



Most of the architecture diagrams in this room are **lies**.

the system

the diagram

Drawn
2 October 2026

Started lying
nobody knows

A wrong diagram and a right diagram look identical.

WHERE THIS TALK ENDS

Can we replace the Rating Engine independently?



No. Mediation calls Rating Engine directly, and Rating Engine directly calls Invoicing; there is no public API boundary, so it is not independently replaceable.

An architecture question, answered with evidence.
Here is how it gets there.

Architecture is a Conversation, Not a Diagram.



Marco Nissen

Senior software developer · Telenor, Trondheim

Earlier: architect and senior developer · Lyse · Ice
Mobil

I helped build the thing

I am about to argue against.

BROKE HERE

- 1 Formal notation
- 2 Visual discipline
- 3 The job is the document

Week 1. Twelve diagrams. Six of fifteen people engaged. Linh knows what will fail. She says nothing.

BROKE HERE

4 Decisions made upfront

5 Documentation solves knowledge

Month 6. Outage. The diagram shows a system that is gone.
The person who knew left in March.

“The death of a program happens when the programmer team possessing its theory is dissolved.

Peter Naur · *Programming as Theory Building*, 1985, section “Program Life, Death, and Revival” · pages.cs.wisc.edu/~remzi/Naur.pdf

“... the expert developers working on that project have a shared understanding of the system design. This shared understanding is called ‘architecture.’

Ralph Johnson, quoted by Martin Fowler in *Who Needs an Architect?*, IEEE Software, July/August 2003, PDF p. 2–3 · martinfowler.com/ieeeSoftware/whoNeedsArchitect.pdf

“organizations which design systems (in the broad sense used here) are constrained to produce designs which are copies of the communication structures of these organizations.

Melvin E. Conway · *How Do Committees Invent?*, Datamation, April 1968, conclusion · melconway.com/Home/pdf/committees.pdf

Conversation first. Then as-is. Then to-be.

CONVERSATION

July 2025

Chat with the system instead of browsing it.

Then create and change it, like Visio or Miro.



