

WHITE PAPER

What Open Process Automation Lets You Do That Your DCS Can't

Eight capabilities an O-PAS architecture enables or materially improves, from orchestration and embedded cybersecurity to advanced process control

Trevor Cusworth

Collaborative Systems Integration (CSI)
October 2026

Contents

Executive summary.....	3
1. Beyond vendor independence.....	4
2. A short primer on the O-PAS architecture.....	5
3. The eight capabilities at a glance.....	7
4. Orchestration and system management.....	8
5. Cybersecurity embedded, not bolted on.....	10
6. Advanced process control that runs where it works best.....	12
7. Control logic the operator owns and can move.....	14
8. Hardware refresh without rewriting the application.....	15
9. Data that carries its meaning.....	16
10. Continuous improvement instead of big-bang upgrades.....	17
11. Migration by coexistence and late binding.....	18
12. Organisational implications.....	19
13. Preparing for an open architecture.....	20
14. Conclusion.....	21
About the author and next steps.....	22

Executive summary

Most discussion of Open Process Automation (OPA) concentrates on vendor independence. That is the procurement argument, and it is an important one, but it undersells what the O-PAS standard does. An open architecture changes what a control system can do, not only who supplies it.

Traditional distributed control systems fix three things at the point of purchase: where software runs, how the system is secured and how the system can change. Those constraints shape every decision for the next 15 to 25 years. O-PAS removes them by separating applications from hardware, standardising how components communicate and are managed, and making security and conformance properties of each component rather than of the platform as a whole.

This paper sets out eight capabilities that follow from that structure:

1. **Orchestration and system management** built into the architecture, so every node is monitored, configured and updated in one consistent way.
2. **Cybersecurity embedded in every component**, inherited by every conformance profile, rather than bolted on around a closed platform.
3. **Advanced process control that runs where it works best**, with execution location chosen by the engineer rather than by the platform.
4. **Control logic the operator owns and can move**, expressed as standard function blocks.
5. **Hardware refresh without rewriting the application**, so end-of-life notices stop triggering capital projects.
6. **Data that carries its meaning**, giving analytics, optimisation and AI tools consistent, contextualised information.
7. **Continuous improvement** through small, tested, reversible changes instead of big-bang upgrades.
8. **Migration by coexistence and late binding**, so modernisation happens unit by unit and project risk moves out of commissioning.

Each capability is described in terms of the constraint it removes, what O-PAS changes, what it looks like in day-to-day operation and what it means for the people making control system decisions. The paper closes with the organisational implications and a set of questions to bring to suppliers and integrators.

1. Beyond vendor independence

Industrial control systems have been engineered for one overriding priority: stability. They have delivered it. The same tight integration that made them reliable also made them hard to change. Hardware, software, engineering tools and communications were designed together by one supplier and sold as one product, so changing any part of the system meant engaging with all of it.

The consequences are familiar to anyone who has run a plant. Control logic accumulates in proprietary formats. Hardware is kept running long after it should be retired because replacing it would force a revalidation of everything above it. Advanced control and analytics are added through custom interfaces that someone has to maintain. Security is layered around a platform that was designed for a closed network. Improvements are deferred to the next major upgrade, and every major upgrade becomes a high-risk capital project.

These are not failures of any particular vendor. They are properties of a platform-defined architecture. The platform determines what the system is, and the operator configures it. Open Process Automation inverts that relationship. The standard defines what the system must be, and the operator composes it from components that conform to that definition.

The O-PAS standard is developed by the Open Process Automation Forum under The Open Group, with ExxonMobil as a founding catalyst and participation from end users, system integrators and suppliers across the industry. It addresses communication, information, execution, security and system management as parts of one architecture. The result is not simply a more open way of buying the same system. It is a different kind of system, with capabilities that a closed platform cannot provide.

The central idea

In a traditional DCS, the platform decides where software runs, how it is secured and how it changes. In an O-PAS system those become engineering decisions made by the operator, inside a standard that keeps the whole system coherent.

