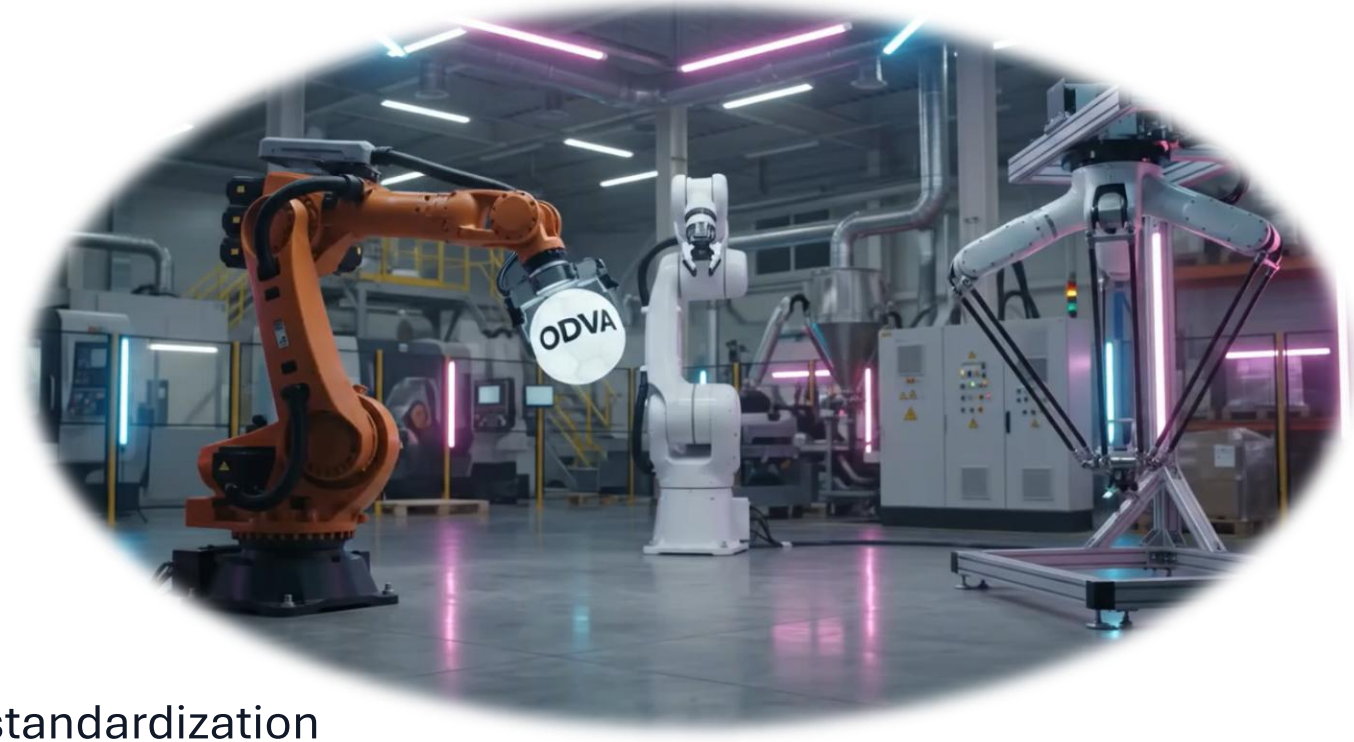


Standardized PLC to Robot Integration using SRCI over EtherNet/IP via CIP Technologies

Joakim Wiberg, ODVA
Tommaso Fochi, Rockwell Automation
Paul Brooks, Rockwell Automation
Todd Wiese, Rockwell Automation

