

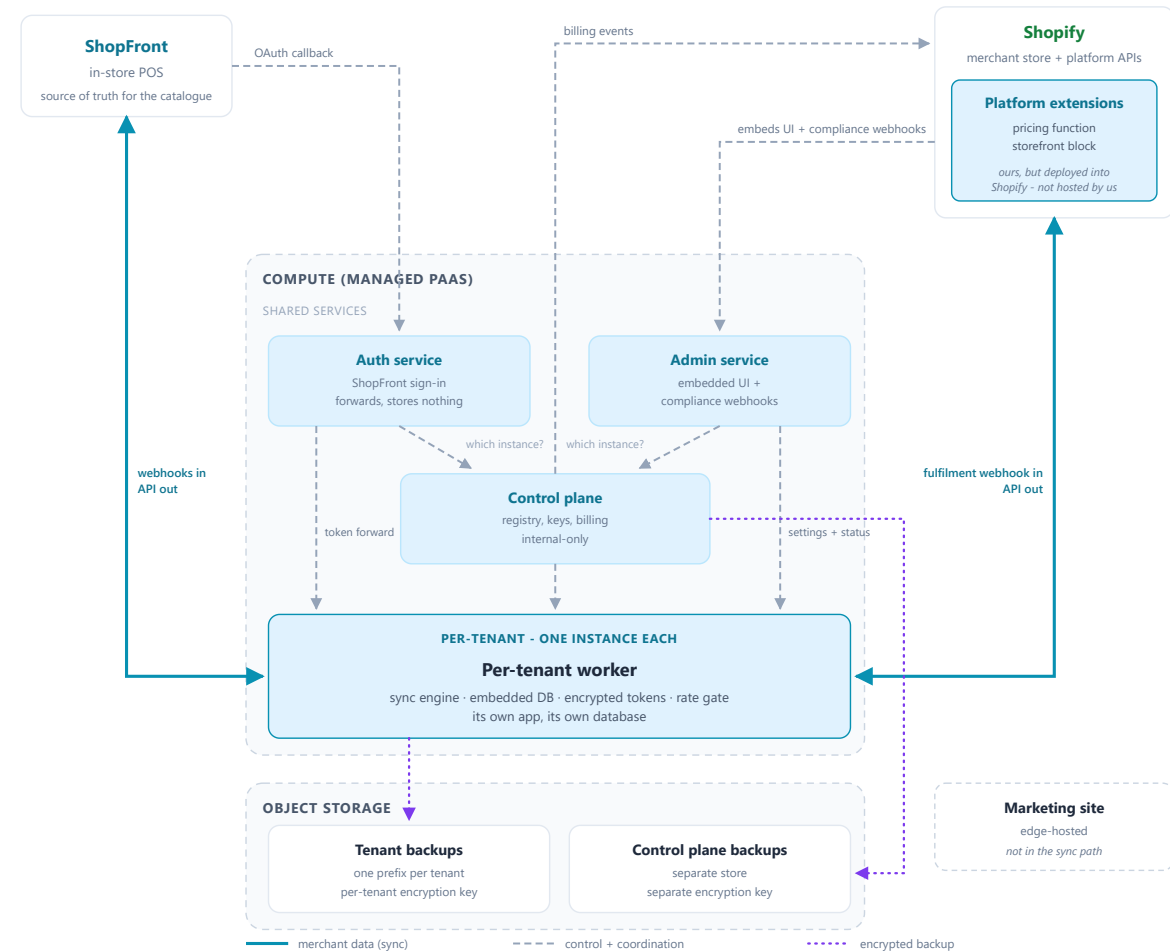
2026-08-11

AI-Enabled SDLC: From Startup to Global Enterprise

Stephen Mc Gowan

Founder, TrolleyRelay · AI Architect

FROM THE SOLUTION DESIGN · HOW IT'S PUT TOGETHER



● Delivery received

Pale Ale · +20

✓ SYNCED

● Online order #1042

Pale Ale ×2 · \$13

✓ SYNCED

● Sold at the till

Pomerol Merlot ×1 · \$22

✓ SYNCED

● New product added

Yuzu Soda · \$6

● Online order #1043

IPA ×6 · \$39

● Sold at the till

Ginger Beer ×4 · \$20

● Price ch

Prosecco · n

18 weeks. One human. One AI agent as primary implementer.

