NetFoundry

Developers of **OpenZiti**

NetFoundry Smart Segmentation

Make the most important assets unreachable,
without touching firewalls

Solution overview
August 2026

NetFoundry’s New Approach: Smart Segmentation

1. Simplify

- Secure priority workloads immediately – don’t need to map every flow first
- Secure priority workloads immediately – don’t impact other workloads
- Meet OT, on-prem and cloud requirements, including legacy

2. Stop risking uptime

- Stop touching firewalls and networking and avoid the ACL ceiling
- Make workloads completely unreachable to avoid patching fire drills
- Progress according to business priorities – not firewall constraints

3. Improve visibility and governance

- Replace ACLs with identities to unify e-w, n-s and remote access control and governance
- Default-deny model: every session tagged with the identity and policy which enabled it
- Centralize identity, policy, controls and audit

NetFoundry

Problem #1: The All-or-Nothing Approach

Map every flow, then change the firewalls — value arrives only at the very end

The network-based approach

Discovery before value

Months of flow mapping and dependency analysis before the first workload is protected. Traffic maps are stale the day they're finished.

Rules multiply

Every mapped flow becomes firewall and VLAN rules to write, test and maintain — thousands of rules for a single data center.

One miss breaks production

Enforcement is binary: a single unmapped flow blocks a production app — so enforcement keeps getting deferred.

How NetFoundry solves it

Start with one workload

Deploy an identity-first overlay for the highest-value workload first. No global flow map required — everything else is untouched.

Identities and policies replace rules

Auth-before-connect policy is written in terms of identities and services, not IPs and ports. A few policies replace thousands of rules.

Visibility as you go

Every connection is attributed to a named identity, not an IP — discovery happens continuously, per workload, as you migrate.

Value from week one: each protected workload is fully microsegmented from day one — at the workload level, not just the host level.

NetFoundry

Problem #2: Uptime Risk of Firewall and Network Changes

Segmentation built on the production network puts the production network at risk

The network-based approach

Every change touches production

Segmenting with firewalls, VLANs and ACLs means every policy update is a change to the live production network.

Change windows and review boards

Each change queues behind maintenance windows and approvals — weeks of tickets and change review board time.

Outages are the failure mode

A bad rule push can take down far more than the workload being segmented — and rollbacks are complex and risky.

How NetFoundry solves it

Zero network and firewall changes

Overlays deploy without touching firewalls, VLANs, ACLs or routing. The existing network keeps doing exactly what it does today.

Deny-all inbound by default

Workloads make auth-only outbound connections to a private overlay — there are no inbound holes to open, manage or get wrong.

Overlays as code

Spin up, change and roll back segmentation as code, in minutes — no change windows, no review board queue.

Microsegmentation that can't take the network down — because it never changes the network.

NetFoundry

Problem #3: Yet Another Silo

New agents, consoles and policies — fragmenting controls instead of converging them

The network-based approach

Another stack to run

Network-based microseg adds its own agents, consoles and policy engine alongside firewalls, NAC, VPN and ZTNA.

Fragmented controls and visibility

Each tool sees only its own slice — blind spots and policy gaps appear at every border between them.

Policy drift

The same intent is expressed differently in every silo — audit and remediation get harder with each tool added.

How NetFoundry solves it

One platform, many use cases

The same identity-first platform delivers microsegmentation, remote access, site-to-site, B2B, APIs and AI workloads.

Unified policy, visibility and audit

One identity-based policy model and one view across clouds, sites and data centers — with visibility, control and audit built in.

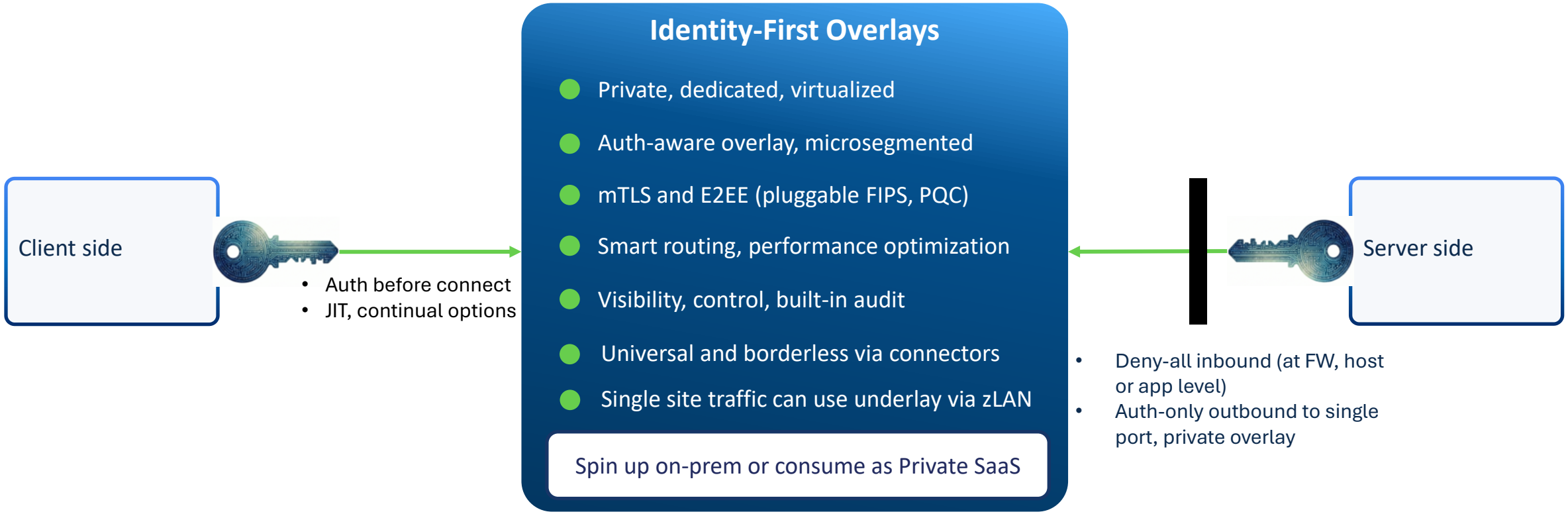
Consolidate instead of adding

Gradually retire VPNs, firewall rule sprawl and point tools as workloads migrate — reducing silos rather than adding one.

