The Infrastructure Factory

A platform that turns requests from humans or agents into compliant, observable environments with clear cost attribution

Everyone is building software factories.

Nobody has built the thing they run

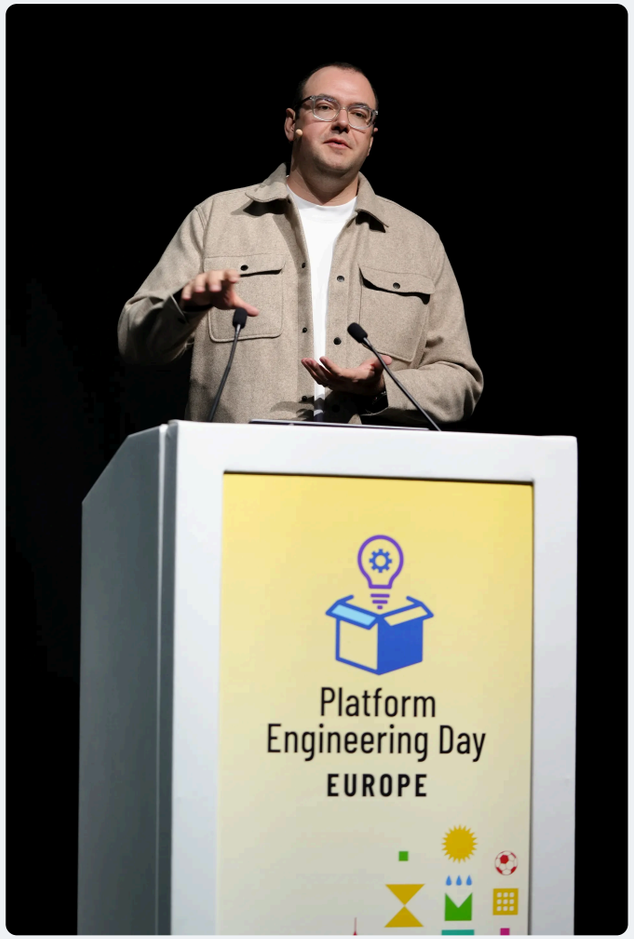
How these slides were made

- Concept: based on experience, research and implementations - humanly crafted
- Website/Slides: based on concept - AI work via Claude
- Updates/Continuous Research: scheduled AI job, human input (e.g. from conferences, workshops)
- Images/Diagrams: Described, sketched by hand - AI generated

ABOUT ME

Max Körbächer

- **Founder at Liquid Reply.** Platform and AI strategy with leadership and engineering teams.
- **Community.** CNCF Governing Board member, Linux Foundation Europe Advisory Board, CNCF and AAIF Ambassador.
- **Speaker.** KubeCon, PlatformCon and beyond, more than 80 talks.
- **Author.** "Platform Engineering for Architects" and the newsletter "Zeitgeist of Bytes".
- **Topics.** Platform engineering, AI engineering, cloud native, digital sovereignty, open source.



max.koerbaecher.io

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

Different speed

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.

An Infrastructure Factory is

> A platform that turns intents from humans or agents autonomously 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.