PR #1 → #597

PULL REQUESTS · 579 MERGED

~1,300

COMMITTS AUTHORED · 602 ON MAIN AFTER
SQUASH

74k / 78k

LINES OF CODE / LINES OF TESTS

372 / 169

BACKLOG ITEMS CREATED / CLOSED

21

ARCHITECTURE DECISION RECORDS

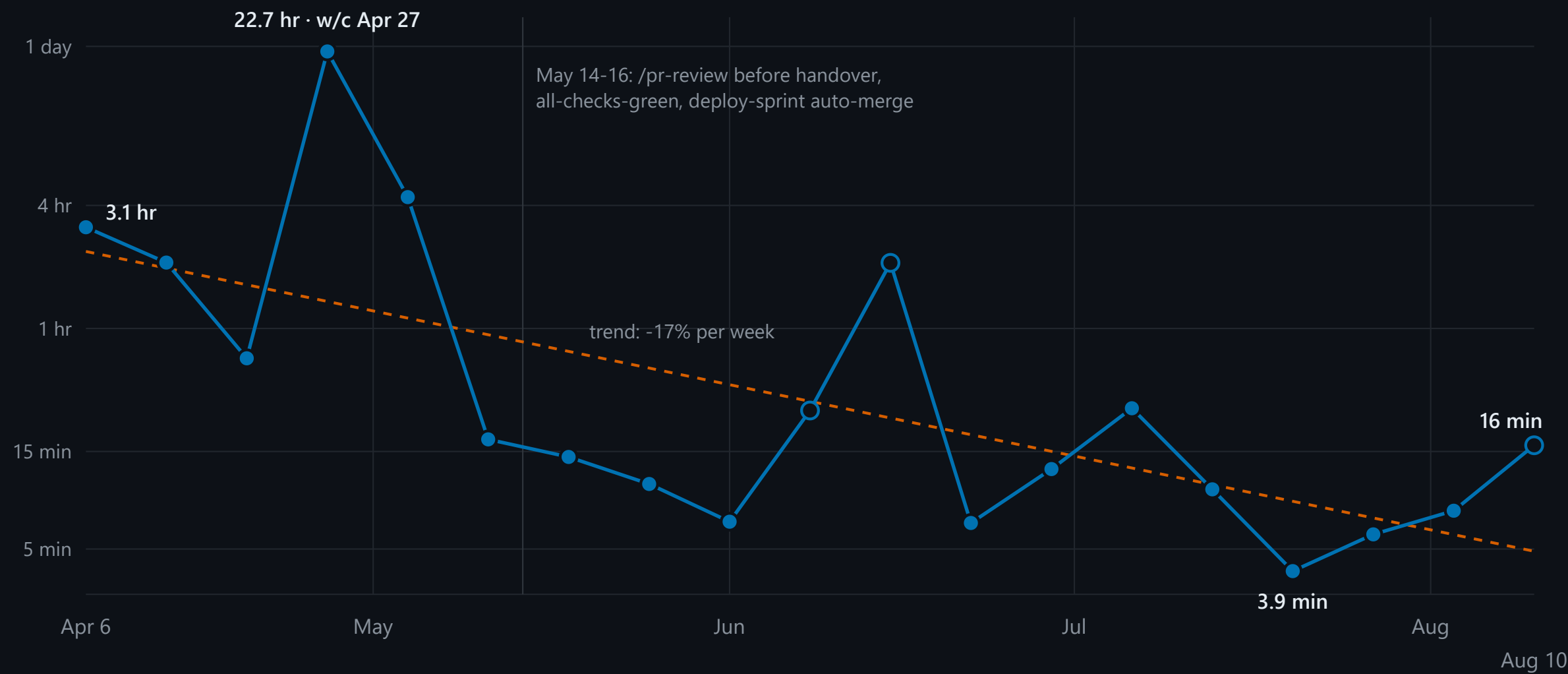
24

SPRINT CLOSE-OUTS

A real product, in pilot with its first customer. Not a demo repo.

