Device Integration Standards for EtherNet/IP

Paul Brooks
Senior Manager, Open Architecture Management
Rockwell Automation, Inc.

Wolfgang Hoeferlin
Technology Manager
Endress+Hauser

Sean Vincent
Director of Technology Programs
FieldComm Group

Joakim Wiberg
Director of Technology
ODVA, Inc.

Presented at the ODVA
2025 Industry Conference & 23rd Annual Meeting
March 19, 2025
Clearwater Beach, Florida, USA

Abstract

Following the acquisition of FDT technology by FieldComm Group (FCG), the industrial automation community has the opportunity to enhance our technologies to allow a single device integration standard to be used through discrete, hybrid and process automation disciplines. Double work on business logic and user interface for a device across different technologies and for use in different applications can be eliminated.

In this paper we outline the use cases that FCG together with ODVA, PNO and OPC Foundation wish to address. We will look at some of the initial technical assumptions that allow this work to dovetail into device description improvements already underway within ODVA. We will discuss the framework that will allow ODVA members to contribute to, and benefit from this work.

Keywords

FDT, FDI, DTM, Device Integration, Device Management, Industrial Automation

Definition of terms

AAS: Asset Administration Shell

API:	Application Program Interface
CIP:	Common Industrial Protocol
DTM:	Device Type Manager
EDD:	Electronic Device Description
EDDL:	Electronic Device Description Language
EDS:	Electronic Data Sheet
FCG:	FieldComm Group
FDI:	Field Device Integration
FDT:	Field Device Tool
iDTM:	interpreter DTM
IIoT:	Industrial Internet of Things
IT:	Information Technology
OPCF:	OPC Foundation
OT:	Operational Technology
PNO:	PROFIBUS Nutzerorganisation e.V.
SDO:	Standards Developing Organization
SIC:	Strategic Integration Committee
UI:	User Interface
UID:	User Interface Description
UIP:	User Interface Plug-in

The merger and high-level objectives for FieldComm Group

In 2023, initial discussions were initiated between the FDT Group and the FieldComm Group, with the goal of exploring potential collaboration. Figure 1 provides a visualization of the timeline of the merger. The primary aim of these talks was to establish a unified organization focused on streamlining device integration across factory, hybrid, and process automation industries. These early conversations continued throughout 2023 and led to the formation of several specialized study teams. Each team was tasked with investigating specific aspects of the potential merger and how it could enhance industry standards and integration efforts. The study teams successfully completed their work at the start of 2024, providing valuable insights that helped shape the direction of the proposed collaboration.

In April 2024, based on the findings from the study teams, a term sheet was drafted to formalize the potential agreement. This document outlined the key terms and conditions of the merger, serving as the foundation for the next steps. The term sheet was presented to the FDT Group's members for approval, and in May 2024, it received unanimous support. This paved the way for the formal announcement of the merger, which was publicly disclosed in a press release on June 10, 2024. The merger marked a significant milestone in the industry's ongoing efforts to create a more cohesive and integrated approach to device management and automation solutions across various industrial sectors.

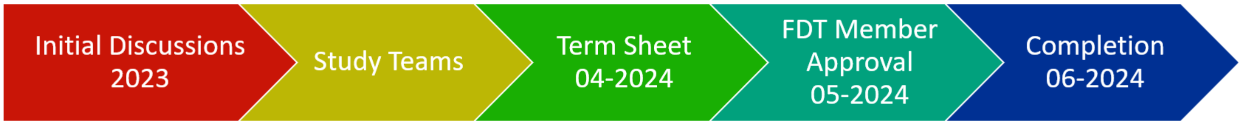


Figure 1 FDT Group and the FieldComm Group merger timeline

The vision behind this initiative is to create a unified framework that bridges the worlds of factory automation, hybrid automation, and process automation. This common technology view aims to streamline the integration of various systems and devices across industries, enabling smoother communication and interoperability between previously siloed automation environments. At the same time, it is crucial that the transition to this unified framework does not disrupt the existing business models or operational structures within these industries. To achieve this, the approach must be evolutionary rather than revolutionary, ensuring continuity and minimizing disruption during the integration process. This is especially important for technologies like Field Device Integration (FDI) and Field Device Tool (FDT), which have long been integral to automation systems.

