April 15, 2026

From AI-Native Development to AI-Native Platform

How agentic workload changes development platforms — using GitHub as an example

What This Article Is About

Navigation Roadmap

Load unit + platform

- Parts 1-2: GitHub platform case.
- Part 3: broken load economics.
- Part 4: AI-native platform failure.
- Parts 5-6: lessons for orgs.

Part 1

01

GitHub as AI-Native Platform

GitHub is a control plane for people and agents

GitHub 2025–2026

AI is first-class load

Scale Scale

180M+ developers

+36M in 2025

80% tried Copilot in week one.

4.3M AI repositories

×2 in <2 years

All new repos: 230+/min.

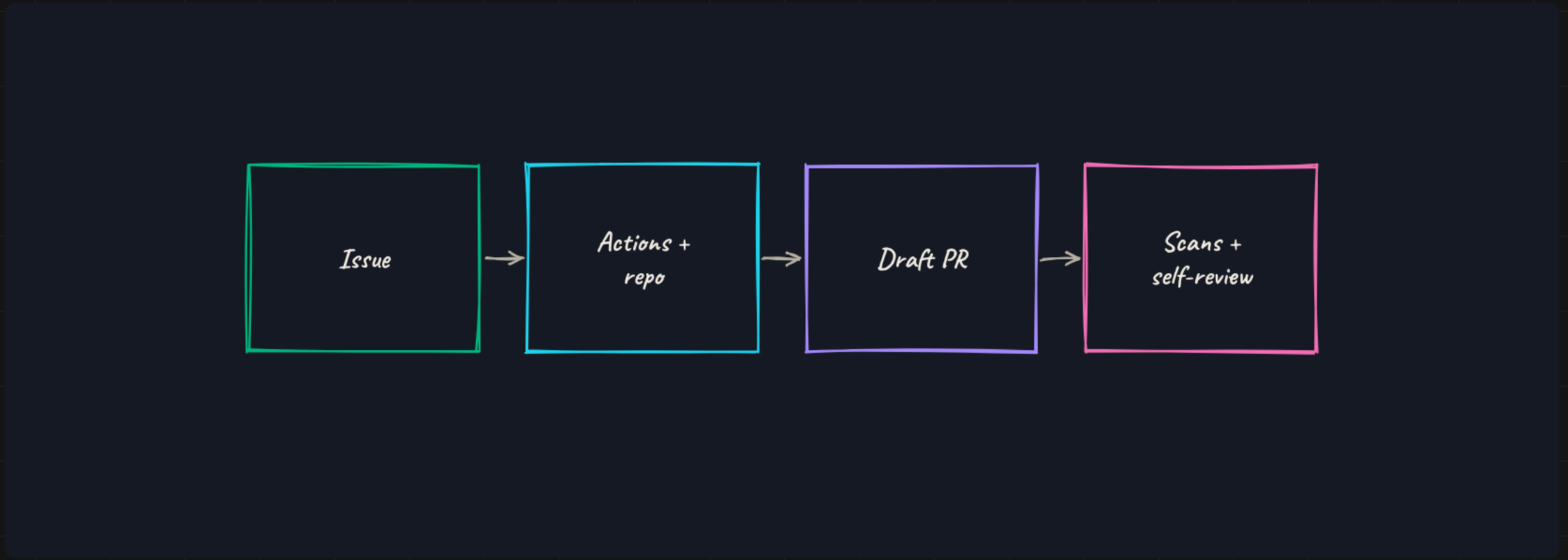
986M commits in 2025

5.5M issues (July)

+25% vs 2024.