October 21, 2026

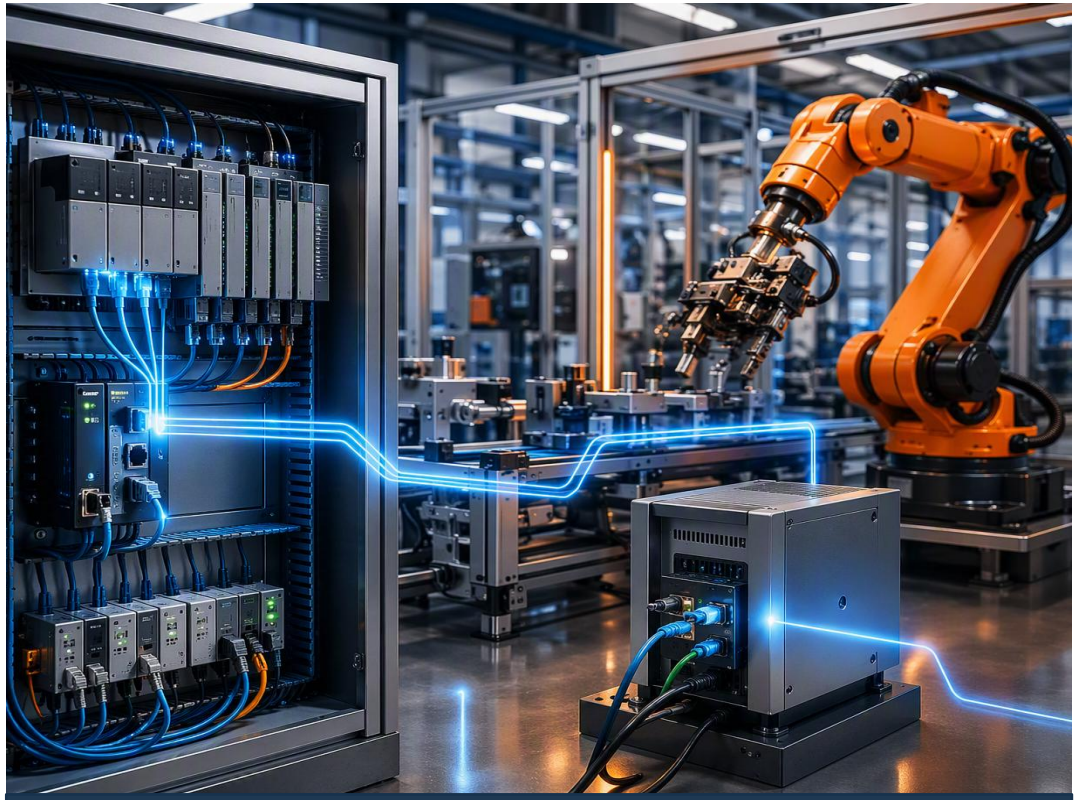


Topics Presented:

- Why robot integration needs another level of standardization
- What SRCI provides today and what gap remains for EtherNet/IP environments
- Proposed CIP-based transport profile and architectural principles
- Command transport, state handling, sequencing, and capability discovery
- Interoperability, commercial value, implementation considerations, and next steps

All rights reserved.

The integration challenge: robots are normal automation assets



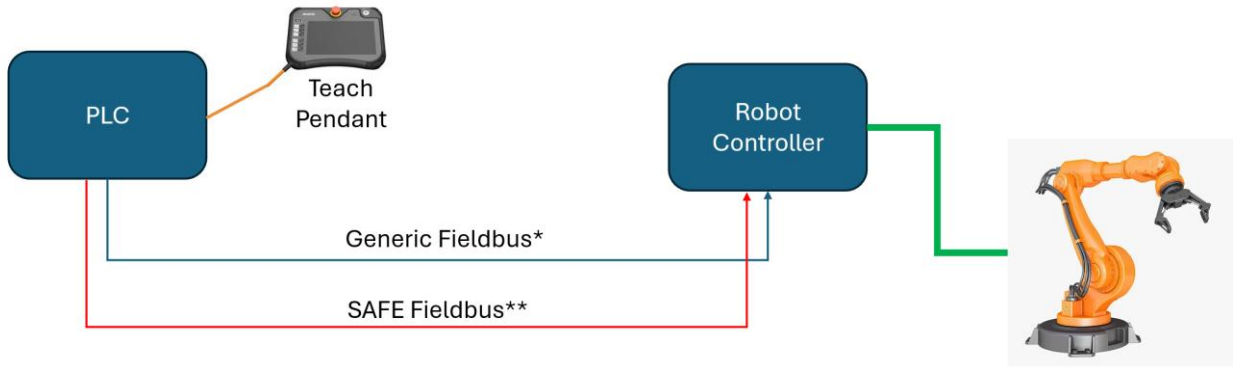
Common PLC integration across multi-vendor robot systems

- Industrial robotics adoption is accelerating across manufacturing environments.
- Multi-vendor robot deployments can introduce proprietary programming models, communications, and integration differences.
- System integrators and end users need scalable, vendor-neutral PLC-to-robot behaviors.
- SRCI addresses the application-level command/state abstraction, but EtherNet/IP transport standardization remains a gap.

All rights reserved.

What is SRCI?

- Standard Robot Controller Interface: a PLC-centric command and state abstraction for robot control.
- Defines command execution, state management, capability access, and operational control.
- Allows multiple robot platforms to be controlled through a consistent application model.
- Reduces application complexity and improves portability across robot manufacturers.

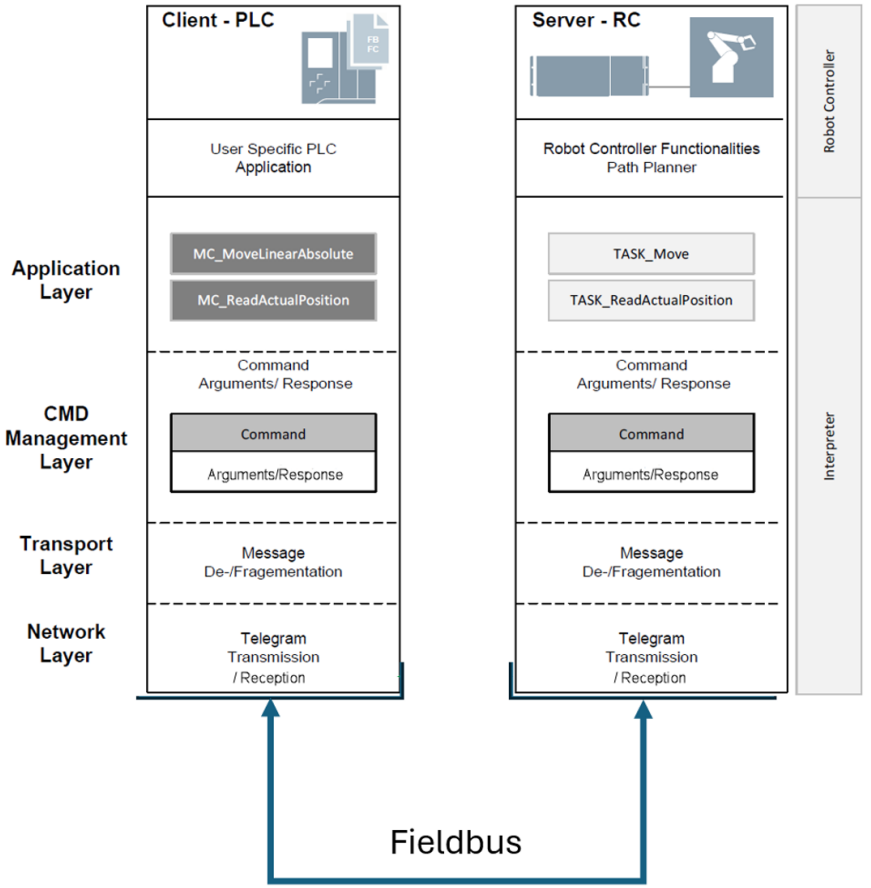


All rights reserved.

