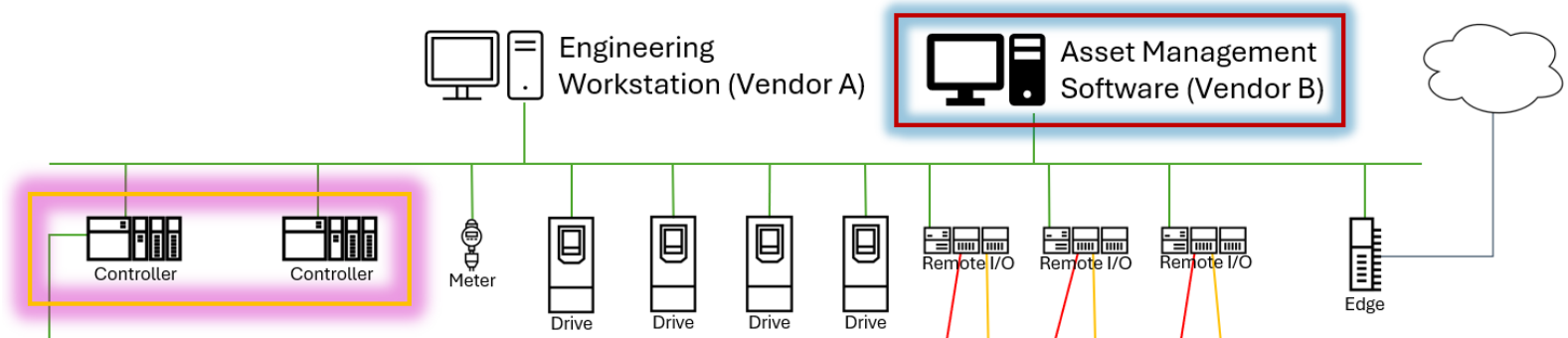


Designed for the Cloud
A Stateless API approach to Multi-protocol Device Integration

Mike Miller, Paul Brooks (Rockwell Automation)

October 21, 2026

A Typical Industrial Automation System Architecture



Org	Protocol	Integration
ODVA	EtherNet/IP	EDS
PI	IO-Link	IODD
FieldComm Group	HART	FDI

© 2026 ODVA, Inc.

All rights reserved.

The problem is not protocol diversity
It's repeated implementation of the same capabilities

For device vendors	For software vendors	For customers	The outcome
• Each protocol can require a separate stack • Repeated exposure of identity, data, configuration and commands • More combinations to qualify and sustain	• Different client libraries and object models • Protocol-specific discovery and error handling • Repeated integration and regression work	• More gateways and configuration • Inconsistent behaviour across tools • Higher lifecycle complexity and slower change	• Device vendor value dependent on host vendor implementation • Host vendor value dependent on device vendor implementation • Everyone does the minimum implementation "that **my** customers are asking for"

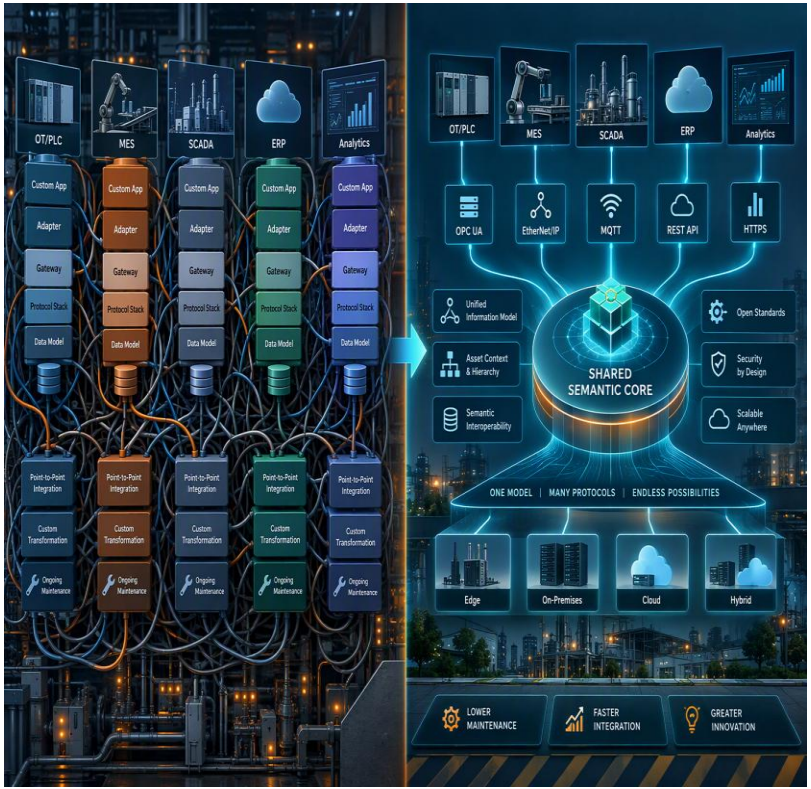
© 2026 ODVA, Inc.
All rights reserved.

