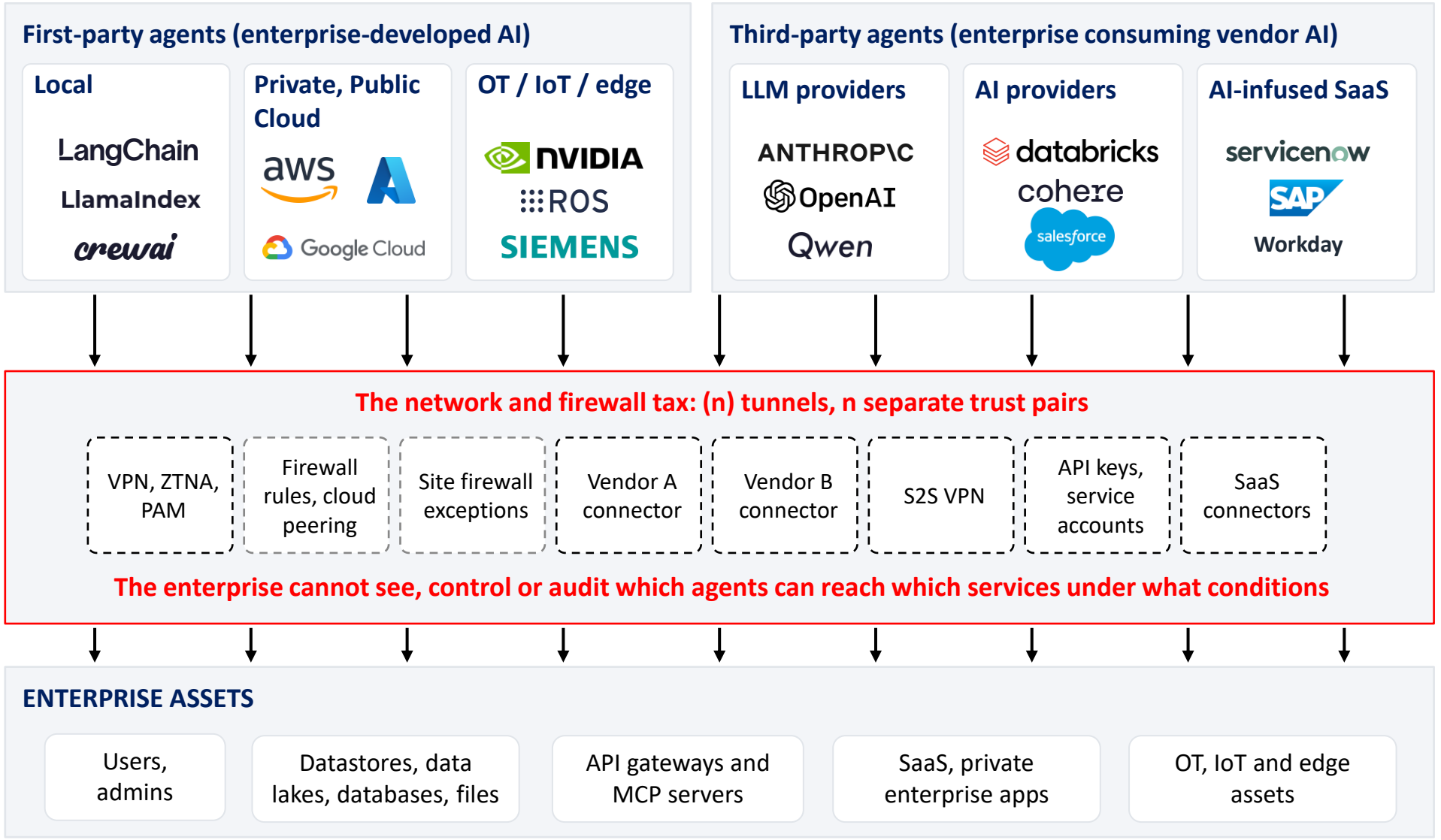
AI can fracture governance

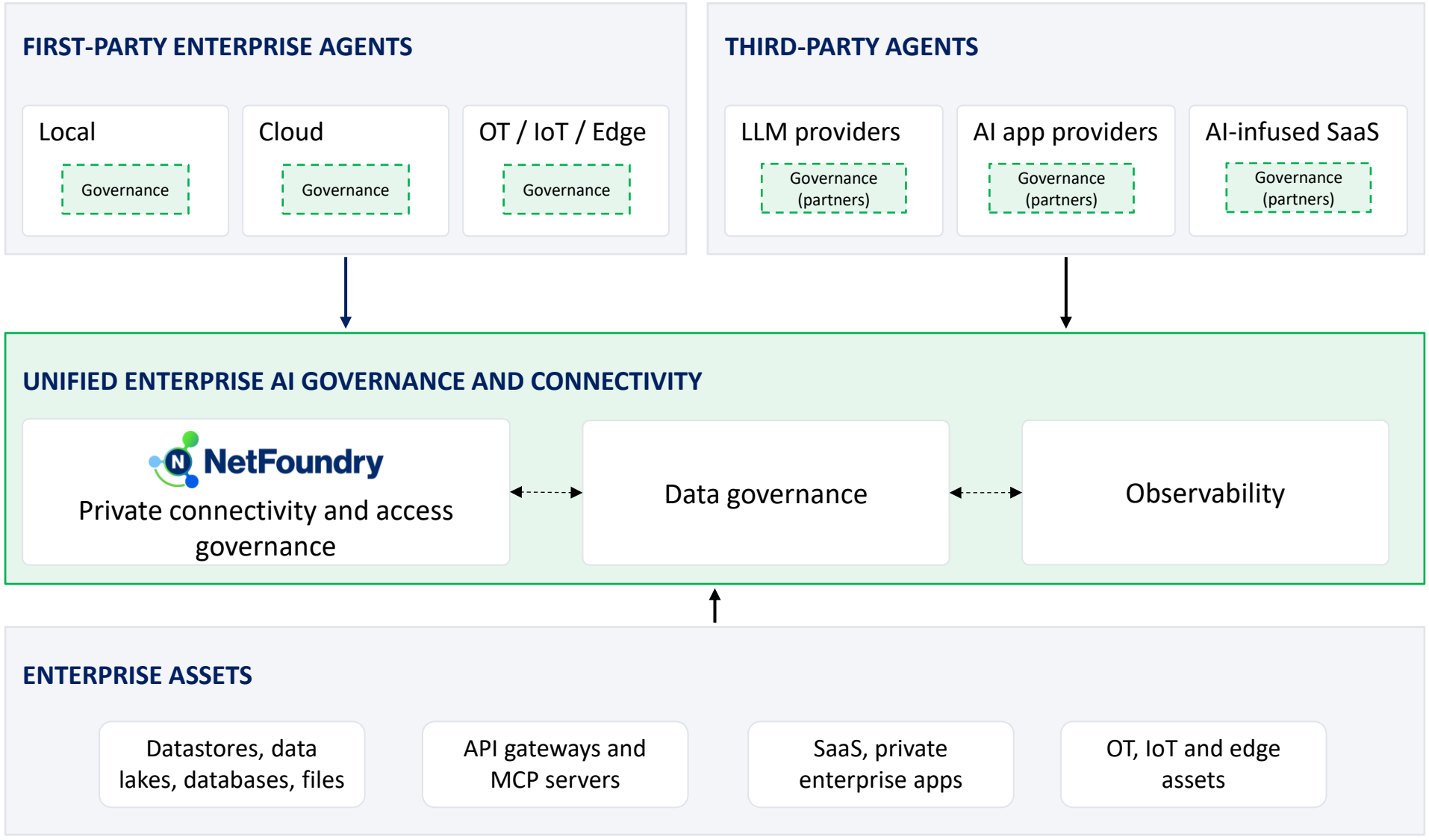


Many enterprises are stuck: slow AI adoption or sacrifice governance?

- Agents need access to enterprise assets.
- Agents break data and access solutions made for pre-AI.
- (n) tunnels for (n) AIs causes fractured governance and controls, with high network and firewall taxes.

NetFoundry

NetFoundry helps provide unified governance, connectivity, control



- NetFoundry helps enterprises get **unified governance and connectivity**.
- NetFoundry puts private connectivity into the governance stack for the first time.
- NetFoundry extends controls as close to the AI as possible, while covering all use cases.

Identity defined AI governance and connectivity – use cases

Enterprise development category: enterprises building AI solutions

1 A2A, server-to-server

First-party agents which need to reach your enterprise assets.

2 Secure remote access

Dev and ops reaching AI cloud-hosted agents and self-hosted environments.

3 OT, IoT and edge AI

Agents reaching IEC 62443 zones, device fleets and edge compute.

Enterprises consumption category: using third-party AI

4 Vendor connectors

Vendors who provide enterprise-hosted connectors to facilitate outbound only connection.

5 Vendor private paths

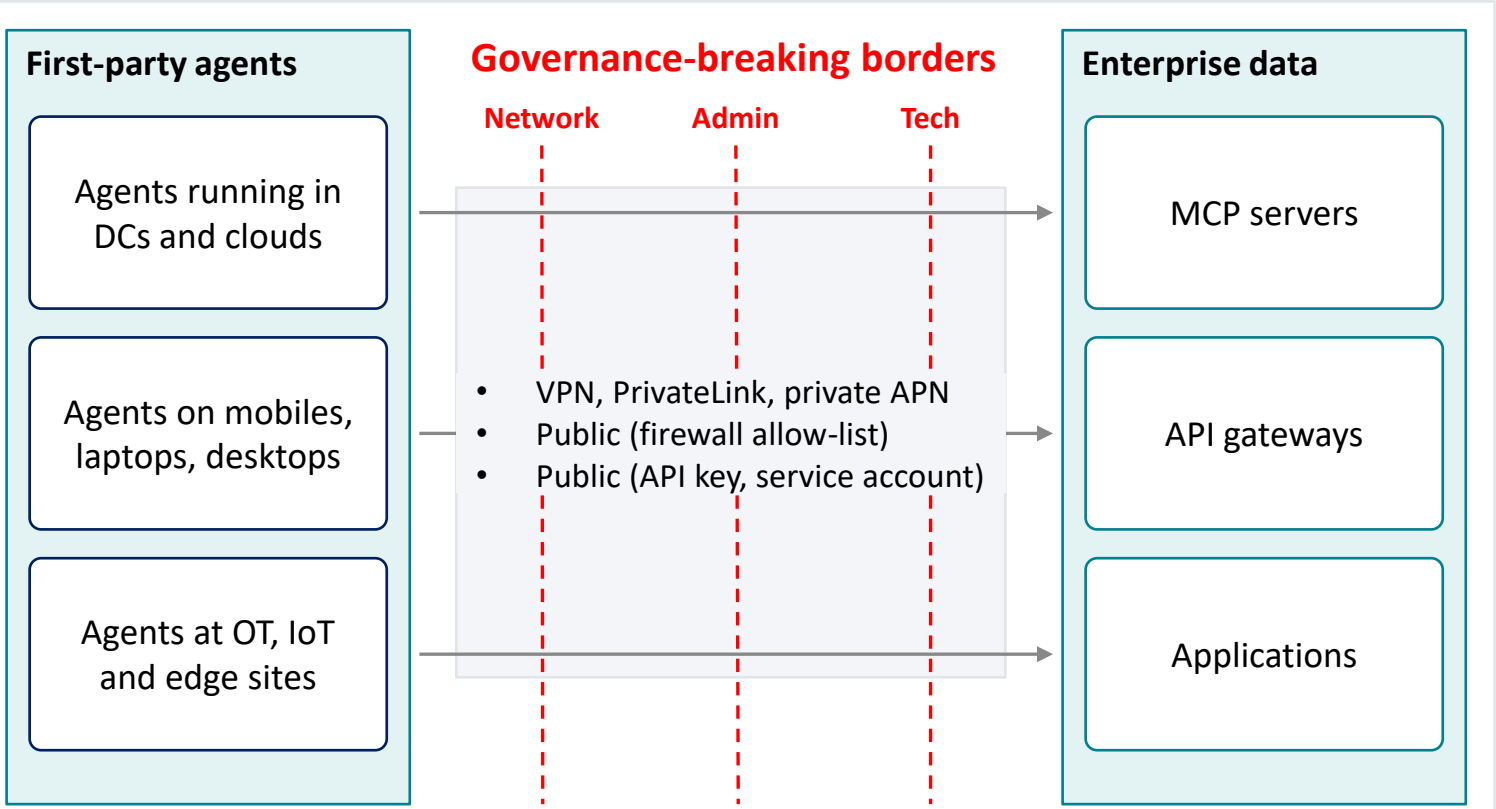
Vendors who provide private link solutions but require enterprise to get the traffic to the site.