The gap: SRCI standardizes behavior, but not with an EtherNet/IP transport

SRCI gives the application contract

- Command semantics
- Execution behavior
- State transitions
- Capability access
- PLC-centric programming model



EtherNet/IP provides an additional transport answer

- Common Industrial Protocol alignment
- Cyclic command/status exchange
- Explicit discovery and diagnostics
- Conformance expectations
- No proprietary gateway assumptions

All rights reserved.

Proposal: a CIP profile for SRCI-compliant robot equipment



Direct PLC-to-robot communication using EtherNet/IP

- Represent the robot controller as a CIP device exposing standardized interfaces and services.
- Preserve SRCI command semantics, execution behavior, and operational assumptions.
- Use EtherNet/IP communication patterns already familiar to ODVA-conformant systems.
- Enable direct PLC-to-robot communication without custom gateways or translation layers.

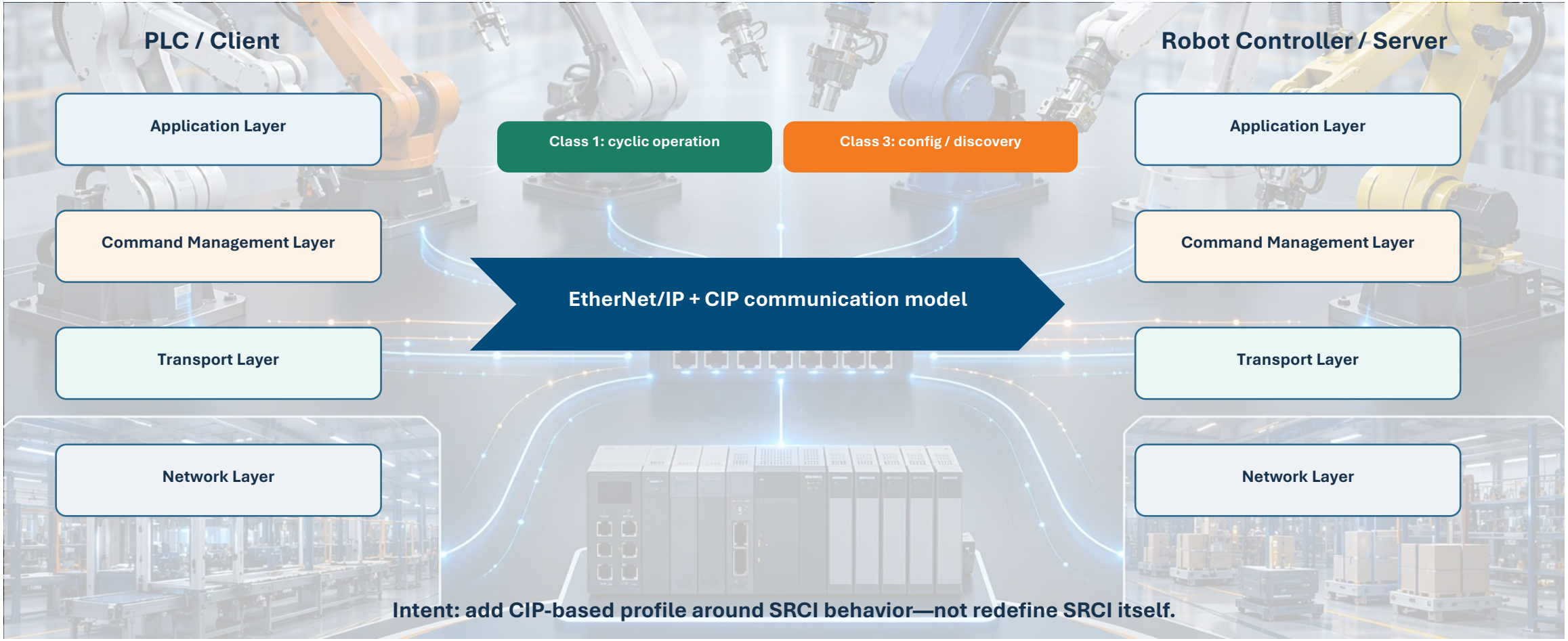
All rights reserved.

Architectural design principles

- Preserve SRCI semantics and command execution behavior.
- Leverage existing CIP technologies for safety, security, connection management, and messaging.
- Use deterministic communication for critical command execution and feedback.
- Separate operation from configuration, diagnostics, and discovery.
- Enable vendor-neutral interoperability with minimal robot vendor implementation impact.
- Align with ODVA conformance models and scalable capability discovery.

All rights reserved.

Proposed architecture: Preserve SRCI, map transport to CIP



2026 ODVA, Inc.
All rights reserved.

Transport strategy: Use the right CIP mechanism for each function

CIP Class 1: deterministic operational exchange

- Command execution
- Command acknowledgement
- Execution state monitoring
- Robot status reporting
- Motion feedback
- Command completion

CIP Class 3: explicit management services

- Capability discovery
- Device configuration
- Diagnostics
- Profile identification
- Maintenance functions

Separation of real-time operation from configuration and discovery reduces coupling and aligns with CIP practice.