Why the current model tends to compound cost

Each new protocol tends to replicate:

- data and command mapping
- security and identity integration
- state and session handling
- diagnostics and observability
- test, documentation and support

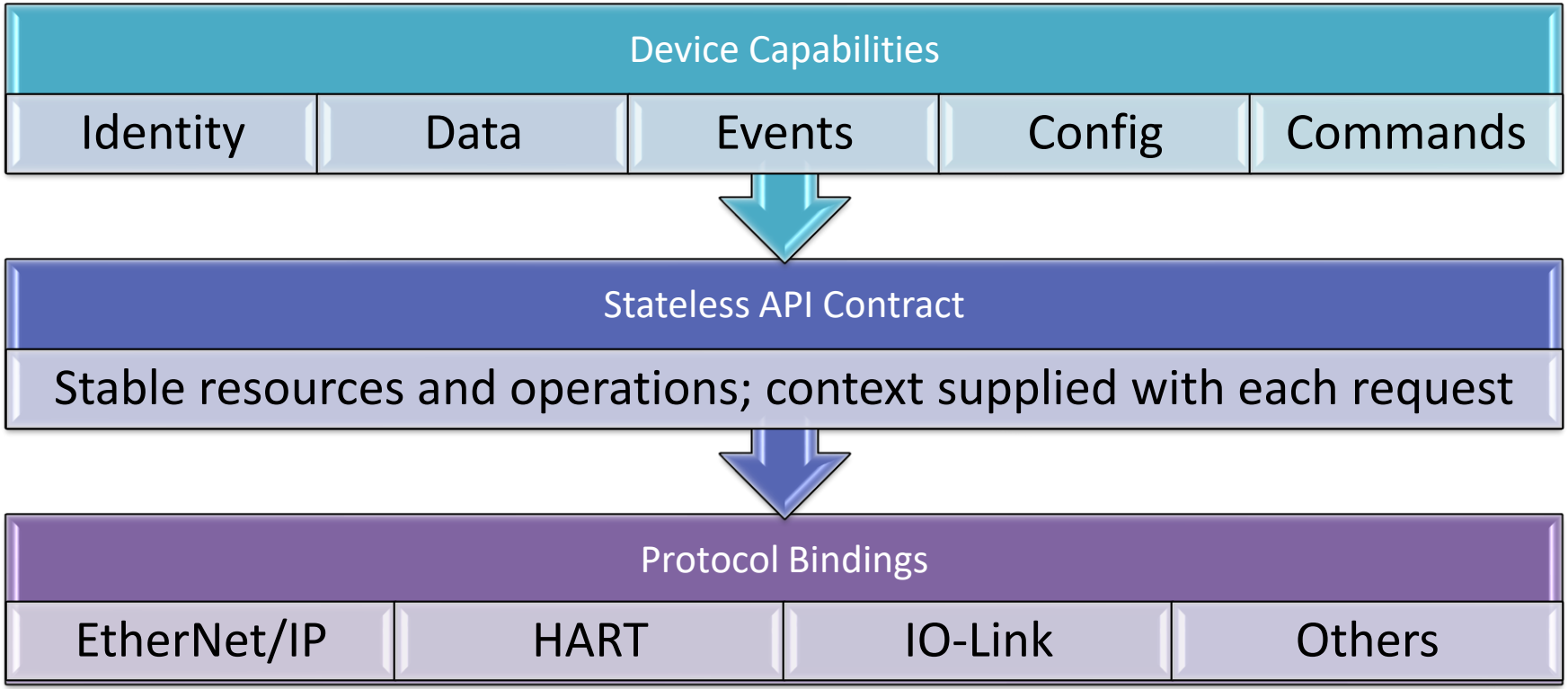
The multiplier is organisational as well as technical.