More gates, faster merges: PR open-to-merge time by week

Weekly median across 579 merged pull requests, log scale (2026-04-08 to 2026-08-10)



Hollow markers: weeks with fewer than 10 merged PRs (w/c Jun 8: 7, w/c Jun 15: 2, Aug 10 partial week: 5).
You can't install an AI-SDLC, article 0. Source: GitHub PR opened/merged timestamps, verified 2026-08-11.

FIRST, WHAT THIS IS NOT

Vibecoding has a time and a place.

This is not it.

Same agent. Same model. Different process.

The two worlds

World 1: a one-person SaaS

TrolleyRelay. Total autonomy over process. The dial goes as high as I dare, and I watch what breaks.

World 2: a global enterprise

Regulated industry. The same questions arrive with risk and compliance attached.

One is my lab. The other is why the lab exists.

"Can I have your process?"

DAY 1: 15 LINES, TWO HEADINGS

```
# Backlog

This is a project to implement a 2-way integration
between ShopFront POS and Shopify.

Ideas, issues, and pre-development plans live in
`.claude/backlog/`. Use this as the source of truth...

## Workflow

- When the user mentions a new idea or issue, check
  if it already exists in the backlog...
- When starting work on a backlog item, read its
  file for context before planning.
- When an item is implemented, mark it as done...
```

TODAY: 165 LINES, 17 SECTIONS · COMPRESSED THIS MORNING FROM 330

Repo layout

Testing (coverage ratchet)

Coding conventions

PR review and merge gate

- Categorical handling of findings

- CI gate (every check green)

- Human-review gates: secret / ADR / pricing

- Mentions and notifications

- Order of operations on a PR

- Attribution and audit cutover

Deployment

- The Shopify app: a fifth deployable

Workflow

- One session, one worktree

- When the agent may stop

- Grill gate enforcement (hard rule)

- Backlog lifecycle (grill-gated)

You can copy the file. You cannot copy what made it.

THE THESIS

You can't install an AI-SDLC.

You grow one.