Infrastructure interacts with the software factory multiple times

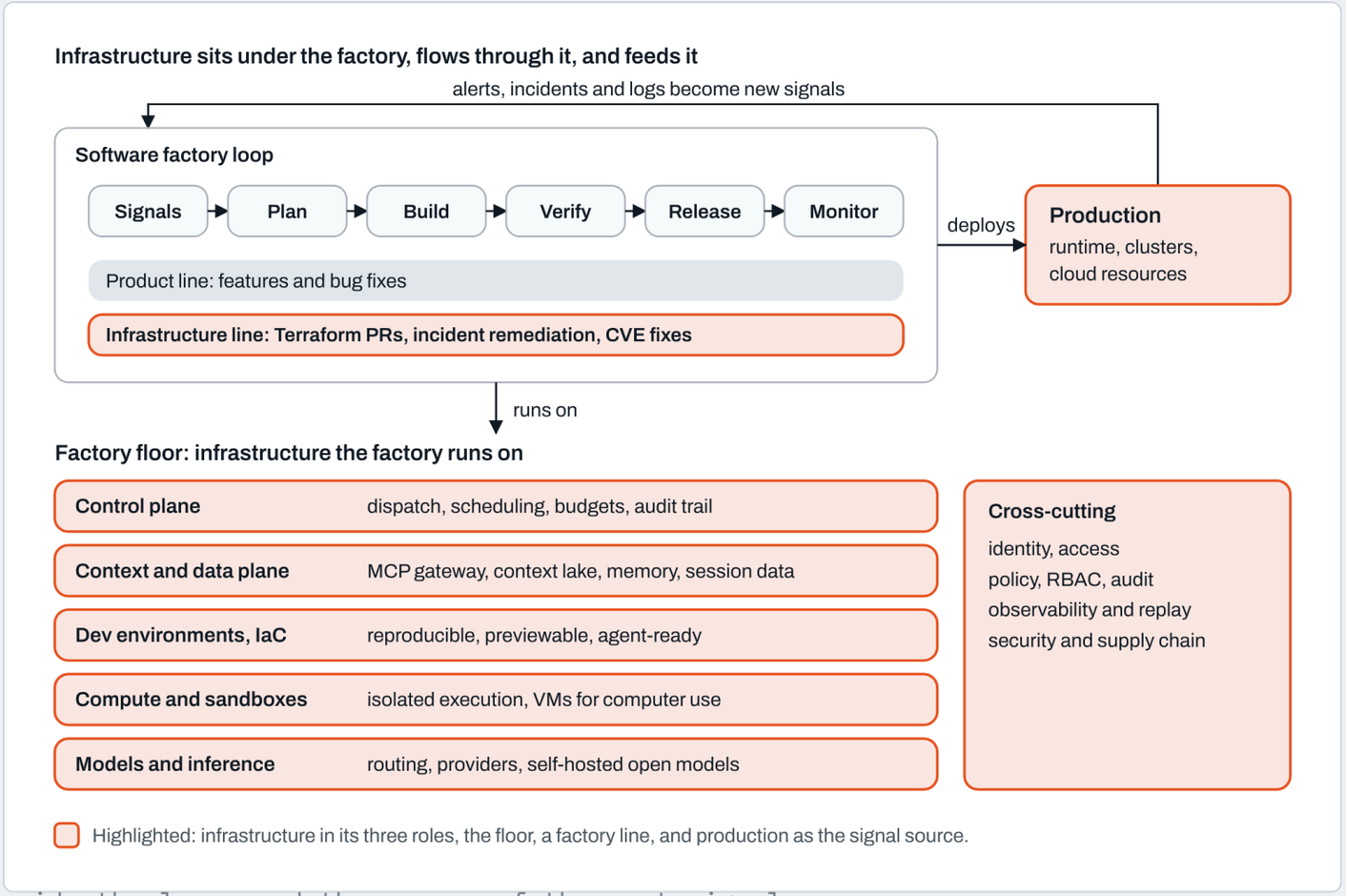
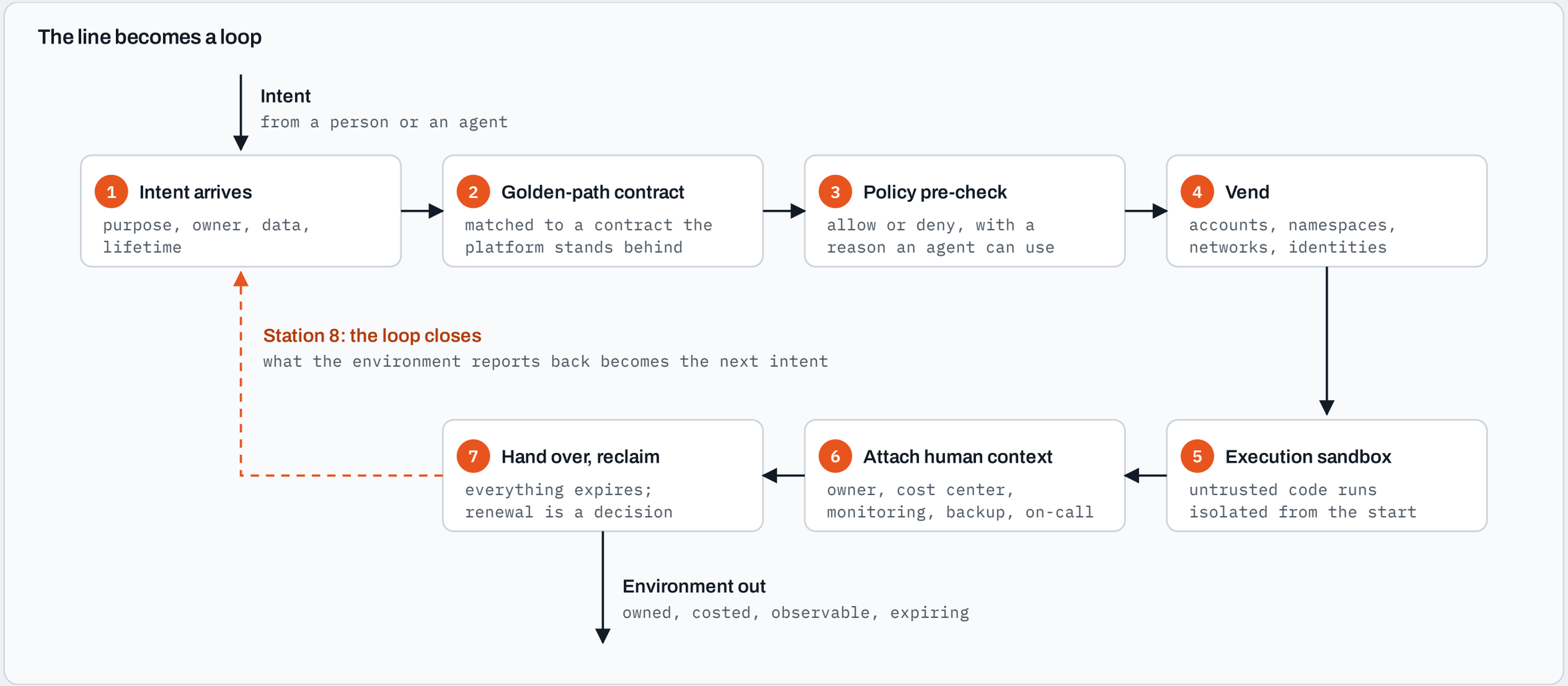


