NETFOUNDRY™

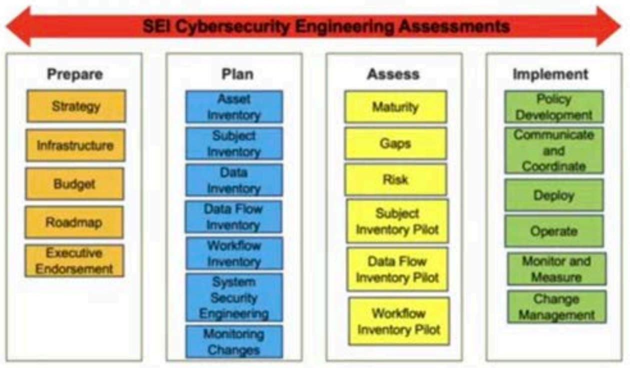
SEI Zero Trust Industry Days 2024

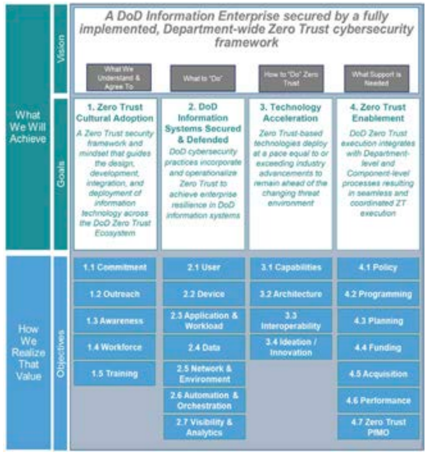
Building **Secluded Semiconductors' Zero Trust** Cybersecurity architecture strategy

- OMB M-22-09, Moving the U.S. Government Toward Zero Trust Cybersecurity Principles[1]
- OMB M-21-31, Improving the Federal Government's Investigative and Remediation Capabilities Related to Cybersecurity Incidents[2]
- CISA Zero Trust Maturity Model, Version 2.0[3]
- National Cybersecurity Strategy[4]
- DoD Zero Trust Strategy[5]
- CISA Zero Trust Implementation Strategy[6]

	Identity	Device	Network / Environment	Application Workload	Data
Traditional	• Password or multifactor authentication (MFA) • Limited risk assessment	• Limited visibility into compliance • Simple inventory	• Large macro-segmentation • Minimal internal or external traffic encryption	• Access based on local authorization • Minimal integration with workflow • Some cloud accessibility	• Not well inventoried • Static control • Unencrypted
	Visibility and Analytics	Automation and Orchestration	Governance		
Advanced	• MFA • Some identity federation with cloud and on-premises systems	• Compliance enforcement employed • Data access depends on device posture on first access	• Defined by ingress/egress micro-perimeters • Basic analytics	• Access based on centralized authentication • Basic integration into application workflow	• Least privilege controls • Data stored in cloud or remote environments are encrypted at rest
	Visibility and Analytics	Automation and Orchestration	Governance		
Optimal	• Continuous validation • Real time machine learning analysis	• Constant device security monitor and validation • Data access depends on real-time risk analytics	• Fully distributed ingress/egress micro-perimeters • Machine learning-based threat protection • All traffic is encrypted	• Access is authorized continuously • Strong integration into application workflow	• Dynamic support • All data is encrypted
	Visibility and Analytics	Automation and Orchestration	Governance		

Software Engineering Institute (SEI) Zero Trust Journey





Building a Zero Trust Cybersecurity architecture strategy

Secluded Semiconductors' existing framework

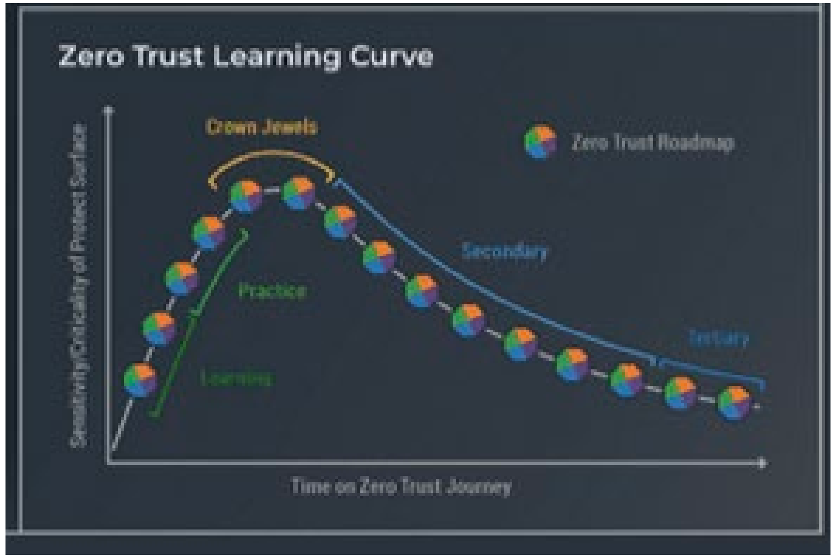
Stage	Prepare	Define	Assess and Design	Implement	Monitor
Steps	• Define business goals and objectives • Document existing systems, architecture and assets • Develop a budget to drive ZT transformation • Outline a HL roadmap	• Inventory of assets, subjects, data, data flows, APIs and workflows within the enterprise • Define the protect surfaces • Identify DAAS elements in each protect surface • Map business goals to DAAS	• Assess current capabilities and define the ZT target state • Define solution criteria mapping to ZT pillars • Pilot for subject, data flow and workflow inventories • Identify and evaluate candidate solutions • Perform cost/benefit analysis • Prioritize initiatives • Finalize roadmap • Aligns initiatives to business goals and protect surface	• Formulate policies for critical DAAS elements • Formulate policies to secure a path to access critical DaaS elements • Deploy people, processes and technology • Operate and maintain the processes, policies, HW/SW of ZT systems • Change management & iterative implementation	• Establish metrics for roadmap tasks (both progress & efficiency metrics) • Track and report metrics • Build communication deck
Mapping to CSA		• Define your protect surface	• Map the transaction flows	• Build a Zero Trust architecture • Create a Zero Trust Policy	• Monitor and maintain the network
Outcomes	• Define business goals • Roadmaps (draft)	• Mapping business goals to protect surface	• Gap analysis of security capabilities • Mapping business goals to protect surface • Evaluation of candidate solutions and roadmap to close gaps	• Method for defining zero trust policies for candidate solutions	Metrics for measuring progress and efficiency of the zero trust implementation

NETFOUNDRY

Building a Zero Trust Cybersecurity architecture strategy

Secluded Semiconductors’ ZT Strategy Principles

- **Journey:** Secluded Semiconductors’ current architect and business requirements mean our transition to ZT will not be quick and easy.
- **Critical Infra:** Our non-IT systems are critical to the safety and security of both employees, residents, and the viability of our business. We must ensure **safety, reliability and uptime, alongside security, of our systems**, even in natural disaster scenarios.
- **Standards:** Alongside ZT guidance OMBs, standards, and guidance, we must always be cognisant of industry standards, such as **62443**. As CISA/NIST guidance is ‘light’ at best on ZT for OT, we will interpret and model back.
- **Learning Curve**. While we have milestones in our ZT plan – e.g., 1 year advanced ZTMM, 2 year optimal – we will utilize a learning curve to ensure programmatic success as well as no negative affect on critical systems.
- **Platforms**. We will identify and utilize technology which allows us to implement zero trust, leveraging a minimum of resources (people, money, and time). This naturally favours **SW-based approaches**. We will also favour **technology platforms** which can support any use case and requirement, as well as move swiftly through levels of maturity.
- **Metrics**: To drive desired behavior and outcomes, we will monitor two key:
    1. Customer Experience
    2. Operational Resilience



NETFOUNDRY

Building a Zero Trust Cybersecurity architecture strategy

Secluded Semiconductors’ ZT Goals

- **Advanced ZTMM 2.0 within 1 year**
- **Optimal ZTMM within 2 years confirmed via assessment**
- **Resilient to identify threats, even in a degraded mode**
- **Policy changes implemented & operational within 30 minutes**
- **All logging and monitoring information obtained via APIs**
- **Integrated security securing all users/all locations consistently with work on the manufacturing process remotely**
- **In the event of a disaster, chip manufacturing and business COOP is successfully operational within 12 hours**
- **ZT applied to chip manufacturing & rest of island’s capabilities**
- **Cybersecurity spending <$3 million over the next two years**

	Identity	Device	Network / Environment	Application Workload	Data
Traditional	• Password or multifactor authentication (MFA) • Limited risk assessment	• Limited visibility into compliance • Simple inventory	• Large macro-segmentation • Minimal internal or external traffic encryption	• Access based on local authorization • Minimal integration with workflow • Some cloud accessibility	• Not well inventoried • Static control • Unencrypted
	Visibility and Analytics	*Automation and Orchestration*		*Governance*	
Advanced	• MFA • Some identity federation with cloud and on-premises systems	• Compliance enforcement employed • Data access depends on device posture on first access	• Defined by ingress/egress micro-perimeters • Basic analytics	• Access based on centralized authentication • Basic integration into application workflow	• Least privilege controls • Data stored in cloud or remote environments are encrypted at rest
	Visibility and Analytics	*Automation and Orchestration*		*Governance*	
Optimal	• Continuous validation • Real time machine learning analysis	• Constant device security monitor and validation • Data access depends on real-time risk analytics	• Fully distributed ingress/egress micro-perimeters • Machine learning-based threat protection • All traffic is encrypted	• Access is authorized continuously • Strong integration into application workflow	• Dynamic support • All data is encrypted
	Visibility and Analytics	*Automation and Orchestration*		*Governance*	

NETFOUNDRY

Building a Zero Trust Roadmap

ZT Implementation Plan – Key Technologies

- **ICAM:** We need to extend or replace our centralized ICAM solution (minimum 'hot standby') to have operational support on the island as well as to provide human and machine identity to systems. It needs to provide standards-based APIs to support enforcement of policy decisions across applications and services. For example, Fornetix. This ensures operations in disaster scenarios as well as being able to apply ZT and ABAC to non-IT systems (IoT, OT, city etc).

- **Zero Trust Network**: We need to implement a zero trust network overlay that can support any use case (incl. N-S, E-W, OT/IoT, M2M etc; while aligned to 62443/Purdue etc), can operate from the Island (minimum HA control plane), support legacy deployment models, ephemeral delivery aligning to ICAM updates and business rules, as well as support our initial discovery / transaction flow mapping. The deny-by-default approach ensures we prevent network attacks from happening, instead of detecting and responding. We expect to utilise NetFoundry, building on the OpenZiti project. The open source angle is crucial to working with industry partners who can build it into their products which we use in our factories, grid, smart city etc as well as other open source projects (e.g., EdgeX).

- **Cloud / Hypervisors**: We need to consider compute protection and hypervisor services which provide secure, isolated virtual environments with confidential computing and autonomous threat protection built in for our applications and services. The deny-by-default approach ensures we prevent compute/application attacks from happening, instead of detecting and responding. An example of this capability is Mainsail's Metalvisor, a type-zero hypervisor that provides hardware-based isolation for workloads.