AS-IS

Added next

Deployed instances

Source code and repos

Live traffic between services



TO-BE

Later

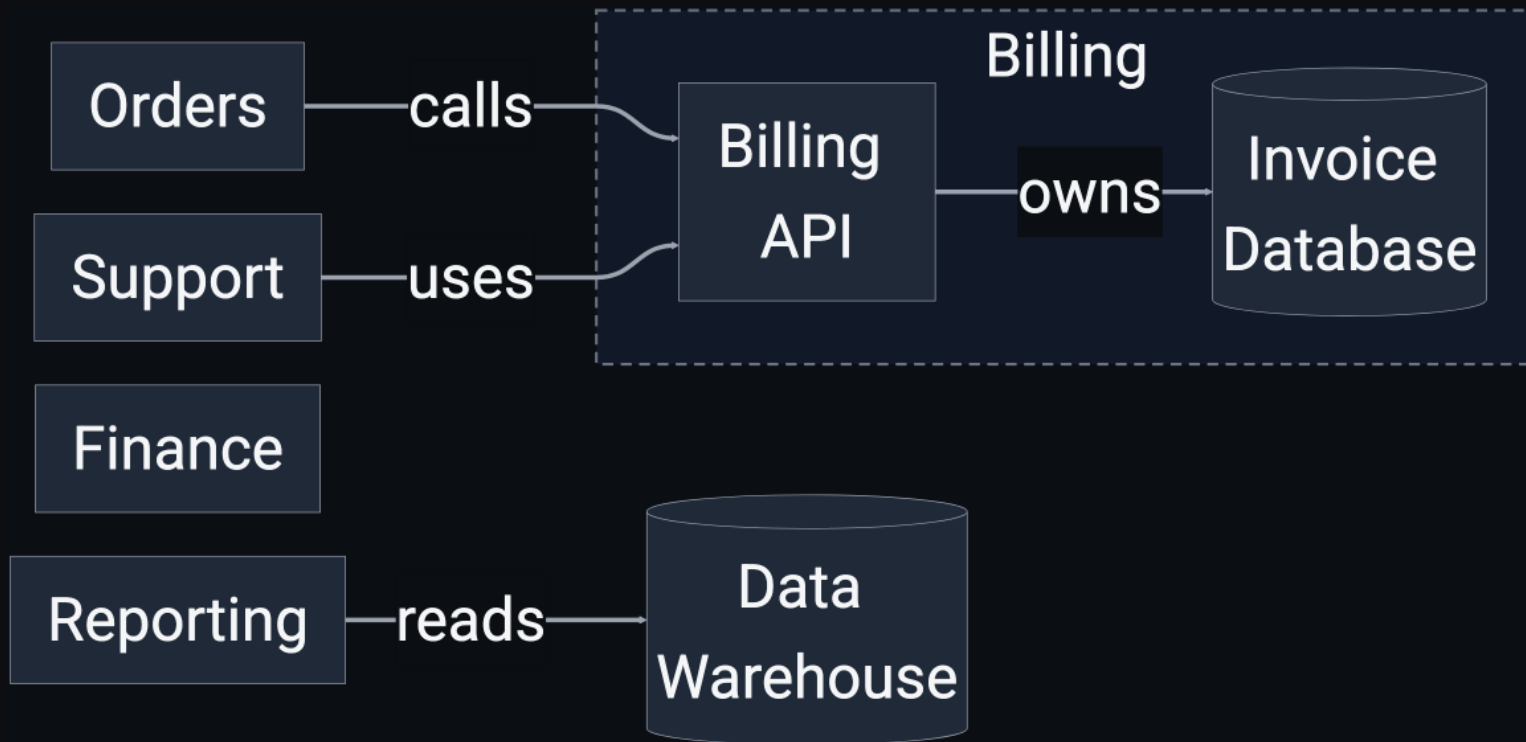
Enterprise architecture

The long-term view

AI adjustment pipeline

Nobody has all three for a live system yet.

1 What the team said: replaceable.



TO-BE · SAID IN THE ROOM

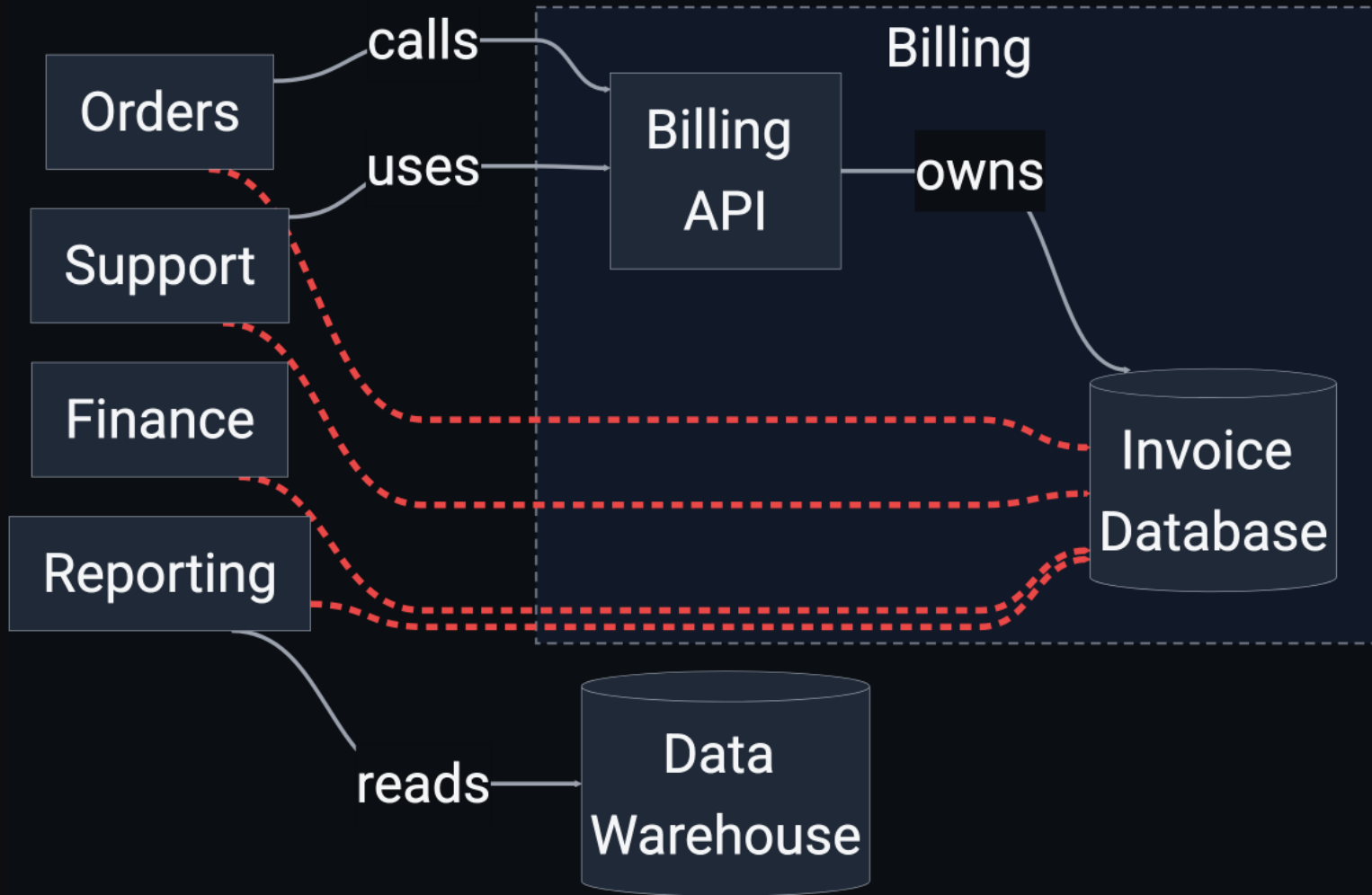
“Orders calls the Billing API.”

