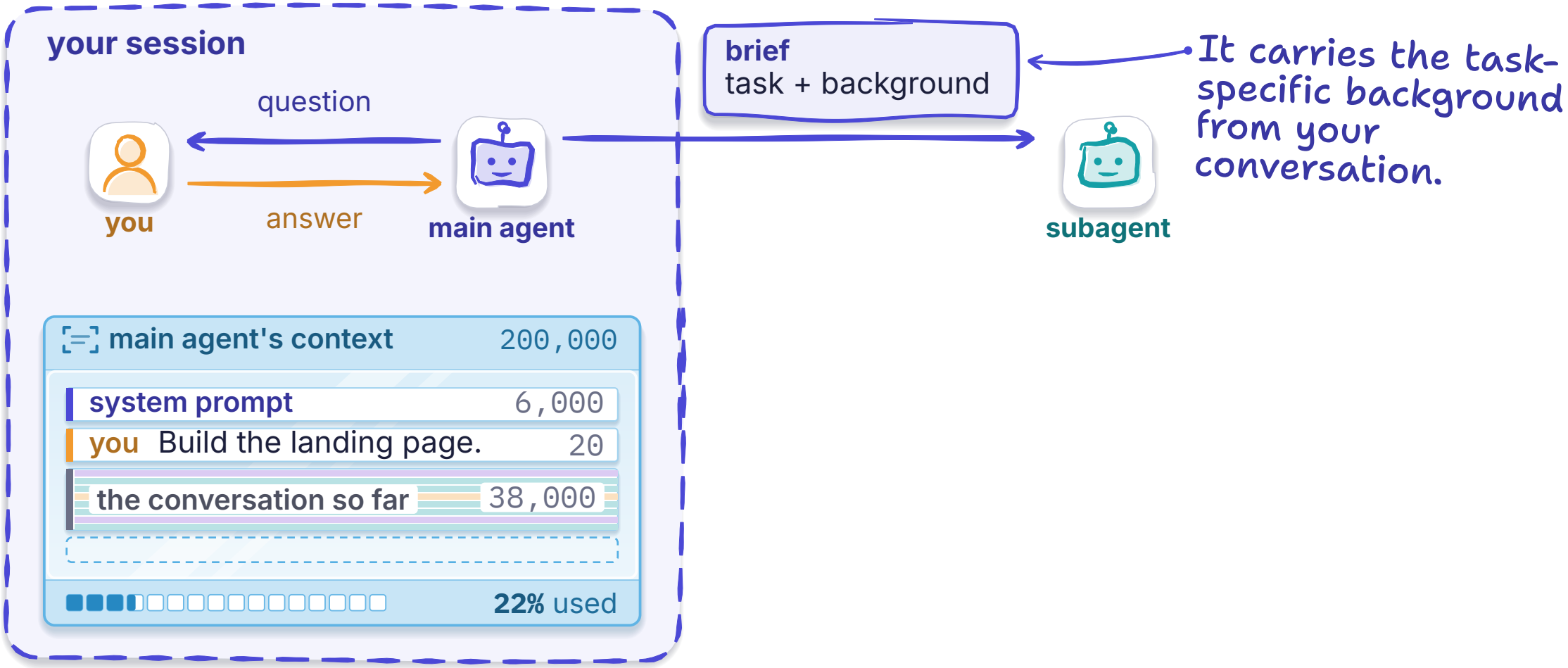
follow a task

The main agent hands a task to a subagent

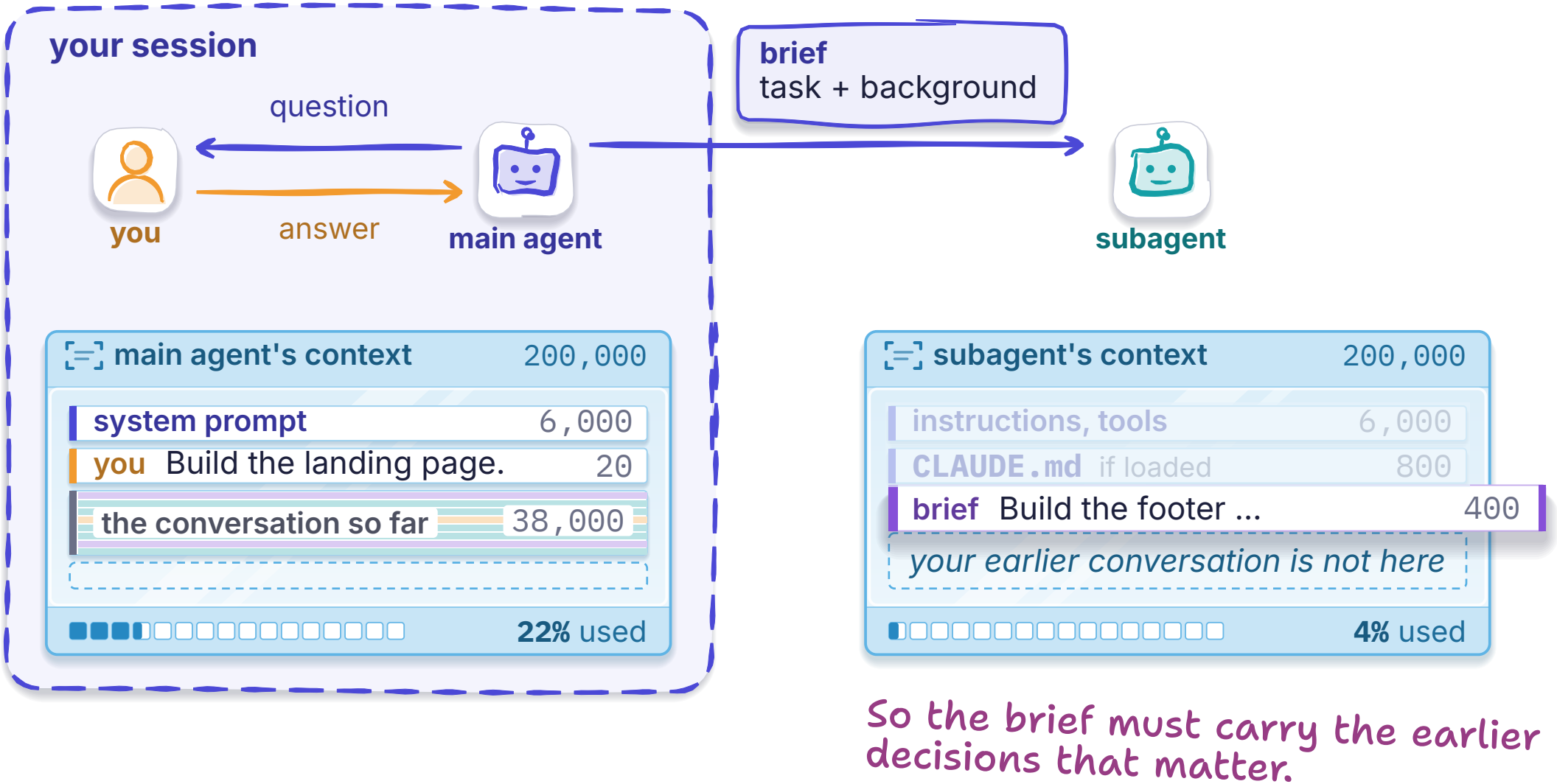


You talk to the main agent. When it has a question, it stops and asks you.

The main agent hands a task to a subagent

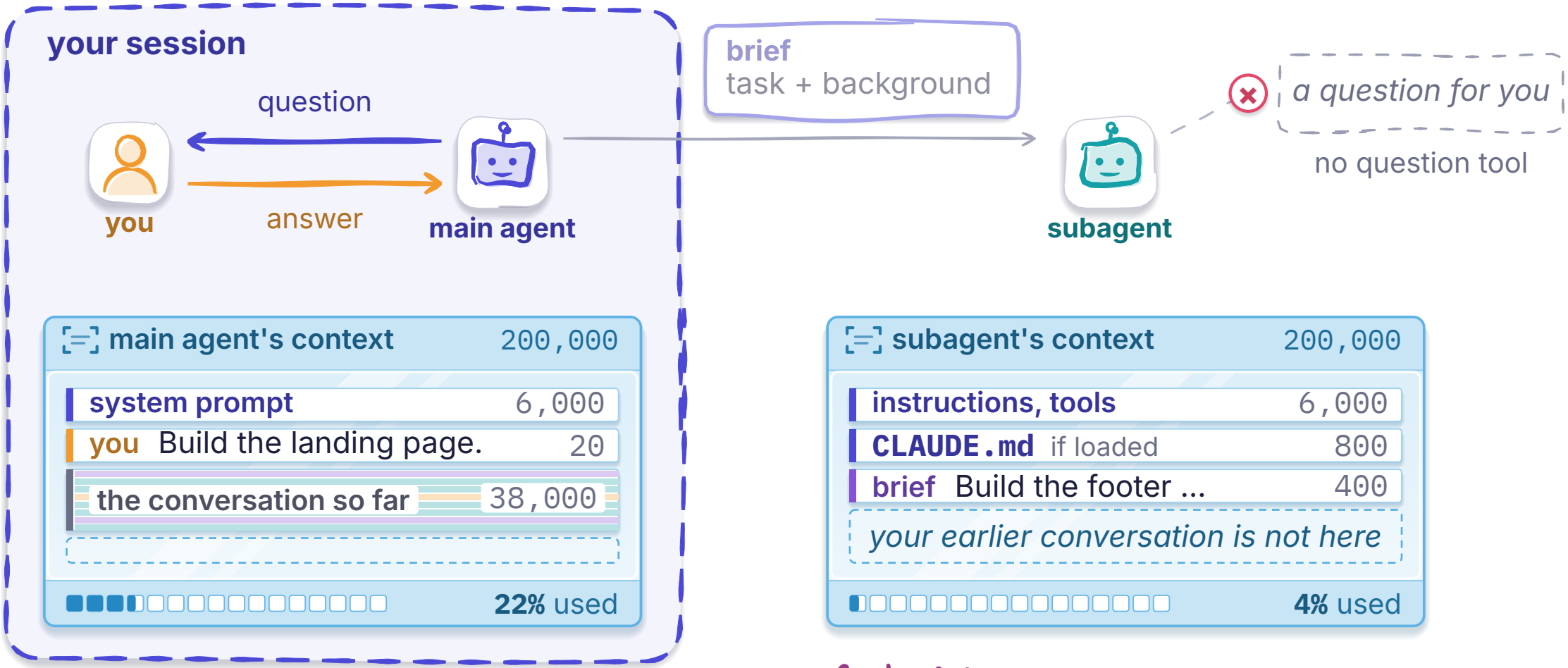


The main agent hands a task to a subagent



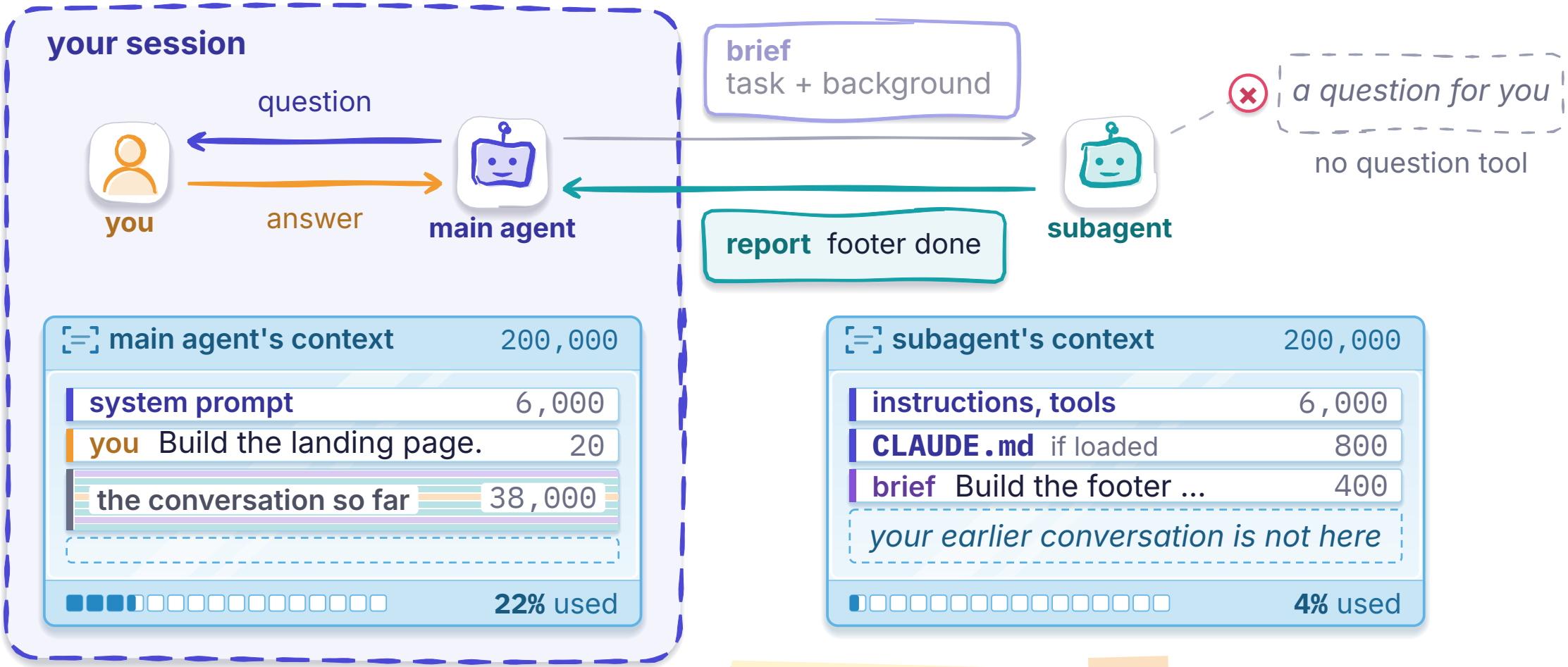
So the brief must carry the earlier decisions that matter.

