Practical Guide to Agent Native Platforms

Øredev 2026

AI Agentic empowered software development is here to stay

and it's evolving fast!

Code is shifting faster than ever before

	Q3 2025		Q3 2026
Issues created	4.85M	→	**15.65M**
Pull requests merged	14.81M	→	**47.04M**

Public GitHub repositories, quarterly totals (Jul 1 to Sep 30). Source: GitHub Search API, retrieved 20 Oct 2026.

We’re moving from writing code to orchestrating change

Internal developer platforms are the foundation for succeeding in the AI era



Jessica Andersson

Fractional CTO & Platform Engineering Leader
CNCF Ambassador
Gothenburg CNCG Organizer
Cloud Native Nordics

Jesande

Fractional and interim CTO for growing companies that need experienced technical leadership to keep their momentum as they scale.

jesande.io

Platforms in the age of agents

01 Fundamentals of agents

02 Internal Developer Platforms: why, and how they relate

03 Agent-Native Platform

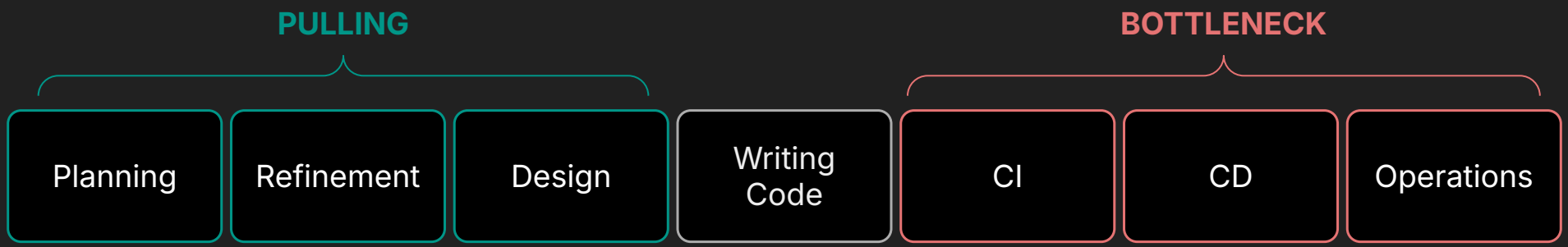
04 Live demo

Code is shifting faster than ever before

	Q3 2025		Q3 2026
Issues created	4.85M	→	**15.65M**
Pull requests merged	14.81M	→	**47.04M**

Public GitHub repositories, quarterly totals (Jul 1 to Sep 30). Source: GitHub Search API, retrieved 20 Oct 2026.