All rights reserved.

Cyclic operation and explicit services work together



Cyclic data flow plus explicit services

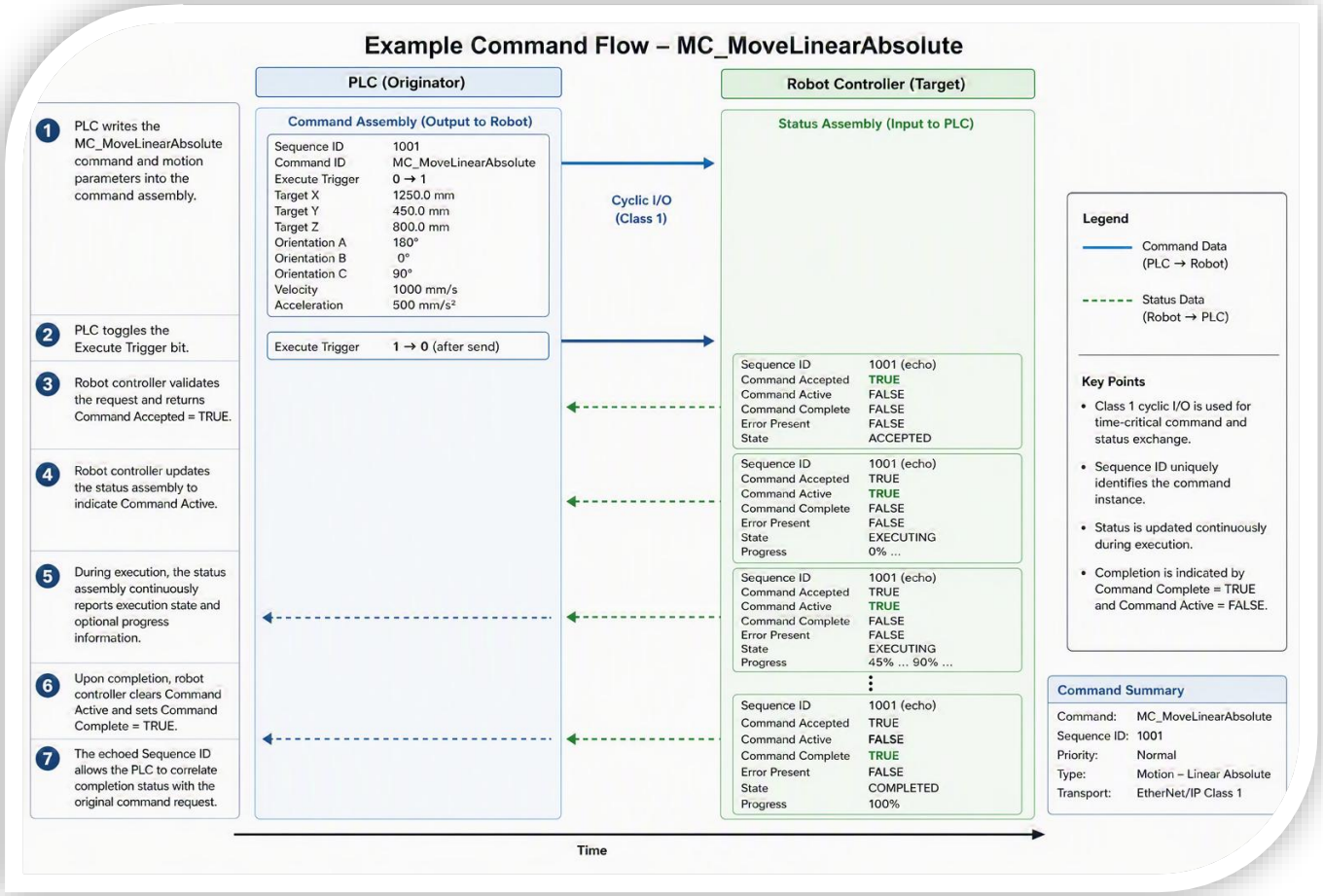
- Class 1 cyclic connections carry operational state and feedback needed during active execution.
- Class 3 explicit messaging supports complementary lifecycle functions such as configuration, diagnostics, and capability discovery.
- Together, they support command execution, state management, sequencing, and feedback while aligning to deployed EtherNet/IP practice.

All rights reserved.

Message model: SRCI elements mapped into CIP data structures

PLC → Robot Controller

- SRCI command invocation
- Command identifier
- Execution trigger
- Command parameters
- Execution control information



Robot Controller → PLC

- Command responses
- Acknowledgement status
- Execution progress
- Completion state
- Error indications and robot feedback

All rights reserved.

MC_MoveLinearAbsolute Message Transaction Flow

Parallel command/response lifecycle across representative robot-controller connections.