“Support uses the Billing API.”

“Finance has no link to Billing.”

Said in English, stored in the model, drawn by the generator.

2 What production shows: four hidden paths.



AS-IS · SEEN RUNNING

Distributed traces

Database audit logs

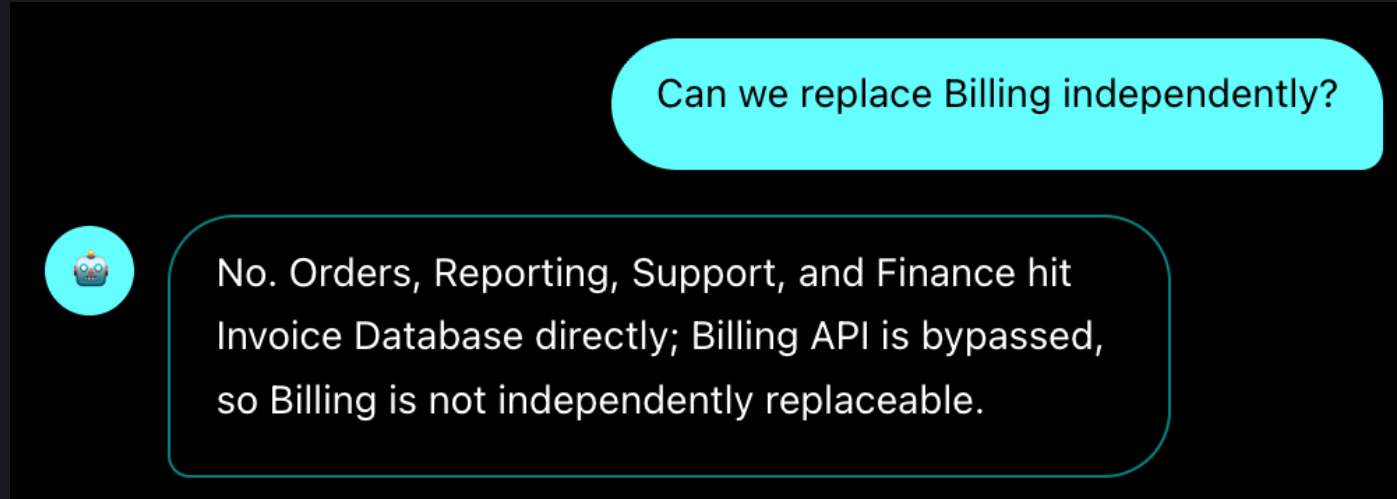
Nightly job schedule

Four new relations,
typed **observed**. None
on the diagram.

Here summarised by
hand. Automatic
ingestion is next.

3 Ask it.

CONVERSATION · ASK IT, IN PLAIN LANGUAGE



Billing is not independently replaceable.

The API boundary is fictional. This is what Linh knew and could not say.

A sentence becomes a test.

NEXT STEP · NOT BUILT YET

SAID

"Billing is
only reached
via its API."

CLAIM

Only Billing
API touches
Invoice DB.

CHECKED

Every trace:
who else
touched
Invoice DB?