The main agent hands a task to a subagent



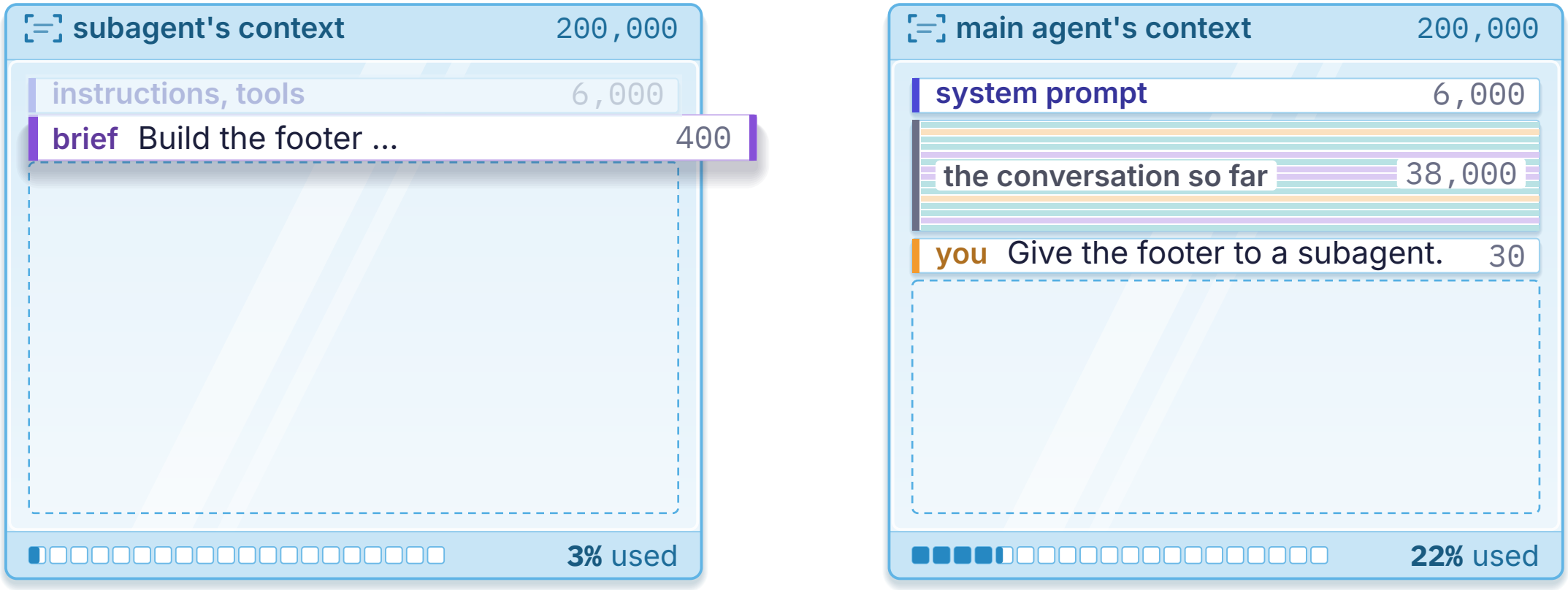
A decision left open: it assumes, or reports it could not finish.

The main agent hands a task to a subagent

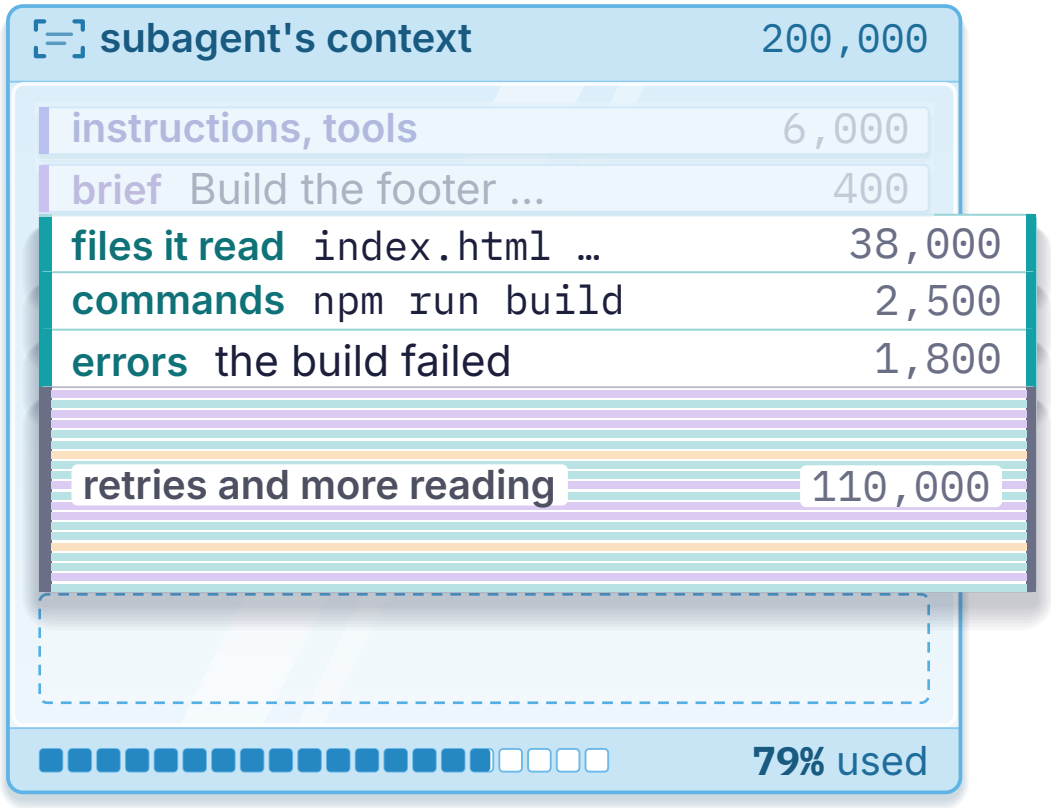


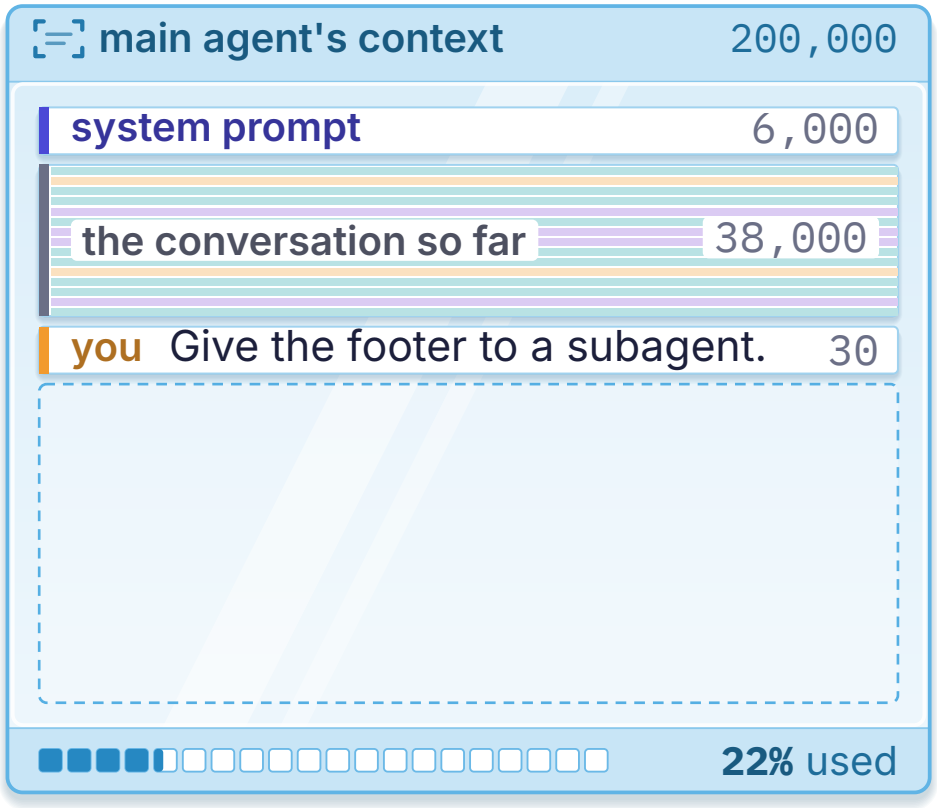
A subagent: a second agent the main agent starts for one delegated task.

Only the report comes back



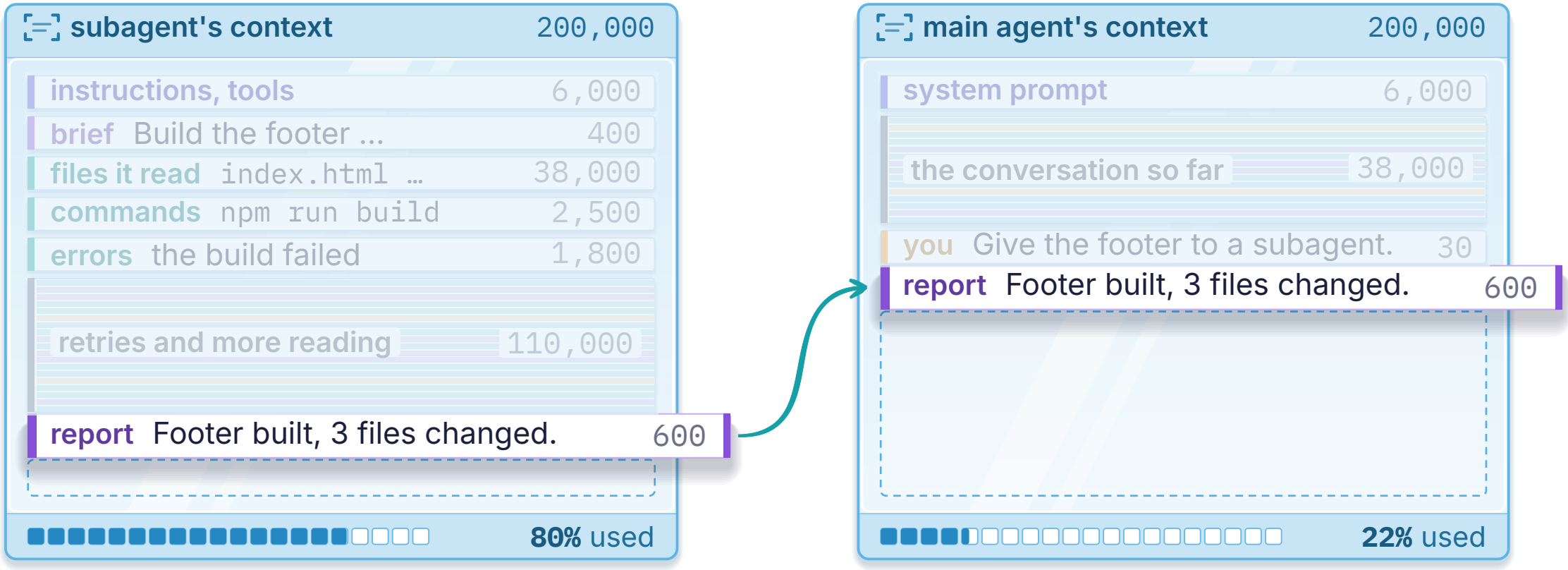
Only the report comes back





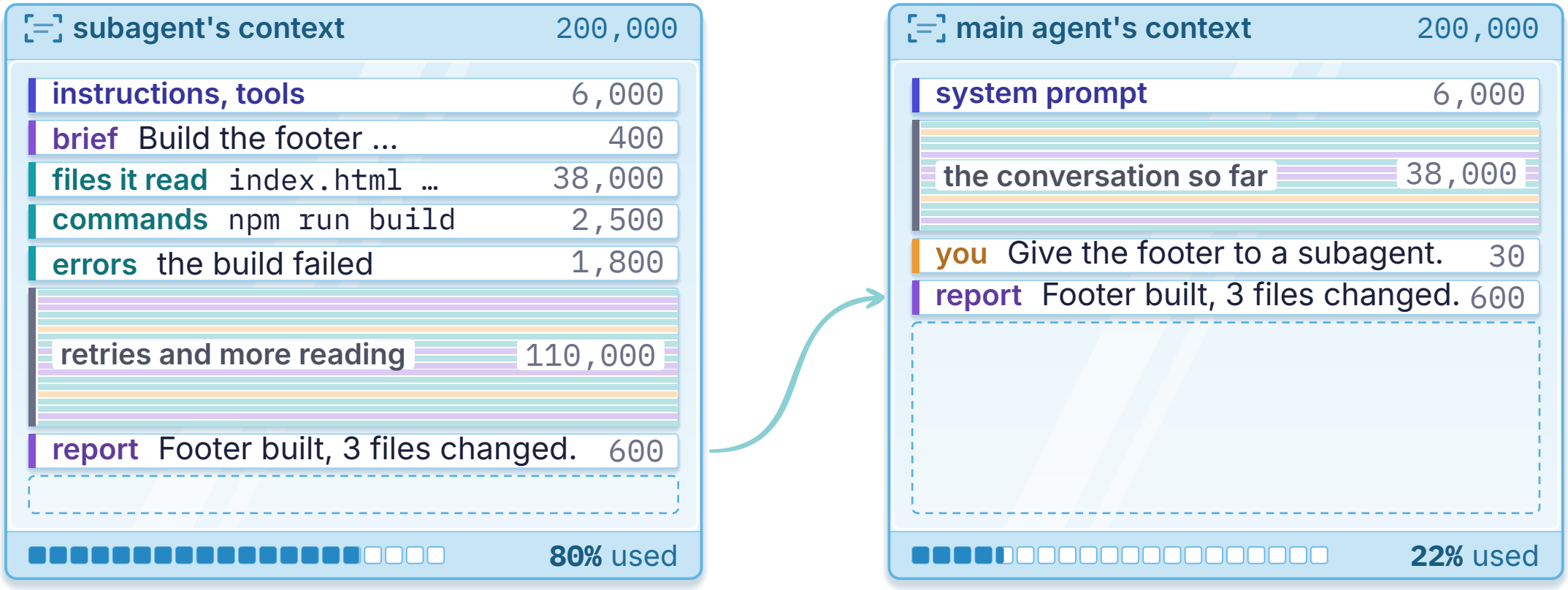
All of its work lands in its own window.

