

A Standardized Approach for Firmware Updates over EtherNet/IP

Matt Frazer, ODVA
Joakim Wiberg, ODVA

October 21, 2026

Agenda

1 **The Problem**
Fragmented firmware updates in multi-vendor environments

2 **Regulatory Drivers**
CRA, NIS2, and IEC 62443

3 **Proposed Architecture**
Three-pillar framework

4 **Use Cases**
Simple to complex scenarios

5 **CIP Objects & Services**
Standardized update interface

6 **Firmware Packaging**
OPC-pkg container and JSON manifest

7 **Related Work**
TUF, Uptane, ENISA advisory

8 **Next Steps**
Path through ODVA standardization

© 2026 ODVA, Inc.
All rights reserved.

The Problem: Firmware Updates Today

Vendor-specific tools	Each vendor requires separate proprietary tooling and processes
No common interface	Different protocols, transport mechanisms — some requiring physical USB connections
No discovery mechanism	No standard way to know when new firmware is available
Manual and error-prone	Updates deferred due to process complexity, leaving devices vulnerable

© 2026 ODVA, Inc.
All rights reserved.

Regulatory Drivers

CRA

EU Cyber Resilience Act

- Mandatory security updates throughout product lifecycle
- Secure update mechanisms required
- Timely vulnerability remediation

NIS2

EU Directive 2022/2555

- Operators must maintain documented security posture
- Consistent, auditable update processes required

IEC 62443

2-1 & 2-2

- Security management system requirements
- Implementation guidance for industrial control systems

Key insight: Vendors each fulfil their CRA obligations individually. End users carry the burden of managing a fragmented landscape.

© 2026 ODVA, Inc.
All rights reserved.

Benefits of a Standardized Approach

Single tool

One tool manages all devices across the facility — regardless of vendor

Scalability

New vendors and devices slot into the same workflow automatically

Reduces errors

A common interface and packaging format make every update predictable

Lower costs

Reduced training, fewer errors, less manual effort at scale

Audit & compliance

Consistent, loggable update process supports NIS2 and IEC 62443 requirements

© 2026 ODVA, Inc.
All rights reserved.

Proposed Architecture: Three Pillars

1

CIP Objects & Services

Standardized vendor-neutral update interface built on existing CIP messaging, Identity Object, File Object, and the new NVS Object

2

OPC-pkg Container

Open Packaging Convention container (ECMA-376) bundling firmware images, signatures, and supporting files

3

JSON Manifest

JSON Schema-validated control document giving conformant tools everything needed to execute an update without vendor-specific knowledge

Any conformant tool can update any conformant device — regardless of vendor.

© 2026 ODVA, Inc.
All rights reserved.

Architecture Components

Update Tool

Reads manifest, discovers devices, orchestrates transfer

Update Package

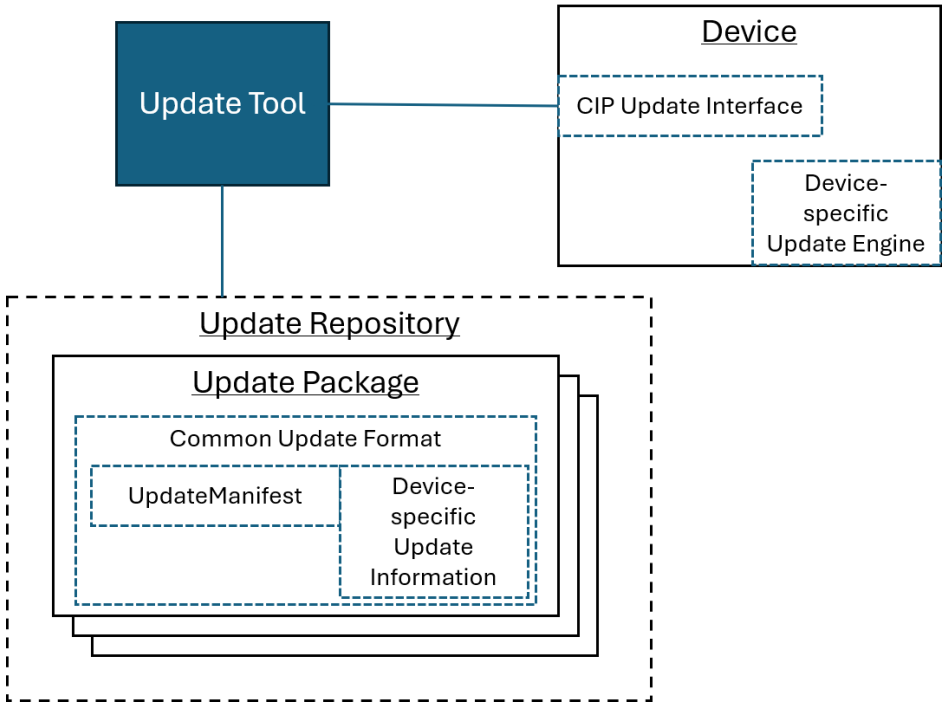
OPC-pkg container with JSON manifest, firmware images, and signatures