RESULT

4 violations
Orders,
Reporting,
Support,
Finance

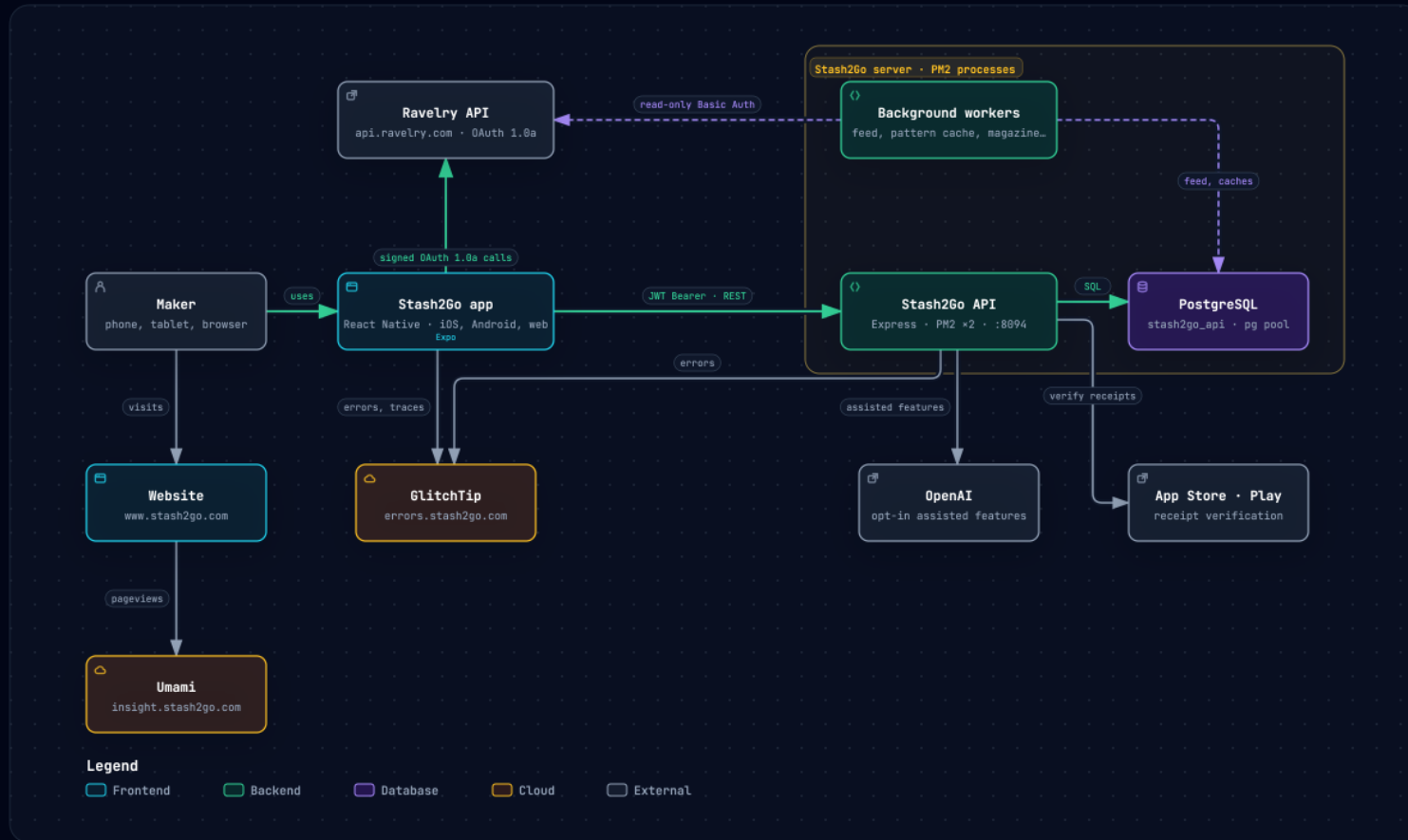
Coding agent
reads the claim first

A fitness function from what the
team said, run against
production.

A decision is a hypothesis. Now it can fail.

Generated diagrams are here.

• Stash2Go system architecture



STASH2GO · 16 OCT 2026

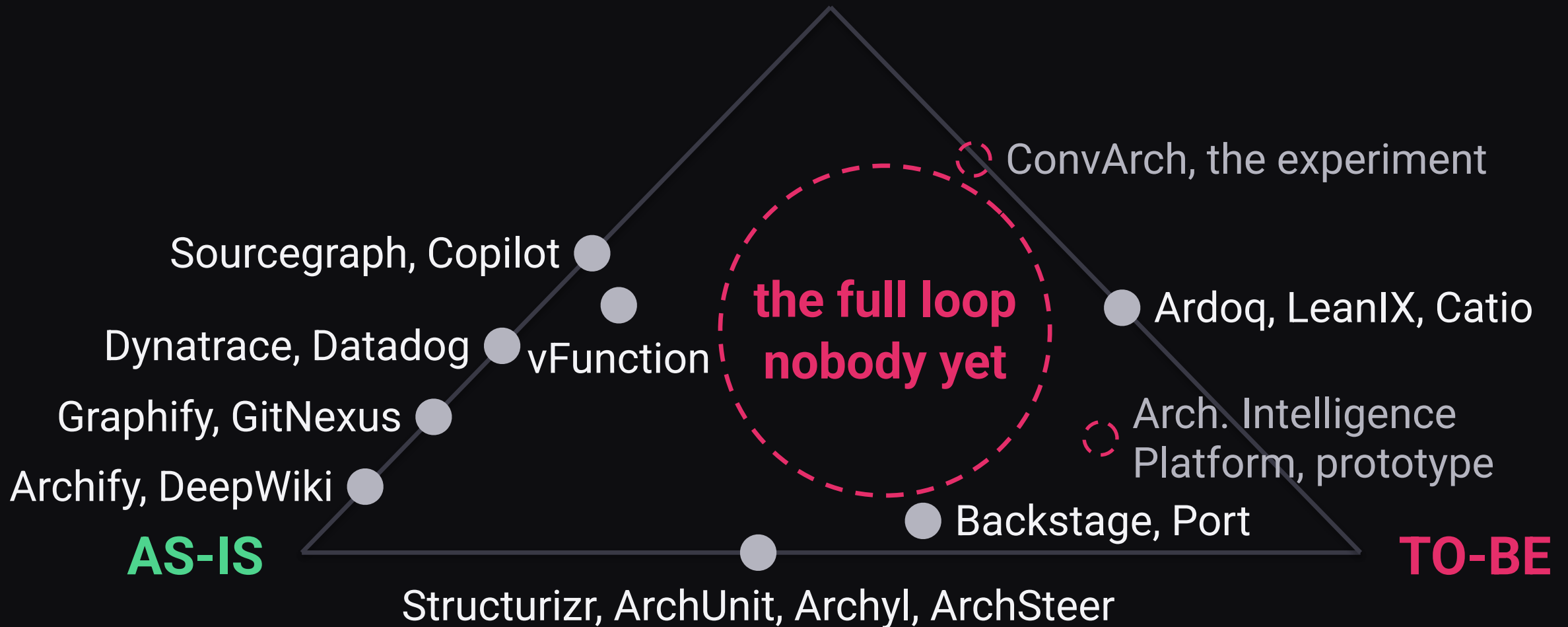
My knitting app. Archify, a Claude Code skill, read the repo and drew this.