6 Vendor MCP server access

Vendors and partners who need access to enterprise MCP servers.

NetFoundry

Use case #1: network defined governance for enterprise developed AI



Challenges

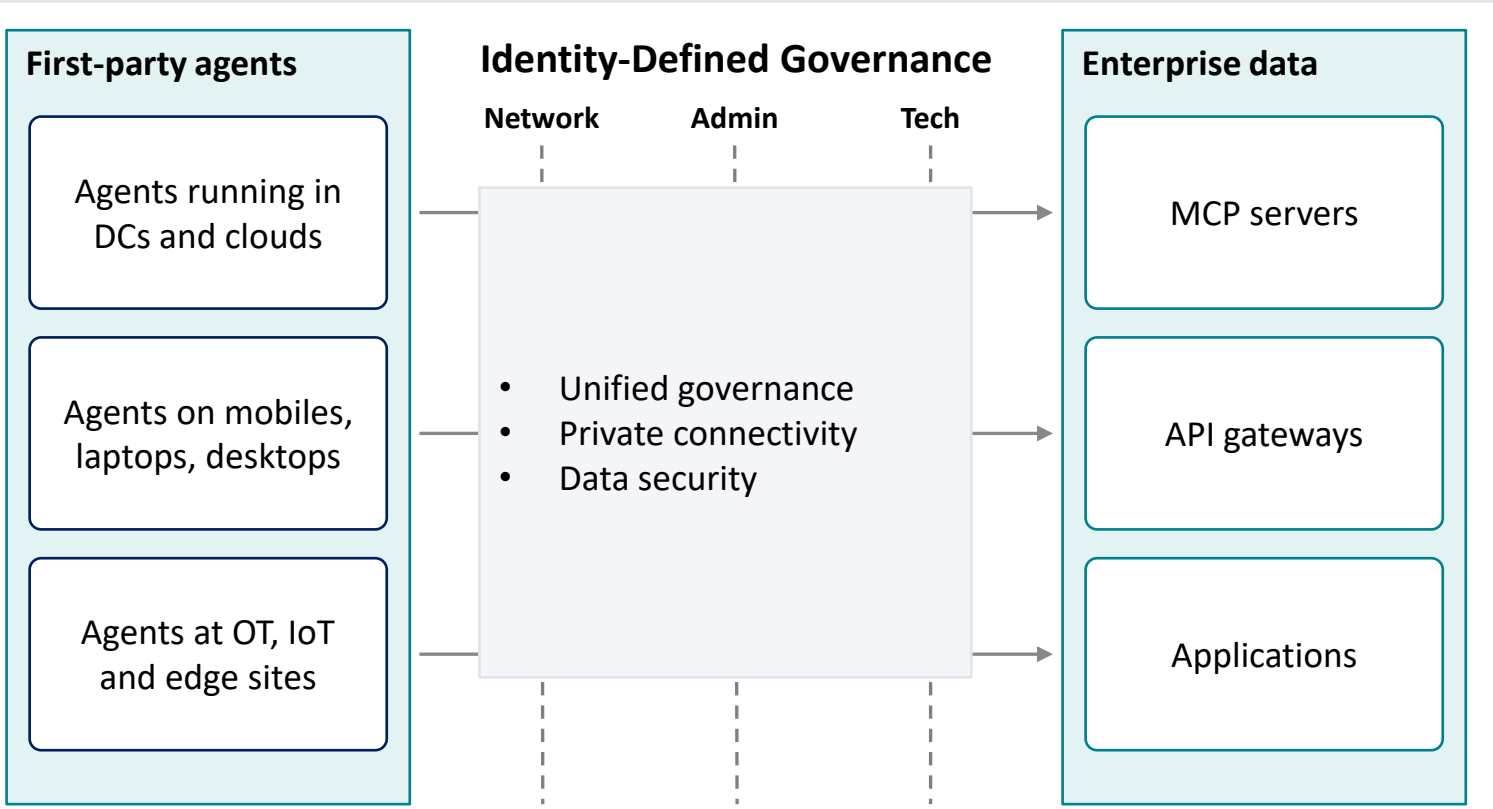
- Giving agents access to enterprise systems is slow and risky. Every new agent, app or site repeats the work; AI slows down, again.
- Governance and visibility break at every border.
- The result is project delays; exposure the business cannot easily see (which agents can reach which systems, right now) and fragmented governance.

Network Defined Connectivity and Governance

Every border adds a firewall rule, VPC peer, transit gateway, PrivateLink, VPN, or an exposed endpoint with an API key. **Each border breaks governance and makes the target reachable to anyone who can reach the border.**

NetFoundry

Use case #1: identity defined governance for enterprise developed AI



Identity defined connectivity and governance

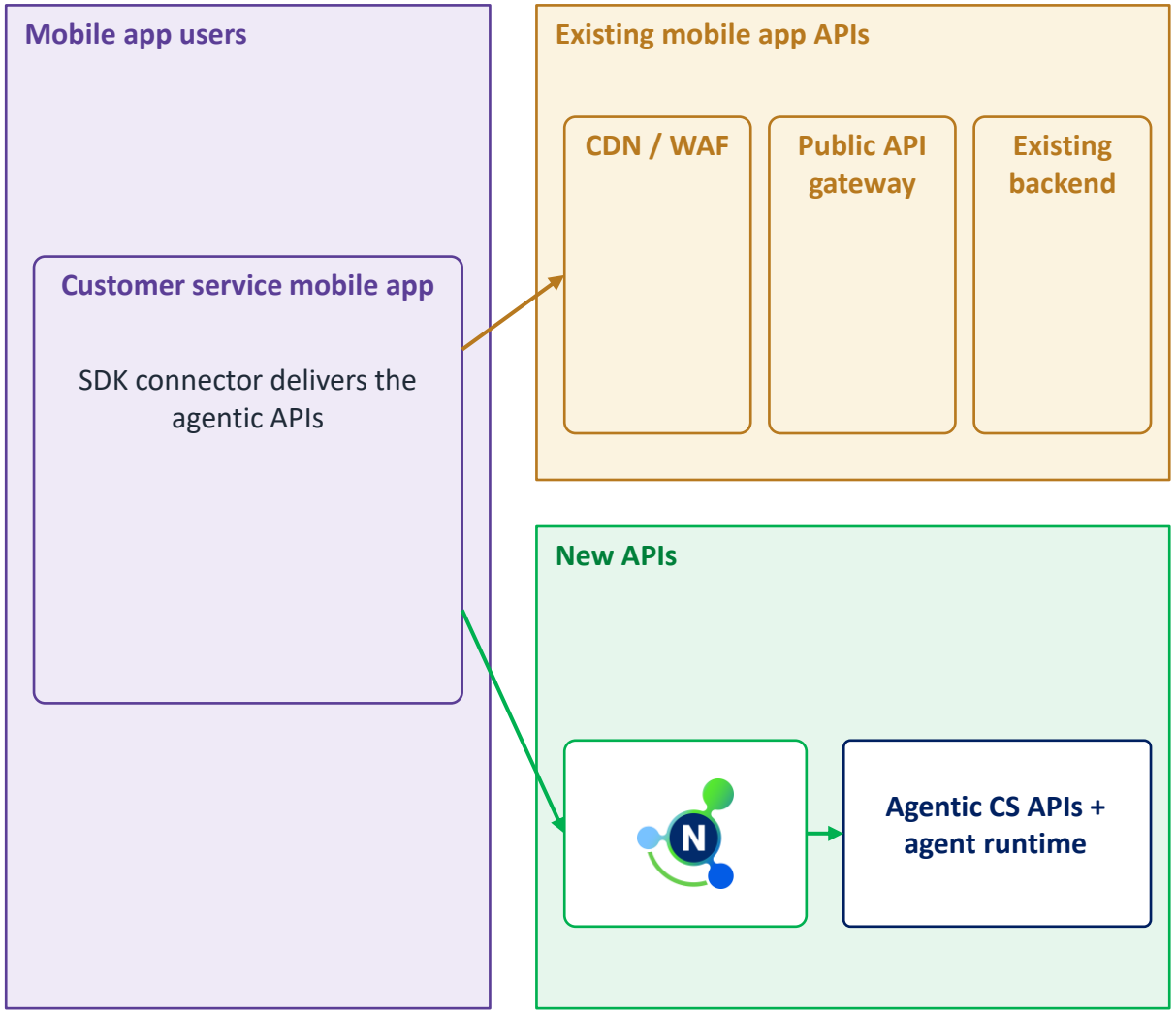
Control and visibility based on identities and policies – no longer fragmented by network borders. Workloads are only visible and reachable to authorized identities.

With NetFoundry

- Connectivity is a policy the team declares, not a queue of trouble tickets and slew of border crossings.
- Agents can only access the services they are authorized for. Nothing else is visible or reachable.
- Minimize the network tax - firewall and network changes which risk uptime and delay projects.

