xiid | PORTAINER.iO

Whitepaper

Visibility and Control at the Industrial Edge

A unified architecture for managing distributed OT infrastructure using containerized workloads and secure commercial networks.

Aug 2026

Executive summary

In most industrial operations today, when a motor fails or a process drifts out of spec, the failure is discovered after the fact. Power generation, HVAC, physical security systems, and building management infrastructure run continuously across plants, distribution centers, utility installations, and remote sites. These systems are, in the vast majority of cases, decades old, proprietary, and entirely unmonitored at scale. There is no unified visibility layer. There is no early warning. There is no way for central operations teams to understand the operational state of distributed OT infrastructure without dispatching personnel.

This white paper describes an architecture that changes that. Portainer provides a containerized workload management plane capable of running on the smallest OT gateway hardware. Xiid's Terniion provides deterministic, outbound-only secure connectivity that makes OT devices non-addressable from the public internet while maintaining continuous, encrypted telemetry back to a self-hosted central operations plane.

The stack is self-hosted end-to-end. There is no SaaS control plane. There is no external infrastructure dependency. Every component runs on hardware the operator controls, in facilities the operator controls.

Because OT environments are heterogeneous, this paper describes the architecture in three layers: a generic technical foundation that applies to every deployment, a set of deployment patterns matched to common industrial scenarios, and a set of worked examples showing how the same building blocks address very different operational realities.

The OT management problem in distributed industrial operations

Industrial operators can manage some of the largest physical footprints in any sector. A national utility operates substations and pumping stations across thousands of sites. A multi-plant manufacturer runs production lines at dozens of facilities. A logistics operator runs warehouses and distribution centers in various regions. Every site depends on a substrate of operational technology: power distribution equipment, HVAC and environmental control systems, physical access control, fire suppression, process control, asset tracking, and a long tail of site-specific machinery accumulated over decades of acquisition.

Taken in aggregate, that infrastructure represents a significant operational risk that is currently largely unmanaged at the system level. Not because operators lack capable site personnel, but because the tooling required to provide centralized visibility into distributed OT environments has historically been either

- Unavailable
- Architecturally incompatible with the security constraints of an industrial environment
- Technically feasible but commercially out of reach

Even where the "right" solution exists, scaling it across an operator's full site portfolio often outruns budgets that don't flex with scope. A pilot at three plants is a different conversation from a deployment across two hundred

The visibility gap

Most OT infrastructure was installed under a simple operational assumption: the devices are local, the technicians are local, and if something breaks, a person on site will notice and respond. That model worked when sites were staffed at levels that made continuous physical monitoring feasible. It does not work in a resource-constrained environment where operations teams cover large footprints with limited headcount, and it fails completely in remote or unmanned locations where specialist personnel are not continuously present.

The result is a visibility gap. Central operations has no real-time view of equipment load across sites. There is no aggregated alerting when HVAC, motor, or process performance degrades below threshold. Predictive maintenance is effectively impossible without baseline telemetry. Equipment runs to failure because there is no early signal. Mean time to detection for OT failures is measured in hours or days, not minutes.

The security gap

Alongside the visibility problem sits a security problem that is less visible but more consequential. OT devices were not designed with network security in mind. PLCs and building management systems from the dominant industrial vendors (Siemens, Allen-Bradley, Schneider Electric, Honeywell, Johnson Controls) frequently communicate over legacy protocols (Modbus, BACnet, DNP3, PROFIBUS) that have no authentication, no encryption, and no concept of access control. When these devices are connected to any network, even a local one, they are inherently exposed.

The standard approach to securing legacy OT, placing devices behind air gaps and firewalls, has two failure modes. The first is that the air gap is not maintained: over time, maintenance access requirements, remote monitoring tools, and ad hoc network extensions erode the isolation. The second is that even a properly isolated OT network provides no protection against insider threats or physical access attacks, and provides zero visibility to the central team that needs to manage the infrastructure.

A modern architecture must do two things simultaneously: provide the visibility that distributed operations require, and do so in a way that reduces rather than increases the attack surface of OT infrastructure that cannot be patched, updated, or replaced on any near-term timeline.

Why current approaches fall short

The obvious responses to the OT visibility problem are not wrong, but they are insufficient. It is worth being specific about why, because the limitations of existing approaches are what drive the architectural choices in this paper.