Diagram D1. Under the loop, inside the loop, and the source of the next signal.

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 for people. 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.

From station to station until the loop closes

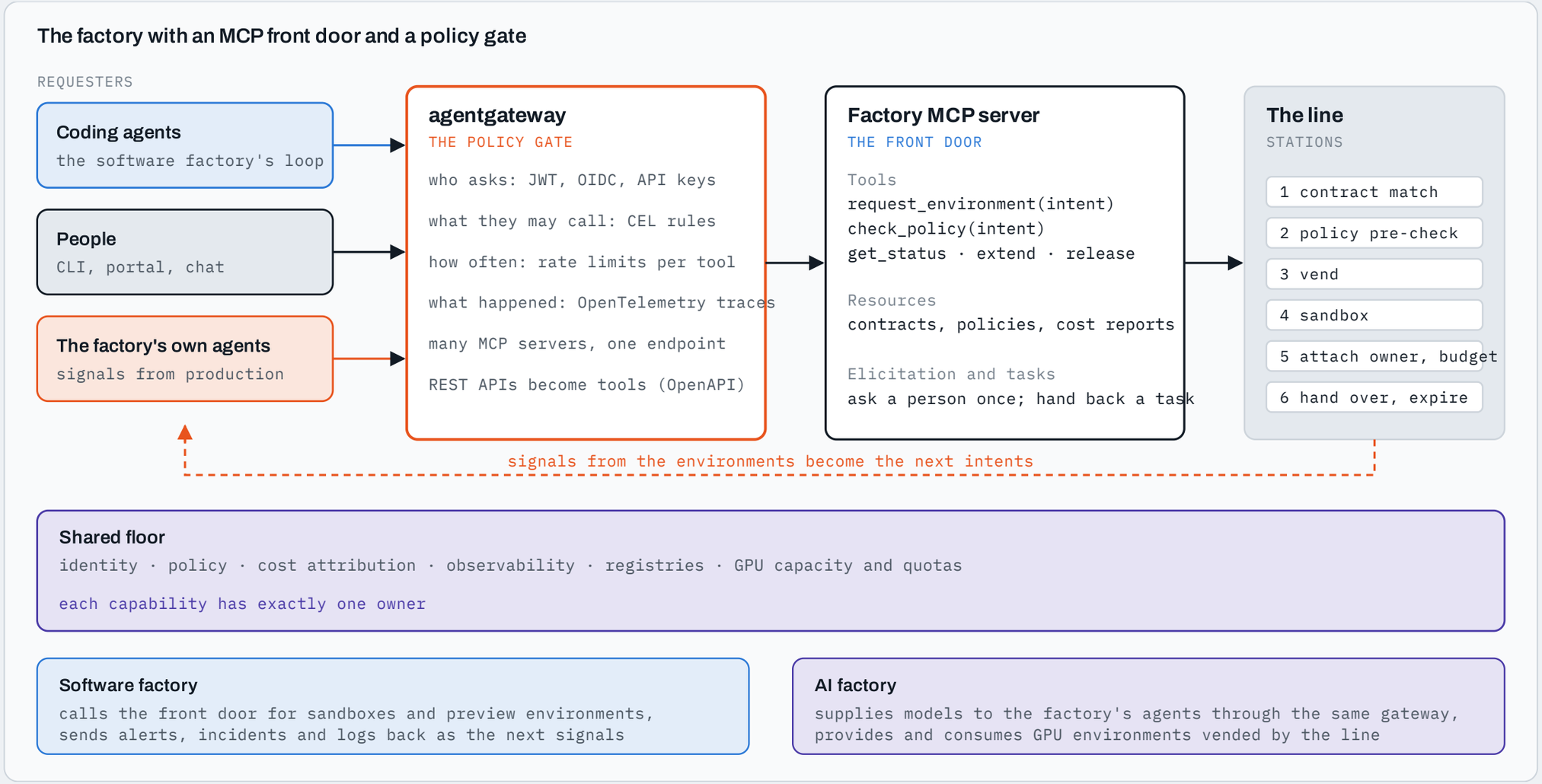


The architecture:

an entry point for agents, a gateway that understand them, a production

line, and a floor

Architecture



MCP as the portier

What the factory exposes

- **Tools**: request_environment, check_policy, get_status, extend, release
- **Resources**: the contract catalog, the policies, cost reports per owner
- **Prompts**: "request a sandbox for this repository", the golden path as a conversation starter

Why MCP

- Coding agents already speak it. The factory joins the tool set they already have.
- A tool is a contract: typed input, typed output, a description that a model and a person both read.
- Open governance: MCP moved to the Linux Foundation's Agentic AI Foundation in December 2025.
- The 2026-07-28 revision made the core stateless: no handshake, no protocol session. The front door scales like any HTTP API.

modelcontextprotocol.io specification 2026-07-28

Important protocol features for the factory

Elicitation

The server asks the person a question through the agent's client: which cost center? It asks once, and the answer becomes the team's default. The ticket's last remaining job, done inside the protocol.

