One Size Does Not Secure All

Why Critical Infrastructure Needs Customisable OT Security

How sector-aware detection, asset-specific context and configurable use cases make OT security more effective across diverse critical infrastructure environments.

Velos Shield Case Study Series

Executive Summary

Critical infrastructure organisations share a need for reliable OT cybersecurity, but they do not share identical operational realities. An airport, power utility, water treatment plant, refinery, railway network and manufacturing facility may all use industrial control systems—yet their critical assets, operating constraints, communication patterns, safety consequences and adversary exposure can differ substantially.

A generic OT security tool can provide useful asset discovery and protocol visibility. Its effectiveness can decline, however, when it treats every environment as if the same assets, behaviours and threat scenarios matter equally. A detection that is meaningful in one sector may be routine in another; an apparently minor deviation in a specialised control process may be highly significant in its local context.

This case study explores the operational case for a highly customisable OT security capability: one that allows defenders to adapt asset models, protocol parsing, detection logic, risk weighting, threat scenarios, alert thresholds and integrations to the needs of a specific critical infrastructure (CII) use case. The central proposition is simple: a common platform can provide consistency, but its detection and response logic must be configurable to reflect the environment it protects.

This is an illustrative design case study, not a claim about a named customer deployment or measured product performance. The scenarios and expected outcomes below are analytical examples intended to show how customisation can improve relevance when implemented, validated and governed appropriately.

"A common security platform may scale across sectors. Effective OT protection depends on whether it can be configured to understand what is normal, abnormal and consequential for each asset."

1. The Standardisation Paradox

Organisations often seek standardisation to simplify deployment, centralise monitoring and compare risk across sites. Those are legitimate objectives. The challenge appears when standardisation becomes rigidity: the tool ships with fixed assumptions about assets, protocols, expected traffic, severity and alert logic, and local teams have limited ability to adapt those assumptions.

What standardisation helps with	Where a fixed model can fall short
Consistent inventory and telemetry collection	May not capture the operational role or consequence of a particular asset
Common alert format and reporting	May rank technically similar events identically despite different process impacts
Reusable baseline detections	May mistake legitimate maintenance or engineering activity for malicious behaviour—or miss a sector-specific deviation
Centralised dashboards	May hide local context behind generic severity scores
Repeatable deployment	May not accommodate legacy devices, unusual protocols, site-specific topology or strict change windows

The goal is not to abandon standardisation. It is to standardise the platform's foundations while allowing controlled customisation of the detection and operational context layers.

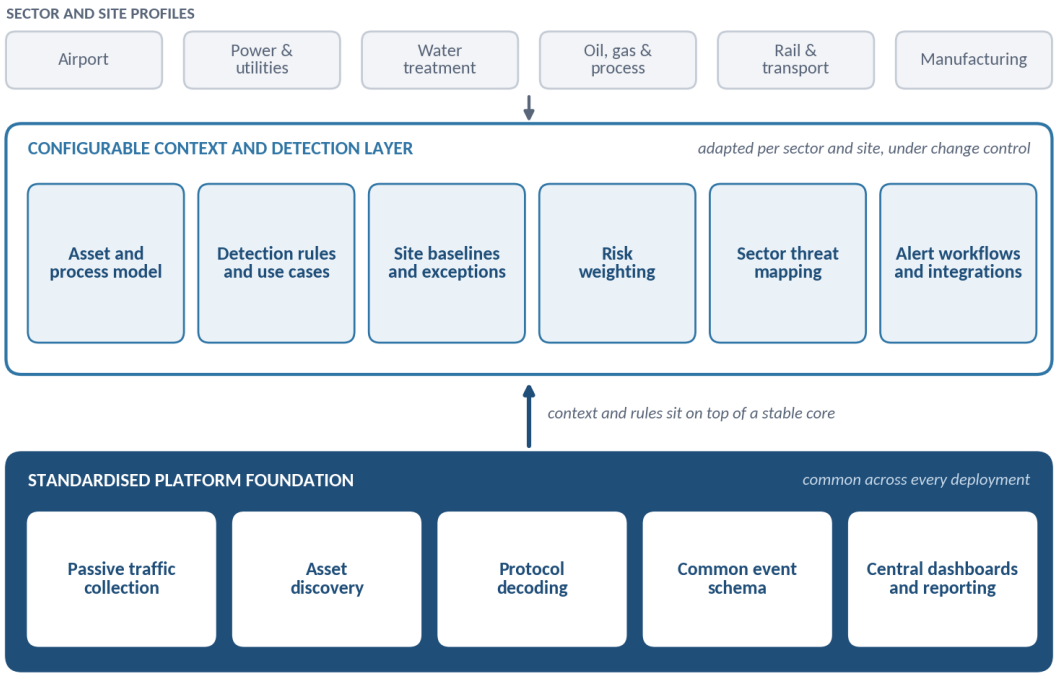


Figure 1. Standardised foundation, configurable context layer: the platform core stays common while sector and site context is configured on top.

