NETFOUNDRY™

SPIN UP YOUR NETWORK

Overview

Contents

Embedding zero trust networking in apps and solutions

Cybersecurity attacks cost us over $1 trillion annually (McAfee). There are many reasons for that outrageous cost, but networks are one of the top vulnerabilities. In fact, 'scan and exploit' network attacks were identified as the #1 infection vector in the 2021 IBM X-Force Threat Intelligence Index.

To defend against these attacks, application and solution providers are now embedding zero trust networking inside their solutions. Embedding functions like zero trust networking is part of the next phase of 'shift left', which started with providers embedding functions like SSL/TLS encryption into their solutions.

Zero trust networking as a 'library' for app and solution providers

The NetFoundry platform and SaaS is used by customers from Fortune 100s to unicorn cybersecurity start-ups to embed zero trust networking into apps, APIs, browsers, security solutions, proxies, management platforms and databases.

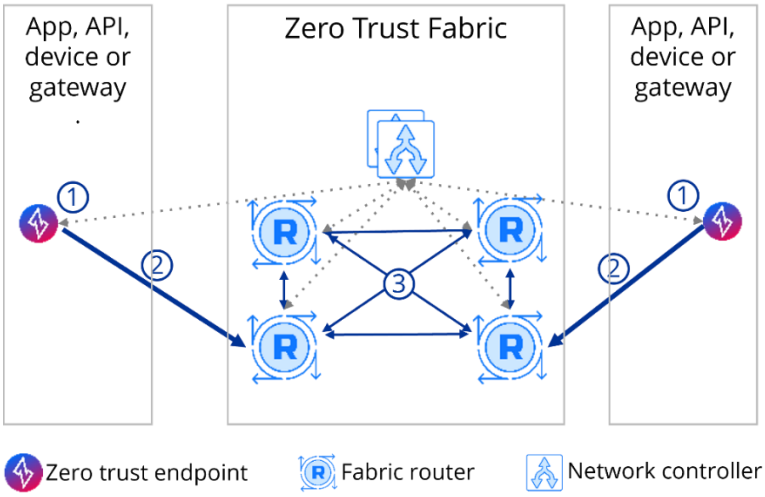
That wide range of solutions and customers is because NetFoundry enables providers to do everything from white-label entire zero trust solutions to integrate specific zero trust networking functions inside their solutions. A few common drivers across this range of use cases and customers:

- **Offer more robust cybersecurity solutions**. Providers add zero trust SD-WAN type functions to sell to their customers as new services, or to improve their existing solutions.
- **Improve business velocity**. Providers no longer need to depend on their customers to provision MPLS, firewalls, VPNs and private APNs. Providers instead enable their customers to access their services securely and reliably via Internet connections, from any device or cloud. This allows providers to both improve and standardize customer access.
- **Shift left to improve security, visibility and control**. Providers insert secure networking into the heart of their development and delivery pipeline, gaining stronger security with end-to-end control and visibility.

The NetFoundry platform is exposed as software-only SaaS, including hosted private network fabrics. NetFoundry also open sourced the core software, OpenZiti, and NetFoundry is the leading maintainer of the software.

Architecture

The NetFoundry platform, APIs and SDKs enables providers to offer new zero trust networking services to their customers, and to build zero trust networking into existing offers. Because the resultant zero trust networks are app-specific, multicloud native, API-first and programmable, it enables providers to offer simple, extensible and scalable services. Providers' customers securely and reliably access the providers' services from any Internet connection, without disrupting or adding to their WANs or infrastructure:



1. **Endpoints**. NetFoundry uniquely protects apps **from** the networks. NetFoundry provides SD-WAN-like software, except it goes anywhere, even inside a provider's app, or on an IoT device, and has zero trust built in. The resultant "AppWANs" deliver traffic for one app, or multiple apps.

2. **Authorize before connect.** Each endpoint uses X.509 based identities, without pre-shared keys, to securely authenticate and request ephemeral overlay network connections based on attribute based authorization, posture checks and MFA when applicable. A key result of this is bad actors can non longer access the apps over networks. All of the involved zero trust functions are provided by NetFoundry as Saas.

3. **Overlay network fabric**. Cloud native virtual routers enable bidirectional overlay networking between endpoints, initiated from either side, by bridging both sides (each side opens outbound from its network towards the fabric). These programmable routers function in a mesh with the endpoints to dynamically optimize routing according to real-time conditions. This dynamic routing is critical for resiliency and quality perspectives - no longer are packets at the mercy of Internet weather.