2. A short primer on the O-PAS architecture

The capabilities in this paper depend on a small number of architectural building blocks. Readers familiar with O-PAS can skip to Section 3.

2.1 Distributed Control Nodes

The Distributed Control Node (DCN) is the primary execution element in an O-PAS system. A DCN is a secure, manageable component that can host control applications, connect to field I/O or act as a gateway to equipment that is not conformant. It is defined by the interfaces it exposes, not by its form factor: a small device with a few I/O channels beside a process unit and a rack-mounted server are both DCNs if they implement the required interfaces.

Inside every DCN are two layers. The Physical Platform, defined in Part 7 of the standard, is the hardware and its base system software. Above it runs a standardised software framework that gives applications the same set of services whatever hardware sits underneath. That separation is the root of portability.

2.2 Advanced Computing Platforms and gateway DCNs

An Advanced Computing Platform (ACP) is a DCN with enough computing resource to host heavier applications such as historians, advanced control, optimisation and visualisation. It typically sits in a server room or operations data centre and participates in the architecture on the same basis as any other DCN.

A gateway DCN exposes the data and functions of existing equipment, such as a legacy DCS, PLCs or intelligent field devices, through conformant interfaces. The legacy equipment is not modified. This is the mechanism that makes migration through coexistence practical.

2.3 The O-PAS Connectivity Framework

Components communicate through the O-PAS Connectivity Framework (OCF), defined in Part 4 and built on OPC UA. The OCF specifies the transport, the structure of the data exchanged, the security that applies to every exchange and the quality-of-service properties distributed control requires. OPC UA supports both client-server and publish-subscribe interaction and carries metadata alongside the data itself. In diagrams the OCF is drawn as one fabric; in a real plant it is segmented networks, virtual LANs and firewall boundaries arranged for security and performance.

2.4 The information model and function blocks

The information model defines what exchanged data means: the structure of signals, function blocks and system relationships. Control logic is expressed as function blocks, and the standard accommodates both the cyclic model of IEC 61131-3 and the event-driven model of IEC 61499. Configuration is exchanged in standard formats that any conformant tool can produce and any conformant execution engine can consume.

2.5 System management, profiles and conformance

System management, defined in Part 5 and built on the DMTF Redfish standard, provides monitoring, configuration control, update coordination and orchestration across components. Security requirements, defined in Part 2, are based on the ISA/IEC 62443 series. Components participate in the architecture by conforming to profiles, defined in Part 3, which group the requirements for a particular role. The Open Group certification programme verifies conformance independently, so compatibility becomes a tested property rather than a supplier assertion.

Element	Role in the architecture	Capabilities it underpins
Distributed Control Node (DCN)	Hosts control applications, connects I/O, or acts as a gateway	APC placement, hardware refresh, migration
Physical Platform + software framework	Separates hardware from the services applications use	Portability, hardware refresh
Advanced Computing Platform (ACP)	A DCN with the compute for historians, APC and optimisation	Advanced process control, analytics
O-PAS Connectivity Framework (OCF)	Standard, secured communication built on OPC UA	Embedded security, data integration
Information model	Common meaning for signals, function blocks and relationships	Data context, portable logic
System management (Part 5)	Monitoring, configuration, updates and orchestration	Orchestration, continuous improvement
Security (Part 2)	Requirements based on ISA/IEC 62443, inherited by every profile	Embedded cybersecurity
Profiles and certification	Testable requirements for each component role	Multi-vendor composition with confidence

Table 1. O-PAS building blocks and the capabilities they support.

3. The eight capabilities at a glance

Table 2 summarises how each capability compares with a traditional platform-based DCS. Sections 4 to 11 describe each one in more detail.