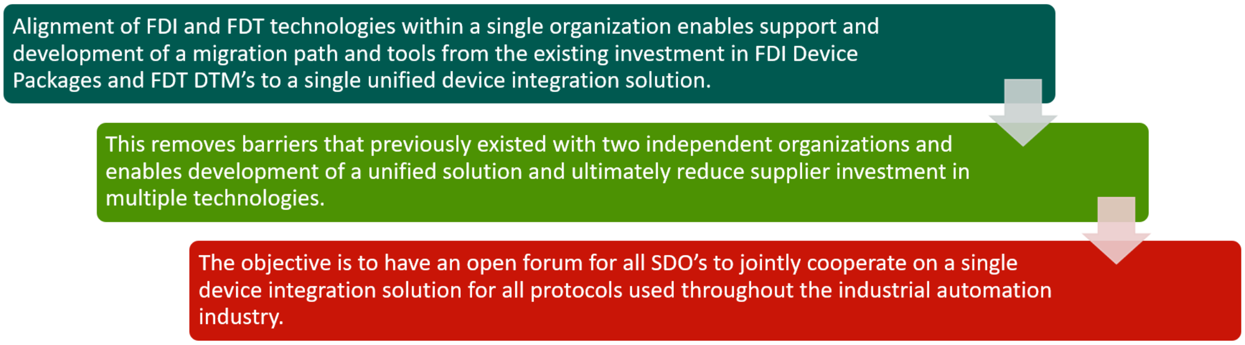
The necessity of an evolutionary approach stems from the reality that many of the existing technologies, standards, and tools in use across factory, hybrid, and process automation were not initially designed to work in today's rapidly evolving digital landscape. As the Industrial Internet of Things (IIoT) continues to expand, and as automation systems become increasingly connected, there is an urgent need to modernize the software environments that support these systems. ODVA, a leader in the development of industrial automation standards, has been at the forefront of advocating for the convergence of Information Technology (IT) and Operational Technology (OT). This convergence is essential for enabling data interoperability, supporting advanced services, and fostering mobility across devices and systems.

However, as we look at this convergence, it becomes clear that many of our existing automation technologies are not adequately equipped for the demands of these new digital ecosystems. This includes both legacy protocols and the infrastructure supporting them, which often lack the flexibility and scalability needed to integrate seamlessly with modern software systems. Therefore, the discussions among industry leaders emphasized the need for a comprehensive solution to these challenges - a solution that would involve collaboration across multiple organizations and technological domains.

By the time the discussions reached a conclusion, there was a strong consensus that these issues were too significant and complex to be solved by FDT Group and FCG alone. Although both organizations bring valuable expertise to the table - FDT being protocol-agnostic and able to support a wide range of device integration, and FCG having deep knowledge of process automation-specific protocols—tackling the broader challenge of integrating factory automation, process automation, and hybrid systems required a more expansive and inclusive ecosystem.

This broader ecosystem needed to encompass a diverse set of Standards Developing Organizations (SDO) that represent the various industrial communication protocols. These include EtherNet/IP and PROFINET, widely adopted protocols in factory automation; OPC UA FX, which supports the exchange of data in a flexible, secure, and scalable way across different industries; and Modbus, a protocol that remains essential in many industrial environments for its simplicity and ease of use. By bringing together representatives from all of these Standards Developing Organizations and ensuring that each protocol is adequately represented, the initiative could move toward creating a more cohesive, interoperable, and future-ready automation ecosystem.

In essence, the collective understanding was that the challenges facing the industry today—ranging from outdated protocols to the complexities of integrating diverse automation systems—required a collaborative approach. FCG and FDT Group, though capable and experienced in their respective areas, recognized that only through the active participation of all relevant stakeholders, including other key protocol organizations, could a truly unified and interoperable framework be achieved. This collaboration would not only address the current limitations of existing technologies but also help prepare the automation industry for the future, where data flows seamlessly between devices, systems, and platforms across different sectors and industries.