CIP Update Interface

Standardized CIP Objects and services on the device

Update Repository

Optional online repository (ODVA-hosted) for firmware discovery



© 2026 ODVA, Inc.

All rights reserved.

Use Cases

UC1	**Bundling multiple update artifacts** Manifest, images, signatures — one container	**UC2**	**Simple device** Single image, constrained hardware, NVS Object
UC3	**Complex device** Multiple images, File Object, sequenced transfer	**UC4**	**Multi-vendor facility** One tool, many vendors, identical experience
UC5	**Parallel updates** Simultaneous download, synchronized reboot	**UC6**	**Sequential/multi-step updates** Version constraints force intermediate hops
UC7	**Custom firmware distribution** Manifest identifier for test and debug builds	**UC8**	**Online repository** ODVA-hosted discovery and download
UC9	**Recovery/Bootloader** Minimal EtherNet/IP stack for rescue updates	**UC10**	**Signature & integrity** PKI-backed verification, supply chain protection
UC11	**Backup & restore** Configuration preservation across updates	**UC12**	**Transition path** Reuse existing firmware and internal update logic

© 2026 ODVA, Inc.
All rights reserved.

Use Case: Simple vs Complex Device Updates

Simple Device

- Single firmware image (+ optional bootloader)
- No filesystem — flat flash storage
- Uses new NVS Object for transfer
- Tool identifies device via Identity Object
- Transfer → Reset → Verify → Apply
- Ideal for sensors, actuators, small components

Complex Device

- Multiple images: bootloader, main FW, app images, filesystem
- Full filesystem — uses File Object
- Vendor-defined sequencing and restart behavior
- Optionally also supports NVS Object path
- Secure connections via CIP Security
- Complexity transparent to end user

© 2026 ODVA, Inc.
All rights reserved.

Use Case: Interaction Model - Simple Device

1-2
Query Identity
Tool reads Vendor ID, Device Type, Product Code, Revision

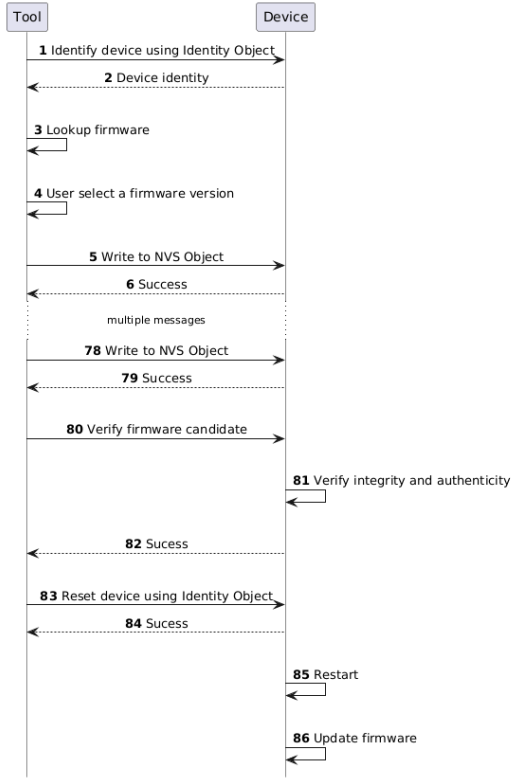
3-4
Match firmware
Tool identifies compatible firmware locally or from repository