© 2026 ODVA, Inc.

All rights reserved.

The architectural move: separate meaning from transport



© 2026 ODVA, Inc.
All rights reserved.

One semantic core, several protocol experiences



Shared Core

- A common resource model for identity, data, events, configuration and commands.

Thin bindings

- Protocol adapters map common operations while preserving protocol-specific strengths

Independent evolution

- New bindings can be added without rebuilding the complete device or application model

© 2026 ODVA, Inc.

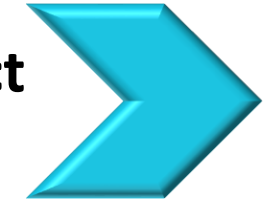
All rights reserved.

The vendor cost equation changes

Traditional pattern

Capability × Protocol × Product

More combinations to
implement, qualify and support



Shared-core pattern

Capability core
+ Protocol bindings

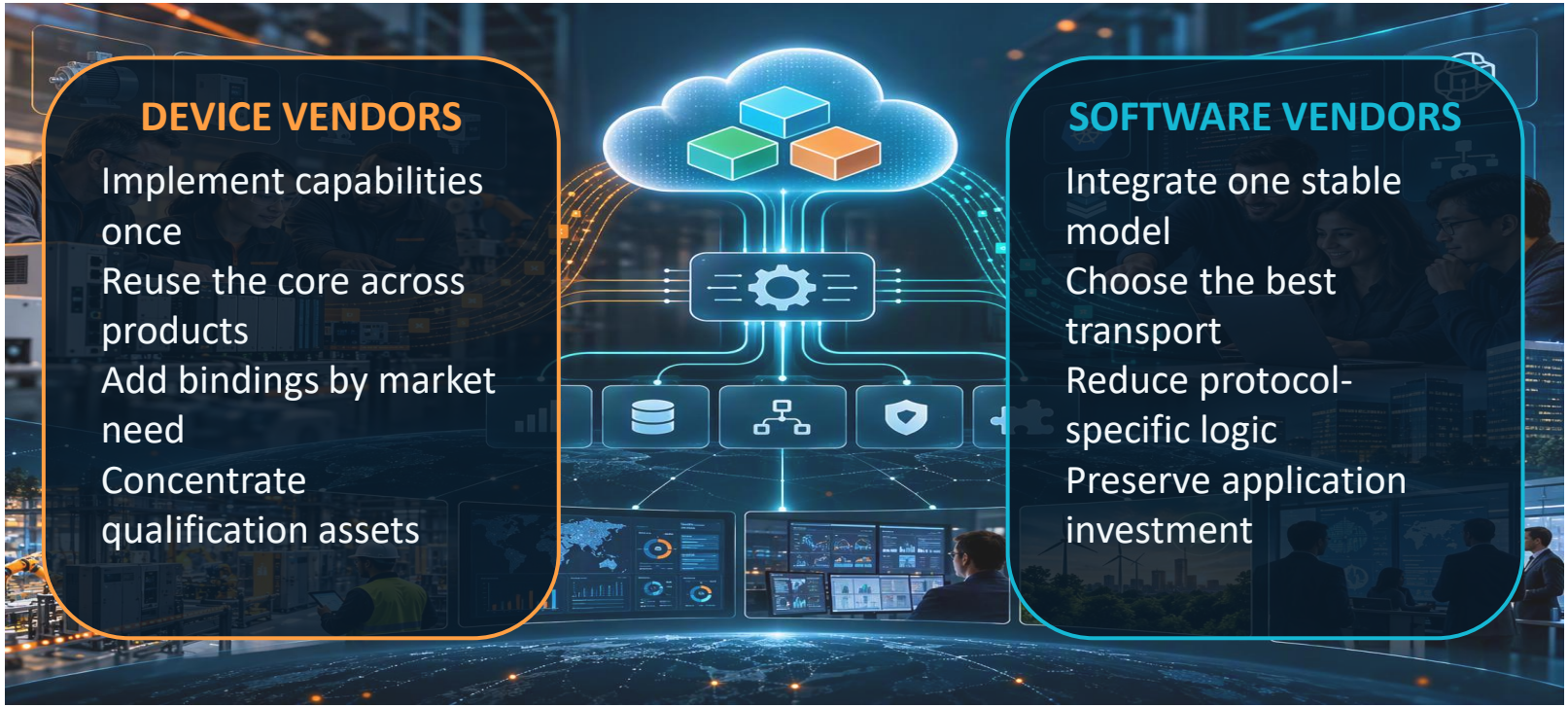
Reuse concentrates investment in the
common model and tests

