NetFoundry

WHITE PAPER

NetFoundry Identity-First™

Accelerate Secure Customer Deployments

Learn more at www.netfoundry.io

Secure networking replaces network security

Business networks (TCP/IP) don't enforce strong identities, continuous authentication and fine-grained, deny-by-default authorization and access. They are default insecure, default expensive to manage and default blockers to business velocity.

The solution is not to pile on more network security. We need a new, secure by default model in which networks enforce identity, authentication and deny-by-default, fine-grained authorization for every workflow (no firewall or VPN holes). We need to move from network security to secure networking.

The network security model doesn't work well in a hyperconnected, AI-enabled world

Why, despite all the money and time we pour into networking and security, can a single compromised identities take down entire ecosystems, from healthcare (e.g. UnitedHealth) to transportation (e.g. Jaguar Land Rover)? Because the networks are default open and ignorant of strong identity, authentication and authorization – networks are open and blind, depending on 'network security' systems to lock their doors. That network security model doesn't work well in a hyperconnected, AI-enabled world of almost infinite doors.

How attacks work when the enterprise network is blind and reachable

1. A session arrives at the network firewall. Unless it is from a known 'rogue' IP address, the firewall sends it to the DMZ. *Firewalls can only filter based on IPs, so they depend on systems like IdPs and DNS to prevent attackers from reaching the firewalls. Because the firewall is Internet-reachable, and can initiate connections into the DMZ, firewall vulnerabilities have resulted in large scale breaches.*

2. The DMZ has 'day two' security, such as WAFs, IDS, IPS. However, unless the attack has a previously identified 'bad' signature, the session is sent to the application server. *WAFs etc. are good but not 100% effective, especially because many attacks are snowflakes (no known signatures) or use compromised identities.*

3. The server attempts to authorize the session. However, the attacker's session is not terminated if the attacker has stolen credentials, or if the server/application has a zero day or business logic gap (or is not hardened, e.g a dev/test instance).

4. The attacker is now in the network and moves laterally to gather other credentials, set up tunnels, gather reconnaissance data, etc. Ultimately to execute its attack. *Microsegmentation would help but is very difficult to do with traditional, default open networking with coarse grained authorization and access.*

This example shows why the network security model is upside down - the model is vulnerable to digital identity (token, assertion, claim) theft and other vulnerabilities because it is default open and default permissive.

NetFoundry flipped the model from **"network security"** to **"secure networking"**

STEP 1

Close the doors. Sessions are not connected unless both sides of the session are strongly identified, authenticated and authorized. The policy enforcement point (PEP) moves to session initiation, and the network enforces identity, authentication and authorization. *In the example above, the session can't reach the network firewall to begin with, even if it has a stolen identity, as long as the policy requires proof of possession (PoP identity (which NetFoundry provides).*

STEP 2

'Day two' systems become part of a defense in depth strategy. Network and server firewalls can actually function as firewalls – block all inbound. The complexity of managing firewall ACLs is eliminated. WAFs etc. are no longer filtering the entire Internet – they focus on insider attacks. Teams are no longer racing against the whole Internet to patch vulnerabilities. *In the example above, even if the server or application was compromised, the attack would be unable to spread if the secure network is extended to the server or app (which NetFoundry provides).*

NetFoundry's secure by default model is implemented as programmable software which does not depend on infrastructure, underlay networks, edges or clouds. It enforces 'layer 7' identity, authentication and authorization principles at layers 3 and 4. This gives control, visibility, ease of use and security to network administrators.