The bottleneck moved



~~2 months~~

2 days

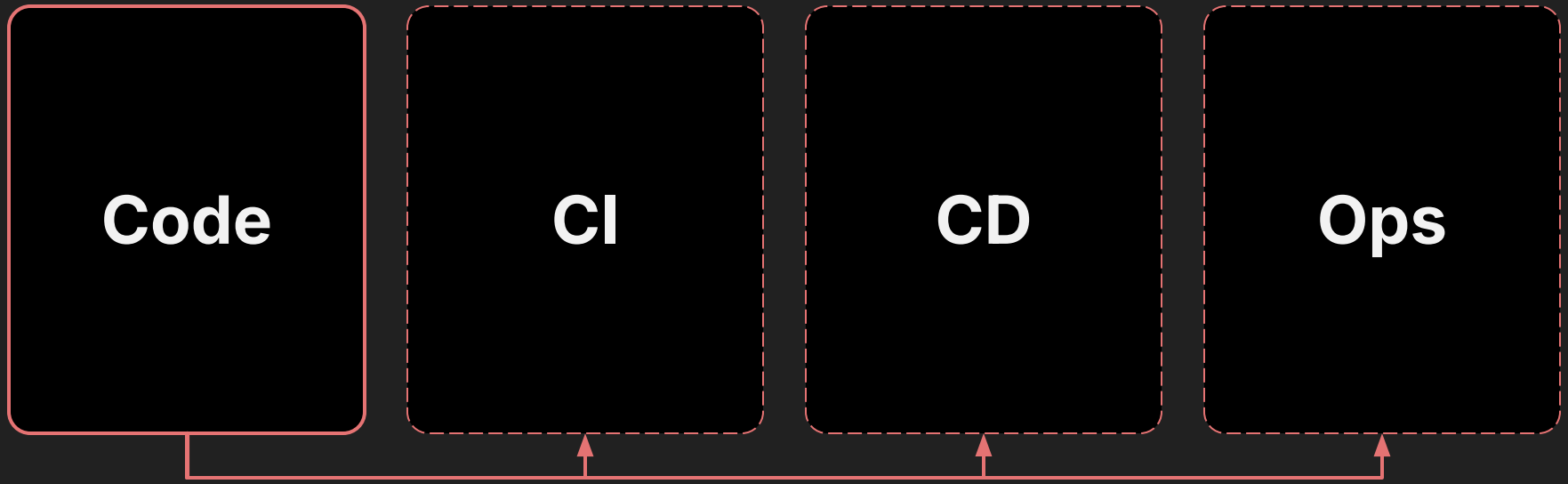
Models are stochastic by design

Inconsistent outputs

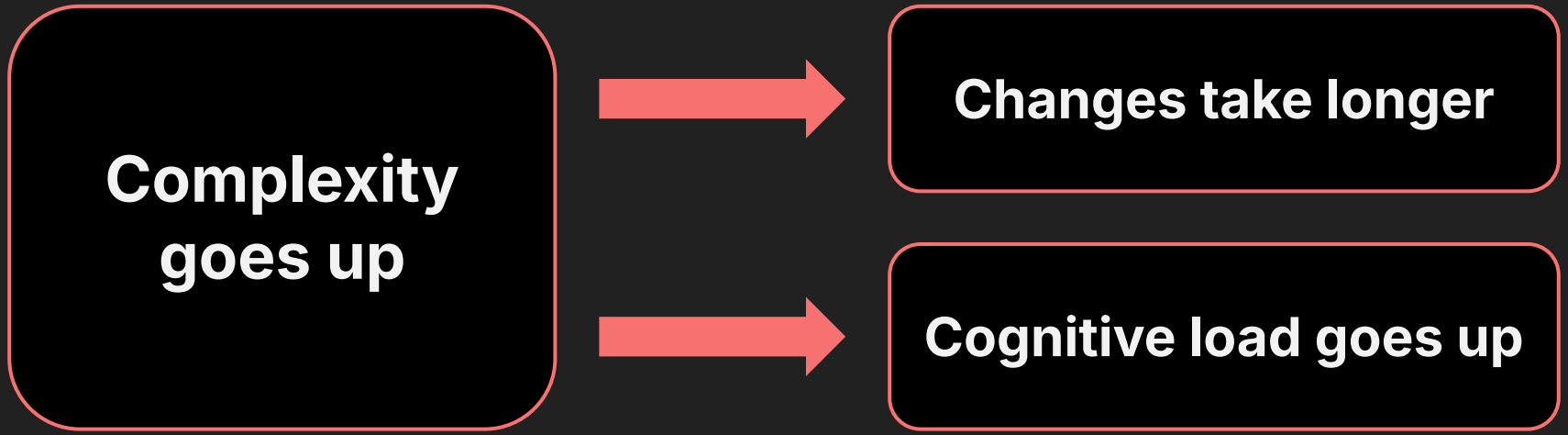
Codebase fragmentation

Fragmentation has an immense blast radius

Introducing a new language or framework isn't *just* a code change, it's a commitment to extend your entire production support stack.