It shows what the code says on one day.

Nothing about what runs. Nothing about what the team decided.

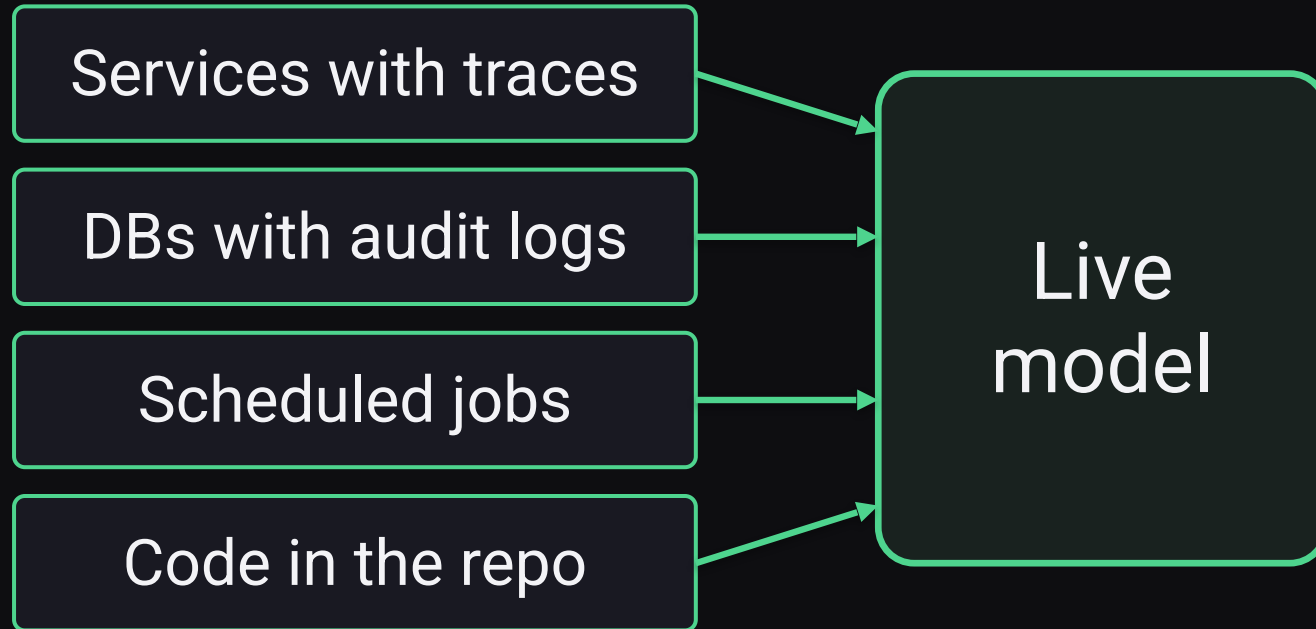
Where each tool sits. The centre is empty.