Capability	Traditional DCS	With O-PAS
Orchestration and system management	Vendor tools, one per platform, with gaps filled by spreadsheets and tribal knowledge	Standard management interfaces exposed by every component, giving one way to monitor, configure, update and deploy across the whole system
Cybersecurity	Added around the platform after design: firewalls, DMZs, whitelisting and compensating controls	Requirements based on ISA/IEC 62443 inherited by every profile, verified in certification and applied to every exchange
Advanced process control	Runs where the platform allows, through custom interfaces that age with the DCS	Runs where it performs best, from edge DCNs to ACPs, coordinated through standard interfaces
Control logic ownership	Locked in proprietary formats readable only by one vendor's tools	Standard function blocks (IEC 61131-3 and IEC 61499) that the operator owns, versions and reuses
Hardware refresh	Hardware and application age together; refresh forces a rewrite and revalidation	Application stays valid on new hardware; refresh is driven by cost, performance and availability
Data with meaning	Point-by-point mapping for every historian, analytics tool and optimiser	Shared information model in which units, scaling, validity and context travel with the signal
Change management	Improvements deferred to major upgrades every 10 to 20 years	Small, tested, reversible changes deployed continuously
Migration and project execution	Rip and replace, sequential projects and late integration	Coexistence through gateway DCNs, parallel engineering and late binding of hardware decisions

Table 2. Traditional DCS compared with an O-PAS architecture.

4. Orchestration and system management

The constraint today

In a traditional DCS, system management means the vendor's tools. That works for the vendor's own equipment, but no plant runs on one supplier alone. Every third-party controller, server, network switch and analyser brings its own management console. Engineers end up with one tool for the controllers, another for the servers, a spreadsheet tracking firmware versions and a great deal of knowledge held in a few people's heads.

The result is poor visibility and slow, cautious change. Nobody can easily answer basic questions such as which nodes are running which firmware, which applications are deployed where, or what will be affected by a particular update.

What O-PAS changes

O-PAS treats system management as part of the architecture, not as a product layered on afterwards. Part 5 of the standard defines management interfaces built on DMTF Redfish, a standard developed for managing data centre infrastructure at scale and adapted here for industrial use. Every conformant component must expose these interfaces as part of its profile.

That requirement is what makes a multi-vendor system operable. A composable architecture in which each component needed a different management approach would not scale. Because management is uniform, an operator can:

- **Monitor** the state and health of every node through one mechanism, whoever supplied it.
- **Control configuration** consistently, with a single record of what is deployed where.
- **Coordinate updates** across components, staging and tracking firmware and software changes across a mixed fleet.
- **Orchestrate** the deployment of applications to nodes, and move them when load, hardware or operating needs change.
- **Allocate resources** as the system evolves rather than fixing them at commissioning.

What it looks like in operation

At shift change, the control systems engineer opens one view of the whole system: every DCN, its health, its firmware, the applications it hosts and any pending updates. A security patch for one family of nodes is scheduled, deployed to a test node, verified and then rolled out in stages, with the status of every node visible throughout. When an ACP approaches its compute limit, an optimisation application is redeployed to a second ACP without touching the regulatory control below it.

Why it matters

Orchestration is the capability that makes the rest possible. Portability, continuous improvement and targeted maintenance all assume that the operator can see and control the whole system consistently.

Without standard system management, a distributed, multi-vendor system would become harder to run than the monolith it replaced. With it, distribution becomes an advantage.

5. Cybersecurity embedded, not bolted on

The constraint today

Most installed control systems were designed for isolated networks. As connectivity increased, security was added as layers wrapped around them: firewalls, demilitarised zones, application whitelisting, patching agents, remote access gateways and a compensating control for every gap the original design never anticipated. Each layer is a separate project with its own supplier, and the platform underneath was never designed with any of them in mind.

Bolted-on security has predictable weaknesses. Protection is concentrated at the perimeter. Patching is slow because changing one component risks the whole platform. Security properties vary from one device to the next, and evidence that a component is secure often amounts to the supplier's word.