2. Critical Infrastructure Is Not One Threat Environment

Sector and site context changes what defenders should prioritise. The following examples are illustrative; the actual technology and threat profile must be validated for each facility.

CII sector / use case	Potentially distinctive assets and dependencies	Examples of context-specific security questions
Airport operations	Baggage handling PLCs, building management, airfield lighting, passenger-processing dependencies	Is a controller communicating with an unexpected engineering station? Could an alert affect baggage throughput or airfield operations?
Electricity / utilities	Protection relays, RTUs, substation automation, SCADA links	Is a control command expected for this operating state? Is the source authorised for this substation and protocol?
Water treatment	Chemical dosing, pumps, pH instrumentation, treatment SCADA	Does a setpoint change fit the approved operating envelope and current work order?
Oil, gas and process industry	DCS, SIS, compressors, valves, process historians	Could the event affect process safety, production continuity or a safety instrumented function?
Rail and transport	Signalling interfaces, traction power, station systems, control centres	Is the communication permitted for this signalling zone and operational mode?
Manufacturing	Robots, CNC equipment, PLC lines, industrial gateways	Is this firmware, logic download or machine-state transition expected for this production cell?

A sector label alone is not enough. Two airports—or two plants in the same sector—can differ in architecture, vendor mix, operating schedules, remote-access arrangements, redundancy and tolerance for disruption. Effective configuration must therefore work at both sector and site level.

3. The Case for a Customisable OT Security Tool

Customisation should be more than changing dashboard colours or setting a generic alert threshold. It should allow a security team to shape how the platform models the environment, recognises relevant behaviours, interprets risk and delivers evidence to the people responsible for the process.

Customisation capability	Why it matters	Example
Asset and process model	Connects device identity to function, zone, owner, dependencies and consequence	Distinguish a baggage-line PLC from a non-critical display controller
Protocol and vendor support	Supports diverse OT stacks, variants and vendor-specific behaviours	Decode the relevant S7, Modbus, OPC UA or BACnet traffic where present
Detection logic and rules	Allows rules to reflect local normal behaviour and threat scenarios	Alert on a PLC logic download outside an approved maintenance window
Site-specific baselining	Accounts for legitimate cycles, shifts, seasonal demand and maintenance	Treat a scheduled vendor session differently from an unknown session at an unusual time
Risk weighting	Combines technical signal with reachability, criticality and process impact	Escalate a low-volume event on a safety-critical asset above a noisier event on a redundant monitor
Threat intelligence mapping	Maps sector-relevant adversary tactics to assets and exposed pathways	Prioritise remote-access and engineering-workstation pathways relevant to the site
Deception and custom use cases	Enables decoys and breadcrumbs to reflect the protected environment	Place a clearly governed decoy representing a likely engineering service or OT asset class
Workflow and integrations	Routes actionable context to SOC, engineering, operations and incident response	Send enriched alerts to SIEM/SOAR with asset role, evidence and recommended validation

