Human in the Agentic Loop

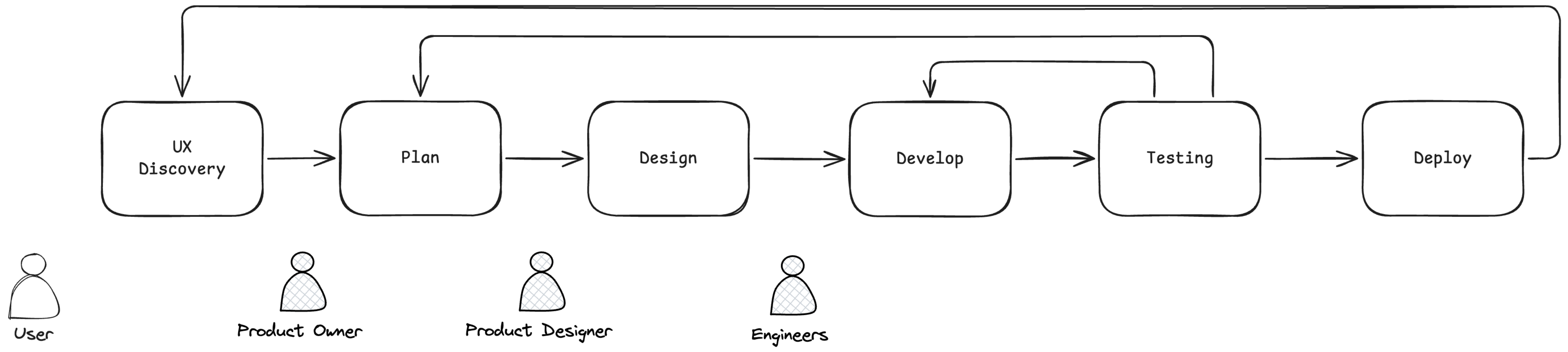
Designing an AI-first Software Engineering Workflow

Changkun Ou, March 5, 2026

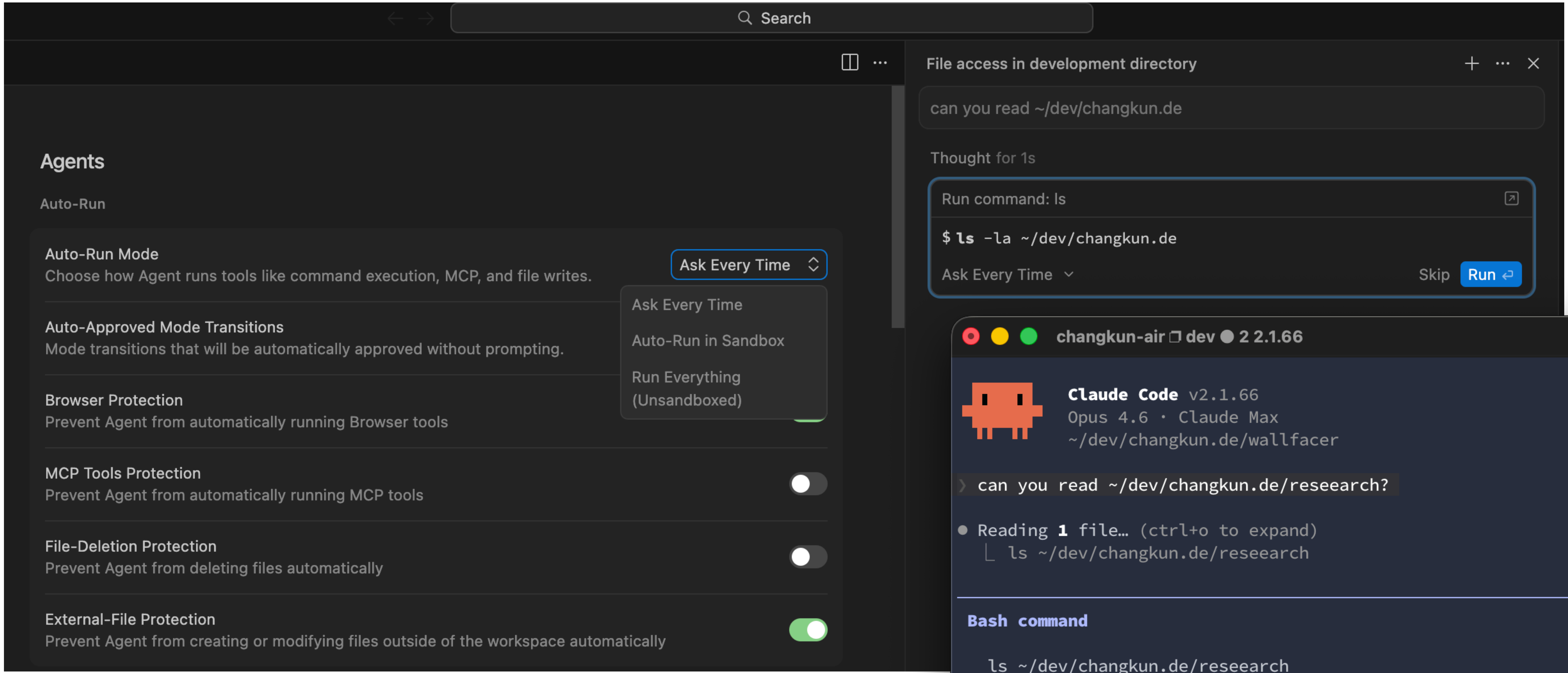
Part I: The Shift and Tension

Conventional Software Engineering Pipeline

- A typical agile pipeline where entire software engineering pipeline involves: User, Product Manager, Designer, Engineers (FE, BE, QA, etc)



The “Accept/Confirm” Button Problem



Search

Agents

Auto-Run

Auto-Run Mode
Choose how Agent runs tools like command execution, MCP, and file writes.
Ask Every Time

- Ask Every Time
- Auto-Run in Sandbox
- Run Everything (Unsandboxed)

Auto-Approved Mode Transitions
Mode transitions that will be automatically approved without prompting.

Browser Protection
Prevent Agent from automatically running Browser tools

MCP Tools Protection
Prevent Agent from automatically running MCP tools

File-Deletion Protection
Prevent Agent from deleting files automatically

External-File Protection
Prevent Agent from creating or modifying files outside of the workspace automatically