NetFoundry

Example of use case #1: add agentic to an existing customer service app

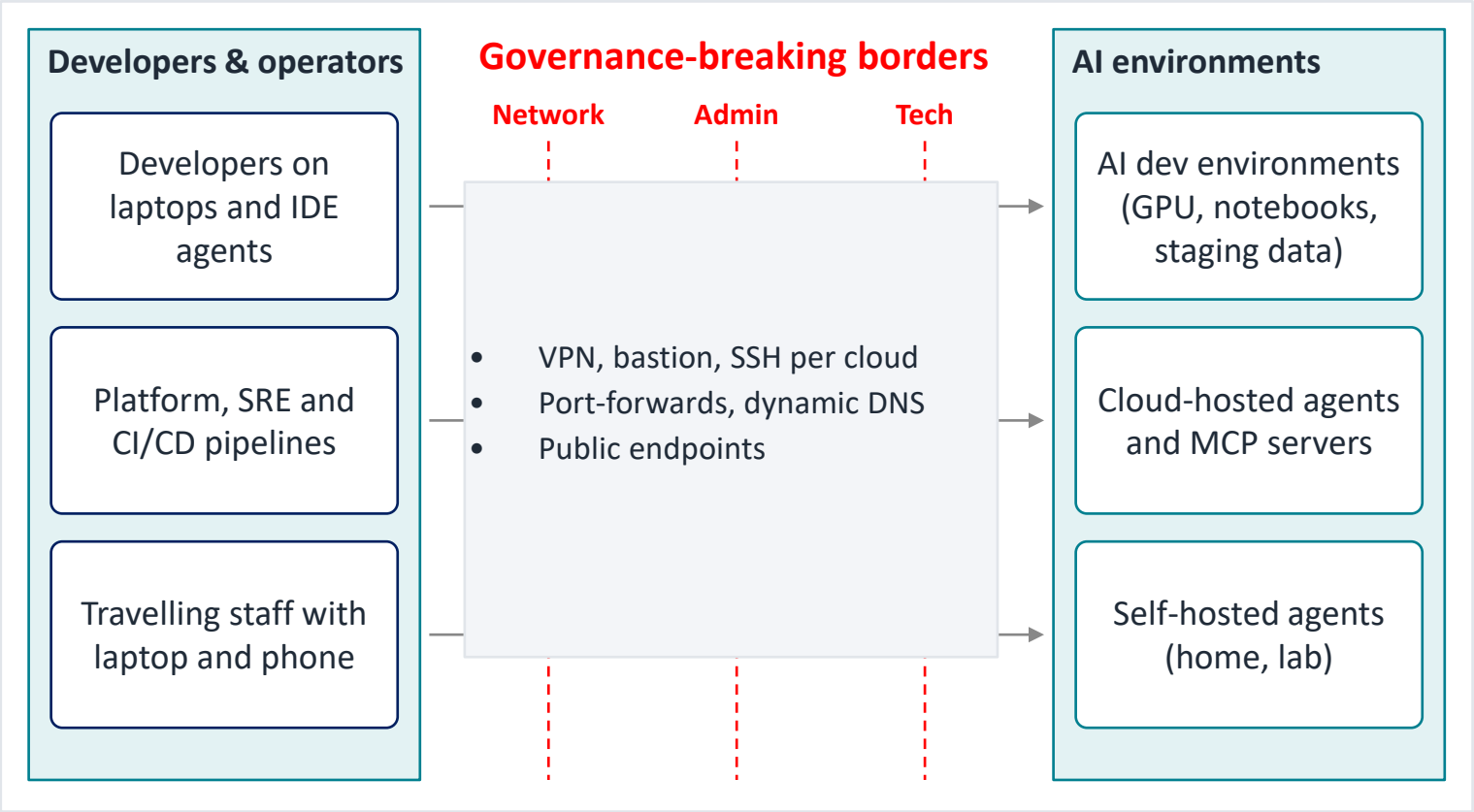


Results

- The agentic APIs are unreachable from the Internet: no public API, no WAF/DDoS surface.
- Governance, visibility and control based on identities and policies – not IP addresses. Instantly revoke a device and the app disappears for them but remains available to authorized users.
- Users just use the app – no VPN, no separate phone. Existing mobile traffic keeps using the old architecture. The new APIs are reachable solely by an authorized user on an enrolled device (user identity, device identity, posture).

NetFoundry

Use case #2: network defined governance for AI developers and operators



Challenges

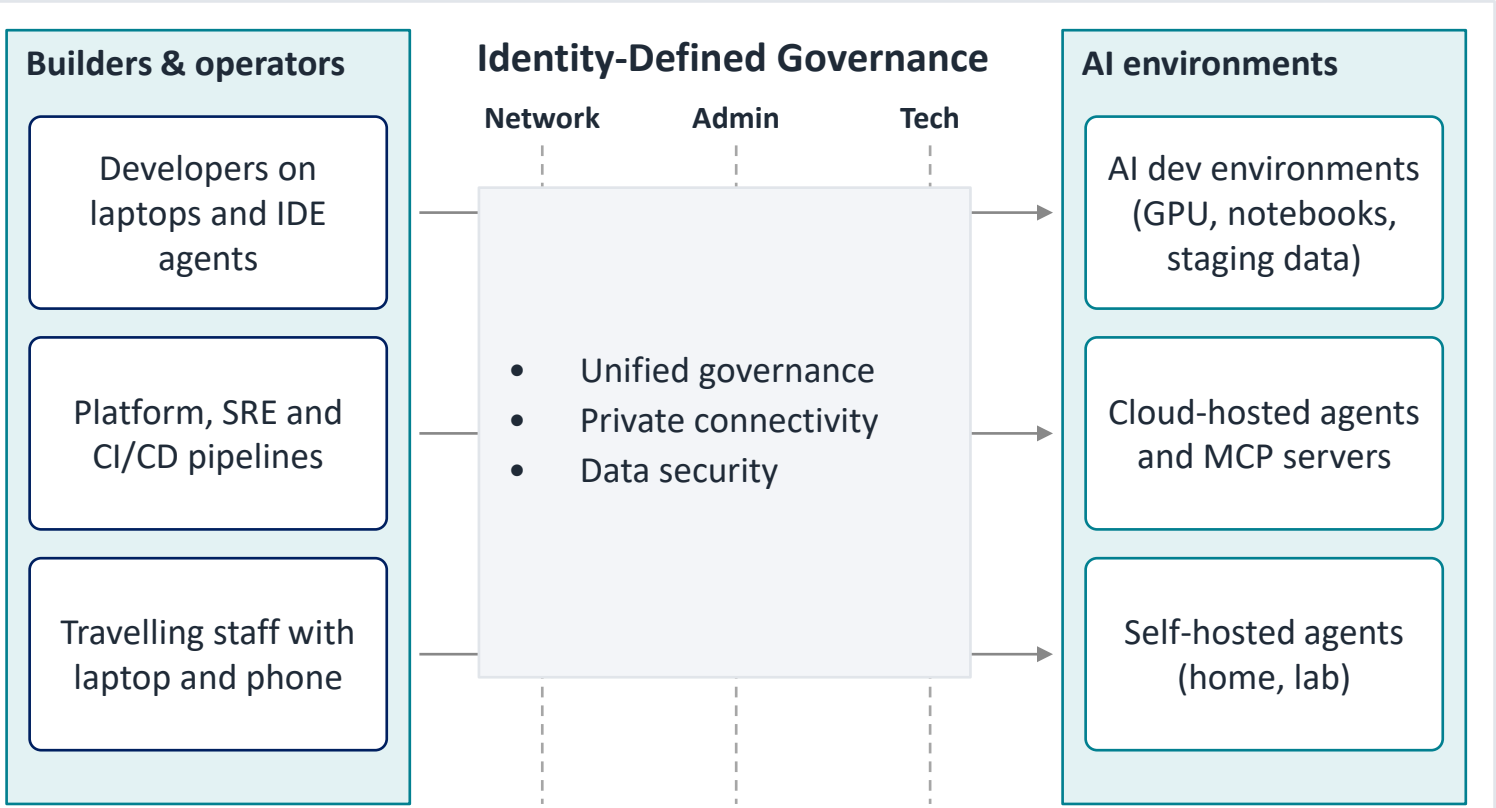
- The people building AI lose hours a week to access plumbing: VPN profiles per cloud, bastion hops, tunnels that break, a ticket for every new environment.
- Shortcuts create exposure: MCP servers and agents published to the Internet, home-hosted agents behind port-forwards.
- No consistent view of who reached which environment or agent, from where, on what device.

Network Defined Connectivity and Governance

Every builder and environment adds a VPN, bastion, tunnel or open port. **Each is a door which is built outside of governance and consumes builder time.**

NetFoundry

Use case #2: identity defined governance for AI developers and operators



Identity defined connectivity and governance

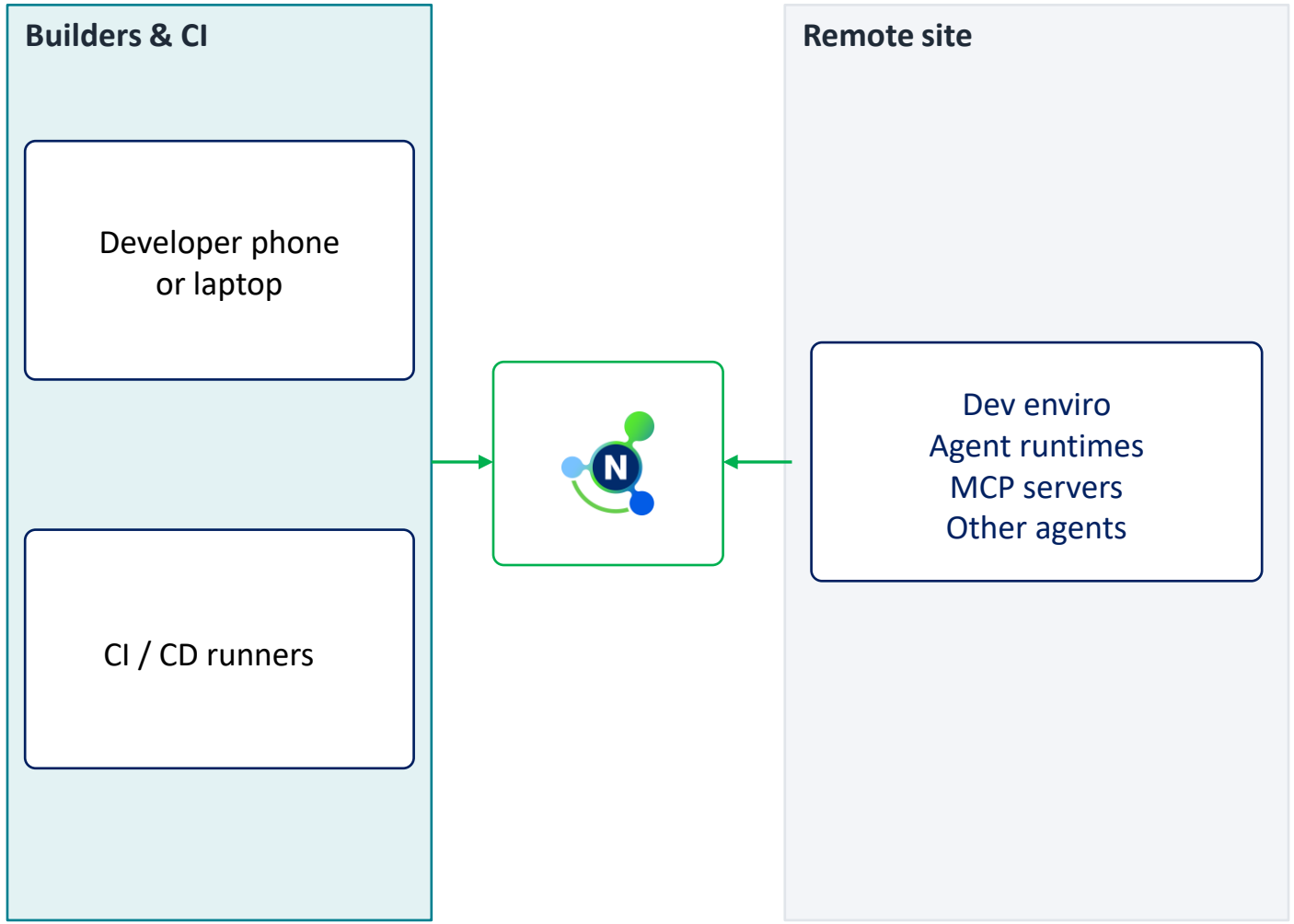
Builders reach exactly what they are authorized for, from anywhere, controlled by enterprise identities and policies. No open doors. Governance meets speed.

