Deterministic Boundaries, Agentic Core: Navigating the AI Fog

Nasser Mohammadiha, PhD, Docent
Expert - Data Analytics for DevOps, Ericsson AB
2026-11-10

What's Covered

ERICSSON

1

Where we stand

What changed in seven months, what the evidence shows, and why it feels like fog

2

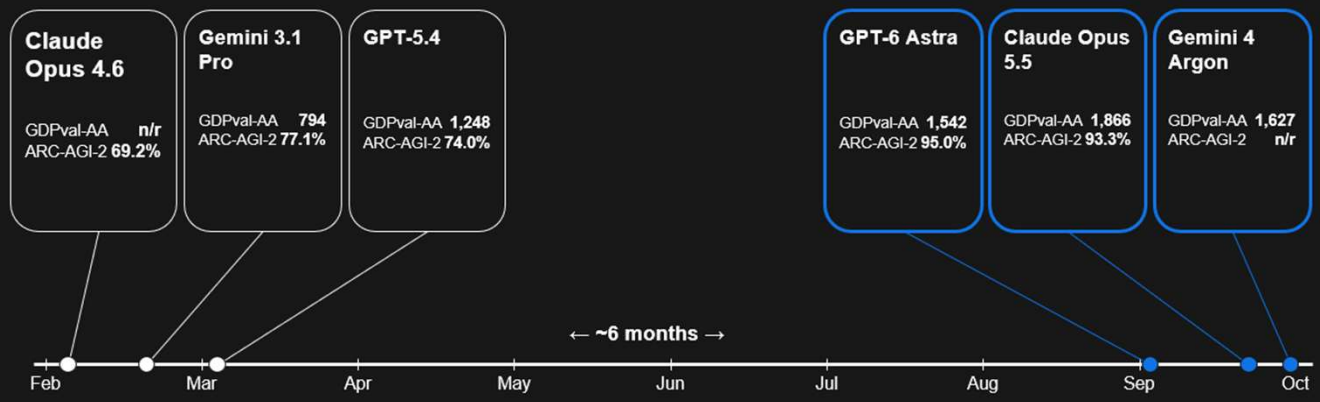
How we transform

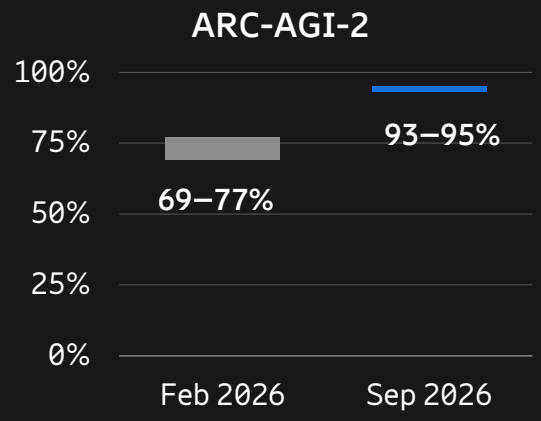
What the agent sees, how we work with it, and what it is allowed to do

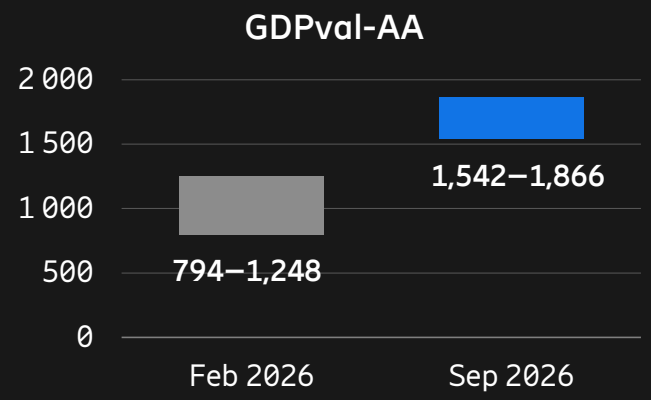
November 2026: where we stand

- Models are no longer the bottleneck
- Model and harness: choose them as a pair
- The hardest part is changing how we work