Notice the failure

→

Write the rule

→

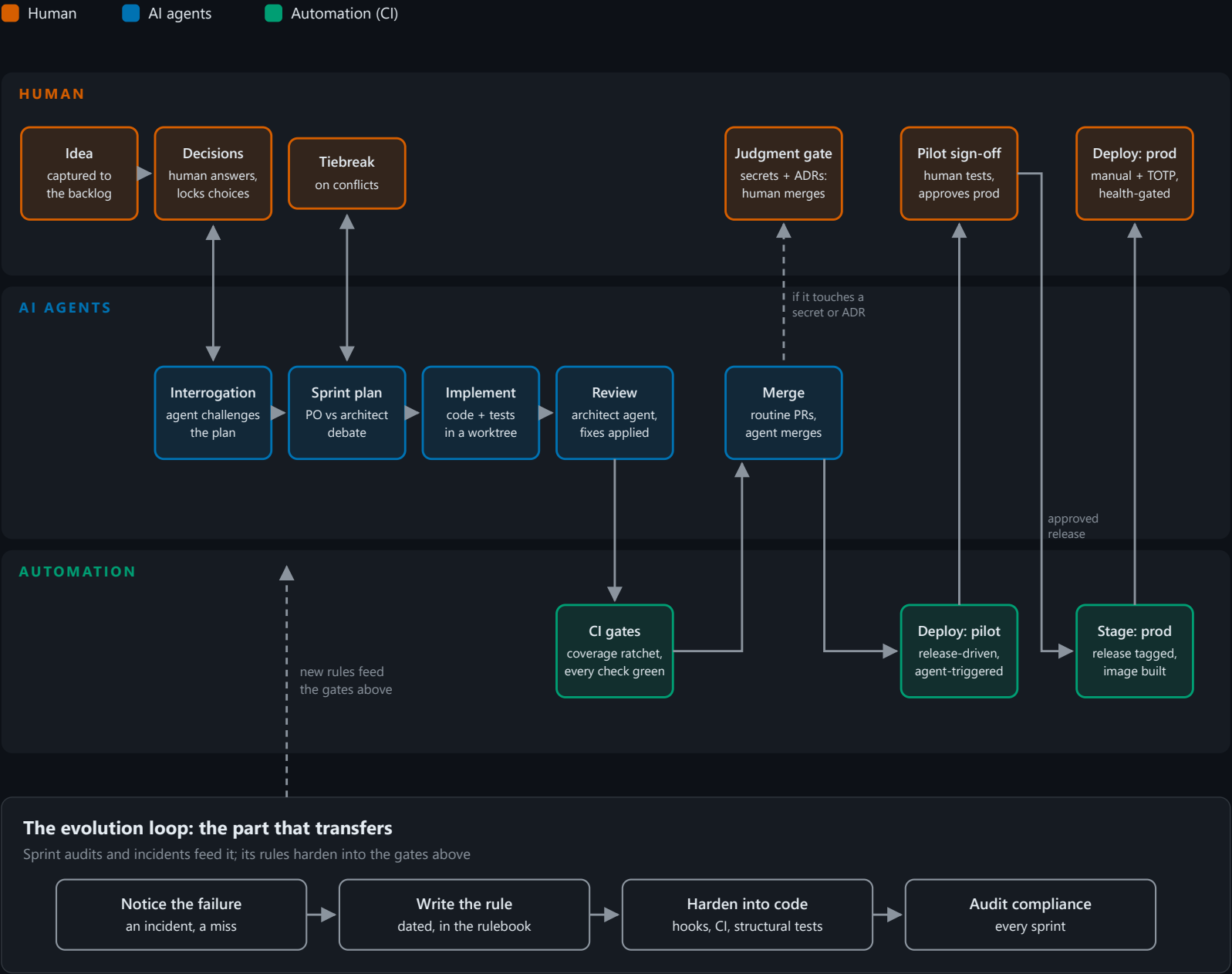
Harden into code

→

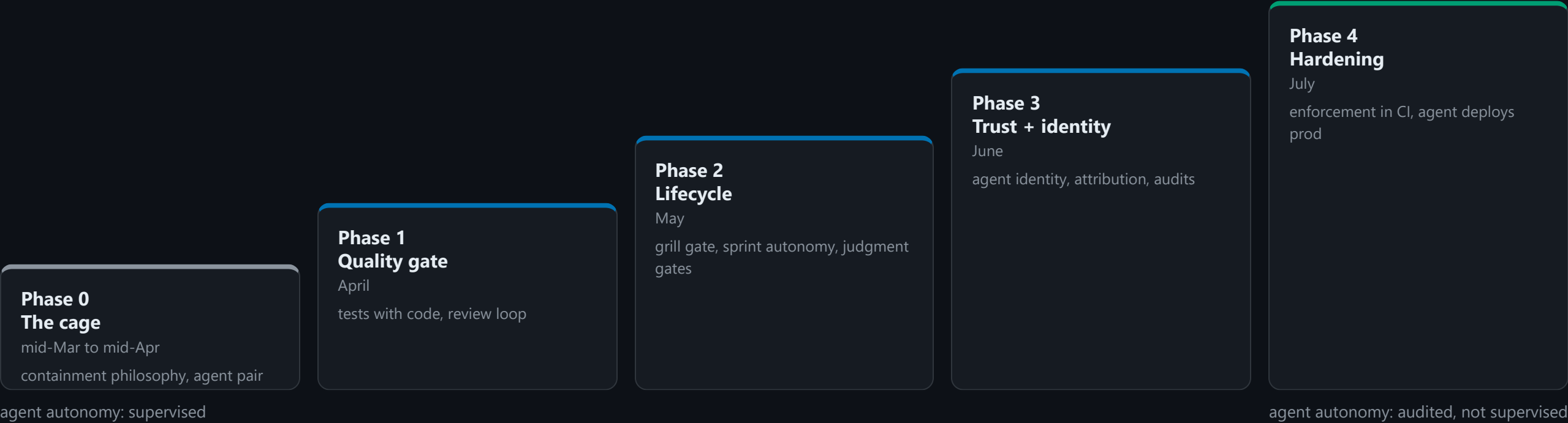
Audit compliance

Mine grew in five deliberate phases. Each phase turned the autonomy dial up one notch, and each notch was paid for with a lesson.