Alignment of FDI and FDT technologies within a single organization enables support and development of a migration path and tools from the existing investment in FDI Device Packages and FDT DTM's to a single unified device integration solution.

This removes barriers that previously existed with two independent organizations and enables development of a unified solution and ultimately reduce supplier investment in multiple technologies.

The objective is to have an open forum for all SDO's to jointly cooperate on a single device integration solution for all protocols used throughout the industrial automation industry.

Figure 2 FDI and FDT technology alignment

The alignment of FDI and FDT technologies, see Figure 2, under a single organization provides significant advantages, particularly in terms of supporting and developing a seamless migration path from existing FDI Device Packages and FDT/DTM's to a unified device integration solution. This approach enables:

- Support for the current installed base of devices, hosts, and communication networks, ensuring they continue to function effectively.
- Enhanced integration capabilities for the existing installed base, making it easier to connect with future, harmonized technologies and systems.
- Streamlined device management, where a single device package can be used throughout the entire lifecycle of the plant, simplifying maintenance, upgrades, and management.

The primary objective is to unify these two integration technologies around a shared vision and target outcome. Previously, the existence of two separate organizations was a barrier to achieving this goal. By bringing both technologies under one umbrella, we eliminate internal competition and ensure that the newly formed organization is fully committed to serving the best interests of both ecosystems.

However, if the goal of a single device package that works with all hosts and uses one protocol cannot be achieved, then the broader ambition of unifying all device packages across multiple protocols will be unfeasible. The success of this integration hinges on the ability to simplify and standardize the device package across the entire system.

The benefits of consolidating into a single organization eliminate the barriers that existed between two independent entities, enabling the development of a unified solution. This, in turn, reduces the need for suppliers to invest in multiple technologies.

To achieve this unified solution and decrease the overall investment required by the industrial automation community, FCG undertook an internal reorganization. A new management team, called the Strategic Integration Committee (SIC), has been appointed to oversee this transition. This team now holds day-to-day responsibility for all integration technologies, and they will manage both the existing and the new working groups that will be established in early 2025. Their key tasks include guiding the direction of these groups and ensuring the delivery of their objectives:

- A dedicated steering body that will focus on device integration strategy and its implementation, independent of any specific fieldbus technology.

- The eventual convergence of all efforts into a single working group will drive the development of a unified integration solution starting early 2025.

- A progressive reduction of redundant investments in technology, tools, and certification, leading to more streamlined and efficient operations.

The goal is to create an open forum that encourages collaboration among all SDOs to develop a unified device integration solution for all protocols used in the automation industry, see Figure 3.

- Protocol-Agnostic Technology: Ensure the development of technology that is agnostic to any specific protocol, making it adaptable and embraceable by all relevant field protocol organizations, including FCG, PROFIBUS Nutzerorganisation e.V (PNO), ODVA, OPC Foundation (OPCF), and others.

- Collaborative Development: The technology will be jointly developed and shared among peer SDO stakeholders, fostering a cooperative approach across the industry.

A key responsibility of the new committee is to ensure that the FCG-managed integration solution maintains an active ecosystem that goes beyond just FCG-managed communication technologies. It is vital that EtherNet/IP has an equal representation alongside protocols like HART-IP and PROFINET. The ultimate aim is to promote cooperation and alignment among all peer stakeholders in the industry.