Only the report comes back



The work stays here; its transcript can be inspected or resumed.

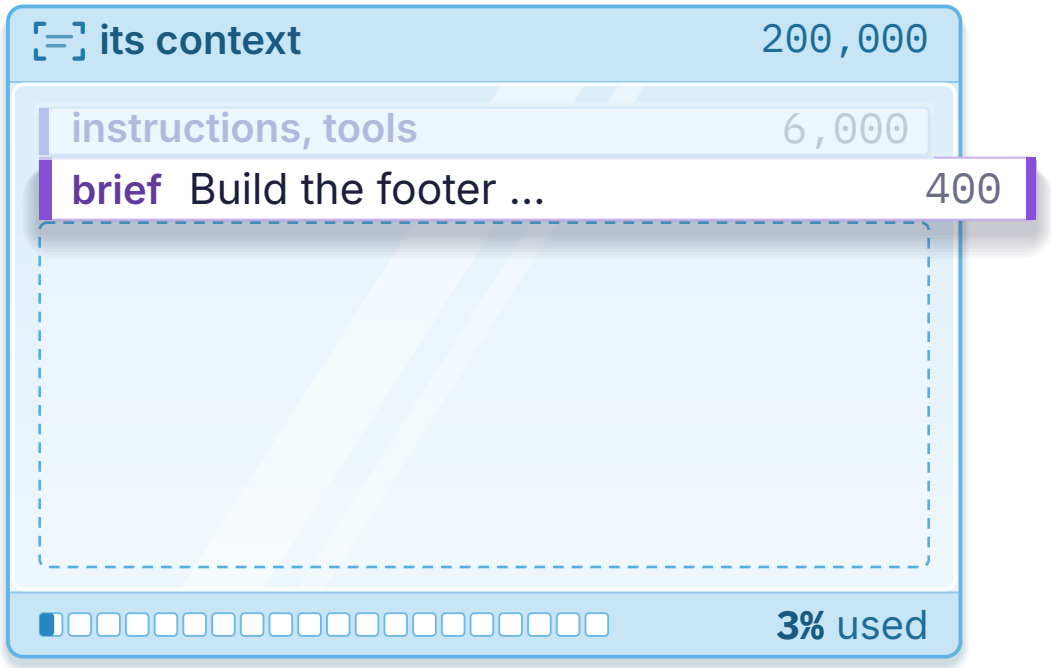
Only the report comes back



The window is limited. Each subtask gathers its own context; the main agent keeps the conclusions.

Two starting points: a fresh subagent or a fork

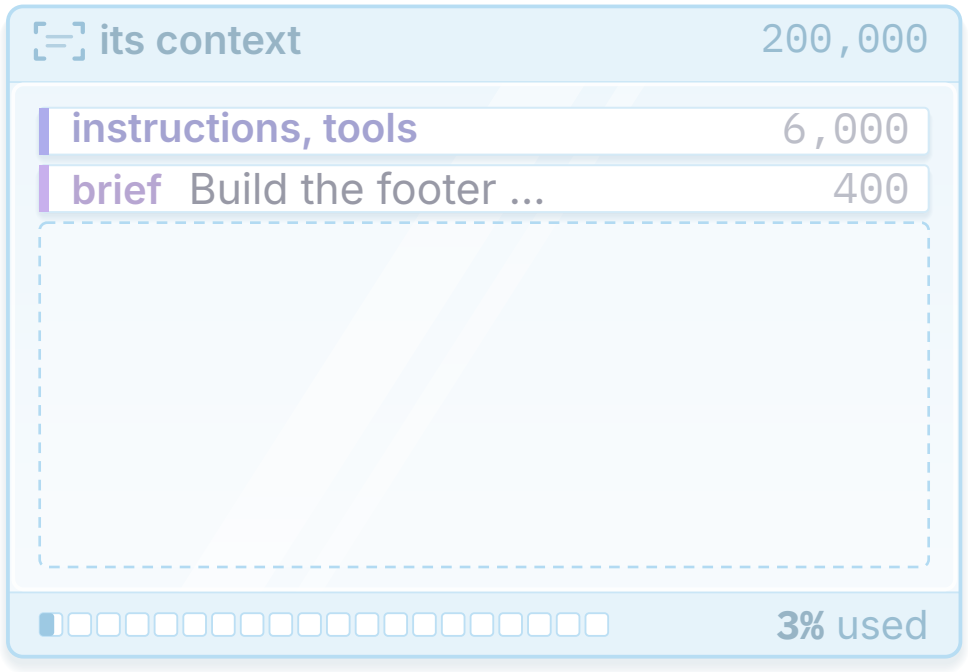
a fresh subagent
starts from a brief



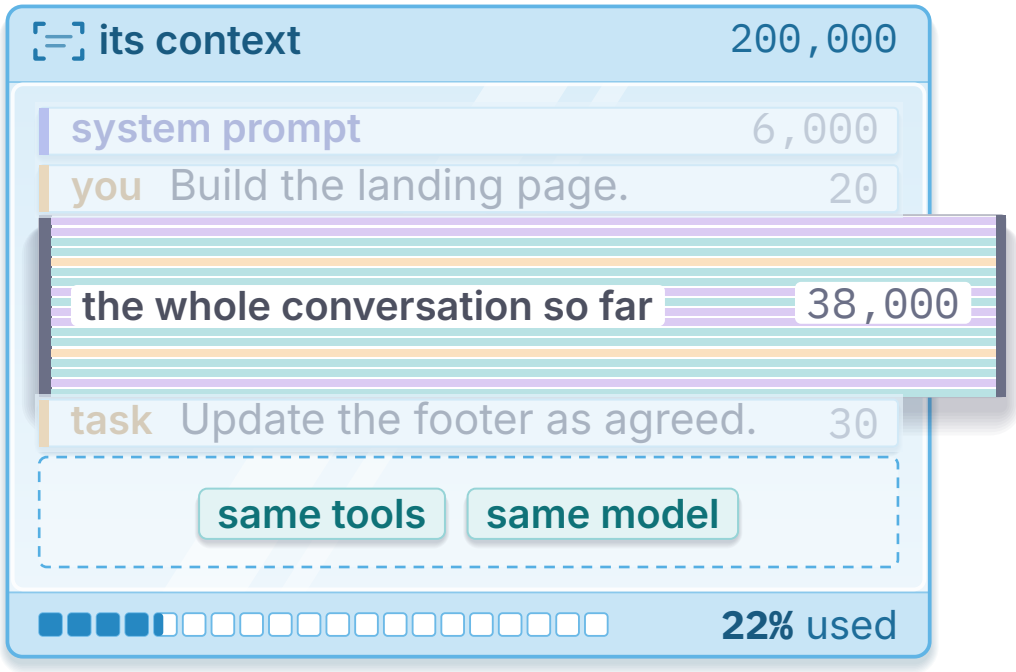
For a self-contained task: a small, focused context.

Two starting points: a fresh subagent or a fork

a fresh subagent
starts from a brief



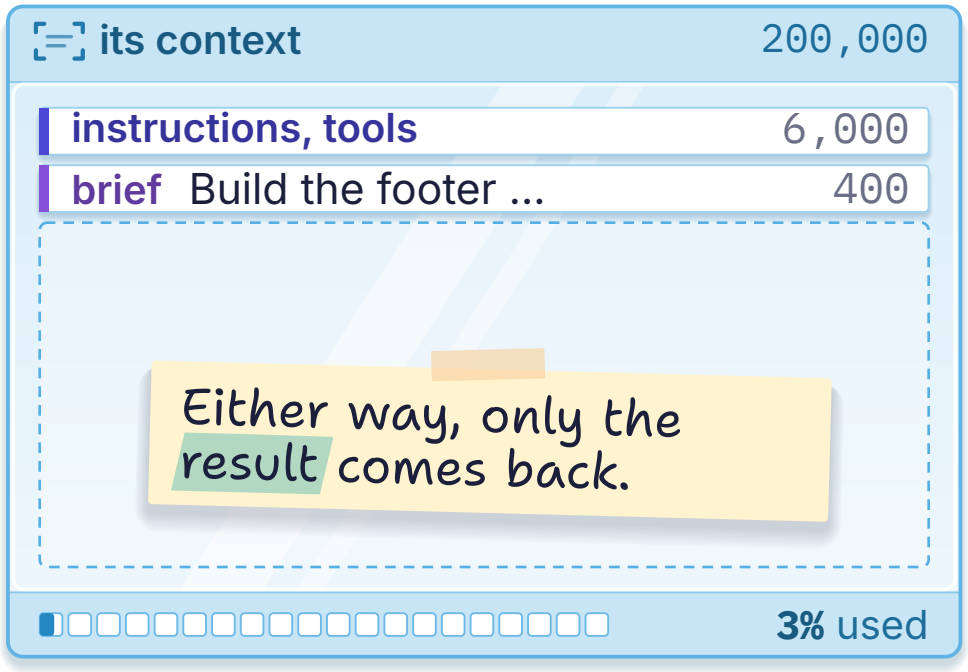
a fork
starts from a copy of the conversation



start one yourself `> /subtask Update the footer as agreed`

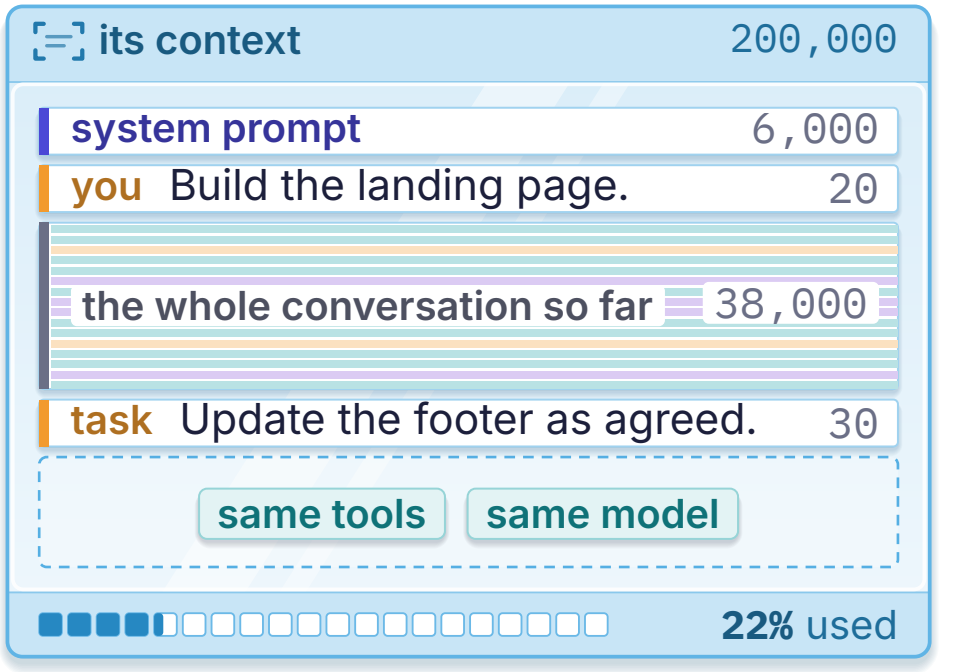
Two starting points: a fresh subagent or a fork

a fresh subagent
starts from a brief



A task that stands on its own.

a fork
starts from a copy of the conversation



A task that needs what you worked out together.

Ask for subagents in plain words

agent session `~/site`

you typed by you

Please open four subagents, one per variant of the landing page: minimal, bold, dark and playful. Run them in parallel, each on its own branch in its own worktree, and report back when all four are done.

You ask in ordinary language: how many, in parallel, and where each one works.