File access in development directory

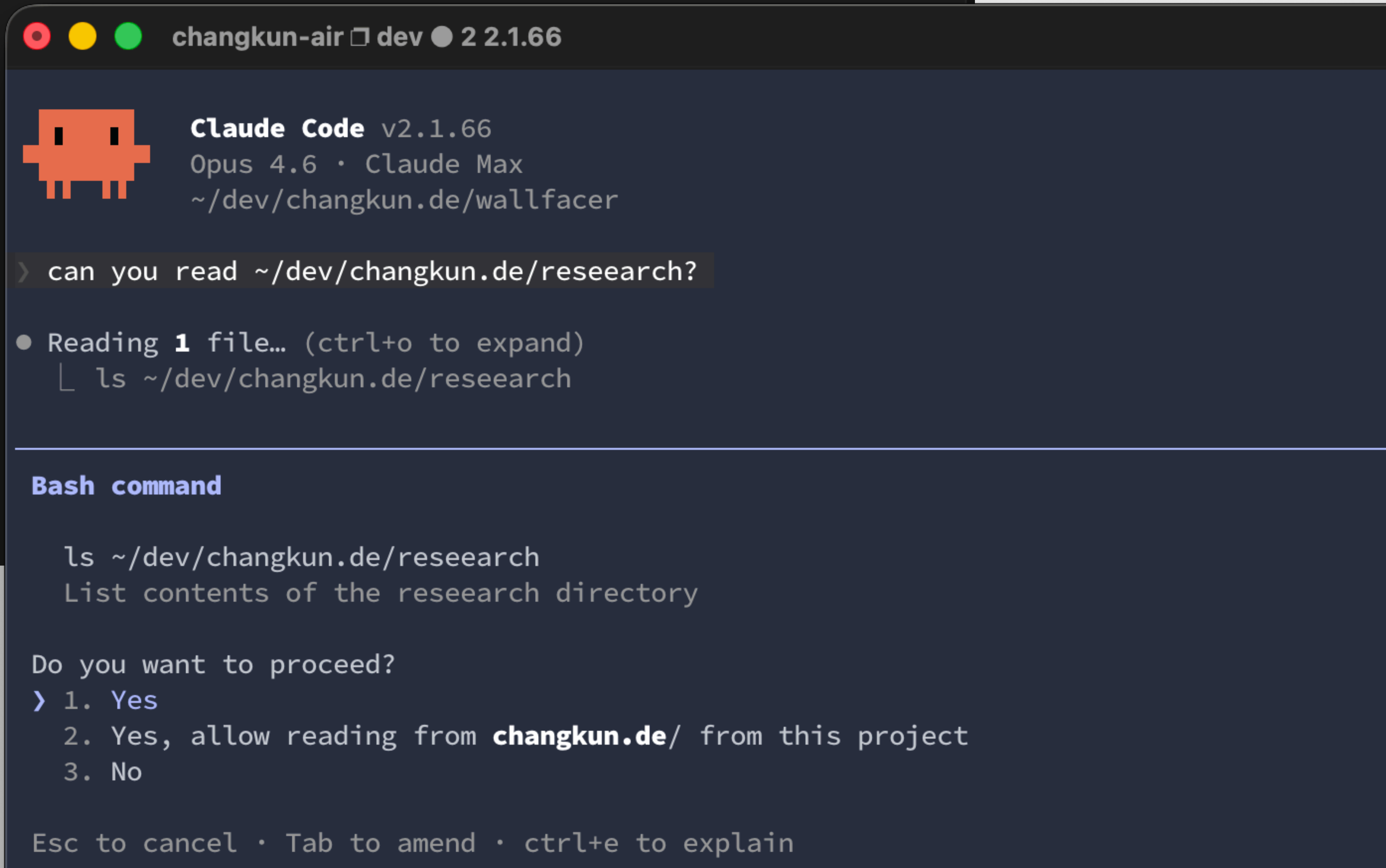
can you read ~/dev/changkun.de

Thought for 1s

Run command: ls

```
$ ls -la ~/dev/changkun.de
```

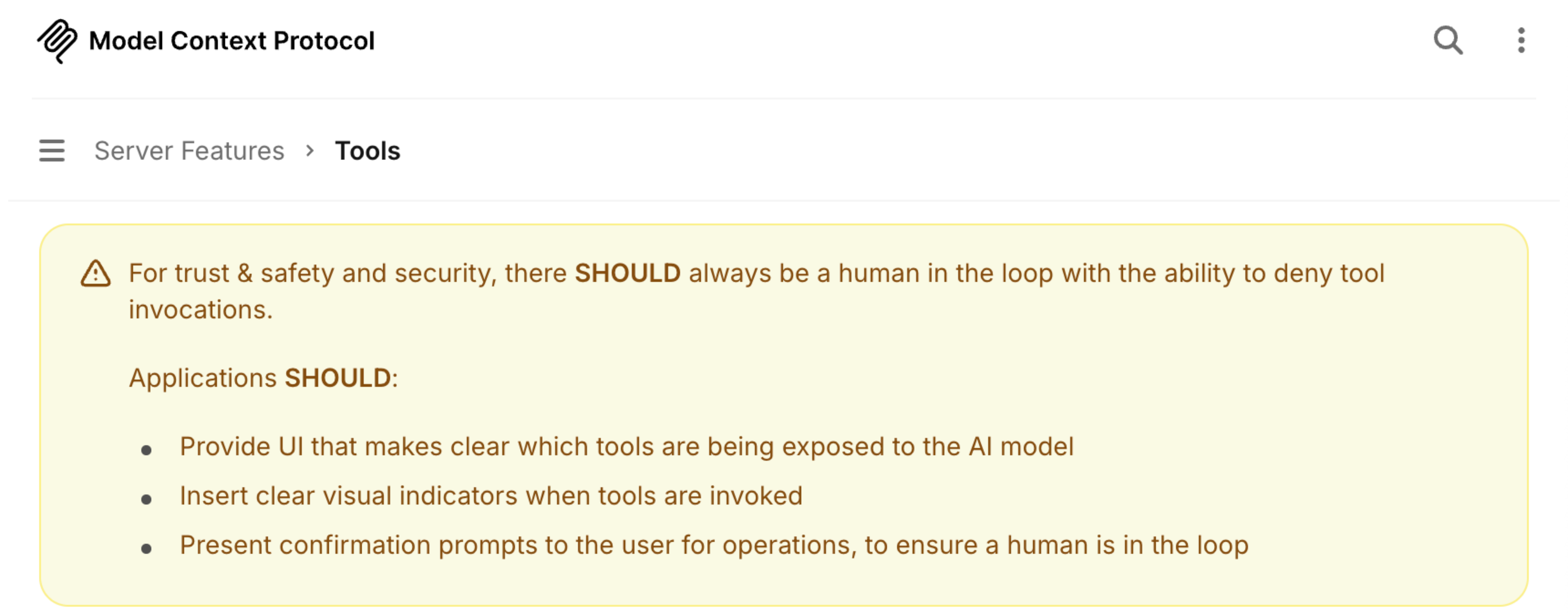
Ask Every Time ⌄ Skip Run ⏎



changkun-air ❐ dev ● 2 2.1.66

```
Claude Code v2.1.66
Opus 4.6 · Claude Max
~/dev/changkun.de/wallfacer

❯ can you read ~/dev/changkun.de/reseearch?

● Reading 1 file… (ctrl+o to expand)
  ⎿ ls ~/dev/changkun.de/reseearch

Bash command

  ls ~/dev/changkun.de/reseearch
  List contents of the reseearch directory

Do you want to proceed?
❯ 1. Yes
  2. Yes, allow reading from changkun.de/ from this project
  3. No

Esc to cancel · Tab to amend · ctrl+e to explain
```

Example: MCP and Tool Annotation

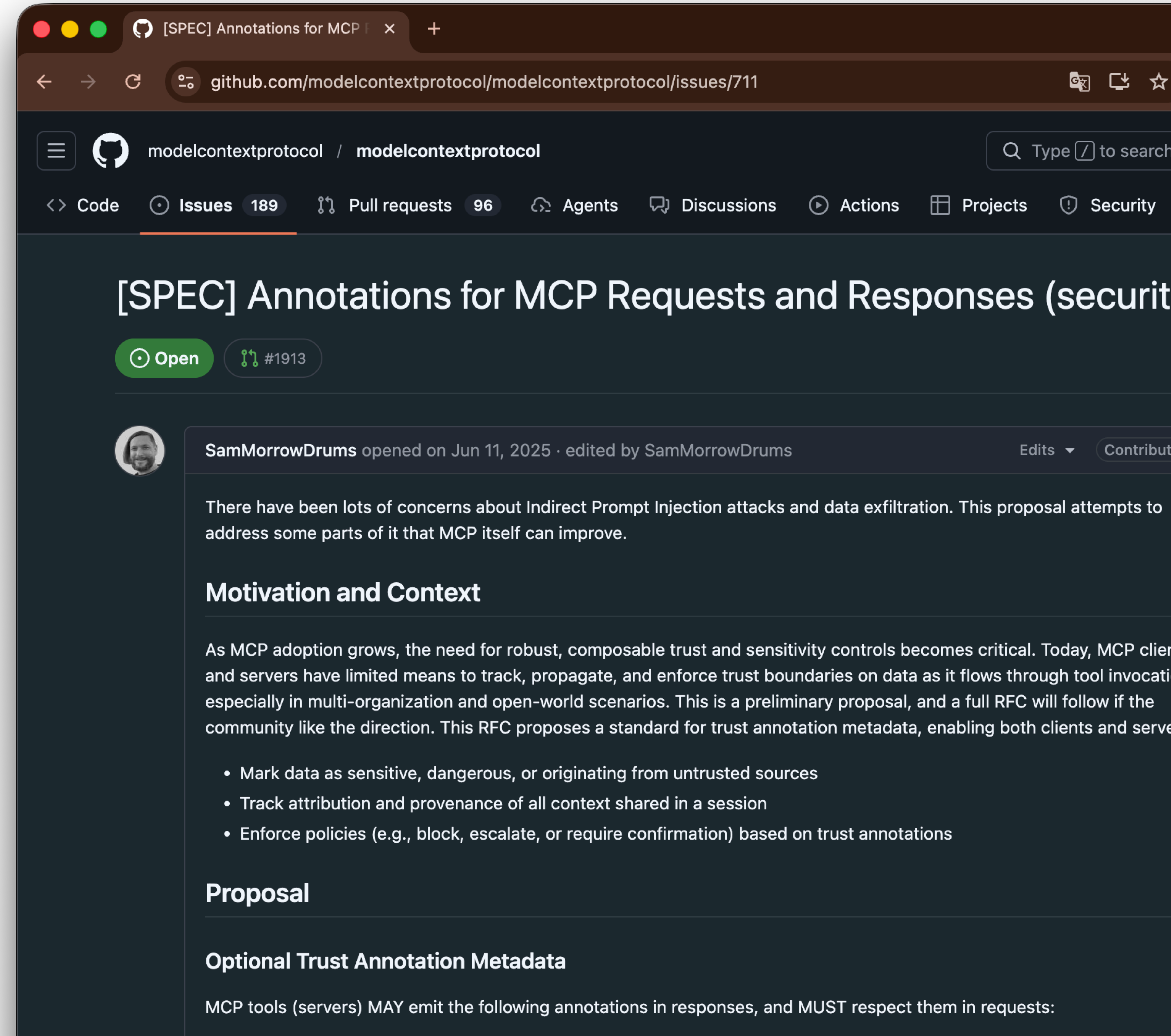

Model Context Protocol

Server Features › Tools

For trust & safety and security, there **SHOULD** always be a human in the loop with the ability to deny tool invocations.