Tasks

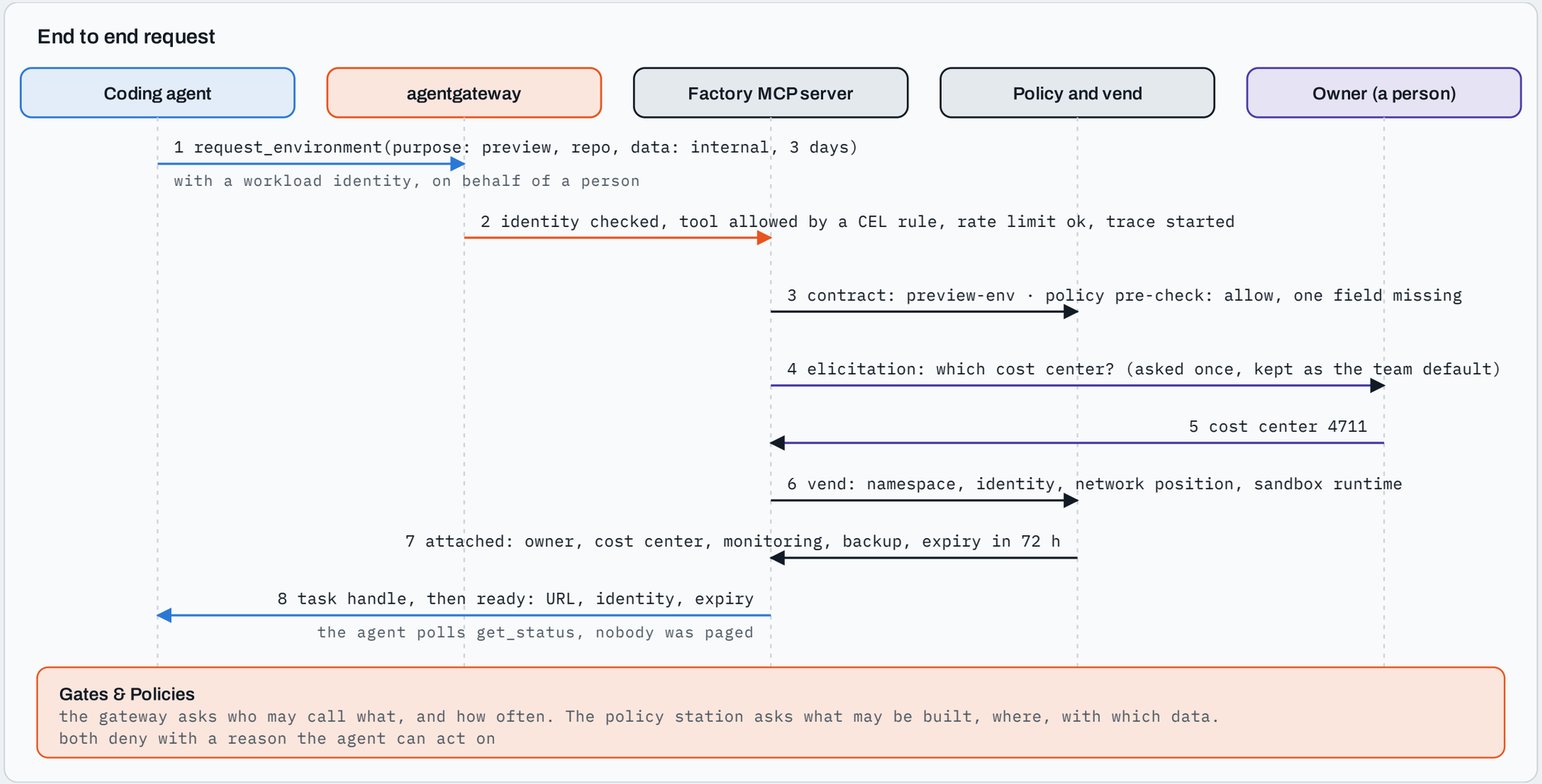
Vending takes minutes. Asynchronous task execution; the agent polls and gets the result when the environment exists. An official MCP extension since the 2026-07-28 revision.

```
{
  "tool": "request_environment",
  "arguments": {
    "purpose": "preview",
    "repo": "shop/checkout",
    "data": "internal",
    "lifetime": "72h",
    "on_behalf_of": "ana@example.com"
  }
}
```

```
{
  "task": "env-7f3a",
  "status": "working",
  "statusMessage": "policy ok, vending namespace"
}
```

Elicitation and the Tasks extension: modelcontextprotocol.io, specification 2026-07-28. The payloads are illustrative.

End to end request



agentgateway: the gate agents already pass through

What it is

An open source proxy for agent traffic: MCP, A2A and LLM calls through one data plane. Rust, Apache 2.0. Created by Solo.io, a Linux Foundation project since August 2025, hosted by the Agentic AI Foundation since June 2026. One binary with a YAML file, or Kubernetes with the Gateway API.



What it does at the factory's gate

- **Identity**: JWT, API keys, OAuth. Every call says who asks.
- **Authorization**: CEL rules per tool and per backend.
- **Rate limits**: per tool, per session.
- **Observability**: OpenTelemetry on every call.
- **Federation**: many MCP servers, one endpoint.
- **OpenAPI**: a REST API becomes MCP tools, no rewrite.

agentgateway.dev docs and logo, github.com/agentgateway/agentgateway, aaif.io/projects/agentgateway