Ask for subagents in plain words

agent session `~/site`

you typed by you

Please open four subagents, one per variant of the landing page: minimal, bold, dark and playful. Run them in parallel, each on its own branch in its own worktree, and report back when all four are done.

agent written by the model

Starting four subagents, each in its own worktree.

tool call asked by the model

```
Agent(Build the minimal landing page variant)
Agent(Build the bold landing page variant)
Agent(Build the dark landing page variant)
Agent(Build the playful landing page variant)
```

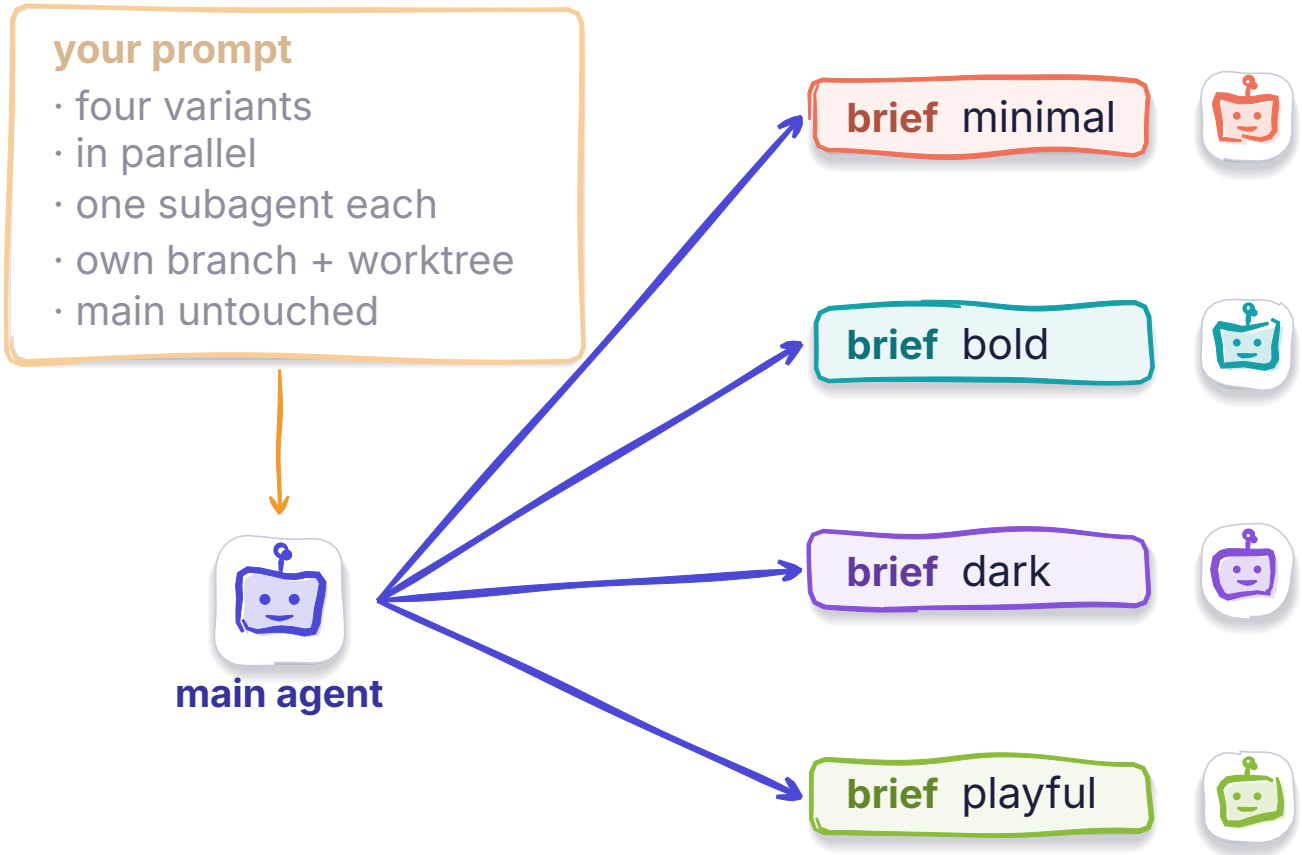
Check: four tasks, four worktrees, as you asked.

One prompt fans out to four subagents

your prompt

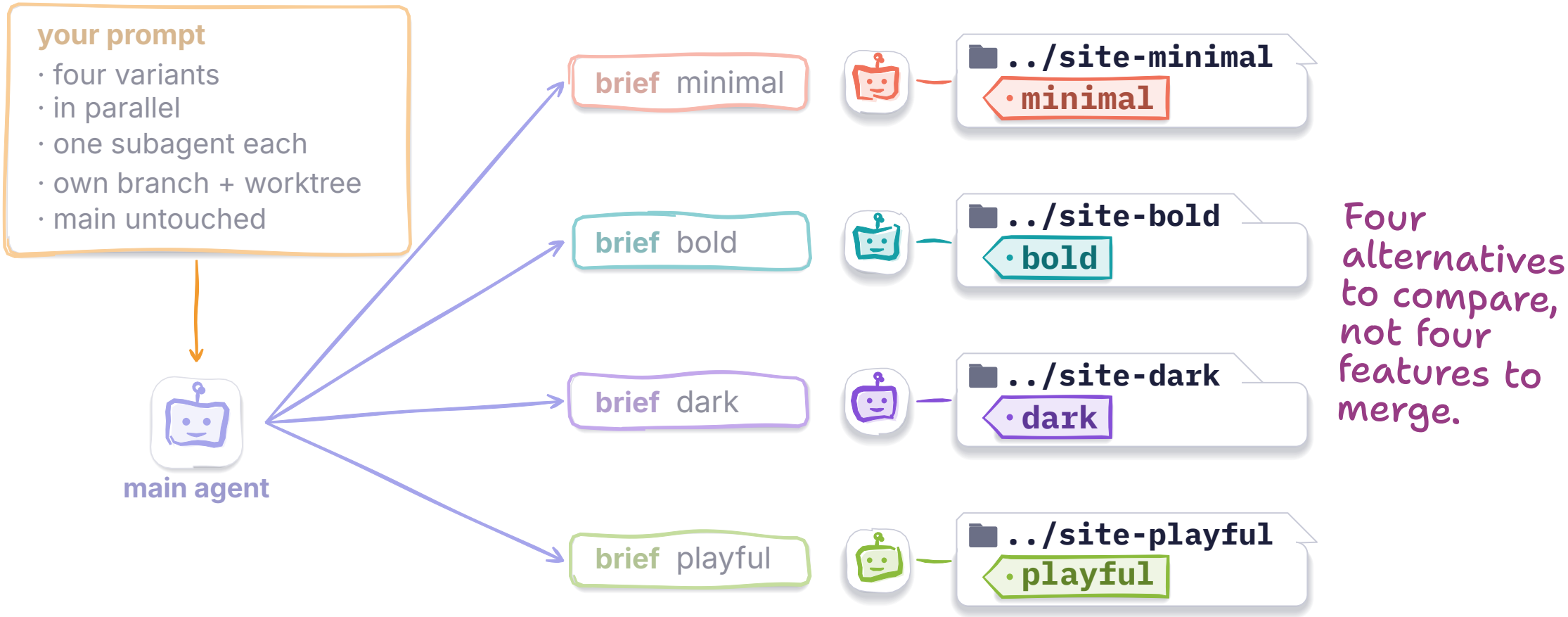
- four variants
- in parallel
- one subagent each
- own branch + worktree
- main untouched

One prompt fans out to four subagents

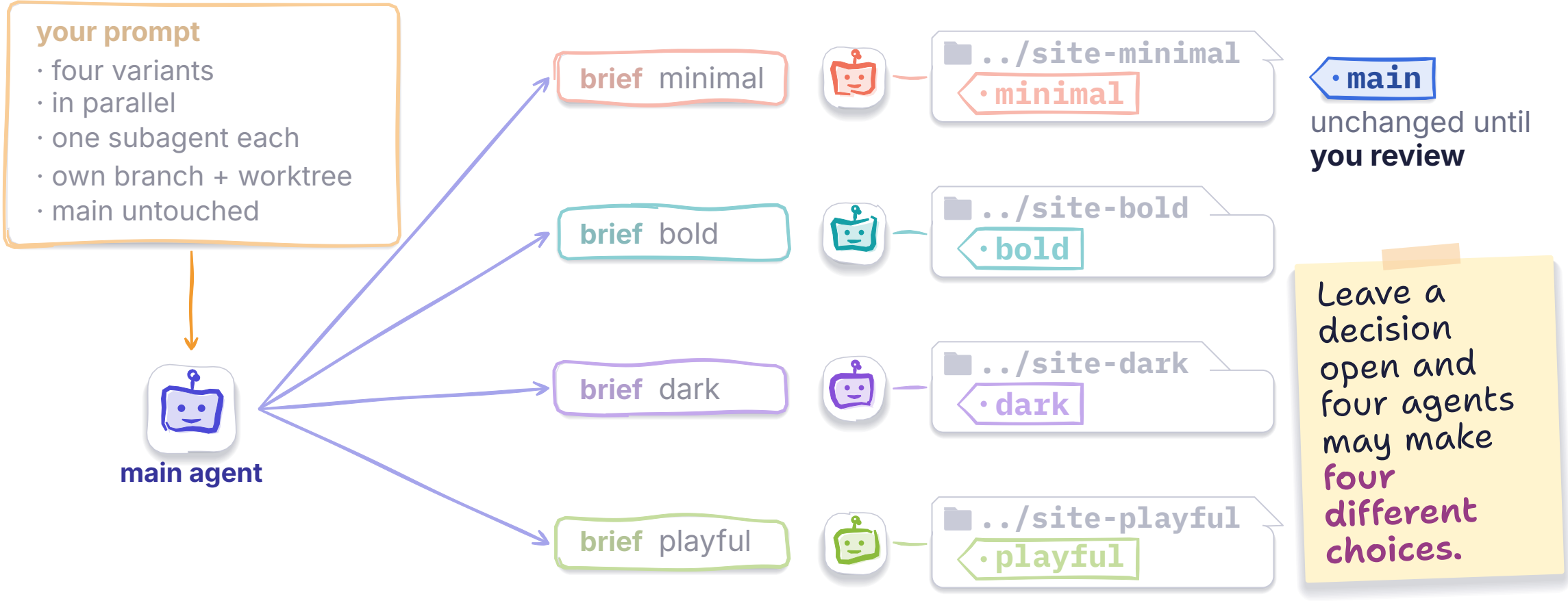


Each one gets a brief for its own variant.

One prompt fans out to four subagents



One prompt fans out to four subagents



Watching subagents while they run

shell

```
> Please open four subagents, one per variant …

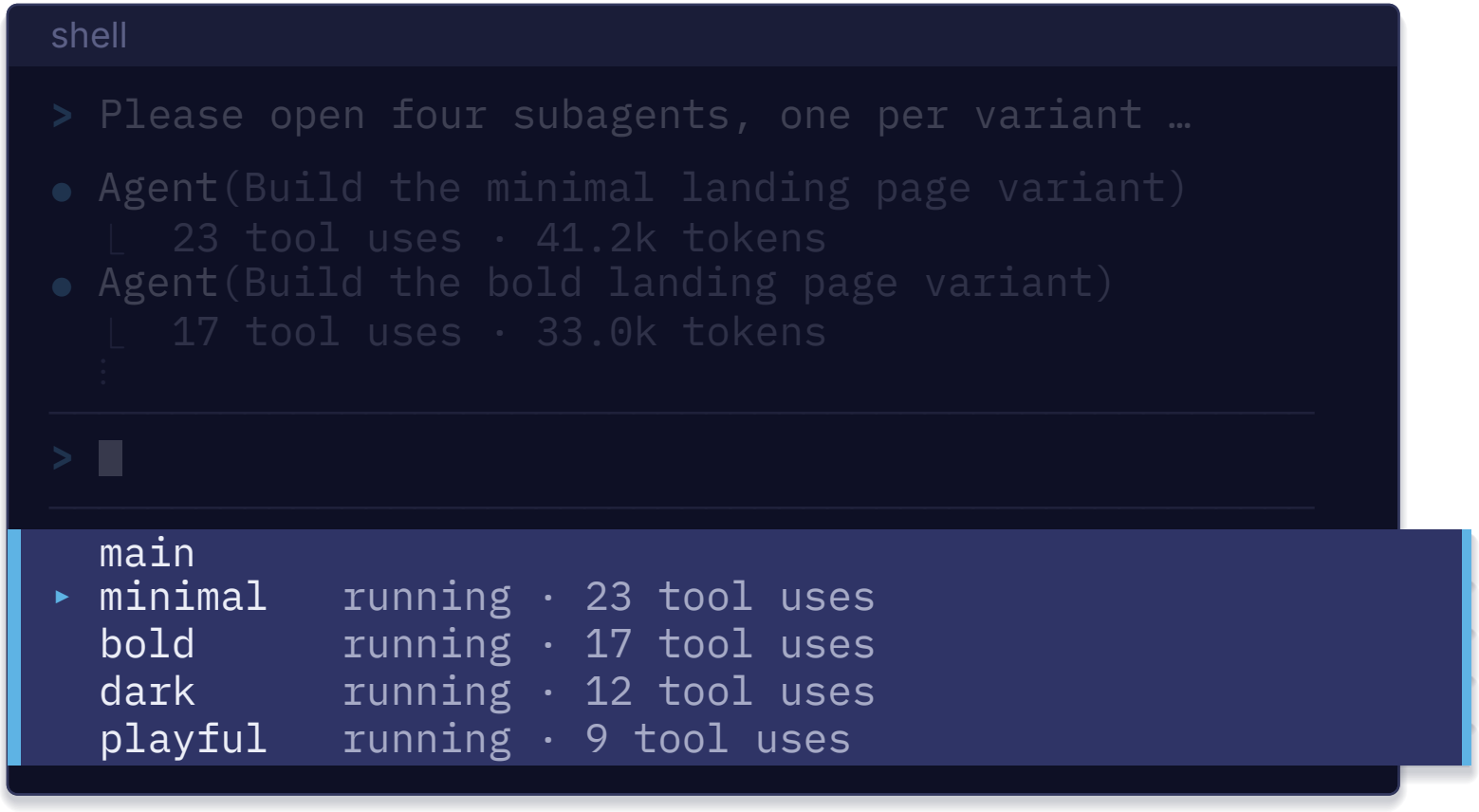
● Agent(Build the minimal landing page variant)
  └  23 tool uses · 41.2k tokens
● Agent(Build the bold landing page variant)
  └  17 tool uses · 33.0k tokens
  ⋮

>

  main
▸ minimal   running · 23 tool uses
  bold      running · 17 tool uses
  dark      running · 12 tool uses
  playful   running · 9 tool uses
```

One row per subagent: the task in brackets, its progress under it.