Applications **SHOULD**:

- Provide UI that makes clear which tools are being exposed to the AI model
- Insert clear visual indicators when tools are invoked
- Present confirmation prompts to the user for operations, to ensure a human is in the loop


[SPEC] Annotations for MCP
github.com/modelcontextprotocol/modelcontextprotocol/issues/711
modelcontextprotocol / modelcontextprotocol
Type / to search
Code
Issues 189
Pull requests 96
Agents
Discussions
Actions
Projects
Security
Open
#1913
Edits
Contribut

[SPEC] Annotations for MCP Requests and Responses (securit

SamMorrowDrums opened on Jun 11, 2025 · edited by SamMorrowDrums

There have been lots of concerns about Indirect Prompt Injection attacks and data exfiltration. This proposal attempts to address some parts of it that MCP itself can improve.

Motivation and Context

As MCP adoption grows, the need for robust, composable trust and sensitivity controls becomes critical. Today, MCP clier and servers have limited means to track, propagate, and enforce trust boundaries on data as it flows through tool invocati especially in multi-organization and open-world scenarios. This is a preliminary proposal, and a full RFC will follow if the community like the direction. This RFC proposes a standard for trust annotation metadata, enabling both clients and serve

- Mark data as sensitive, dangerous, or originating from untrusted sources
- Track attribution and provenance of all context shared in a session
- Enforce policies (e.g., block, escalate, or require confirmation) based on trust annotations

Proposal

Optional Trust Annotation Metadata

MCP tools (servers) MAY emit the following annotations in responses, and MUST respect them in requests:

Per Action Consent Doesn’t Scale

There are three challenges:

- Catastrophic single actions (rm -rf /)
- Review large changes doesn’t scale
- Adversaries target oversight

No single mechanism resolves all three simultaneously.

The Real Threat: Confirmation Fatigue

- Overwhelming human-in-the-loop: Adversaries flood human reviewers with alerts or tasks to exploit cognitive overload
- “[…] fully autonomous AI agents should not be developed” cf. [Mitchell, et al. 2025]

What information granularity enables effective human confirmation of agent actions?

Part II: Wallfacer

Wallfacer A UX for for Sandboxed, Parallel, HITL Orchestration

github.com/changkun/wallfacer

Wallfacer   wallfacer main 8↑ Push

BACKLOG 3

+ New Task

backlog 15m 9m ago
Cloud or Native Deployment Optio...
investigate options where I can begin making this app running on cloud, or running locally as

backlog 15m 12m ago
Kanban Mobile Responsive Design
the current kanban UI doesn't seem to support mobile mode. Update the UI styling (layout,

backlog 15m 9h ago
Resilient Task State Recovery
when server is accidentally crashed or reboot, the inprogress tasks are

IN PROGRESS 5

in progress 15m 13m ago
Docker Container Monitoring Panel
in the system setting, i want to have a monitoring panel to mimic the behavior of "docker

in progress 15m 11m ago
Fix backlog prompt height
Prompt input field for the task card in "backlog" column has too little height, update and

in progress 15m 8m ago
Fix offline Kanban CSS dependen...
the current kandban css have some dependencies loaded from remote. this makes

in progress 15m 18m ago

WAITING 2

waiting 15m 22m ago
Debug GitHub Actions workflow
.github/workflows/release.yml doesn't seem to get triggered. do I need to configure
There's no `.github/workflows/release.yml`
`no changes`

waiting 30m 18h ago
Drag Drop File Attachment Resear...
as of now. the prompt input can only contain pure markdown text. I want to have it also
Both agents have finished — all results were already synthesized in
`no changes`

DONE 83 32,144 in / 1,001,089 out / $99.7326

done 15m 10m ago
Fix markdown bullet rendering
In UI, markdown rendering for prompt seems do not render bullet point very well and
The fix adds explicit `list-style-type` values to both `.prose-`

done 15m 14m ago
Resume Failed Timeout Tasks
waiting tasks can only be moved back to "backlog"? why is that the case? there should
Clean build. Here's what was changed and why:

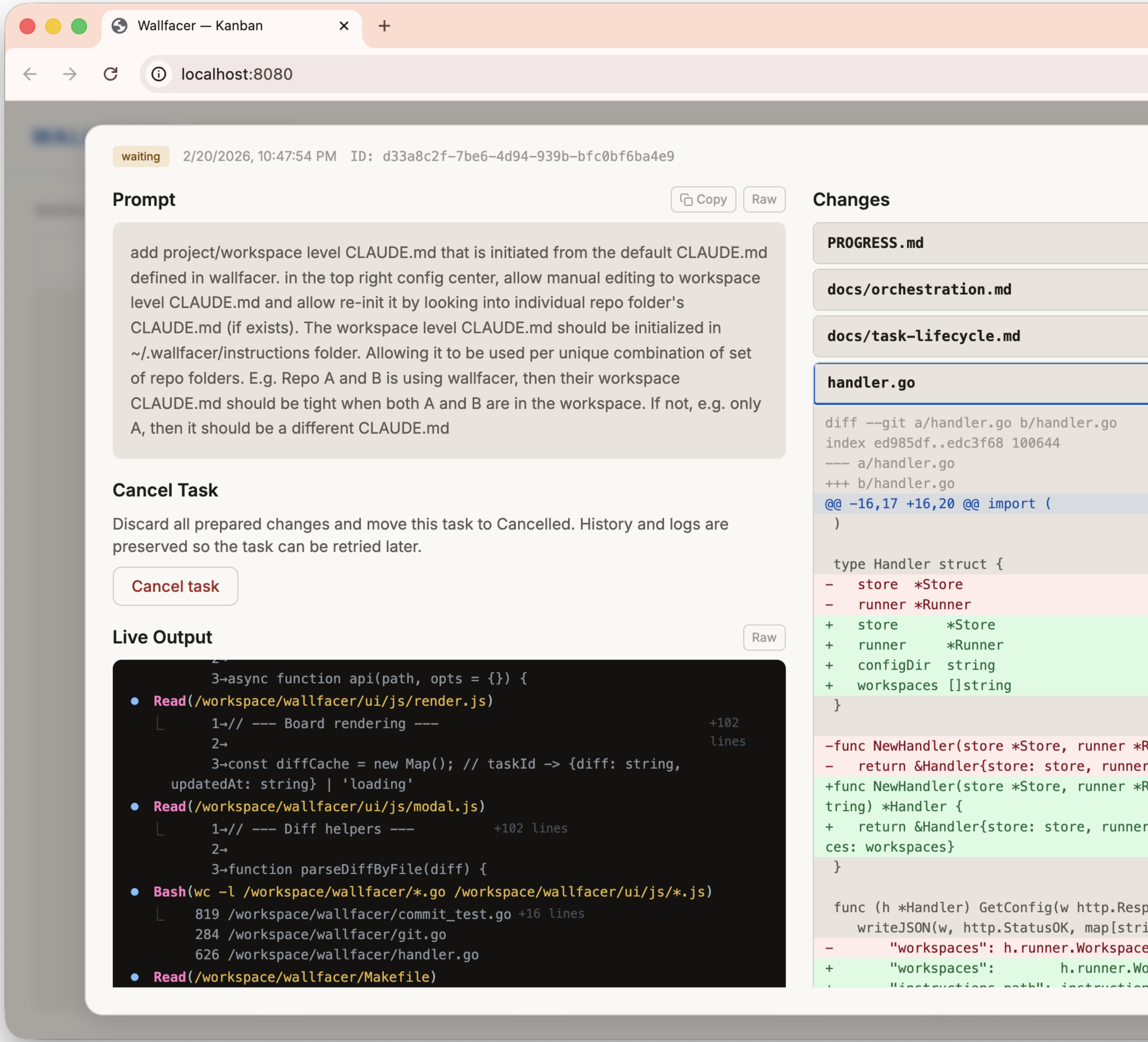
done 2h 4h ago
Git Util Test Coverage
write unit tests and make sure the test coverage up to 80%

CANCELLED 2

cancelled 15m 17h ago
Fix failed task resume
click resume button for the failed task seems to get this error

cancelled 15m 17h ago
Investigate Missing Git Changes
this task 38e607c3-68c8-46d7-adbe-316730c5ad46 was executed with some changes
container terminated: context deadline exceeded

Demo



Wallfacer — Kanban

localhost:8080

waiting 2/20/2026, 10:47:54 PM ID: d33a8c2f-7be6-4d94-939b-bfc0bf6ba4e9

Prompt

add project/workspace level CLAUDE.md that is initiated from the default CLAUDE.md defined in wallfacer. in the top right config center, allow manual editing to workspace level CLAUDE.md and allow re-init it by looking into individual repo folder's CLAUDE.md (if exists). The workspace level CLAUDE.md should be initialized in ~/.wallfacer/instructions folder. Allowing it to be used per unique combination of set of repo folders. E.g. Repo A and B is using wallfacer, then their workspace CLAUDE.md should be tight when both A and B are in the workspace. If not, e.g. only A, then it should be a different CLAUDE.md

Cancel Task

Discard all prepared changes and move this task to Cancelled. History and logs are preserved so the task can be retried later.

