Your platform matters more than ever with AI

Jessica Andersson

AI Agentic empowered software development is here to stay

and it's evolving fast!

The Multi-Directional AI Pressure

EXTERNAL PRESSURE

Market Position & Competition

Ensure you hold or advance your market position against aggressive competition in an AI-first landscape.

INTERNAL PRESSURE

Mandates & Fragmentation

- Top-down: C-suite demands to “do something with AI.”
- Bottom-up: Shadow AI leading to increased fragmentation.
- Security: Unregulated experimentation creating new risks.

GitHub monthly developer activity metrics

	2024		2025
Issues closed	≈3.4M	→	**4.25M**
Pull requests merged	35M	→	**43.2M**
Code pushes	65M	→	**82.19M**

monthly average developer activity metrics. source: Octoverse 2025.

Fail at Machine Speed

THE RISK

Unintentional Fragmentation

Without intentional strategy, AI accelerates poor decisions and creates a fragmented, unmanageable landscape at scale.

THE SOLUTION

Intentional Strategy & Platforms

- Intentional AI Policy & Roadmap
- Internal Platform Standardization

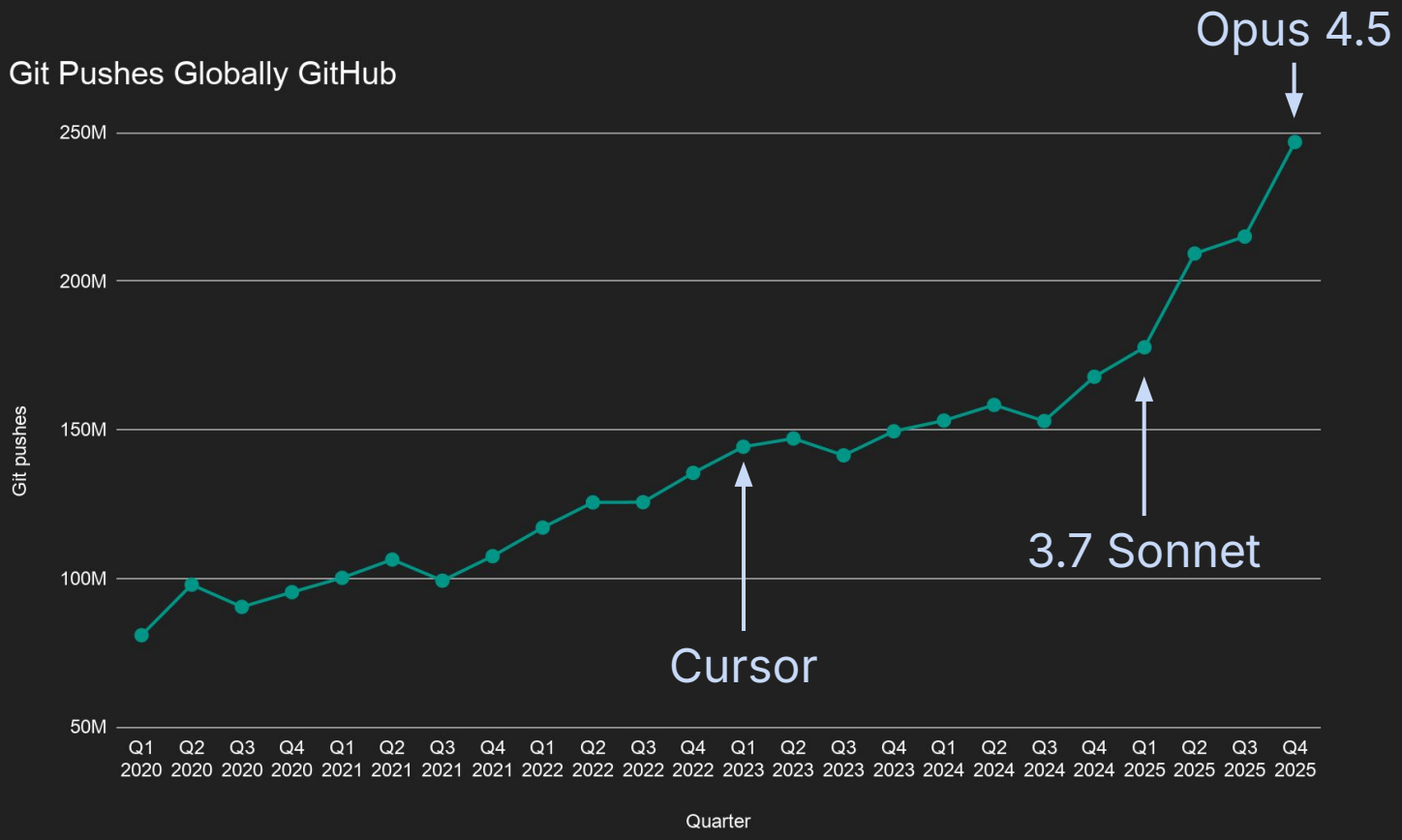
Your platform matters more than ever with agentic AI