Where it ended up: who does what today



Five phases, one direction



PHASE 0 · MID-MARCH TO MID-APRIL · BEFORE THIS PROJECT EXISTED

The cage

- > I set out to build a **declarative security stack for self-hosted coding agents**: enterprise governance standards, for people running agents on their own metal.
- > A sandbox with **three controlled ways out and nothing else**. The agent never sees a real secret.

The failure, in its own words. First commit, 15 March: "**First commit, overly complex. Time to simplify.**"

What it bought: a posture and a team. Do not trust the agent: constrain, observe, review. And an agent pair, a backlog, tickets - it felt less like coding and more like hiring. **Vibecoding became discipline, and everything since grew from that.**

PHASE 1 · APRIL · PRS #1-62

The quality gate

LANDED

- › Every change ships with its tests, **in the same PR**.
- › Coverage ratchets that **can never go down**.
- › An architect agent reviews every PR, findings graded **must / should / nice**.

April 28: the agent put a **siren emoji** on the most compliance-critical log line in the system.

What it bought: an agent trusted to write most of the code, because the quality holds whether or not I read every line.

PHASE 2 · MAY · THE HEAVIEST DELIVERY MONTH: 189 COMMITS

The lifecycle

LANDED

- > The **grill gate**: no implementation before the plan survives interrogation.
- > Sprint-scale **auto-merge autonomy**.
- > The first **human-only judgment gate**: secrets.
- > Durable close-out docs.

May 23, one operator session, three lessons: public values shipped as secrets (30 minutes debugging blind); retrospectives lost to context compaction; **/healthz green for 5+ days over a crash loop**.

What it bought: sprint-scale autonomy: the agent merges on green inside a sprint, and the human merges only when the PR touches something sensitive.

Trust and identity

LANDED

- > The agent gets **its own GitHub identity**, distinct from mine.
- > Every action records **the directing human**.
- > Sprint audits: **every gate checked**, findings graded.

May: the grill gate **skipped twice** - the agent wrote itself a "Grilled basis" note instead of being interrogated. **I caught both by reading**. At 189 merged PRs that month, I was the only process check, and I do not scale.

What it bought: the confidence to stop watching. **Process review itself was handed over** - the audit checks every gate after the fact, so I am no longer the thing that catches a skipped gate.

Hardening

LANDED

- > Key rules pushed **from prose into CI**: structural tests, deploy health gates.
- > The **stop-gate**: a second model, in its own context, decides whether the agent is allowed to end its turn.

Long autonomous runs kept **stopping for input I had already given**. July 18: it handed me a deploy it was authorised to run itself. Every one cost a manual "**continue**".

What it bought: the question inverted. Not "what should make it stop?" but "**what could possibly justify stopping?**" A stop is legitimate only when the next action is not the agent's to take, or there is no next action.

The pattern

Every phase: a lesson, a rule, a notch on the dial.

The failure log is the process backlog.

And it caught my failures too.

Levels of enforcement



A rule an agent can rationalise around is a suggestion.

The phases are these levels applied to a whole process: prose in May, machinery by July.

STARTUP → ENTERPRISE

What transfers

The loop, and the phase discipline.

Gates as a pattern.

Attribution before autonomy.

Enforcement as code.

Judgment stays human at named points.

None of this is scale-dependent.

Alignment is more important than ever.

A managed, two-way connection between your in-store ShopFront point of sale and your Shopify online store, keeping stock, products, and orders in step automatically.

TrolleyRelay's architecture decisions are version-tracked. This document reflects decisions up to ADR-0019. If you are working from an older copy, ask us for the current version.

STARTUP → ENTERPRISE

What changes

The operator is not the owner: gates need owners and escalation paths.

Compliance is external and non-negotiable.

Many humans, many agents: attribution becomes identity management.

An existing SDLC to integrate with, not replace.

The dial starts lower and climbs slower. Same staircase.

Nobody vibe-codes a claims system.