Admin, tech, cloud and network borders are erased — controls, visibility and policy converge instead of fragmenting.

NetFoundry

NetFoundry's Identity-First Connectivity™ Approach



1 **No lateral movement**
No network access. Includes AI agent sandboxing.

2 **Vulnerability & credential mitigation**
No unauthorized access. Proof of possession, continually authenticated, JIT options.

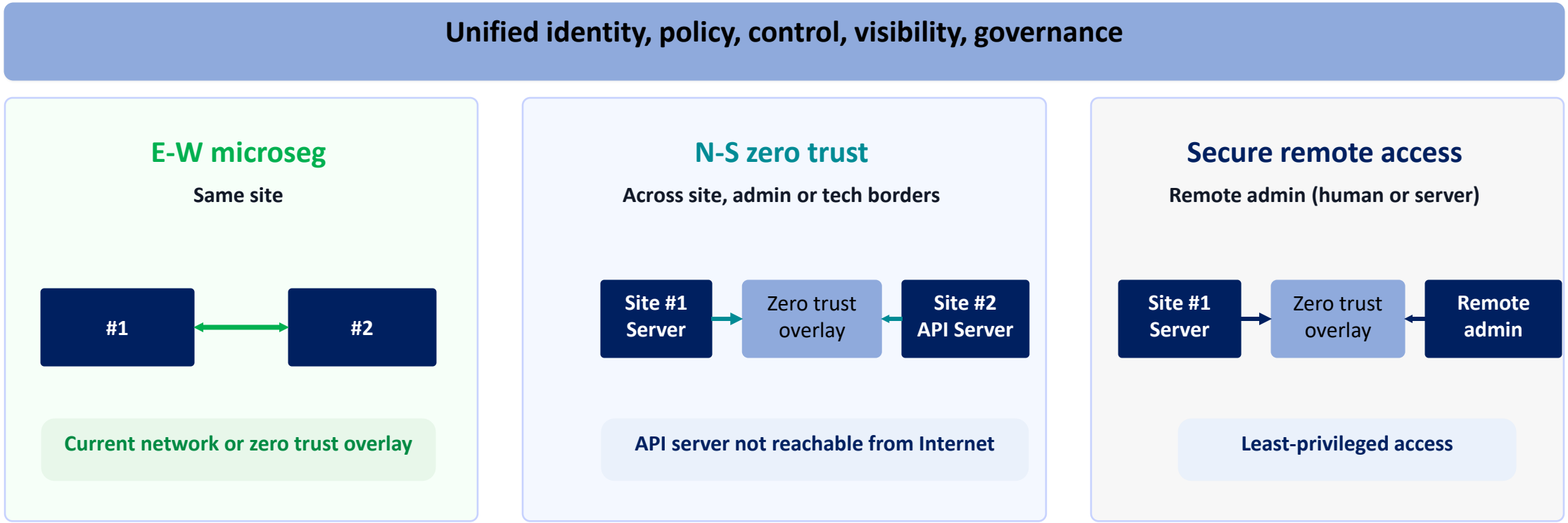
3 **Operations and performance**
Identities and policies replace firewalls, VPNs, networking. Spin up overlays as code. HA and performance optimization across overlay.

4 **Remediation, compensating controls, audit**
Less breaches, better visibility, built-in audit, less compensating controls