A rule the agent can read

```
mcp:
  policies:
    mcpAuthentication:
      mode: optional
      issuer: http://localhost:9000
      audiences:
      - http://localhost:3000/mcp
      jwks:
        url: http://localhost:9000/.well-known/jwks.json
    mcpAuthorization:
      rules:
      # Allow anyone to access tools on the public-tools target
      - 'mcp.tool.target == "public-tools"'
      # Only authenticated admins can access tools on the admin-tools target
      - 'mcp.tool.target == "admin-tools" && has(jwt.sub) && "admin" in jwt.roles'
```

The token is verified, a rule decides whether this identity may call this tool, and the rate limit decides whether it may call it now. A denial carries a reason the agent can act on. A tool the caller may not use is left out of the tool list.

agentgateway.dev docs, MCP authorization

Technically one gate, practically an onion style is better

At the gateway: who may call what, and how often

The identity of the agent and of the person behind it. Tool-level rules. Rate limits. A trace for every call. This gate knows nothing about regions or data classes, and it should not.

At the policy station: what may be built

Region and residency, data classification, network position, cost ceiling, lifetime. Evaluated before anything is provisioned, by an engine: OPA, Kyverno, CEL. Allow, or deny with a reason.

Both answers are machine-readable. An agent that is denied changes its request and asks again. Nobody opens a ticket, and nobody reads one.

Engines and agents

Vend

Crossplane compositions, Terraform or OpenTofu modules, Kubernetes namespaces. For untrusted code: agent-sandbox, the Kubernetes SIG Apps project, with gVisor or Kata Containers underneath.

Attach

What people used to add by hand: owner, cost center, monitoring, backup, on-call. Inherited from the contract and the team's defaults, applied by the engine, never typed.

Expire

Everything has a lifetime. Renewal is a decision an owner makes when asked, not a default. Orphans cannot form, because nothing is born without an owner or lives without an end.

agent-sandbox: github.com/kubernetes-sigs/agent-sandbox

Every call traced, every environment attributed

At the gate

OpenTelemetry on every tool call: who called, which tool, how long, allowed or denied, and why. The audit trail is a by-product of the gateway, not a project of its own.

On the floor

Owner and cost center are labels applied at birth, so cost reports need no reconciliation. The factory publishes them as MCP resources: an agent can ask what its sandboxes cost this week.

The factory measures itself

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

Published to the factory's customers: the teams and the agents that use it.

Each metrics has their software-side counterparts: infrastructure-factory.com/metrics

The factories are overlapping (share the floor)