Edge to cloud use cases (users, IT and IoT apps)

Case studies: Java apps (video, article); Private 5G (Microsoft); web servers; user and admin access (Ramco); Postgres; edge compute (Arrow); IoT analytics (TOOQ); MPLS replacement (FWD Insurance)

These use cases are applicable for providers focused on their customers' end users or devices accessing the provider's services such as:

1. Expand cybersecurity services to include zero trust networking, either as another service to sell to end customers, or to enhance existing services.
2. Enable customers to consume apps or services from the solution provider without needing to extend or disrupt traditional WANs, and without implementing different solutions for different edges and clouds.
3. Implement stronger security. The resultant zero trust networking is more secure than VPNs and MPLS. For example, a result is enabling both sides to reply complex ACLs with one inbound policy: deny-all! The security is for both the provider and the customer – each app session can't be accessed from the networks, and neither can the app or database servers. The secure networking goes anywhere the app goes – no need for agents.
4. Expand addressable markets to the most security and compliance sensitive vertical and organizations. For example, the 2022 US Government mandate to upgrade to zero trust networking.

Server to server use cases (APIs, multicloud, hybrid cloud)

Case studies: Kubernetes (Ozone); APIs (Oracle); multicloud (IBM); hybrid cloud (CERM); zero trust MSSP services (Ohka)

Highlighting two of the main use cases:

1. **API security**. Orgs have strengthened API security by improving the **authorization of the users** of the apps. With NetFoundry, orgs **authorize the network connection** to the API servers. The result is the API server becomes inaccessible from the networks, even for B2B APIs. That is important because the top 10 OWASP API attacks require the attacker to get network access to the API server.
2. **Database access**. Data access has become challenging to secure because app servers and web servers are increasingly in different private data centers, edges and clouds then the datastores which they query (the data stores tend to be more centralized). With NetFoundry, zero trust database connections, including agentless via zero trust JDBC drivers,

replace existing VPNs and ACLs. This is used internally by solution providers, as well as externally (e.g. to secure data collection agents on customer premises or in customer VPCs).

Remote management use cases (internal and B2B)

Case studies: B2B (Lemongrass SAP); Data collectors (e.g. SIEM and APM solutions), ops and CI/CD database access, remote management without bastions; multicloud remote management (Novis)

Three main use cases:

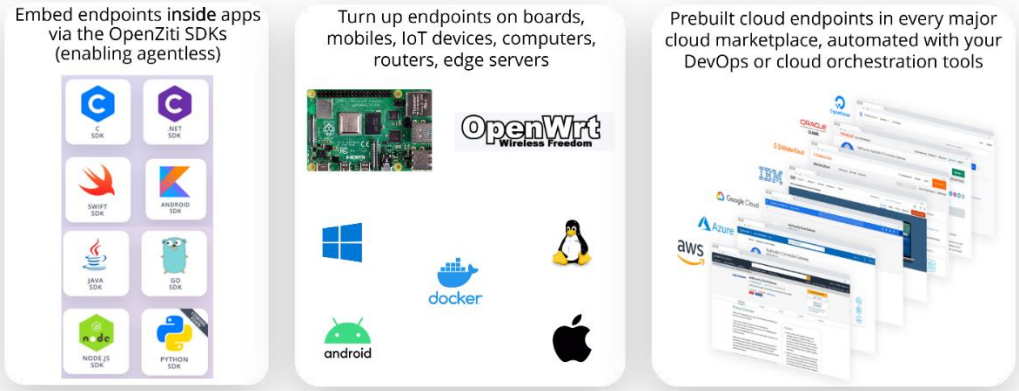
1. **VPN/bastion replacement**. On the security side, NetFoundry enables all inbound firewall ports to be closed (not the case with bastions and VPNs); enables app session level microsegmentation (rather than network, VLAN, or subnet level segmentation); and enables orgs to not need to pre-share keys or manage them. On the agility side, NetFoundry APIs enable CI/CD, DevOps and cloud orchestration systems (example: Ansible) to use the same zero trust solution as human admins. Meanwhile, API integrations with ticketing solutions result in automation scenarios like only enabling ephemeral connections for the life of an approved work order, auto closing them when the work is done.
2. **B2B remote management**. In many B2B cases, including IoT and cases in which a provider needs to manage an app or device on a third party network, secure networking has historically been very difficult. VPNs and MPLS often don't fit. In the cases when VPN is used, it is very difficult to cost-effectively scale, and often leads to finger pointing between the orgs, because neither has end-to-end visibility or control. Increasingly, it is understood that the VPN solution is simply not secure (hence the mandates of orgs like the US government executive mandate M-22-09 for agencies to move from VPN to zero trust). For these use cases, the NetFoundry solution has the same zero trust security benefits described above, and its app specific architecture enables third parties to run agents which only secure the B2B remote management sessions (this avoids the problem of admins running multiple VPN clients on the same device, e.g. for their internal needs, and for their B2B needs). The same solution protects the MSP or MSSP from being the next Kaseya-type victim.
3. **IoT remote management**. In the case of IoT devices, the zero trust security benefits, and software-only, API-first agility benefits apply. In addition, NetFoundry enables remote management without the need to provision private business APNs, or to deploy separate solutions for the management and the actual IoT data. In many cases, like Supermicro,