Network attribute	Traditional SCADA / BMS	VPN-based remote access	Air-gap + manual inspection	Joint architecture: Portainer + Xiid
Centralized visibility across sites	⊗ vendor-siloed, per-deployment islands	○ partial; depends on per-site setup	⊗ none beyond what's on-site	✓ unified across the fleet
Inbound network exposure	○ central server exposed; vendor cloud connection	⊗ open ports per connected device	✓ none, by design	✓ none, outbound-only by design
Scale across many sites	⊗ each deployment is its own silo	○ degrades operationally at scale	⊗ requires personnel travel	✓ thousands of sites from one console
Reduces on-site visits / truck rolls	○ partial; some remote capability	○ partial; tied to network access	⊗ every routine event requires a visit	✓ remote deployment, updates, and config
Air-gap and regulatory fit	⊗ cloud-dependent in most deployments	⊗ exposes legacy OT to public networks	○ at the cost of visibility	✓ self-hosted, air-gap capable
Central update management	○ within vendor stack only	⊗ per-device, manual	⊗ none	✓ GitOps, fleet-wide

✓ Capability is well supported and proven at scale

○ Partial; degrades with scale or operational context

⊗ Capability not provided, or actively counter-productive

Traditional SCADA and BMS platforms

Supervisory Control and Data Acquisition (SCADA) systems and enterprise Building Management Systems (BMS) from vendors like Honeywell Forge, Siemens Building X, and Schneider EcoStruxure are designed to solve exactly this problem for commercial and industrial customers. They provide centralized dashboards, historian databases, alerting, and in many cases remote control capabilities. They are mature, well-understood products.

The problem is the deployment model. Cloud-hosted SCADA/BMS raises data sovereignty and regulatory concerns. On-premises deployments require either significant infrastructure at every site or a secure network path from each site to a central server. And critically, every existing SCADA or BMS deployment at a site is often its own island — its own data model, its own vendor contract, and no interoperability with the sites next to it.

VPN-based remote access

Adding VPN connectivity to legacy OT devices is the conventional answer to the remote access problem. It is also the wrong answer: VPNs require open inbound ports, which means every VPN-connected OT device has a publicly routable presence that can be enumerated and attacked. A PLC running firmware from 2009 with a VPN client bolted on in 2024 is not more secure than it was before. It is a larger attack surface with a more complex failure mode.