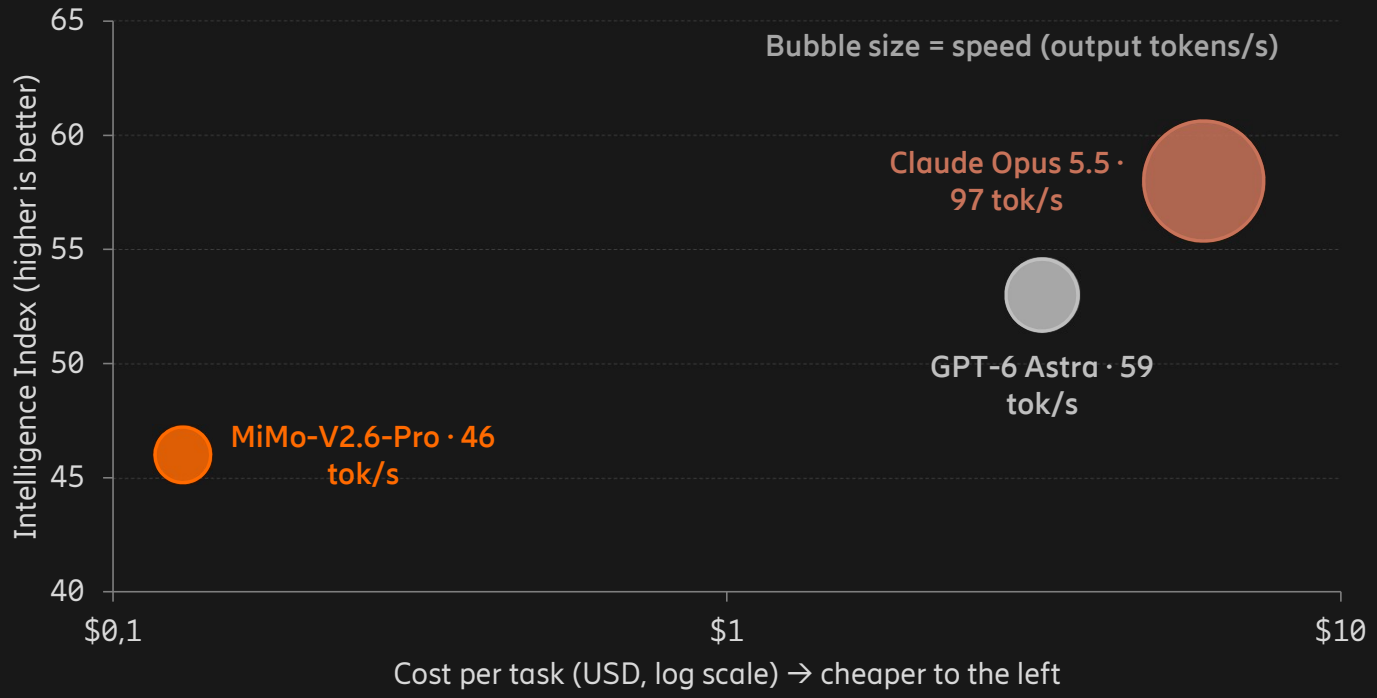
Seven Months Step







Better? Faster? Cheaper?