Parallel command transmission

PLC / Client

Command Assembly

- Seq ID: 1001
- Command: MC_MoveLinearAbsolute
- Cmd ID: 0x0021
- Execute: Rising Edge
- Target X/Y/Z + ABC
- Velocity / Acceleration

Same SRCI command sen
are reused for each vendor lane

Cnxn 1 — CMD 1001 MoveLinearAbs

Cnxn 2 — CMD 1001 MoveLinearAbs

Cnxn 3 — CMD 1001 MoveLinearAbs

CIP Class 1 Cyclic I/O

Command / status assemblies

Transaction identity

Sequence ID 1001 correlates:

- Command request
- Acknowledgement
- Execution progress
- Completion / fault status

Vendor A RC

Consumes command assembly

Returns status assembly

Receiving Command



Vendor B RC

Consumes command assembly

Returns status assembly

Receiving Command



Vendor C RC

Consumes command assembly

Returns status assembly

Receiving Command



1 Write command

2 Parallel transmit

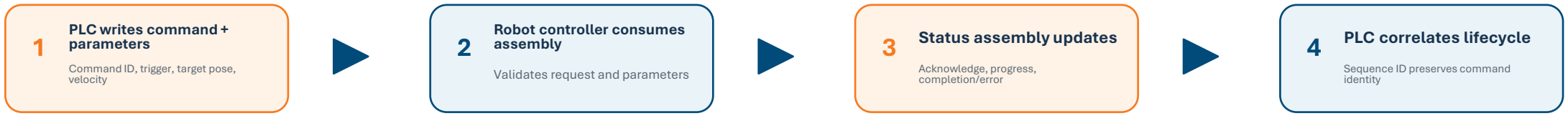
3 Validate / execute

4 Return ACK + progress

5 Return completion status

 © 2026 ODVA, Inc. All rights reserved.

Command execution example: MC_MoveLinearAbsolute over a cyclic connection



Payload elements example

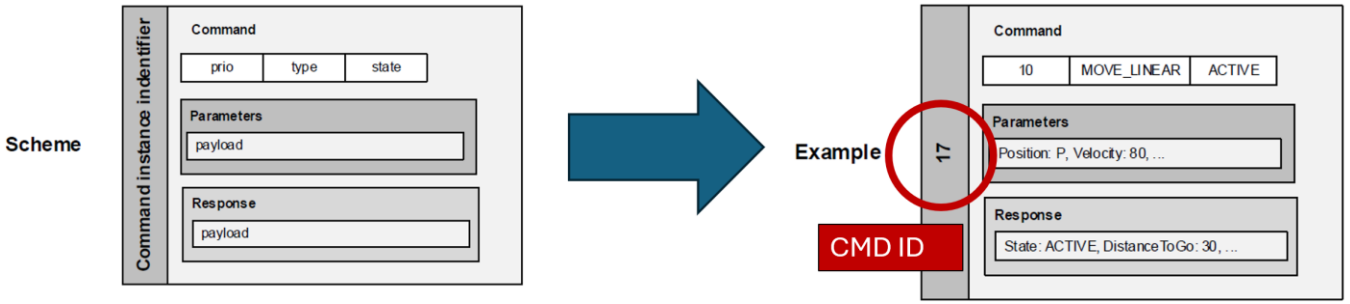
Command Sequence ID	Command ID = MC_MoveLinearAbsolute	Execute Trigger	Target X/Y/Z
Orientation A/B/C	Velocity	Acceleration	

Design principle: transaction identity + cyclic status preserve SRCI execution semantics while using EtherNet/IP data exchange.

All rights reserved.

Sequencing and execution integrity

- Each command transaction is associated with a unique execution identifier.
- The identifier correlates command requests, acknowledgements, execution state updates, completion notifications, and faults.
- Sequencing rules preserve intended SRCI execution behavior, including buffered commands, sequential execution, interruption, and prioritization.
- Cyclic status assemblies provide a continuously updated view of command progress and include “LifeSign” information in the messaging headers, as an integrity check to discover discrepancies.



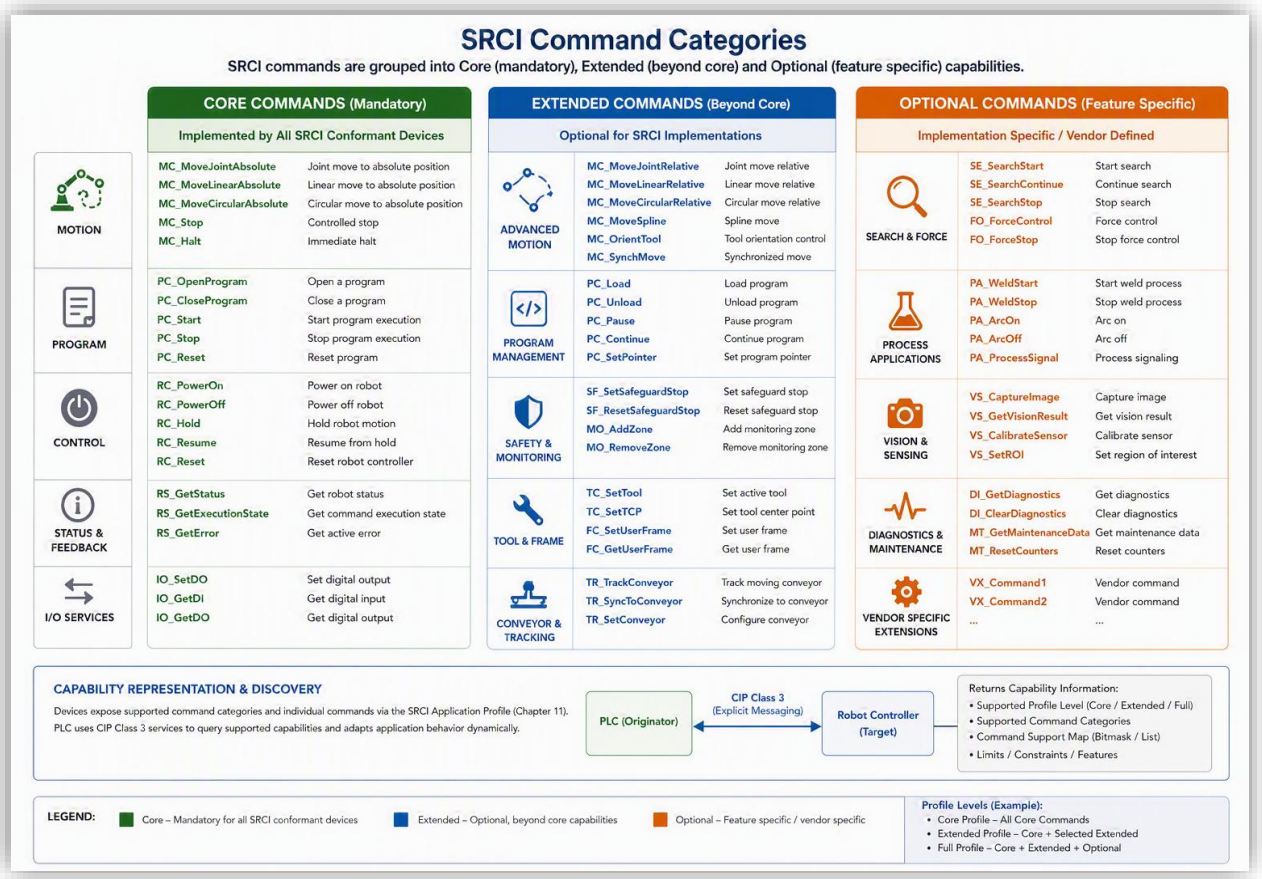
All rights reserved.