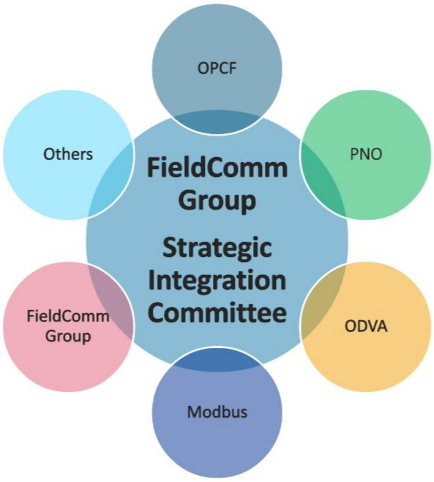


Figure 3 SDO’s jointly cooperating

FDI Primer

The most fundamental aspect of FDI technology is the device package itself, which is designed to provide all the necessary information for the device to function seamlessly within an automation system. The core component is the Electronic Device Description (EDD) file, which was initially developed for communication protocols such as HART, FOUNDATION Fieldbus, and PROFIBUS PA. The EDD serves similar purposes as the EDS in certain respects but has a few key differences:

- Limited Real-time Exchange Information:
  Unlike EDS, the EDD does not provide as much information regarding the real-time communication between the controller and the device. EDS focuses more on the structure and communication protocols, whereas the EDD emphasizes device-specific information.

- Richer Business Logic:
  One of the significant advantages of the EDD is its ability to contain more sophisticated business logic (METHODS). This provides a deeper level of configuration and operational control compared to EDS, allowing for more complex interactions and calculations that can be executed on the device.

- Flexible User Interface:
  FDI packages consist of a User Interface Description (UID) and, optionally, a User Interface Plug-in (UIP). The UID serves as the foundation for the user interface, defining its basic structure, layout, and essential elements. EDS doesn’t provide any means for the device vendor to provide an user interface.

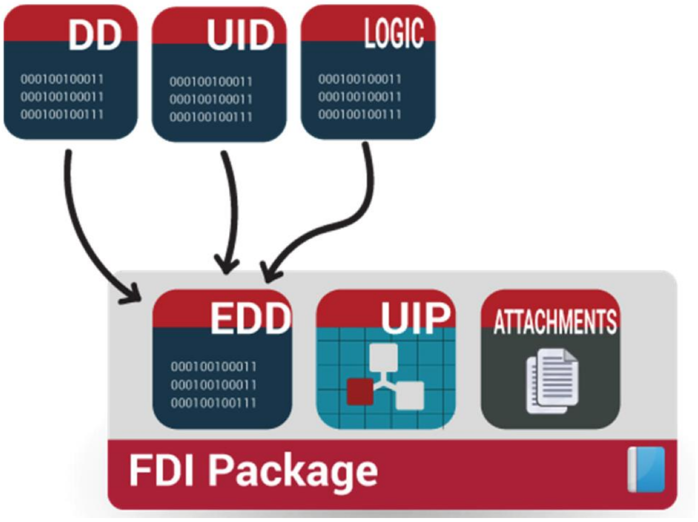


Figure 4 FDI device package contents

An FDI Device package is a comprehensive software package that device vendors must supply along with their instruments, ensuring compatibility and functionality within the FDI framework. This package includes several essential components that work together to describe the device's operation, user interaction, and business logic. Figure 4 provides a visualization of the FDI package contents. The key elements of the FDI device package are:

1. Electronic Device Description:
   The EDD is a crucial component written in Electronic Device Description Language (EDDL), based on IEC 61804, which provides detailed information about the device's parameters, configuration, and capabilities. It is essential for enabling communication between the device and host systems.

2. User Interface Description:
   The UID is a text-based description of the device’s user interface, also written in EDDL. It defines the structure and presentation of how users interact with the device, making it easier to set up, configure, and monitor the device through a graphical interface.

3. User Interface Plug-in:
   The UIP is an optional component that, when provided, must be written using HTML5. It allows device vendors to create a customizable, interactive user interface, offering greater flexibility for the display and interaction with the device. However, there is no guarantee that hosts will support the UIP, so its use depends on system compatibility. The UIP, when included and supported by hosts, enhances the user experience by enabling the development of highly flexible and interactive interfaces. Utilizing HTML5, the UIP allows for advanced customization, rich graphical elements, and dynamic content, providing users with an intuitive and efficient way to interact with complex devices.