NETFOUNDRY

Building a Zero Trust Roadmap

Mapping Key technologies to DoD 7 Pillars of Zero Trust

- 3 technologies provide a robust foundation and lays the groundwork for to fully meet all 152 Zero Trust activities
- Need to bring in SIEM/SOAR, EDR, data mngt and possibly more

NETFOUNDRY

Building a Zero Trust Roadmap

Some Design Principles – Manufacturing and Legacy

Topic	Concern	Solution
Resiliency / Loss of connectivity	Operating environment mandates our systems can operate disconnected and local without connection to mainland and ensure continuity of operations if power flips on/off.	• Control & data planes must be hosted on island. Administration plane & 'bridge' to mainland should be ok in cloud. Both control and dataplanes have HA for no SPOF. • Data planes should be ephemeral & degrade gracefully • ICAM needs backup/ability to operate on Island
Safety and Reliability	Manufacturing environments require safety and reliability 1st, with with alignment to 62443	• Ensure ZT solutions have no single point of failure/HA & scalable • Purdue (outbound connections from higher trust to lower trust environment) & OT (e.g., L2) compliant & work with 62443 cell structure • Ability to intercept packets, drive ephemeral connections

ZT & OT/Legacy	3 fabrication systems as well as IoT/Smart City need to be able to support ZT architecture where it makes sense	• Acceptance that not all ZTMM can be applied to OT use cases
		• Pick solutions which can:
		  • Support M2M, ~80% of traffic in manufacturing environment
		  • Machine/its own identity (password-less; support legacy apps which cannot do SAML/SCIM), zero touch deployments
		  • Support 'ZT' light with NAC type capabilities
		  • Interoperability of ZT overlay with underlay monitoring (MOSAICS & ElaZtic)
		  • HBZST for high value endpoints which cannot deploy SW
		  • Technology designed for edge & constrained, e.g., SDKs, lightweight
		• **Work with existing vendors to ascertain their ZT journey, product capabilities (SOTA), and ability to embed native ZT with OSS (see example, IPCs, IFWs, PLCs, etc).**

Accessibility and availability	In a disaster scenario our engineers still need to be able to access the production site, even if not able to be in the factories	• Minimum OOB access which does not depend on ICAM but still uses strong identity • Control & data planes must be hosted on island.

NETFOUNDRY

Building a Zero Trust Roadmap

Some Design Principles – Smart City and IoT

Topic	Concern	Solution
Safety and Reliability	Highly connected systems which can cause downtime to critical functions if compromised (e.g., water or grid). Safety and reliability are primary concerns.	• Assess and prioritise use cases which are most foundational to saving and maintaining lives in a disaster scenario. This includes using ZT learning curve approach as we iteratively apply ZT to use cases where high security is important and it can be achieved. • Ensure ZT solutions have no single point of failure/HA & scalable
Risk reduction	IoT/Smart Cities are highly connected systems which could have vulnerabilities	• Mandate our vendors are compliant to the UK Product Security and Telecommunications Act (PSTI) to deliver critical measures to safeguard connectable consumer products against cyber threats. Strongly encourage PSTI compliance for residents. This mandates strong/unique passwords, security issue reporting, and security updates (secure-by-design)

ZT & Smart City	IoT/Smart City need to be able to support ZT architecture where it makes sense	• Acceptance that not all ZTMM can be applied to Smart City use cases – systems may not be able to even support encryption
		• Pick solutions which can:
		  • Technology designed for edge & constrained, e.g., SDKs, lightweight.
		  • Embedded identity, zero touch, secure device onboarding (e.g., Dell NativeEdge)
		  • Utilise LPA and deny-by-default to connect disparate systems without explicit trust
		  • Support 'ZT' light & HBZST for high value endpoints which cannot deploy SW
		  • App-embedded ZT (where possible) to ensure IoT/smart apps do not 'listen' to the network even if degraded
		• **Work with existing vendors to ascertain their ZT journey, product capabilities (SOTA), and ability to embed native ZT with OSS (see example automation and control systems for generator and transmission).**

NETFOUNDRY

Building a Zero Trust Roadmap

Some Design Principles – Connected Services

Topic	Concern	Solution
Resiliency / Loss of connectivity	Operating environment mandates our systems can operate disconnected and local without connection to mainland and ensure continuity of operations.	• ICAM needs backup/ability to operate on Island. This ensures that if internet and cloud are unavailable, our systems continue to have access. This includes redundant/HA capabilities.
Accessibility and availability	In a disaster scenario our engineers still need to be able to access the production site, even if not able to be in the factories	• Minimum OOB access which does not depend on ICAM but still uses strong identity • Control & data planes must be hosted on island.

NETFOUNDRY

Building a Zero Trust Roadmap

Near-Term Planning (1-2 years); IT & ZTMM

Year	1				2			
Quarter	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
Program	Goals, Budget Draft Roadmaps	Ongoing Governance, Stakeholder Mngt, Communications, Reporting Metrics						Optimal ZTMM Assessment
Discovery	Identify Protect Surface & Critical apps Initial Inventory Map Transaction Flows Roadmap update	Continued discovery and inventory; mapping transaction flows Update and implement ZT policies Roadmap updates & Monitoring Metrics						
Identity	Enable MFA for supported apps Audit Permission	Automated joiner/leaver			ICAM & ZTN integration for automated policy updates	ICAM/ZTN integration with digital workflow		
Devices	Implement endpoint security controls, baselines, posture			Automated Patching/vuln mngt	Real-time analytics & ZTN integration			
Workloads	Compliance controls cloud	Implement ZTA to apps Encryption to accounts/APIs		ICAM for apps		Immutable workloads and security testing		
Networks	ZTN macro & discovery Internet GW	ZTN micro segmented N-S & dark	Private DNS	ZTN micro segmented E-W	JIT/JEA ephemeral business rules All traffic encrypted & ML TD	OT packet capture Quantum Encryption		
Data		Ensure data encryption at rest (cloud and on-prem)		Implement least privilege	Continuous Data inventory	Data categoriz. & labelling	DLP blocking & dynamic access	
Visibility			Implement SOC/SIEM/SOAR					

Building a Zero Trust Roadmap

Near-Term Planning (3-5 years); OT, IIoT, Smart City

Year	3				4				5			
Quarter	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
Program	Ongoing Governance, Stakeholder Mngt, Communications, Reporting Metrics											
Discovery	Continued discovery and inventory; mapping transaction flows Update and implement ZT policies Roadmap updates & Monitoring Metrics											
Identity	• Assessing and progressively implementing ZTMM Optimal for OT, IIoT, Smart City Use Cases • Roadmap TBD once we have assessed further what is and isn't possible • Replacement cycles for HW/SW devices compliant to PSTI • Work with existing vendors to ascertain their ZT journey, product capabilities (SOTA), and ability to embed native ZT											
Devices												
Workloads												
Networks												
Data												
Visibility												

NETFOUNDRY

Making zero trust successful

Organisation's training needs

- **Executives and Managers**: Require training on the business implications and benefits of Zero Trust security (see CSA, Jason's working group/papers). Need to understand the risks associated, potential impact operations, and bottom line, as well as upsides.
- **IT Personnel**: Require training on Zero Trust principles, concepts, and best practices (again, CSA has some great resources) with ongoing training and certifications. Also on how technology/process changes impact their area (e.g., new tools, protocols, etc).
- **OT and IIoT Specialists**: Training on securing their systems can operate in a Zero Trust environment, mapping more tightly to their standards (e.g., 62443) rather than the 6 documents.
- **Manufacturing Personnel**: Need basic cybersecurity training to understand their role in maintaining a secure manufacturing environment. Incl. recognizing and reporting security incidents, adhering to security protocols, and understanding the importance of data protection in manufacturing operations.
- **Clerical and Logistic Support Personnel**: Training on basic cybersecurity hygiene practices, such as password management, phishing awareness, and data handling procedures. Incl. raising awareness about common cyber threats and educating employees on their role in maintaining a secure work environment.
- **City Services Personnel**: May require cybersecurity training tailored to their specific roles and responsibilities, incl. guidance on identifying and mitigating cyber risks in their respective areas of operation.

Making zero trust successful

Projected costs and budget

- **Budget:** Cybersecurity spending <$3 million over the next two years is low, particularly for a high security, distributed business such as Secluded Semiconductions. Deloitte research shows 1k employees ranged from $1.5 million to $3.5 million in 2020, let alone 2024-26, as well as implementing the zero trust programme. We need to decide how much of that we allocate to the programme.

- **Implementation Costs:**
  1. Initial Assessment and Planning: Approximately 5-10% of the budget allocated for assessments, gap analysis, and developing ZT implementation plan.
  2. Infrastructure Investments: 30-40% dedicated to acquiring and deploying hardware and software components.
  3. Training and Education: Approximately 5-10% allocated for cybersecurity training programs, workshops, and certifications.
  4. Implementation and Integration: Another 20-30% of the budget may be reserved for hiring external consultants or dedicating internal resources to implement and integrate Zero Trust solutions into the organization's infrastructure.
  5. Licensing and Subscription Fees: Ongoing costs for software licenses, subscriptions, and maintenance agreements may consume 10-20% of budget annually.

- **Potential Cost Savings:**
  - Reduced Risk of Breaches and Attacks: Average cost of a data breach $4.24 million (IBM 2021). Implementing ZT reduces likelihood, potentially saving $Ms.
  - Operational Efficiency Gains: Possibly 10-20% reduction in operational costs related to cybersecurity management from automation.
  - Improved Resource Utilization: Savings of 5-15% could be achievable from optimal resource allocation and reducing over-provisioning of IT resources.
  - Enhanced Incident Response and Recovery: A 20-30% reduction in incident response costs, including forensics, legal fees, and downtime, could be anticipated.
  - Compliance and Regulatory Costs: Reduce costs from regulatory fines, penalties, and audits

NETFOUNDRY

Making zero trust successful

Effects on users

- **Design Goals**:
  - **Least-burden**: While changes will be needed to processes, workflows, and user experiences, our long term goal is to have little to no change/remove burdens from users. This is why 'deny by default', 'app-embedded', etc are crucial. We can achieve our technology goals (e.g., MFA, not P/U) without necessarily demanding user TOTP MFA (i.e., strong identity). Users should not be blamed, our end goal is a system which individual user mistakes cannot cause systematic attacks, disruptions, and degradation - i.e., secure by default with asymmetry in our favour.
  - **Risk appropriate**: Additional security checks will be layer on for critical applications which require higher security by default.

- **Authentication, Authorization & Workflows:** Users may experience changes in how they log in to various systems and applications (e.g., MFA, OTP, biometrics) but ideally this will be temporary in most cases. Access to resources will follow least privilege with access to specific resources and data necessary to perform their job functions, ultimately automatically through the ICAM. Accessing sensitive or high-risk resources may require re-authenticate or reauthorize or approval process, again, with the goal to automate through digital workflows.

- **Logging and Monitoring:** Users will be educated through training on logging and monitoring we do, as well as alerts or notifications they would receive if their actions trigger security policies or if suspicious activity is detected.