We're moving from writing code to orchestrating change

Trend - output increases

	2024		2025
Issues closed	≈3.4M	→	**4.25M**
Pull requests merged	35M	→	**43.2M**
Code pushes	65M	→	**82.19M**

monthly average developer activity metrics. source: Octoverse 2025.


Git Pushes Globally GitHub
Opus 4.5
250M
200M
150M
100M
50M
Git pushes
3.7 Sonnet
Cursor
Q1 2020
Q2 2020
Q3 2020
Q4 2020
Q1 2021
Q2 2021
Q3 2021
Q4 2021
Q1 2022
Q2 2022
Q3 2022
Q4 2022
Q1 2023
Q2 2023
Q3 2023
Q4 2023
Q1 2024
Q2 2024
Q3 2024
Q4 2024
Q1 2025
Q2 2025
Q3 2025
Q4 2025
Quarter

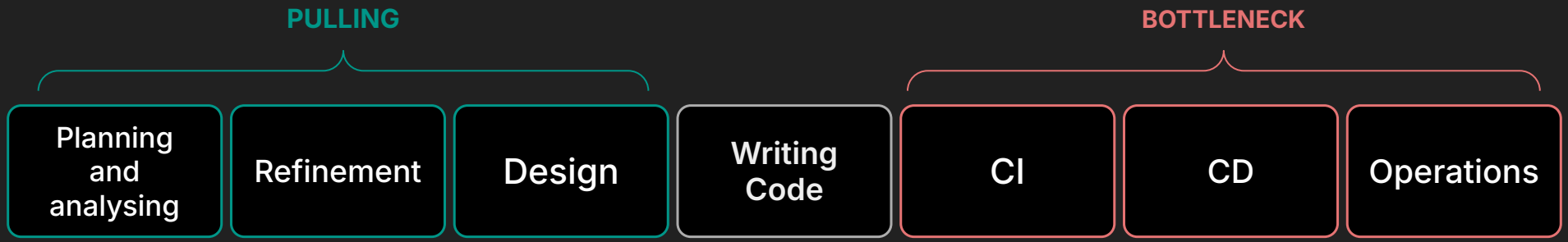
jesande.io

Research looks back..

..the future is unknown

But what do we know?