Watching subagents while they run



A panel under the prompt: the main agent and each subagent.

↑ ↓	move between rows
Enter	open its transcript
x	stop it
Esc	back to the prompt

Watching subagents while they run

shell

```
> Please open four subagents, one per variant …

● Agent(Build the minimal landing page variant)
  └ Done · 31 tool uses · 52.0k tokens
● Agent(Build the bold landing page variant)
  └ 17 tool uses · 33.0k tokens
  ⋮

> 

  main

▸ bold       running · 17 tool uses
  dark       running · 12 tool uses
  playful    running · 9 tool uses
```

Finished: it leaves the panel, and its report reaches the main agent.

/tasks: everything this session runs in the background

shell

```
> /tasks

  Agent  Build the minimal landing page variant  running
  Agent  Build the bold landing page variant     running   ⓐ
  Agent  Build the dark landing page variant     done
  Bash   npm run build                           running
  Bash   npm test                                running   ⓑ
```

ⓐ subagents, running or recently finished

ⓑ commands the agent started and did not wait for: a build, a test run

/tasks: everything this session runs in the background

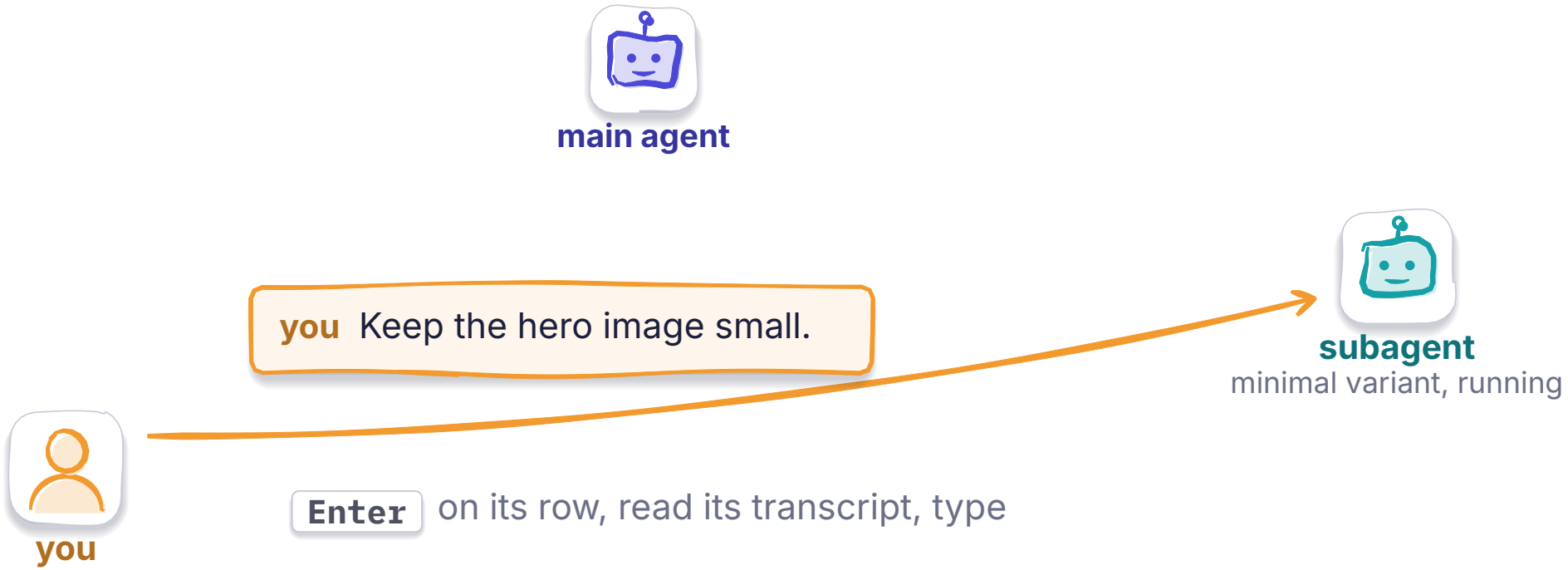
shell

```
> /tasks

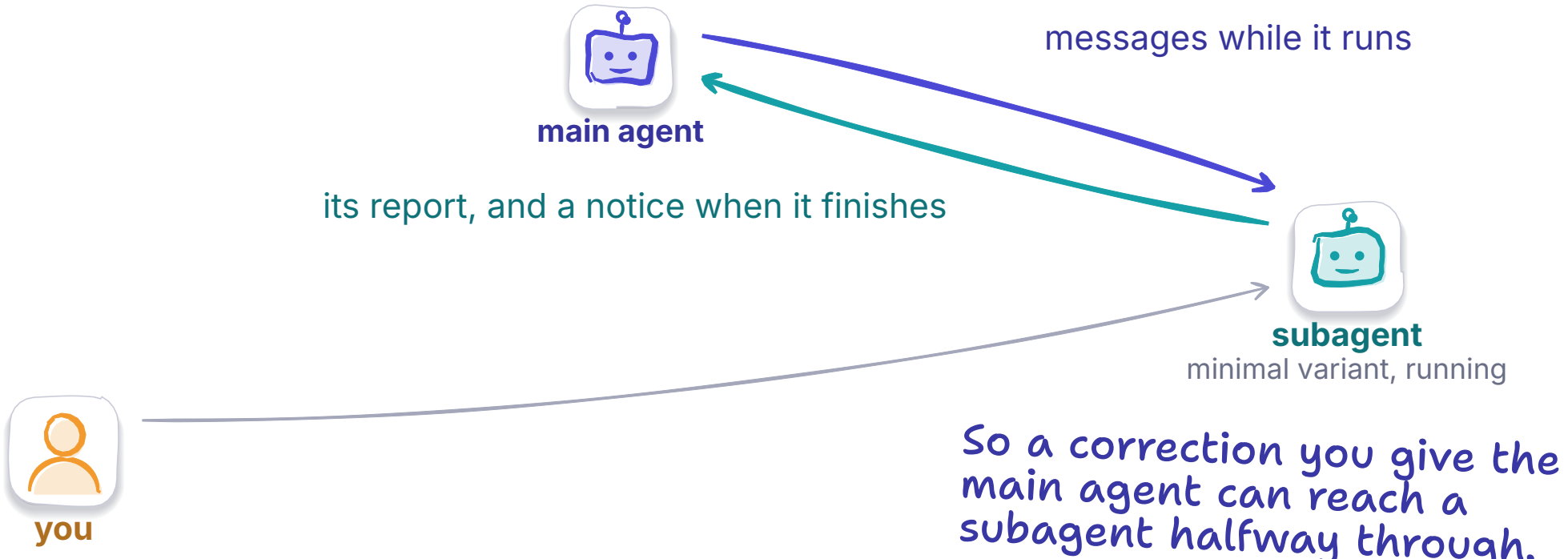
  Agent  Build the minimal landing page variant  running
  Agent  Build the bold landing page variant     running
  Agent  Build the dark landing page variant     done
  Bash   npm run build                           running
  Bash   npm test                                running
```

One list for the background work of this session: its commands and its subagents.

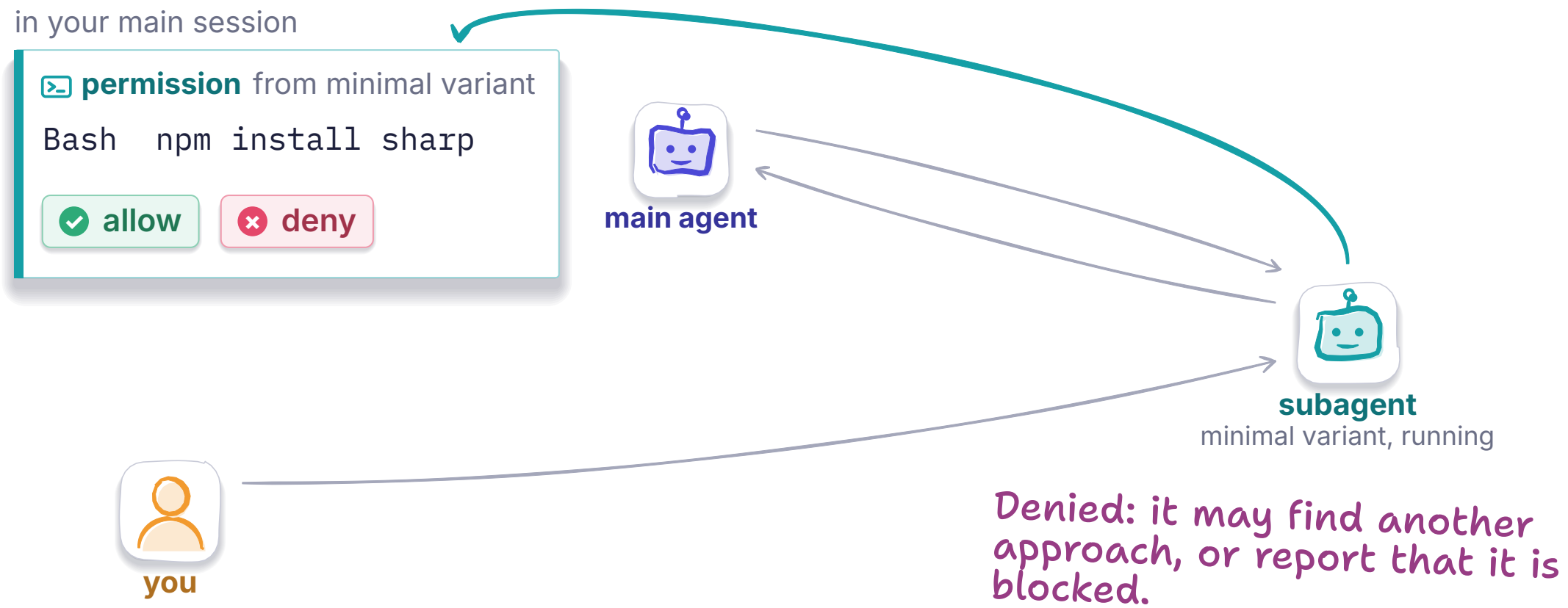
You can still reach a running subagent



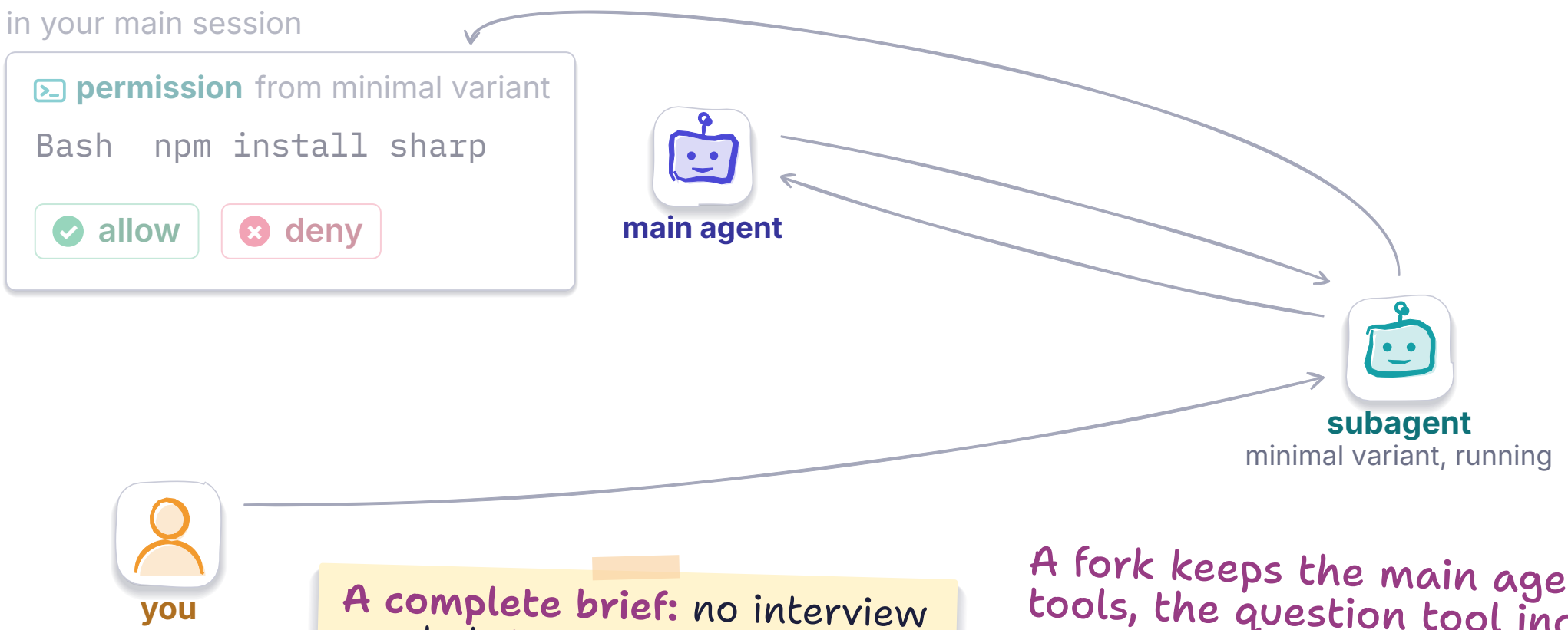
You can still reach a running subagent



You can still reach a running subagent



You can still reach a running subagent



A complete brief: no interview needed. **Messages:** corrections during the run.