- **Training and Awareness:**
  1. Users will need to undergo training and awareness programs to familiarize themselves with the new authentication methods, access workflows, and security policies associated with Zero Trust.
  2. Training will emphasize the importance of adhering to security protocols, recognizing potential security threats, and reporting any suspicious activity or security incidents promptly.
  3. Users should be educated about the rationale behind Zero Trust principles and the role they play in maintaining the organization's cybersecurity posture.

NETFOUNDRY

NETFOUNDRY™

Tech Deep Dive

NETFOUNDRY

Tip of the iceberg

What's the commonality?

CHANGE
HEALTHCARE

Feb '24. **Lack of multifactor authentication**.

Paid a $22 million ransom. Expected costs around **$1.6 billion through 2024** (excludes litigation or potential regulatory fines).

NetFoundry makes this exploit impossible.

EQUIFAX

March '17. **Failed to patch a basic vulnerability.**

Cost more than than $1.7 billion. Had a $125 million cybersecurity insurance coverage which paid out maximum reimbursement.

NetFoundry makes this exploit impossible.

Colonial Pipeline Company

May '21. **Exposed VPN password**.

Paid a **$4.4 million ransom**.

NetFoundry makes this exploit impossible.

Progress
MOVEit

May '23. **Zero-day vulnerability allowed SQL injection**.

Total costs of USD 20 million for Q3 2023. Using average cost of customer PII involved in a data breach, incident could have a total cost of **up to USD 12.15 billion**.

NetFoundry means this exploit cannot be exploited.

CYBERSECURITY & INFRASTRUCTURE SECURITY AGENCY
CISA

Jan '24. Remote code exploitation CVE affecting **16,500 Ivanti gateways**.

CISA was exploited.

NetFoundry makes this exploit impossible.

JBS

March '21. **Leaked employee credentials**.

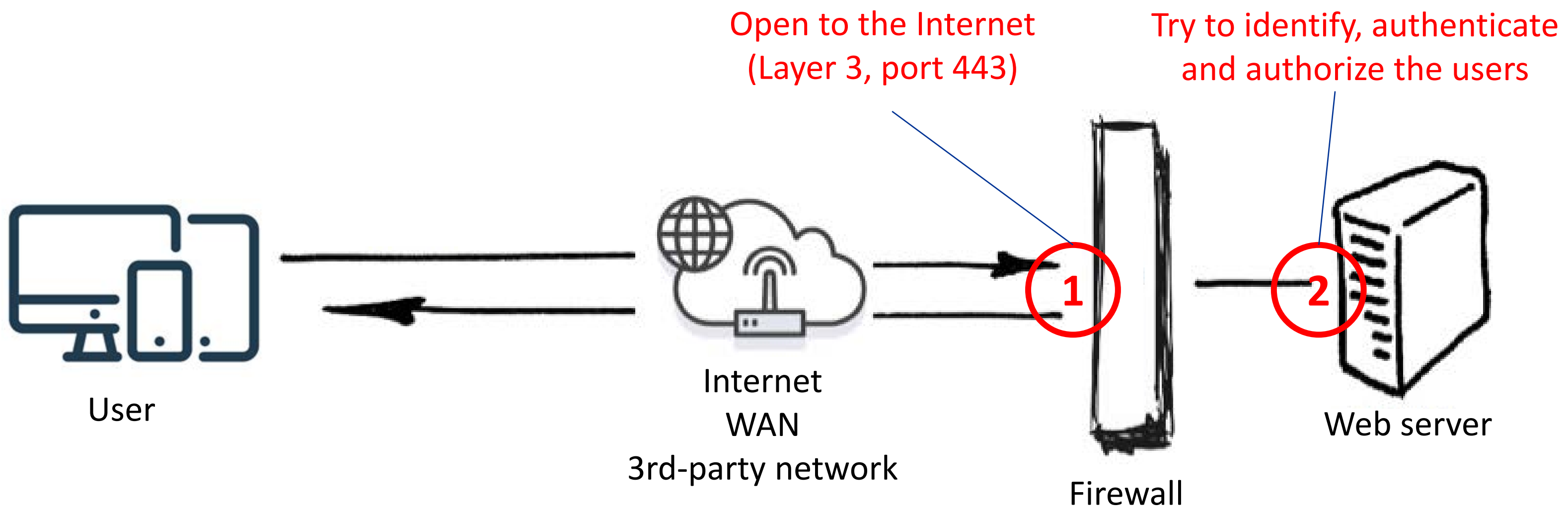
Paid a $11 million ransom and temporarily shut down some operations.

NetFoundry makes this exploit impossible.

NETFOUNDRY

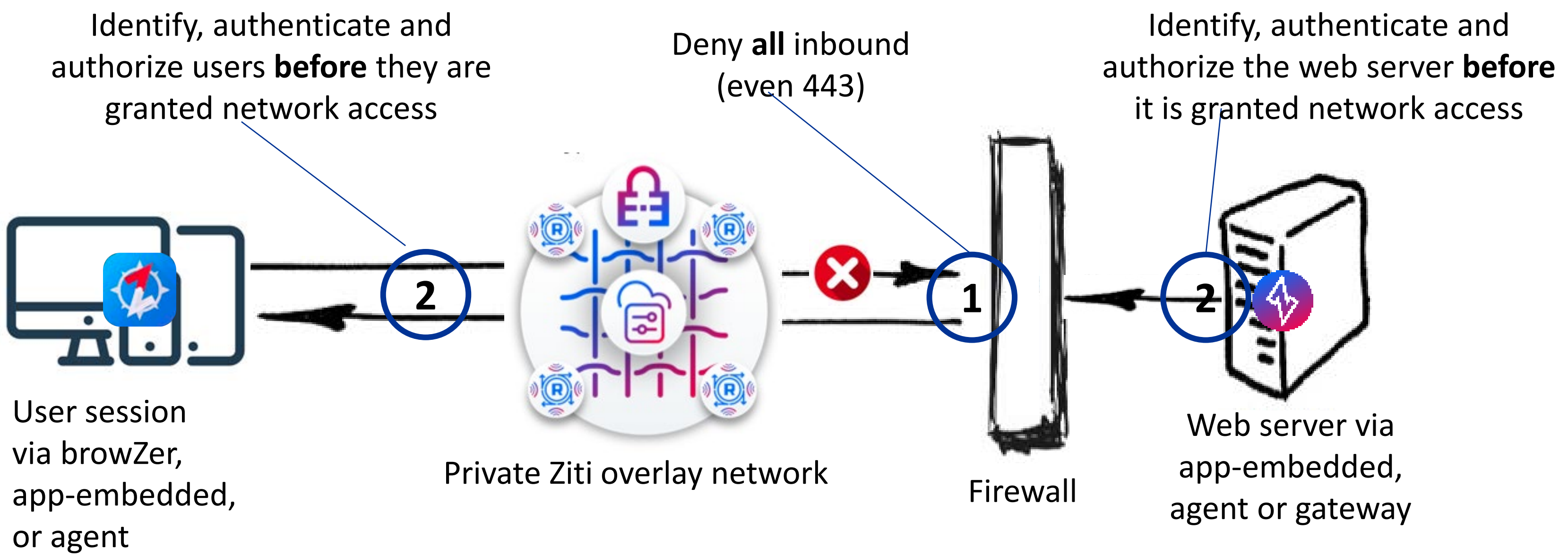
Our network architecture doesn't stop attacks

We 'listen' on the network interface



The NetFoundry architecture makes these attacks impossible

We do not need to listen on network interfaces – WAN, LAN, or host OS network



Gartner – "Zero Trust Network Access (ZTNA) will replace 60% of VPNs by 2023". Gartner recognises NetFoundry as a ZTNA provider as well as giving 'Enhanced Internet' instead of using MPLS

The details

Comparing traditional networking with NetFoundry Cloud

Current networking architecture	Secure-by-Design architecture
Networks are prebuilt and open (scan and exploit)	Ephemeral, app-specific overlay connections are spun up, on demand (minimize attack surface and blast radius).
Any endpoint can initiate a connection with network identities (scan)	Endpoints require a strong cryptographic identity to initiate a connection (minimize attack surface)
Authentication is attempted *after* the connection is permitted, deep inside the network (exploit)	Authentication is required *before* a connection is permitted (minimize attack surface)
Most network elements permit inbound connections (scan and exploit)	All connections are outbound from higher to lower trust zones. All inbound connections are prevented (minimize attack surface)
Endpoints are given subnet or network level authorization (exploit)	Endpoints are governed by least privileged access with ability to microsegment (minimize attack surface and blast radius)
Networks are bolt-on, after the fact, often with physical and virtual appliances (VPNs, FWs, load balancers etc.)	Zero trust network can be built into the solution or application during development, using code and APIs

NETFOUNDRY

The details

No listing ports?

Philip Griffiths • You
Open source zero trust networking
2w •

When I say on socials that app embedded zero trust has no listening ports on the network so is literally unattackable via conventional IP-based tooling, people often respond with some variation of:
- "That would help with open ports, but it also complicates listeners and introduces new attack vectors", "they don't understand (the zero trust people) almost every thing you add, adds to your attack surface", or "Any app or software you add, increases attack surface. It's that simple"
- Another is "If I gain access to a host that has your ZTNA on it, I can now touch everything it has access to touch. That is an increased attack surface. This is called priv esc and lateral movement. Its literally no different than if i gained access to a host thats connected to a corp VPN, i can now traverse that VPN tunnel as long as its up.
- Yet another is: "Once that machine is known, and authorized, thats it, its on. If I exploit a host that has an IP4 address from its hardware NIC and it has a ziti address, i can slide over Ziti, because the PKI is already authorizing that HOST."

All of the above is not true. Here is a great blog from a colleague which describes in greater depth, what 'no listening ports' means.

No Listening Ports?
blog.openziti.io • 7 min read

Michael T. (He/Him) • 2nd
Why not? - FIP, CIPP/C/E, CIPT/M
2d •••

Seems to me this is moving the goal posts and now Ziti does the listening for all services. I am not seeing the appeal. **Sandor Slijderink** thoughts?

Like | Reply · 2 Replies

Load previous replies

Philip Griffiths • You
Open source zero trust networking
2d •••

"You're are just moving the listening port from the service/ edge to the OpenZiti fabric (control/data plane)"... thus the question becomes, how can I compromise OpenZiti? ... you need to do all of the following:
- bypass (or have an exploit for) the mTLS requirement necessary to connect to the data plane (all parts of the overlay are exclusively mTLS)
- have a strong identity that authorizes them to connect to the remote service in question (or bypass the authentication layer the controller provides through exploits)
- know the remote service name, allowing the data to target the correct service (Ziti has a private DNS which does not need to comply with TLDs)
- bypass whatever "application layer" security is also applied at the service (ssh, https, whatever)
- know how to negotiate the end-to-end encrypted tunnel to the 'far' identity

So you have moved the port while reducing the attack surface. Also, you no longer know which Ziti controller/fabric relates to which customer/apps/environment, so that's further obfuscation.

One other point, we haven't even touched on app-embedded ZTN... with this model your app has no listening ports on the underlay network. It's literally unattackable via conventional IP-based tooling.