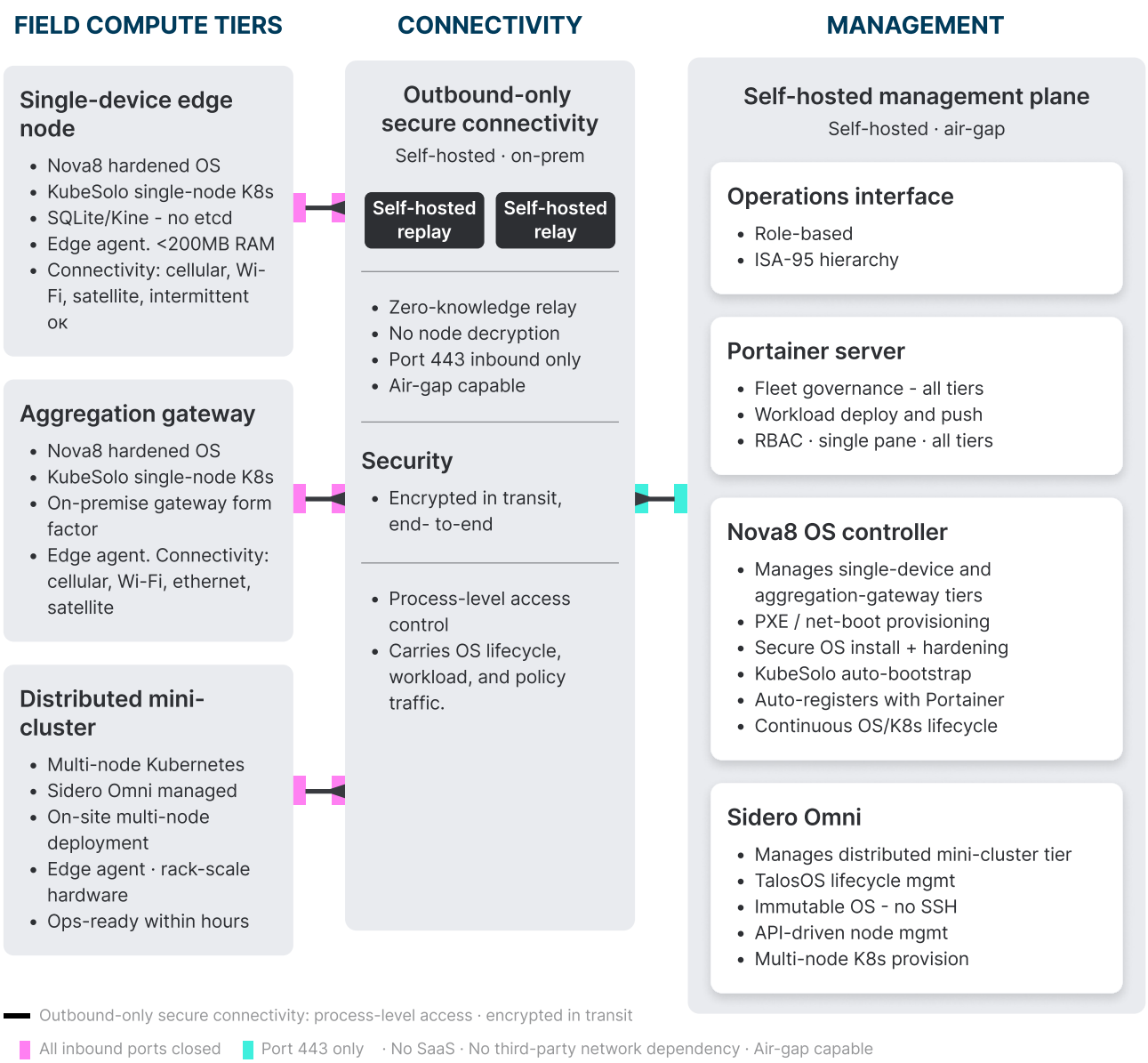
Manual inspection and periodic audits

The current baseline for many distributed industrial operators is periodic physical inspection by site personnel, supplemented by reactive maintenance when failures are reported. This model is not a security or visibility strategy. It is the absence of one. It cannot provide early warning of equipment degradation, cannot support predictive maintenance planning, and cannot give central leadership any meaningful picture of infrastructure health across a distributed portfolio.

The joint architecture: Portainer and Terniion

The architecture combines two core technology layers. Portainer provides the containerized workload management plane. Terniion provides the secure, outbound-only connectivity layer. Together, they create a platform on which OT data collection, normalization, and transport can run at scale across any number of sites, without requiring inbound ports, external cloud dependencies, or vendor-specific integration work at the central management level.

The third layer, the software that actually speaks to PLCs and building systems in the field, is not specified in this document by design. That layer depends on what hardware is actually deployed at sites, what protocols those devices use, and what the operator wants to do with the data once it is collected. It is treated in a dedicated section later in this paper.