Expected outcome

- less duplicated code
- smaller regression surface
- faster protocol additions
- more consistent behaviour

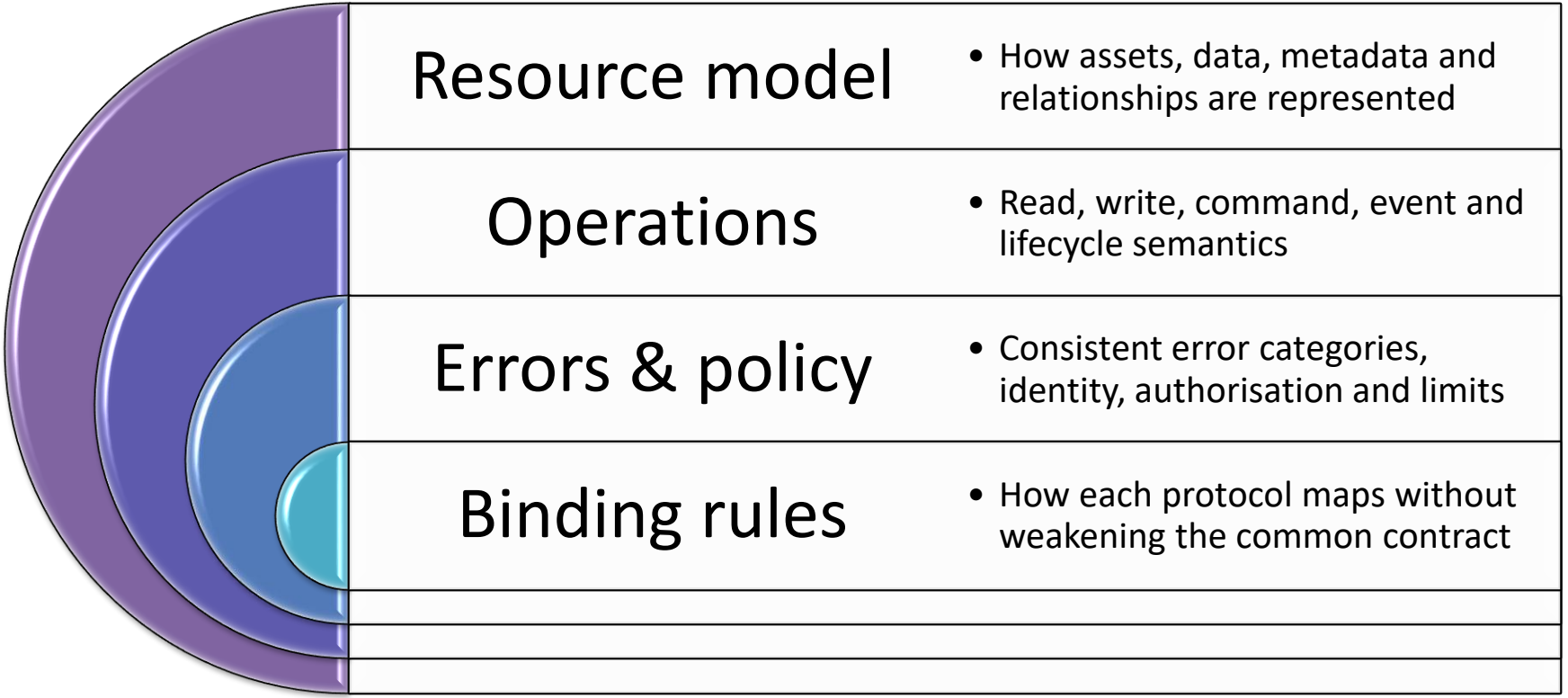
© 2026 ODVA, Inc.
All rights reserved.