4. Business Logic:
   Business logic, often referred to as METHODS, outlines the operational behavior of the device. This includes defining parameter dependencies, calculation algorithms, and other dynamic operational aspects. These are also written in EDDL and enhance the device’s functional capabilities.

5. Catalog File:
   The catalog file is a mandatory element of the FDI package. It contains fundamental information about the device, such as its model number, version, manufacturer details, and other metadata. This file plays a key role in identifying the device and ensuring its proper integration into the network.

6. Attachments:
   Attachments such as PDF documents, certificates, and other supporting materials can also optionally be included in the FDI package. These are not mandatory but may provide additional documentation and certification for the device.

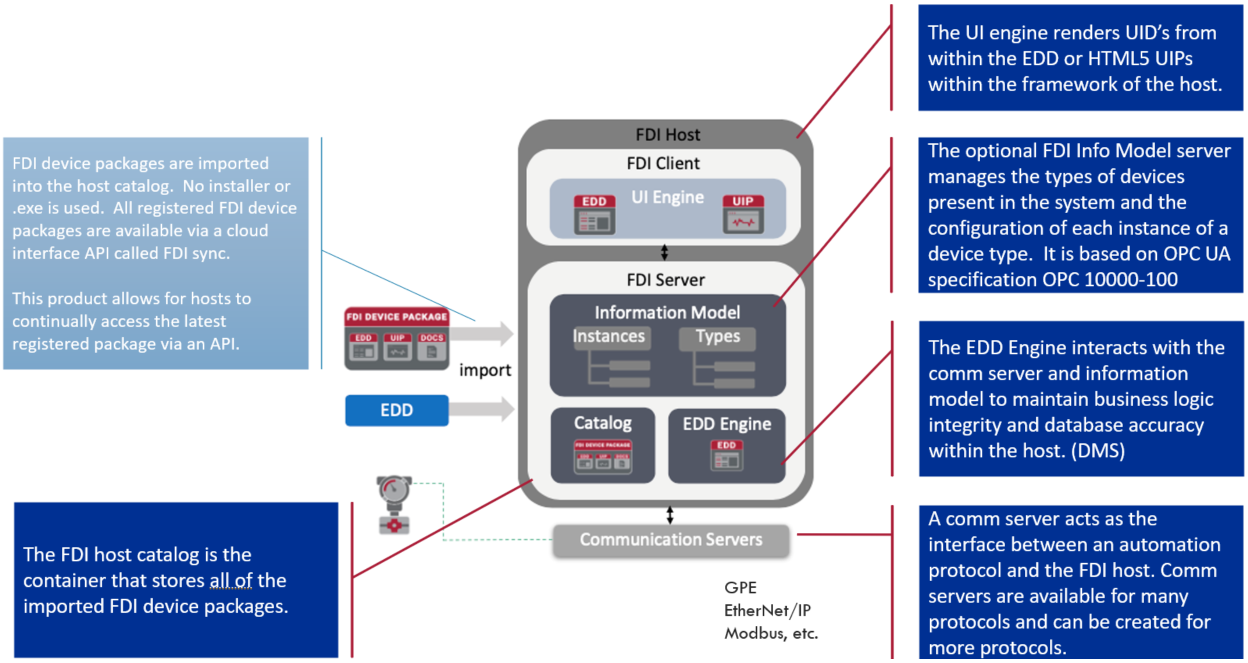
From the perspective of a host system, FDI packages give vendors the ability to implement capabilities that are currently out of scope for EDS.

One of the most notable aspects of FCG’s support is that the FDI package is included in the conformance testing process. Once tested, FCG archives the FDI packages in the cloud repository known as FDI Sync, providing a well-defined Application Program Interface (API) that allows host vendors to automatically retrieve these packages for instruments they have never encountered before.

Additionally, FCG provides communication drivers for both its own technologies and those of its partners. For example, an EtherNet/IP device directly connected to the network can be supported by a communication server co-developed by FCG and ODVA. However, this support is limited by the lack of capabilities for bridging, routing, and EDS in the FDI architecture.