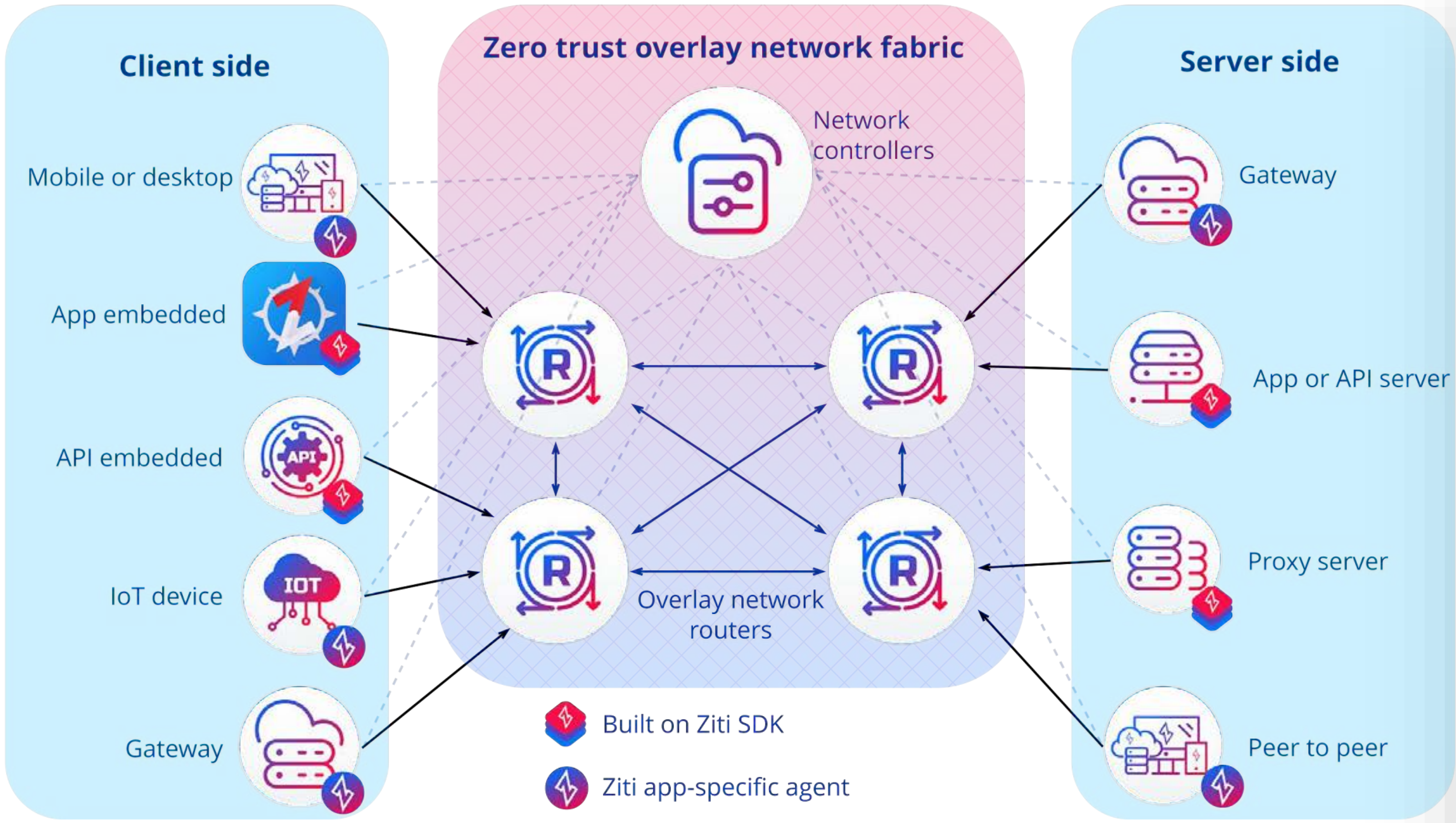
Like · 1 | Reply

NETFOUNDRY

https://blog.openziti.io/no-listening-ports

Key Technology Deep dive - NetFoundry

Zero trust networking from IoT to APIs, and all other use cases, as software

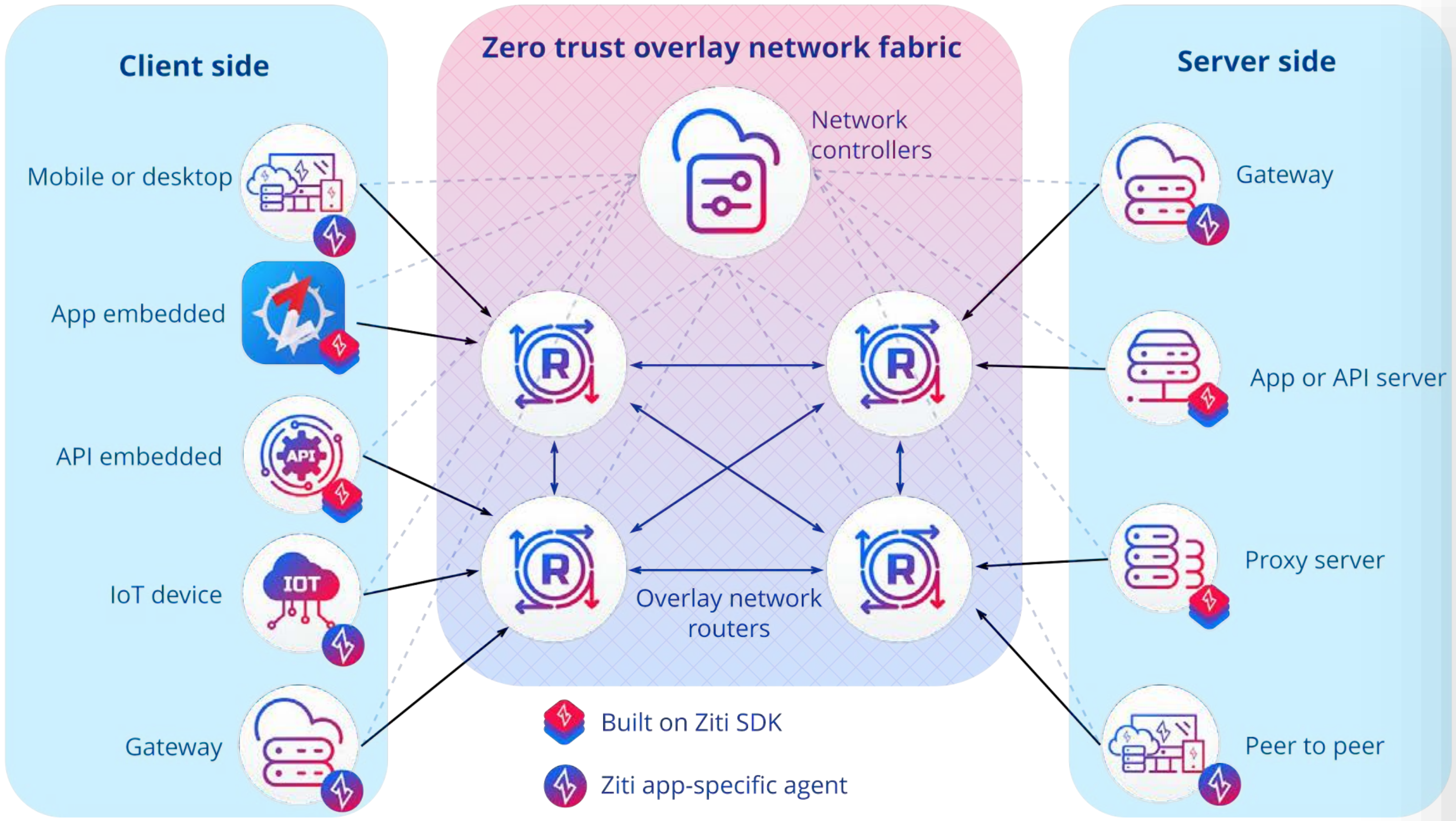


- Ziti provides networking **and** zero trust functions.
- Distribute endpoints with built-in IDs and auth **anywhere** - even to unmanaged devices.
- Authenticate & authorize **before** granting access to private mTLS overlays (least privilege, ephemeral) – i.e., the PEP.
- Authorized servers open **outbound** sockets to the overlay - enabling you to close all inbound ports. Always from trusted to untrusted (aligning to Purdue).
- Ziti routers provides 'SD-WAN' like functions on each full mesh, multipoint overlay. **No backhaul.**

NETFOUNDRY

Key Technology Deep dive - NetFoundry

Zero trust networking from IoT to APIs, and all other use cases, as software



Secluded Semiconductors Requirements:

- Support any use case (incl. N-S, E-W, OT/IoT, M2M etc) as well as legacy deployment
- Purdue compliant (always outbound, from high to low trust)
- Operate anywhere, with no SPOF, incl. air gap. Completely DDIL compliant, with for full authentication, policy config, enrolment etc
- Ephemeral overlay, with strong identity, with 3rd party IdP/ICAM
- Support our initial discovery / transaction flow mapping
- Deny-by-default, zero trust model

NETFOUNDRY

Key Technology Deep dive - NetFoundry

Service discovery for mapping data and transaction flow

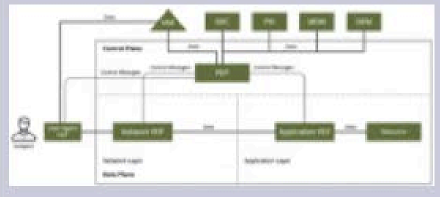
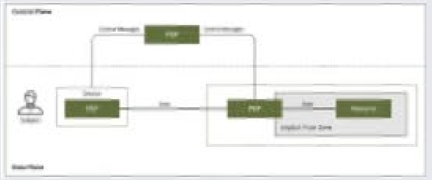
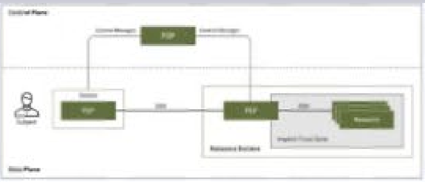
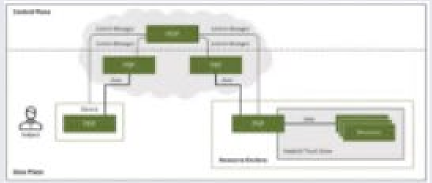
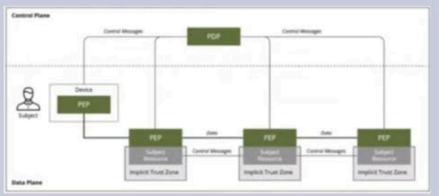
ServiceDiscovery