Best, at a price

The middle ground

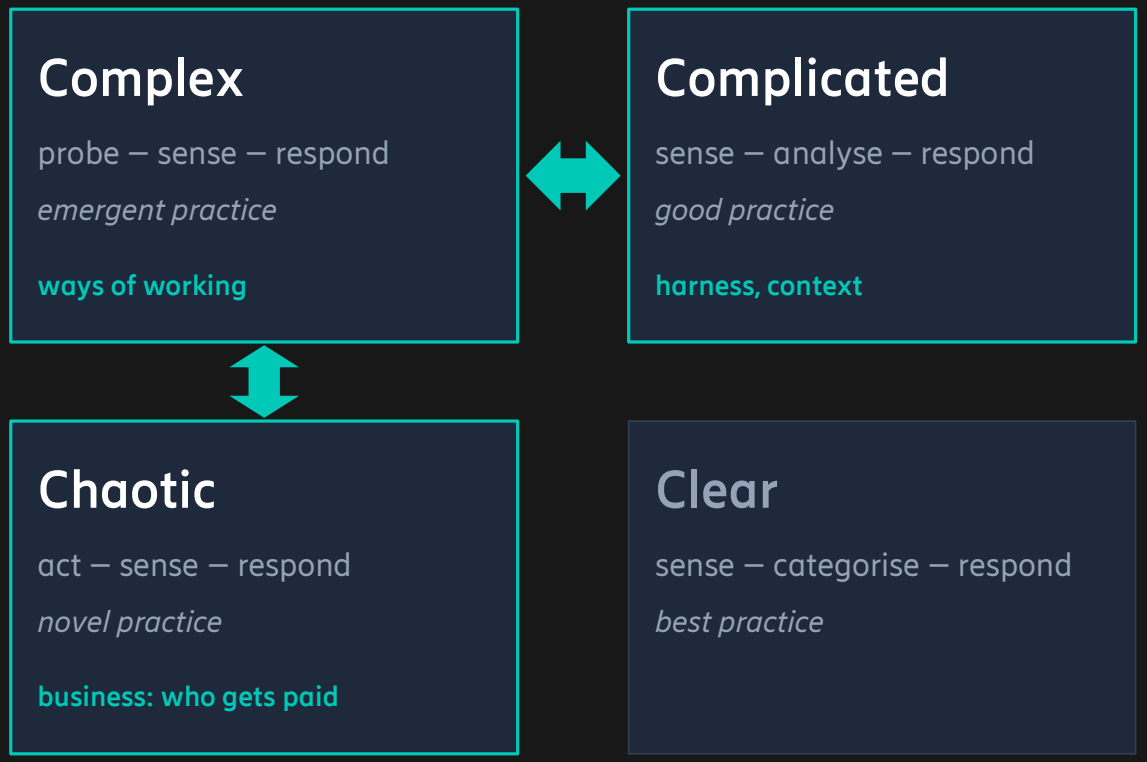
Good enough, very cheap

https://artificialanalysis.ai/

The AI Fog: Learning Through Experimentation

AI moves our ways of working here.

Same needs, new winners.

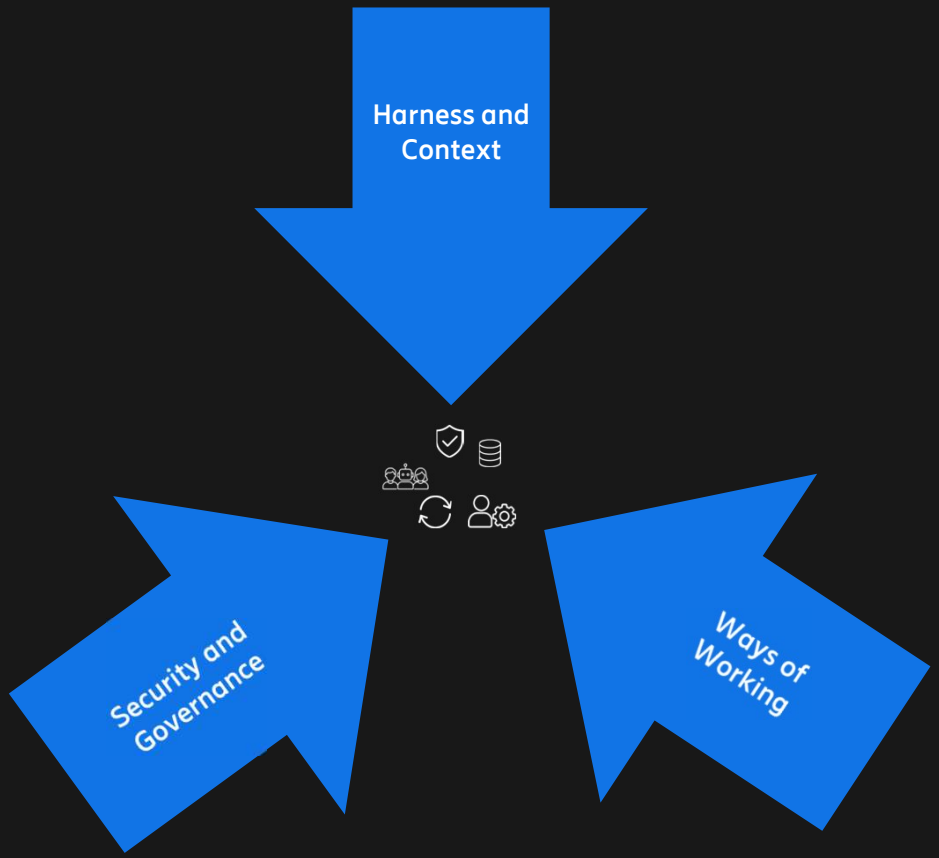


Where much of our engineering used to live.