NetFoundry

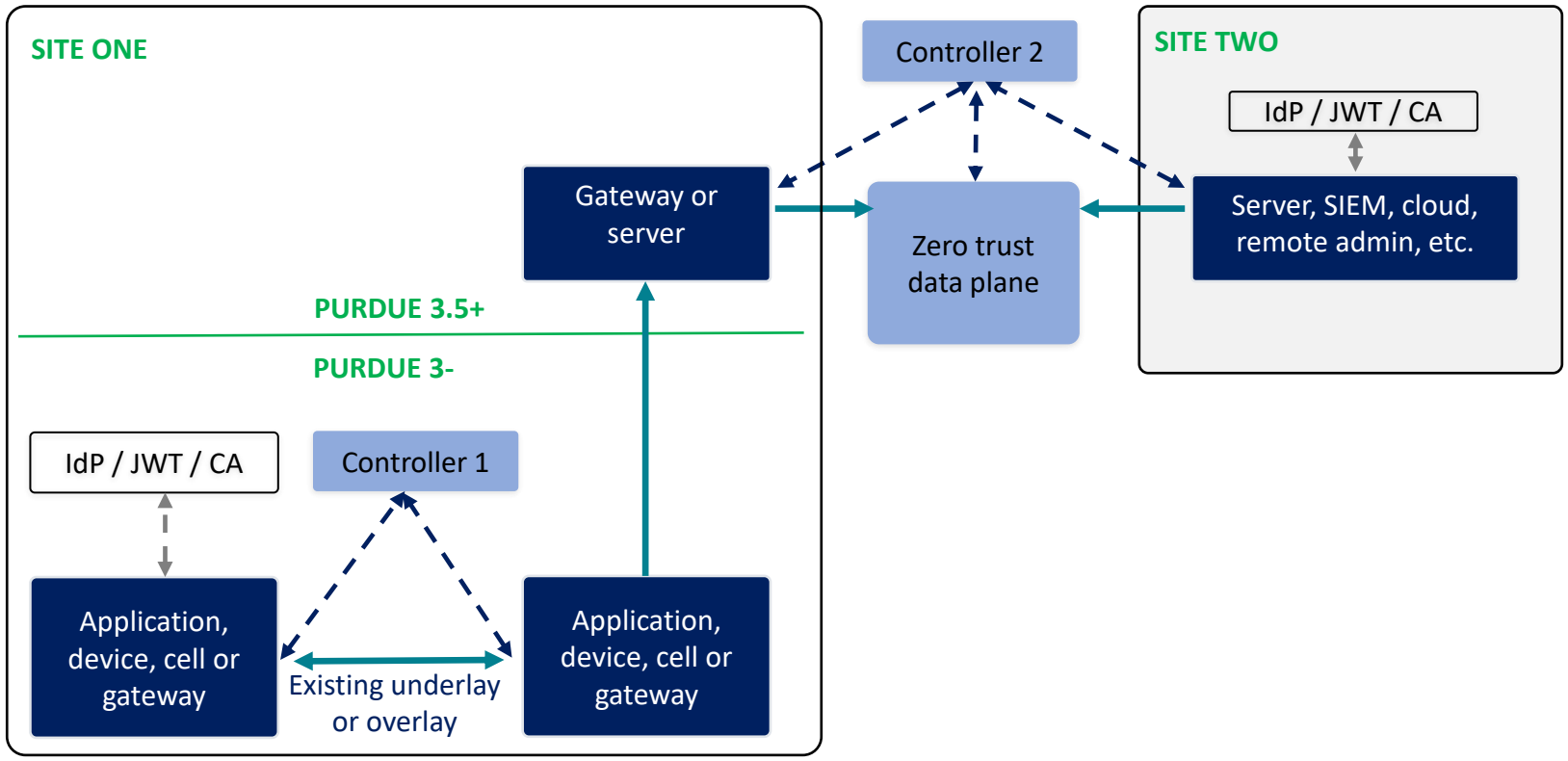
NetFoundry

Borderless, Unified Smart Segmentation



- Identities for human, OT, servers, AI – NetFoundry bootstrapped and 3rd-party
- Borderless and universal - connectors for applications, devices, gateways, firewalls
- Use of underlay and Zero Trust-native overlay networks – all sessions authorized first and initiated outbound
- Solution is self-hosted or NetFoundry hosted
- Includes mTLS, E2EE (pluggable FIPS, PQC), MFA, posture
- Rich integrations with operational and security systems

Unified Visibility and Control of Microsegmentation, Zero Trust and Remote Access



Legend

Policy enforcement points (PEPs)

- PEP in the app (SDK), device or gateway
- Multiple identity and auth options
- Options for legacy, clientless, JIT, IPsec

Virtualized overlay

- Only accepts PEP authorized sessions
- Software-only, deploy as code in minutes
- Self-hosted, NetFoundry hosted or hybrid
- Distributed controller clusters

Simplified Policy Creation, Testing, Enforcement

Organizations combine these options to get immediate wins with a realistic path towards full microseg

1. Application-based

- Create the policy directly from the application via NetFoundry SDK, or from the host via NetFoundry connector
- Run in passive mode
- Turn on enforcement when ready

Best method when available. Direct data, action immediately, without collateral risk.

2. Host-based

- NetFoundry connector (host connector or zero trust firewall) creates the policy
- Run in passive mode
- Turn on enforcement when ready

Direct data, action immediately, with minimal collateral risk.

3. Network-based

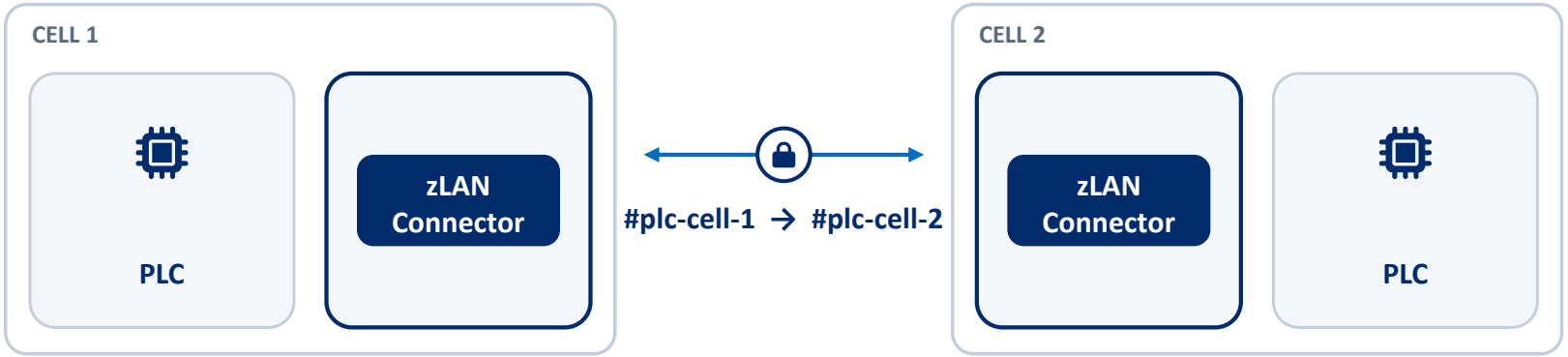
- Get data (cloud, IPFIX, VM tags, NetFlow, DHCP, DNS, NSX/DFW (if avail), SPAN/TAP, syslog, etc.)
- Enrich (AD, Okta, CS, Defender, SentinelOne, SN, Claroty, Armis, Dragos, Nozomi, etc.)
- Use the data to build policies, run in passive mode, turn on enforcement

Traditional microseg discovery but NetFoundry derisks the enforcement and destination.