Live Output

```
        3→async function api(path, opts = {}) {
● Read(/workspace/wallfacer/ui/js/render.js)
  ⎿      1→// --- Board rendering ---                              +102
         2→                                                        lines
         3→const diffCache = new Map(); // taskId -> {diff: string,
   updatedAt: string} | 'loading'
● Read(/workspace/wallfacer/ui/js/modal.js)
  ⎿      1→// --- Diff helpers ---         +102 lines
         2→
         3→function parseDiffByFile(diff) {
● Bash(wc -l /workspace/wallfacer/*.go /workspace/wallfacer/ui/js/*.js)
  ⎿    819 /workspace/wallfacer/commit_test.go +16 lines
       284 /workspace/wallfacer/git.go
       626 /workspace/wallfacer/handler.go
● Read(/workspace/wallfacer/Makefile)
```

Changes

PROGRESS.md

docs/orchestration.md

docs/task-lifecycle.md

handler.go

```
diff --git a/handler.go b/handler.go
index ed985df..edc3f68 100644
--- a/handler.go
+++ b/handler.go
@@ -16,17 +16,20 @@ import (
 )

 type Handler struct {
-   store  *Store
-   runner *Runner
+   store      *Store
+   runner     *Runner
+   configDir  string
+   workspaces []string
 }

-func NewHandler(store *Store, runner *R
-   return &Handler{store: store, runner
+func NewHandler(store *Store, runner *R
tring) *Handler {
+   return &Handler{store: store, runner
ces: workspaces}
 }

 func (h *Handler) GetConfig(w http.Resp
   writeJSON(w, http.StatusOK, map[stri
-       "workspaces": h.runner.Workspace
+       "workspaces":         h.runner.Wo
```

The Journey: Maximize Agentic Coding Throughput

- Step 1: Switch from Cursor Agent to Claude Code
  - Too big one shot code changes
  - Too many reviews
  - Complex UI

The Journey: Maximize Agentic Coding Throughput

- Step 1: Switch from Cursor Agent to Claude Code
- Step 2: Use a Sandbox
  - `claude --dangerously-skip-permissions`

The Journey: Maximize Agentic Coding Throughput

- Step 1: Switch from Cursor Agent to Claude Code
- Step 2: Use a Sandbox to Isolate Fully Autonomous Actions
  - `claude --dangerously-skip-permissions`
- Step 3: Use "Ralph Loop” to keep work

The Journey: Maximize Agentic Coding Throughput

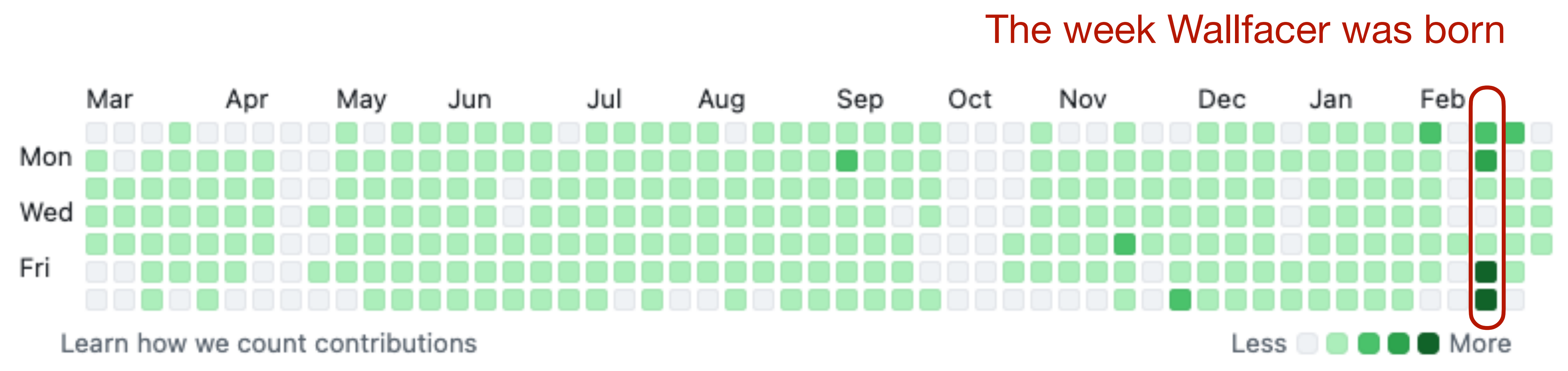
- Step 1: Switch from Cursor Agent to Claude Code
- Step 2: Use a Sandbox to Isolate Fully Autonomous Actions
  - `claude --dangerously-skip-permissions`
- Step 3: Use "Ralph Loop” to keep work
- Step 4: Use Git worktree for work parallelization
-

The Journey: Maximize Agentic Coding Throughput

- Step 1: Switch from Cursor Agent to Claude Code
- Step 2: Use a Sandbox to Isolate Fully Autonomous Actions
  - `claude --dangerously-skip-permissions`
- Step 3: Use "Ralph Loop” to keep work
- Step 4: Use Git worktree for work parallelization
- Step 5: Let Wallfacer develop Wallfacer

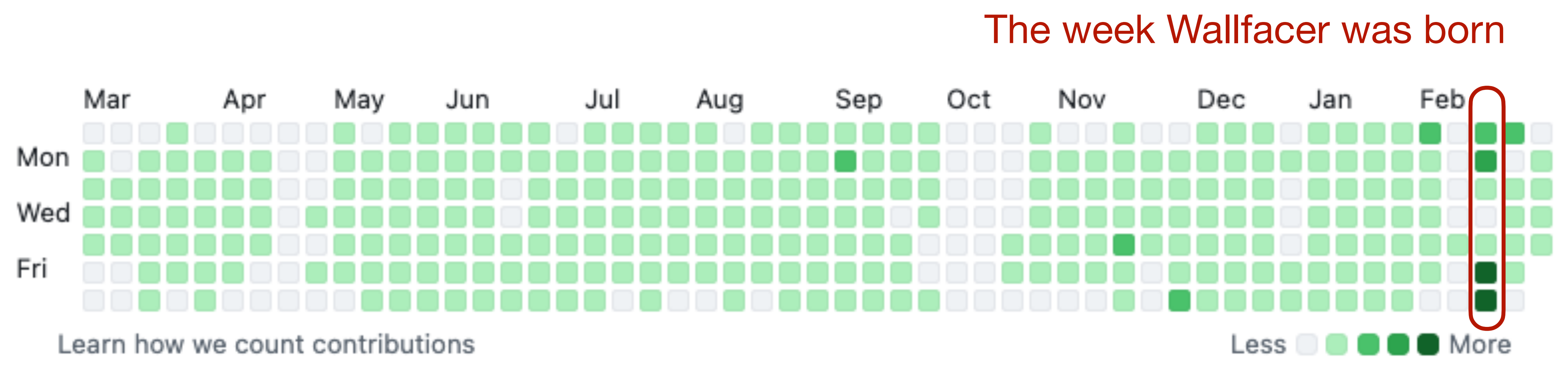
The Journey: Maximize Agentic Coding Throughput

- Step 1: Switch from Cursor Agent to Claude Code
- Step 2: Use a Sandbox to Isolate Fully Autonomous Actions
  - `claude --dangerously-skip-permissions`
- Step 3: Use "Ralph Loop” to keep work
- Step 4: Use Git worktree for work parallelization
- Step 5: Let Wallfacer develop Wallfacer