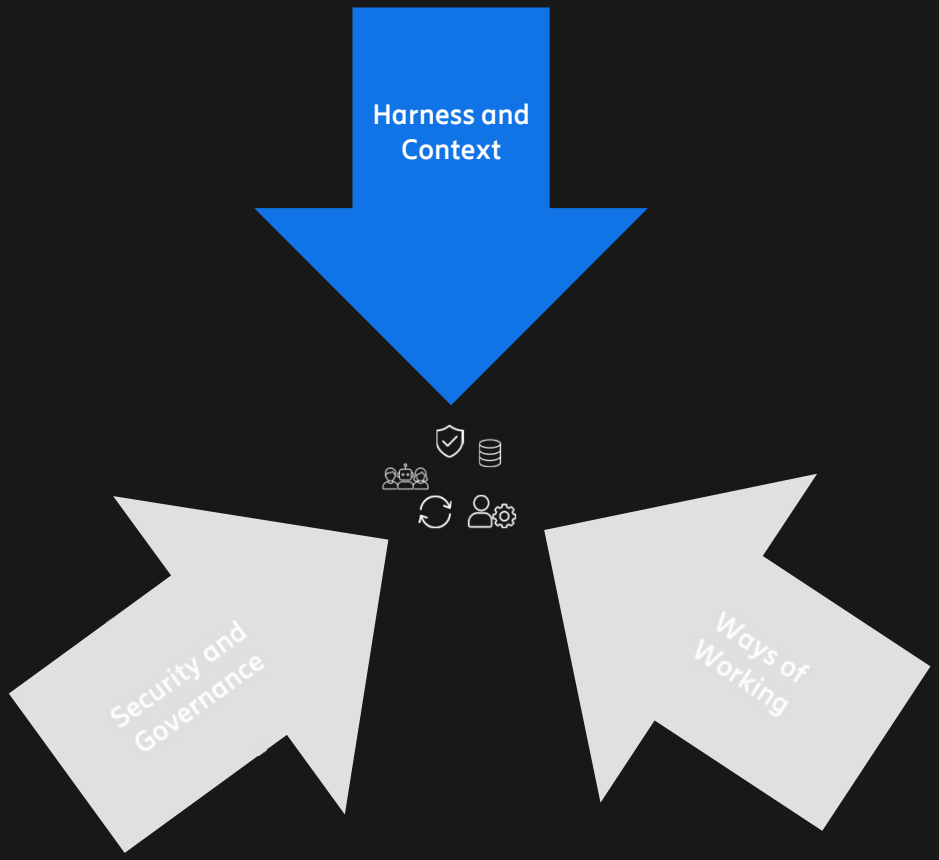
AI-Native R&D



Three Pillars for AI-Native R&D



Three Pillars for AI-Native R&D



The Model

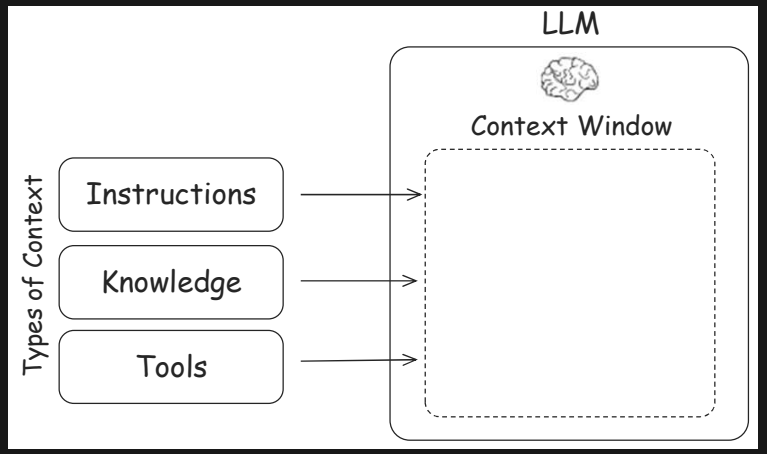
ERICSSON

The Model: reasons about situation, decides actions

The Context

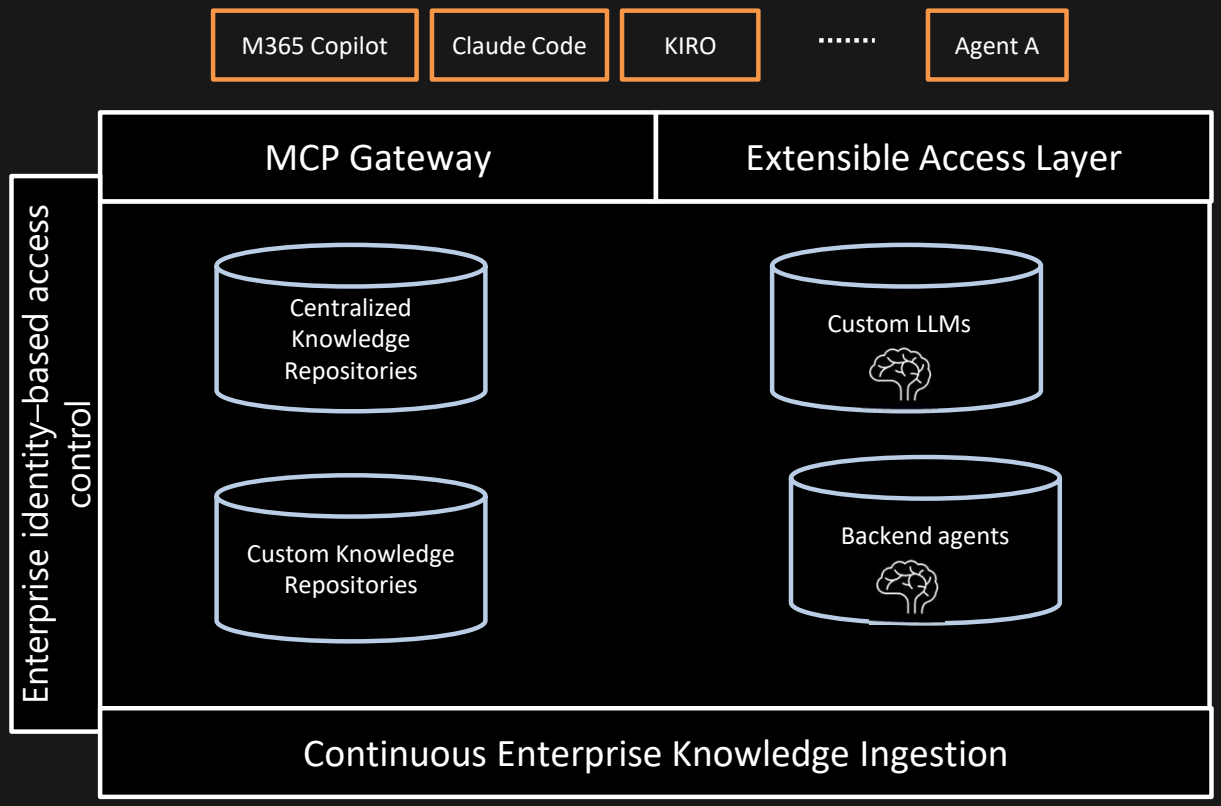
The Model: reasons about situation, decides actions

The Context: what the model sees



Context Engineering at Scale

(Infrastructure, not a user responsibility)



The Harness

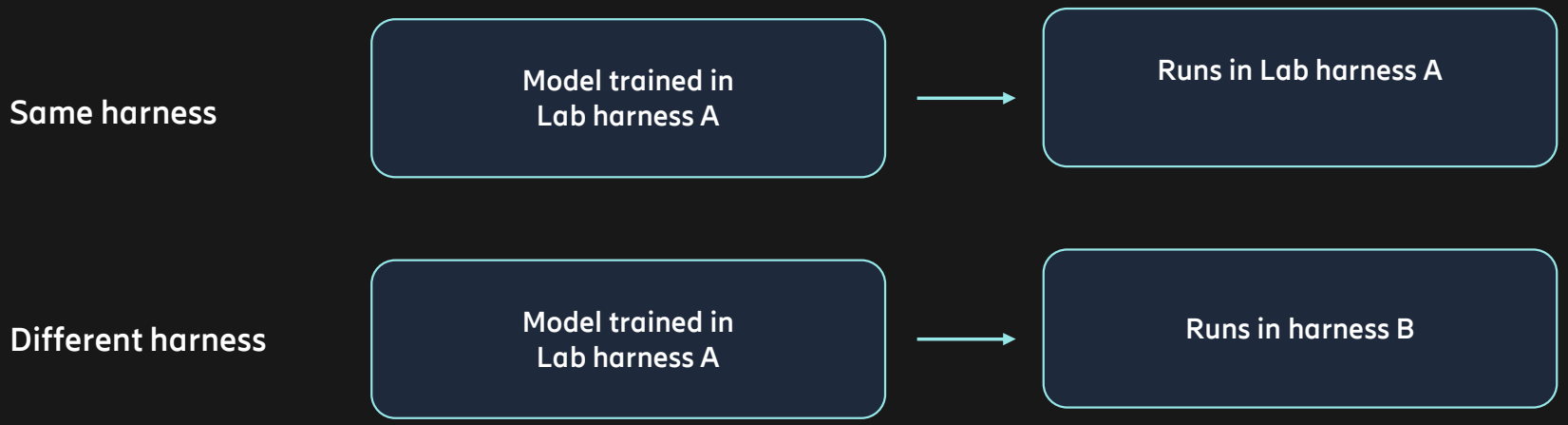
The Model: reasons about situation, decides actions

The Context: what the model sees

The Harness: program around the model, enabling agency



Training Inside The Harness, As A Loop



Sources: arxiv.org/abs/2610.00890 · arxiv.org/abs/2609.22247 (CHART) · cognition.com/blog/swe-1-5 · cursor.com/blog/composer · epoch.ai/benchmarks/swe-bench-verified · aider.chat/docs/more/edit-formats

Harness vs. Model: What the Evidence Supports

Both matter: changing the harness can change which model comes out ahead.