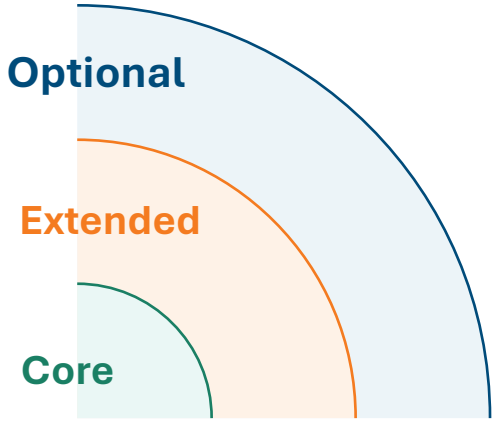
Fragmentation, reassembly, and data handling

- SRCI messages may exceed practical cyclic data sizes and therefore require a transport profile to define fragmentation and reassembly behavior.
- The receiving side must reconstruct the SRCI message while preserving ordering, integrity, and command association.
- Fragment-level errors shall be detectable and reported in a way that preserves the command lifecycle and supports diagnostics.
- The paper positions fragmentation and data handling as part of a complete transport profile rather than an application workaround.

All rights reserved.

Capability model: Core, extended, and optional SRCI support





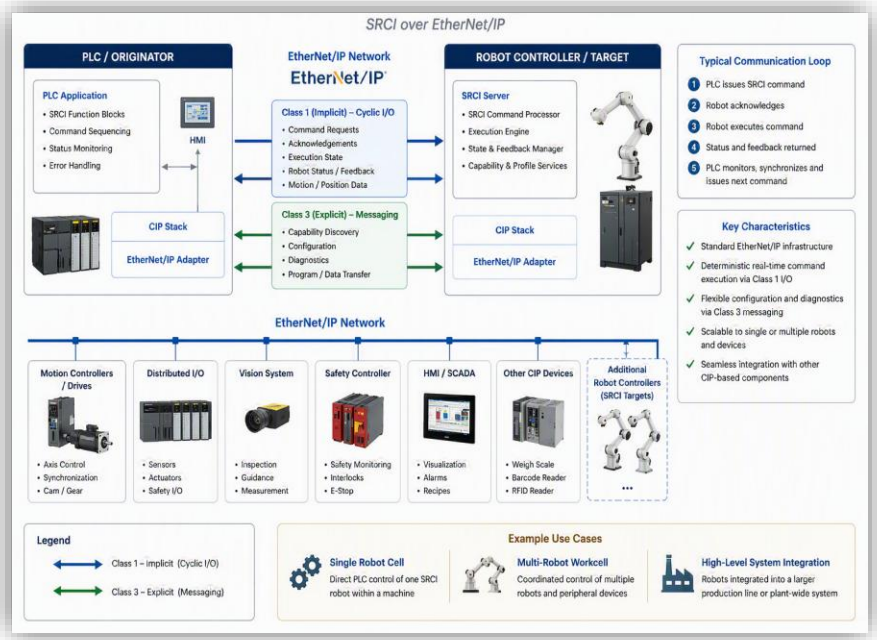
- Expose supported SRCI profiles and command categories through CIP profile information.
- Enable scalable adoption for simple and advanced robot controllers.
- Support conformance discussions around required vs. optional behavior.
- Avoid assuming every robot implements every SRCI function.

All rights reserved.

PLC and robot controller integration models

PLC-centric control model

- PLC invokes SRCI function blocks or commands.
- Command Management Layer formulates command requests and interprets responses.
- Robot behavior remains aligned to SRCI semantics.