Value for both sides of the ecosystem



© 2026 ODVA, Inc.
All rights reserved.

What must be standardised



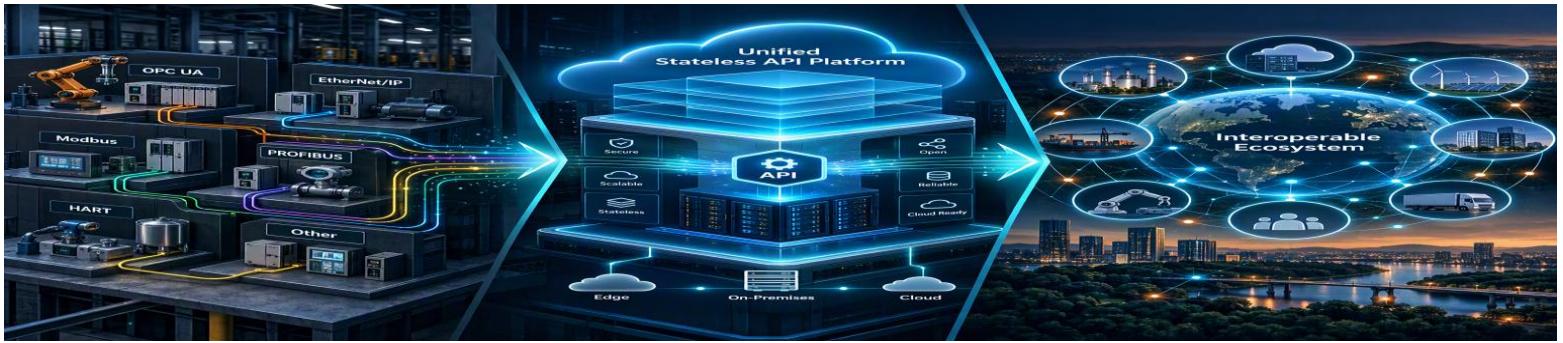
© 2026 ODVA, Inc.
All rights reserved.

Design Guardrails

- Lowest-common-denominator API
  - If protocol strengths disappear, vendors will bypass the common layer.
- Hidden state in adapters
  - Session dependence undermines scale-out and recoverability.
- Semantic drift
  - Bindings must not reinterpret the same capability differently.
- Monolithic gateway
  - A single component can recreate lock-in and slow independent evolution.

© 2026 ODVA, Inc.
All rights reserved.

A pragmatic adoption path



Define

Agree the minimum common resources and operations.

Prove

Bind two materially different protocols to one implementation

Measure

Compare code, tests, latency, operability and support effort

Expand

Add bindings and migrate products incrementally.

© 2026 ODVA, Inc.

All rights reserved.

What a credible proof should demonstrate

Functional equivalence

- The same asset capabilities are available through multiple bindings

Stateless operation

- Requests can move between instances without hidden session recovery

Semantic consistency

- Values, errors and commands mean the same thing across bindings

Operational evidence

- Scale, failure recovery, observability and security can be assessed

Reuse evidence

- Common code and tests outweigh binding-specific implementation

© 2026 ODVA, Inc.
All rights reserved.



DEMO

© 2026 ODVA, Inc.

All rights reserved.

The paper’s central proposition

Industrial interoperability becomes more economical when protocols become bindings to a shared capability model, rather than separate product architectures.

Cloud-native

Designed for scale, automation and replaceable instances.

Stateless

Context is explicit; durable state is separately managed.

Multi-protocol

One semantic core supports several optimised bindings.

Lower cost

Less duplicated implementation, test and support effort.


2026
ODVA
Industry Conference and Annual Meeting