// claude-as-coworker

My new coworker Claude

a way to bring Claude Code into your dev team

// whoami

~/about-me

```
$ whoami
Florian Wenzel
company — Bits GmbH · mybits.de
role — software architect / developer / generalist
focus — platform engineering · Kubernetes · AI

# I'm just very bad at saying no.
# probably why I am giving this talk right now.
$
```

// PART 1 — we all start here

Claude as a tool

```
❯ add a NotificationService that emails a user when their order ships — retry failed sends

● Write(src/notifications/notification.service.ts)
  ⎿ Wrote 10 lines
export class NotificationService {
  constructor(private users: Users, private mail: Mailer,
              private q: RetryQueue) {}

  async onOrderShipped(order: Order) {
    const user = await this.users.byId(order.userId);
    await this.q.run(() =>
      this.mail.send(user.email, shippedEmail(order)));
  }
}
```

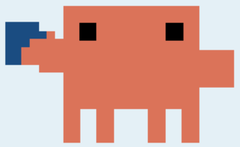
// PART 1 — the turn

…but it feels like more

colleague

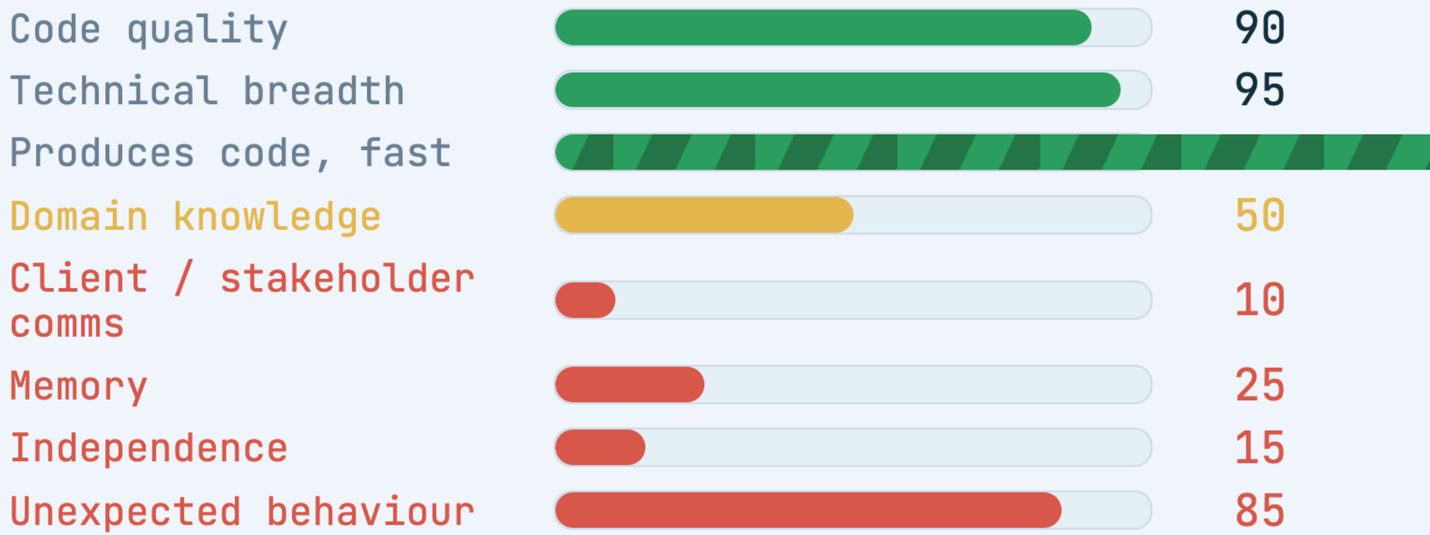
// PART 1 — meet the coworker

Meet our new coworker, Claude.



Claude

// junior-senior hybrid



mascot rig: Ayotomiwa Wale-Durojaye · ayotomcs.me

// PART 1 — placement

What do we trust them with?