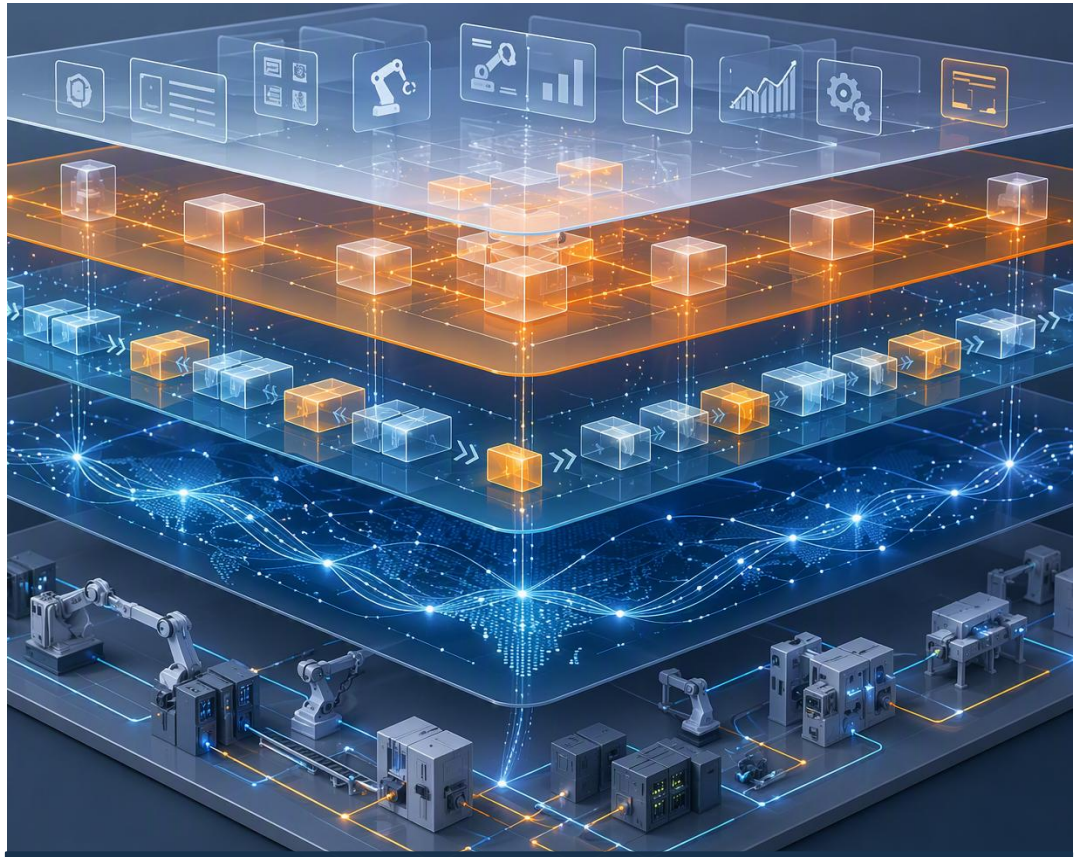


Robot controller as CIP server

- Exposes standardized communication structures.
- Consumes command assemblies and produces status/feedback assemblies.
- Provides discovery, diagnostics, and configuration through explicit services.

All rights reserved.

Value: less custom glue, more reusable integration



Commissioning through a familiar PLC engineering workflow

- Reduce integration complexity by avoiding proprietary gateways or vendor-specific adaptation layers.
- Increase application portability across compliant robot platforms.
- Reuse EtherNet/IP infrastructure, engineering practices, and ODVA-conformant device models.
- Create a stronger foundation for multi-vendor robot ecosystems.

All rights reserved.

Technical value: a disciplined mapping instead of ad hoc integration

- CIP device modeling supplies a consistent place for robot controller capabilities, services, and attributes.
- Cyclic I/O provides an established mechanism for observable command/status behavior.
- Explicit messaging provides a complementary model for discovery, diagnostics, profile identification, and maintenance.
- Common data representation rules support consistent interpretation across controller platforms.

All rights reserved.

Interoperability, safety, security, and compliance considerations

- Interoperability requires well-defined models, assemblies, command handling behavior, and capability discovery.
- Safety and security should leverage existing EtherNet/IP and CIP technologies where applicable rather than redefining the robot safety model.
- Diagnostics shall make transport, command, and capability issues visible to engineering and conformance workflows.
- Profile definition should support ODVA-style validation and conformance expectations.

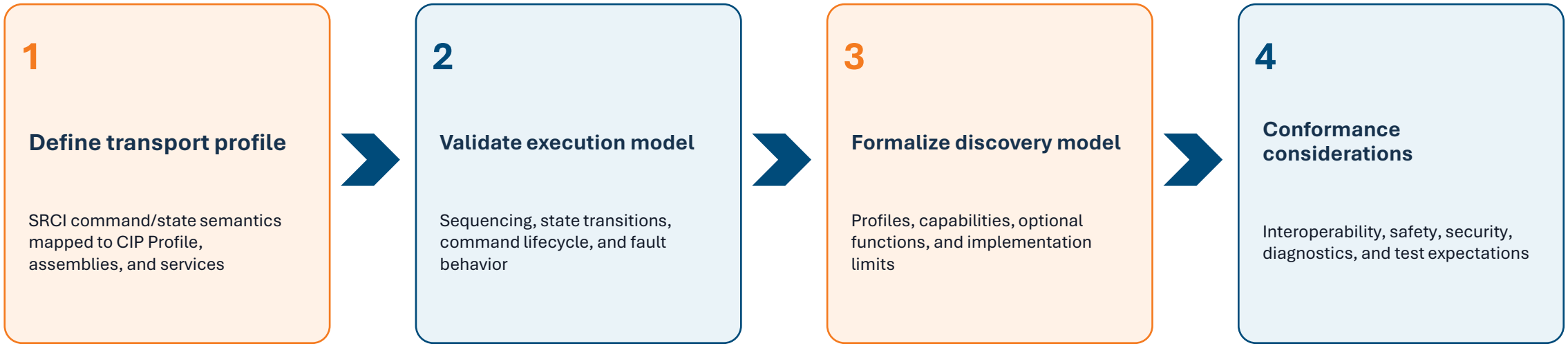
All rights reserved.