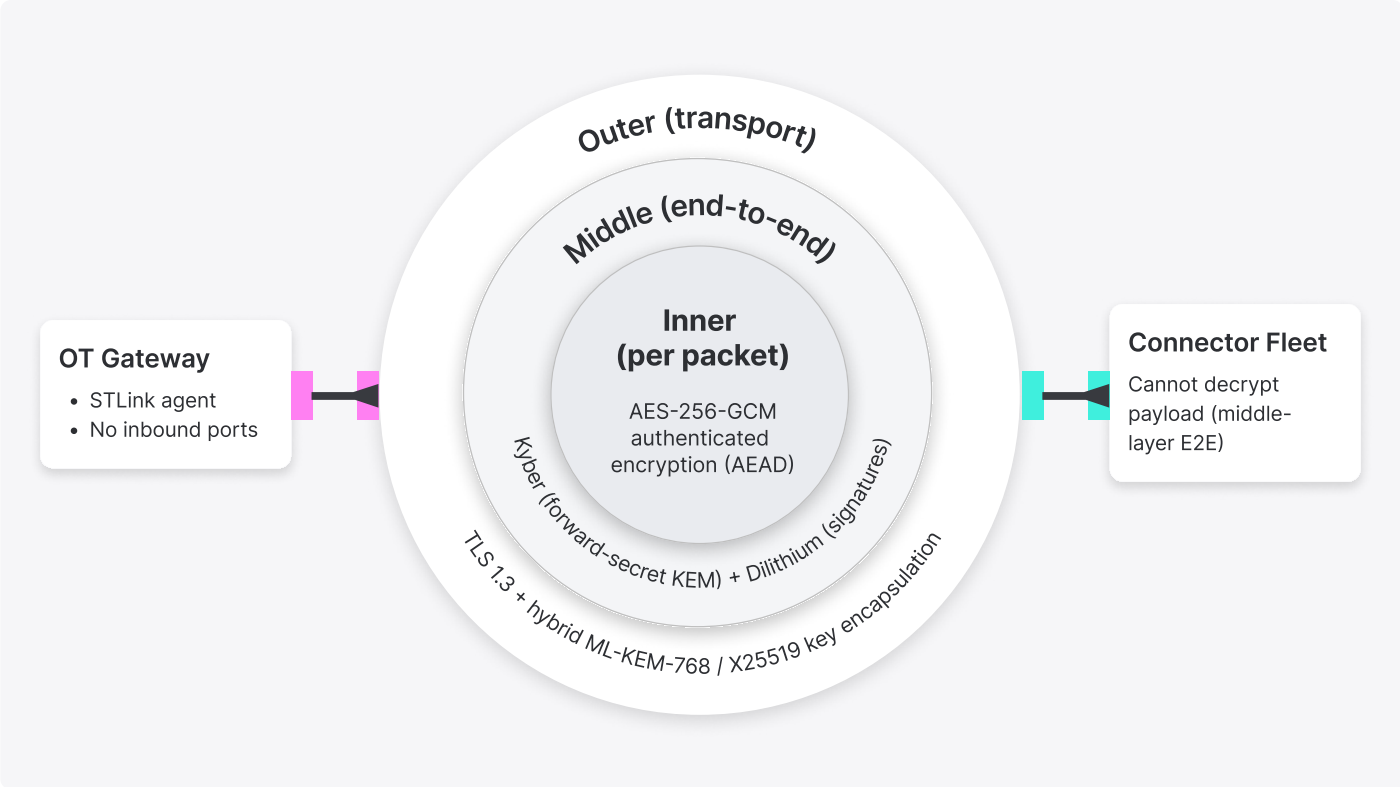
Terniion: erasing the OT attack surface

Every OT gateway deployed under this architecture runs a Terniion STLink agent as a container. On startup, the STLink immediately closes all inbound firewall ports on the host device. The gateway stops listening for incoming traffic entirely. It has no public IP presence. From the perspective of any external network observer, it does not exist.

The STLink then initiates a secure, outbound-only encrypted tunnel to the self-hosted Terniion Connector fleet, co-located with the operator's central management infrastructure. Because modern stateful firewalls allow outbound traffic to pass and accept the corresponding return traffic natively, the STLink establishes a full bidirectional communication channel without a single inbound port being opened. OT telemetry flows out through the tunnel. Management commands, configuration updates, and software deployments flow back in through the same tunnel. The underlying commercial network sees only encrypted outbound traffic on port 443.

The encryption stack carries three simultaneous layers. The outer layer handles transport using TLS 1.3 with hybrid ML-KEM-768 and X25519 key encapsulation, providing protection against both classical and post-quantum attacks. The middle layer establishes true end-to-end key negotiation using Kyber for forward-secret key encapsulation and Dilithium for digital signatures, ensuring the Connector fleet itself cannot decrypt the payload. The inner layer applies AES-256-GCM authenticated encryption with per-packet tamper detection.

Network access within the tunnel is bound to the process level. A specific containerized OT collector on a gateway is granted a tunnel bound exclusively to the specific process at the central operations site it needs to communicate with. Tunnel binding can be configured at the process, container, deployment, or host level depending on the operational granularity needed. A compromise of the local operating system on a gateway device cannot be used to pivot into the broader operations network because the tunnel does not grant subnet-level access. It grants process-to-process access and nothing else.



Why three layers?

Each layer protects against a different attack class. A compromise of any single layer does not expose the payload. The stack exceeds CNSA 2.0 requirements, and every component is FIPS 140-3 validated.

Layer	Cryptographic component	Function / what it does	Protects against
Outer (transport)	TLS 1.3 + hybrid ML-KEM-768 / X25519 key encapsulation	Transport security at network boundary Post-quantum key encapsulation	Network observers; future quantum attacks against classical key exchange
Middle (end-to-end)	Kyber for forward-secret key encapsulation + Dilithium for digital signatures	End-to-end key negotiation Zero-knowledge to connectors	Compromised relays; forged signatures; man-in-the-middle attacks at the connector tier
Inner (per-packet)	AES-256-GCM authenticated encryption with associated data (AEAD)	Per-packet encryption and tamper detection, applied to the actual payload	Packet tampering; data manipulation in flight; replay attacks

An optional hardened OS layer: Nova8

For industrial deployments that require stronger isolation between the operating system and the containerized workloads it runs, the architecture supports Nova8 as the underlying OS. Nova8 is a hardened, immutable micro-OS purpose-built to run KubeSolo. The OS image is a single signed file verified at boot by a hardware root-of-trust, with disk encryption tied to the device's onboard TPM.

The root filesystem runs in RAM rather than from disk, removing an entire class of OS-level drift and tampering. Nova8 handles its own installation, hardening, and KubeSolo bootstrap from a pre-staged image, and the central management plane handles all subsequent OS updates, security patches, and Kubernetes version management without manual intervention at the site. Useful where the deployment context demands cryptographic verification of every node from boot through workload, or where field hardware cannot be re-provisioned by a technician.