Each developer owns more and the bottleneck moved



~~2 months~~

2 days

Models are stochastic by design

Inconsistent Outputs

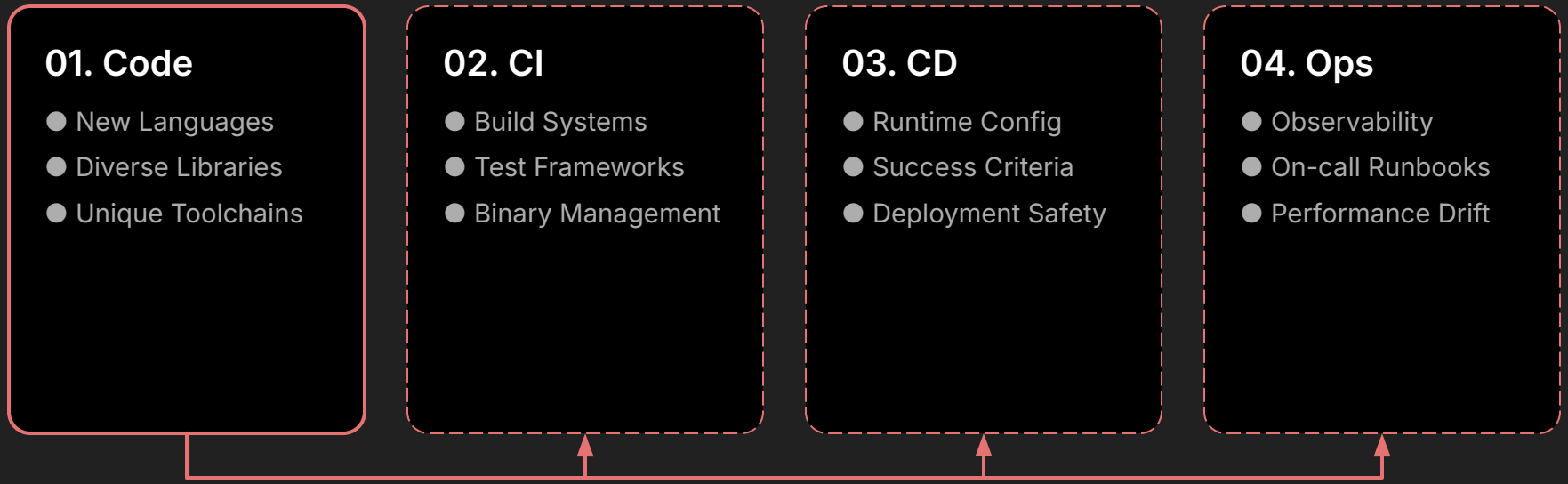
The core challenge of LLM integration is inherent **randomness**.

Codebase Fragmentation

Stochastic behavior leads to **unmaintainable systems** without technical guardrails.

Fragmentation has an immense blast radius

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



The Cascade Effect: Fragmentation in code demands 4x the infrastructure investment.

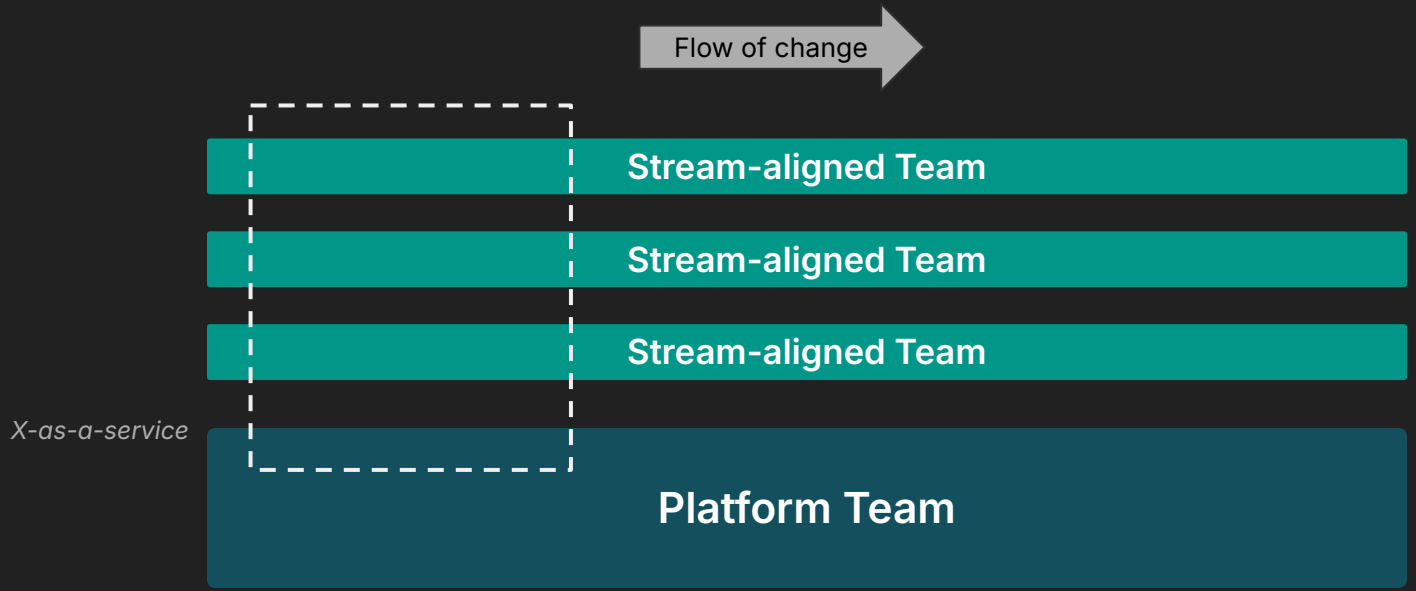
But what do we know?

