# The Infrastructure Factory

Software factories need infrastructure factories

Max Körbächer · infrastructure-factory.com · October 2026

Everyone is building software factories.

Nobody has built the thing they run

THE SHIFT

# Code got cheap. The environments it needs still move through tickets.

## 75%

of new code at Google is AI-generated and approved by engineers.

Sundar Pichai, Google Cloud Next, April 2026

## +441%

median time a change spends in review, while throughput per developer rose 33.7%.

Faros AI, April 2026

## −7%

main-branch throughput for the median team, while daily pipeline runs rose 59%.

CircleCI, February 2026

All three figures are reported by the organization that benefits from them. Read them as direction, not precision. Sources 4, 7 and 8 at infrastructure-factory.com/sources

# What thirteen sources agree on

> A codified, largely automated production system that carries work through the entire software lifecycle, from incoming signals through planning, implementation, verification and release to monitoring. Agents do most of the execution inside standardized workflows and quality gates. Humans set intent and approve at defined checkpoints.

Engineers build and tune the factory. The factory builds the product.

Consensus definition from thirteen practitioner sources: infrastructure-factory.com/software-factory

# The queue is the thesis

## Software lane

Agents write the change, tests run, review happens in minutes. Pull requests ship at agent speed, around the clock.

## Environment lane

Each change needs an environment with an owner, a network position and a budget. That request stops at a gate a person opens a few times a day.

Both lanes receive the same work at the same rate. One ships. One waits.

Live two-lane model with the policy-check switch: infrastructure-factory.com/evidence

# An Infrastructure Factory is

> A platform that turns intents from humans or agents into compliant, owned, observable, cost-attributed environments, with no ticket in the path.

**Input: an intent.** What the environment is for, who owns it, what data it touches, how long it lives.

**Output: an environment.** Already owned, already compliant, already observable, already attributed to a cost center.

# Six words carry the weight

| WORD | WHAT IT RULES OUT |
| --- | --- |
| Intents | Resource lists. The request names a purpose, not a machine size. |
| Compliant | Compliance checked afterwards. Policy runs before anything is built. |
| Owned | Orphans. Cost center, compliance owner and on-call are inherited. |
| Observable | Dark environments. The factory can see what it made. |
| Cost-attributed | Unassigned spend. Every environment has a budget line from birth. |
| No ticket in the path | People deciding per request. They decide once, at the contract. |