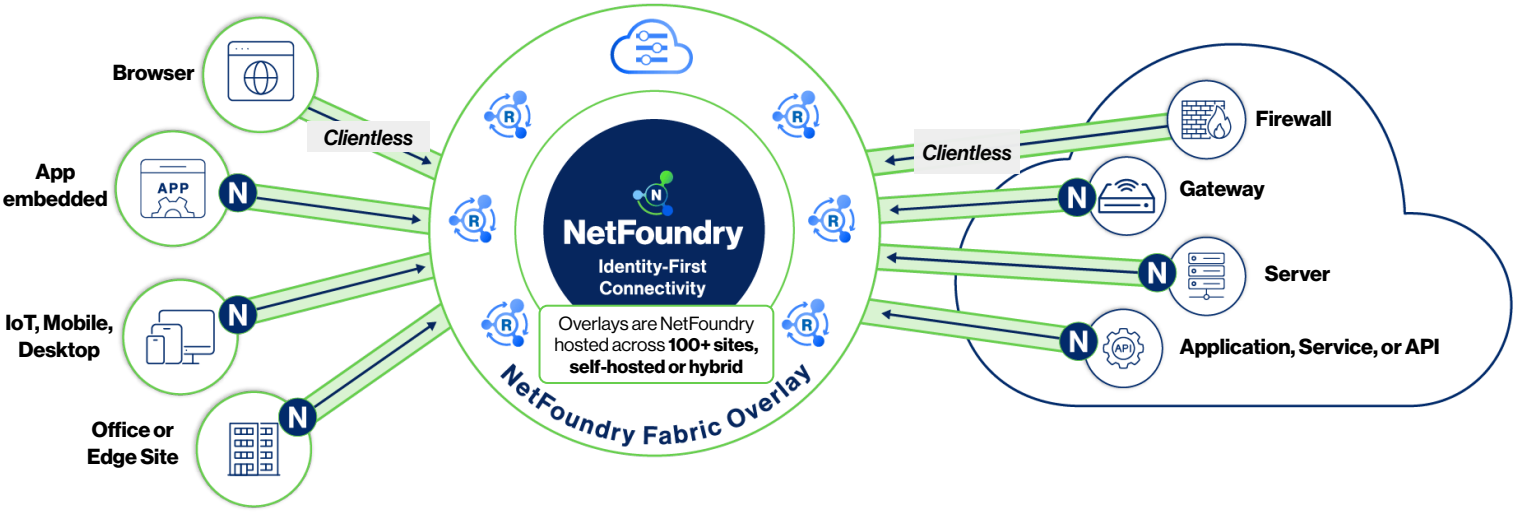
FOR EXAMPLE:

1. Simple to control, view and audit agentless, one-time or just in time (JIT) access.
2. Implement new initiatives, sites or business partners without any dependencies on firewalls, IP addresses, infrastructure, edges or clouds.
3. Easily dial up security strength to fit the needs of each situation, like requiring proof of possession identity because NetFoundry builds-in the PKI solution – adaptive access.
4. Firewalls, WAFs, API gateways and application servers are no longer reachable from unauthorized users. Authorized users – even third parties – are easily 'turned off' if they are breached and only have microsegmented access to start with.
5. Zero days or existing malware can't easily exfiltrate data when the secure network is extended to the application or app server (which are NetFoundry options).

NetFoundry's secure networking model doesn't just radically reduce risk - it simplifies operations and unblocks new initiatives because most of the complexity was due to using infrastructure dependent network security to try to compensate for default insecure, authorization unaware networks. This is why NetFoundry's new secure by design model helps end the tug of war between security and the rest of the business.

APPENDIX

NetFoundry's universal, secure-by-default architecture



- Continually authenticated or JIT access for IT, OT, IoT and AI
- Easily extend anywhere, including unmanaged and 3rd party
- Choose proof of possession, PKI-based identity or IdP enforced

- High-performance, E2EE, identity defined networking on dedicated overlays
- Distributed revers proxy, WAF, firewall, API and AI gateway functionality
- Centralized contols, telemetry, audit with end-to-end control

- Extend secure connection to FW, GW, server or all the way to the application
- Make the firewall, server or application unreachable from unauthorized endpoints
- Can self-host entire network at a site, including air-gapped