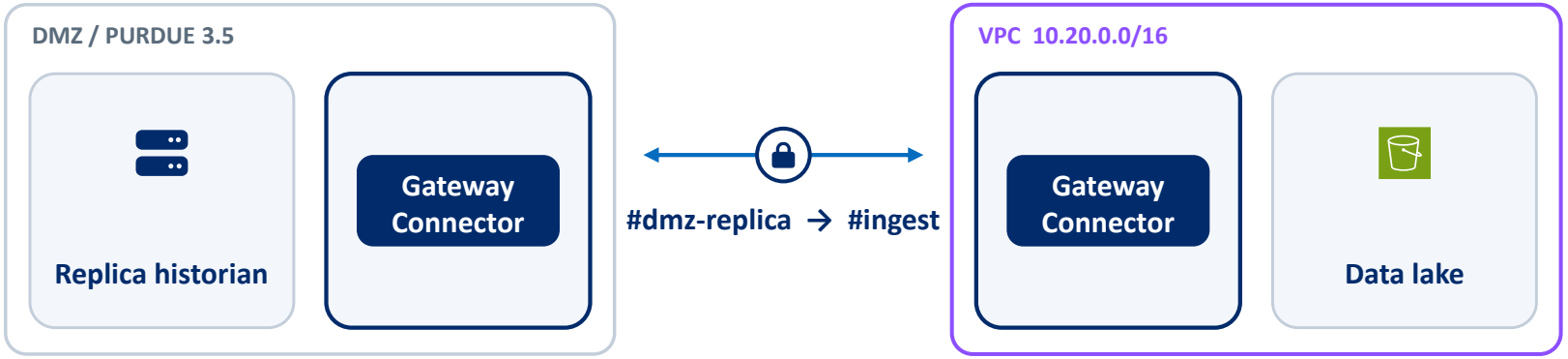
NetFoundry

Smart Segmentation for OT: Implementation Choices

1 **zLAN connector**: OT inter-cell: only authorized flows between cells



2 **Gateway connector**: plant historian replica streams to the cloud data lake

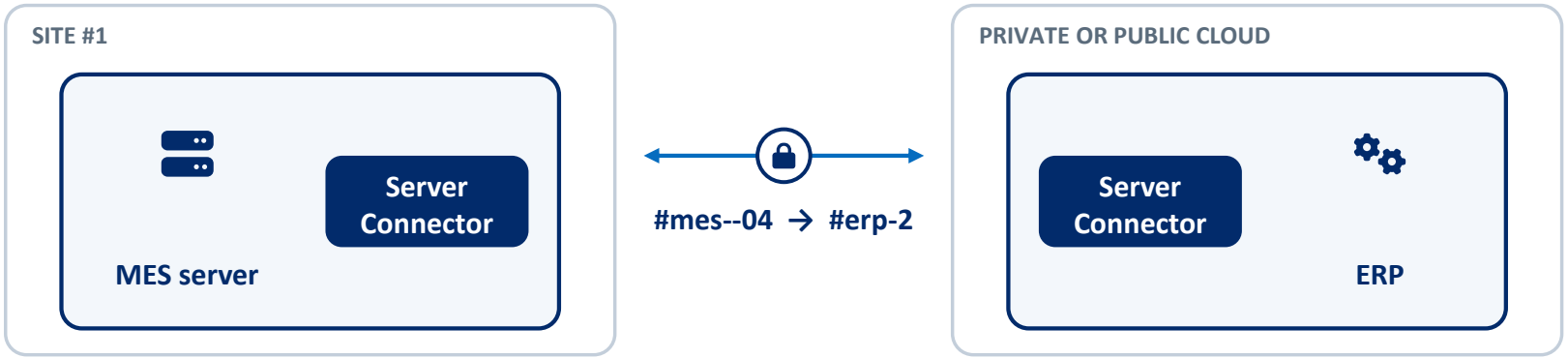


- zLAN connector often deploys on existing top of cell hardware and routes across the existing network.
- zLAN connector starts as a filtering mechanism for cell-to-cell traffic (equivalent to top of cell firewall), but enables progression to zero trust over time.
- zLAN connector has built in zero trust connector for north-south or remote access.
- Gateway connector deploys on existing industrial compute (VM, container, binary) or on a standalone box. It routes on a zero-trust overlay.

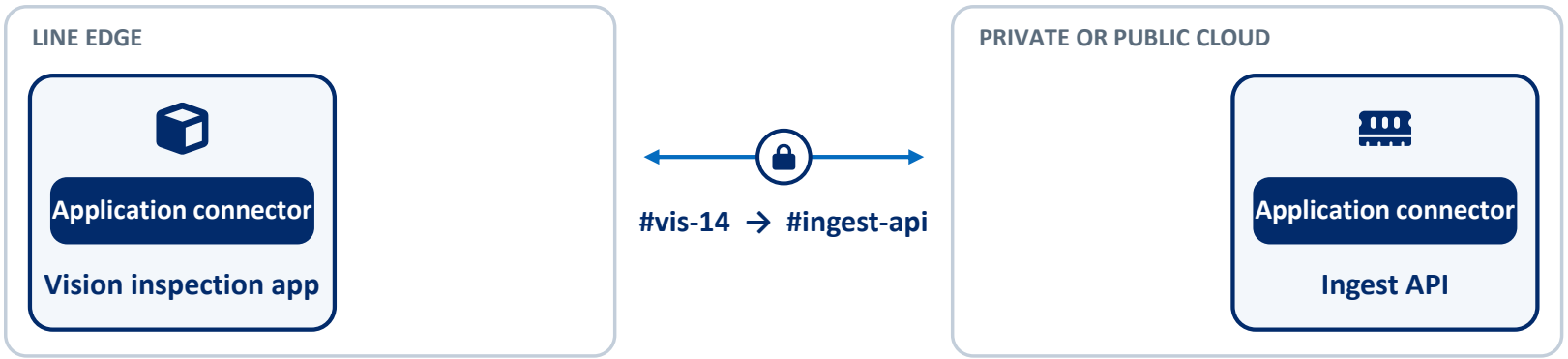
NetFoundry

Smart Segmentation for OT: Implementation Choices

3 **Server connector**: MES exchanges orders and confirmations with ERP



4 **Application connector**: vision inspection app posts telemetry to its cloud API

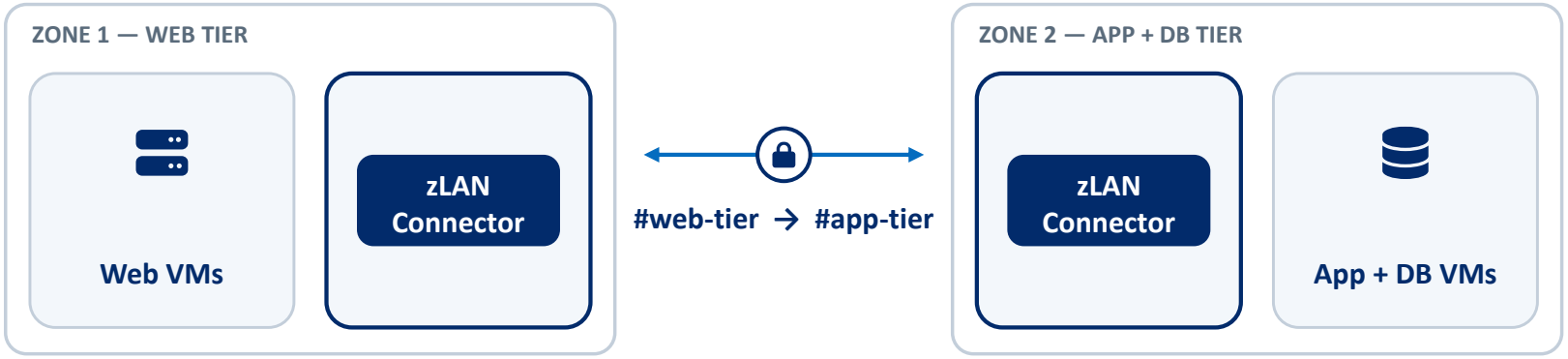


- Server connectors are deployable on existing compute and extend the zero trust connection to the server.
- This simplifies networking and firewalls, while improving segmentation, control and visibility.
- Application connectors use SDKs to make the application itself be zero trust. This eliminates need for separate connectors and is the ultimate in segmentation, control, visibility and governance.
- All connectors also support remote access and north-south zero trust