More output

Fragmentation default

Fragmentation increases complexity

Why we built IDPs in the first place



Skelton & Pais, Team Topologies

Same problems, bigger blast radius



Velocity vs. Control

Agents generate code at volume; manual gates can't keep up with machine speed.



Invention vs. Standards

No paved road = the agent invents one badly. Cognitive load explodes without defaults.



The Mirror Effect

AI amplifies whatever is in place. Agents amplify it faster, scaling existing issues.

Platforms now serve humans & agents

Humans need	Agents need
Ergonomic UI / developer portal	Machine-readable interfaces (APIs, MCP)
Mental shortcuts, golden paths	Idempotent, deterministic operations
Discoverable docs	AI-accessible context (parseable, current)
Audit trails (for compliance)	Audit trails (for trust + rollback)
Examples to copy	Well-described tools with clear contracts

Strategic blueprint: Adaptable by design

A living strategy requires intentional communication and the courage to evolve.

Adaptable

Build for the shape of change, not specific models. Prototyping speed dictates that strategy must remain fluid.

Communicate

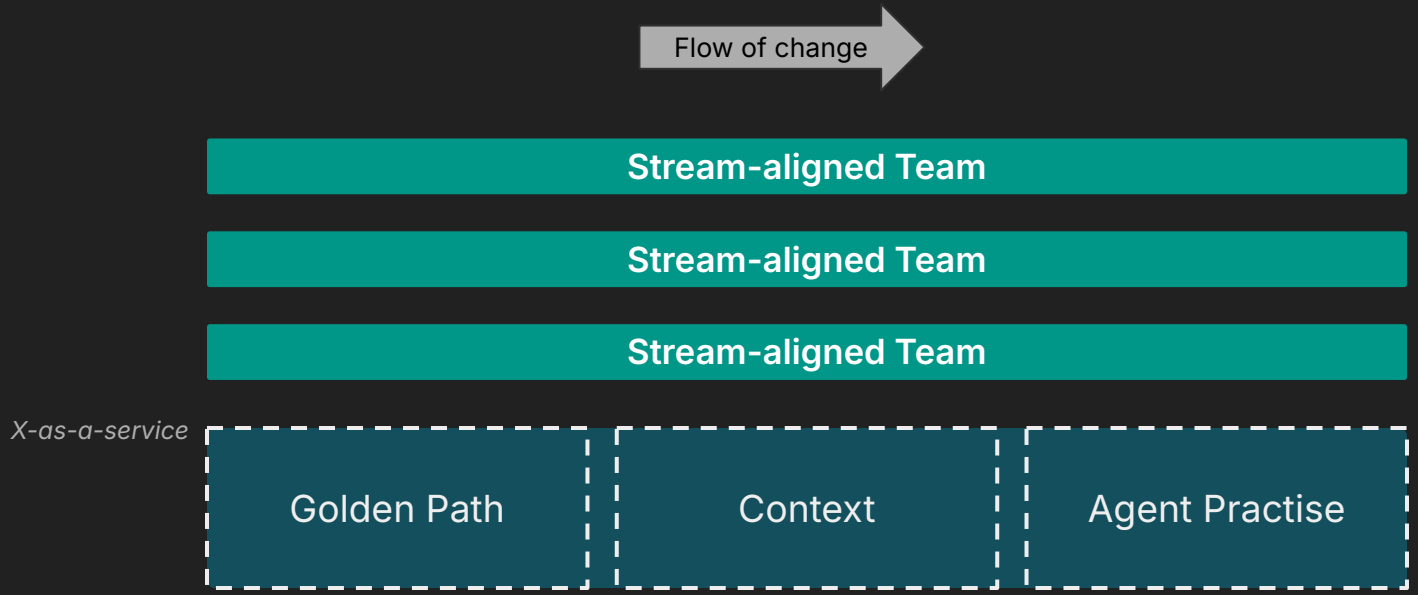
Clear and often. A clear AI stance must be communicated to capture org-wide alignment.