What O-PAS changes

O-PAS starts from the other end. Part 2 of the standard defines security requirements based on the ISA/IEC 62443 series, the international framework for industrial automation and control system security. Those requirements are not a separate add-on. Every profile inherits them, so a component cannot be conformant to its profile without also meeting the corresponding security requirements.

Three properties follow:

- **Security is a property of every component.** It is designed in by the supplier and verified through certification, rather than added by the integrator at the end of the project.
- **Every exchange is secured.** The OCF applies security to communication between components, drawing on the security mechanisms OPC UA already provides for authenticating applications and protecting the messages they exchange.
- **Security is maintainable.** Because components are managed through standard interfaces and can be updated individually, a patch on one node does not force a revalidation of the whole system. Shorter patch cycles become realistic.

What it looks like in operation

When a new controller is procured, its security capabilities are part of its certified profile, alongside its functional capabilities. The security architect designs zones and conduits for the site knowing what every component will support. When a vulnerability is disclosed in one supplier's node, that node is patched through the standard management interfaces on a planned schedule, and the rest of the system is untouched.

What embedded security does not replace

Embedded security is the starting point, not the whole answer. A real OCF is still built from segmented networks, virtual LANs and firewall boundaries arranged for the site. Defence in depth, monitoring, incident response, access management and trained people remain essential. The difference is that the

security programme begins with components that were designed to be secured, instead of trying to secure components that were not.

For the CISO and the plant manager

Embedded security changes the procurement conversation. Security requirements can be written into specifications by reference to the standard, and evidence of conformance comes from certification rather than from supplier questionnaires.

6. Advanced process control that runs where it works best

The constraint today

In a platform-based system, advanced control is implemented inside the boundaries of the platform. Model predictive control (MPC), optimisation and other advanced strategies run where the vendor's architecture allows. Frequently that means a separate server connected to the DCS through custom interfaces that someone has to build, test and maintain.

Three problems follow. **Execution location is fixed:** if the available environment lacks computing headroom or the right timing characteristics, the controller is limited by the box rather than by the process. **Integration is fragile:** advanced control has to interact with regulatory loops, process data and external systems, and every non-standard interface adds engineering and risk. **Evolution is slow:** when the DCS is upgraded, the APC interfaces are retested or rebuilt, so improvements wait for the next project.

What O-PAS changes

O-PAS decouples control logic from infrastructure, which makes execution location a design decision. Engineers decide where each layer of control should run based on performance, latency, resilience and computing need. A typical arrangement uses three layers that interact through standard interfaces and share one information model:

Layer	Where it runs	What it does
Regulatory and high-speed control	DCNs close to the process	Fast, deterministic loops such as combustion control and high-speed regulation
Supervisory and advanced control	ACPs or larger DCNs	MPC coordinating several loops or units, adjusting setpoints within constraints
Optimisation and analytics	ACPs in the operations data centre	Economic optimisation across units, drawing on data from across the system

Table 3. Distributing control across an O-PAS architecture.

Because the layers share a common information model and communicate through the OCF, the plumbing between them stops being a custom project. MPC reads and writes regulatory loops through the same interfaces any other application uses. Optimisation draws on consistent data from across the plant.

Four improvements

- **Placement.** Each layer runs where it performs best, not where the platform has space. High-frequency loops sit at the edge for determinism; computationally heavy controllers get the compute they need.

- **Scale.** Computing capacity for advanced control can be added by adding ACP capacity, without disturbing the control layers below.
- **Portability.** An APC application built to the standard can move to new hardware, or be reused on a similar unit at another site, without redevelopment. Proven controller designs become reusable assets.
- **Refinement.** Controller models can be updated and redeployed on their own schedule, so tuning and model maintenance become routine rather than deferred to the next turnaround.

What it looks like in operation

A refinery deploys an MPC application for a crude unit on an ACP. Regulatory control runs on DCNs beside the unit. When the crude slate changes, the APC engineer updates the controller model, validates it in a test environment against recorded plant data and deploys it without any change to the regulatory layer. Two years later, the same controller design is reused on a sister unit at another site, running on hardware from a different supplier.