FCG also offers an EDD engine that can be deployed on any Windows-based host tool, significantly easing the development process by simplifying the handling of the EDD language and the EDDL binary format. Figure 5 provide a visualization of how an FDI host can be constructed.

Moreover, FCG provides common components for both information model servers and user interface clients, with an OPC UA interface connecting the two. However, few host systems take advantage of this setup. Many prefer a client-server model tailored specifically to their vendor's needs.



FDI device packages are imported into the host catalog. No installer or .exe is used. All registered FDI device packages are available via a cloud interface API called FDI sync.

This product allows for hosts to continually access the latest registered package via an API.

The FDI host catalog is the container that stores all of the imported FDI device packages.

The UI engine renders UID's from within the EDD or HTML5 UIPs within the framework of the host.

The optional FDI Info Model server manages the types of devices present in the system and the configuration of each instance of a device type. It is based on OPC UA specification OPC 10000-100

The EDD Engine interacts with the comm server and information model to maintain business logic integrity and database accuracy within the host. (DMS)

A comm server acts as the interface between an automation protocol and the FDI host. Comm servers are available for many protocols and can be created for more protocols.

Figure 5 FDI host visualized

FDI device packages are imported directly into the host catalogue without the need for an installer or executable (.exe) file. All registered FDI device packages can be accessed via a cloud-based interface FDI Sync. This service enables hosts to continuously retrieve the latest registered device packages through the API.

The FDI host catalogue serves as a repository, storing all the imported FDI device packages. The User Interface (UI) engine within the host framework renders UIDs from both the EDD files and the optional HTML5-based UIP.

An optional FDI Information Model server manages the types of devices within the system and oversees the configuration of each instance of a device type. This server is based on the OPC UA specification, specifically OPC 10000-100.

The EDD Engine works in conjunction with the communication server and information model to ensure the integrity of business logic and maintain database accuracy within the host system (DMS). A communication server functions as the interface between the automation protocol and the FDI host. Communication servers are available for various protocols, and new ones can be developed for additional protocols as needed.

FDT Primer

A Device Type Manager (DTM) package provides a device-specific component in accordance with the FDT3 Standard. It includes the following elements:

- DTM Package Manifest:
  A description of the package's contents and structure.
- DTM Business Logic:
  This component provides APIs and information about:
  - Device identification (both offline and online)
  - Device status (online)
  - Device parameters (both offline and online)
  - Network configuration (including support files)
  - Device process values (I/O data)
  - Device-related functions (UI and commands)
  - Documentation (for the current device as well as references to additional documentation)
- DTM User Interface(s):
  Typically implemented in HTML5.