Transparent Pivot

Be clear on when and why the roadmap shifts. Every pivot is an intentional step toward higher leverage.

Adapting is not failing — it is learning and improving at machine speed.

The platform executes on the strategy



Golden Path

Paved roads from idea → production. The happy path both humans and agents follow.

- **Template repos / project scaffolds:** Standardized starting points for consistent development.
- **CI/CD pipelines:** Default security and quality gates baked into every deployment.
- **Policy-as-code:** Branching, review, and deployment rules encoded once for the whole org.
- **Agent-specific tools:** Review-bot integration, deterministic rollback, and automated security review.

Less variability, stronger defaults — AI gains **compound** across teams instead of fragmenting.

What counts as context?

The spectrum runs from deterministic (verifiable) to stochastic (interpreted).

Deterministic (verifiable)	Stochastic (interpreted)
Linters & Static Analysis policy-as-code	Markdown guides & AGENTS.md
Type systems & IDLs	ADRs & Design Docs
Schema validators & Data contracts	Runbooks & Troubleshooting guides
CI rules & Infrastructure-as-code	Internal Wikis & Handbook

If the agent can’t find your context, it will invent one.

just-in-time context

Adding context ≠ adding value. **Right context at the right time** is what helps agents succeed.

Too much irrelevant context → more failures, not fewer

Feedback loops close this gap:

- Which context did the agent use? Which did it ignore?
- Which sessions failed despite having relevant context loaded?
- **Refine. Trim. Re-rank. Repeat.**

Agent Practice: Setting Boundaries

What every organization needs, even without sophisticated infrastructure.

Centralized Governance

Single point of control for sanctioned models and shadow spend prevention.

Data Boundaries

Explicit perimeter controls for PII and code training clauses.

Calibrated Access

Tiered guardrails: human-in-the-loop for critical tasks, flexible for experiments.

You need to know what data goes where

Agent Practice: Awareness & Learning

The flexibility layer. Model capabilities change in months — this pillar is where you stay nimble.

Cross-Org Learning

Shared libraries and "what we tried" digests ensure Team A's discovery reaches Team B.

Visibility, not Surveillance

Lightweight telemetry to spot trends and spend without micromanaging people.