Top values of nf_service_name.keyword	Top values of path.terminator_remote_addr.keyword	Top values of nf_endpoint_name.keyword	Count of records
AzuNorth-DC01-10.101.5.4	10.101.5.4:53	DXB-ER01	13,125
AzuNorth-DC01-10.101.5.4	10.101.5.4:53	AUH-ER01-NEW	7,524
AzuNorth-DC01-10.101.5.4	10.101.5.4:53	PRT-ER	4,997
AzuNorth-DC01-10.101.5.4	10.101.5.4:53	BAH-ER01	583
AzuNorth-DC01-10.101.5.4	10.101.5.4:53	OMN-ER01	254
AzuNorth-DC01-10.101.5.4	10.101.5.4:88	DXB-ER01	729
AzuNorth-DC01-10.101.5.4	10.101.5.4:88	AUH-ER01-NEW	100
AzuNorth-DC01-10.101.5.4	10.101.5.4:88	PRT-ER	36
AzuNorth-DC01-10.101.5.4	10.101.5.4:88	BAH-ER01	9
AzuNorth-DC01-10.101.5.4	10.101.5.4:389	DXB-ER01	424
AzuNorth-DC01-10.101.5.4	10.101.5.4:389	AUH-ER01-NEW	111
AzuNorth-DC01-10.101.5.4	10.101.5.4:389	PRT-ER	81
AzuNorth-DC01-10.101.5.4	10.101.5.4:389	BAH-ER01	6
AzuNorth-DC01-10.101.5.4	10.101.5.4:389	shellye-v7	1
AzuNorth-DC01-10.101.5.4	10.101.5.4:445	DXB-ER01	213
AzuNorth-DC01-10.101.5.4	10.101.5.4:445	AUH-ER01-NEW	87
AzuNorth-DC01-10.101.5.4	10.101.5.4:445	PRT-ER	31
AzuNorth-DC01-10.101.5.4	10.101.5.4:445	shellye-v7	1
AzuNorth-DC01-10.101.5.4	10.101.5.4:135	DXB-ER01	183
AzuNorth-DC01-10.101.5.4	10.101.5.4:135	AUH-ER01-NEW	70

NETFOUNDRY

Key Technology Deep dive - NetFoundry

Mapping to NIST Deployment Models

Deployment Model	Diagram	Notes
Reference PEP Types		• NF provides PEPs on device*, network, and apps* • AuthN/AuthZ before connectivity is allowed to network PEP, using crypto, outbound-only from low to high risk, deny-by-default model
Resource-Based		• NF supports resource-based deployment with endpoints for apps, hosts, devices, and more • PEP can be hosted locally or externally (for 0 implicit trust in WAN)
Enclave-Based		• NF supports enclave-based deployment with endpoints for apps, hosts, devices, and more • PEP can be hosted locally or externally (for 0 implicit trust in WAN)
Cloud-Routed		• NF supports Cloud-Routed deployment, whether hosted in public or private clouds for 0 implicit trust in WAN
Micro segmentation		• NF supports host micro segmentation for 0 implicit trust in LAN • NF supports app micro segmentation for 0 implicit trust in host OS network; unattackable via conventional IP-based tooling. • Achieved with external network PEPs (no implicit network trust) & no need for external WAN products (VPNs, MPLS, bastions, etc)

Key Technology Deep dive - NetFoundry

App embedded: Zitifications

Before Ziti

```
AsynchronousServerSocketChannel server = AsynchronousServerSocketChannel.open();
server.bind(new InetSocketAddress(InetAddress.getLocalHost(), 8080));

while (true) {
  AsynchronousSocketChannel client = server.accept().get();
  processClient(client);
}
```

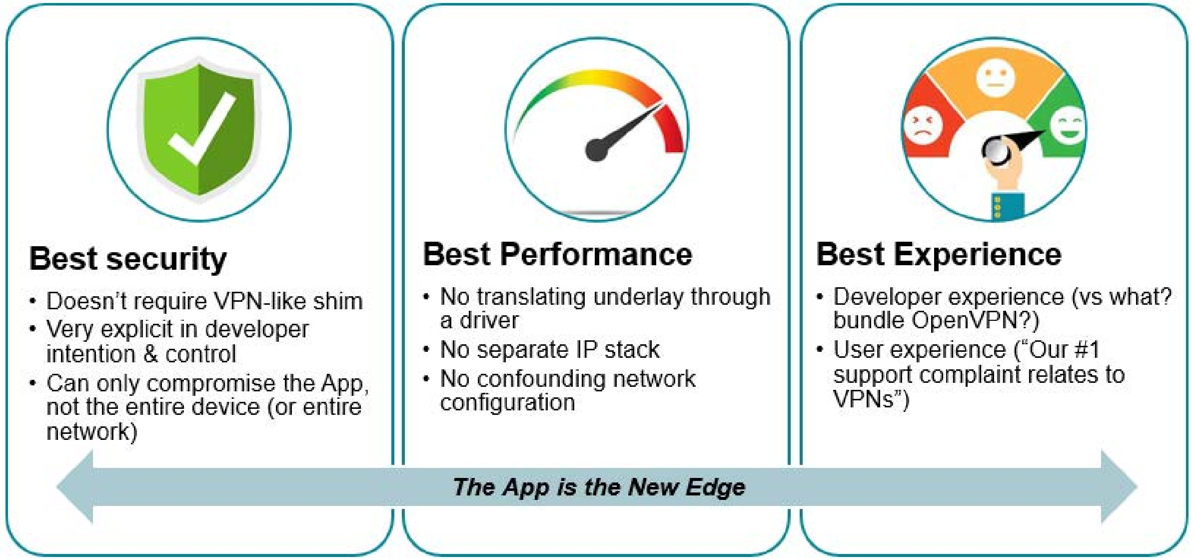
Apres Ziti

```
AsynchronousServerSocketChannel server = ziti.openServer();
server.bind(new ZitiAddress.Service("super-service"));

while(true) {
  AsynchronousSocketChannel client = server.accept().get();
  processClient(client);
}
```

Embedding Ziti JVM SDK

- **Ziti SDKs help enable Zero Trust**
- **Zero Trust requires:**
  - **End-to-end identification**
  - **End-to-end authorization**
  - **End-to-end encryption**
- **End-to-end means embedded in customers' client and server applications**
- **Embedded means SDKs**



NETFOUNDRY

Key Technology Deep dive - NetFoundry

Zitifications

- ZSSH
- ZSCP
- Mattermost
- Webhooks Github/Gitlab
- Generified JDBC Wrapper - ZDBC
- Kubeztl
- Helmz
- Prometheuz
- Ansible
- SPIFFE Integration
- “Zitify”
- Caddy
- Beats & Logtash (Elastic)

Blog:
https://openziti.io/zitifying-ssh

Uses:
Golang SDK

By:
Jon Kochanik

GitHub:
openziti-test-kitchen/zssh/tree/main/zssh

- Augments ssh/sshd. Replaces local ssh client app
- Covers basic functionality not advanced usage
- Features Use of Addressable Terminators

```
zssh ziti-identity-name
```

NETFOUNDRY

Key Technology Deep dive - NetFoundry

Stopping all external network threats

OPENZITI TECH BLOG

Follow

```
        }
    }
}
```

Not much difference, is there? So what changed?

1. We needed to create an OpenZiti `Context`. We need to authenticate and authorize the process with the OpenZiti overlay. Only identities **authorized** to host a service are allowed to do so. Contrast that to the IP-based underlay. With IP-based servers, **any** process can be a listening server. This allows for attacks like DNS poisoning. With a zero trust overlay, this whole style of attack is impossible.
2. The OpenZiti context contains a function to return a `Listener` interface. When using `ListenAndServe`, the standard library assumes the server should be listening on the common, IP-based underlay network because that's how it's always been done, but this is not the case with OpenZiti. Now, your server has **no listening ports** on the underlay network. It's literally **unattackable** via conventional IP-based tooling. Seriously, stop and consider that for just a moment. By adopting an OpenZiti SDK into the server, **all conventional network threats** are immediately useless.
3. The same amazing Go standard library then uses the `Listener` and starts listening for connections from clients on the overlay network. Now, all clients need to be authenticated and authorized to the OpenZiti overlay to connect to the server. If the server itself also requires authentication (a password, for example), it no longer accepts connections from **unauthenticated** and **unauthorized** clients, … ss threatening.