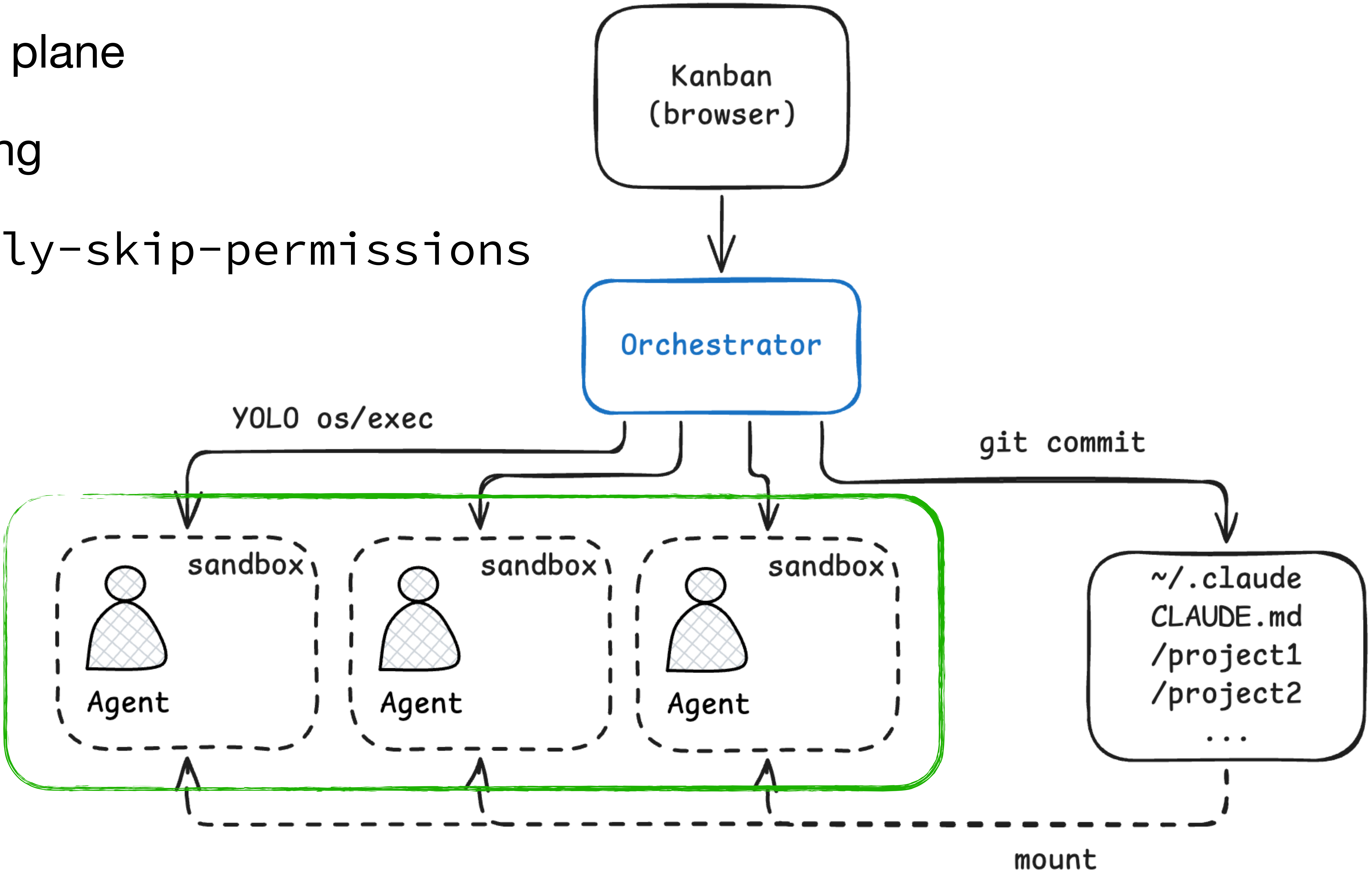
The Journey: Maximize Agentic Coding Throughput

- Step 1: Switch from Cursor Agent to Claude Code
- Step 2: Use a Sandbox to Isolate Fully Autonomous Actions
  - `claude --dangerously-skip-permissions`
- Step 3: Use "Ralph Loop" to keep work
- Step 4: Use Git worktree for work parallelization
- Step 5: Let Wallfacer develop Wallfacer
- Step 6: Multi-workspace and CLAUDE.md
- …



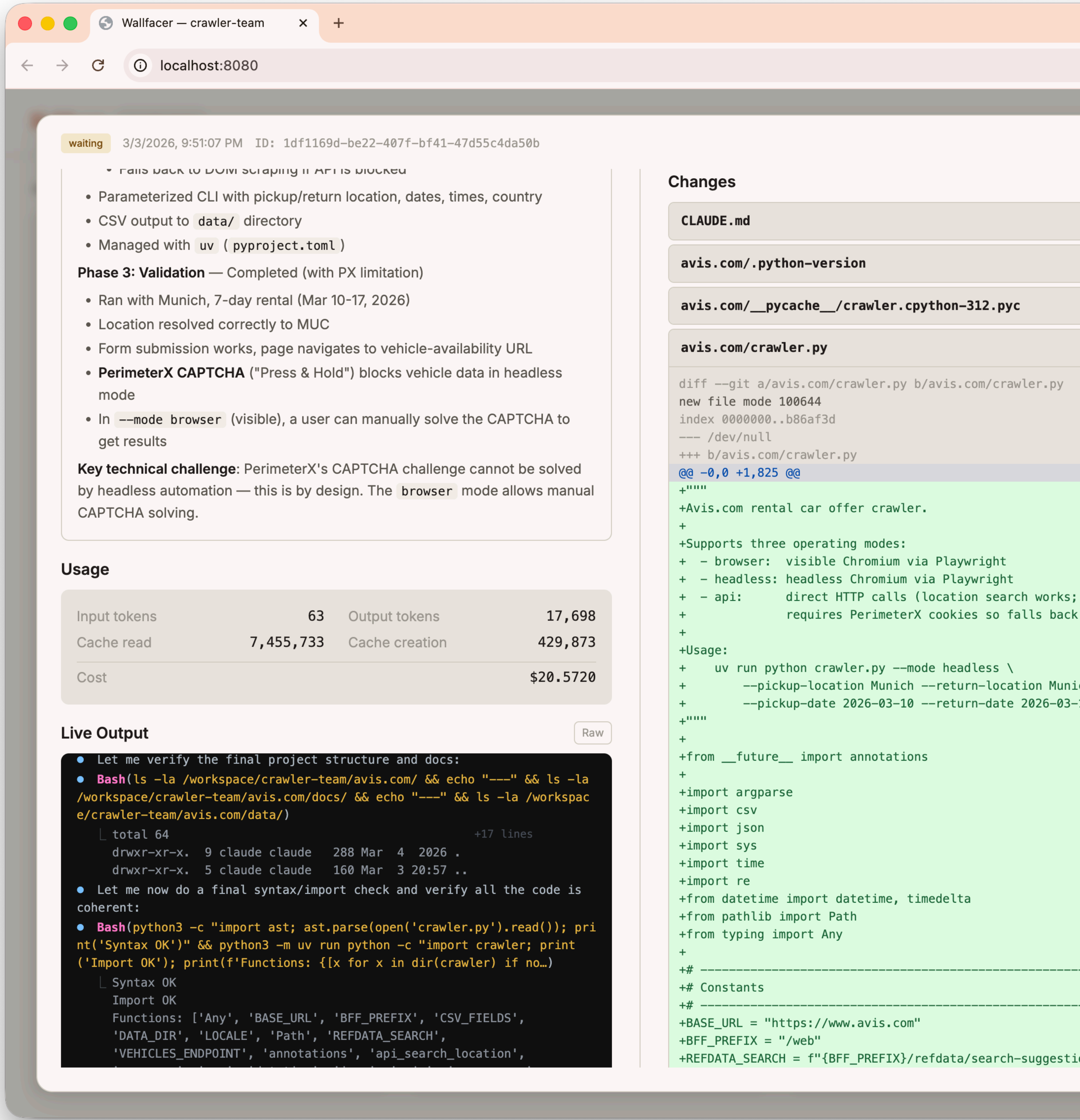
Wallfacer Architecture

- Control plane vs. data plane
- Isolation and monitoring
- YOLO: `--dangerously-skip-permissions`



Design Choices

- Task lifecycle: TODO -> In Progress -> Waiting -> Done (Cancelled)
- Oversight mechanism:
  - Level 1: High-level intent summary
  - Level 2: Execution traces
  - Level 3: Change-diff
- Automate the coding but not the deploying


Wallfacer — crawler-team

localhost:8080

waiting 3/3/2026, 9:51:07 PM ID: 1df1169d-be22-407f-bf41-47d55c4da50b

  - Falls back to DOM scraping if API is blocked
- Parameterized CLI with pickup/return location, dates, times, country
- CSV output to `data/` directory
- Managed with `uv` (`pyproject.toml`)

Phase 3: Validation — Completed (with PX limitation)

- Ran with Munich, 7-day rental (Mar 10-17, 2026)
- Location resolved correctly to MUC
- Form submission works, page navigates to vehicle-availability URL
- **PerimeterX CAPTCHA** ("Press & Hold") blocks vehicle data in headless mode
- In `--mode browser` (visible), a user can manually solve the CAPTCHA to get results

Key technical challenge: PerimeterX's CAPTCHA challenge cannot be solved by headless automation — this is by design. The `browser` mode allows manual CAPTCHA solving.