With NetFoundry

- Dev environments, cloud agents and home servers no longer need to be made visible and reachable in order for builders to access them.
- Eliminate the network tax – VPNs, firewall changes, port forwarding.
- Access is per service and per session with posture, with full visibility, controls and governance.

NetFoundry

Example of use case #2: developers and CI reaching cloud AI environments

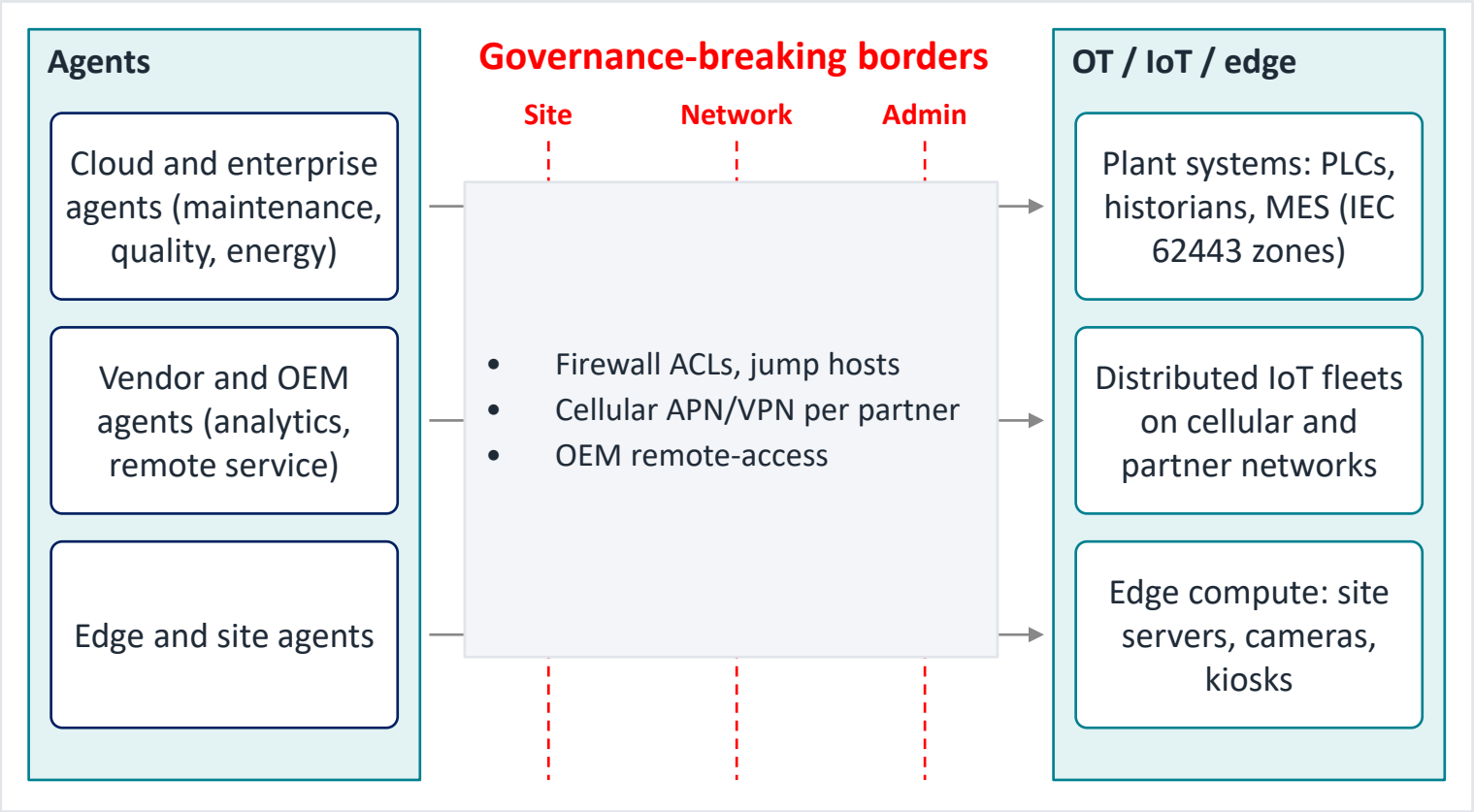


Results

- Developers, their tools and their agents just connect; no VPN profile per cloud, no tunnel to set up, no networking tax.
- The remote site has no inbound access, no bastion and no public IPs; unreachable and not visible to unauthorized identities.
- Governance, visibility and control based on identities and policies: developer, device and CI runner each get only the services their role allows.

NetFoundry

Use case #3: network defined governance for OT, IoT and edge AI



Challenges

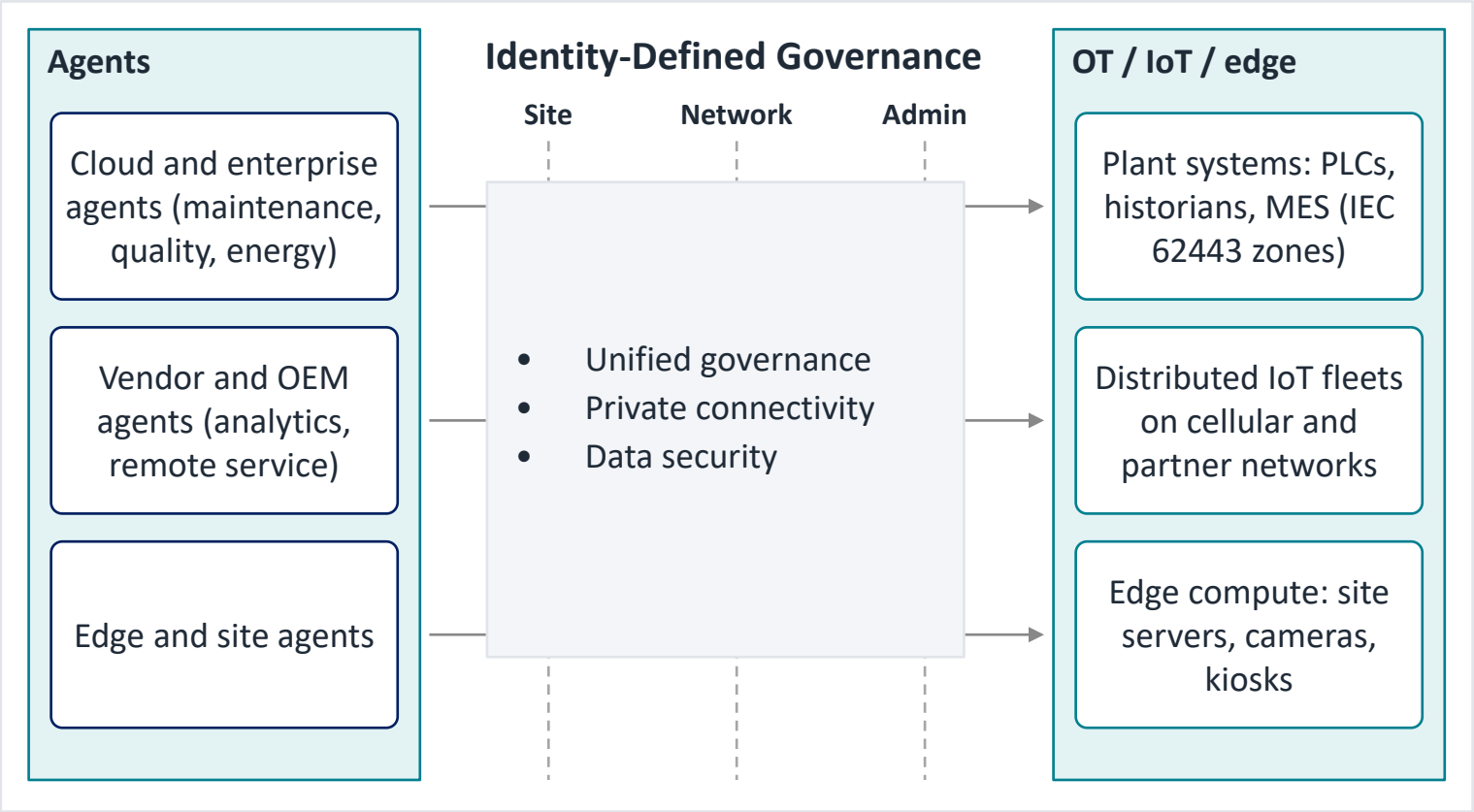
- OT is where agentic ROI is largest and adoption slowest: every connection is an exception to a safety-driven and uptime-driven architecture.
- Zones and conduits were designed for people and fixed systems – agents can cause chaos.
- Each plant, vendor and fleet brings its own remote-access path; nobody can answer which may be adding AI agents and what they can access.

Network Defined Connectivity and Governance

Every agent request becomes an OT change request: a firewall exception, a jump host, an APN, a vendor box. Each opens a path into a flat Purdue level. Per vendor VPNs do not differentiate between humans and AI agents.

NetFoundry

Use case #3: identity defined governance for OT, IoT and edge AI



Identity defined connectivity and governance

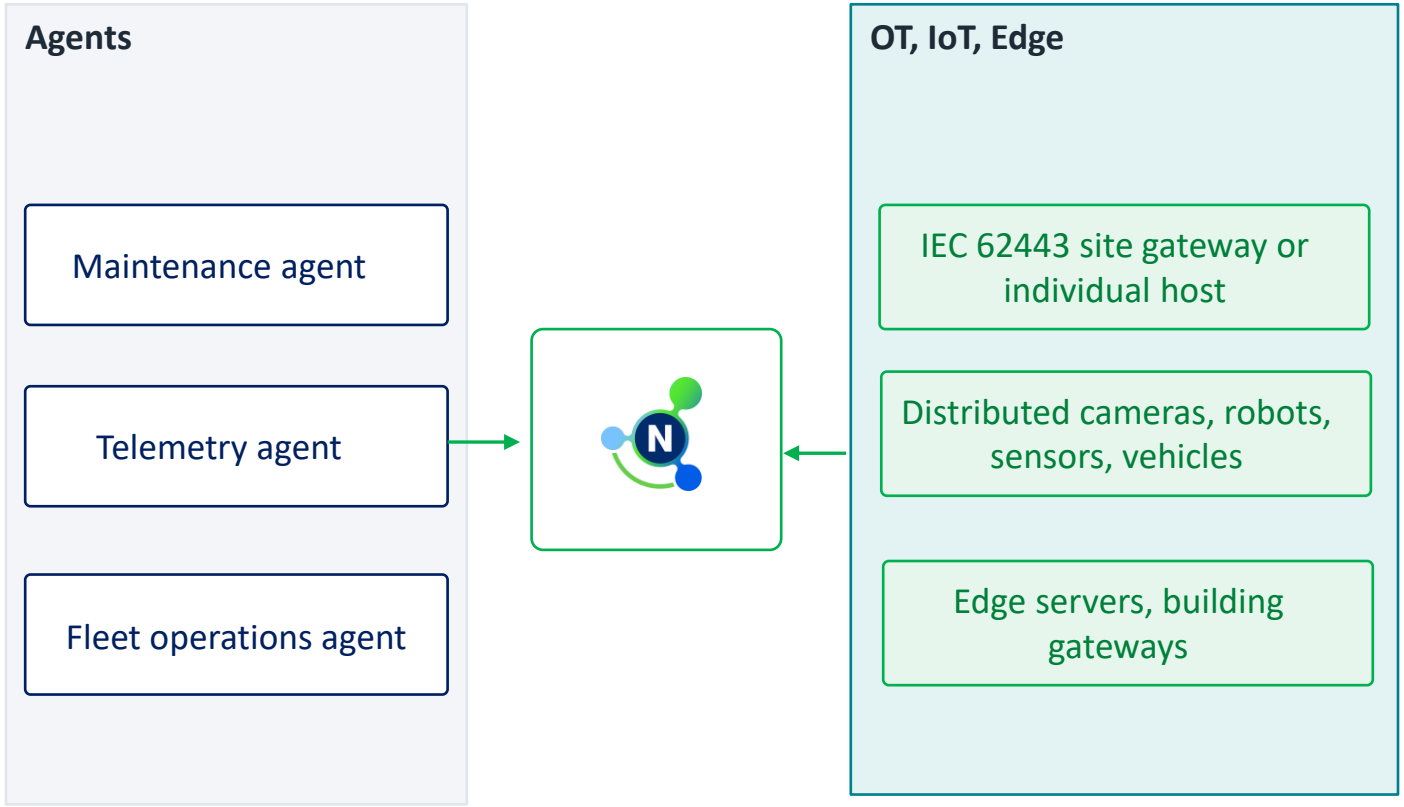
Replace open conduits with identities and policies. Agents reach exactly the plant systems and device groups they are authorized for; the plant and the devices remain invisible. Every user or agent has an identity for proper controls, governance and attribution.