What it means for decision makers

The value of OPA is usually expressed in lifecycle cost. Advanced control shows that it extends to how the plant performs. When advanced control is easier to deploy, cheaper to maintain and quicker to improve, more of the plant runs closer to its constraints for more of the time. For suppliers of advanced control, the opportunity is to deliver their products as conformant O-PAS applications that can run on any conformant platform.

7. Control logic the operator owns and can move

The constraint today

A plant's control strategies represent decades of process knowledge. In a traditional DCS that knowledge sits in proprietary configuration formats that only one vendor's tools can open. The logic is the operator's intellectual property in principle, but in practice it is rented from the platform it was written on. Every migration starts with paying someone to translate it, and every translation carries the risk of subtle errors.

What O-PAS changes

O-PAS expresses control logic as standard function blocks, each encapsulating behaviour, inputs, outputs, internal state and parameters in a consistent, reusable form. The standard accommodates both major traditions: the cyclic execution model of IEC 61131-3 and the event-driven model of IEC 61499. Different processes have different timing needs, and the standard does not force one programming model on all of them.

Configuration is packaged in standard formats. A conformant engineering tool produces it, and any conformant execution engine consumes it, with neither needing to know who supplied the other. Portability is achieved not by translating applications between platforms but by defining them in a way that does not depend on any single platform.

What it looks like in operation

- Control strategies are developed, tested and validated independently of the hardware they will eventually run on.
- Engineering tools and runtime environments are procured separately, each on its own merits.
- Logic is kept under version control, with a clear history of what changed, when and why.
- Proven strategies are packaged as libraries and reused across units and sites.

What it means for decision makers

Owning the logic changes the balance of every future negotiation. When control strategies can move, the operator is no longer locked into a supplier by the cost of translating its own intellectual property. It also changes how engineering is organised: software practices such as version control, structured testing and controlled deployment become part of control engineering.

8. Hardware refresh without rewriting the application

The constraint today

In a traditional DCS, hardware and software age together. When controllers or I/O reach end of life, the upgrade drags the application with it, because the logic is bound to the platform. Plants respond rationally by keeping obsolete hardware running well past its sensible life, buying spares on the secondary market and deferring the inevitable. When the refresh finally happens, it is a full capital project with a full revalidation.

What O-PAS changes

Inside every DCN, O-PAS separates the Physical Platform from the standard software framework above it. Applications interact with the framework, never with the hardware directly. Scheduling, resource management and access to signals are all mediated by the framework.

The framework is, in effect, a contract. As long as the framework on the target node provides what the application expects, the application can be deployed on that node without modification. The hardware can be refreshed, the supplier can change and the form factor can vary from a small embedded device to a rack-mounted compute platform, and the application remains valid.

What it looks like in operation

An end-of-life notice arrives for a family of controllers. Instead of opening a capital project, the reliability team plans a rolling replacement over two turnarounds. New DCNs, possibly from a different supplier, are installed, the existing applications are redeployed to them through the standard management interfaces and the old hardware is retired. The control strategies are not rewritten.

What it means for decision makers

Hardware becomes a resource managed on cost, performance and availability rather than a constraint to be preserved. Capital planning becomes smoother, obsolescence risk falls and the operator can take advantage of improvements in computing hardware on its own schedule, not the vendor's.

9. Data that carries its meaning

The constraint today

Any two systems can pass a temperature value across a network. The difficulty is that each end may have its own interpretation of the units, the scaling, whether the value is valid and which piece of equipment it belongs to. Connectivity is necessary for interoperability, but it is not sufficient.

This is why every historian, analytics platform, optimiser and reporting tool in a traditional plant needs its own point-by-point mapping exercise. Those mappings are expensive to build, they break quietly after changes and they are one of the main reasons analytics projects stall between pilot and production.