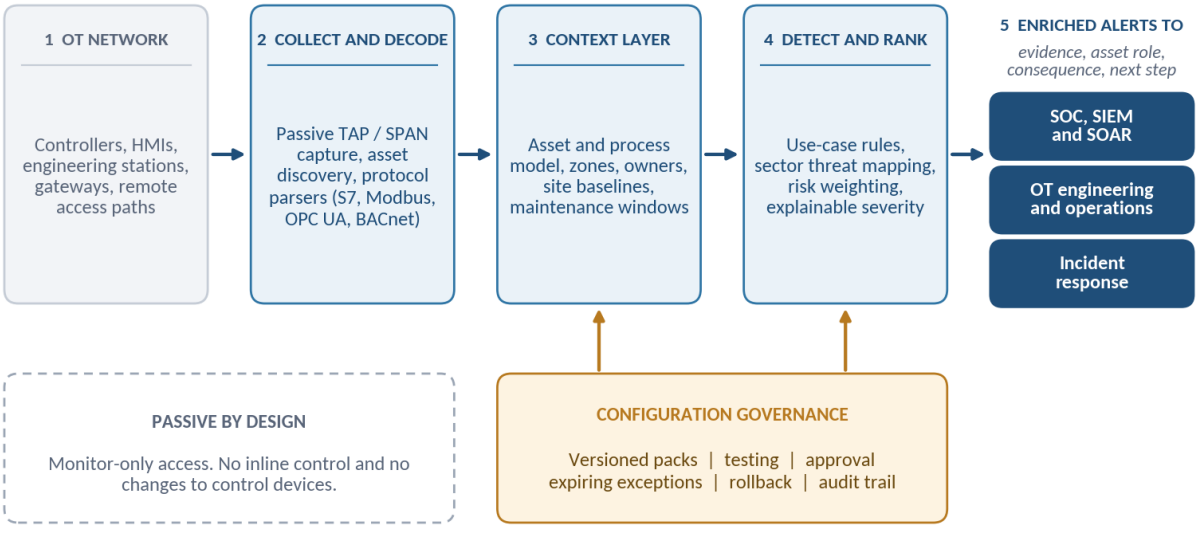


Figure 2. Reference architecture for a customisable OT security platform, from passive collection to enriched alerts, under configuration governance.

The platform should make these capabilities configurable without requiring every customer to rebuild the product or write unreviewed code. Configuration should be versioned, testable, reversible and subject to change control.

4. Illustrative Scenario: One Platform, Three Different Contexts

Consider the same technical signal: a controller receives a write command from an engineering workstation. A generic rule may flag the command based on protocol function alone. A context-aware and customisable tool can interpret the event against approved sources, asset role, time window, operating state and process consequence.

Environment	Context supplied to the tool	More useful detection outcome
Airport baggage handling	PLC identity, line dependency, approved vendor endpoints, maintenance schedule, impact on baggage flow	Prioritise an otherwise valid write command when it originates from an unapproved host or outside the approved window
Water treatment	Controller role, approved setpoint range, process state, dosing safeguards, work order	Correlate an unusual setpoint write with process limits and source identity; alert at the appropriate urgency
Manufacturing cell	Machine recipe, production shift, authorised engineering station, change ticket	Distinguish an approved recipe change from an unexplained logic download or out-of-pattern write

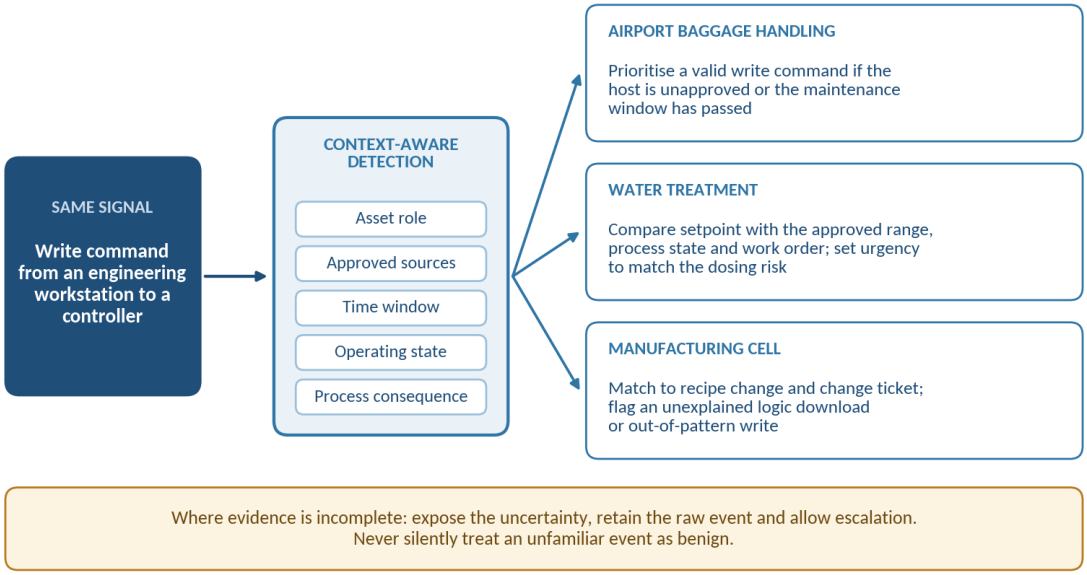


Figure 3. One technical signal, three operational meanings: context changes how the same write command is interpreted and prioritised.

The signal is technically similar; the operational meaning and response are not. Configuration allows the platform to preserve a common detection foundation while tailoring its interpretation and prioritisation to each CII environment.

This does not mean context should suppress alerts automatically. Where evidence is incomplete, the tool should expose uncertainty, retain the raw event and allow escalation rather than silently treating an unfamiliar event as benign.