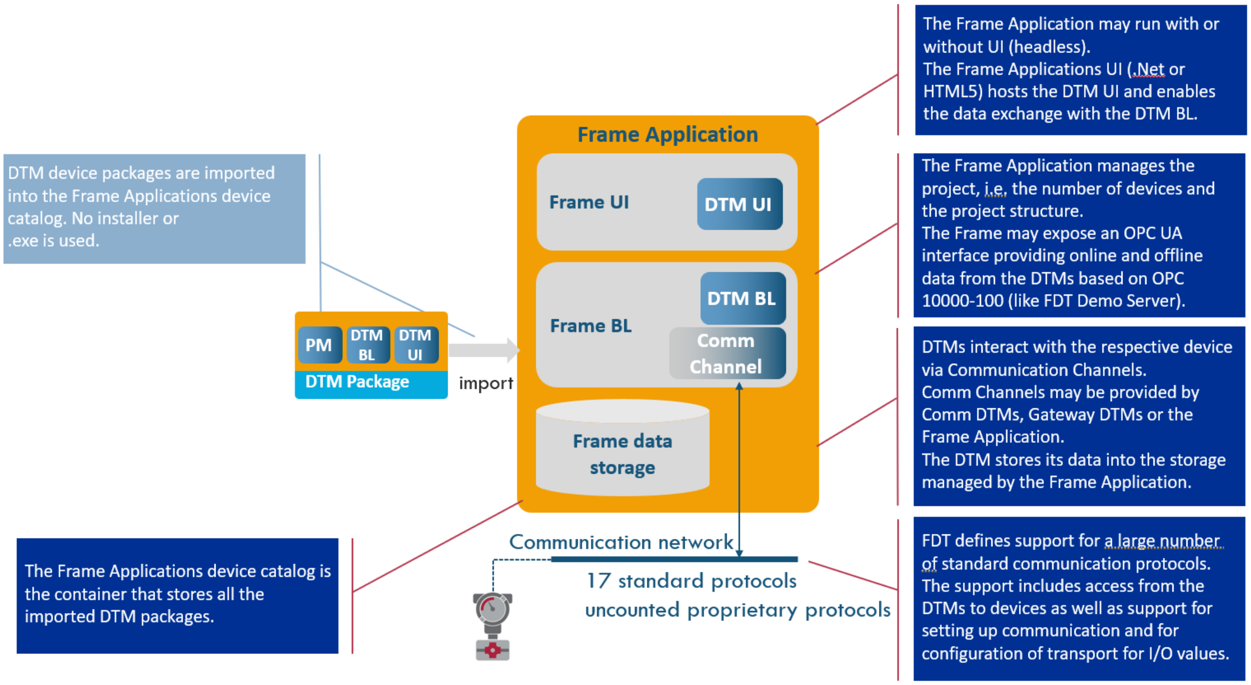
The package may also contain additional files, such as binary elements, documents (e.g., DD files), and other media related to the DTM or the device itself.

Unlike FDI, the DTM package is primarily delivered as executable code, accompanied by defined APIs and a vendor-specific user interface. This approach allows for language independence and offers maximum flexibility to the device vendor. However, except in the most recent versions, it often carries dependencies on specific operating systems.

DTM technology is designed to be protocol-agnostic, meaning it can support all capabilities of the underlying communication technology. For example, in technologies like EtherNet/IP, where both system and device configuration are described using the EDS file, a DTM can manage both aspects of configuration.

Since the DTM interface operates through an API, it is essentially a 'black box.' This means that the device’s business logic can be implemented in any programming language the developer prefers. In some cases, the same source code used for the real-world device can even be employed for the DTM.

The latest iteration of FDT technology, FDT 3.0, is optimized for cloud-native operations. While cloud-native systems can easily scale down to on-premises environments, the reverse is not always the case. A key feature of cloud-native FDT 3.0 is its ability to run business logic seamlessly, with a clear separation between business logic (language-neutral) and presentation (HTML5).



The Frame Application may run with or without UI (headless).
The Frame Applications UI (.Net or HTML5) hosts the DTM UI and enables the data exchange with the DTM BL.

DTM device packages are imported into the Frame Applications device catalog. No installer or .exe is used.

The Frame Application manages the project, i.e. the number of devices and the project structure.
The Frame may expose an OPC UA interface providing online and offline data from the DTMs based on OPC 10000-100 (like FDT Demo Server).

DTMs interact with the respective device via Communication Channels.
Comm Channels may be provided by Comm DTMs, Gateway DTMs or the Frame Application.
The DTM stores its data into the storage managed by the Frame Application.

The Frame Applications device catalog is the container that stores all the imported DTM packages.

FDT defines support for a large number of standard communication protocols.
The support includes access from the DTMs to devices as well as support for setting up communication and for configuration of transport for I/O values.

Figure 6 FDT frame application visualization

Both FDT and FDI leverage OPC UA in similar ways, with companion specifications that define how to create the information model. Harmonizing these models will be one of the first deliverables of the technical working group. Just like in FDI, the use of OPC UA information is optional and ultimately depends on the host vendor's choice, not the device vendors.

DTM device packages are imported directly into the Frame Applications device catalogue, without requiring an installer or .exe file, see Figure 6. The device catalogue acts as the container for all imported DTM packages. Frame Applications can operate with or without a user interface (headless).

When a UI is present, the Frame Applications UI (whether .NET or HTML5) hosts the DTM UI, enabling data exchange with the DTM’s business logic. The Frame Application also manages the project structure, including the number of devices and how the project is organized.