CONVERSATION

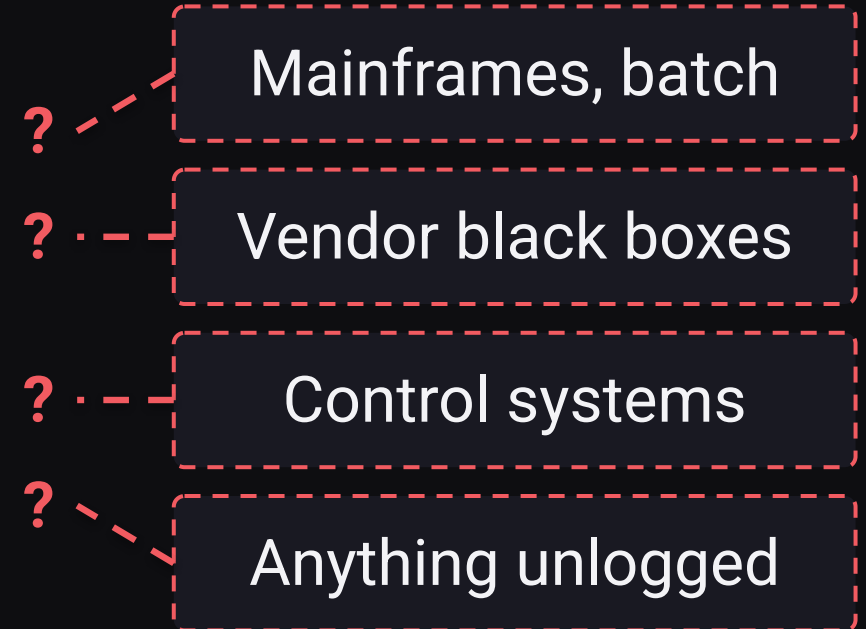


It only works where the system can be observed.

SEES WELL



STILL BLIND

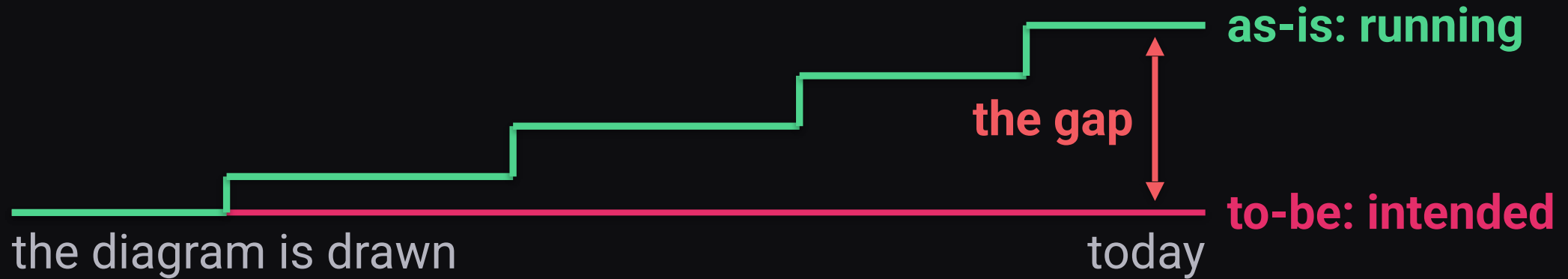


Treat the language model as a witness.

You ask. The evidence comes from the running system.

To-be stands still. As-is keeps moving.

Every step on the green line is a change nobody drew.



That gap is your technical debt.

WARD CUNNINGHAM, WHO COINED THE TERM · PARAPHRASED

Debt is code that no longer matches what the team understands.

Billing: believed replaceable. Running with four hidden paths.

TRY THIS WEEK · no tool needed

1 CONVERSATION

Record your next design meeting.

Twenty minutes of talk before the first box. Transcribe it.

2 AS-IS

Check one sentence against production.