More complexity means slower changes and higher cognitive load



Complexity must go down as output goes up

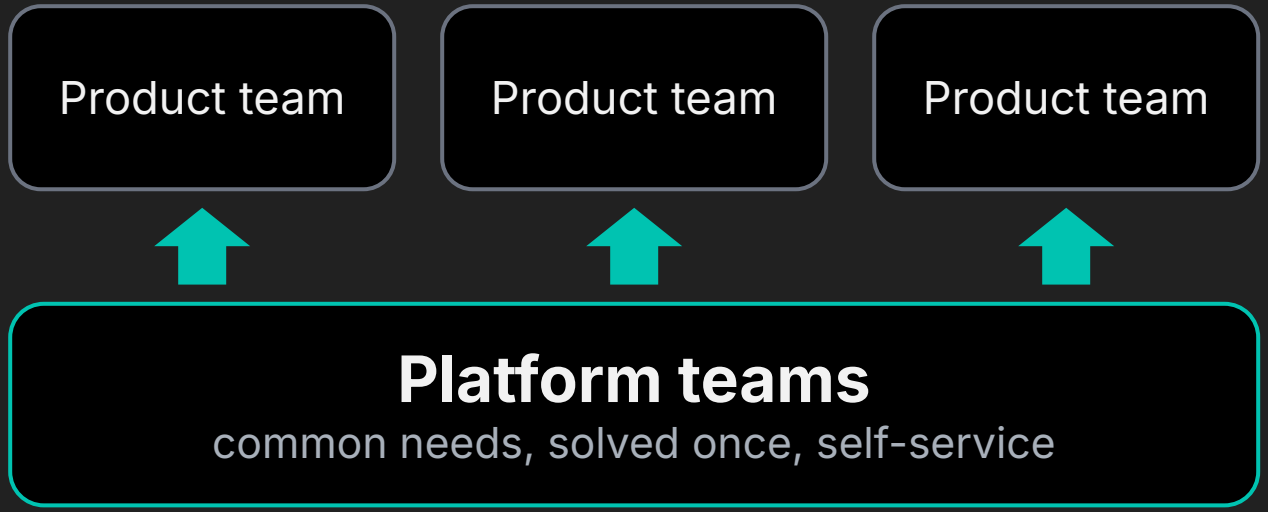


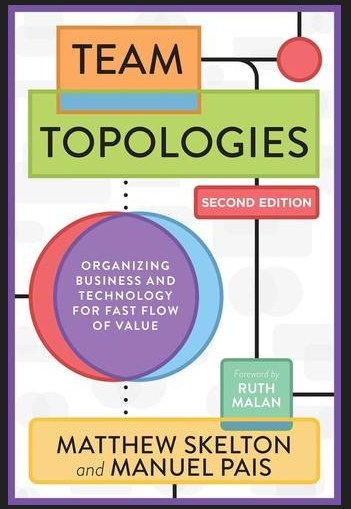
Output increases



Complexity and cognitive load must decrease

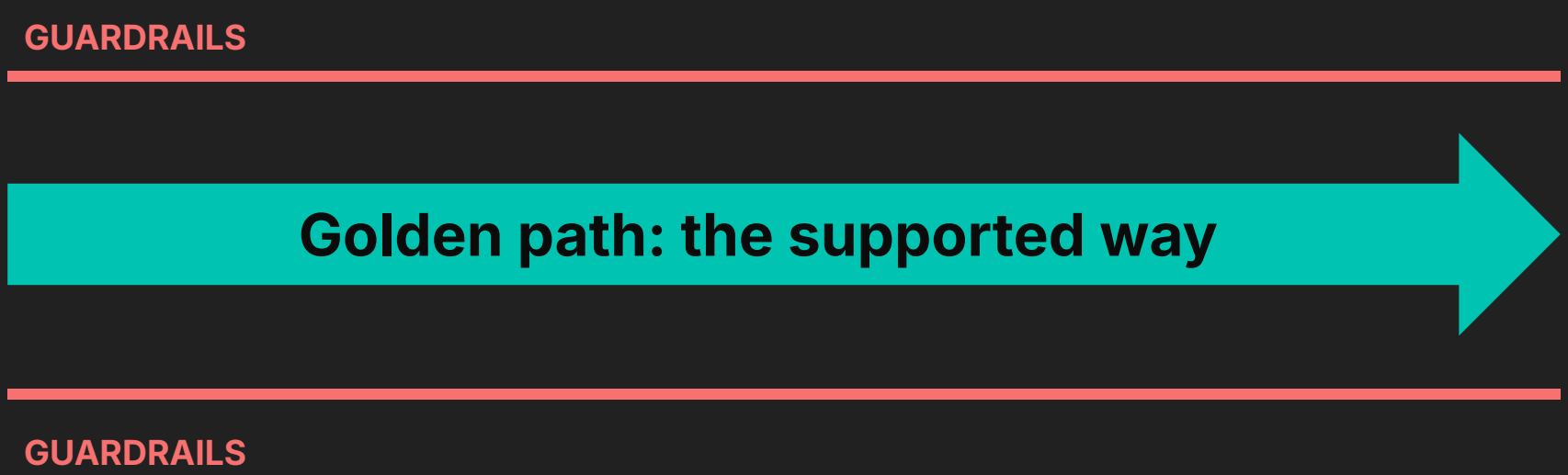
We build platforms to solve common needs once





Platform and stream-aligned teams, from Team Topologies (Skelton & Pais)