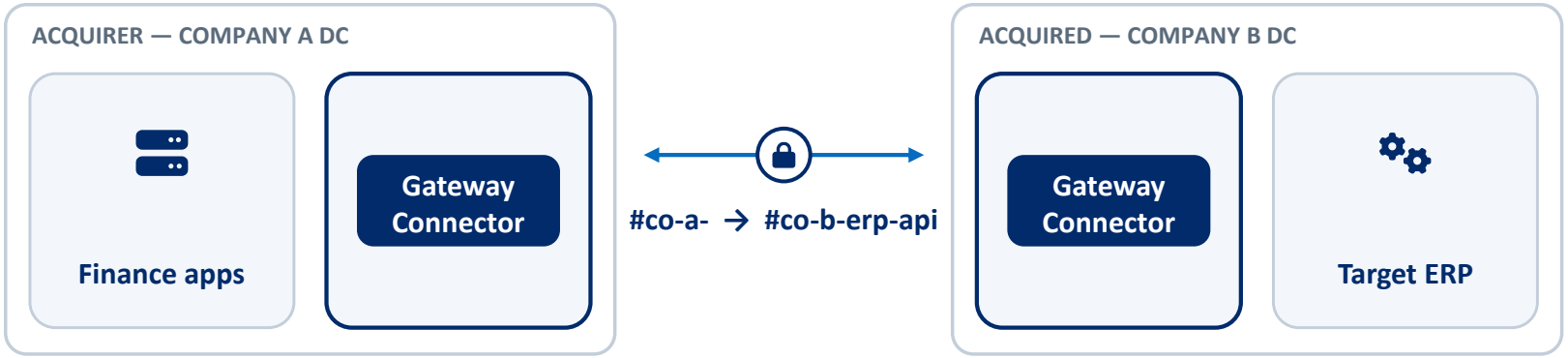
NetFoundry

Smart Segmentation for DCs and Cloud: Implementation Choices

1 **zLAN connector**: DC east-west: only authorized flows between zones



2 **Gateway connector**: M&A day one — app access without merging the networks

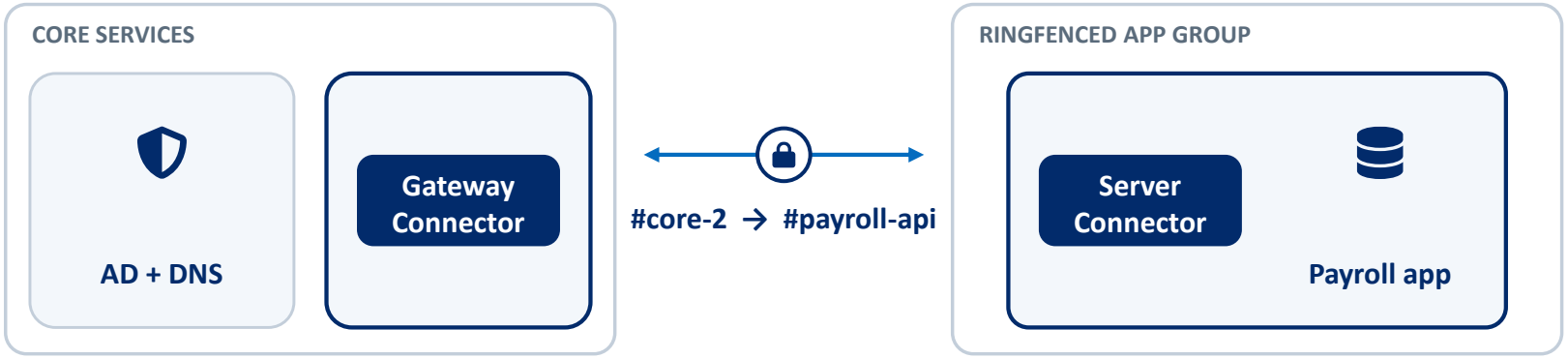


- zLAN connector often deploys on existing hardware and routes across the existing network.
- zLAN connector starts as a filtering mechanism but enables progression to zero trust over time so orgs don’t hit the firewall ACL problem.
- zLAN connector has built in zero trust connector for north-south or remote access.
- Gateway connector deploys on existing compute (VM, container, binary) or on a standalone box. It routes on a zero-trust overlay and brings zero trust to any

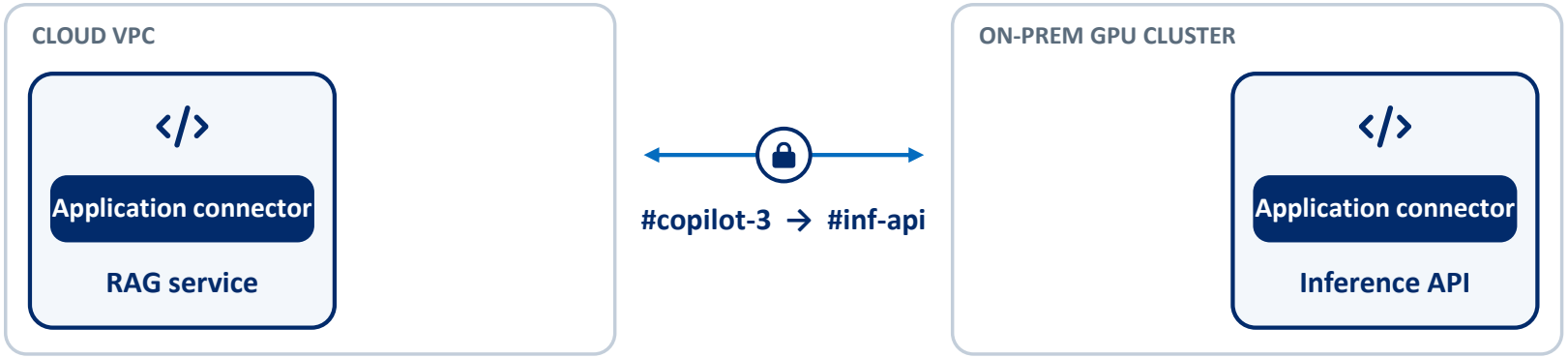
NetFoundry

Smart Segmentation for DCs and Cloud: Implementation Choices

3 **Server connector**: ringfencing the crown-jewel application



4 **Application connector**: cloud copilot calls its private inference API on the GPU cluster