A fork keeps the main agent's tools, the question tool included.

Several sessions, side by side

terminal 1

agent session `~/contact`

you typed by you

Add a contact form.

agent written by the model

Writing the form and its checks.

Several sessions, side by side

terminal 1

agent session `~/contact`

you typed by you

Add a contact form.

agent written by the model

Writing the form and its checks.

terminal 2

agent session `~/pricing`

you typed by you

Build the pricing page.

tool call waiting for your permission

`Edit pricing.html`

terminal 3

agent session `~/blog`

you typed by you

Lay out the blog index.

agent written by the model

One column or two?

Ordinary sessions, like the one you have used since week one: only several at once.

Several sessions, side by side

terminal 1

agent session `~/contact`

you typed by you

Add a contact form.

agent written by the model

Writing the form and its checks.

own context window

accept edits on

terminal 2

agent session `~/pricing`

you typed by you

Build the pricing page.

tool call waiting for your permission

```
Edit pricing.html
```

own context window

default: asks first

terminal 3

agent session `~/blog`

you typed by you

Lay out the blog index.

agent written by the model

One column or two?

own context window

plan mode on

They work independently: tell each one the decisions that affect its task.

Several sessions, side by side

terminal 1

agent session `~/contact`

you typed by you

Add a contact form.

agent written by the model

Writing the form and its checks.

own context window

accept edits on

working

terminal 2

agent session `~/pricing`

you typed by you

Build the pricing page.

tool call waiting for your permission

`Edit pricing.html`

own context window

default: asks first

waiting for you: permission

terminal 3

agent session `~/blog`

you typed by you

Lay out the blog index.

agent written by the model

One column or two?

own context window

plan mode on

waiting for you: question

Agent view: every session in one list

shell

```
Needs input
  * pricing page      permission: edit pricing.html  3m
  * blog layout       question: one column or two?   6m
Working
  * contact form      editing form.js                1m
Completed
  · minimal variant   done, ready to review          14m

> describe a task and press Enter to start a new session
```

← on an empty prompt

puts this session in the background and opens the list

One row per session, grouped by state: the ones that need you, then the ones still working.

Agent view: every session in one list

shell

```
Needs input
  * pricing page      permission: edit pricing.html  3m
  * blog layout       question: one column or two?   6m
Working
  * contact form      editing form.js                1m
Completed
  · minimal variant   done, ready to review          14m

> describe a task and press Enter to start a new session
```

`Enter`	open that session
←	back to the list
`Space`	peek, and reply

A task typed here starts a new session, which moves into a worktree of its own before it edits.

Agent view: every session in one list

shell

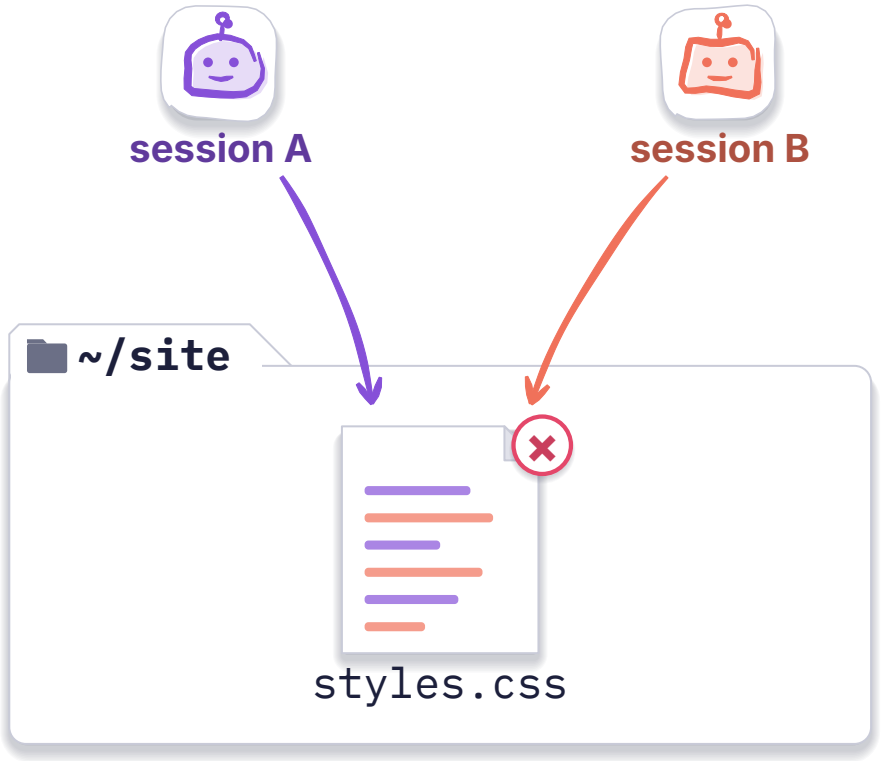
```
Needs input
  * pricing page      permission: edit pricing.html  3m
  * blog layout       question: one column or two?   6m
Working
  * contact form      editing form.js                1m
Completed
  · minimal variant   done, ready to review          14m

> describe a task and press Enter to start a new session
```

Check this group often: answering lets their work go on.

Give each session its own worktree

two sessions, one folder



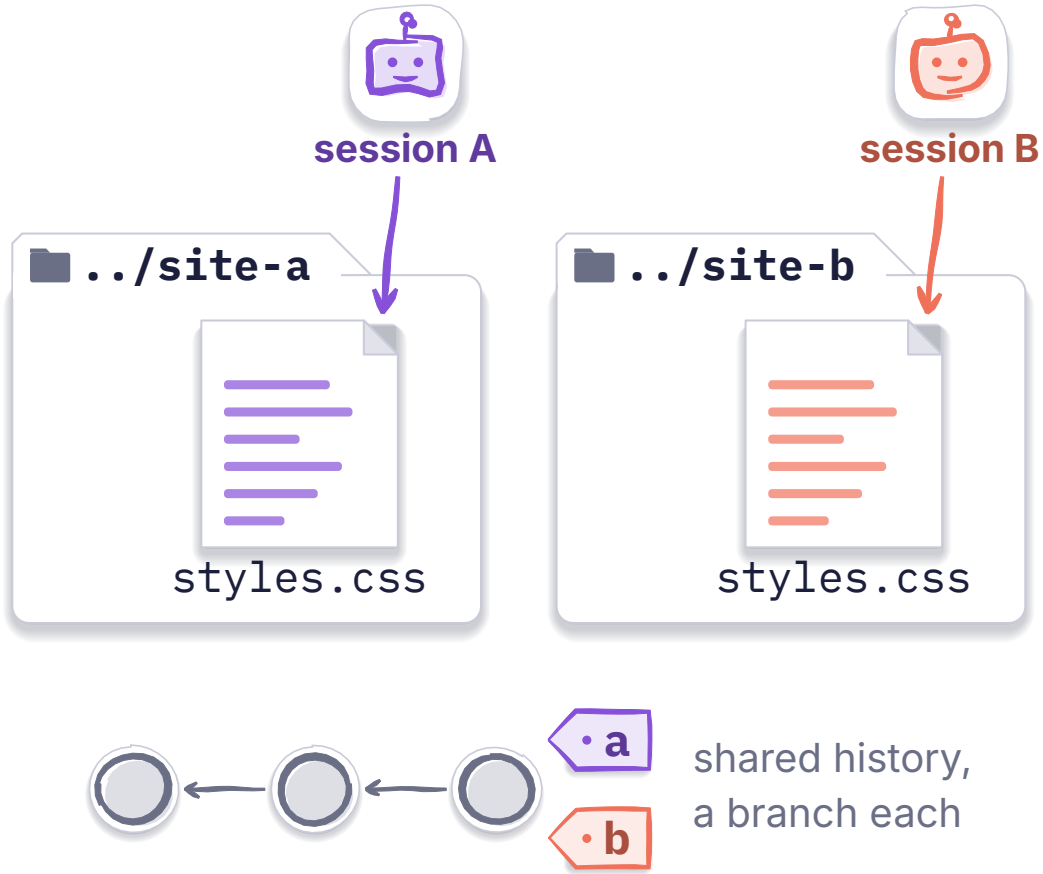
One overwrites the other's change, or works from contents that have already changed.