Usage

Input tokens	63	Output tokens	17,698
Cache read	7,455,733	Cache creation	429,873
Cost			$20.5720

Live Output

```
● Let me verify the final project structure and docs:
● Bash(ls -la /workspace/crawler-team/avis.com/ && echo "---" && ls -la
/workspace/crawler-team/avis.com/docs/ && echo "---" && ls -la /workspac
e/crawler-team/avis.com/data/)
  └ total 64                                          +17 lines
    drwxr-xr-x.  9 claude claude   288 Mar  4  2026 .
    drwxr-xr-x.  5 claude claude   160 Mar  3 20:57 ..
● Let me now do a final syntax/import check and verify all the code is
coherent:
● Bash(python3 -c "import ast; ast.parse(open('crawler.py').read()); pri
nt('Syntax OK')" && python3 -m uv run python -c "import crawler; print
('Import OK'); print(f'Functions: {[x for x in dir(crawler) if no…)
  └ Syntax OK
    Import OK
    Functions: ['Any', 'BASE_URL', 'BFF_PREFIX', 'CSV_FIELDS',
    'DATA_DIR', 'LOCALE', 'Path', 'REFDATA_SEARCH',
    'VEHICLES_ENDPOINT', 'annotations', 'api_search_location',
```

Changes

`CLAUDE.md`

`avis.com/.python-version`

`avis.com/__pycache__/crawler.cpython-312.pyc`

`avis.com/crawler.py`

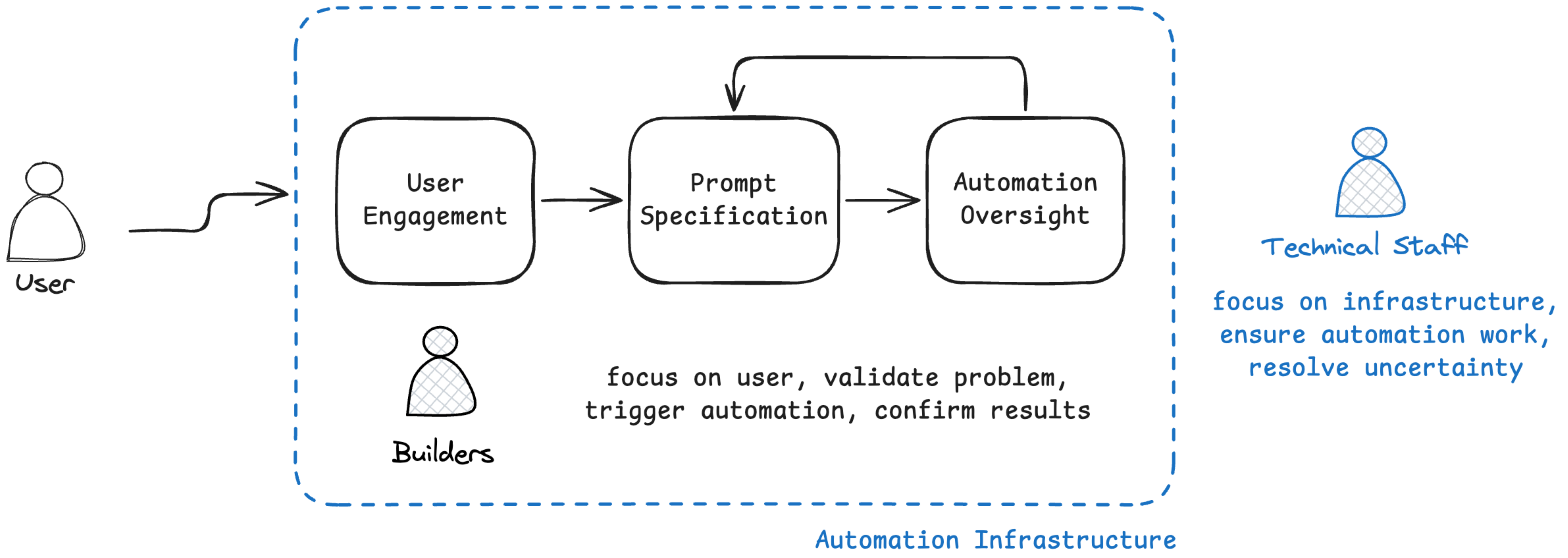
```
diff --git a/avis.com/crawler.py b/avis.com/crawler.py
new file mode 100644
index 0000000..b86af3d
--- /dev/null
+++ b/avis.com/crawler.py
@@ -0,0 +1,825 @@
+"""
+Avis.com rental car offer crawler.
+
+Supports three operating modes:
+  - browser:  visible Chromium via Playwright
+  - headless: headless Chromium via Playwright
+  - api:      direct HTTP calls (location search works;
+             requires PerimeterX cookies so falls back
+
+Usage:
+    uv run python crawler.py --mode headless \
+        --pickup-location Munich --return-location Muni
+        --pickup-date 2026-03-10 --return-date 2026-03-
+"""
+
+from __future__ import annotations
+
+import argparse
+import csv
+import json
+import sys
+import time
+import re
+from datetime import datetime, timedelta
+from pathlib import Path
+from typing import Any
+
+# ---------------------------------------------------------------
+# Constants
+# ---------------------------------------------------------------
+BASE_URL = "https://www.avis.com"
+BFF_PREFIX = "/web"
+REFDATA_SEARCH = f"{BFF_PREFIX}/refdata/search-suggesti
```

What Will Human (Engineer) Actually Do?

- Using expertise as risk control
- Knowing when and what to interrupt
- Engineering context for the agents

Shifting from Execution to Judgement

- The per-keystroke style of programming is gone
- With the current capability, “builders” role became more dominant, where they apply judgements heavily to decide what to build and what to not build hence ensure system operates continuously


User
User Engagement
Prompt Specification
Automation Oversight
Builders
focus on user, validate problem, trigger automation, confirm results
Technical Staff
focus on infrastructure, ensure automation work, resolve uncertainty
Automation Infrastructure

Part III: Reflections

What to Automate and What Not To

Automate

- Implementation & boilerplate
- Migration scripts
- Test scaffolding
- Anything with clear spec and verifiable results

Don’t Automate

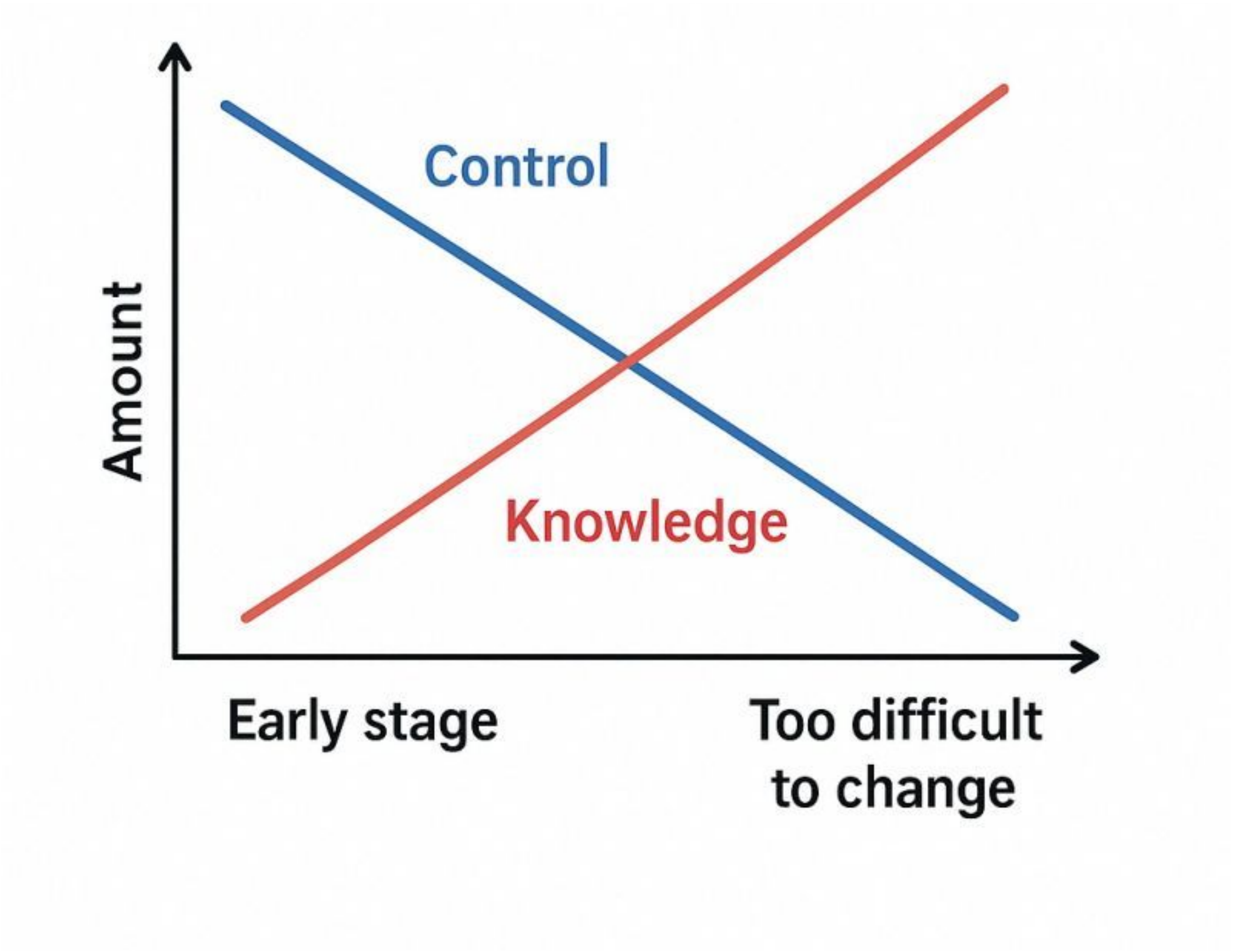
- Architectural judgement
- Review of irreversible actions
- Cross-system integration decisions
- Anything requiring organizational context