What O-PAS changes

The O-PAS information model makes meaning part of the data. The structure of signals, function blocks and the relationships between system elements is standardised, so units, scaling, validity and equipment context travel with each signal. The receiving component does not need to know which system produced a value in order to interpret it correctly.

What it looks like in operation

- A new analytics application subscribes to the data it needs and understands it immediately, without a bespoke tag-mapping project.
- Optimisation routines incorporate real-time information from across units on a consistent basis.
- Insights from analytics can be fed back into control strategies through the same standard interfaces, under the same security and management rules.

What it means for decision makers

Contextualised data is the foundation for every ambition built on plant data, including machine learning and AI. In a standards-based architecture, the cost of adding the next analytics use case falls, and the boundary between control and analytics becomes something to design deliberately rather than a wall to engineer through.

10. Continuous improvement instead of big-bang upgrades

The constraint today

Traditional control systems are operated to preserve stability, which in practice means minimising change. Improvements accumulate until the next major upgrade, and that upgrade concentrates years of change, and all of its risk, into one outage window. The system is at its best on the day it is commissioned and slowly falls behind operational needs from then on.

What O-PAS changes

Because applications and infrastructure are independent assets, each can change on its own. The principles of continuous integration extend from the project into operations:

1. A change, whether a new application, an update or a configuration adjustment, is developed and placed under version control.
2. It is tested in a controlled environment that reflects operating conditions.
3. It is deployed through the standard management interfaces to the nodes that need it, often in stages.
4. Its effect is monitored, and it is rolled back if it does not behave as intended.

Maintenance becomes targeted. A single component can be updated or replaced without system-wide downtime, and issues are isolated and addressed where they occur.

A different risk profile

Risk is not eliminated. It is distributed across many small, well-understood changes instead of being concentrated in one large event. That reduces the likelihood of major disruption, but it depends on discipline: configuration control, realistic testing, monitoring that detects problems quickly and a governance process that prevents the system from drifting away from its defined architecture as changes accumulate.

What it means for decision makers

The benefits compound. Over a 20-year life, the gap between a system that is frozen at commissioning and one that is refined continuously becomes large, both in operational performance and in the cost of keeping the system aligned with the business. The role of operations shifts from preserving stability to managing change safely.

11. Migration by coexistence and late binding

The constraint today

The traditional modernisation choice is rip and replace: a large capital project, a long engineering phase, a cutover plan that keeps operations leadership awake for months and integration testing compressed into the final weeks before start-up. Projects run sequentially, because application engineering cannot start in earnest until the hardware platform is chosen.

Coexistence

O-PAS provides the gateway DCN, which exposes an existing DCS, PLC or intelligent field device through conformant interfaces without modifying it. The legacy installed base keeps running while new O-PAS components grow around it. Control functions move across one unit or one area at a time, and the legacy system is progressively encapsulated and absorbed rather than replaced in a single event.

Parallel execution and late binding

Because applications are developed against the standard rather than a chosen hardware platform, project work can run in parallel. Application engineering proceeds alongside hardware selection. Integration is validated continuously from early in the project instead of at the end. Hardware decisions are deferred until more is known, a practice known as late binding, so they can be made with better information and closer to the point of use.

Project dimension	Traditional sequence	O-PAS approach
Hardware decision	Made first; constrains everything after it	Deferred until needed and better informed
Application engineering	Waits for the platform	Starts early, independent of hardware
Integration	A late, discrete phase	Continuous throughout the project
Cutover	One large event	Unit by unit, through coexistence
Where risk concentrates	Commissioning and start-up	Spread across the lifecycle

Table 4. How project execution changes in an O-PAS architecture.

What it means for decision makers

Modernisation becomes a controlled programme rather than a single disruptive bet. Capital can be phased, operations can learn as they go and risk moves out of commissioning, where it is most expensive, into earlier stages where it is cheaper to address.

12. Organisational implications

These capabilities do not arrive through technology alone. They depend on the organisation that defines, integrates and sustains the system.