Give each session its own worktree

two sessions, one folder

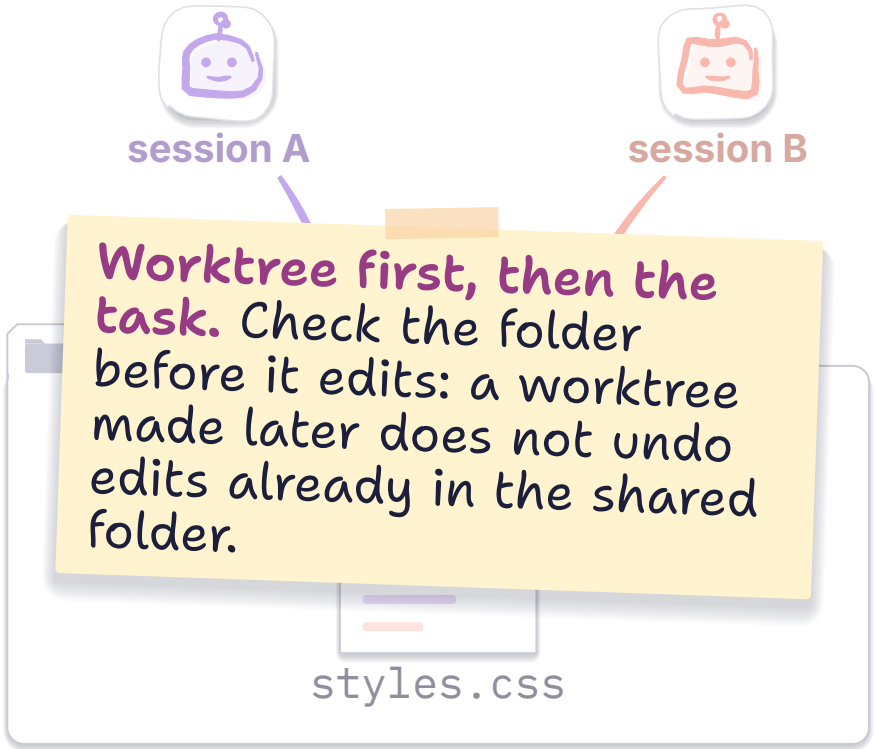


one worktree per session

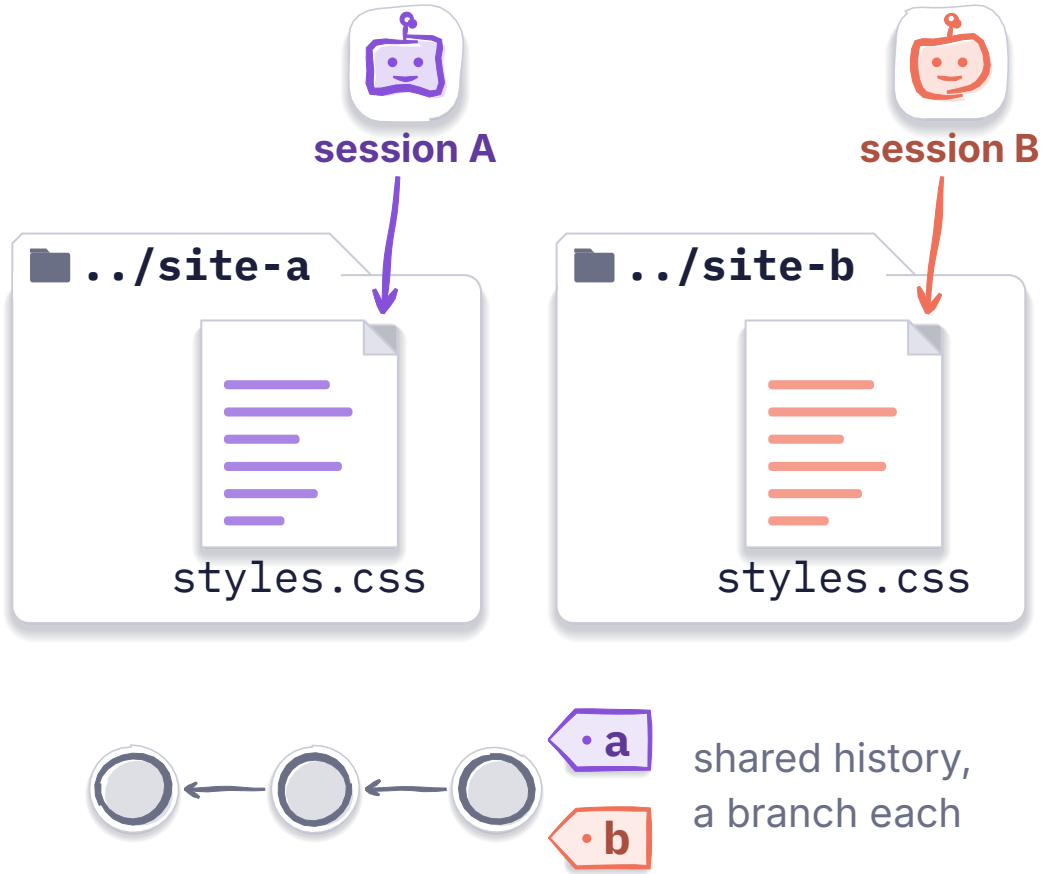


Give each session its own worktree

two sessions, one folder



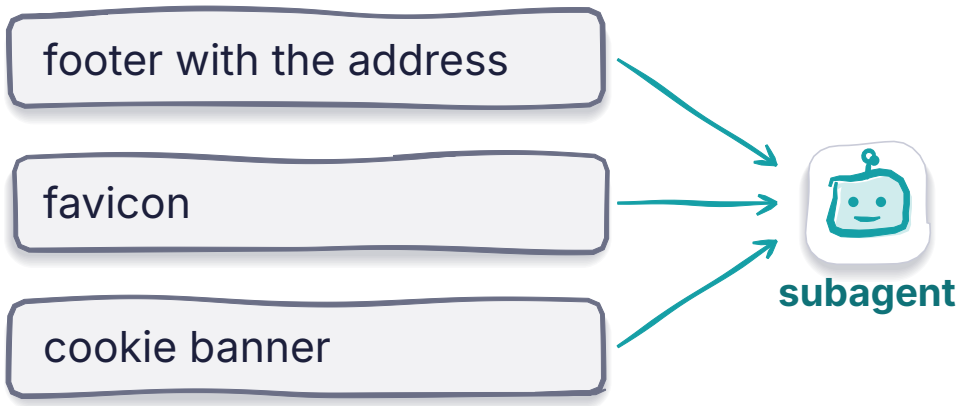
one worktree per session



Delegate it, or follow it?

a complete brief

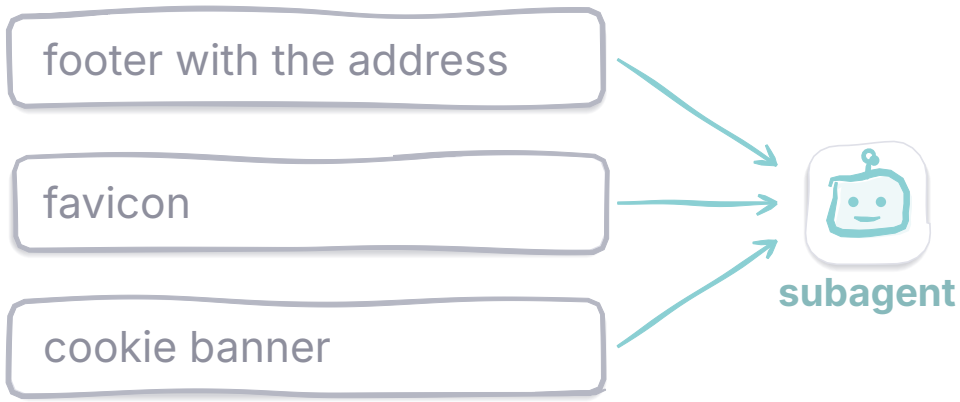
nothing will need asking



Delegate it, or follow it?

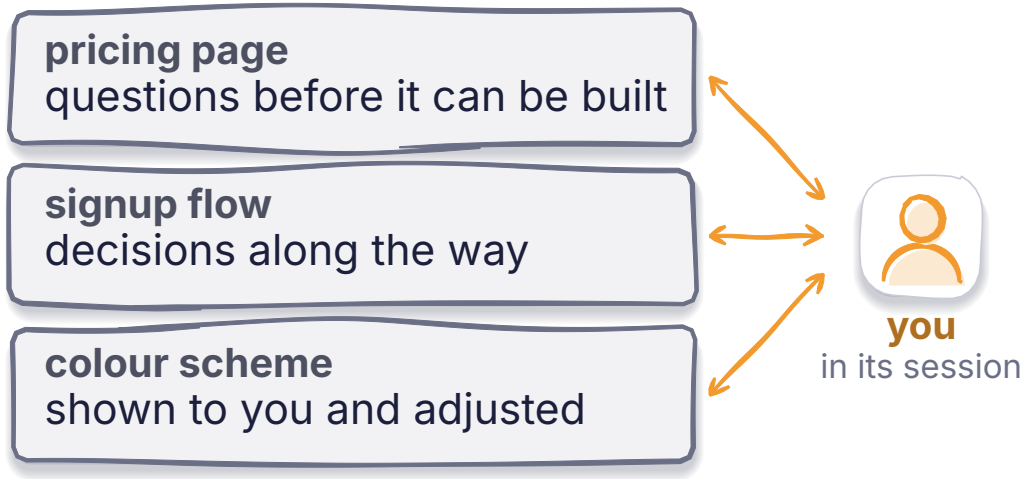
a complete brief

nothing will need asking



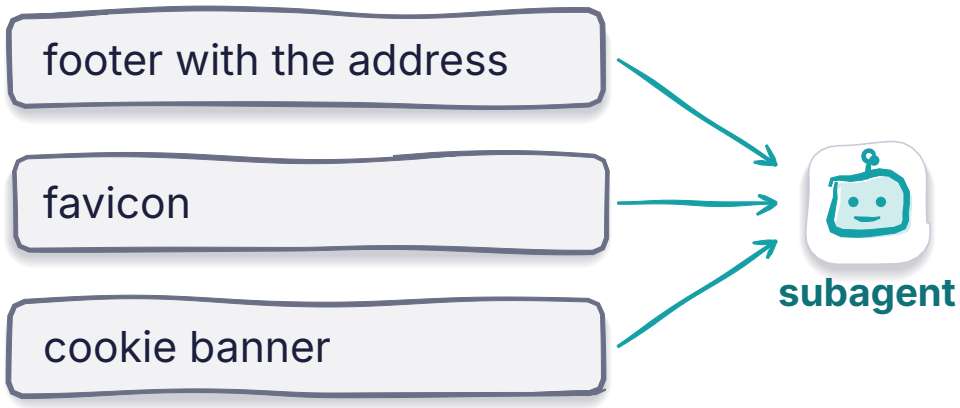
a task that needs a conversation

you stay in the loop

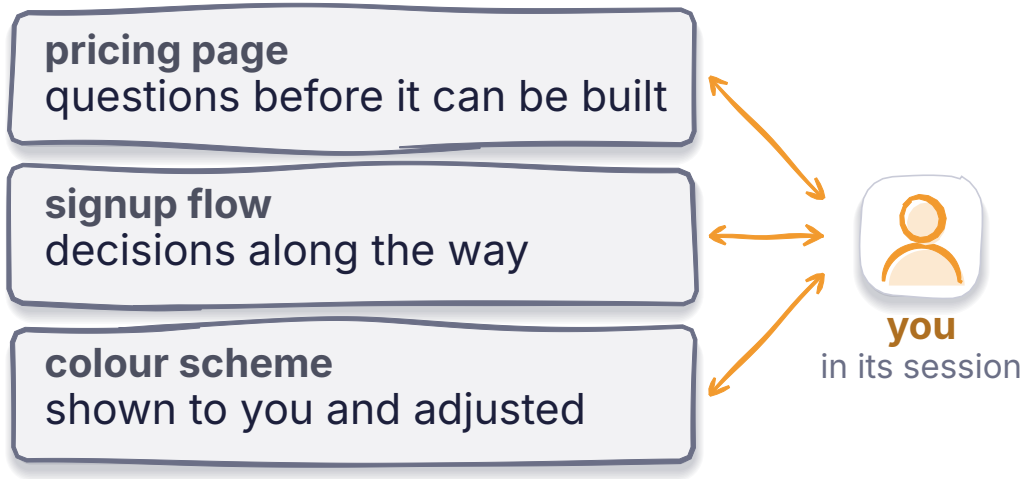


Delegate it, or follow it?

a complete brief
nothing will need asking



a task that needs a conversation
you stay in the loop

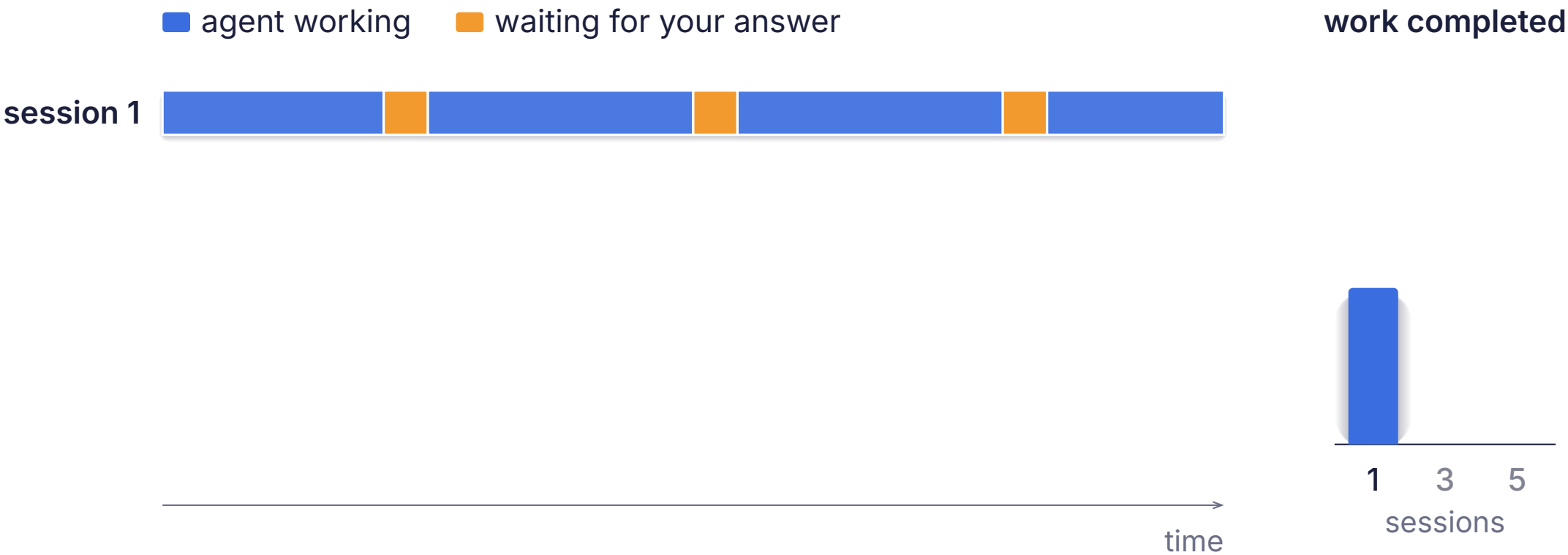


Will the task need you during the run?

No: delegate it.

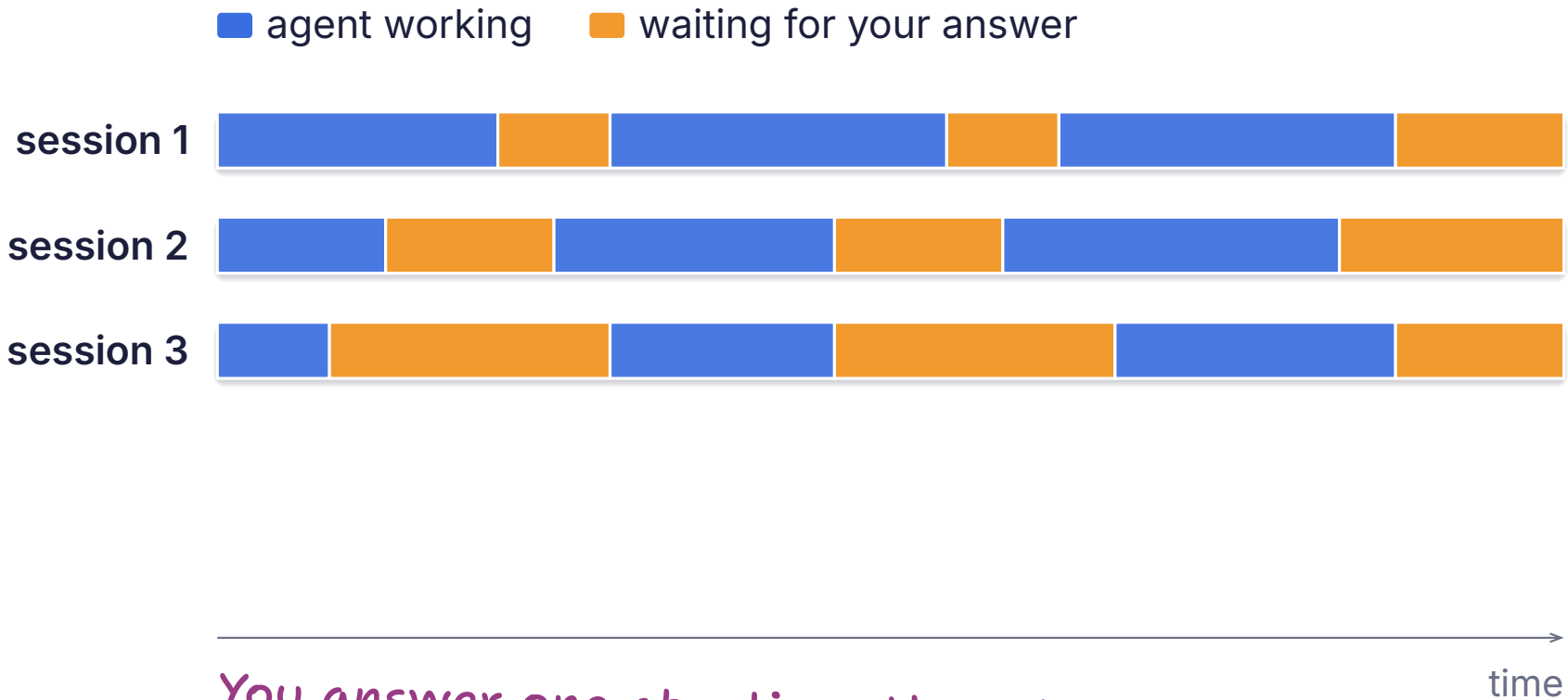
Yes: keep it where you can answer.