- Server connectors are deployable on existing compute and extend the zero trust connection to the server.
- This simplifies networking and firewalls, while improving segmentation, control and visibility.
- Application connectors use SDKs to make the application itself be zero trust. This eliminates need for separate connectors and is the ultimate in segmentation, control, visibility and governance.
- All connectors also support remote access and north-south zero trust

NetFoundry

NetFoundry

Case Studies

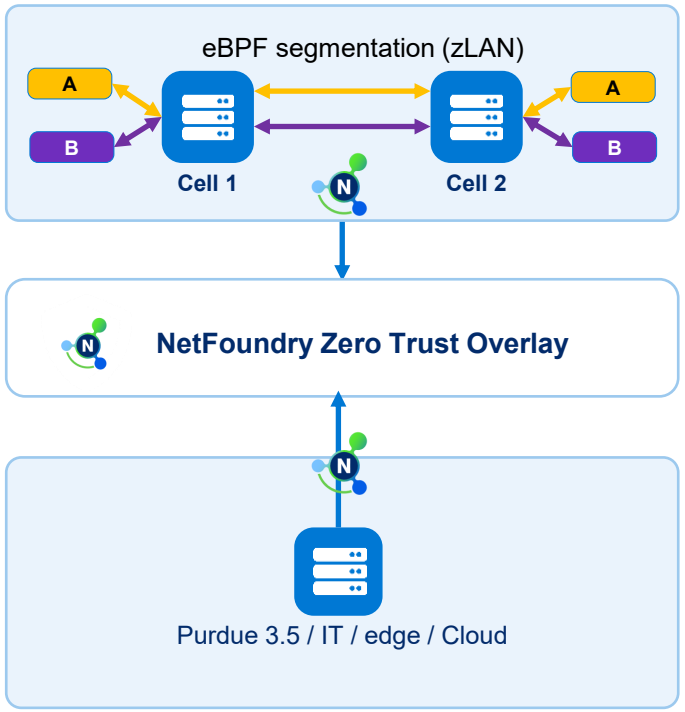
Zero Trust Platform for Fortune 50

One identity-based overlay spanning OT segmentation, machine-to-cloud data flows, and secure access

1. OT segmentation

Customer's requirements:

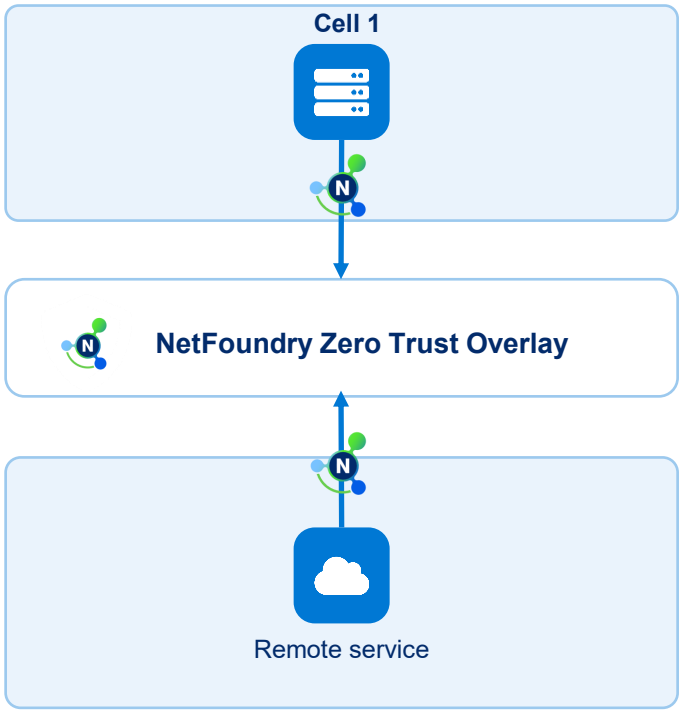
- No app, device, firewall, hardware changes.
- Support layer 2.
- Support on prem and hybrid.
- Enable future zero trust extension



2. Machine to cloud

Customer's requirements:

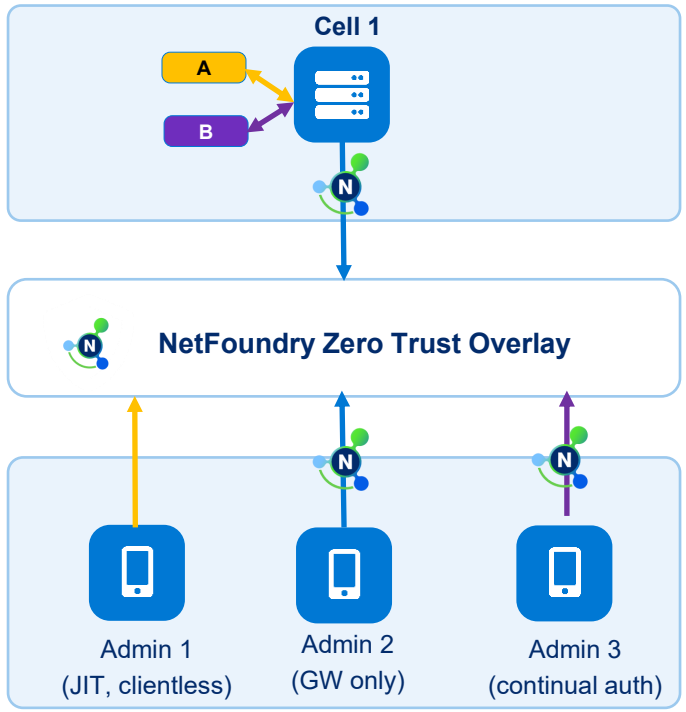
- No open inbound ports
- Microsegmented outbound
- Connect anywhere with single model
- No network, VPN, FW dependencies