Harness and Context in Practice

OS sandboxing

Permissions and credentials

Hybrid model integration

Breakthrough detection

Data transfer approvals

Agentic enterprise access

Context engineering across levels

Workflow orchestration

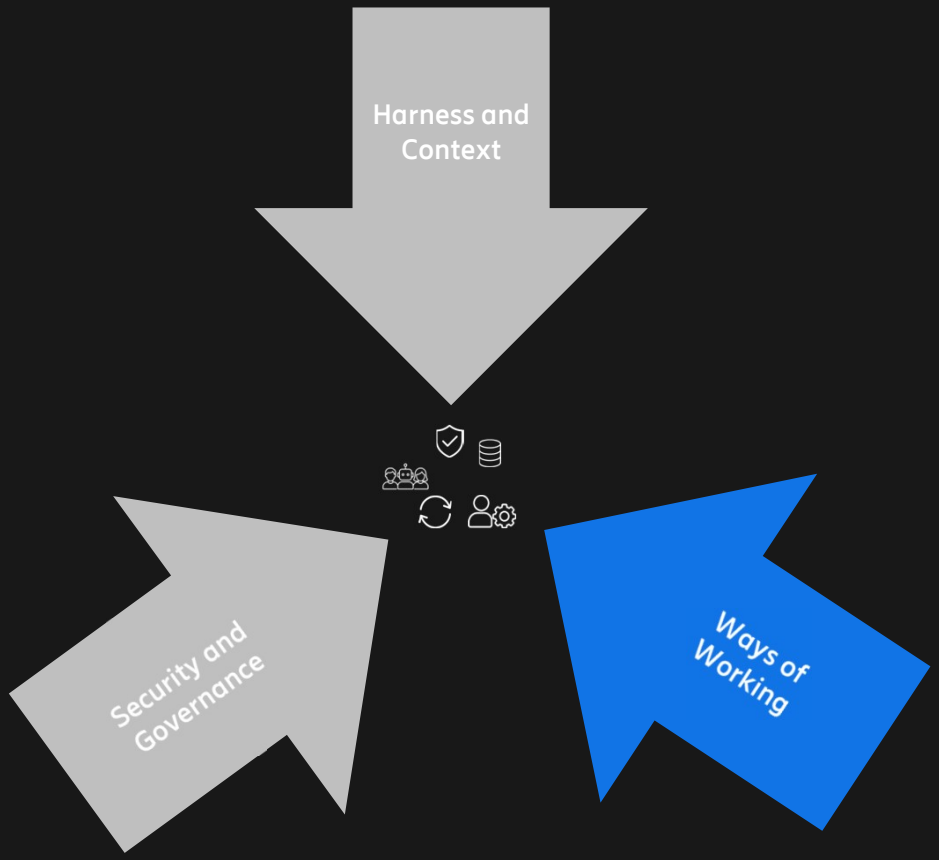
Discovery and reuse of assets

Multi-agent workspace

Agent observability

...