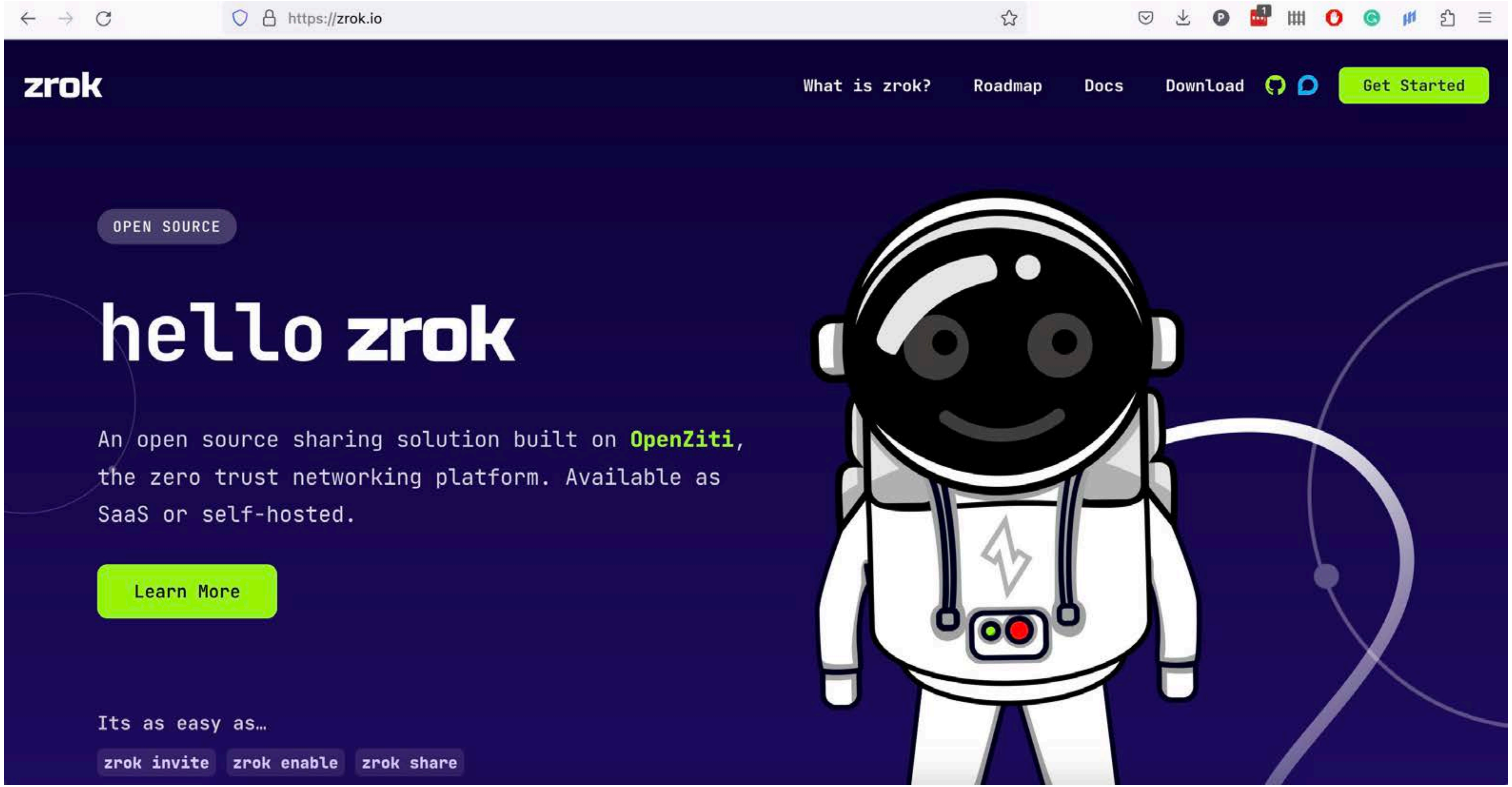
3 1

NETFOUNDRY

https://blog.openziti.io/go-is-amazing-for-zero-trust

Key Technology Deep dive - NetFoundry

zrok: Discreet, Secure by Default Apps, faster…



NETFOUNDRY

Key Technology Deep dive - NetFoundry

CISA Zero Trust Maturity Journey Mapping

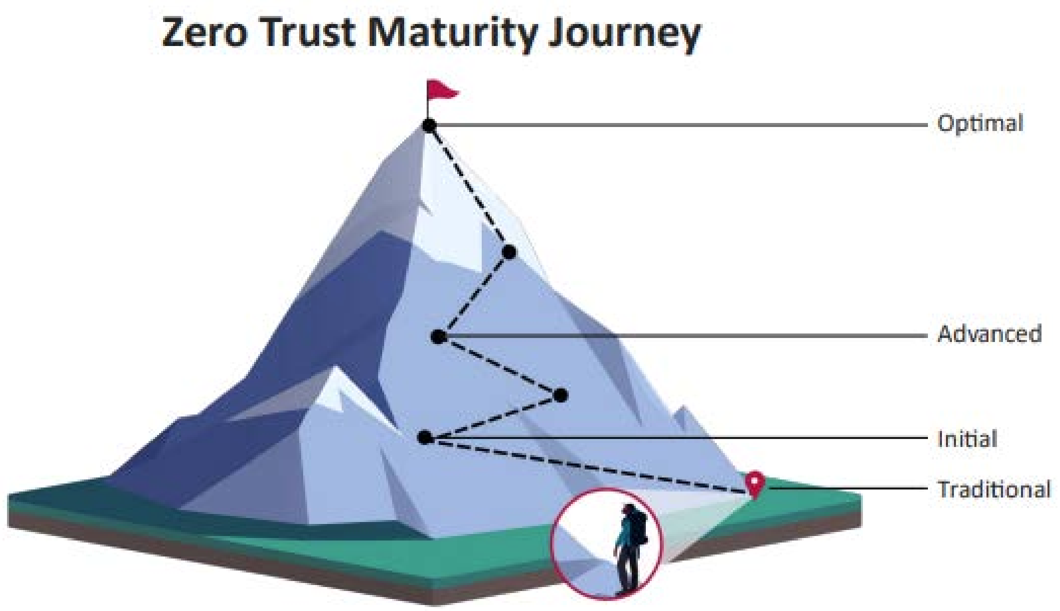


Figure 2: Zero Trust Maturity Journey

Summary: 'Optimal' where applicable, across the 5 of the 6 pillars, particularly when integrated with other technologies

1. **Identity**: Zit has its own CA/PKI with ability to utilize third party & integrate with IdPs as well as usage metrics per service per endpoint
2. **Devices:** Various device posture checks (incl. TOTP, domain join OS, process identification, MAC address) with periodic renewal. Further ability to be app-embedded to not trust host OS even if compromised.
3. **Networks**: Supports different service specific micro segmentations across ZTNA/ZTHA/ZTAA, least-privilege, with various encryption & BYOE, and HA resilience. Deep telemetry with 'default-deny' and closed ports. Driven with SW & APIs.
4. **Apps and workloads**: Ability to make apps available across public networks with highest security, with ability to act as strong kill point for attack chains.
5. **Data**: OpenZiti only impacts data in motion, it plays only it's access controlling role within this category.
6. **Cross-Cutting**: Provides events and metrics (up to L4) and provides single view for auditing across the entire network instance(s) with automation and orchestration.

NETFOUNDRY

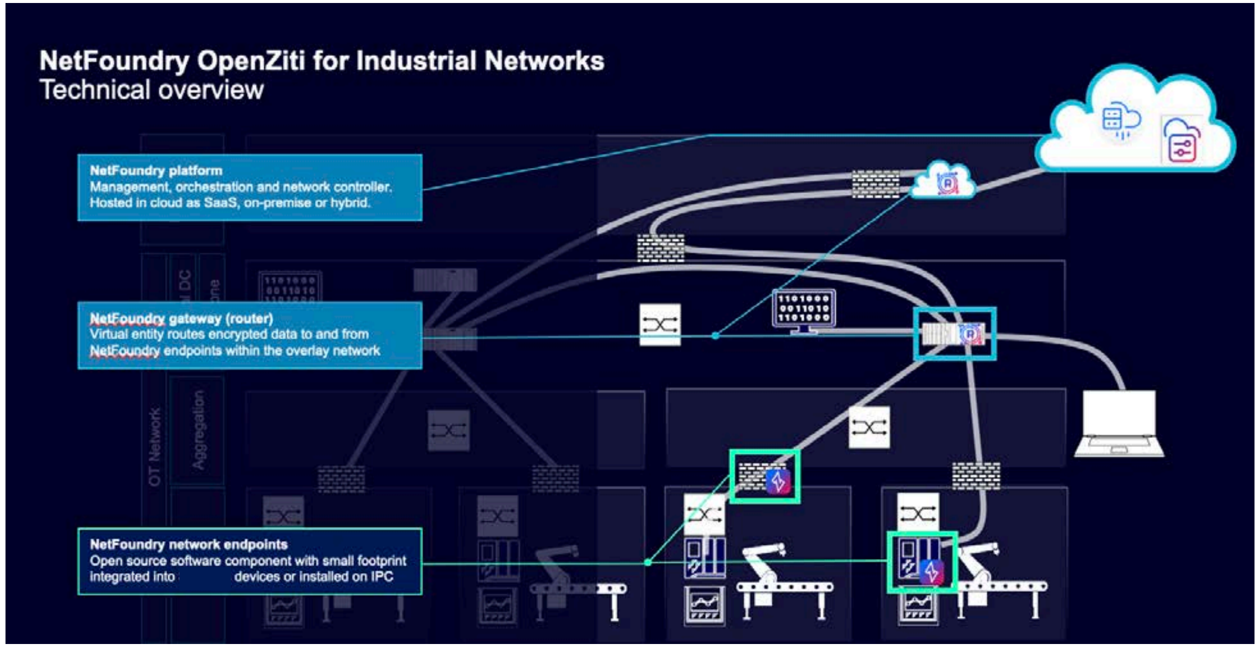
NETFOUNDRY™

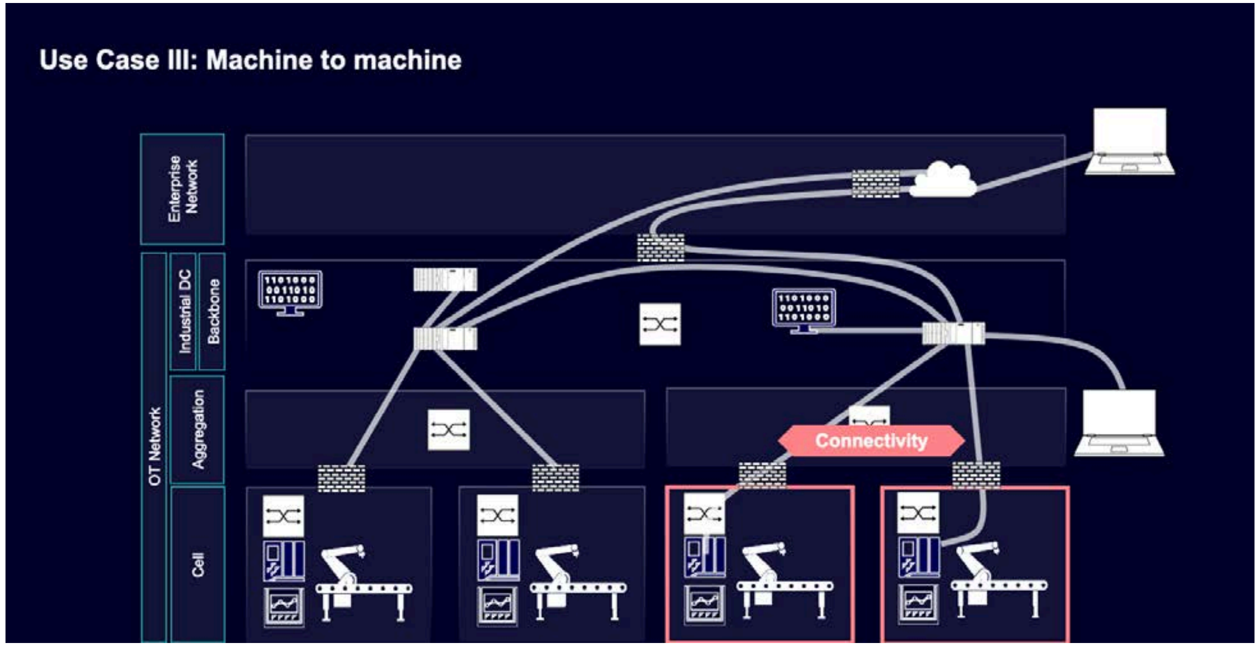
Why embedded ZTN is so key

NETFOUNDRY

Native Zero Trust Networking

Industry Partners: Manufacturing





"Ziti provides us with ZT that can support any use case with no SPOF, including air gapped"

Head of Cyber for Portfolio

"Ziti can support any use case, uniquely including M2M which is 80% of traffic in a factory"

Leader for Zero Trust

Use Case I: Machine to cloud connectivity
Challenge for e.g. FA factory Singapore

Use Case II: Machine to MES in datacenter
Challenge for e.g. EWA

Use Case III: Machine to machine

Use Case IV: Remote Access

Native Zero Trust Networking

Industry Partners: Edge / IoT

https://www.lfedge.org/2021/12/15/edgex-3-0-the-future-of-edge-computing/

EdgeX Ireland Release Services	Image Footprint[1]	Memory Consumption[1]	CPU Consumption[2]
EdgeX Core Data	20 MB	11 MB	0.02%
EdgeX Core Metadata	17 MB	11 MB	0.11%
EdgeX Core Command	16 MB	11 MB	0.02%
EdgeX Device Virtual	24 MB	13 MB	0.01%
EdgeX Device REST	21 MB	11 MB	0.01%
EdgeX Support Notifications	17 MB	11 MB	0.03%
EdgeX Support Scheduler	16 MB	11 MB	0.09%
EdgeX App Service Configurable	25 MB	12 MB	0.01%
EdgeX Security Proxy Setup	26 MB	0 MB [3]	0% [3]
EdgeX Secret Store Setup	29 MB	0 MB [3]	0% [3]
EdgeX Security Bootstrapper	19 MB	0 MB [3]	0% [3]
Consul	122 MB	41 MB	0.58%
eKuiper	25 MB	37 MB	0.01%
Redis Database	32 MB	6 MB	0.2%
Kong (with Postgres DB)	199 MB	748 MB	0.67%
Vault	207 MB	126 MB	2.97%

EdgeX micro services and 3rd party services (below the red line) performance metrics

1 – image footprint (as determined by its container image size), and memory consumption rounded to the nearest MB
2 – Average CPU consumption when run on an HP MP9 G4 Desktop Mini PC, single Intel Core I7 processor with 16GB RAM
3 – these security services are only involved in setup/bootstrapping and then do not run or consume resources (memory or CPU) after startup.

https://www.linkedin.com/feed/update/urn:li:activity:7064745301881847808/

Bryon Nevis · 1st
IEEE Senior Member, Light Bulb Security Professional
3mo ·

As security working group chair of Linux Foundation's EdgeX Foundry, I have been working with Clint Dovholuk of NetFoundry for close to a year on a proof-of-concept integration of EdgeX with the OpenZiti zero-trust networking fabric. EdgeX has matured to the point that some customers are no longer running EdgeX on a single node exclusively, but solving authenticated secure network communication is a problem that distracts EdgeX adopters from the real problems that they are trying to solve.