With NetFoundry

- Unified governance via identity-based conduits: e.g. an agent is authorized for specific historian tags, not for a site.
- Sites and devices dial out only. Nothing inside the plant or on the device listens; zones stay intact, exceptions disappear.
- New ransomware can’t enter. Existing malware can’t exfiltrate data to unauthorized destinations.

NetFoundry

Example of use case #3: zero trust into OT, IoT, edge

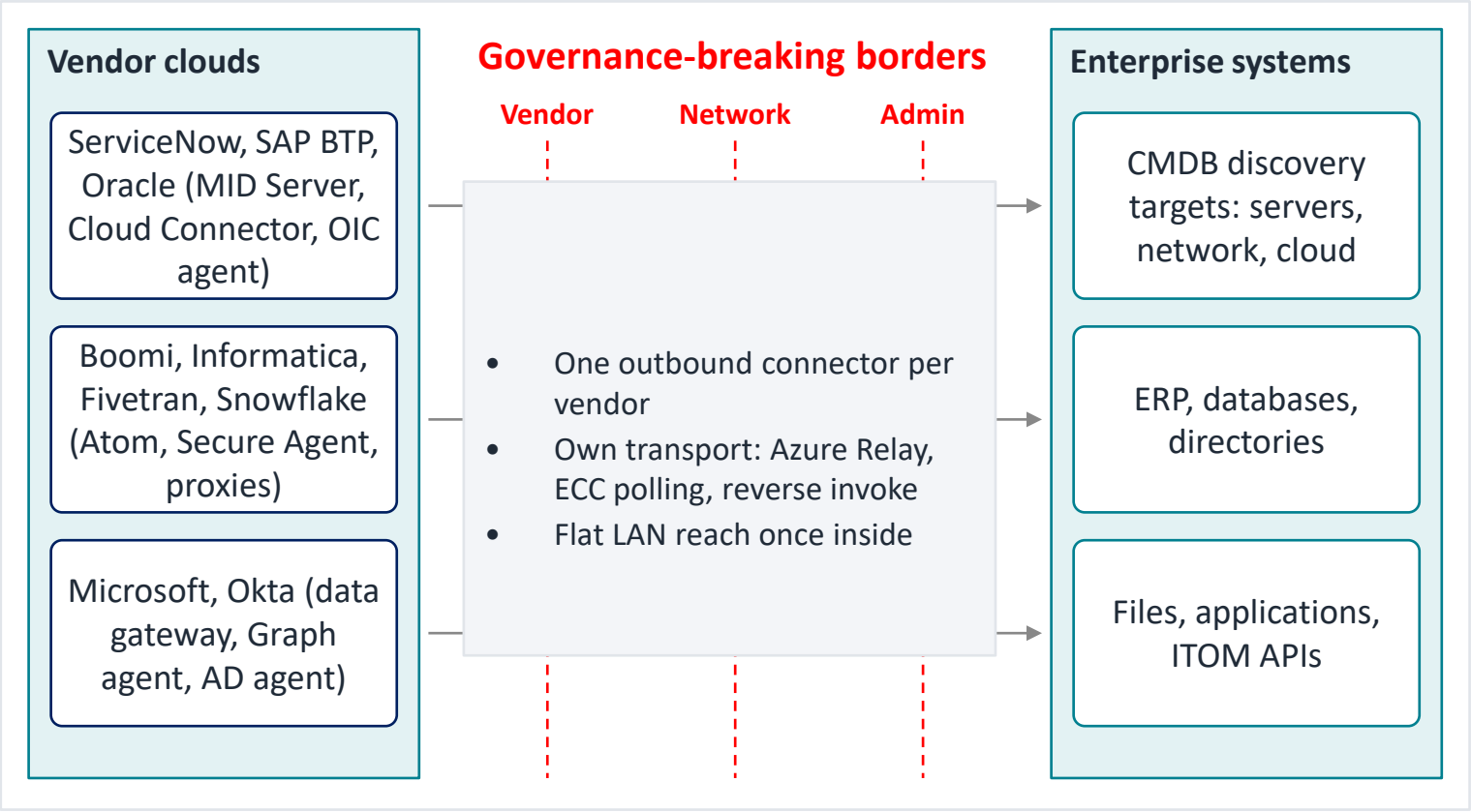


Results

- A plant or other IEC 62443 site has no inbound path: the site gateway is the only thing that dials out.
- Each conduit is an identity-to-service authorization: the maintenance agent may read historian tags and call the MES work-order API, nothing else. Vendor agents get an identity and a policy, not a remote-access box or open conduit.
- Zones stay intact; no firewall exceptions, no jump hosts, one audit trail.

NetFoundry

Use case #4: network defined governance for customer-hosted vendor connectors



Challenges

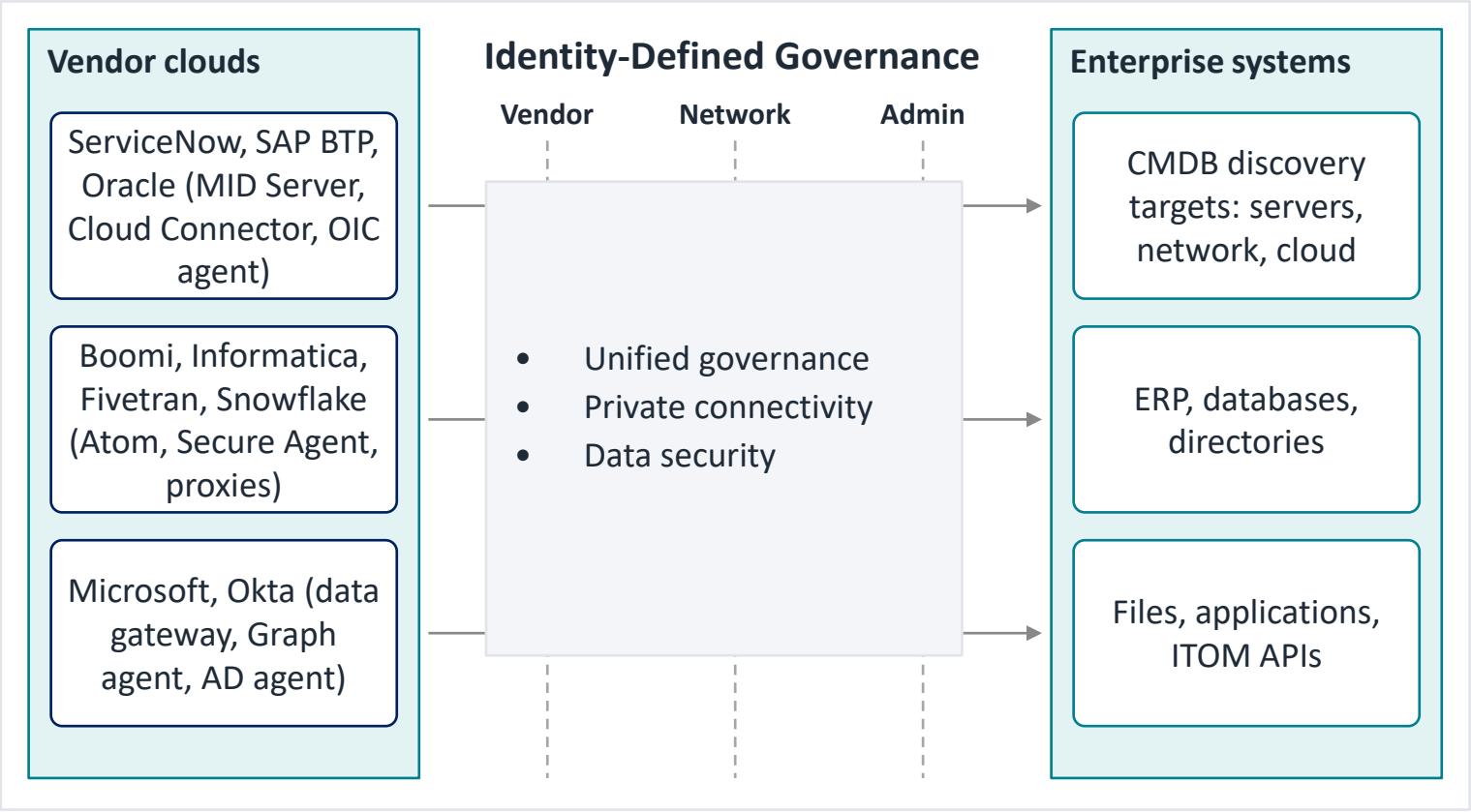
- Each vendor connector was the "zero trust" of its era: outbound only. But once inside it can reach far more than it needs, with its own credentials, policy model and logs.
- Agentic features on these platforms now drive the connectors with non-deterministic requests.
- (n) vendors, (n) trust relationships; a compromised connector has network-wide reach.

Network Defined Connectivity and Governance

Each connector is a separate trust relationship the enterprise cannot see or govern consistently. With AI causing more Saas vendors to need inbound access, the problem is about to multiply.