Three Pillars



Flash back to Feb 2026: Building a C compiler

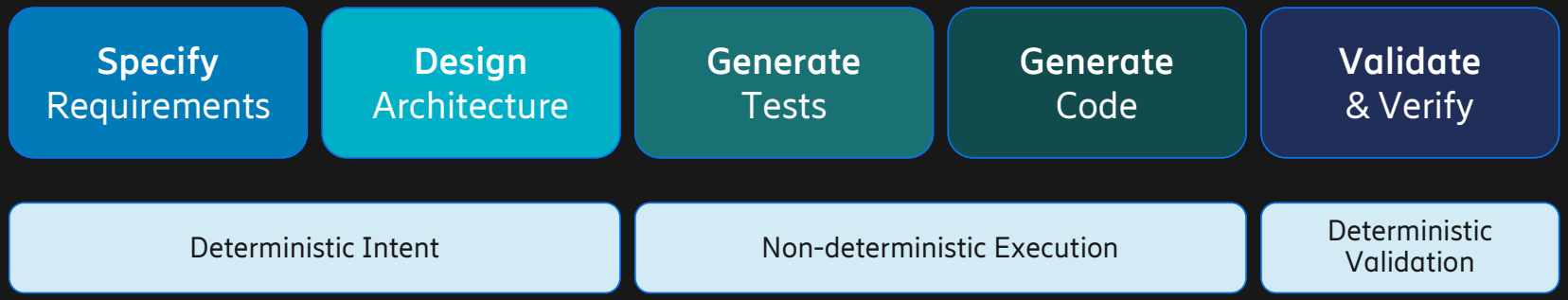
ERICSSON

1) Write extremely high-quality tests
2) …

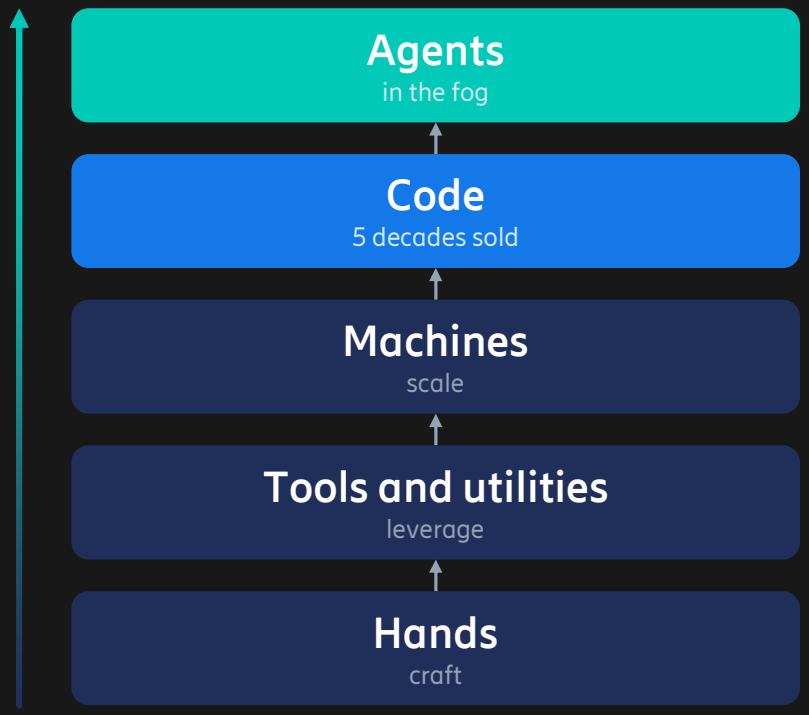
Claude will work autonomously to solve whatever problem I give it. Improving the testing harness required finding high-quality compiler test suites, … and watching for mistakes Claude was making, then designing new test.

Agents Fail on Ambiguity, Not Ability

~~*Traditional: Prompt → Code → Test → Hope it works*~~



Code: a How, Never a Why



The How keeps moving up

The shoe

Buyers want the “shoe”; only lately also the “how”.

Code

Code became our agreed translation of the How.

The Need stays

The How is shifting again. The Need stays.

Code Review- a Closer Look

ERICSSON

- Find and fix issues before they ship
  - CI took a big responsibility in that
- Shared context and codebase understanding

- You land it, you own it.

Source: https://claude.dev/blog/solving-code-review-at-anthropic/

Sources: claude.dev/blog/solving-code-review-at-anthropic (Osmani, Lunt, Wang, 15 Oct 2026) · claude.com/blog/how-anthropic-secures-its-ai-native-software-development-lifecycle