3. Secure access

Customer's requirements:

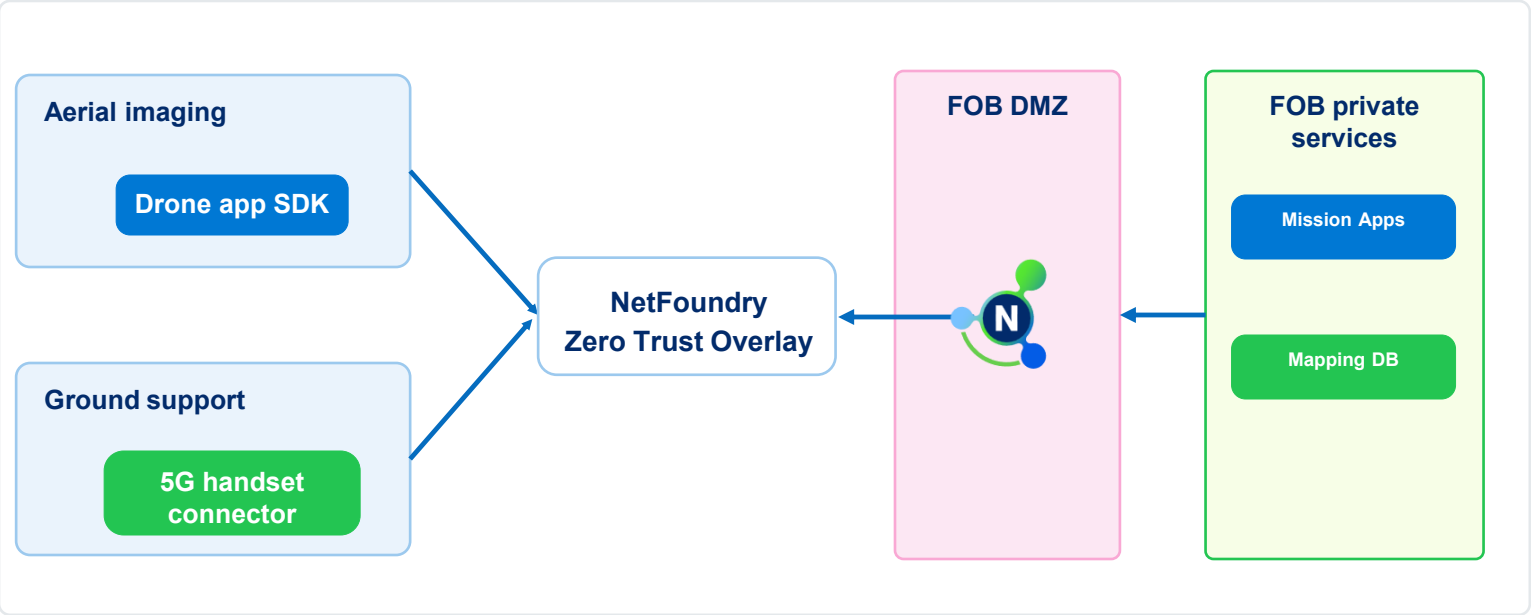
- Support layer two to avoid on-site work
- JIT, clientless, continual auth options
- Microsegmentation – no VPN level access
- Central visibility and control



NetFoundry

Private 5G Zero Trust for Drones

Air-gapped on-prem overlay connects drones, handsets, mission apps, and mapping databases



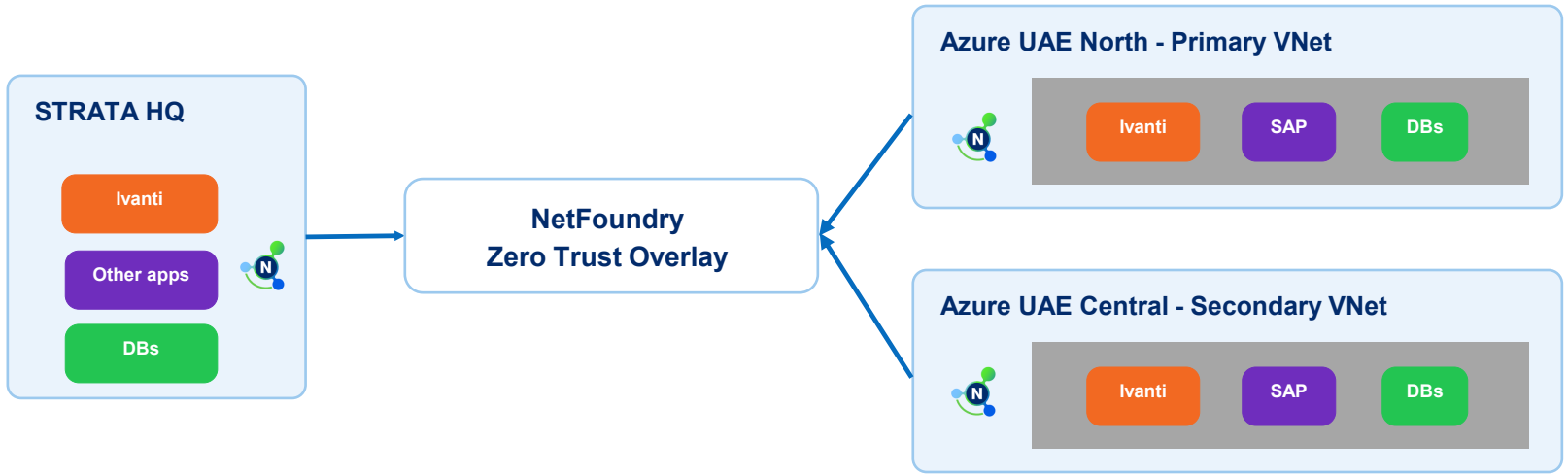
Customer's requirements:

- Drone to database and drone remote controls without public network exposure
- High-low security environment demarcation
- No RFC 1918 and NAT complexity when connecting with other units