New Tradeoffs baked with Undertainty

- Automation vs. Steering
- Delegation vs. Hands-on

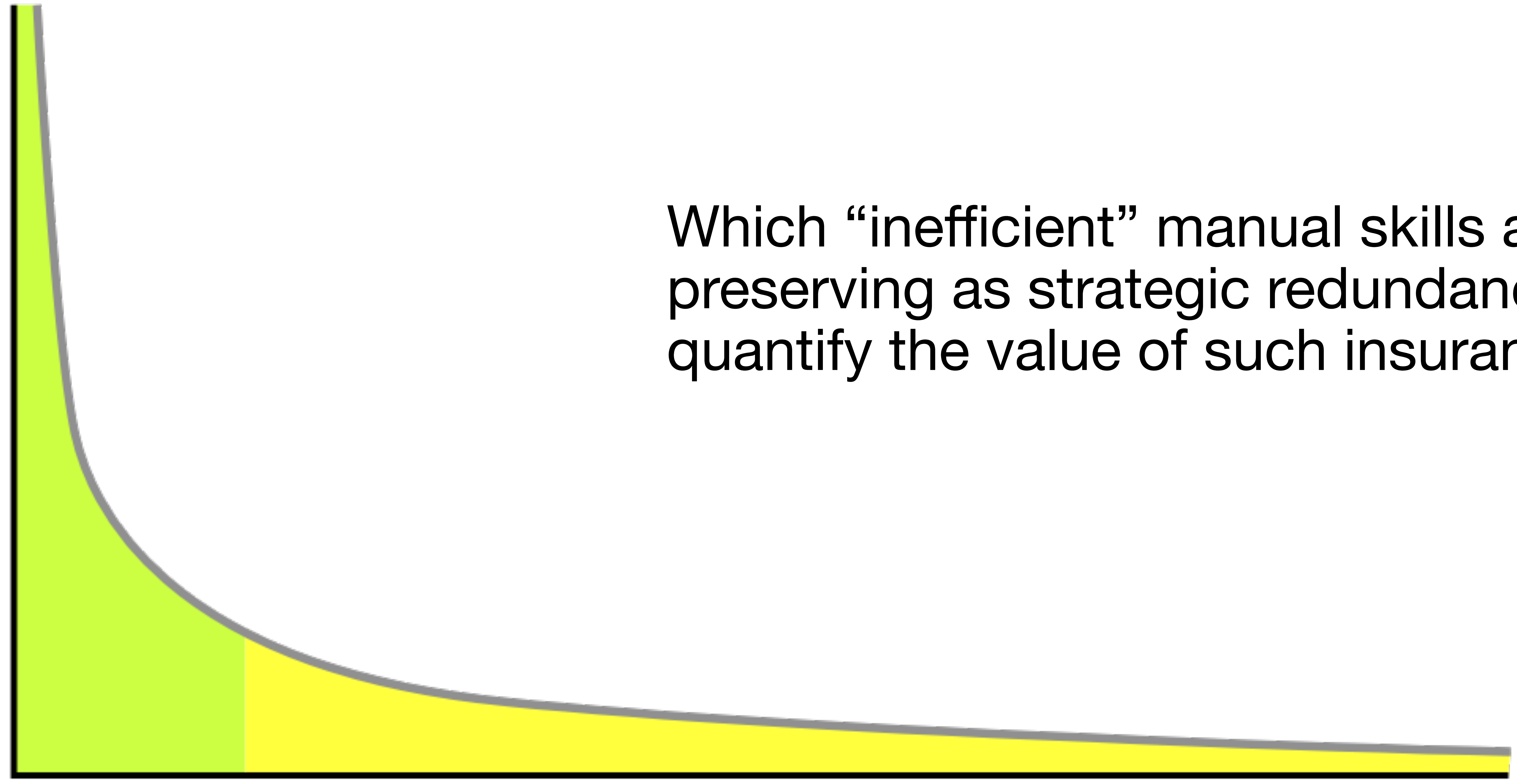
The Collingridge Dilemma [Colligridge 1980]

- The dilemma: Technology is easy to modify when you don't yet know the problems. By the time the problems appear, modification is expensive



Cognitive Offloading and Adaptation Curve

Collapse of Complex Societies: the maintenance costs of complex systems may ultimately exceed their marginal benefits, leading to simplification and reversion

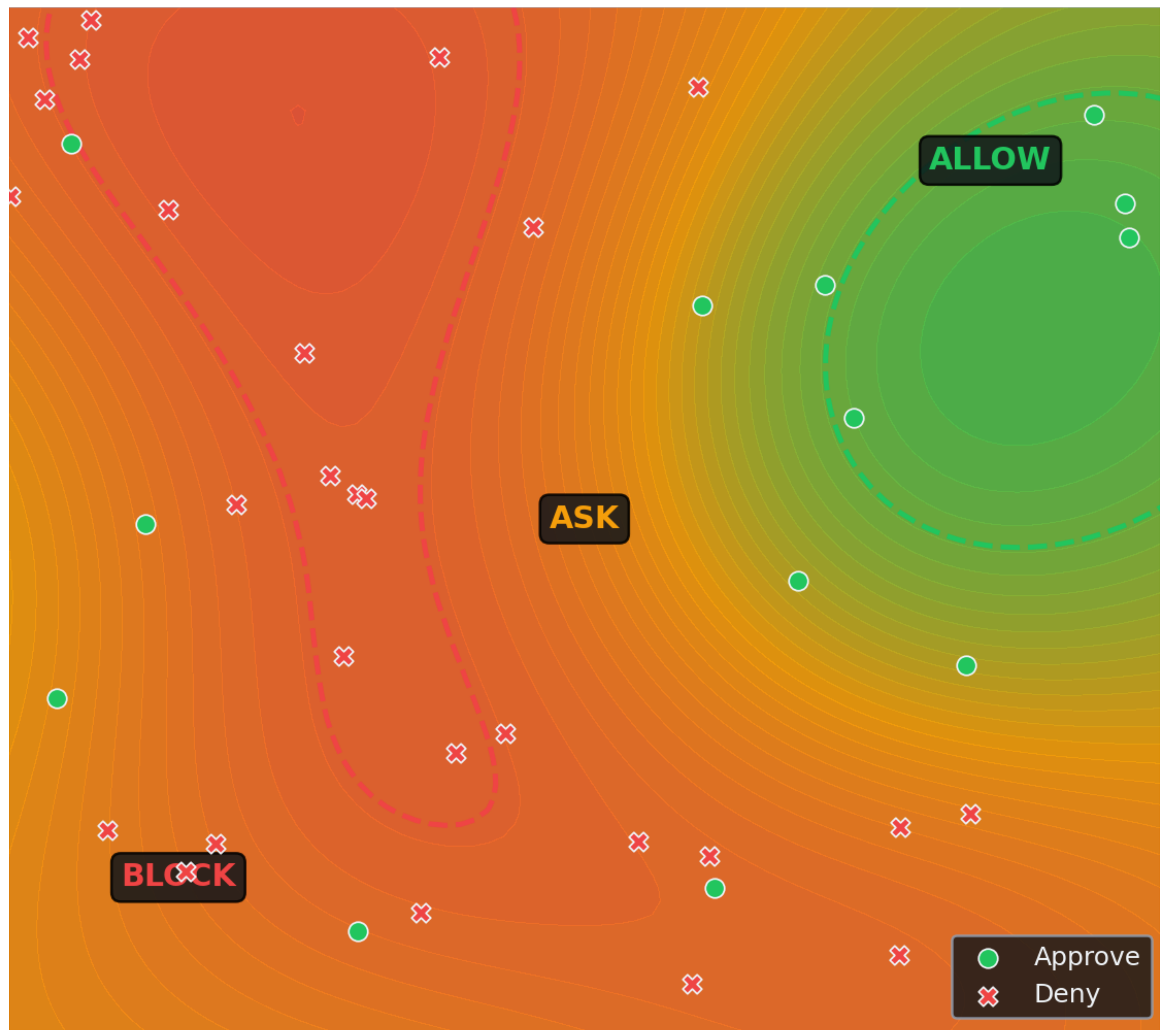


Which “inefficient” manual skills are worth deliberately preserving as strategic redundancy? How should we quantify the value of such insurance?