Life of a Change

Blue: agents act | White: deterministic gate

1 **Before The Pull Request**

- Agent runs tests
- Second agent review

2 **On Every Pull Request**

- Required CI
- Multi-agent review
- Security review gates
- Risk-based approval

3 **Merging**

- Merge queue re-checks main

4 **After The Merge**

- Behind a flag, internal first
- Gradual release
- Watch, fix or roll back

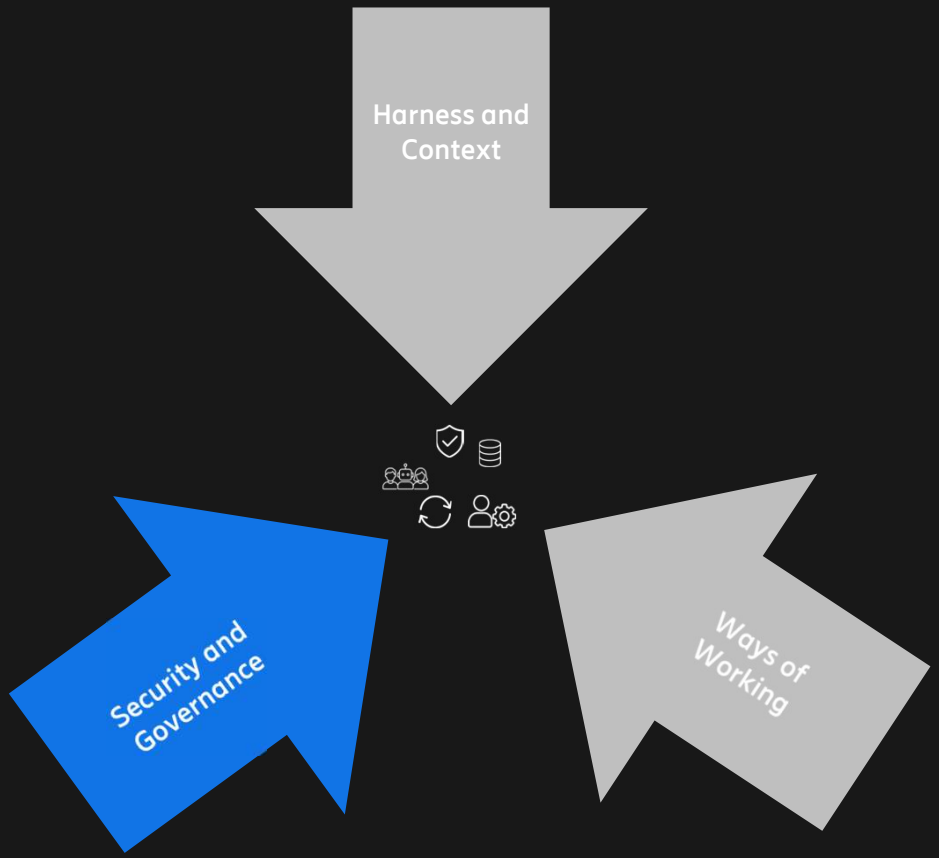
Gates that scale with agent output

- **Shift left:** verify before shared CI, where agents scale
- **CI and merge:** build the gates to carry the volume
- **Source control:** scale with the change, not the repo

Sources: https://www.anthropic.com/research/redesigning-source-contro, https://claude.dev/blog/solving-code-review-at-anthropic/

Sources: claude.com/blog/how-anthropic-secures-its-ai-native-software-development-lifecycle · anthropic.com/research/redesigning-source-control (Oct 2026) · code.claude.com/docs/en/code-review

Three Pillars



Business Risk Factors when Using ~~Gen~~AI

Deployment model: On-premises vs. SaaS

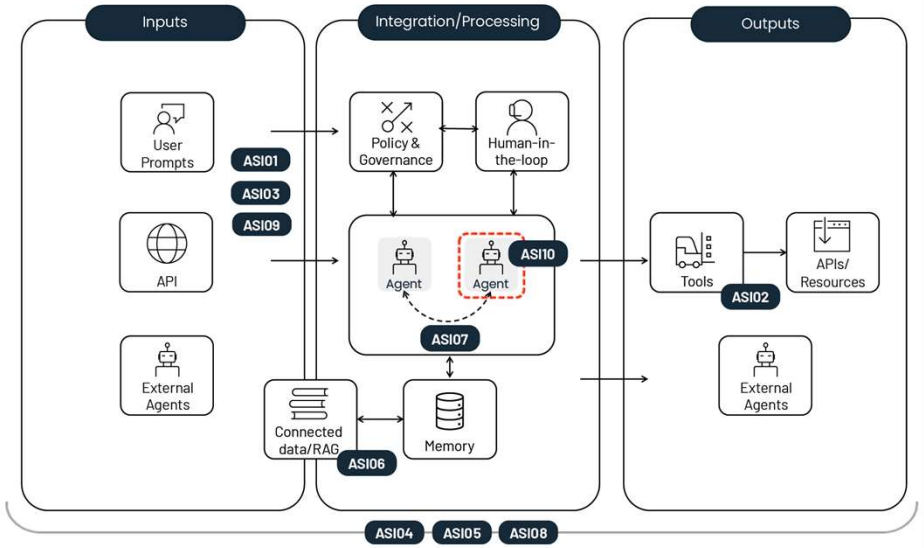
Usage context: Product type and intended usage

Geopolitics and trade compliance

Data: Does access mean permission?

Autonomy level: Human-~~in~~on-the-loop

Top 10 OWASP Threats for Agentic Applications



ASI01: Agent Goal Hijack

ASI02: Tool Misuse & Exploitation

ASI03: Identity & Privilege Abuse

ASI04: Agentic Supply Chain Vulnerabilities

ASI05: Unexpected Code Execution (RCE)

ASI06: Memory & Context Poisoning

ASI07: Insecure Inter-Agent Communication

ASI08: Cascading Failures

ASI09: Human-Agent Trust Exploitation

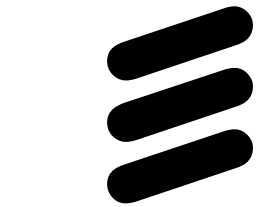
ASI10: Rogue Agents

Governing AI is a Journey



Thank You

Users, providers, each of us: together, we all own this.



ERICSSON