● Owns it

- Implement → open a PR
- Fix failing CI
- Resolve merge conflicts
- Address review feedback

◐ Assists

- Review a PR

○ Different colleague (human)

- Requirements & client talk
- The decision to ship
- Keep prod up (ops)

“Different colleague” ≠ “no Claude”

// PART 1 — access control

Access control

- ✓ Its own **account**
- ✓ **Restricted** to the permissions it needs
- ✓ **Open PRs**
- ✓ **Read** the code, CI & the docs it needs (SonarQube, internal docs)
- ✗ Prod & deploy keys
- ✗ Secrets / customer data
- ✗ Push directly to master

Worst case ≈ **a bad PR.**

// PART 2 — in my case

Enough theory.

- ✓ **2× Mac mini**
- ✓ each on its **own GitHub account**
- ✓ host set up with **what the work needs**
- ✓ **Claude Code** on each → ● ready, waiting...

// PART 2 — getting it to actually work

They won't do anything just sitting there.

So I tried the loop command:

```
claude · /loop

❯ /loop 5m "check GitHub —
  an issue assigned to me?  → implement it
  my PR with red CI?        → fix it
  a PR waiting on review?   → review it"
```

...and it worked!

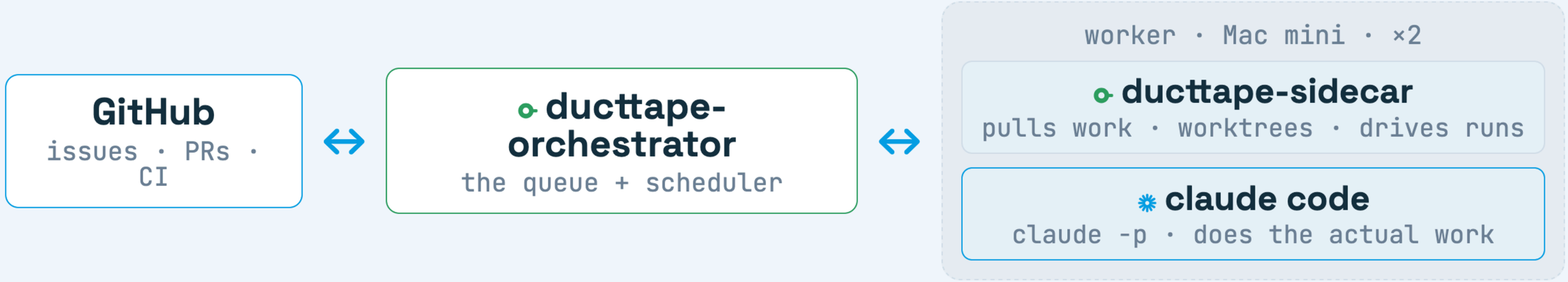
...until it didn't.

It broke too frequently, so I had Claude write me some **ducttape**.

// PART 2 — architecture

ducttape

- **ducttape** is what attaches **GitHub** to my two Claude coworkers.



- each task runs in its own **git worktree**
- it **resumes its Claude session** for follow-ups

// PART 2 — see it in action

Let’s see it in action.

claude · local

copy

```
❯ open an issue on NovyraHQ/petal, label it agent:ready:
  "on the marketing page, change the CTA button that reads
'Sign up' to 'Start journaling'"
```

// PART 2 — off-script

Sometimes they go off-script.

It opened an **Ops issue** because CI was flaky.

Main takeaways

// DX is for them too

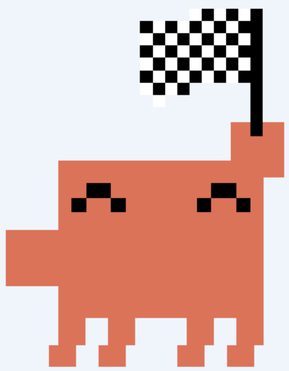
- Trivial **local setup**
- Mock external APIs
- Strong **e2e coverage**
- CI is the judge

// place & sandbox

- **Sandbox** hard
- ...but feed it **context**
- Play to strengths, design around weaknesses
- **Write your own ducttape**

// the money

- Now: **MAX flatrate** covers it
- **Jun 15** → `claude -p` goes metered



// fin

Questions?

mybits.de
Bits GmbH

Thanks!