# Infrastructure plays three roles in a software factory

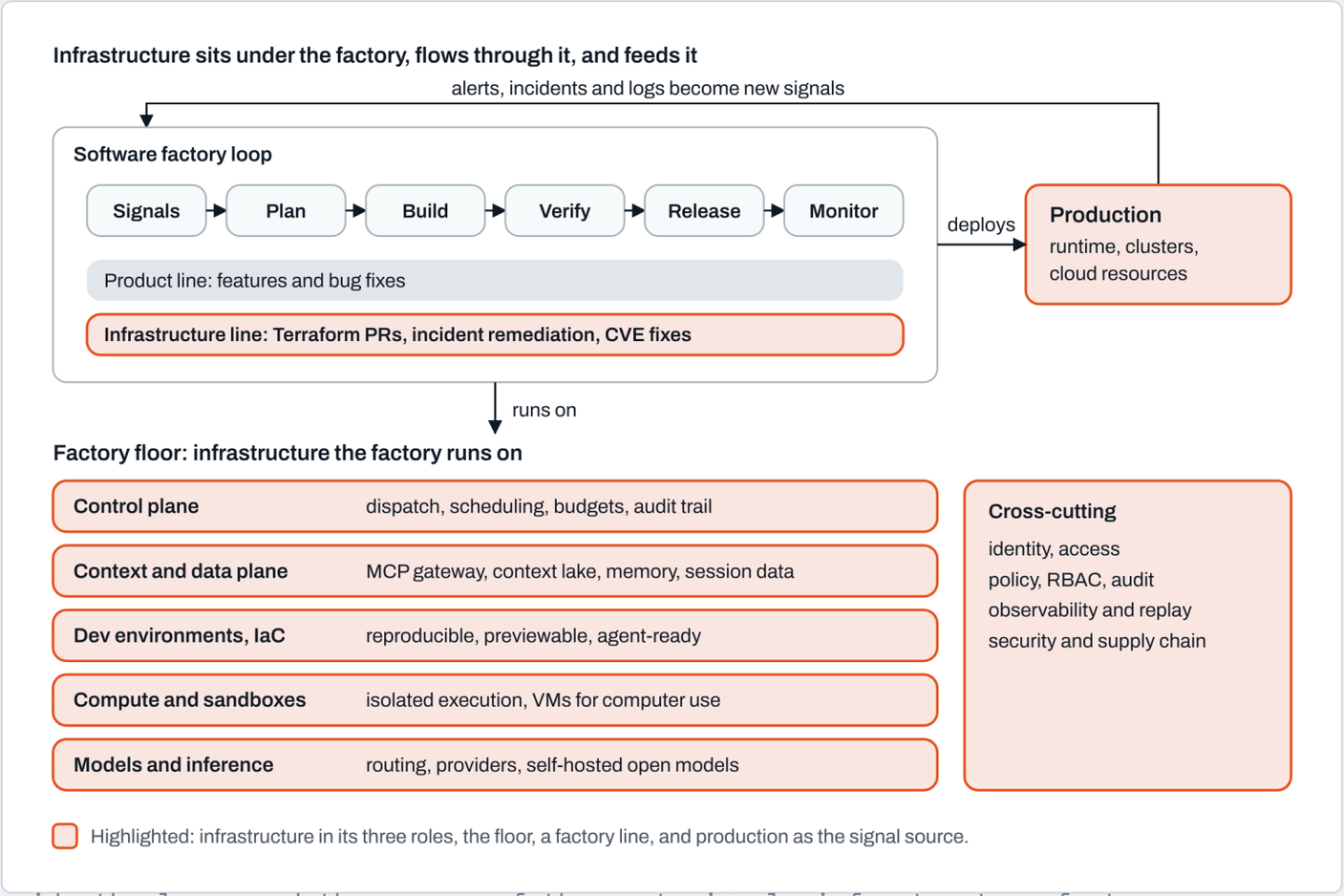



Diagram D1. Under the loop, inside the loop, and the source of the next signal. infrastructure-factory.com