5. From Generic Alerting to Use-Case-Led Detection

A practical implementation begins with defined questions—not with an unrestricted list of features. Each use case should specify the asset or process in scope, the behaviour of interest, the relevant telemetry, the expected baseline, the threat hypothesis, the consequence and the response owner.

Use-case element	Questions to define before enabling a detection

Operational objective	What process, service or safety function are we protecting?
Asset scope	Which devices, zones, communication paths and dependencies are in scope?
Threat hypothesis	What plausible misuse, intrusion path or unsafe change are we trying to detect?
Required evidence	Which packets, logs, identity events, configuration changes or process signals support the detection?
Baseline and exceptions	What legitimate behaviour is expected, and who approves exceptions?
Severity and response	Who must act, within what time, and what action is safe for operations?
Validation	How will the rule be tested against representative benign and malicious scenarios?
Governance	Who owns the rule, reviews drift, approves updates and can roll back a change?

This use-case-led approach reduces the risk of deploying broad rules that generate large volumes of low-value alerts while missing behaviours unique to the environment.

6. Customisation Must Be Governed

Flexibility is valuable only when it remains controlled. Excessive or poorly documented customisation can fragment deployments, weaken detection consistency and create rules that nobody understands or maintains.

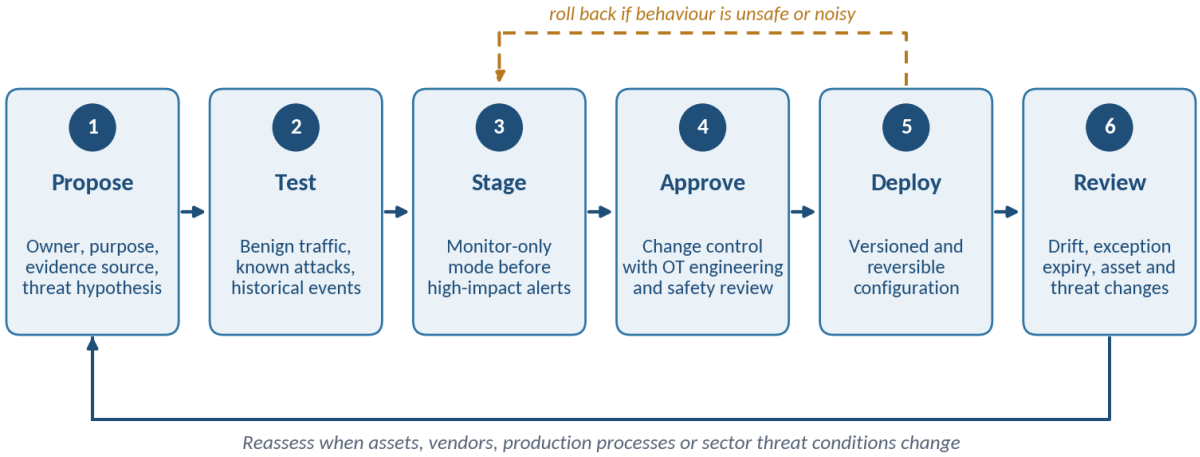


Figure 4. Governed configuration lifecycle: propose, test, stage, approve, deploy and review, with rollback and periodic reassessment.

- Use approved templates and a common core of baseline detections, then add sector- and site-specific rules.
- Version every configuration and record the owner, purpose, evidence source, approval and change history.
- Test rule changes against representative benign traffic, known attack scenarios and historical events where available.
- Use staged deployment or monitor-only mode before enabling high-impact alerting or automated response.
- Define expiry and review dates for exceptions; avoid permanent allowlists without justification.
- Keep raw telemetry and an auditable explanation of why an event was classified, prioritised or suppressed.
- Ensure response recommendations are OT-safe. Do not automatically isolate or alter control devices without an approved operational safety process.
- Reassess configurations when assets, network paths, vendors, production processes or sector threat conditions change.

7. How to Measure Whether Customisation Works

The value of customisation should be demonstrated through validation, not assumed from the number of available settings. Baseline and customised configurations should be compared using the same defined test scenarios and comparable data.

Measure	What it indicates	Evaluation approach
Detection precision	How many alerts correspond to validated events of interest	Review alert outcomes against labelled test cases or analyst adjudication
False-positive rate	How often benign activity triggers the detection	Run approved maintenance, normal production and routine traffic scenarios
Detection coverage	Which defined threat scenarios are detected and which remain uncovered	Map tested cases to the use-case register and document gaps
Time to detect / triage	How quickly an event is identified and understood	Measure event-to-alert and alert-to-triage intervals under comparable conditions
Context completeness	Whether alerts include asset role, zone, dependencies and operational consequence	Audit a sample of alerts against required context fields
Rule maintainability	Whether configurations can be understood, tested and safely updated	Review ownership, documentation, version history and rollback readiness
Operational safety	Whether monitoring and response avoid unacceptable process disruption	Validate passive monitoring and response actions with OT engineering and safety teams

No numerical improvement is claimed in this illustrative case study. Organisations should establish their own baseline, define acceptance criteria and publish results only after repeatable testing or measured deployment.