Portainer KubeSolo: containerized workload management on minimal hardware

Modern software development and deployment rely entirely on containerization and Kubernetes orchestration. However, standard Kubernetes distributions are fundamentally designed for high-availability cloud data centers, rendering them entirely unsuitable for the constraints of an edge environment.

Standard Kubernetes relies on highly complex clustering logic. It requires multiple nodes to communicate constantly to establish a quorum, and it uses etcd as a distributed key-value store to maintain the state of the cluster. In an edge setting, a device is often a single ruggedized computer or a small-form-factor gateway operating independently. If a standard Kubernetes node loses its connection to the central control plane due to an intermittent network link, the clustering logic fails, and the local applications crash.

Additionally, standard Kubernetes distributions require significant memory and processing power simply to run the underlying orchestration components. Small edge devices, such as those used in smart meters, industrial controllers, or forward-deployed field sensors, lack the system resources to support this massive overhead.

The joint solution addresses these constraints by implementing Portainer KubeSolo. KubeSolo is a single-node Kubernetes distribution built for environments currently running Docker or Podman that need full Kubernetes API compatibility without the operational overhead of a multi-node cluster. It runs on resource-constrained edge hardware where conventional Kubernetes cannot. It strips away the unnecessary complexity of cloud-based Kubernetes while retaining full compatibility with modern containerized workloads. KubeSolo eliminates clustering logic entirely. There is no multi-node orchestration, no quorum requirement, and the resource-heavy etcd database is replaced with a highly efficient SQLite backend managed via Kine. This architectural shift allows KubeSolo to run comfortably using less than 200MB of RAM, making it ideal for deployment on minimal hardware, including simple devices running on SD cards.

Despite its exceptionally small footprint, KubeSolo remains fully OCI-compliant, Helm-ready, and supports Custom Resource Definitions (CRDs) alongside standard Kubernetes Operators. This standard compliance ensures that developers can build applications using familiar cloud-native practices and deploy them directly to edge nodes without modification.

KubeSolo fits single-node edge deployments. For sites that require multi-node Kubernetes clusters (aggregation servers in a plant data center, regional clusters consolidating telemetry from multiple gateways, or operations centers running heavier workloads) Portainer's recommended OS is Talos Linux: an immutable, API-driven operating system purpose-built for Kubernetes, with no SSH access and no mutable runtime state. Portainer manages KubeSolo single-node deployments and Talos multi-node clusters from the same management plane.

Portainer: managing the OT gateway fleet

Portainer serves as the central management plane for every OT gateway in the deployment, managing containerized workloads across Docker, Podman, and Kubernetes environments through a single console. Administrators have a unified dashboard from which they can deploy containerized OT workloads to any gateway in the fleet, monitor the health of running containers, push configuration updates, and manage the full software lifecycle of field devices, regardless of the underlying hardware or the network those devices are currently connected through.

This matters because OT gateway environments are inherently heterogeneous. A legacy building management system at a distribution center in one region may require a different protocol adapter than a process control system at a manufacturing plant in another. Portainer's container-based deployment model means those differences are handled at the software layer, in containers that can be updated, replaced, or extended without physical access to the gateway hardware. The gateway itself is treated as a standardized compute platform. What runs on it is managed centrally.

For installations where existing PLC or IPC hardware has sufficient compute headroom to run containers natively, no additional gateway hardware is required. Portainer deploys directly to those devices. For legacy devices that cannot run containers, a small ruggedized OT gateway (such as a Raspberry Pi CM4 industrial carrier, an NVIDIA Jetson Orin, or any commercially available industrial mini-PC) is positioned physically adjacent to the existing hardware and handles all external communication on its behalf.

This approach avoids any need to modify or replace existing field hardware. Portainer offers three agent deployment models matched to connectivity profile: a LAN Agent for trusted networks with bidirectional in-band communication, a Remote Agent that uses outbound-only communication with no inbound firewall rules, and an Async Edge Agent built for offline-safe operation with commands queued centrally and executed when the gateway next reaches the management plane. For industrial sites with intermittent connectivity, the Async Edge Agent is typically the right choice; the Remote Agent suits sites with reliable connectivity that still need strict outbound-only behavior.

CONTROL PLANE FAILURE DOES NOT STOP WORKLOADS.
Portainer separates management operations from runtime execution. If the Portainer Server becomes temporarily unavailable, managed gateways continue running, application workloads continue executing, and Terniion tunnels remain established. Management access is temporarily unavailable; field operations are not.