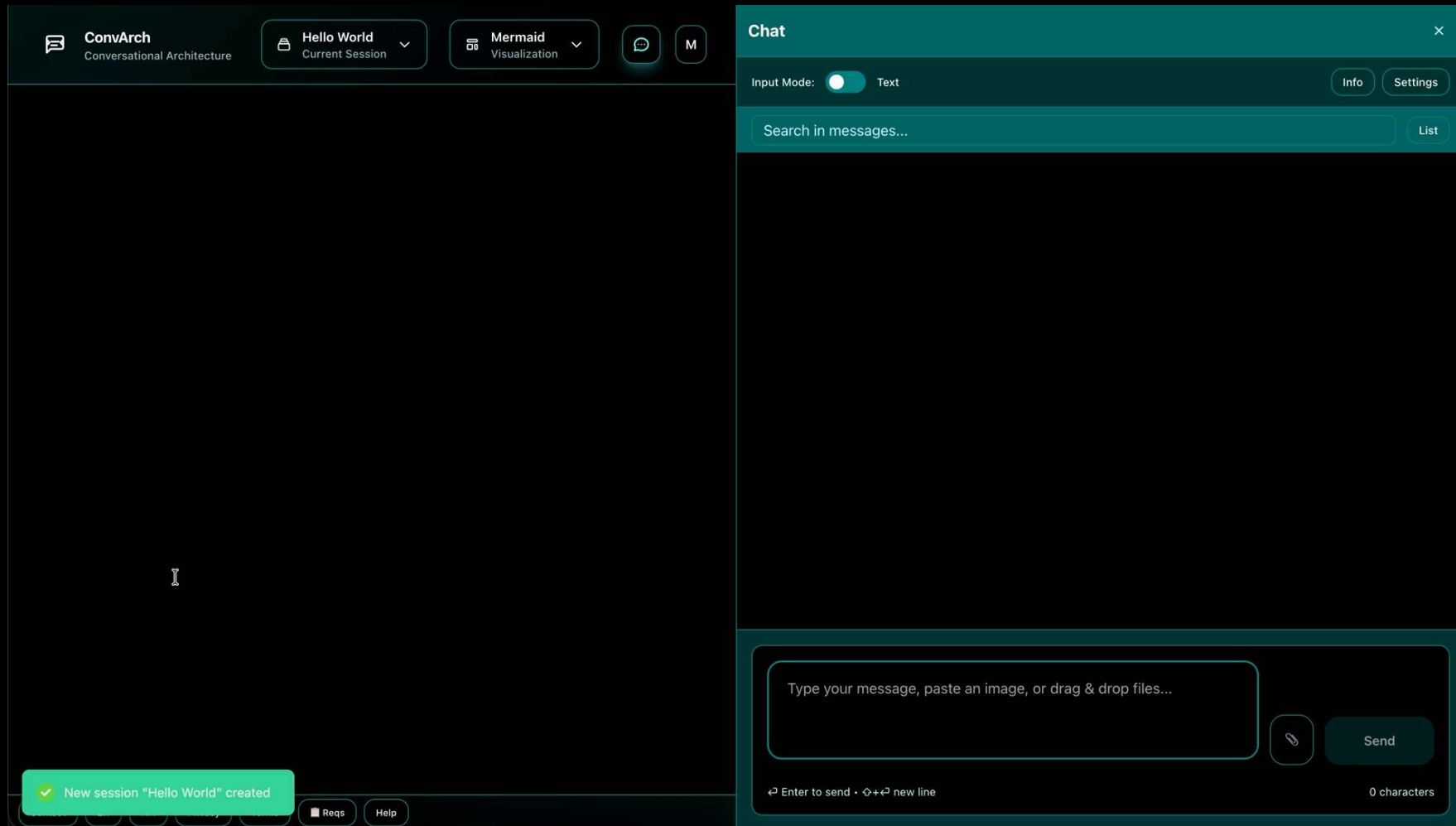
“Only Billing touches invoices.” Ask your traces who really did.

3 TO-BE

Keep decisions as sentences, not boxes.

A box cannot fail. A sentence can.

ConvArch, recorded 15 October 2026 · an experiment · raw take, 0:49



marco-nissen.com/tdc · the full film (3:55), slides and the demo

Stop documenting afterwards.

Build systems you can ask.

Marco Nissen · marco-nissen.com/tdc
slides, film, demo and the newsletter



Each tool covers a different part of the vision.

	<u>CONVERSATION</u>		<u>AS-IS</u>			<u>TO-BE</u>	
	Ask	Modify	Runs	Code	Deploy	Team	Enterpr.
vFunction	yes	part	yes	yes	—	part	—
Dynatrace, Datadog	yes	—	yes	—	yes	—	—
Backstage, Port	part	—	part	part	yes	part	—
Ardoq, LeanIX, Catio	yes	part	part	—	part	part	yes
Structurizr, ArchUnit	—	—	—	yes	—	yes	—
Archyl, ArchSteer	part	—	—	yes	—	yes	—
Sourcegraph, Copilot	yes	—	—	yes	—	—	—
Graphify, GitNexus	part	—	—	yes	—	—	—

yes = covers it today part = covers some of it — = not at all

No row is green across. Nobody has all three for a live system.

Architecture scanning skills: code in, picture or graph out.

Agent skills, MCP servers and web tools that read a repository and draw or graph it. GitHub stars, 16 Oct 2026.

Tool	What it is	Reads	Stars	Where it stops
Graphify	Skill · Claude Code, Cursor, Codex	code, docs, SQL, configs, PDFs	127,081	knowledge graph; code only
Archify	Agent skill · 4 coding agents	the repo, read by the agent	84,271	picture; no runtime, no intent
GitNexus	CLI + MCP · web UI	the repo, parsed (tree-sitter)	47,913	code graph for agents; code only
GitDiagram	Web tool · gitdiagram.com	a public GitHub repo	18,252	picture
DeepWiki (open clone)	Web tool; Cognition's is closed	a public repo	18,205	wiki with pictures
Cocoon diagram generator	Claude skill	the repo, read by the agent	7,449	static picture; idle since May
CodeFlow	Browser tool	JS, TS, Python files	5,421	file-level map
fallow	CLI · GitHub Action · LSP	TS and JS source	5,098	code graph and rules; code only
CodeBoarding	CLI · VS Code · GitHub Action	static analysis, LLM names	2,587	picture, PR impact; code only
Swark	VS Code extension	code, via the Copilot model	1,696	Mermaid; idle since Mar 2025
OpenTrace	MCP server + plugin	source, 12 languages	108	OSS code only; runtime promised
ADR Analysis Server	MCP server	ADRs and the code	35	doc vs code drift
Arch. Intelligence Platform	MCP server · Neo4j	OpenAPI, AsyncAPI, OTel	8	declared vs observed; no asking
Ridge	MCP server	Go, TS, Python source	7	code graph; drift between refs

If asked: They make the drawing cheap. Thirteen read the code, one reads traces. None stores what the team decided.