Microsoft and Arrow, NetFoundry is already built into the gateways. In other cases, the org simply deploys NetFoundry inside the app (no agent), or on devices like Nvidia Jetson and Raspberry Pi.

Appendix 1: architecture details

Endpoints

There are three main types of endpoints:

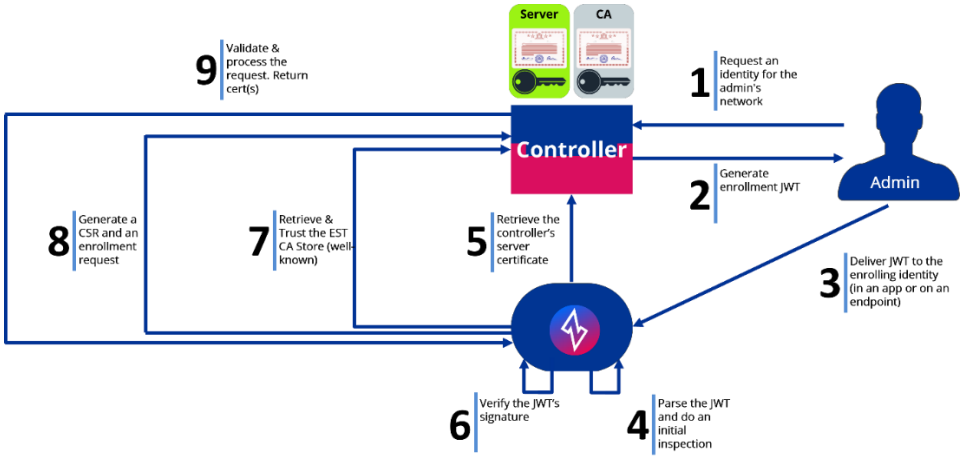


These endpoints enable providers to extend zero trust networking by embedding it directly into apps via the SDKs (agentless model), or enabling providers or provider customers to deploy agents or gateways. The endpoint software:

1. Contains X.509 identities and full network stacks (to serve every protocol and use case).
2. Give providers a cloud-managed control point. Visibility can be shared with customers and partners as well.
3. Enable MFA and posture checks when applicable.
4. Secure the app, data and device (make it unreachable from the networks).
5. Enable extensions into security and networking tools and solutions.

These endpoints are akin to SD-WAN CPE, except they go anywhere, and enable you to use Zero Trust principles instead of WAN principles.

Bootstrapped endpoint identification and authentication

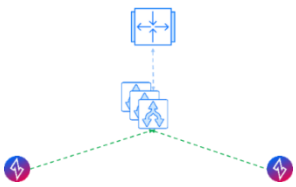


Above is the basic diagram of the endpoint identity and authentication (further details are here), which enables your endpoints, and only your endpoints, to access your private overlays.

Endpoints need to be authenticated and authorized via their X.509 identities to access your private overlay. The bootstrapping and Certificate Authority (CA) are provided as SaaS, and you can add your own CA (RFC 7030). The platform also supports PKCS #11 - enables you to store certificates in secure ways (e.g. HSMs).

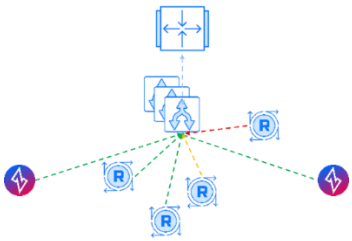
Least privileged access via distributed controllers

Your authorization and access policies are enforced via your private, NetFoundry hosted (in the SaaS model) or self-hosted (in the open source model) controllers:



Least privilege access can be paired with your IDP or SSO type solutions such that even if your IDP or SSO solution is compromised, there is still no network connection (layer 3). Similarly, if your NetFoundry solution is compromised, the attacker also often needs to thwart the IdP solution as well.