HARDWARE NOTE
The reference demo architecture for this joint solution runs on commercially available, off-the-shelf hardware: industrial controllers such as the WAGO CC100 and Revolution Pi Connect 4, ARM-based compute as small as a Raspberry Pi Zero 2W, and x86 mini PCs. Power compatibility spans 100-240V input. Nothing in the architecture requires proprietary or single-vendor hardware.

Deployment patterns across industrial environments

The architecture supports three deployment patterns that map to the operational realities of distributed industrial operations. Most operators have more than one of these patterns running across their footprint.

PATTERN A

Fixed industrial installations

Manufacturing plants, distribution centers, fixed utility installations, water and wastewater treatment, building management at office and retail estates. The defining characteristic is that the OT hardware is permanent: it was commissioned years or decades ago, it runs continuously, and it is rarely if ever physically moved.

For installations where existing PLC or IPC hardware has sufficient compute headroom to run containers natively, no additional gateway hardware is required. Portainer deploys directly to those devices. For legacy devices that cannot run containers, a small ruggedized OT gateway is positioned physically adjacent to the existing hardware and handles all external communication on its behalf. The deployment is one-time. Once the gateway is installed, it stays there, and all subsequent updates - protocol drivers, monitoring rules, configuration changes - happen over the Terniion tunnel without anyone returning to site.

PATTERN B

Mobile or transient industrial deployments

Field service vehicles, mobile inspection units, mining vehicles and equipment, agricultural machinery, construction sites, oil and gas field operations, utility line crews, deployable communications and command units. The defining characteristic is that the compute is physically moving or temporarily deployed, on networks that change as the asset moves.

The same Nova8 and KubeSolo stack extends to vehicle-mounted compute, installed as a micro-rack in service vehicles or mobile industrial assets. The OS lifecycle is managed identically to the fixed tier: the central management plane provisions and maintains the device, KubeSolo provides the container orchestration layer, and the Terniion STLink agent manages the outbound tunnel. The additional compute headroom relative to a stationary tier supports heavier inference workloads and allows the vehicle node to act as a local aggregation point for nearby sensors or smaller field devices. Connectivity uses cellular and SATCOM links, with Terniion re-establishing the tunnel automatically each time the vehicle moves between networks.

PATTERN C

Hybrid operations

Most industrial operators run a combination of Pattern A and Pattern B in a mixed deployment. A multi-plant manufacturer has fixed lines (Pattern A) plus mobile maintenance and inspection rigs (Pattern B). A utility has substations and pumping stations (Pattern A) plus field crew vehicles (Pattern B). A logistics operator has warehouses (Pattern A) plus mobile fleet telemetry (Pattern B). The architecture handles both patterns through the same management plane, the same security model, and the same containerized workload approach.

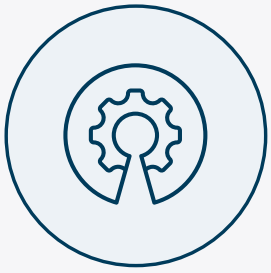
Operations does not have to learn two systems. Portainer's hierarchical environment model lets operators organize sites into a user-defined tree (geography, business unit, OT line, compliance boundary), where policies, access controls, and application deployments applied at a parent node cascade automatically to child sites. Operators apply intent once at the level that makes sense, rather than configuring each site individually.

The software layer: what runs in the containers

The architecture described is a delivery platform. Portainer manages what runs on the gateways. Terniion secures how it communicates. What actually runs on the gateways, the containerized software that speaks Modbus to a PLC, translates BACnet from a building management system, or pulls SNMP data from an environmental sensor, is the layer that must be specified based on what the operational requirements of a given site.

There are three possible shapes this layer takes, depending on the operator's existing technology relationships and operational requirements.

Option A



Open-source collector stack.

If the operator's primary requirement is visibility and alerting, a containerized stack built from established open-source OT middleware (Node-RED for protocol translation, Telegraf for metric collection, InfluxDB for time-series storage, Grafana for dashboards) can be deployed within the Portainer management plane against any standard industrial protocol. This approach has no additional software licensing cost, is vendor-neutral, and can be adapted to new protocols by updating the collector container without replacing the gateway hardware.

Option B

Commercial OT middleware.

If the operator already has a relationship with a commercial OT platform vendor (Kepware, Inductive Automation Ignition, PTC ThingWorx, Softing, or similar), the platform's data collector or edge gateway component can be containerized and deployed via Portainer to field gateways, with Terniion providing the secure backhaul. This positions the joint Portainer and Terniion stack as the secure deployment and connectivity layer underneath the operator's existing software investment.