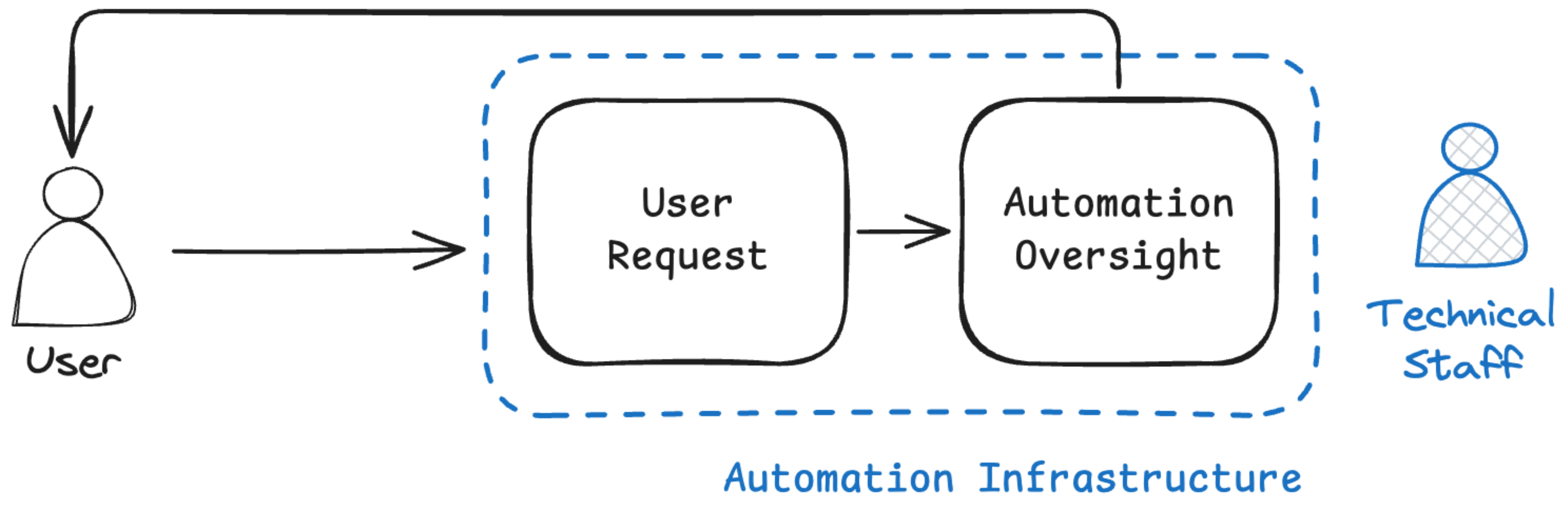
* Kosmyna et al. Your Brain on ChatGPT: Accumulation of Cognitive Debt when Using an AI Assistant for Essay Writing Task. arXiv preprint. https://arxiv.org/abs/2506.08872

Progressive Autonomy

- Metacognition adaptation
- When judgements fade out
- Trust calibration is a preference learning problem: learn P(approve | action, context)



Ou. Trust calibration as preference learning. Preprint. 2026.
https://changkun.de/research/papers/ou2026trustcalib.pdf



The Identity Question

Lex Fridman (03:04:40) […] I never thought that the thing I love doing would be the thing that gets replaced. You hear these stories about things like the steam engine. I've spent thousands of hours poring over code, pouring my heart and soul into it. Some of my most painful and happiest moments were alone behind a screen. I was an Emacs person for a long time—man, Emacs. And there's an identity there, there's meaning. When I walk about the world, I don't say it out loud, but I think of myself as a programmer. And to have that possibly change in a matter of months…

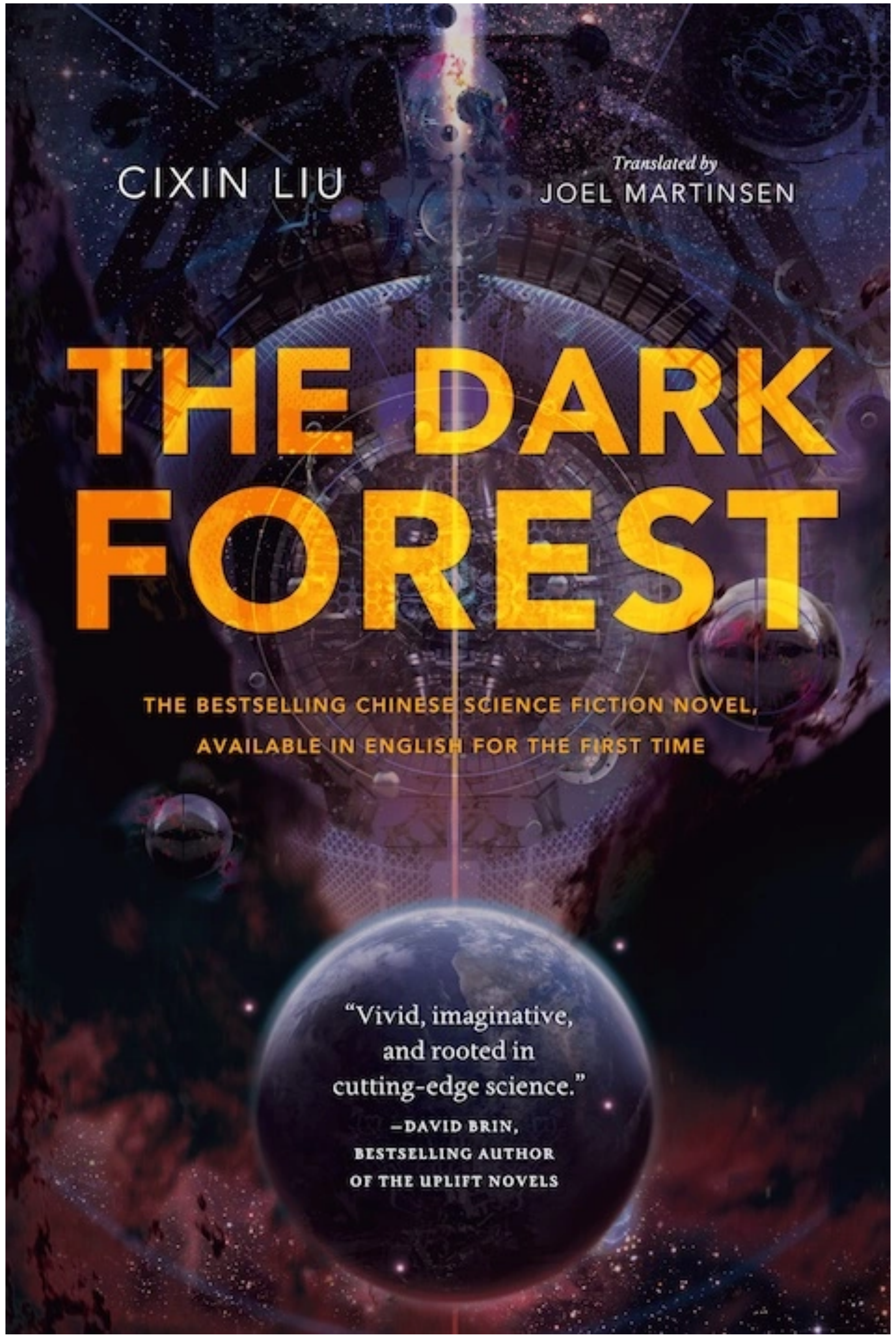
https://youtu.be/YFjfBk8HI5o?si=kpCuIKaaAgfC6l47&t=11080

But this is a life long question :)

* Klein, S. B., & Nichols, S. (2012). Memory and the sense of personal identity. Mind, 121(483), 677-702.

Problems of Other Minds

- In *Three Body Problem*, Trisolarans deploy sophons that can observe everything human can do (i.e. pure behaviorism):
  - All actions are observable
  - All communication is monitored
- "Wallfacer" from *Three Body Problem II: The Dark Forest:* Strategists whose plans are opaque to observers. Each human works independently on an opaque task; only results are inspected.


CIXIN LIU
Translated by
JOEL MARTINSEN
THE DARK FOREST
THE BESTSELLING CHINESE SCIENCE FICTION NOVEL, AVAILABLE IN ENGLISH FOR THE FIRST TIME
"Vivid, imaginative, and rooted in cutting-edge science."
—DAVID BRIN, BESTSELLING AUTHOR OF THE UPLIFT NOVELS