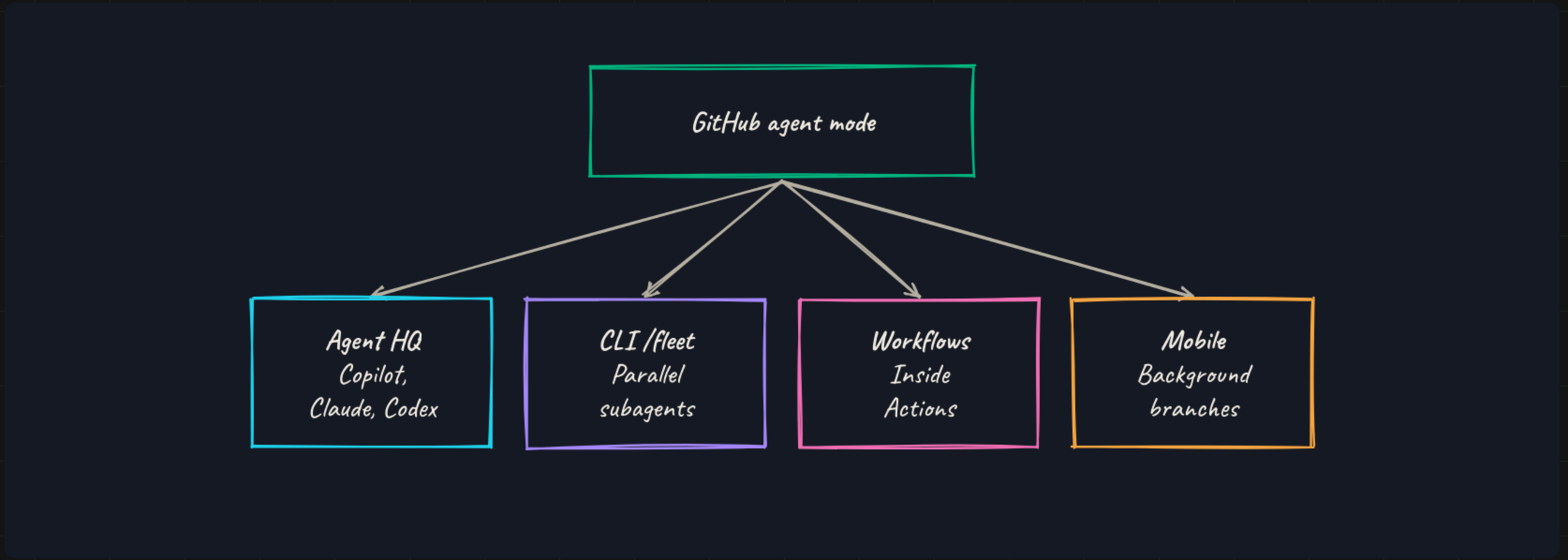
Task Reaches an Agent

Work Model | Agent Flow



GitHub Strengthens Agent Mode

Evolution Agent HQ



Part 2

02

What the Numbers Show

Actions, code review, usage metrics — three signals of new pressure

AI-load signals

Signals Metrics

AI is first-class load

GitHub Actions

11.5B minutes · +35% YoY

71M jobs/day; agents run on Actions.

Copilot Code Review

60M reviews · ×10 growth

>1/5 reviews; 71% actionable feedback.

AI-native telemetry

First-class traffic

Copilot PRs, cloud agents, CLI tokens.

Review Quality vs Computational Depth

Trade-off | Quality vs Cost

Quality

Reasoning model: +6% positive feedback

Agentic architecture: +8.1%

71% actionable

Cost

+16% latency on stronger models

Each review is a platform workload

Part 3

03

Why Old Architectural Assumptions Break

Agentic fan-out: one issue spawns a tree of platform operations

Old vs New Unit of Load

Load Model Fan-out

Classic platform	AI-native platform
Human changes code, waits for CI Metrics: MAU, API QPS, CI, storage	Intent → platform action tree Fan-out and operations per intent