Option C

Discovery-defined specification.

If the OT hardware inventory is unknown or highly fragmented across sites, a site assessment will produce a protocol map and a vendor distribution. The software layer is then specified from that map, selecting the right combination of drivers, adapters, and middleware for what is actually in the field rather than what is assumed to be there.

How it works in practice: three industrial scenarios

The architectural capabilities are consistent regardless of the specific OT hardware in the field. Three industry scenarios illustrate how the same architecture maps to different operational priorities. These are deployment patterns drawn from common industrial contexts, not references to specific customer environments.

SCENARIO 1

Containerized edge analytics across global manufacturing plants

Manufacturing companies running platforms like Siemens Industrial Edge, Bosch IoT Suite, or custom containerized OPC-UA analytics applications operate hundreds or thousands of industrial PCs and edge servers across plants worldwide. Each plant might run dozens of devices; each device runs containerized applications that read from PLCs, run local analytics, and feed data to plant historians or cloud platforms. The operational requirement is consistent: deploy and update containerized applications across the fleet, monitor health centrally, and apply policy uniformly, without on-site IT staff at most locations and without expanding the inbound network exposure of plants that were never designed for safe remote connectivity.

The joint architecture deploys containerized OT collector workloads on gateways adjacent to each plant's existing controllers. Portainer manages the workload lifecycle across the fleet, deploying protocol adapters, pushing configuration updates, and surfacing alerts on a unified dashboard. Xiid Terniion secures every gateway's connection to the central management plane: outbound-only, no inbound ports opened at the plant, and process-level binding so a compromise of any individual gateway cannot pivot into the broader operations network. Plants that were historically managed as isolated islands gain central operational visibility without expanding the attack surface.

SCENARIO 2

Containerized analytics at utility substations and grid edge

Utility operators run containerized edge analytics at substations, water treatment facilities, and smart meter aggregation points. Applications process local sensor streams, run anomaly detection, and manage demand response automation. Sites are often unstaffed and may have limited WAN connectivity. The operational model requires a management system that tolerates intermittent connectivity and delivers application updates without site visits, paired with a security posture that does not depend on the integrity of the underlying network.

The joint architecture meets both requirements. Portainer's Async Edge Agent on each gateway buffers state locally and continues operating during connectivity gaps; deployment commands queue centrally and execute when the site reconnects. Xiid Terniion ensures that when the site is online, the connection back to central operations is outbound-only and post-quantum encrypted, appropriate for utility environments with strict regulatory and supply-chain requirements. For a utility operating hundreds or thousands of distributed sites with no on-site IT, this means the management plane works regardless of the connectivity profile of any individual site.

SCENARIO 3

Containerized in-vehicle and mobile-asset software management

Automotive OEMs, fleet operators, and industrial equipment manufacturers managing containerized software on mobile assets — connected vehicles, autonomous platforms, mobile inspection units, off-road equipment — face the same operational challenge as fixed-site industrial operators, intensified by movement. The asset's network connectivity changes as it moves. Updates land when the asset is in range. The fleet may number in the thousands or hundreds of thousands. Each asset runs containerized navigation, telemetry, payload processing, or operational software that must be consistently deployed and updated without physical intervention.

The joint architecture supports this pattern through agents engineered for low-bandwidth, intermittent vehicle networks. Portainer manages container orchestration across the entire fleet; the Async Edge Agent buffers state through coverage gaps and applies queued updates on reconnection. Xiid Terniion secures the connection between each mobile asset and the central management plane, with outbound-only tunnels that re-establish automatically as the asset moves between cellular, satellite, and other commercial networks. For a vehicle OEM or mobile industrial operator, this means the same containerized software approach can be deployed safely across asset populations where physical access for maintenance is not operationally feasible.

Compliance and audit posture

OT environments are increasingly subject to regulatory frameworks that mandate specific controls around network segmentation, audit logging, access management, and incident response. The architecture described in this paper aligns with these requirements by construction rather than by configuration.

The outbound-only connectivity model means there are no inbound firewall modifications required, no port forwarding rules to audit, and no perimeter exposure to assess. Devices are non-addressable from the public internet. Network access is constrained to the process level. These are architectural properties of the system, which means they are permanent rather than contingent on configuration discipline.

The post-quantum encryption stack (TLS 1.3 with hybrid ML-KEM-768, Kyber/Dilithium end-to-end, AES-256-GCM inner) is suitable for environments with strict data-in-transit requirements.