# Not NVIDIA's AI factory, but it overlaps

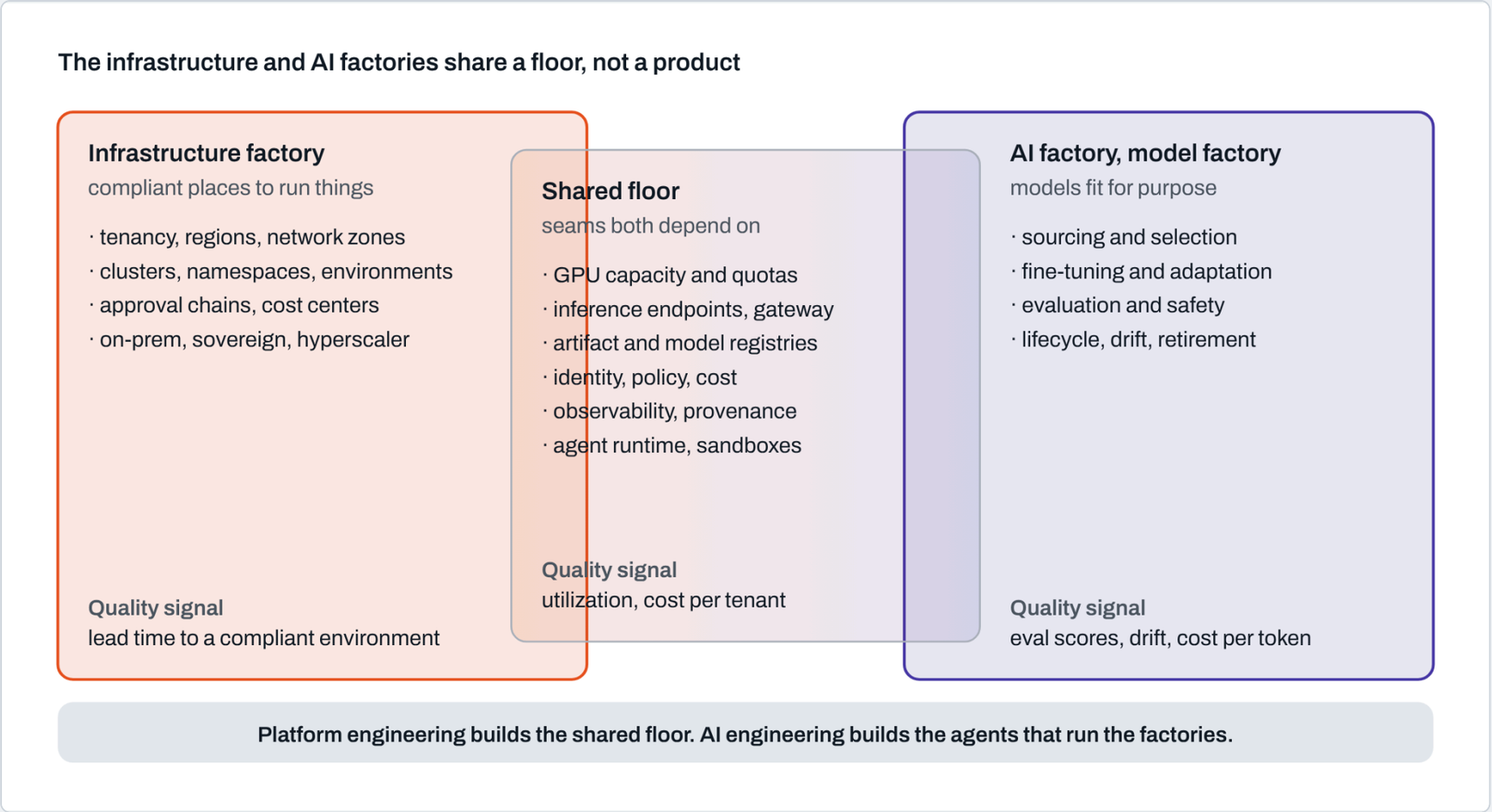



Diagram D3. The AI factory produces models. The infrastructure factory produces environments. Every shared capability has exactly one owner. Source 48.

# Each of these is at most one part of it

- **An account vending machine.** Vending is one station. Without contracts, policy and ownership it vends orphans faster.
- **A developer portal.** A portal is a front door. The factory is what happens after the form is submitted, without a person.
- **An infrastructure-as-code generator.** Generated Terraform is still a pull request that waits for someone to plan, review and apply it.
- **An AI factory.** That one produces models. This one produces the places they run.

# Seven stations, then the loop closes

1. Intent arrives. Purpose, owner, data, lifetime, from a person or an agent.
2. Golden-path contract. The intent is matched to a contract the platform stands behind.
3. Policy pre-check. Allow or deny, with a reason the agent can act on.
4. Vend. Accounts, projects, namespaces, networks, identities.
5. Execution sandbox. Untrusted code runs isolated from the start.
6. Attach what humans added by hand. Owner, cost center, monitoring, backup, on-call.
7. Hand over, then reclaim. Everything expires. Renewal is a decision, not a default.

Station eight: what the environment reports back becomes the next intent.

# The hard part is ownership, not Terraform

Four fields no agent can sign:

- Cost center
- Compliance owner
- Data classification
- On-call owner

Encode them once: defaults by team and purpose, delegations with limits, policies that explain a denial. After that, a request at any hour is served without a person in the path.

# The cloud is the easy case

| TERRAIN | WHAT CHANGES ON THE LINE |
| --- | --- |
| Hyperscaler | Elastic capacity behind an API. The factory is mostly contracts, policy and ownership. |
| Virtualization estates | Capacity is finite and shared. Vending has to check inventory before it promises. |
| Bare metal and edge | Lead times, physical placement, hardware lifecycles. Capacity is inventory, and the factory manages stock. |
| Sovereign and regulated | Evidence ships with the product: residency, attestation, audit trails that a regulator accepts. |

Terrain analysis with NIST's five cloud characteristics as the yardstick: infrastructure-factory.com/evidence

# Measure it like a factory

- **Lead time** to a compliant environment
- **Zero-touch rate**: environments delivered with no human in the path
- **Human interventions** per environment
- **Time to owner** and **orphan rate**

**Little's law.** Environments waiting = request rate × wait time. The request rate is set by the agents. The wait time is yours.

**Outcomes, not tokens.** Count environments delivered and owned, not model calls made.

Six metrics with their software-side counterparts and a Little's law calculator: infrastructure-factory.com/metrics

# Twelve factors of an Infrastructure Factory

1. Intent in
2. Environment out
3. Machine-readable contracts
4. No ticket in the path
5. Policy before provisioning
6. Ownership by default
7. Scoped identity for every requester
8. Everything expires
9. Evidence ships with the product
10. Capacity is inventory
11. The factory measures itself
12. Built from replaceable parts

Each factor explained: infrastructure-factory.com/infrastructure-factory/#factors

# Most teams are somewhere on this ladder

| 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Ticket desk | Automated fulfilment | Self-service portal | Contracted API | Infrastructure factory |

Find your stage, then read what the next one removes from the path. A ticket desk automates nothing. Fulfilment automates the work but not the decision. A portal moves the form, not the wait. A contracted API removes the person from the request. The factory removes the person from the path and measures itself.

# Start with the sandboxes

1. Ephemeral sandboxes for coding agents. The demand exists already: sandboxes on Google Kubernetes Engine grew 16x in under five months after the Agent Sandbox preview. A sandbox needs an identity, a lifetime and a policy, and nothing a human has to sign.
2. Preview environments per pull request. Ownership defaults and cost attribution get exercised at volume.
3. Production vending. By then the contracts, policies and inventory already exist.

Source 9 at infrastructure-factory.com/sources. The three lines in order: infrastructure-factory.com/infrastructure-factory/#start

# Argue with it

The concept, the evidence, the twelve factors and every source: **infrastructure-factory.com**

Corrections and counter-examples: **github.com/mkorbi/infrastructure-factory**