NetFoundry

Use case #4: identity defined governance for customer-hosted vendor connectors



Identity defined connectivity and governance

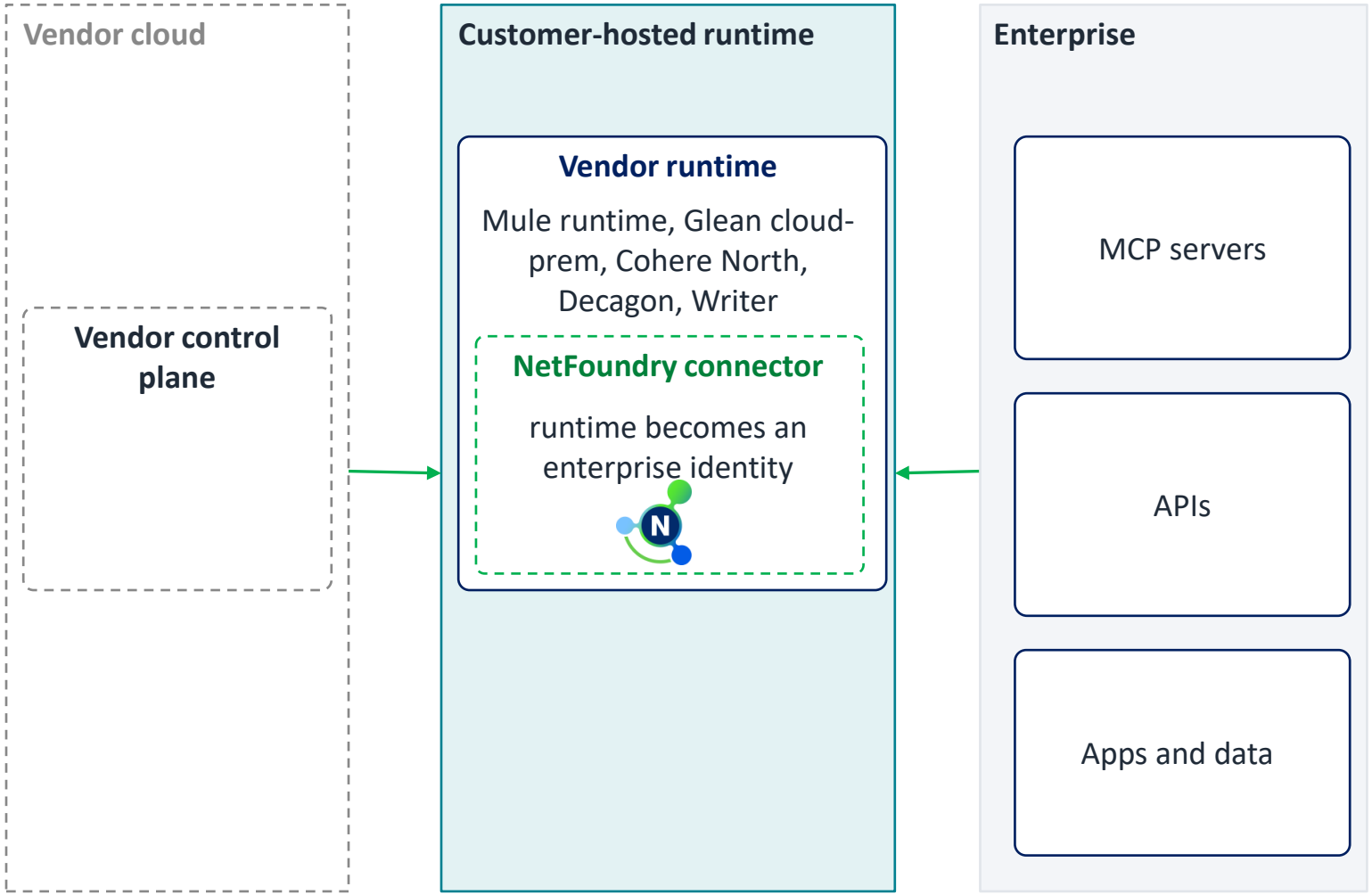
The connector keeps talking to its vendor; what it can reach inside the enterprise becomes an identity and a policy. Least privilege per connector, unified governance across vendors.

With NetFoundry

- Keep the vendor's northbound leg exactly as designed; govern the southbound leg: the connector reaches enterprise systems only through NetFoundry.
- Never need to grant inbound access to another vendor – even if they don’t have their own connector.
- One policy model and one audit trail across all vendor connectors, without asking any vendor to change.

NetFoundry

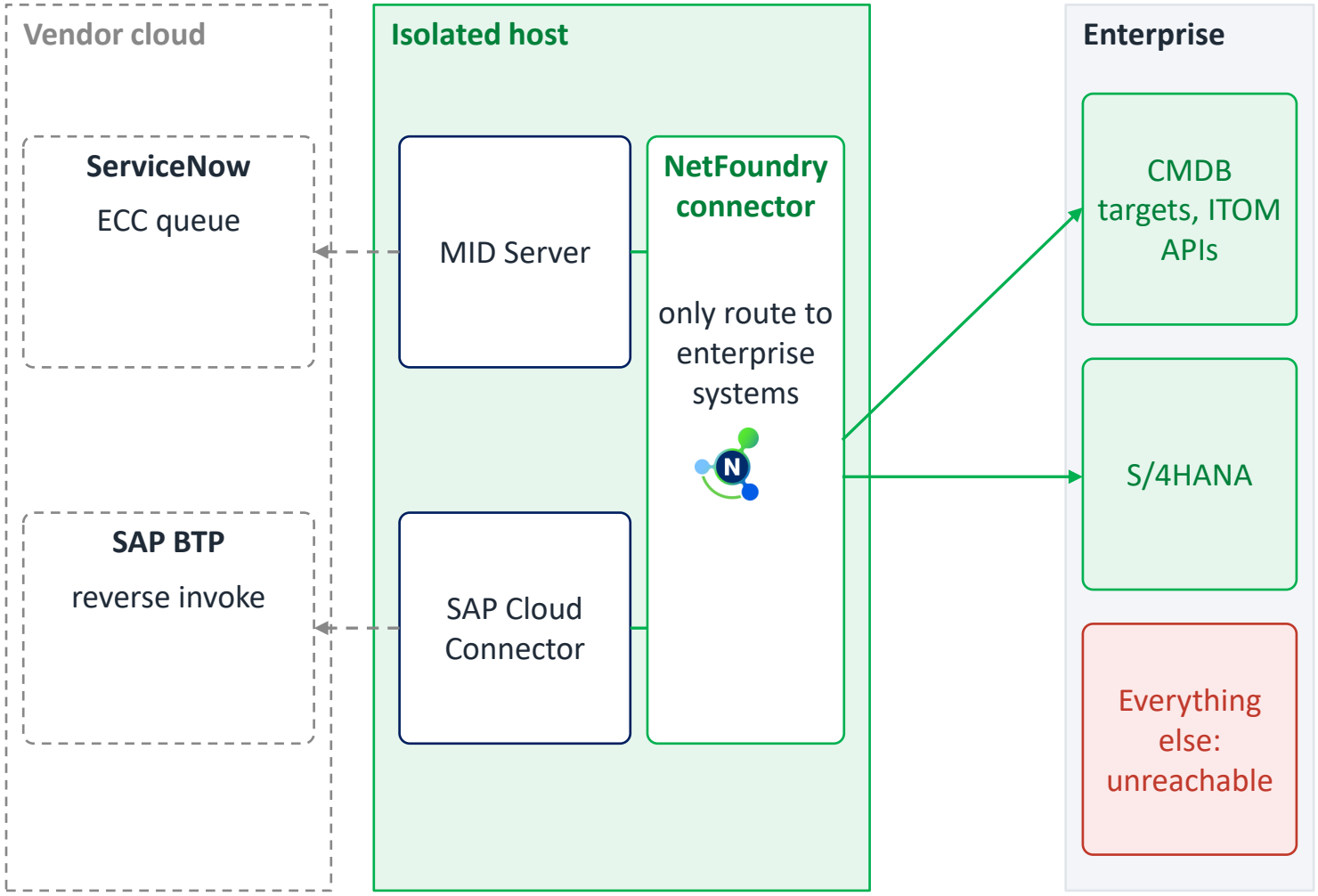
Example of use case #4: NetFoundry connector inside the vendor runtime



Results

- The vendor control plane still connects as designed.
- Access to enterprise systems flows only through NetFoundry, so the runtime becomes a first-party enterprise identity inside of the enterprise governance model.
- Enterprise identity and policy determines which services may reach which resources, and maintains visibility, audit and governance.

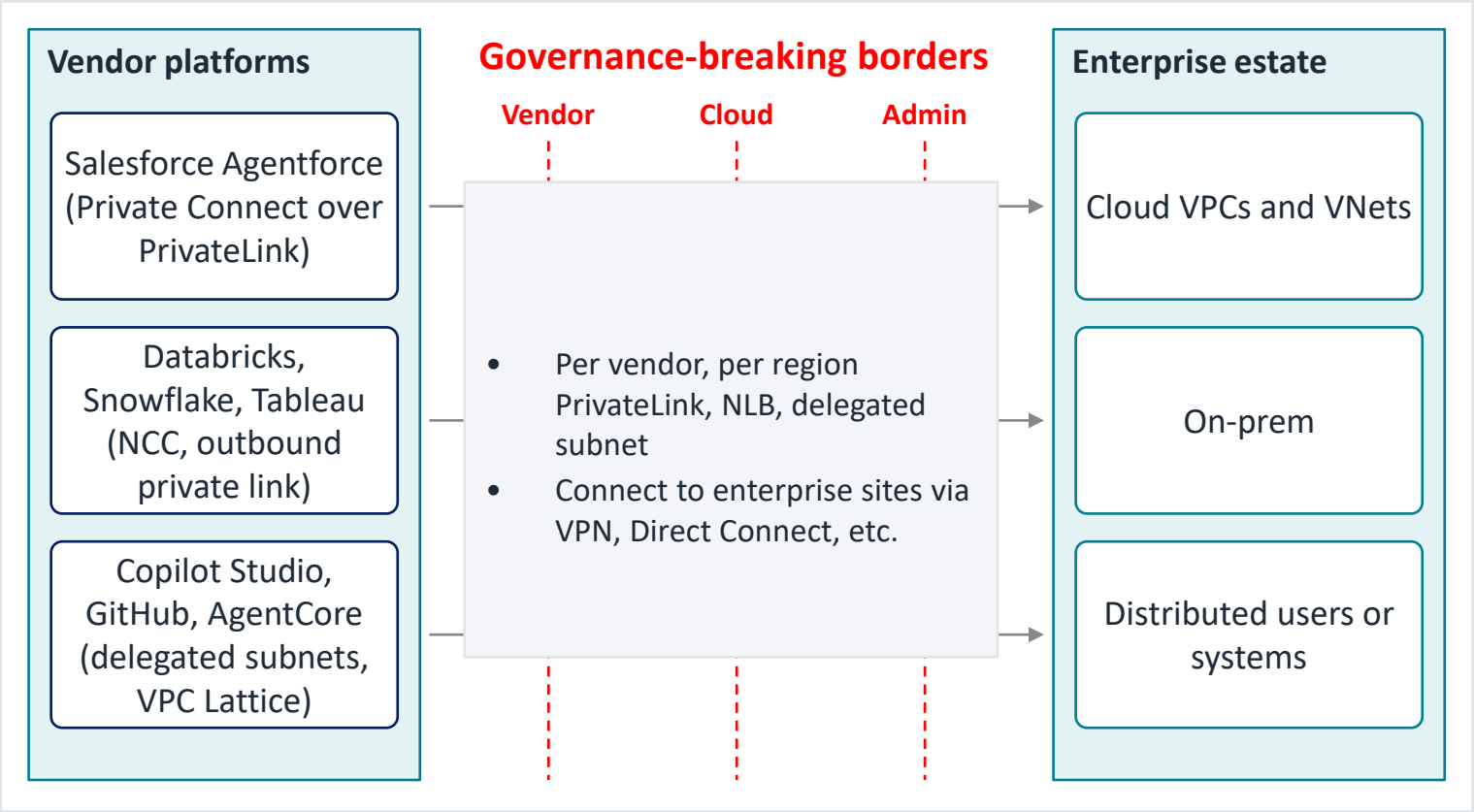
Example of use case #4: vendor connector in a NetFoundry isolated host



Results

- The vendor connector keeps dialling its vendor cloud exactly as designed; egress is pinned to the vendor's ranges.
- Southbound, the host's only path to enterprise systems is its NetFoundry connector: e.g. the MID Server can reach CMDB discovery targets and specific ITOM APIs and nothing else, unresolvable and unreachable.
- (n) vendor connectors, one policy model, one log, without any vendor cooperation.