NetFoundry

Zero Trust for STRATA Azure Migration

Outbound-only virtual gateways connect specific HQ and Azure workloads without site-level exposure

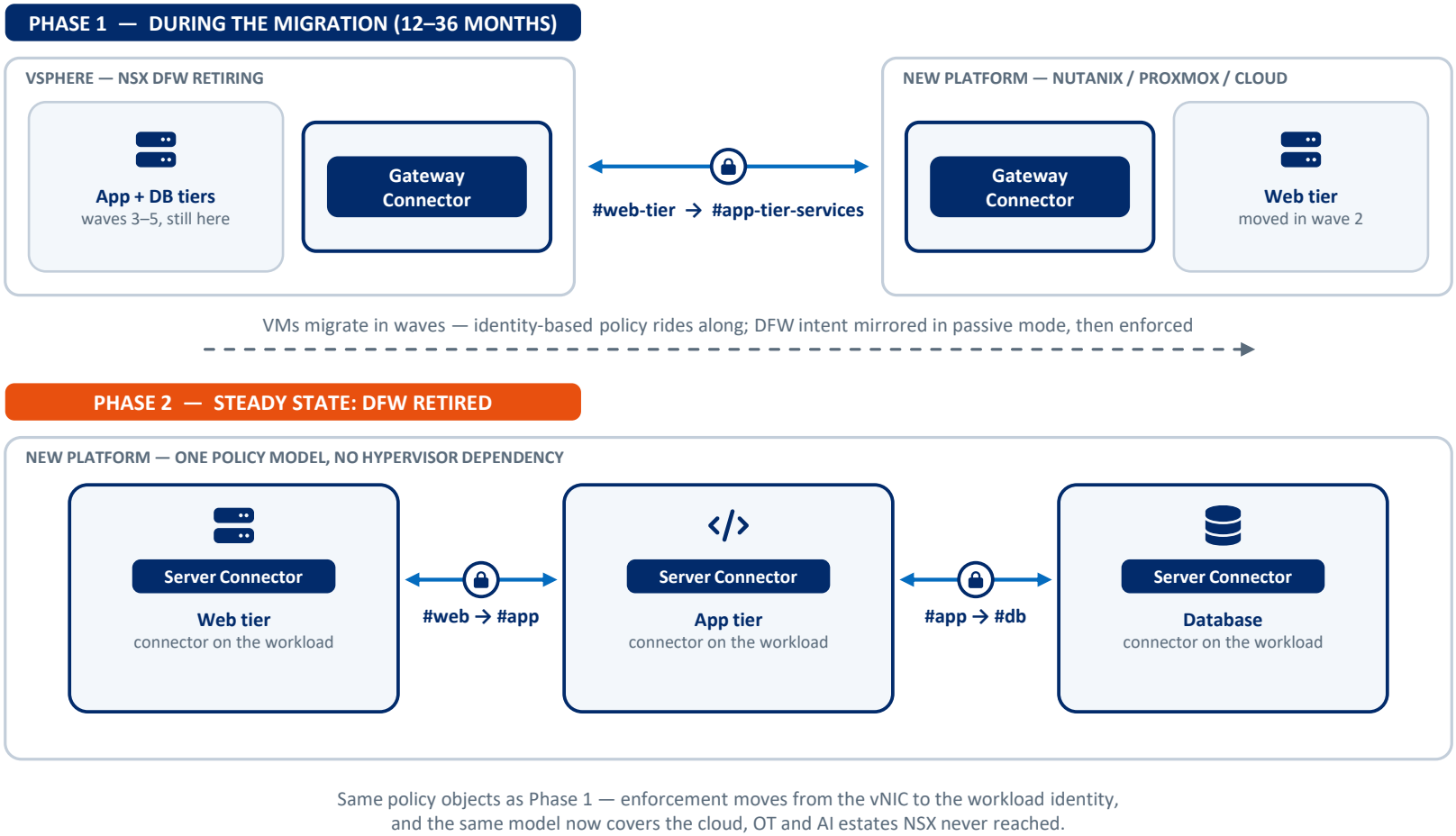


Customer's requirements:

- No public endpoints on Azure
- No inbound to HQ
- No dependency on IP addresses, cloud, infrastructure

NetFoundry

Replacing NSX DFW: Segmentation That Survives the VMware Exit



Same policy objects as Phase 1 — enforcement moves from the vNIC to the workload identity, and the same model now covers the cloud, OT and AI estates NSX never reached.

- NSX DFW enforces in the ESXi kernel — policy evaporates the moment a VM leaves vSphere.
- Migrations run in waves for 12–36 months; app tiers are split across two platforms the whole time.
- Phase 1: overlay gateways front both platforms — DFW intent mirrored as identity → service policy, passive first, then enforced.
- Policy is attached to workload identity, not the vNIC — each wave carries its policy with it.
- Phase 2: server connectors on the tiers retire DFW for good — same policy, no hypervisor dependency.
- The same policy model now extends to the cloud, OT and AI estates NSX never covered.

NetFoundry

NSW DFW Migration-Window Segmentation: The Three Real Choices

	With NetFoundry	Accept the gap (today's default)	Re-anchor on zone firewalls
Continuity during the waves	Full — the same identity → service policy spans both platforms for the whole window	None — migrated VMs lose DFW policy entirely	Coarse zones only — nothing like per-app DFW policy
Pre-work before wave 1	Days — gateways on both sides; passive mode mirrors DFW intent, no mapping project	None — which is exactly the problem	Re-IP / VLAN redesign, firewall rule build-out, change tickets
Legacy & EOL workloads	Covered — a gateway fronts whatever can't take an agent	Exposed	Covered, but only at zone granularity
Network change required	None — overlay on existing compute	None	Heavy — VLANs, routing, traffic hairpins through firewall pairs
Fidelity vs DFW	Comparable intent, enforced per identity — and no listening ports at all	—	Much coarser; address-based rule sprawl replaces group policy
End state after migration	Permanent segmentation layer — extends to cloud, OT and AI estates	Unsegmented; the project restarts from zero	Rule sprawl and hairpin latency, welded to the network design
Cost shape	Overlay subscription — the same asset is reused after the migration	Breach risk carried for 12–36 months	Firewall capacity plus ongoing change labor

NetFoundry