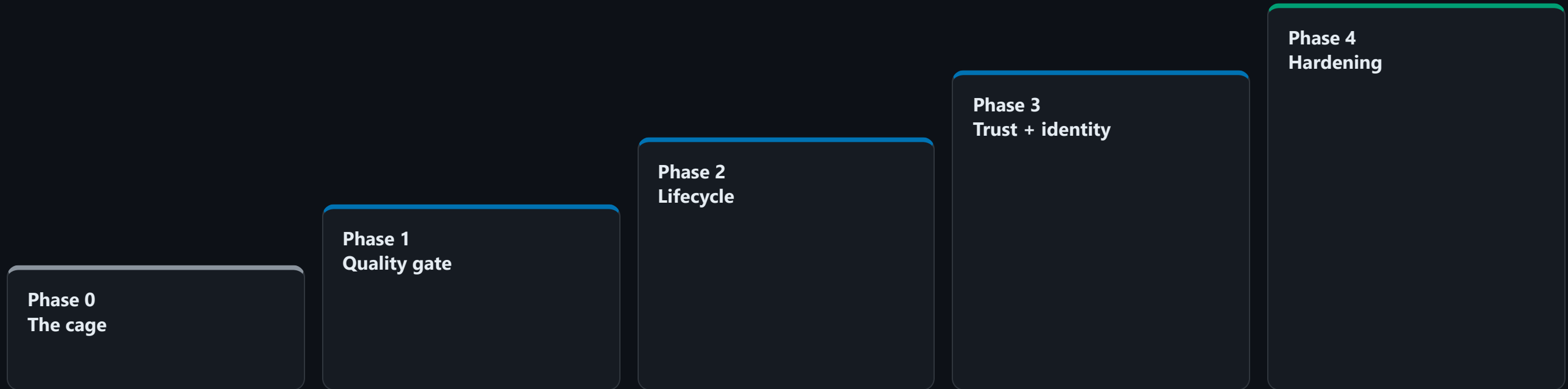
Where to start

You do not need my 165 lines. You need Phase 1 of your own climb:

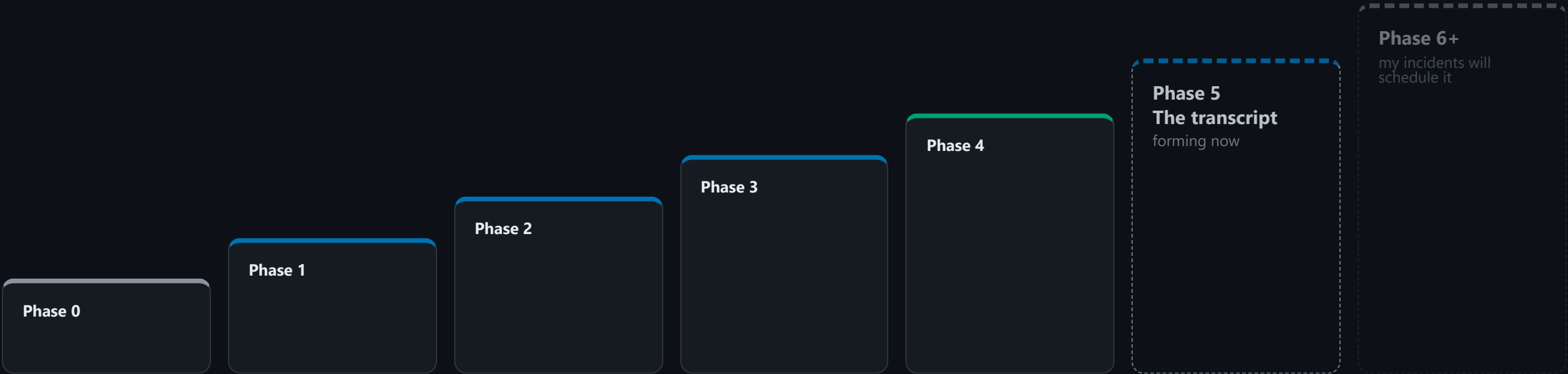
1. **Agent identity and attribution.**
2. **A review gate** whose findings cannot be silently skipped.
3. **A judgment gate** naming what only humans approve.

Then let your incidents schedule the next phase.

The climb so far



The climb is not over



Your dates and incidents will differ.
Your need to climb with them will not.

The long version: an article series, "You can't install an AI-SDLC". The first article is live at smcgowan.me.

Thank you.

Questions.

Stephen Mc Gowan

Founder, TrolleyRelay · AI Architect



smcgowan.me