5-79
Transfer image
Firmware image sent to device via NVS Object

80-82
Verify
Device verifies digital signature

83-84
Reset
Tool issues Reset service via Identity Object

85-86
Apply
Device applies new firmware

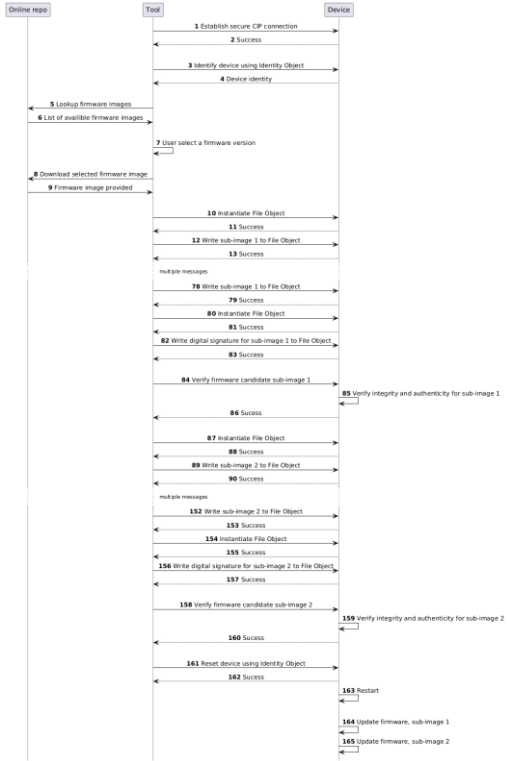


© 2026 ODVA, Inc.

All rights reserved.

Use Case: Interaction Model - Complex Device

1-2	**Secure connection** Tool establishes a secure connection to the device (CIP Security Confidentiality Profile)
3-4	**Query Identity** Tool reads Vendor ID, Device Type, Product Code, Revision
5-7	**Match & select firmware** Tool identifies compatible firmware at online repository; user selects version
8-9	**Download package** Tool downloads selected firmware package from the online repository
10-86	**Transfer and verify sub-image 1** File Object instantiated; sub-image 1 transferred; File Object instantiated again; signature for sub-image 1 transferred and verified
87-160	**Transfer and verify sub-image 2** File Object instantiated; sub-image 2 transferred; File Object instantiated again; signature for sub-image 2 transferred and verified
161-162	**Reset** Tool issues Reset service via Identity Object
162-165	**Apply** Device then applies both firmware images



Signatures are provided as separate files in the firmware package and transferred independently after each sub-image.

© 2026 ODVA, Inc.
All rights reserved.

Use Case: Update Multiple Devices in Parallel

Scenario: A group of devices needs to run the same firmware version — for example after a security patch or a coordinated system upgrade.

1	**Select devices**	User selects a set of devices — may span multiple vendors
2	**Initiate update**	Tool downloads appropriate firmware to all selected devices in parallel
3	**Wait for completion**	Different device types take different times — tool waits for the last transfer to complete
4	**Synchronized reboot**	User triggers simultaneous reboot across all devices — new firmware activated at the same time
5	**Confirm**	Tool polls each device and confirms new firmware version is active

Because the update interface is standardized, selecting a mix of devices from different vendors is no different from selecting devices from a single vendor.

© 2026 ODVA, Inc.
All rights reserved.

Use Case: Recovery / Bootloader Mode

Scenario: A firmware update fails or leaves the device in an inoperable state. The device must be recoverable over EtherNet/IP without physical intervention.

Minimal EtherNet/IP stack required

- ListIdentity, RegisterSession, SendRRData encapsulation services
- Simple unconnected and Class 3 messaging (single connection)
- Identity Object
- Message Router Object
- NVS Object

Why the NVS Object fits here

- Deliberately limited scope — minimal resource requirements
- Designed for flat flash storage — no filesystem needed
- Simple read/write services only
- Practical to implement in a constrained bootloader environment where a full CIP stack is not available

The device runs just enough EtherNet/IP to accept a new firmware image and return to normal operation.

© 2026 ODVA, Inc.
All rights reserved.

Use Case: Signature, Integrity and Authenticity

Firmware directly controls device behavior. A tampered or corrupted package could compromise the safety and security of an entire installation.

Container signing

- OPC-pkg built-in digital signature mechanism
- Covers the package as a whole
- Any conformant tool can verify — no vendor-specific knowledge
- Tools must reject packages that fail signature validation
- Protects against supply chain attacks

PKI hierarchy

- ODVA acts as root CA
- Member organizations operate as subordinates
- Alternatively: globally trusted certificate chain
- Open, verifiable trust — no proprietary mechanisms

Per-image integrity

- Required for each firmware image inside the container
- Specific mechanism not mandated — vendor choice
- Manifest declares the method used
- Tool performs the check without vendor-specific knowledge

Backup and restore is designed as a general-purpose capability — useful independent of firmware updates, not tied to the update workflow alone.