The load unit is an operation tree per intent

Part 4

04

AI-Native Platform Failure

New load meets old architecture

Incident: Feb 9, 2026

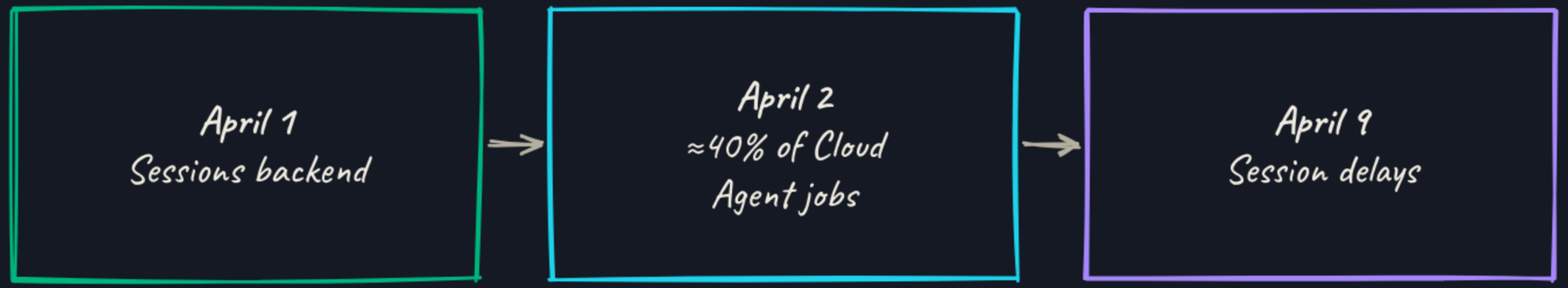
February 9 Incident



Gaps: isolation, load shedding, monitoring

April Incidents

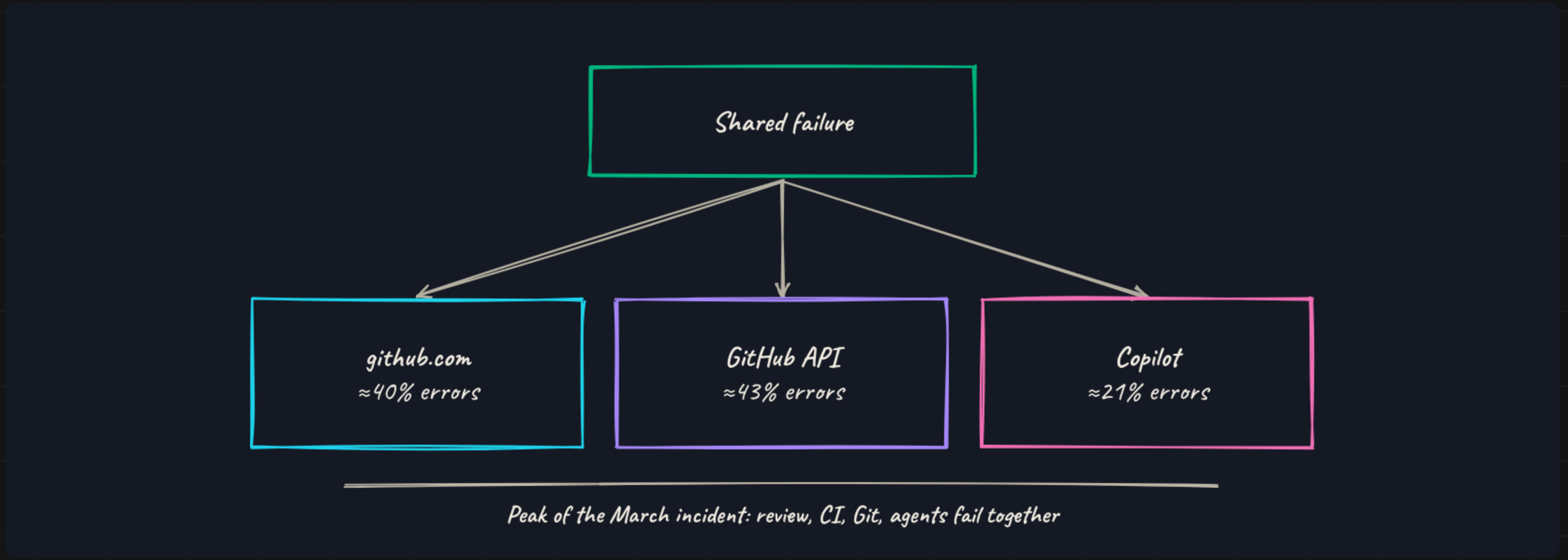
April Incidents



Separate incidents reveal pressure on the shared runtime

Shared Failures Spread Wider

Blast radius Availability



Part 5

05

For Large Engineering Orgs

Process-wise — integrate; infrastructure-wise — isolate

Process Integration, Infrastructure Isolation

Principle Architecture

Integrate workflow

PR · CI · audit
Shared checks

Isolate load

Failure domains
Per-tenant priority

DORA: AI Amplifies Platform Properties

DORA Evidence

Discipline becomes visible

- DORA 2024: better docs, quality, review speed.
- Cost: lower throughput and stability.
- DORA 2025: throughput vs stability.
- GitHub shows this dynamic.