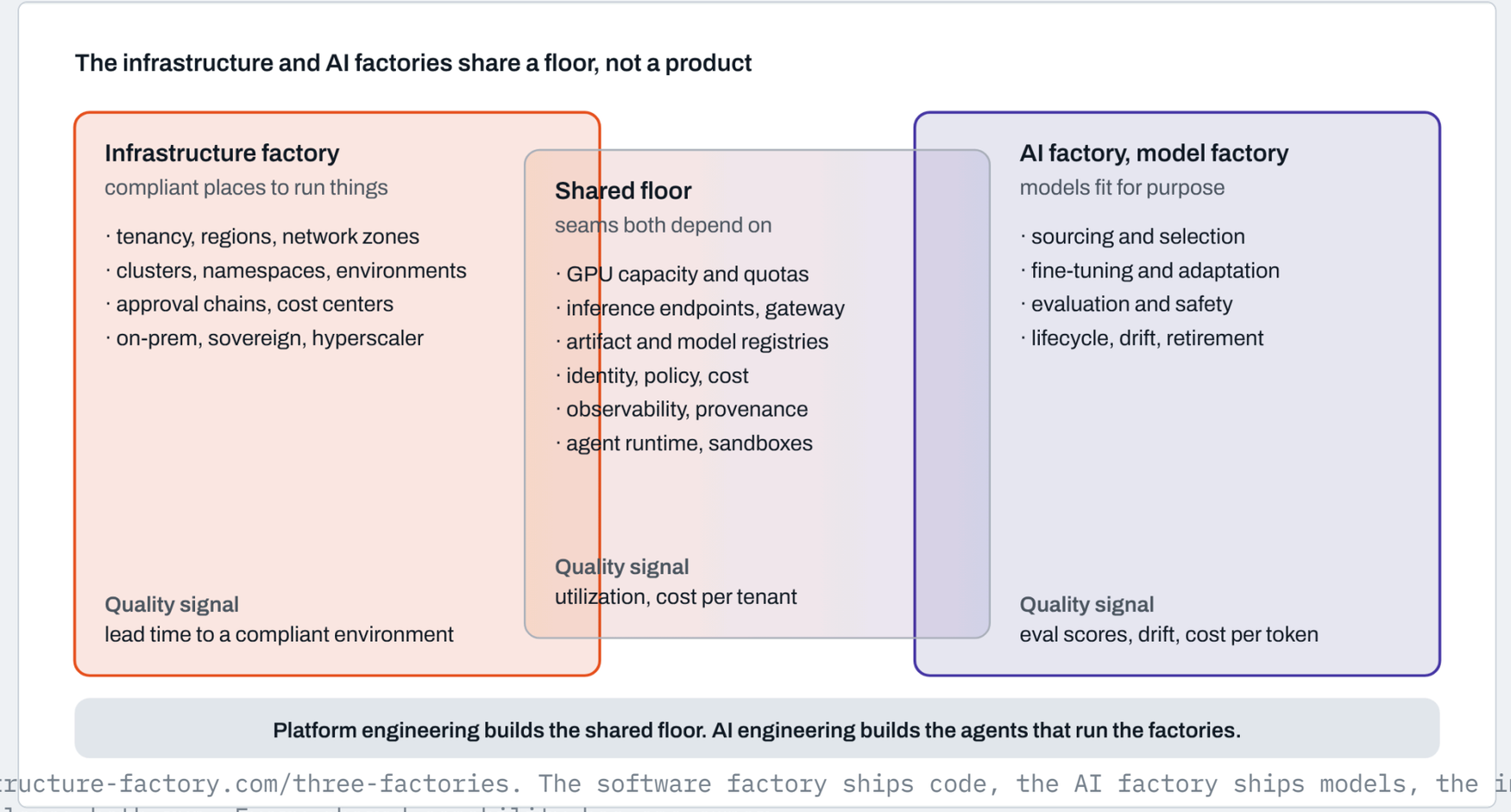