Use case #5: network defined governance for vendor private paths



Challenges

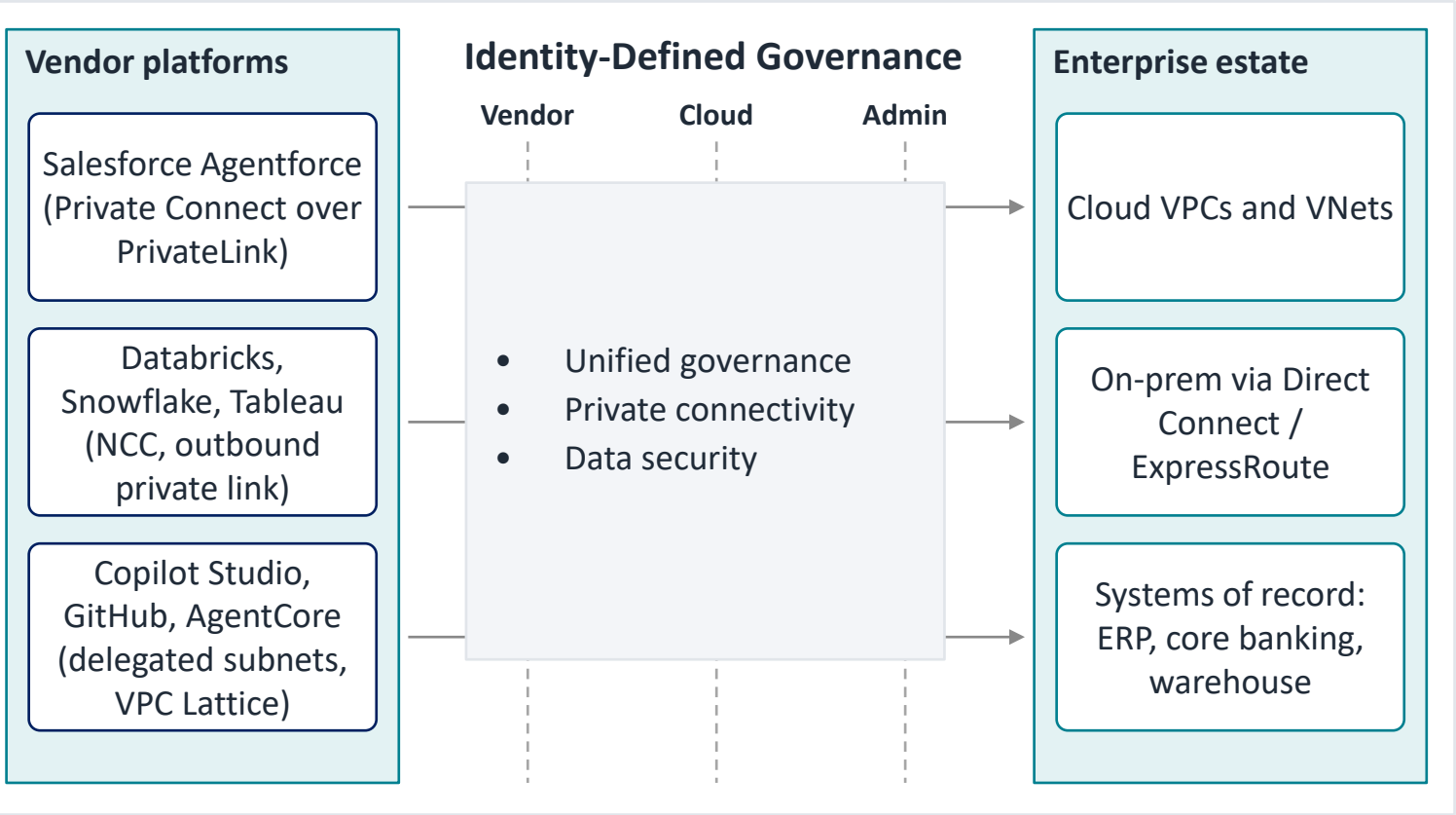
- PrivateLink keeps traffic off the Internet but gives (n) vendors (n) paths into the estate without centralized governance or clear visibility.
- Crossing the borders into other sites, tech stacks or admin zones means paying a high network tax – bridging with VPNs and firewalls.
- Tough decision for enterprises – as the vendor platforms become agentic, they want more access, but that adds to the governance problem and connectivity tax.

Network Defined Connectivity and Governance

Each vendor's path terminates in an enterprise VPC and routes onward through transit gateways, VPN, firewalls. Each path is a high network tax snowflake and no governance layer can provide controls and visibility across all the snowflakes.

NetFoundry

Use case #5: identity defined governance for vendor private paths



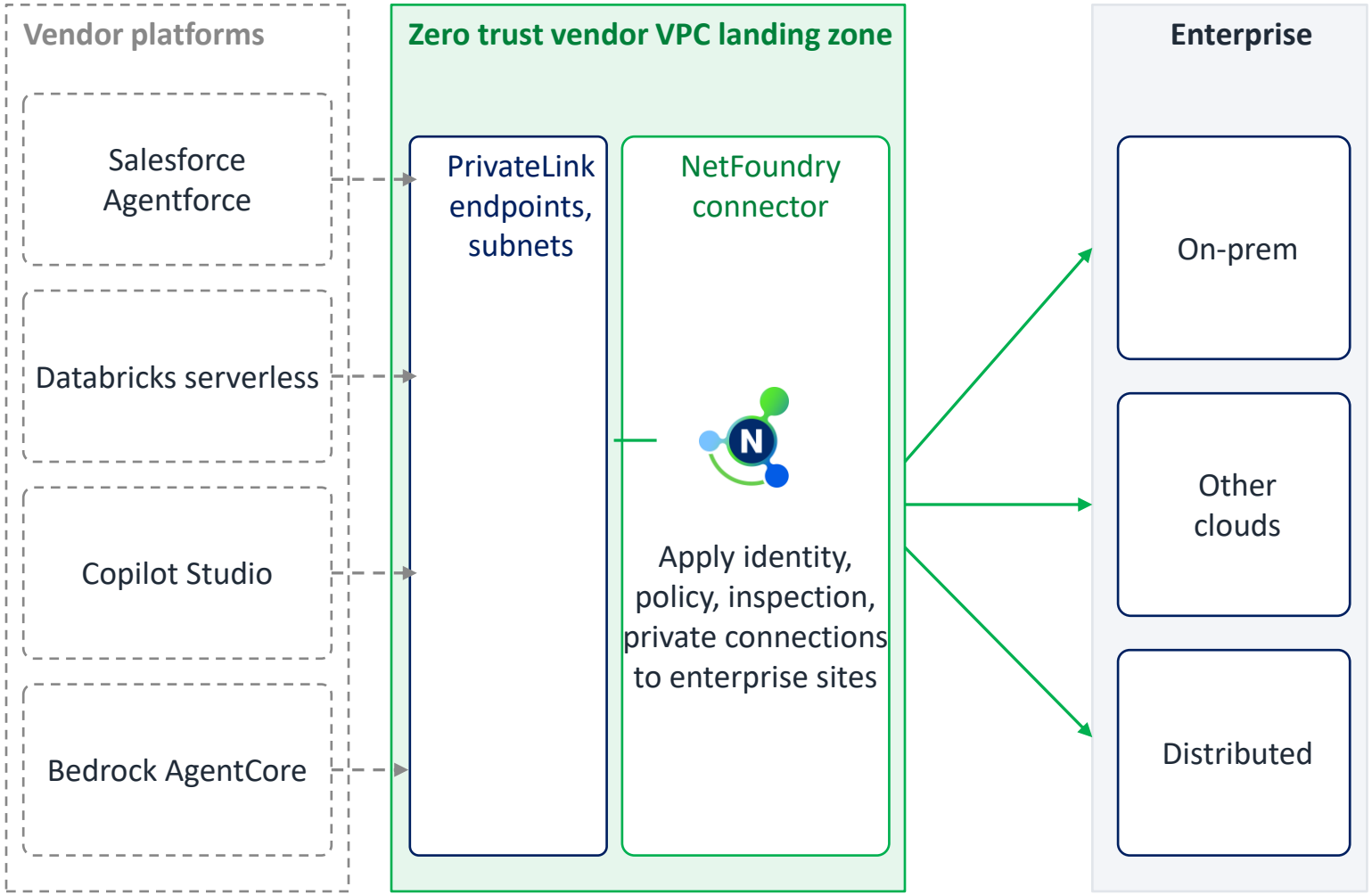
Identity defined connectivity and governance

Vendor private paths land in a governed zone, not in the estate. Vendors become identities; reach becomes policy; one place to inspect, prove and revoke.

With NetFoundry

- One landing zone per cloud: every vendor private path terminates in a small zero trust vendor VPC holding only a NetFoundry connector, inspection and evidence (NetFoundry partners).
- Vendor traffic is mapped to a vendor identity and authorized per service over the fabric; internal addresses are never exposed.
- Unified governance and architecture across all vendors and sites.