Part 6

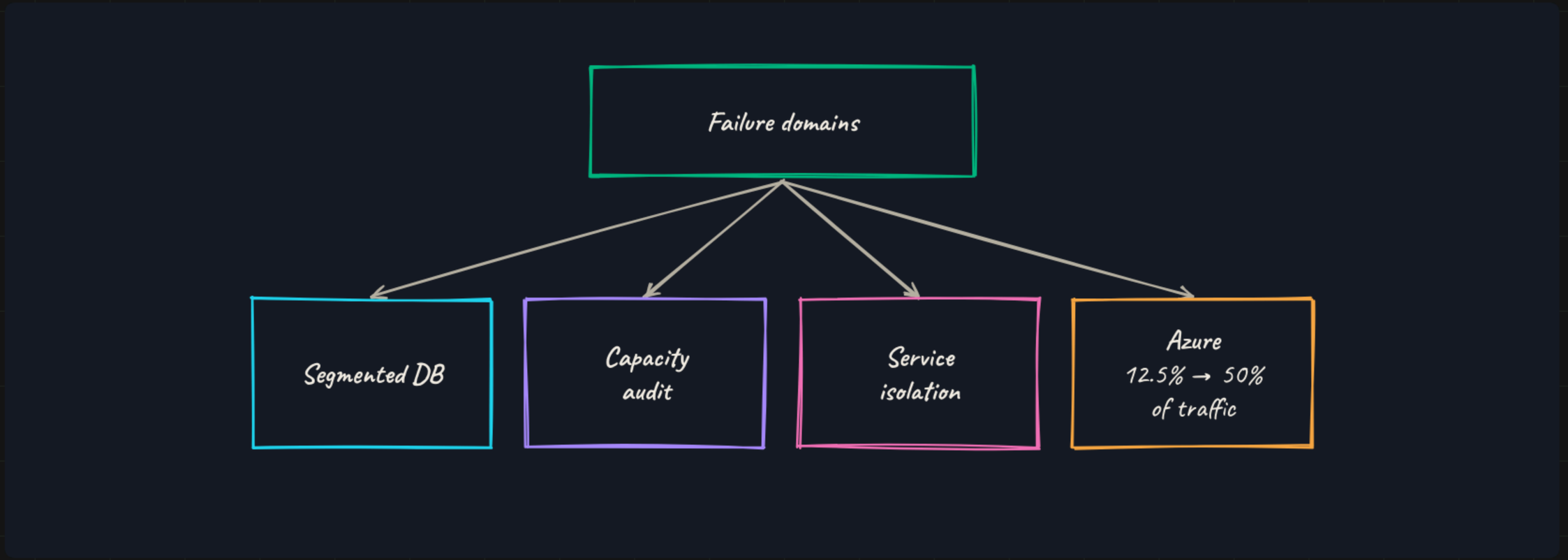
06

GitHub Outlook and Conclusions

The right problem class — the right category of actions

GitHub Response: Platform Work

GitHub Response



Outlook Outlook

Moderately Positive Medium-Term Outlook

Right direction; pressure grows

- Fix substrate while load grows.
- Multi-provider and mobile agents.
- Runtime SDK and remote CLI sessions.
- Short-term cautious; medium-term positive.

Key Takeaways

What to take away

- Bottleneck: platform layer, not authoring.
- Agentic fan-out spawns platform operations.
- Guardrails are production load.
- Platform engineering manages fan-out and machine traffic.

- ↗ tellmeabout.tech · ↗ ai4sdlc-research.space

References and Materials

Core sources

- Classic PDLC to AI-native article.
- GitHub Blog, GitHub Status and Copilot updates.
- GitHub Actions, Code Review, org metrics.
- DORA 2024 report.

AI-Native Platform

Thank You!

For more materials on this topic, visit the ”Book Cube” channel — all links are collected there

Alexander Polomodov

Technical Director & Fellow,
T-Technologies

@Book_Cube