Implementation considerations and open questions

- How should SRCI capabilities be exposed: attributes, profile, or another discovery structure?
- What command categories are mandatory for initial conformance versus optional extensions?
- How should fragmentation errors, sequence errors, and command faults be normalized for diagnostics?
- What byte-level mapping should be normative versus implementation-defined?
- How should existing SRCI implementations be reused with minimal impact on robot vendors?

All rights reserved.

Standards-aligned extension



Recommended: Provide ecosystem feedback on interface, capability discovery, and conformance scope.

All rights reserved.

Ecosystem Outcome: reusable framework for ODVA-compliant robotics



Standards-aligned, interoperable, extensible

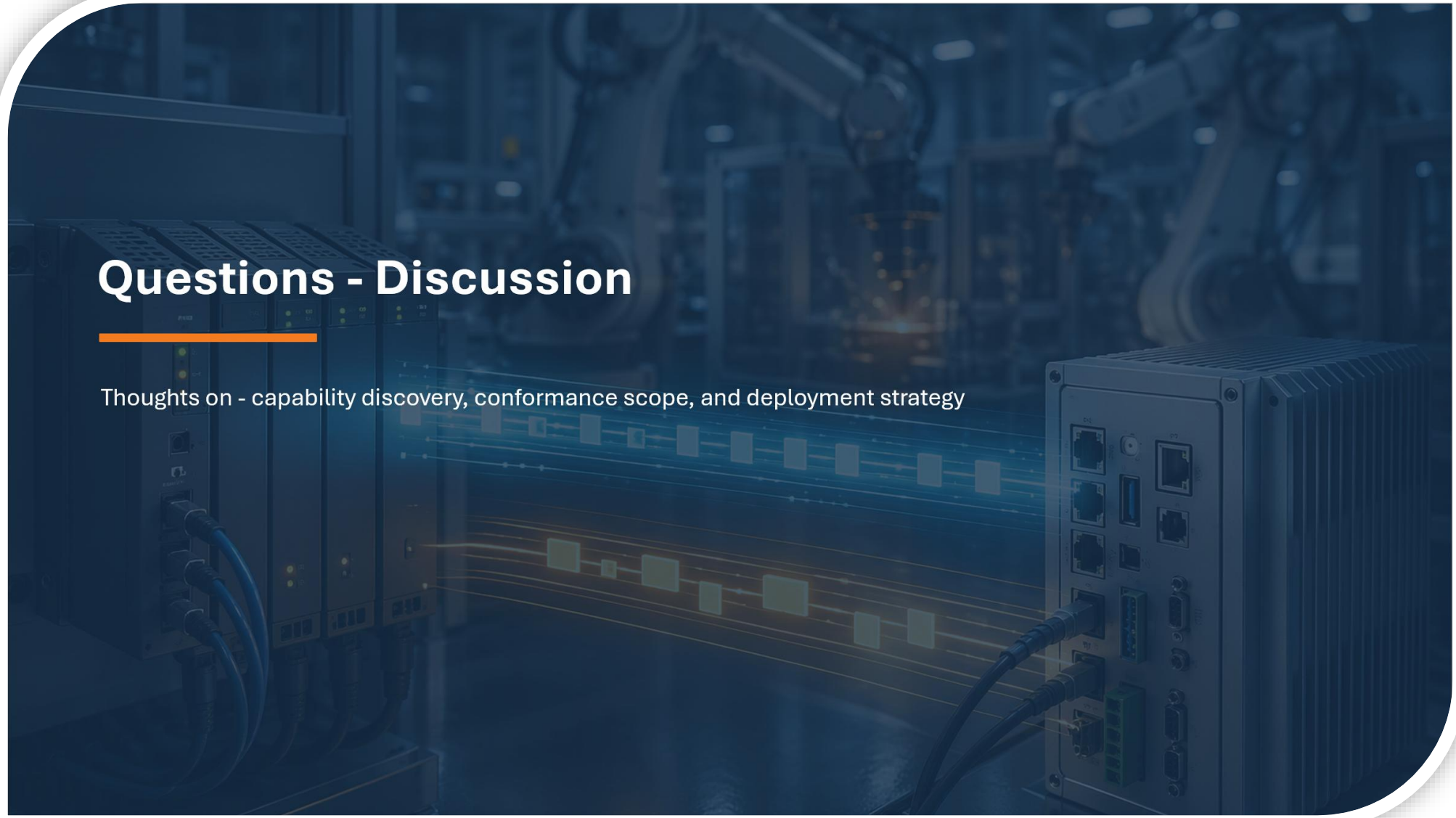
- Reusable standards-aligned framework for PLC-to-robot communication over EtherNet/IP.
- Preserves SRCI behavior while extending applicability into CIP-based architectures.
- Supports multi-vendor compatibility and future enhancements in industrial robotics control.
- Creates a common interoperable interface point for vendors, system integrators, and end users.

All rights reserved.

Key takeaways

- SRCI provides the application-level robot command and state abstraction.
- EtherNet/IP environments need a standardized CIP-based transport/profile to avoid proprietary bridging.
- The proposed architecture preserves SRCI semantics while leveraging Class 1 cyclic I/O and Class 3 explicit messaging.
- Capability discovery, command categorization, sequencing, and reassembly/error handling are central to interoperability.
- The commercial value is reduced integration complexity and a stronger multi-vendor robotics ecosystem.

All rights reserved.



Questions - Discussion

Thoughts on - capability discovery, conformance scope, and deployment strategy

All rights reserved.


2026
ODVA
Industry Conference and Annual Meeting