Example of use case #5: one landing zone per cloud

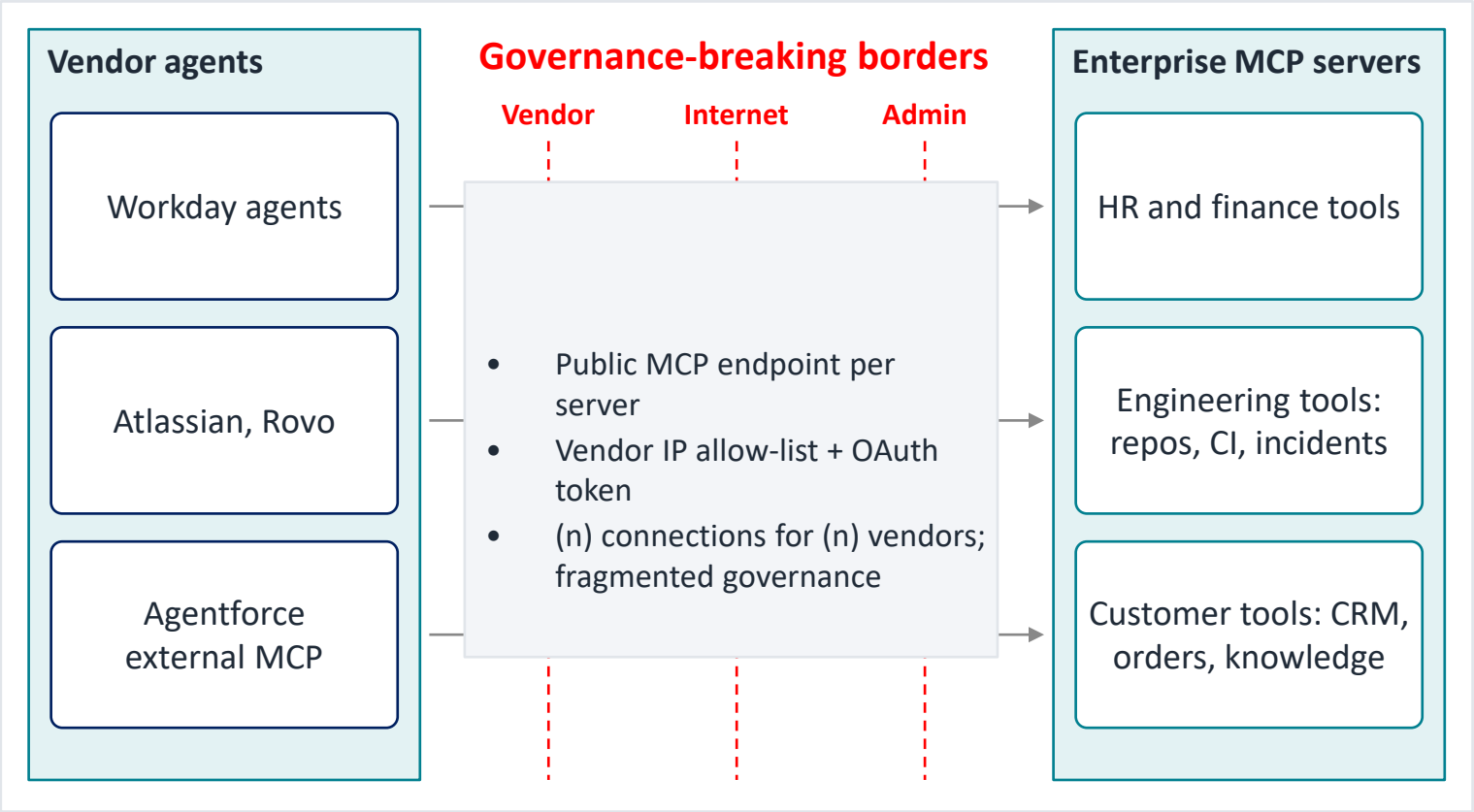


Results

- Vendor connects all terminate in one zero trust vendor VPC.
- The landing zone holds only PrivateLink endpoints, a NetFoundry connector and policy enforcement; vendor traffic is mapped to a vendor identity and authorized per service. Vendors never see internal addresses.
- Private connections to each enterprise site with unified governance, control and visibility.

NetFoundry

Use case #6: network defined governance for vendor agents calling enterprise MCP servers



Challenges

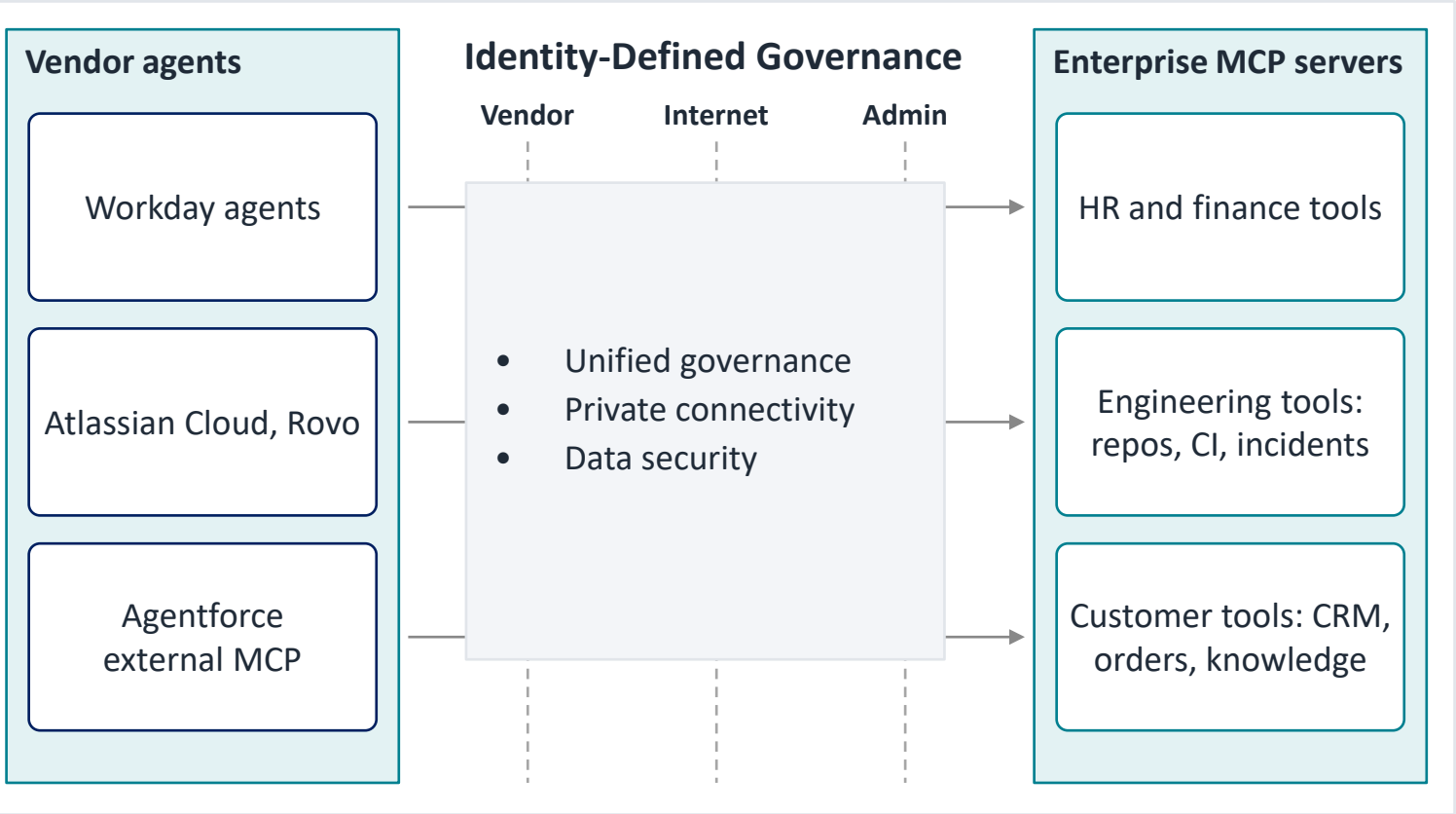
- Vendor agents are the fastest-growing consumers of enterprise MCP tools.
- Each MCP server compounds the network tax – each is a new network-facing application with its own allow-list, token handling and logs.
- A prompt-injection can result in enterprise data being exfiltrated from the MCP servers.

Network Defined Connectivity and Governance

The enterprise exposes MCP servers behind vendor IP ranges and OAuth. Every MCP is another door to enterprise data, and another door outside of governance. Every MCP compounds the network and firewall tax.

NetFoundry

Use case #6: identity defined governance for vendor agents calling enterprise MCP servers



Identity defined connectivity and governance

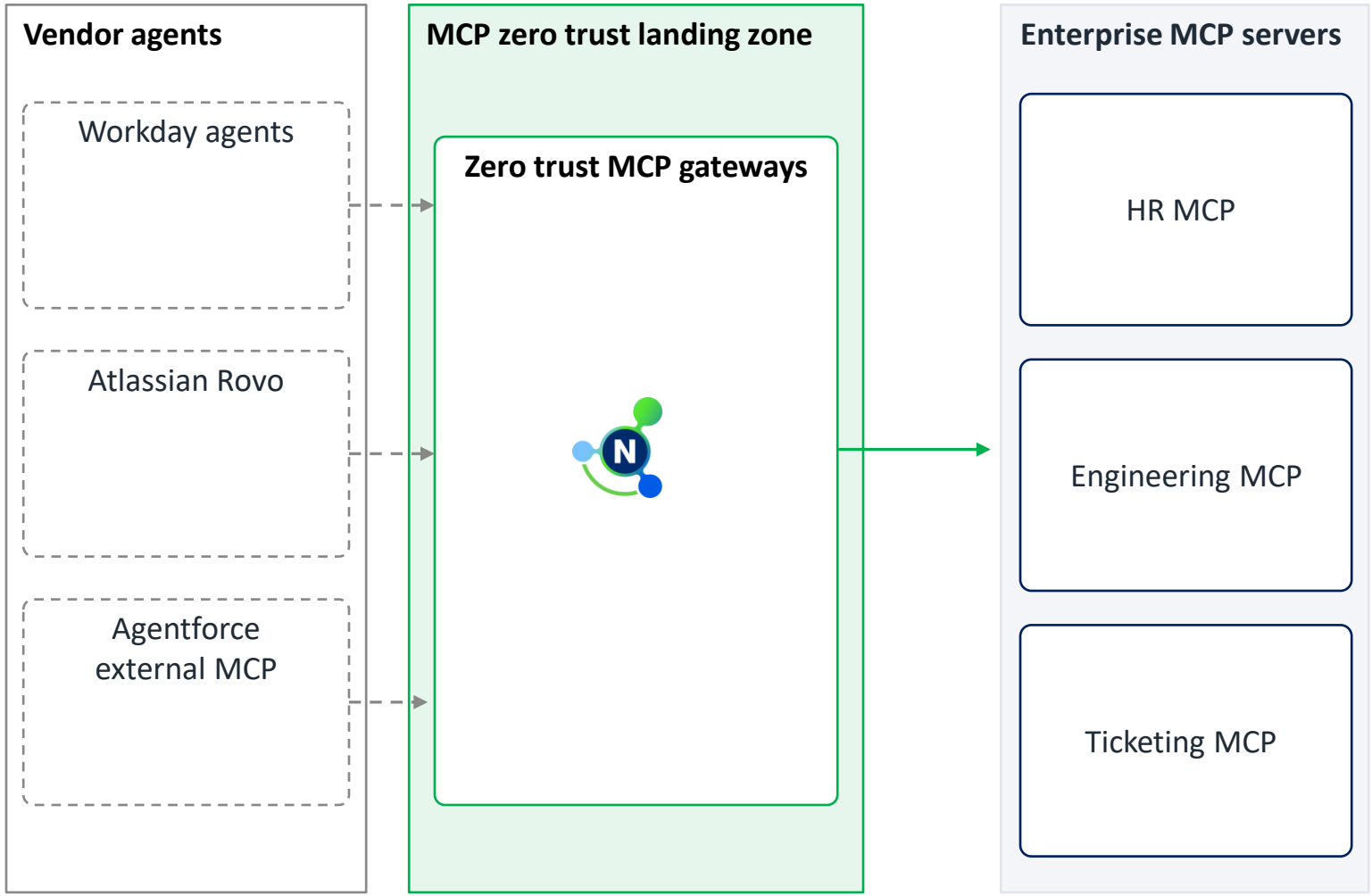
One governed door instead of one door per MCP server. Vendor agents are transformed into identities with per-tool authorization; MCP servers become invisible and unreachable to unauthorized endpoints.

With NetFoundry

- Publish exactly one thing: the NetFoundry MCP gateway, hardened, allow-listed and OAuth/mTLS-fronted, the only door for every vendor agent.
- The gateway authorizes per vendor agent and per tool, then reaches the real MCP servers over the fabric; those servers are dark to the Internet.
- Where the vendor allows, flip the direction: an enterprise worker on the fabric pulls from the vendor API and nothing is published at all.

NetFoundry

Example of use case #6: zero trust MCP gateway as the only published endpoint



Results

- Per-vendor-agent, per-tool authorization: Workday's agent gets HR tools, not customer tools. Enterprise MCP servers have no public name or port.
- Centralized governance, visibility and controls across all (n) vendors.
- Eliminate the networking and firewall tax and vulnerabilities – compromised service accounts no longer provide access, and existing malware can't easily exfiltrate data

NetFoundry