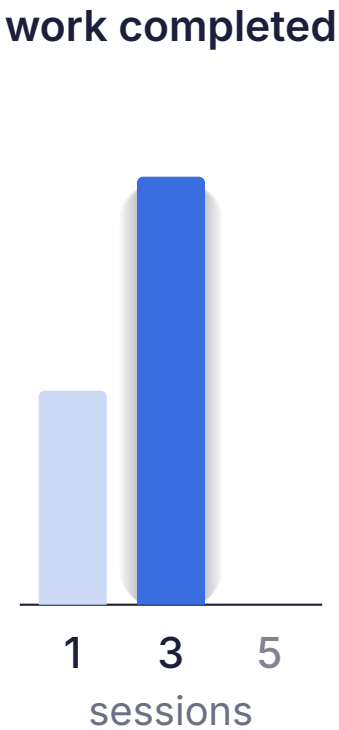
Waiting for you limits parallel work



illustrative values

Waiting for you limits parallel work

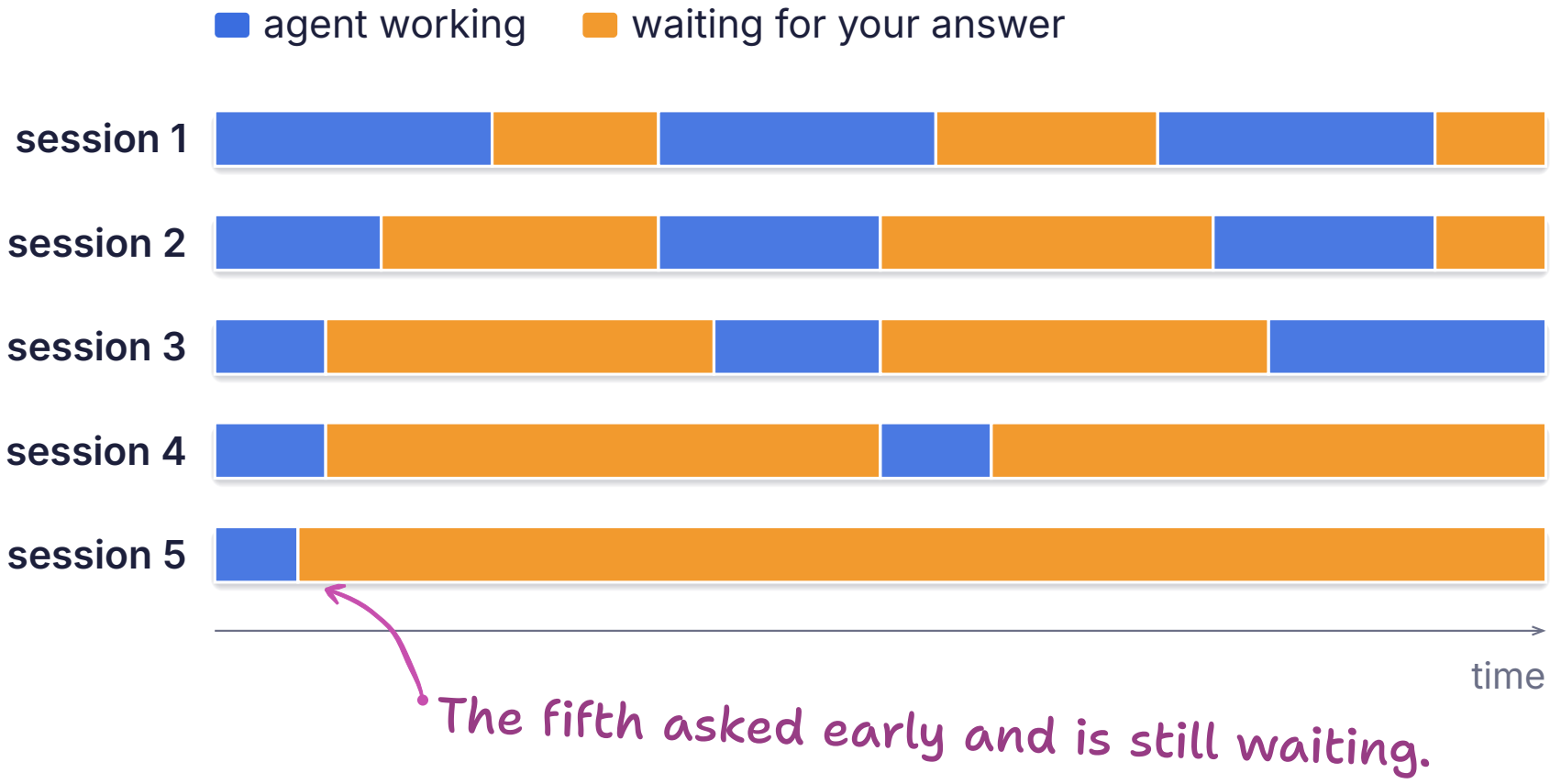


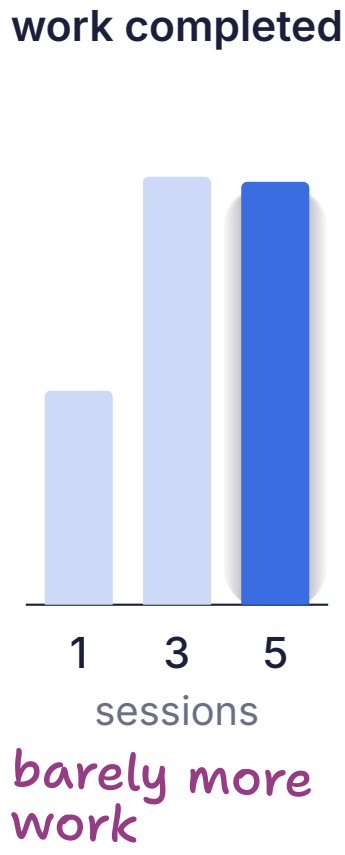


You answer one at a time; the others' questions wait in the same queue.

illustrative values

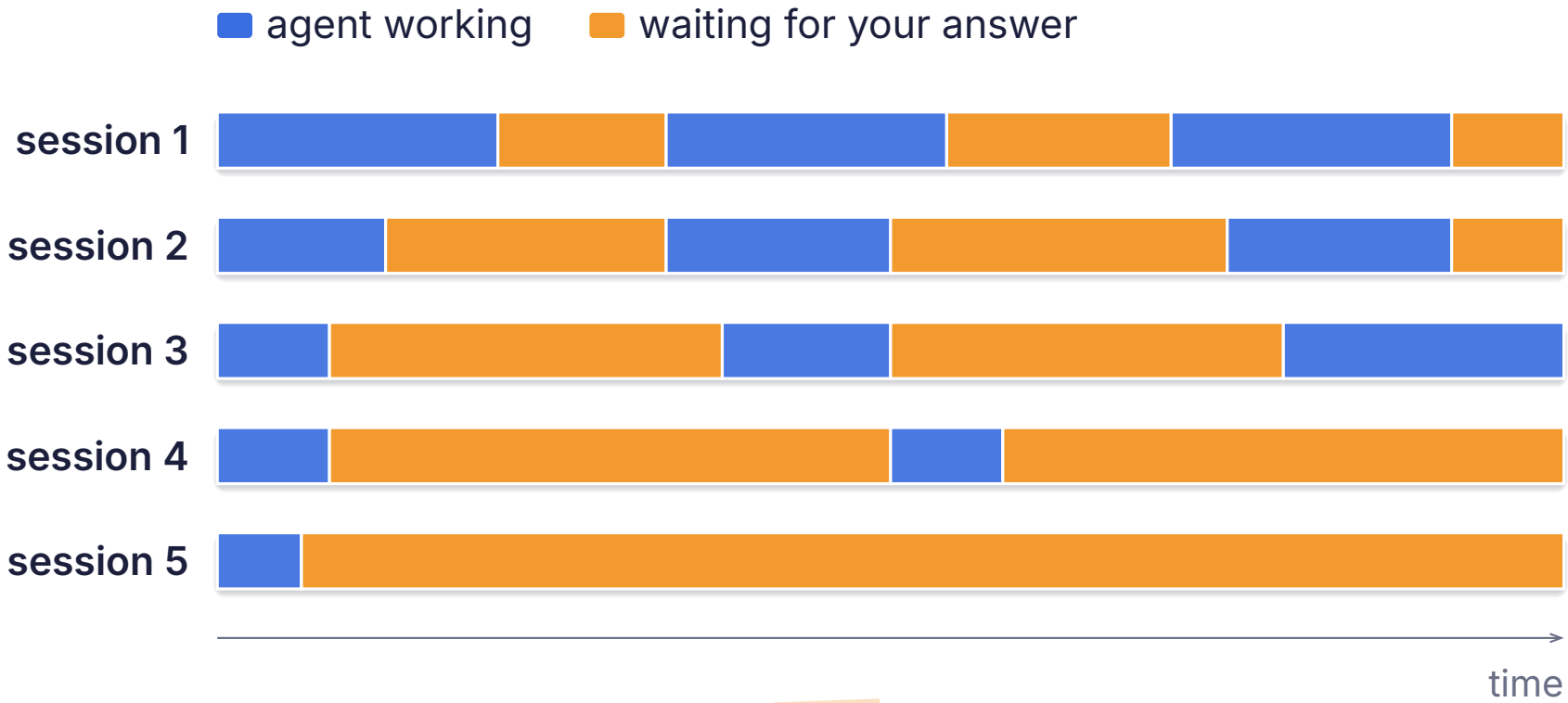
Waiting for you limits parallel work

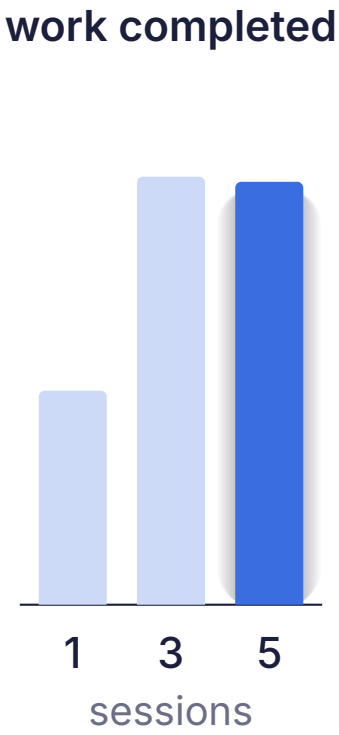




illustrative values

Waiting for you limits parallel work





Your attention is one limit. Find how many sessions you can answer and review without a growing backlog.

illustrative values

/cost: model time and elapsed time

```shell
> /cost
  └ Total cost:            $0.55
    Total duration (API):  6m 20s
    Total duration (wall): 41m 10s
    Total code changes:    212 lines added, 38 lines removed
```

An estimate at API prices. With a subscription it is not an extra bill, unless you turned on extra usage.

/cost: model time and elapsed time

```shell
> /cost
  └ Total cost:            $0.55
    Total duration (API):  6m 20s
    Total duration (wall): 41m 10s
    Total code changes:    212 lines added, 38 lines removed
```



/cost: model time and elapsed time

```shell
> /cost
  └ Total cost:            $0.55
    Total duration (API):  6m 20s
    Total duration (wall): 41m 10s
    Total code changes:    212 lines added, 38 lines removed
```



Waiting for you, but also tools and other delays: not a direct measure of your response time.

Compare it across parallel sessions with the questions left waiting.

Delegate, follow, and keep the edits apart



Delegate to a subagent

A complete task, and a report you review. Its detailed work stays out of the main context.



Follow it in a session

A task that needs a conversation: you answer decisions as they arise.

Delegate, follow, and keep the edits apart



Delegate to a subagent

A complete task, and a report you review. Its detailed work stays out of the main context.



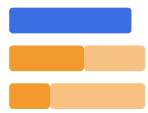
Follow it in a session

A task that needs a conversation: you answer decisions as they arise.



A worktree each

Set it up before the agent changes files; merge after review.



Watch the queue

Extra agents help only while you keep up with their questions and reviews.