Key Components Benefits	
Client-side connectors	Host endpoints include lightweight software for OT, IoT and IT 'client side' devices, available in the Win, Mac, iOS and Android marketplaces.
Server-side connectors	Host-based endpoints are available for the 'server side', so that connections are controlled end-to-end between client and server.
Site-level connectors	Endpoint types also include 'gateways' which front-end edges, site and clouds. This includes images in every major cloud marketplace.
Infrastructure integrated endpoints	NetFoundry endpoints are already built-in or easily integrated with browsers, modems, firewalls, edge servers, reverse proxies and API gateways.
Software integrated endpoints	Use NetFoundry SDKs to add the endpoints into any software – the software then ships with built-in zero trust connections, without requiring the deployment of endpoints or gateways.
Clientless access	NetFoundry provides TLS and mTLS options, without any endpoints. These are often mixed with the options above so that all sites are supported.
Fabric overlay routers	Routers enable endpoints to make outbound-only connections, without firewall hole punching, NAT or static IP address dependencies. The routers provide telemetry and control and are either hosted by NetFoundry or self-hosted. They are single-tenant when hosted by NetFoundry.
Network controllers	Controllers enable the management of identities, services, configurations and policies, including authentication and authorization. Controllers include PKI, as a managed service, and support third-party CAs. Controllers are hosted by NetFoundry or self-hosted (dedicated to each tenant).

This architecture enables businesses to use "zero trust native" overlays in which all devices and services (internal and supply chain) are strongly identified, authenticated and authorized before any communication, and no network ports are left open for reconnaissance, attack, lateral movement or data exfiltration. The overlays are self-hosted or NetFoundry hosted and are based on NetFoundry's industry leading open source software, OpenZiti.

Identity-based authentication & authorization

Every endpoint (user, device, or service) is enrolled via a unique, RSA or EC cryptographic identity (X.509 certificates with optional MFA and IdPs), used for mTLS connectivity. The mTLS private key never leaves the endpoint and is unique to each endpoint (invalid if moved). Hardware roots of trust or TSMs can be used for the local storage of each key (PKCS11). This functions as a Proof-of-Possession (PoP) identity such that credential theft is thwarted – the attacker must take control of the human and associated machine to get any access – not simply use stolen credentials, tokens or assertions from the Internet.

Connections are only allowed after verifying identity and posture for both sides of the connection. It is a continuous authentication model – meaning if there is a change in posture or policy during a session, then that session is terminated. Third-party CA, third-party IdP, external JWT, username, password and TOTP are all supported (all RFC 6238 is supported), as primary or secondary auth methods. The CA/PKI solution is included in NetFoundry NaaS, and other CAs are optionally integrated via RFC 7030 support.

All access is explicitly controlled by policies – the default policy is no access. Endpoints and services only do what they are explicitly permitted to do – "positive security model" instead of "negative security model". This makes it easy for administrators to setup and enforce one-time access, just-in-time (JIT) access and continually authenticated access models. For example, a ticket system or CI/CD workflow can instantly create an identity or policy, and instantly revoke it upon a certain event or timeframe, while continual auth and posture checks mean that state-based events can also instantly terminate the access.

Businesses provision NetFoundry endpoints via standard methods including SCIM and JWT (or build the endpoints into their software so that the network goes anywhere their software goes without agents or gateways).

Because NetFoundry endpoints don't depend on specialized hardware, and can overlay into existing infrastructure, they often retrofit zero trust security into legacy sites, for example front-ending anything from an OT device to an AS400, or simply enforcing posture and MFA for applications which don't support mechanisms like SAML, OAuth or OIDC.

End-to-end encryption & data cloaking

All data is secured with strong encryption (configurable ECDSA or RSA for mTLS 1.2+, ChaChar20-Poly 1305, libsodium). In addition, other ciphers such as FIPS-compliant encryption and post-quantum encryption (PQE) are optionally plugged in.

Application data is end-to-end encrypted between the communicating parties - intermediate routers in the overlay (even if hosted by NetFoundry) cannot read or tamper with the content, and neither can network administrators.