A cloud-based service might solve these problems by running their services in Kubernetes, with sidecar injection and service meshes. The elasticity of the cloud means that it is easy to deploy control plane add-ons, daemonsets and sidecars with ease with and incremental operational cost. But the on far edge, constrained devices that have limited processing power, maybe only 1 or 2 GB of RAM, a few hundred gigabytes of disk, and no elasticity at all present a unique challenge.

In this [video](https://lnkd.in/gbnCcFek), Clint demonstrates a deep integration of OpenZiti into EdgeX. In this prototype, OpenZiti client libraries have been directly linked in to EdgeX's basic microservices and have replaced the standard TCP/IP listeners and dialers that most REST-based microservice architectures rely on. The demo also includes a "Zitified" a third-party component, the eKuiper rules engine, which was done with only a few lines of code. The Zitified services have no open HTTP ports that can be attacked, and all inbound REST calls are authenticated by an OpenZiti-linked identity. Only the OpenZiti control plane and edge router components bear the risk of exposed ports.

All this functionality is bootstrapped from EdgeX services' Vault-based identity that was added to the forthcoming EdgeX 3.0 release. No work was needed to create complex PKI hierarchies for TLS servers and TLS clients. No work was needed to convert HTTP listeners to HTTPS listeners and HTTP dialers into HTTPS dialers. No work was needed to select TLS algorithms or TLS ciphers or to manage certificate and key rotation. No work was needed pass a JWT on every outgoing microservice call and the check incoming JWT in every handler (although EdgeX 3.0 does exactly this, as the OpenZiti integration is still in the architectural design and prototyping phase). All that was needed was a way to bootstrap client and server identity, override the standard listeners and dialers, and connect to the OpenZiti infrastructure

NETFOUNDRY

Native Zero Trust Networking

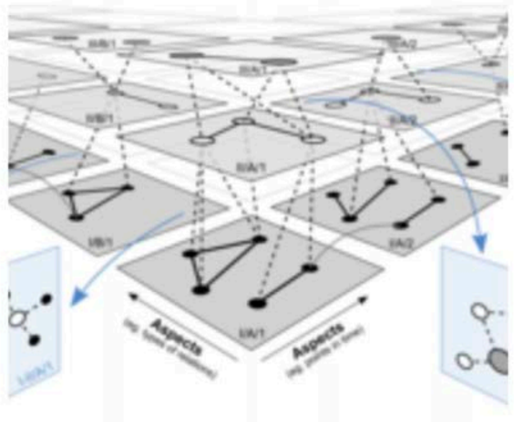
Industry Partners: Grid and Energy

Inter-Network Coordination Needs

- How do I remotely connect to my DER?
- How do I grant a remote OEM support staff access to my system?
- How do I share secure information between two distributed DMS (FLISR) deployment?
- How do I affect DRA access policy changes across multiple Blueframe systems?
- How do we securely aggregate data from in-substation to central systems? How do we administrate centralized access controls to in-substation solutions.
- How do I move data from my OT on-prem solutions to my business systems which include the cloud?

Today's approach with Legacy Technologies

- Connect many different networks together with routers and firewalls
- Create various tunnels (IPSec etc..) with encryption to span or tunnel between networks
- Add additional firewall rules to restrict accessibility
- Spanning encryption zones requires reverse proxies with extra certificates or service accounts.



"We created a theoretical whitepaper on what 'service mesh for DER' should look like, it basically described OpenZiti before we knew Ziti existed"

Senior Engineer

Problem

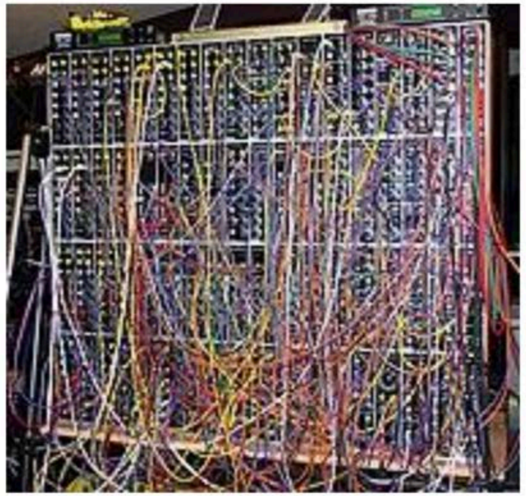
Multiple network administration systems must be manually choreographed!

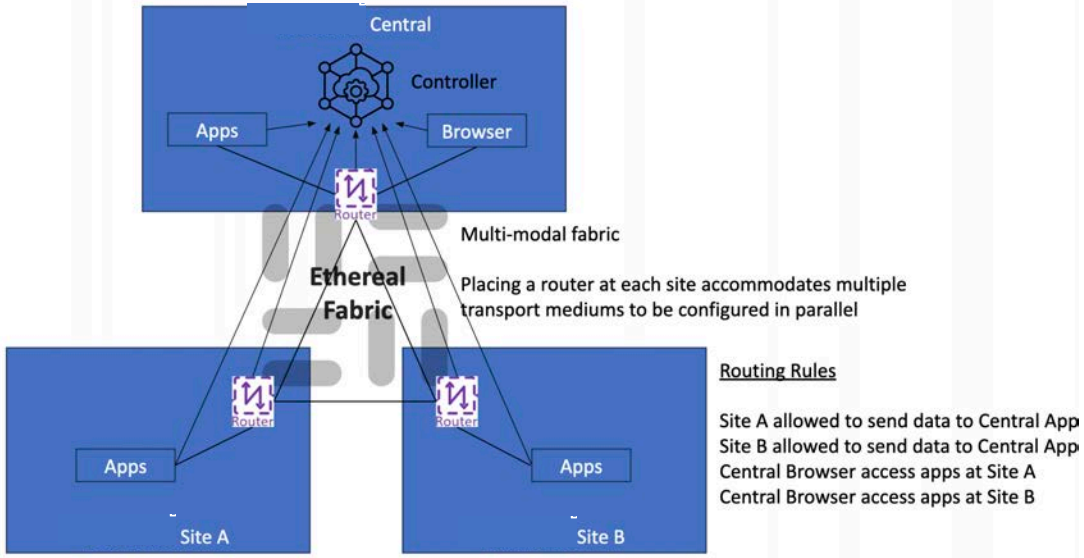
Expensive to Keep Working

- Getting new data reliably through the system
- Certificates expire and PKI becomes an ongoing critical-path maintenance issue

Difficult to Audit & Control

- How do we audit that the resulting system doesn't allow abuse
- Problem - Service accounts = shared credentials





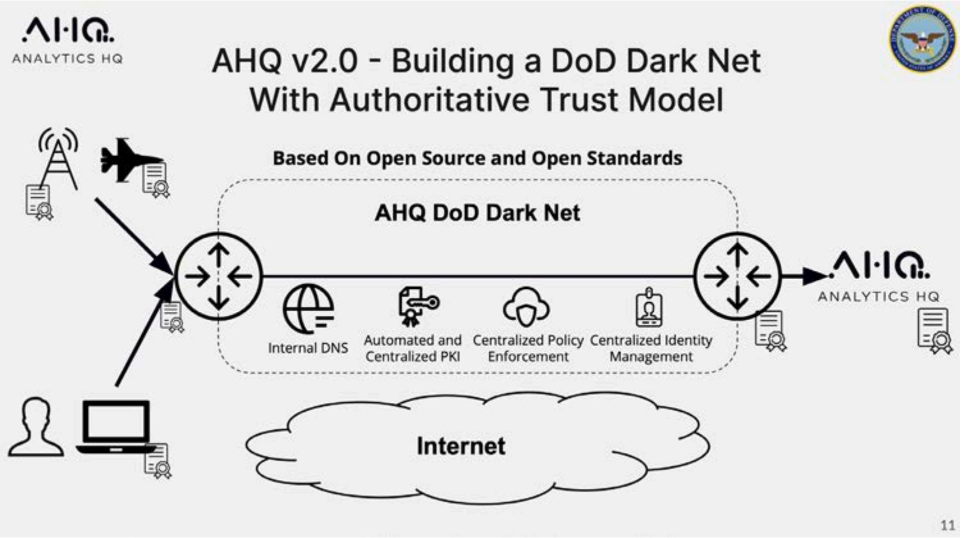
NETFOUNDRY

Native Zero Trust Networking

Industry Partners: Analytics and K8S

"We can connect services, from K8S to legacy (non-K8S), with each service completely 'air gapped', 'dark' and Secure by Default"