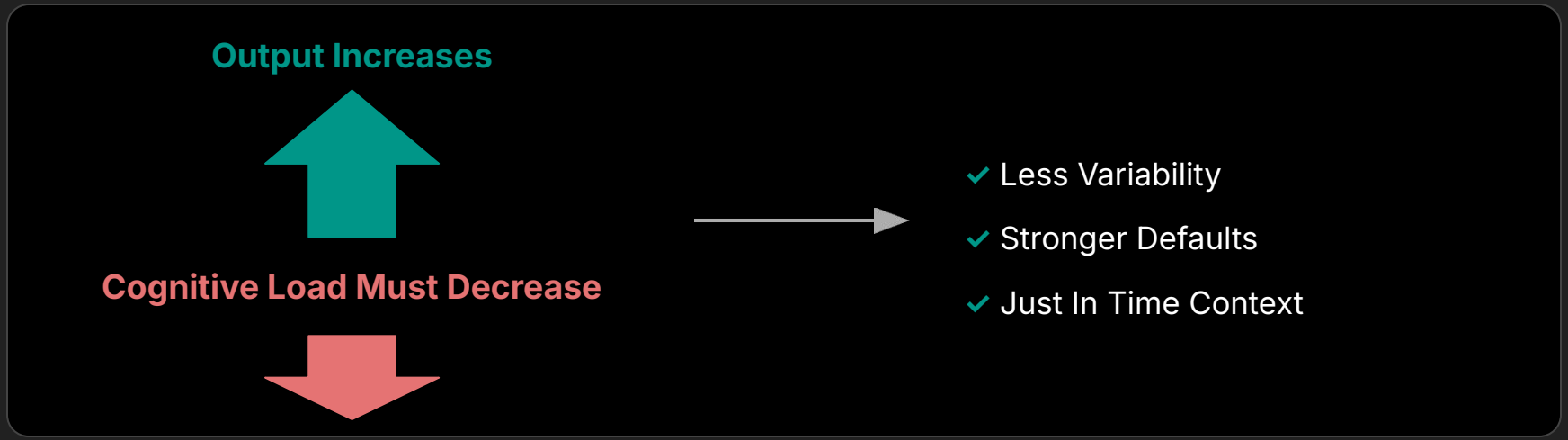
Built for Model Churn

Stay model-agnostic. Avoid hard-wiring to specific SDKs as the model layer evolves.

Keep your platform *alive* in a world of evolving models

Less variability, stronger defaults

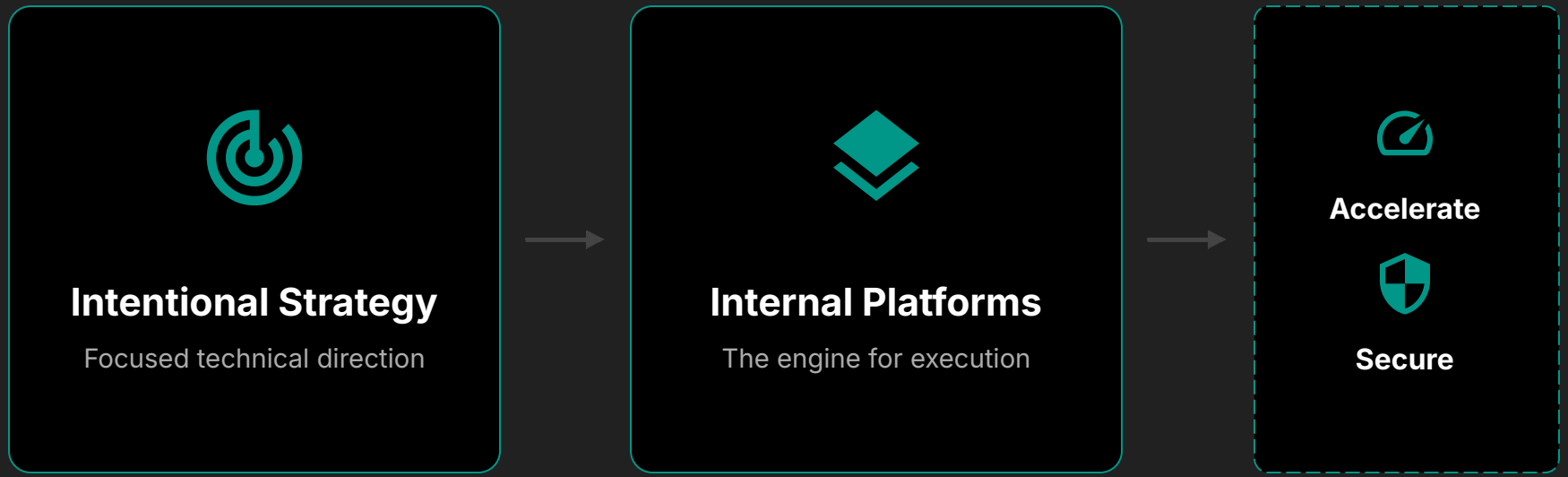
Reducing cognitive load as output increases



Complexity must go ***down*** even as volume of code goes ***up***.

Strategy Drives AI Success

Intentional platforms secure and accelerate development



Powering your ***AI-empowered development*** through consistency.

Three things to do on Monday

Execute the DORA capabilities under platform-team control



Publish AI Stance

Stop relitigating policy team-by-team.

Org-wide alignment



AI-Accessible Context

Stop agents inventing what they can't find.

Accuracy + Trust



Invest in Quality

Stop bespoke experiments that don't compound.

Scalable improvements

Each removes a ***fragmentation cost*** and captures a piece of ***leverage***.

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

jesande.io



Jessica Andersson

Fractional CTO & Platform Engineering Leader

Jesande

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

jesande.io