© 2026 ODVA, Inc.
All rights reserved.

Use Case: Backup and Restore

Some firmware updates reset or overwrite device configuration. Before proceeding, the tool must be able to upload and preserve the current configuration.

1	**Check configuration flag**	Identity Object 'Configured' bit indicates whether a configuration is present and should be backed up
2	**Warn user**	Firmware container metadata instructs the tool to inform the user before proceeding if configuration will be affected
3	**Backup**	Tool uploads current configuration from the Config Instance (File Object or NVS Object instance) and stores it locally
4	**Apply firmware update**	Standard update workflow proceeds as normal
5	**Restore configuration**	Tool pushes stored configuration back to the device after the update is complete

Because the update interface is standardized, selecting a mix of devices from different vendors is no different from selecting devices from a single vendor.

© 2026 ODVA, Inc.
All rights reserved.

CIP Objects & Services

Identity Object	Device identification, status, Reset service, Config Instance attribute
Message Router	Required when using Send_Receive_Fragment for binary transfer
Connection Manager	Required for Class 3 connections used with Send_Receive_Fragment
CIP Security	Confidentiality Profile — authenticity and optional encryption
File Object	For complex devices and CIP Security-capable devices; handles filesystem storage
NVS Object (NEW)	Lightweight new object for constrained devices — flat flash storage model

NVS Object is also well-suited for recovery/bootloader mode — minimal implementation footprint.

© 2026 ODVA, Inc.
All rights reserved.

The NVS Object: A New CIP Object

Why a new object?

The File Object assumes a richer storage model. Many constrained devices use flat flash memory — the NVS Object is designed for this from the ground up, keeping implementation lightweight.

Instances:	One or more instances, each representing a contiguous non-volatile storage area
Access model:	Simple read and write services — no filesystem assumptions
Multi-image support:	Single instance can handle multiple images sequentially (one at a time)
Configuration storage:	Dedicated instance can hold device configuration, referenced by Identity Object's Config Instance attribute
Bootloader fit:	Minimal scope makes it practical for constrained bootloader environments

© 2026 ODVA, Inc.
All rights reserved.

Firmware Packaging Format: OPC-pkg

OPC-pkg Container

Based on the Open Packaging Convention (ECMA-376) — the same open standard underlying Office Open XML formats.

- Well-documented open standard
- Broad tooling and platform support
- ZIP-based — easy to inspect
- Relationship model links manifest to firmware parts
- No restriction on firmware file types
- Vendors embed existing binaries unchanged

Package Contents

manifest.json	Control document — validated against JSON Schema
Firmware images(s)	One or more binary images, vendor format unchanged
Digital signatures	Per-image or package-level signatures
Release notes	Optional human-readable documentation
License files	Optional legal artifacts

Note: 'OPC-pkg' is used to avoid confusion with OLE for Process Control (also abbreviated OPC).

© 2026 ODVA, Inc.
All rights reserved.

The JSON Manifest: Control Document

Device identity	Vendor ID, product type, product code — matched against Identity Object
Version info	Firmware version, min/max version constraints for compatibility checks
Image list	Each image: role, target object (NVS or File), instance, and transfer order
Signature refs	Per-image signature file references, transfer timing (before/after image)
Reset control	Per-image or deferred reset — defined by vendor, executed by tool
Security config	Whether CIP Security is required and whether confidentiality is enabled
Owned/Configured	Instructions to check device status bits before proceeding

Validated with JSON Schema — malformed or incomplete packages caught before any transfer begins.

© 2026 ODVA, Inc.
All rights reserved.

Security: Integrity & Authenticity

Container signing

- OPC-pkg built-in digital signature mechanism
- Covers package contents as a whole
- Any conformant tool can verify — no vendor-specific knowledge
- Tools must reject packages that fail signature validation

PKI hierarchy

- ODVA acts as root CA
- Member organizations operate as subordinates
- Alternatively: globally trusted certificate chain
- Enables open, verifiable trust model

Per-image integrity

- Integrity verification required for each firmware image
- Mechanism not mandated — vendor choice
- Manifest declares the method used
- Tool performs check without vendor-specific knowledge

Security is a first-class design concern — the update mechanism itself must not become an attack vector.

© 2026 ODVA, Inc.
All rights reserved.