8. A Practical Adoption Roadmap

Phase	Activity	Deliverable
1. Discover	Inventory assets, protocols, zones, owners and critical dependencies	Validated asset and process context
2. Select	Choose a small number of high-value, sector-relevant use cases	Prioritised use-case register
3. Configure	Adapt parsers, baselines, rules, context fields and workflows	Versioned configuration pack
4. Validate	Test benign, suspicious and approved adversarial scenarios in a safe environment	Evidence-backed test report and known limitations
5. Operationalise	Integrate alerts with SOC, OT engineering and incident response	Defined triage and escalation workflow
6. Improve	Review outcomes, changes in assets and evolving threat scenarios	Controlled tuning and periodic revalidation

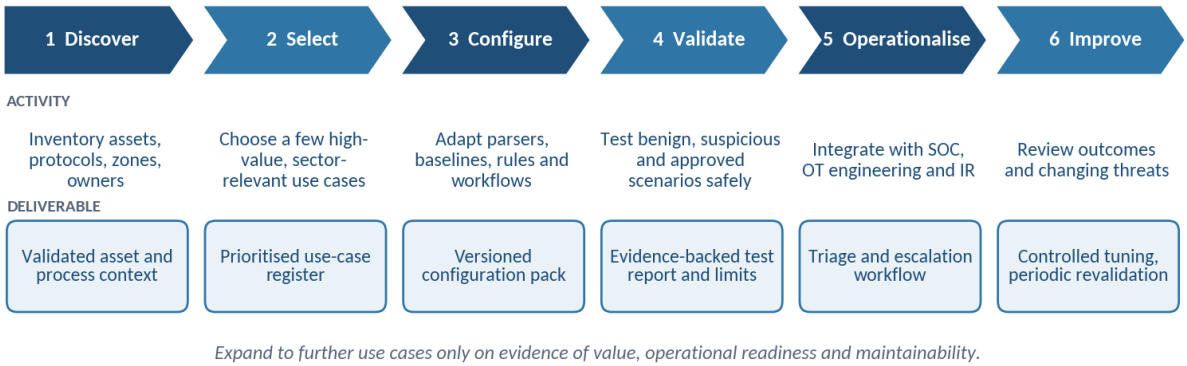


Figure 5. Adoption roadmap: six phases from discovery to continuous improvement, each with a defined deliverable.

A small number of well-validated use cases is usually a more defensible starting point than enabling every available rule. Expansion should follow evidence of value, operational readiness and maintainability.

9. What Buyers Should Ask Their OT Security Provider

- Can we model our own asset roles, process dependencies, zones and consequence levels?
- Can detections be configured for our sector, site, vendor mix and operational schedule?
- Which protocol variants and vendor-specific behaviours are actually supported, and how is that support validated?
- Can our team build, test, version, approve and roll back a custom use case without a product-code change?
- Can alerts explain the evidence and context behind their priority instead of exposing only a score?
- How are false positives, exceptions and changes in normal operating behaviour managed?
- Can the platform integrate with our SOC while preserving OT engineering authority over operational actions?
- What evidence can the provider share for detection coverage, false-positive performance and safe deployment?

Conclusion

Critical infrastructure is diverse by design. The same protocol can serve different processes; the same alert can carry different consequences; and the same sector can contain sites with materially different architectures and threat exposures. A one-size-fits-all OT security model risks flattening those differences into generic detections and scores.

The stronger approach is a consistent platform with meaningful, governed customisation: sector-aware threat scenarios, site-specific asset and process context, adaptable detection logic, explainable prioritisation and workflows aligned with the people who operate the infrastructure.

For CII operators, the key question is therefore not simply whether an OT security tool can see devices or recognise protocols. It is whether the tool can be configured—and safely maintained—to understand the specific environment, detect the behaviours that matter there, and present evidence that supports timely operational decisions.

A position, not a product claim

This case study presents a general design perspective on customisable OT security. The scenarios are illustrative, and the benefits described are expected outcomes to validate rather than measured results from a named deployment. Actual effectiveness depends on data quality, architecture, configuration, testing, governance and operational adoption.