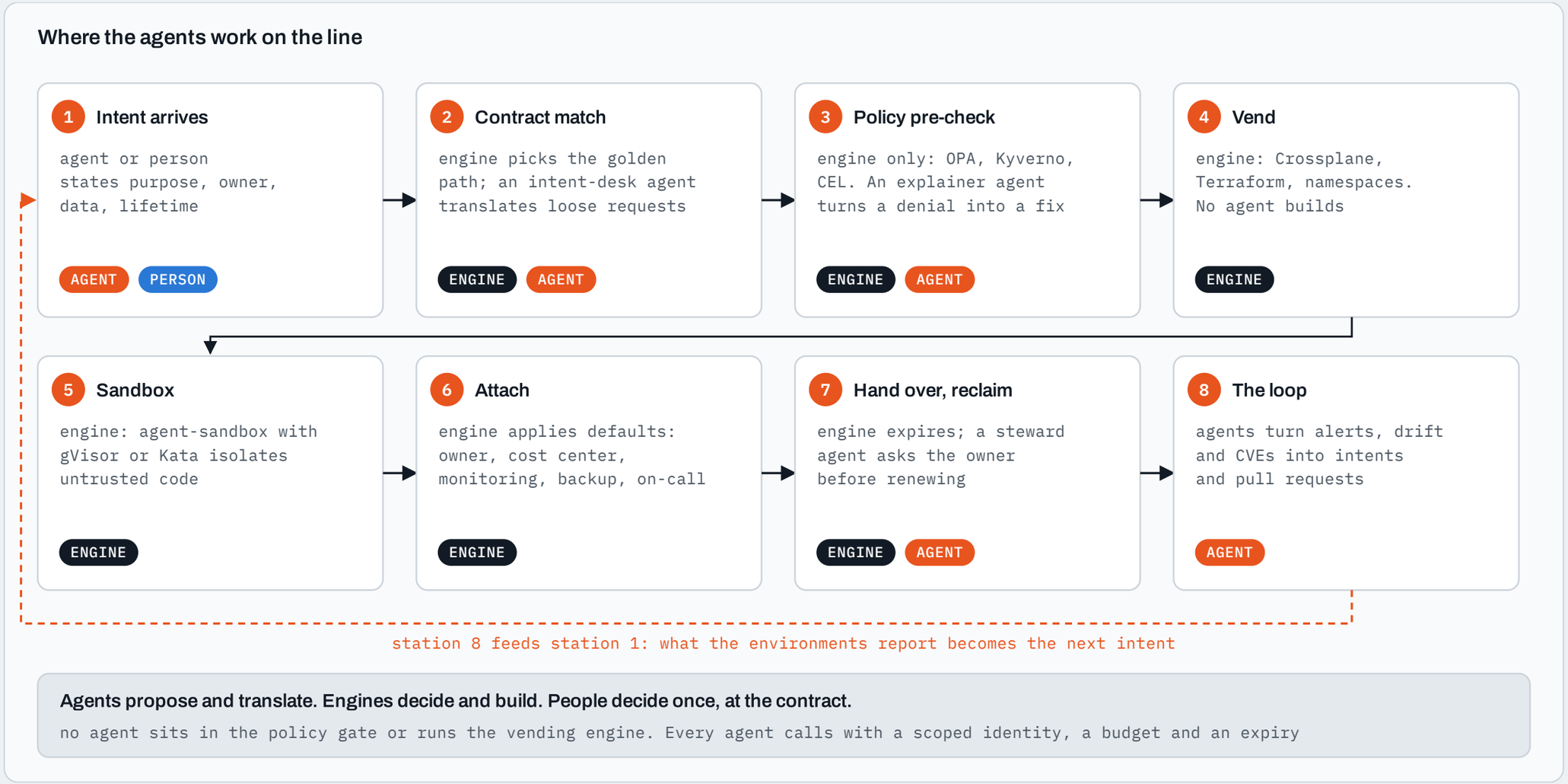