Transition Path for Existing Devices & Vendors

Cannot be reused

- Existing proprietary packaging formats
- Existing transfer protocols (even CIP-based ones)

Can be preserved

- Internal firmware binary format
- Existing internal update procedures
- Vendor-specific authenticity checks on-device

Constrained devices and tool-resident logic

Some devices rely on tool-resident update logic with minimal on-device logic. Specification development will need to investigate how these devices can be accommodated within the standardized interface.

The externally visible interface is standardized. What happens inside the device remains under vendor control.

© 2026 ODVA, Inc.
All rights reserved.

Online Repository: Firmware Discovery



Four-attribute lookup:	Vendor ID, Device Type, Product Code, Revision — the same the tool reads from the Identity Object
Version selection:	Latest available firmware or any prior version (e.g., for rollback)
Air-gap support:	Tool can pre-cache firmware for installations without internet connectivity
Cross-vendor visibility:	Browse available firmware across all vendors from one interface
ODVA hosted:	Reasonable assumption: all conformant devices already subject to ODVA testing and certification

© 2026 ODVA, Inc.
All rights reserved.

Related Work

TUF

The Update Framework Specification v1.0.34

General-purpose framework securing software update systems. Role-separated metadata (Root, Targets, Snapshot, Timestamp) limits impact of key compromise. Addresses arbitrary installation, rollback, freeze, and replay attacks. Technology-agnostic — foundational reference rather than a deployable industrial solution.

Uptane

Standard for Design and Implementation 2.1.0

Extends TUF for automotive ECUs. Two-repository model (Director + Image), conformance requirements for OEMs. Addresses heterogeneous resources, intermittent connectivity, and safety-critical systems. Relevant parallels to multi-image EtherNet/IP device scenarios.

ENISA

Technical Advisory on Secure Update Mechanisms (2026)

Practical guidance for CRA-compliant update mechanisms. Seven-step lifecycle model: build, publish, check, retrieve, verify, install, record. Controls covering signing, metadata freshness, anti-rollback, and observability. Explicitly references TUF and Uptane as consistent frameworks.

This proposal complements these works — a standardized CIP interface and package format through which these security properties can be realized within the ODVA ecosystem.

© 2026 ODVA, Inc.
All rights reserved.

Conclusion

Three-pillar framework

CIP Objects & Services · OPC-pkg container · JSON manifest — any conformant tool can update any conformant device.

- **Scales end-to-end:** From a single-image constrained sensor to complex multi-processor controllers
- **Transition-friendly:** Vendors retain internal update mechanisms — only the external interface is standardized
- **Security first:** Container integrity, PKI authenticity, and per-image verification built in from the start
- **Regulatory aligned:** Supports CRA, NIS2, and IEC 62443 compliance for end users across the facility

If a secure update mechanism is formally defined within the CIP Networks Library, TUF, Uptane, and the ENISA advisory warrant thorough evaluation as part of that effort.

© 2026 ODVA, Inc.
All rights reserved.

Future Work

1	**JSON manifest schema**	Define the full JSON Schema for manifest validation and publish as part of the specification
2	**Conformance testing**	Specify conformance test requirements for both devices and update tools
3	**Device architecture validation**	Work with ODVA members to validate approach against a broader range of device types
4	**PKI & certificate management**	Define the PKI hierarchy and processes around certificate management for member organizations
5	**ODVA online repository**	Define responsibilities, processes, and SLAs for ODVA-hosted firmware repository
6	**Generic update tool**	Determine whether ODVA should provide a reference/community firmware update tool
7	**Related framework evaluation**	Thorough evaluation of TUF, Uptane, and ENISA advisory guidance for any formal specification work

© 2026 ODVA, Inc.
All rights reserved.

Next Steps: ODVA Standardization Path

1

Member interest

Gather feedback from ODVA members and the broader EtherNet/IP community

2

Project charter

Create a project charter addressing scope, responsibilities, and ODVA obligations

3

MAC &TRB approval

Submit charter to the MAC and Technical Review Board for approval

4

SIG

Hand charter to relevant Special Interest Groups to drive specification work

Charter responsibilities for ODVA

- Development and ongoing maintenance of the online firmware repository and its associated processes
- Whether ODVA should provide a generic firmware update tool for use by the community

© 2026 ODVA, Inc.
All rights reserved.

Join the Conversation

ODVA members and the broader EtherNet/IP community are encouraged to:

- Engage with this proposal and provide feedback from implementation experience
- Evaluate the approach against your own device architectures
- Contribute to the standardization effort through ODVA's SIG process

© 2026 ODVA, Inc.
All rights reserved.


2026
ODVA
Industry Conference and Annual Meeting