A deployment's overall compliance posture depends on how the full solution is implemented. Device hardware, operating system, change control, and operational process all matter. Within that context, Portainer and Xiid together provide the visibility, auditability, consistency, and isolation that industrial regulatory frameworks require. Each component can be configured to behave appropriately for the deployment's regulatory context. Portainer itself is FIPS 140-3 compliant and operable in FIPS-only mode where the broader solution calls for it.

The self-hosted deployment model means data sovereignty is preserved by default. No telemetry leaves the operator's infrastructure unless the operator explicitly configures it to. This matters for operators in regulated industries (utilities, pharmaceuticals, food and beverage, critical infrastructure) and for any operator with intellectual property concerns about exposing process data to a vendor-hosted control plane.

What the architecture delivers

The practical outcome of deploying this stack across an operator's sites is a central operations picture that does not currently exist. The specific shape of that picture depends on what is in the field and what the operator decides it wants to do with the data. But the architectural capabilities the platform enables are consistent regardless of the specific OT hardware in the field.

Four operational outcomes

Unified infrastructure visibility

Every OT device across every site contributes telemetry to a single normalized data stream. Real-time picture of distributed infrastructure without dispatching personnel

Secure remote access & configuration

An engineer at central operations can interact with a PLC at a remote site without exposing it to any public network and without opening any inbound ports. Complete audit trail.

Predictive maintenance + anomaly detection

Once baseline telemetry is established, deviation becomes detectable. Equipment that signals a problem before it fails can be serviced before it fails.

Fleet-scale software management

Containerized OT software updated, reconfigured, or replaced centrally without physical access to field hardware. Pushed once, propagated to the relevant subset.

Enabling unified infrastructure visibility

Power consumption, motor load, environmental readings, access events, process variables: most of this data already exists on OT devices, but central operations historically can't reach it without site visits or always-on network access that creates its own risk. This joint architecture changes that. Containerized data collection workloads on each gateway pull telemetry from connected devices and move it securely to whatever visibility layer the operator chooses, whether that's a commercial OT historian, an existing SCADA front end, or an open-source stack the team has assembled. The architecture removes the structural barrier to seeing across distributed sites. What that picture looks like, where it lives, and who can query it remain the operator's to design.

Secure remote access for maintenance and configuration

Where bidirectional control is in scope, the same Terniion tunnel that carries telemetry outbound can carry authorized management commands inbound. Access is constrained to the process level: the engineer's authorized session reaches the specific device they are authorized to manage and nothing else.

Enabling predictive maintenance at scale

Once telemetry is flowing from distributed sites to a central analytics layer, deviation from baseline becomes detectable. Equipment that signals a problem before it fails can be serviced before it fails. The architecture doesn't perform the analysis itself; it makes the data available at scale to the operator's chosen historian, time-series database, or anomaly detection platform. Predictive maintenance becomes a fleet-wide capability instead of a site-by-site project.

Fleet-scale software management

When a protocol driver needs to be updated, that update is pushed from the management plane to the affected gateways. When a new monitoring capability is added, it is deployed as a container update across the relevant subset of the fleet. The gateway hardware in the field does not need to be touched.

The architecture does not eliminate every challenge in distributed industrial operations. It does not normalize OT protocols at the semantic level. It does not interpret machine-level data. It does not replace domain expertise about what specific equipment is supposed to do and when intervention is warranted. What it does is remove the structural barriers - network exposure, hardware fragmentation, manual site visits - that have historically prevented operators from getting visibility and control across distributed infrastructure in the first place

Conclusions

Distributed industrial operations depend on OT infrastructure that is currently managed without the visibility tools the scale and distribution of that infrastructure requires. The risk is not hypothetical: it is the accumulated operational exposure of running power, process, environmental, and access systems across many sites with no centralized telemetry, no early warning capability, and network connectivity models that were designed before OT security was a meaningful concern.

The joint Portainer and Terniion architecture directly addresses this. A containerized, centrally managed OT gateway platform secured by outbound-only, post-quantum encrypted tunnels provides the visibility layer industrial operators need without introducing the attack surface that conventional remote monitoring architectures create. The entire stack is self-hosted, air-gap capable, and built on commercially available off-the-shelf hardware. The architecture is not proposed in theory. It is deployed and operational in significantly more constrained environments today.

Industrial operators looking to start with this architecture should do so the way any responsible OT modernization begins: with a structured site assessment that maps the actual hardware, protocols, and network conditions across a representative sample of sites. The architecture is mature; what is unknown for any specific operator is the precise composition of their own field environment. Specifying a full deployment without that information would produce a project that looks credible on paper and fails in practice. A site assessment closes that gap and provides the technical foundation for a precise deployment plan.


xiid
PORTAINER.iO