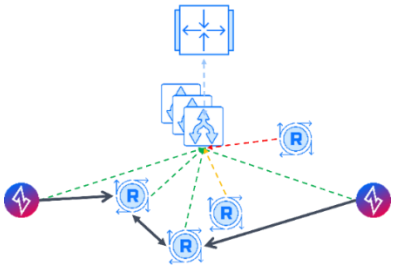
Private overlay fabrics



Your Routers function as combined routers/firewalls, but operate in the opposite model: instead of delivering packets unless told differently (IP based firewall rules), Fabric Routers deny all packets unless told differently (cryptography authorized flows). The Routers are hosted by NetFoundry in the SaaS, in every cloud marketplace and can be deployed as VMs anywhere. The Routers are governed by your policies (geofencing etc.) and are ephemeral – spin them up and down, programmatically. Your Routers form mesh overlay fabrics to:

1. Provide you with end-to-end control, across the overlay.
2. Enable you to close all your inbound ports. Instead, authorized NetFoundry endpoints will open outbound-only connections to authorized Routers. The Routers bridge the connections, enabling bi-directional data across the private overlay.
3. Provide you with optimized routing. This is detailed in the next section.
4. Enforce least privilege access on your overlay. This enables you to only grant permissions which are absolutely required for a given workload or session – no network level access. This applies to systems as well, for example a CI/CD system can only access certain ports on databases, and the connection will be ephemeral (only available during a code push).

Optimized, multipoint routing

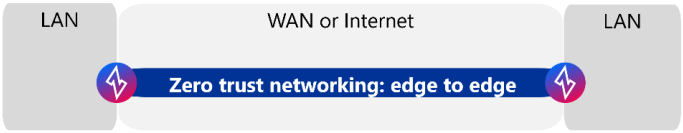


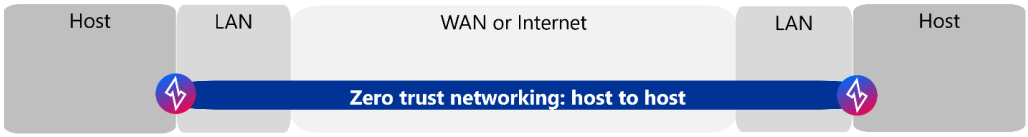
Real-time routing algorithms select the best path across your mesh, based on your metrics. In the example above, two of the four Routers are selected at first. This will change, automatically, as conditions change. Your Fabric Routers form your private, programmable Fabric, which can be used to:

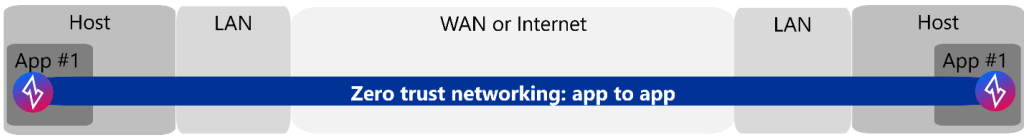
1. Enable your assets to deny all inbound connections, making your data unreachable from the networks
2. Enforce geofencing and similar policies. In the example above, the Router with the red connection was not eligible due to its location.
3. Use specific clouds for specific sessions.
4. Enable resiliency and improve quality (e.g. leverage routers across multiple edges and clouds). With Routers across multiple clouds and networks, there are many potential routes. When "Internet weather" makes certain routes perform better than others, or if certain routes have outages, the algorithms will automatically (per programmable policy) select the lowest latency routes.
5. Create ephemeral data planes (periodically spin up new sets of routers, creating a moving target).

Appendix 2: app-embedded, host-integrated, edge-integrated

Here is a quick visual of how each option extends zero trust networking towards the end-to-end solution of app-embedded, and then we'll look at each option:





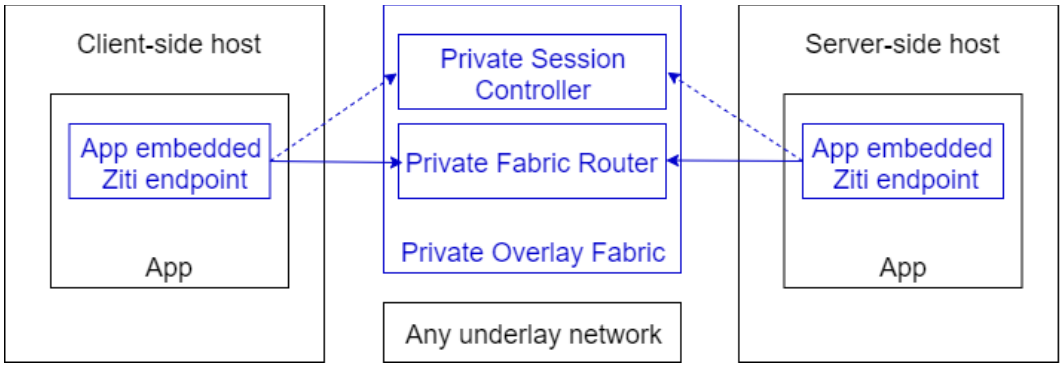


App-embedded option (app, browser, proxy, agent or driver embedded)

App-embedded (agentless) solves these problems:

1. **Security and control**. Use cases which require the strongest security and control can now securely connect without even trusting the hosts. The solution provider is no longer at the mercy of the networks.
2. **Topology**. Use cases in which it is difficult to deploy agents, including B2B APIs and third party endpoints. By embedding in the app, data or database, secure networking goes anywhere your app goes, eliminating DNS, VPN, etc. problems (this video shows how to use the SDKs).

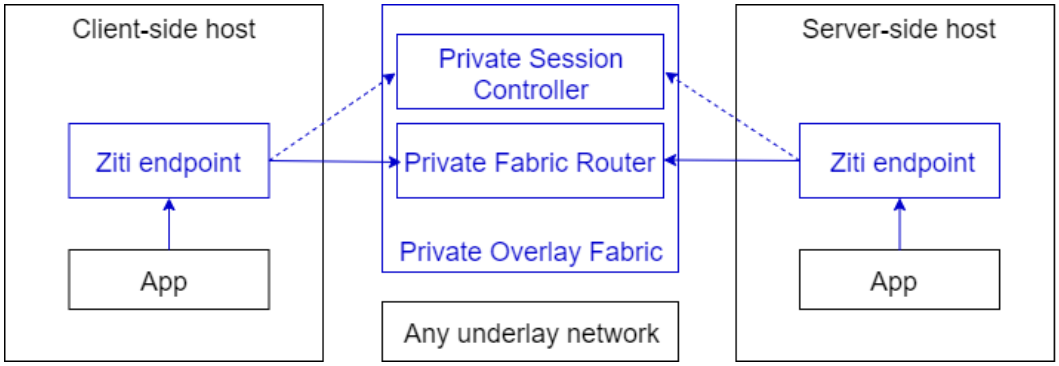
3. **Integrations**. Add code to existing security solutions, agents, proxies, browsers, API servers etc.



You can embed the software directly into the app, browser, proxy, API, database driver, etc (see the video in this blog post about "Zitifying" a Postgres driver).

Host integrated option (app specific agents for IT and IoT devices)

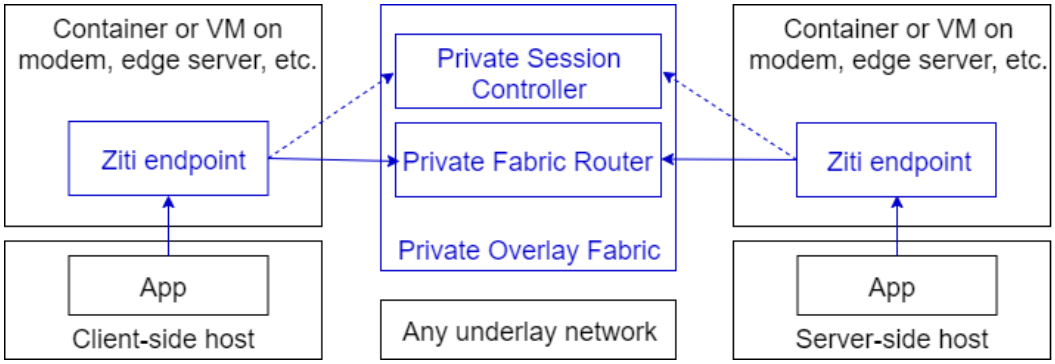
This option leverages NetFoundry software endpoints, which support every device type, OS and cloud, and are built on the OpenZiti SDKs described above:



Your endpoint is 'moved' from the app or browser to the device hosting the app.

Edge-integrated option (containers or VMs):

In this option, NetFoundry software endpoints are deployed as containers or VMs on modems, DMZ routers, edge servers and cloud instances:



This places your endpoints on aggregation points.