Architecture leadership

When the system is composed rather than delivered, someone must own the architecture: defining it, keeping components aligned and preserving coherence as the system changes. Without that function, a multi-vendor system fragments into components that each work but do not form a whole.

Integration as a continuing capability

In a traditional system, integration is largely internal to the vendor. In OPA, integration is the system. Interfaces must be validated, dependencies managed and behaviour understood across boundaries, and that work continues into operations. System integrators take on a central role in assembling and maintaining coherent systems.

IT and OT working together

The technologies that enable OPA, including virtualisation, distributed computing, orchestration and standard interfaces, come from IT. The requirements, including determinism, reliability and safety, come from OT. Success depends on coordinating the two disciplines through shared frameworks and processes, not on merging them.

Skills

Engineers move beyond platform-specific expertise towards an understanding of architecture, interfaces and system behaviour across the full stack. Software practices such as version control, structured testing and controlled deployment join established control engineering disciplines. The aim is to extend existing expertise, not to replace it.

Governance

OPA introduces flexibility, and governance keeps that flexibility under control. Clear rules for how changes are introduced, how interfaces are managed and how conformance is maintained over time prevent drift: the gradual divergence of a system from its architectural definition.

13. Preparing for an open architecture

Each capability in this paper depends on components that are certified against the relevant O-PAS profiles. The Open Group certification programme is what turns a supplier's claim into something independently tested, and conformance must be maintained as both the standard and the components evolve. Operators who want these capabilities should build conformance into their specifications and their conversations with suppliers now.

Questions for suppliers and integrators

1. Which O-PAS profiles is this component certified against, and against which version of the standard?
2. Can it be monitored, configured and updated through the standard O-PAS system management interfaces, or only through your own tools?
3. Which security requirements from Part 2 does it meet, and how is that evidenced?
4. Can this application be redeployed onto another supplier's DCN, and what evidence can you show?
5. Are your advanced control and analytics products delivered as conformant O-PAS applications, and on what timescale?
6. How will you support coexistence with our installed DCS through gateway DCNs?

Steps operators can take now

- Reference O-PAS conformance and certification in specifications for new control system work.
- Identify an architecture owner and begin building internal capability in interfaces, system management and software practices.
- Map the installed base and identify where coexistence through gateway DCNs would let modernisation start unit by unit.
- Choose an early application, such as an advanced control or analytics use case, where the benefits of an open architecture are easy to demonstrate.

A simple test

The answers to these questions show quickly whether you are buying an architecture or a platform with an OPC UA port on the side.

14. Conclusion

Open Process Automation is often presented as a way to buy control systems from more than one supplier. That is true, but it is the smaller part of the story. By separating applications from hardware, standardising communication, information and management, and making security and conformance properties of every component, O-PAS changes what a control system can do.

Orchestration gives operators one consistent way to see and manage the whole system. Embedded security gives them components designed to be secured. Advanced control can run where it performs best and improve continuously. Control logic becomes an asset the operator owns. Hardware can be refreshed without rewriting applications. Data carries its meaning to every consumer. Change becomes routine rather than exceptional, and modernisation becomes a programme rather than a bet.

The most important decision in the next generation of control systems is not which product to buy. It is which architecture will govern the system for the next 20 years.

About the author and next steps

Trevor Cusworth is a Principal at Collaborative Systems Integration (CSI), a vendor-neutral industrial automation consultancy. He has 47 years of process manufacturing experience across ICI, Imperial Oil, Aspen Technology, Invensys, Hagen & Company, Deloitte and Schneider Electric. He co-chairs Marketing and Outreach for the Open Process Automation Forum, having served as Forum Co-Chair from 2017 to 2023. CSI is a co-founding member of COPA.

Learn more

Start with the free [Introduction to Open Process Automation](https://csi.learnworlds.com) course at csi.learnworlds.com.

For the full treatment, including architecture, project execution, migration, procurement, governance and the business case, read *Open Process Automation: Architecture, Execution, and Transformation* by Trevor Cusworth, available on Amazon.