Sources: the GitHub repositories and READMEs, stars on 16 Oct 2026. Tried: Archify once, on Stash2Go (slide 13). The others: read, not run.

Market 2026: strong pieces, still no full loop.

Current offerings are getting closer, but they still cluster around separate strengths.

Architecture observability

vFunction · CAST · Lattix

vFunction now adds an agent, an architectural query engine, and chat over distributed apps built from runtime data. Still starts from the running system, not the design conversation.

Runtime truth

Dynatrace Smartscape · Datadog · New Relic

Dynatrace now pairs Smartscape, Assist, and MCP. Strong live topology and runtime truth. Weak on explicit design intent.

EA / architecture IDE

Ardoq AI workforce · Catio Archie (beta)

Ardoq now runs architecture agents over its documented graph; Catio lets coding agents query its model over MCP. Strong on plans and decisions; runtime truth is still partial.

Developer portals

Cortex EngOps · Port AI · OpsLevel

Port now supports AI agents and external MCP in chat and workflows. Cortex has moved toward Engineering Operations. Strong on ownership, standards, and action; weak on structural truth.

Architecture as code

ArchUnit · Structurizr vNext · CodeScene · Archyl · ArchSteer

Explicit rules and conformance checks. Archyl and ArchSteer now score drift between declared intent and code for coding agents over MCP. Runtime proof still lives elsewhere.

Code intelligence

Sourcegraph 7 · Copilot cloud agent · Meta Glean OSS

Sourcegraph 7 brought a versioned integrations API and MCP GA. Strong repo and agent context. Not the full intent-plus-runtime loop.

Bottom line: Ardoq and SAP LeanIX lead on the central graph; vFunction and Dynatrace on runtime truth. Small tools now check intent against code. No single product joins intent and runtime evidence into one queryable model.

Why now: five capabilities converged in 2024-2026.

In 2023 this stack was awkward. In 2026 it is becoming practical.

Codebase context

Copilot cloud agent deep research and Sourcegraph 7 now support repo-scale questions, not just file-local help.

Live tool access

MCP is now a foundation-backed tool standard, so systems like Port and Sourcegraph can be called instead of merely read.

Telemetry maturity

Dynatrace and Tempo normalize service graphs, and OpenTelemetry eBPF instrumentation lowers the cost of collecting runtime evidence.

Architecture intelligence

Ardoq, LeanIX, vFunction, and Catio now reason over architecture structure, not just tickets and code.

AI delivery pressure

Cloud agents and agentic autofix increase change volume, so static diagrams decay even faster.

Result: everyone is making existing graphs agent-callable, but the standing intent-versus-runtime loop is still missing.

Archyl: C4 documentation that scores its own drift.

"Architecture documentation that stays true to your code." Small, new, 2026.

Conversation
no

Declared intent
yes

Code
yes

Runtime
no

Agent access
yes

What it does

- AI discovers a C4 model from the Git repo; or declare it in archyl.yaml.
- Drift Score 0–100: how much of the documented model still matches the code.
- Runs as a CI gate; fails the build below a threshold.
- MCP server: coding agents read the model, ADRs and guardrails.

What it does not do

- No runtime evidence: analysis is structural; behaviour is listed as future work.
- Drift means "doc vs. code", not "what runs vs. what we meant".
- Does not start from the design conversation.
- Vendor claims only; no independent evaluation found.

If asked: Archyl keeps the diagram honest about the code. The running system and the conversation are still missing.

ArchSteer: steer coding agents with declared intent.

"Your agents ship faster than you can govern." Small, new, launched mid-2026.

Conversation
no

Declared intent
yes

Code
yes

Runtime
no

Agent access
yes

What it does

- Derives an architecture model from code; zero config (archsteer xray).
- You write one invariant in architecture.yaml; it is enforced as a fitness function.
- Injects the rule into CLAUDE.md / AGENTS.md; ratchet blocks new drift only.
- Local MCP server and Claude Code plugin; Drift Index tracked over time.

What it does not do

- No runtime evidence: the model comes from source code only.
- Intent is a rule file someone writes, not the team conversation.
- Governs agents at edit time; does not answer "can we replace Billing?" from evidence.
- Vendor claims only; no independent evaluation found.

If asked: ArchSteer checks the code against what you declared. The running system and the conversation are still missing.