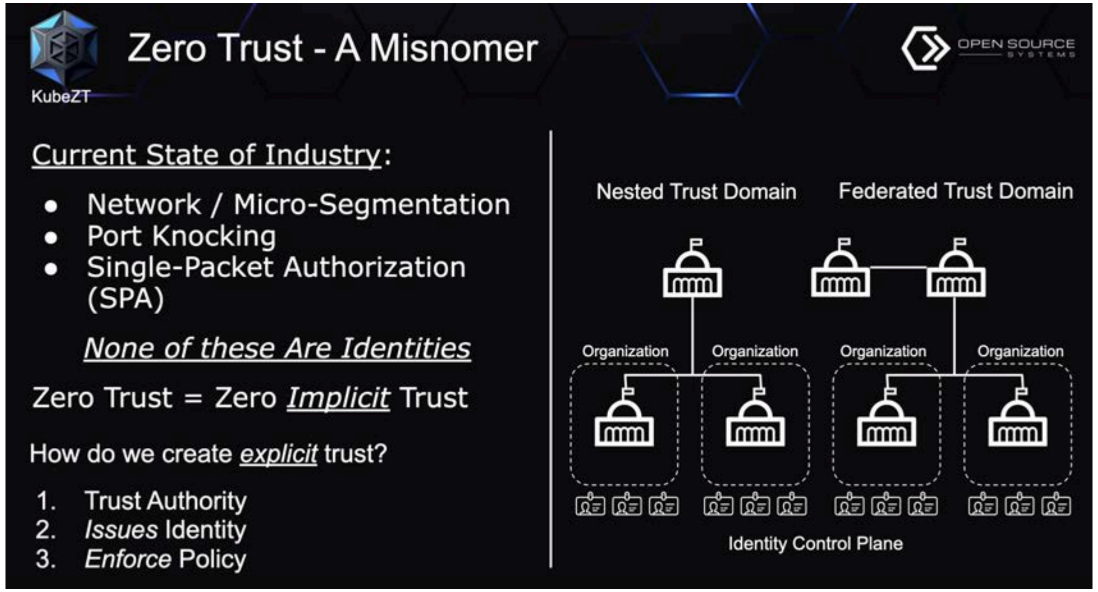
Chad, CTO, OSSYS

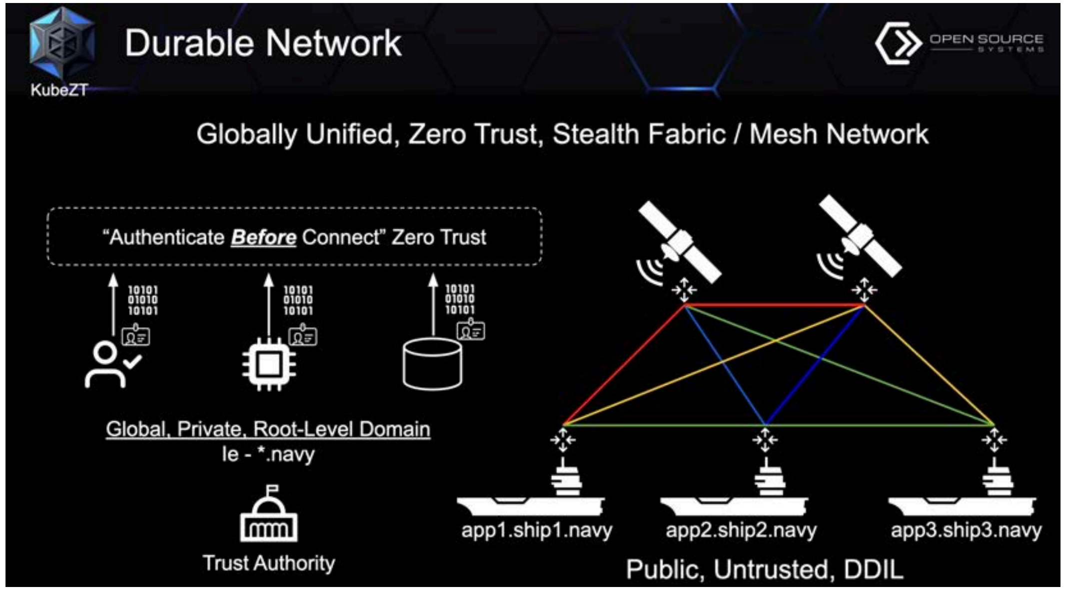


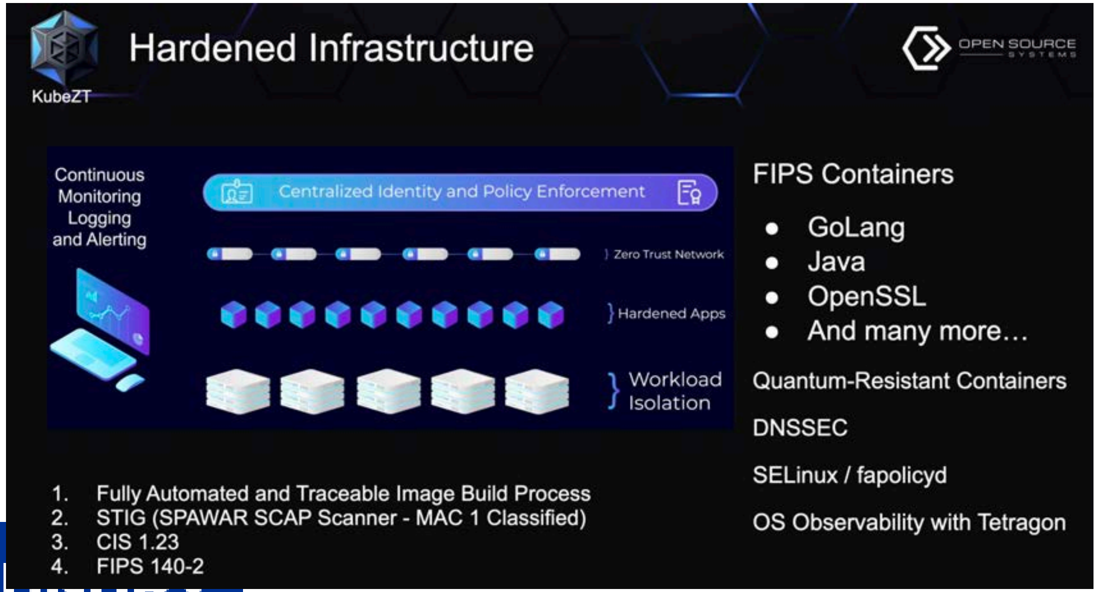
NETFOUNDRY

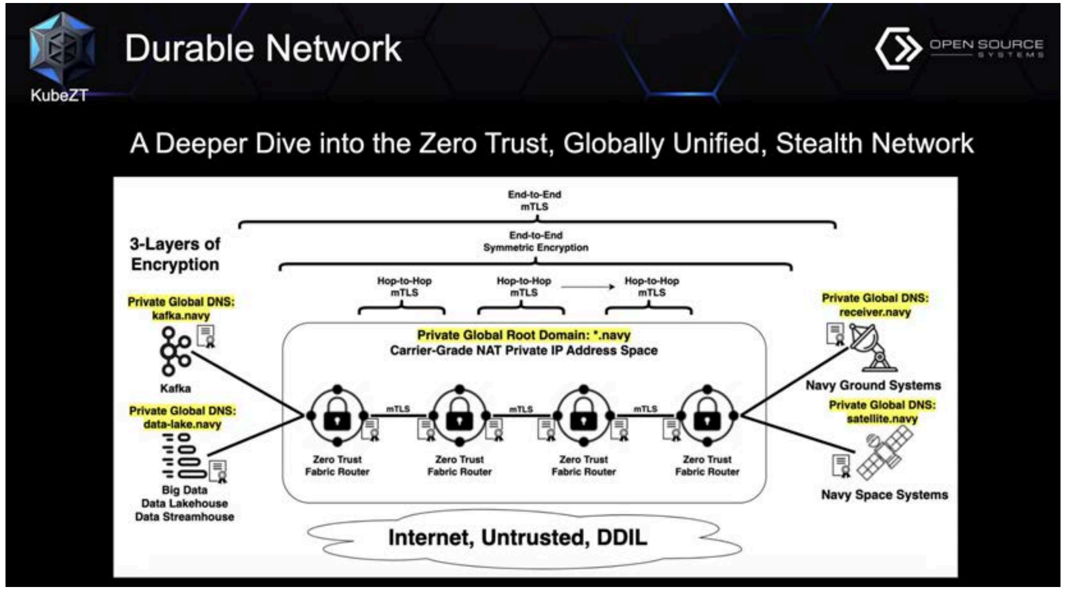
Native Zero Trust Networking

Industry Partners: Next-Generation Application Platform









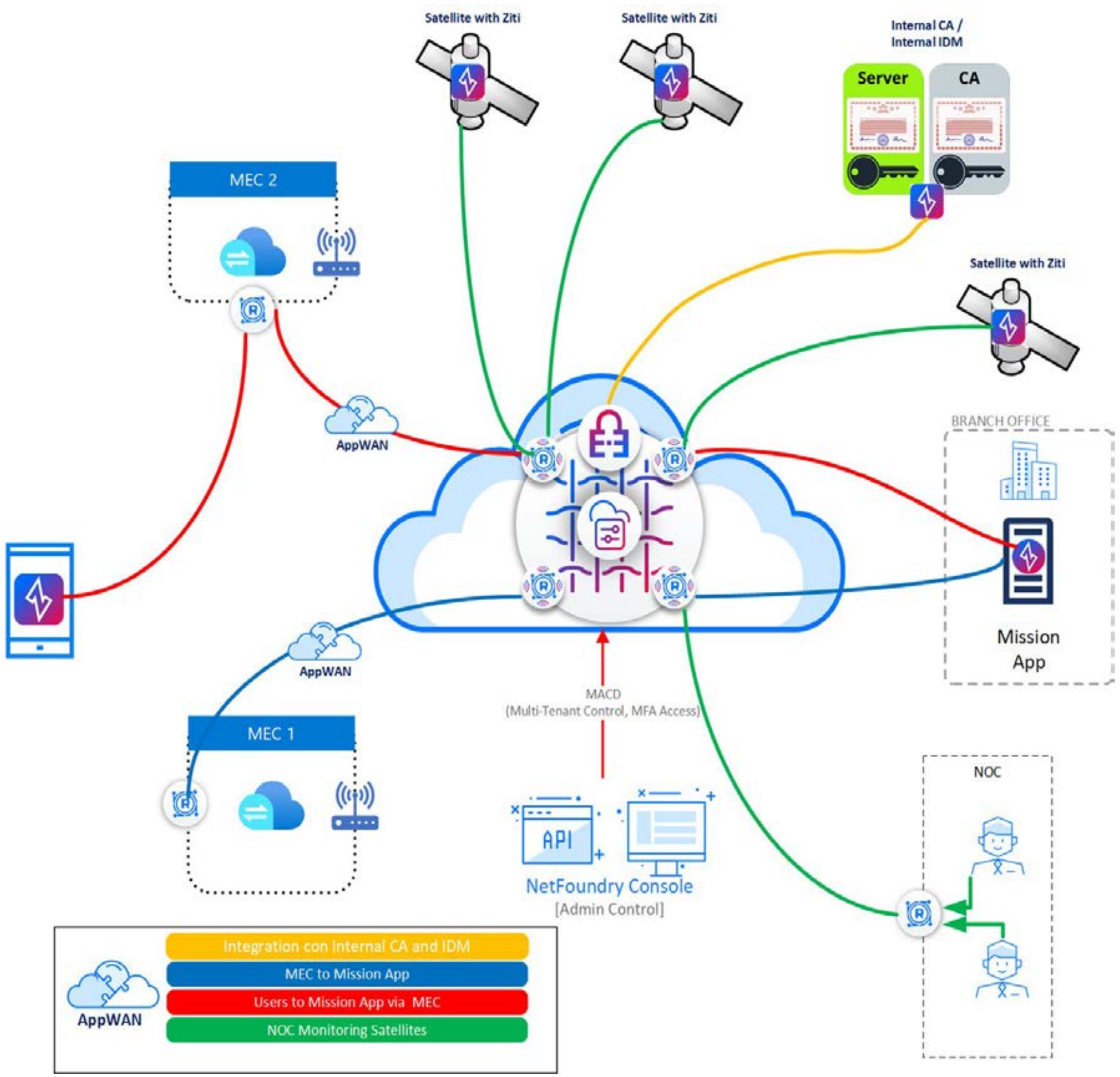
Native Zero Trust Networking

Industry Partners: F100 US Defence Contractor

Air gapped, military 5G network, with drones, mobiles and more, incl. micro segmentation between trust zones

"the best adherence to NIST 800-207, including micro-segmentation and E2E encryption… with a breadth of architectures… so we can run on anything—from containers to embedded, including less resource-intensive far edge. It includes its own CA/PKI to start without doing any expensive integrations like AD, as well as the ability to provide their own CA. Completely air gapped."

Fellow, ZT Leader, US Defence Contractor



NETFOUNDRY