Diagram D3, infrastructure-factory.com/three-factories. The software factory ships code, the AI factory ships models, the infrastructure factory ships the places both run. Every shared capability has one owner.

Part three:

where the agents work, and where they do not.

Where the agents work



Your agents on the payroll, and what they may not do

AGENT	JOB	NEVER
Intent desk	Turns a loose request into a contract-shaped intent and asks the person for what is missing	Approves anything
Explainer	Turns a policy denial into a concrete fix, and proposes contract changes as pull requests	Changes a policy
Steward	Finds orphans and idle environments, asks owners before expiry, negotiates renewals	Deletes without an owner's answer
Loop	Turns alerts, drift and CVEs into intents and pull requests on the infrastructure line	Applies to production on its own

Every one of them is a requester like any other: a scoped identity, a budget, an expiry, and the same gate in front of it.

Agents run most of operations

SIGNAL	AGENT	THE ENGINE DECIDES	A HUMAN, ONLY WHEN
Alert	Triage: environment or workload? Routes with evidence	Owner from the catalog	It is the workload's: its on-call owner
Drift	Remediation: the fix as a pull request	Policy: apply alone or ask	The contract does not describe the fix
New contract version	Rollout: upgrades, certificates, CVEs, canary first	Waves inside the maintenance window	Outside the window, or above a blast radius
Cost anomaly	Cost: right-size, propose a throttle	Budget ceiling	The owner disagrees
Expiry	Steward: asks the owner, renews or reclaims	Lifetime in the contract	Renewal above a threshold

Scoped identity for every requester

- An agent never borrows a person's credentials. It has a workload identity of its own, OIDC or SPIFFE, and carries the person it acts for as a claim.
- The gateway reads both: a rule may allow the tool for the workload and the budget for the person.
- Budgets and lifetimes attach to the identity. A coding agent gets a sandbox quota and 72-hour environments, not a cloud account.
- Model calls go through the same gateway. Token spend lands on the same cost center as the environments. The AI factory is a supplier with an invoice.

FOR DEVELOPER RELATIONS

Your next developers are agents

- **Tool descriptions are documentation.** A model reads them before a person does. Write them the way you write a quickstart.
- **Contracts are the catalog.** The golden paths are the product. Name them, version them, show what each one gives and what it asks for.
- **Denials are onboarding.** The reason behind a "no" is the most-read text in the factory. Make it say what to do next.
- **Publish the metrics.** Zero-touch rate and lead time tell the community whether the factory is real or a portal with a new name.
- **Argue in the open.** Contracts and policies as pull requests, and a place to disagree.

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**