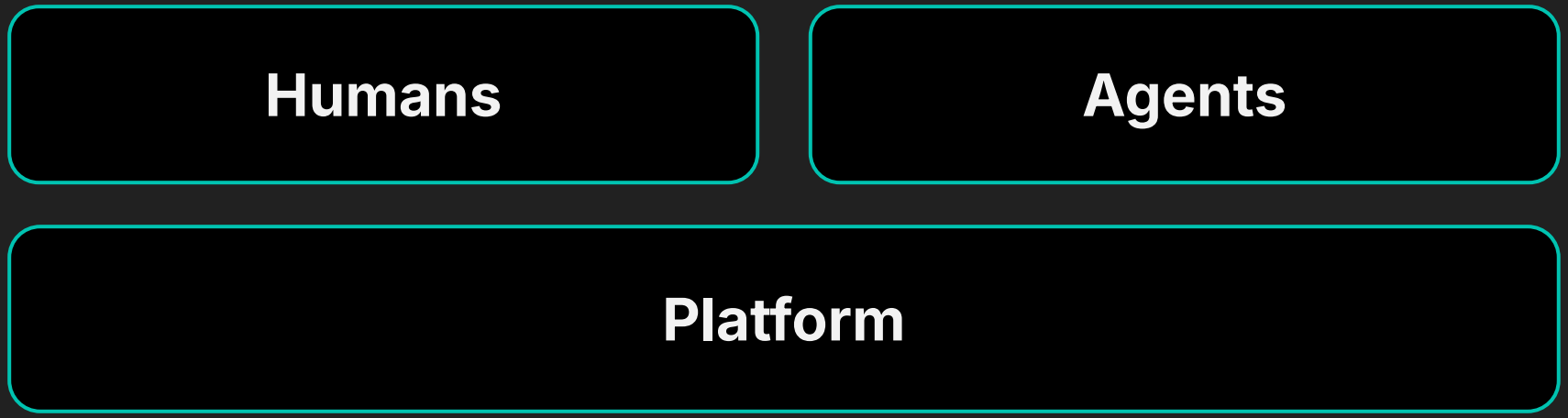
Golden paths and guardrails keep complexity and cognitive load down



Citizen developers are new platform users



A good platform for humans is the foundation for agents



No platform yet? Now is a good time to start.

What platforms already solve works for agents too

CI/CD

Runtime

Observability

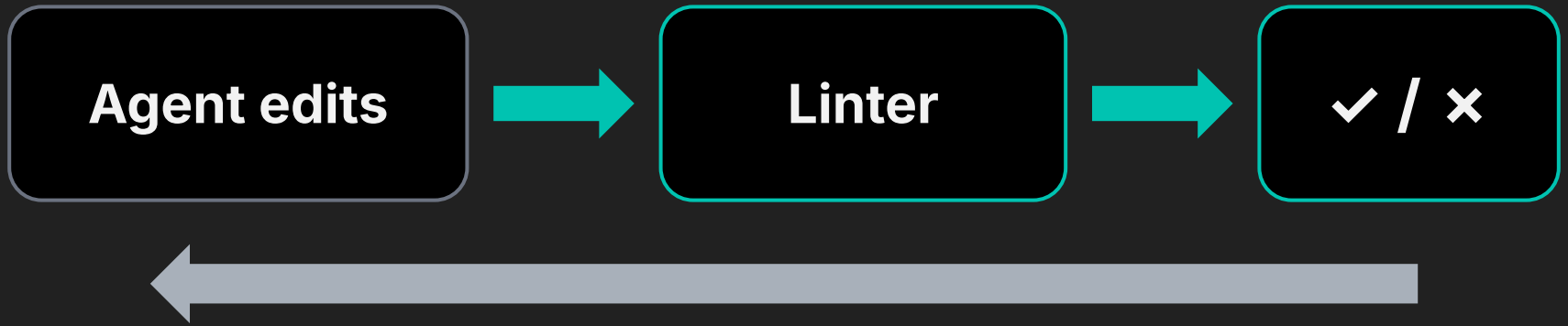
Security

Design systems

API-first and self-service

So what else helps agentic development?

Deterministic checks give agents fast feedback



Same input, same result, every time. And the same linter for humans.

Just-in-time context: search, then look up

```
$ goldenpath search storage
storage   Storing data
buckets   Object storage (buckets)

$ goldenpath lookup buckets
# Object storage (buckets)
- The bucket name comes from STORAGE_BUCKET ...
```

Only what the agent needs, when it needs it

Keep the laptop thin, control it from the platform



Hard to update what's on laptops, easy to update a service

Demo: building on the golden path



github.com/Solisol/goldenpath-demo

You have the same tools as your agent



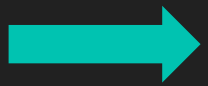
The same information and the same guardrails, for both

If the agent fails, you can still move forward

× Your agent

provider outage, rate limits, a bad day

✓ You



Same tools, same platform

Nobody gets locked out

Takeaways

01 Reduce complexity with golden paths

02 Invest in qualitative self-service platforms

03 Enable agents with fast feedback loops and just-in-time context

Thank you