NetFoundry's encapsulation and encryption helps obfuscate metadata – packets look like TLS traffic on port 443 (or whatever port has been configured in NetFoundry), with no useful info in either content or headers for an adversary to exploit. This level of cloaking helps thwart common reconnaissance (port scanning, protocol fingerprinting, traffic analysis) and reduces actionable intelligence that enemies could gather from the network.

'Dark' services, devices or networks

NetFoundry makes services, devices or sites unreachable from unauthorized endpoints:

- When using NetFoundry SDKs, an application, session or service is unreachable from anywhere – it is end-to-end encrypted from the memory space of the application – even hosts don't have access. This is proof-of-possession (PoP), identity-based microsegmentation, at the session level. For example, an AI agent can be scoped to not have any network access, and not to be accessible from the networks, but to only have access to connect to specifically defined microservices, with NetFoundry providing the mechanisms for identity, authentication, authorization, encryption, policy, delivery and telemetry.
- When using NetFoundry host-based endpoints with host firewalls (or NetFoundry's integrated firewalls), the endpoint (e.g. a server) can be made to only open sessions outbound to its private NetFoundry overlay and not accept inbound requests from the Internet or underlay network. This is proof-of-possession (PoP), identity-based microsegmentation. A common use case is to make a server (API, application, AI, MCP or web server) unreachable from the Internet or underlay network.
- When using NetFoundry's edge, site or cloud endpoints, the site devices can be configured to only route outbound sessions to the NetFoundry site endpoint (it functions as the default gateway and network firewall), and to not accept inbound requests from the Internet or underlay network. A common use case is to make an edge site – for example, a site with IoT devices like an energy site or smart building – unreachable from the Internet (instead of relying on a firewall, which is Internet reachable, and therefore extremely vulnerable to attacks). This supports remote access, including one-time, JIT and agentless, as the outbound session from the site devices supports full-duplex connectivity from authorized endpoints.

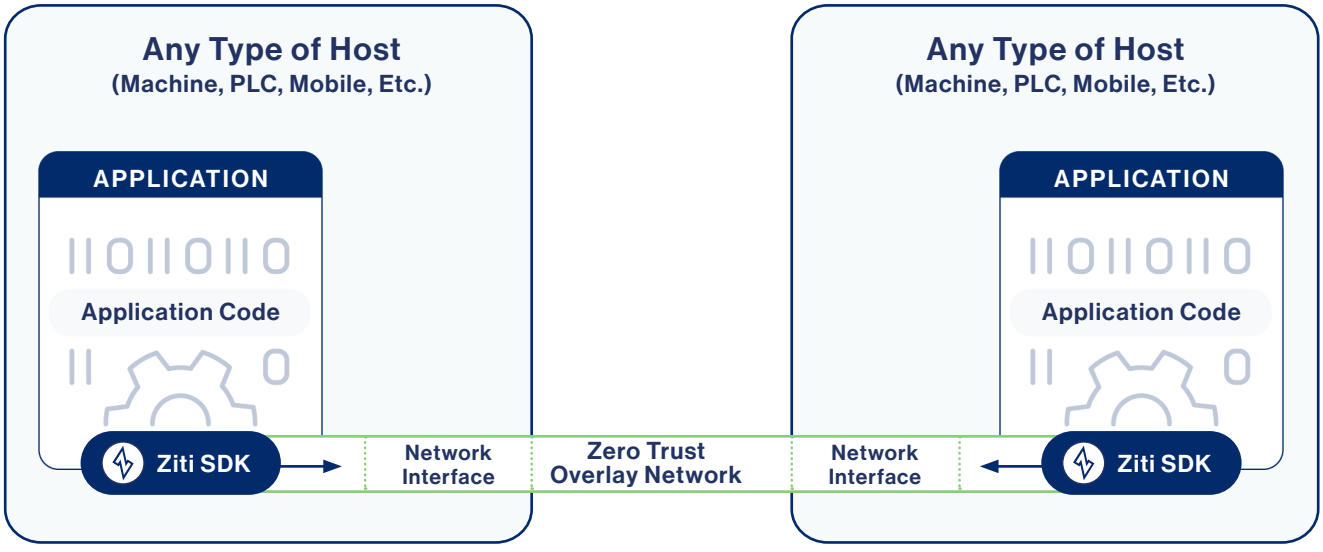
This overall concept is to shrink the attack surface to insider attacks only - even if an attacker is physically on the same network, they cannot access or even enumerate a NetFoundry-protected resource without being enrolled in the system and authorized for that specific session. NetFoundry secured resources will not respond to any unauthorized ping or connection attempt – the service, device or server might as well not exist.

Microsegmentation and least privilege

Through NetFoundry, administrators define fine-grained segments and access policies, with full visibility from telemetry and audit perspectives (NetFoundry helps discover the network flows and define policies for them).

Each application or device is segmented so they can't communicate, even if on the same physical network, unless policy specifically allows. The matching of authorized identities to services helps mitigate against malicious lateral movement and data exfiltration.

Every service can function as a 'network of one', with authorized endpoints on that virtualized network only able to access the specific service. Likewise, a single device may have access to a service but be completely unable to even see a different service, even if they are on the same physical network. This also sandboxes what a compromised node can do. The ultimate microsegmentation is via the NetFoundry Ziti SDKs which embed zero trust overlay network endpoints into any application or software:



This means the zero trust native network overlay goes anywhere the application goes, without agents or gateways. Since NetFoundry also provides the private overlay as NaaS, operations are massively simplified. The SDK option also provides extreme security – even a compromised host has no access to the session.

Adaptive, HA networking with end-to-end control and visibility

NetFoundry NaaS uses full mesh adaptive routing across over 100 data centers on the world's top IP backbones to optimize reliability and performance, with options to limit to certain geographies or providers. Each network is private and dedicated to its administrator, who can then decide to add tenants or clients. Overlays can be self-hosted, including in air-gapped sites. This overlay architecture helps NetFoundry provide end-to-end control and telemetry, especially because it can be used for any use case, including supply chain partners.

Overlay network controllers instruct the endpoints and routers on the best performing route for each session. Endpoints simultaneously use different paths for different sessions (no single tunnel backhaul), changing paths as necessary, and doing load balancing. Routers do not have access to the encrypted payloads. Every link is mutual TLS (mTLS).

NetFoundry endpoints leverage diverse transport networks – if a radio link is slow or down, traffic might reroute over a satellite or wired link. The architecture is designed for high availability; there's no single point of failure, and the system can tolerate outages or route around network damage. This provides resilience in DDIL conditions.

Security policy enforcement is distributed – once endpoints are authenticated and have their session keys, they can often continue communicating through available edge routers even if the central controller is temporarily not reachable.

Both client and server-initiated sessions open outbound towards the overlay. This enables firewalls and servers to deny all inbound. The overlay routers bridge the session.

Universal, secure-by-design networking: summary

NetFoundry is the first to provide a new networking model: secure-by-design networking. The model provides far stronger security, improves reliability and simplifies operations.

As importantly, there are realistic paths towards this north star, with ROI at each step, and businesses choose to use NetFoundry for all use cases or select use cases. NetFoundry helps future proof as well, even including moving off NetFoundry services, if necessary, via its software-only, open source based, API first approach.

The result is virtualized, zero trust native overlays, independent of networking and infrastructure. All sessions (internal and supply chain) are authorized before any communication, and no inbound underlay ports are left open. It includes true end-to-end encryption (the keys remain local to the endpoints), with every link using mTLS. Each overlay delivers authorized, microsegmented sessions, mitigating against data exfiltration. NetFoundry is consumed as cloud native NaaS or self-hosted on-premises, including air gapped. NetFoundry is an all batteries included service, with no external dependencies, but includes prebuilt integrations and APIs for the cases when integrations make sense.