The Frame may also expose an OPC UA interface, allowing access to both online and offline data from the DTMs based on OPC 10000-100. DTMs communicate with their respective devices through Communication Channels. These channels can be provided by Comm DTMs, Gateway DTMs, or directly through the Frame Application.

DTMs store their data in the storage managed by the Frame Application. FDT supports a wide range of standard communication protocols, enabling DTMs to access devices, set up communication, and configure the transport of I/O values.

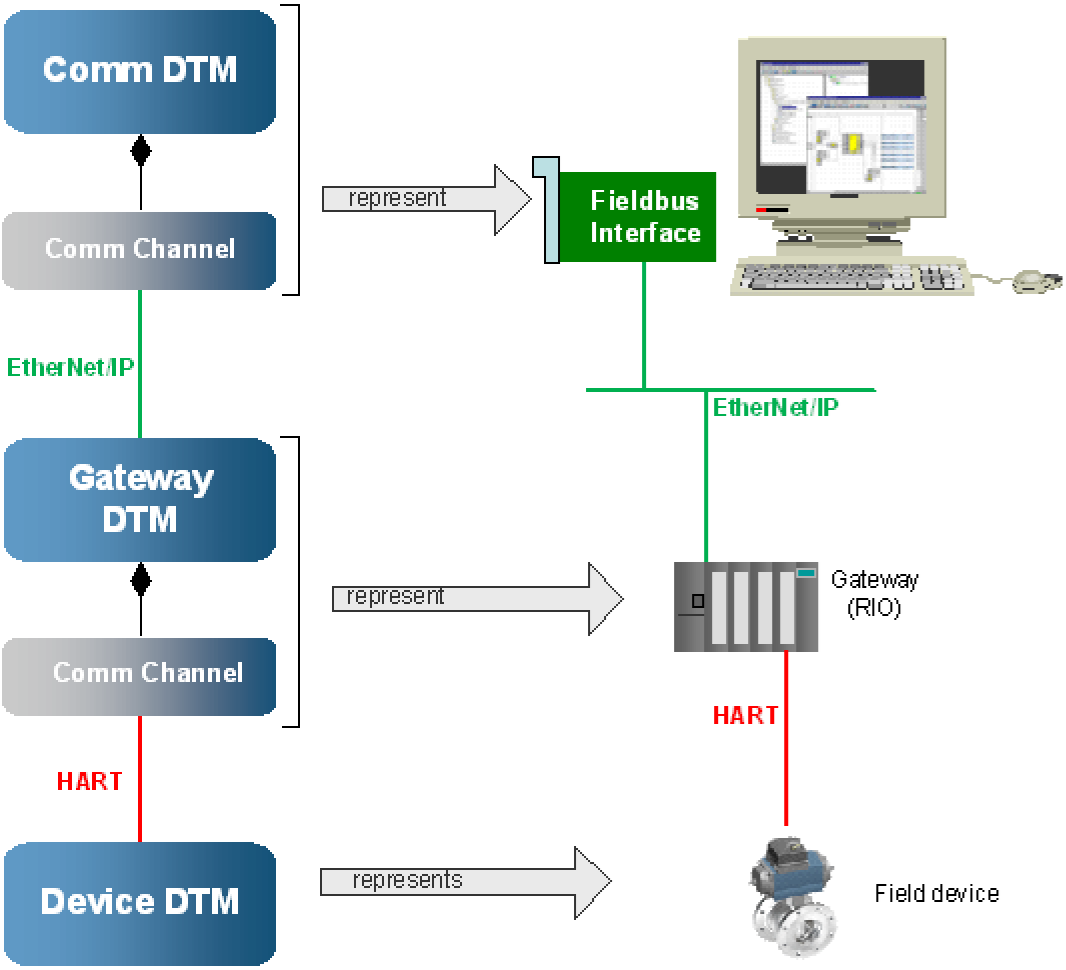


Figure 7 Nested communication

Nested communication in FDT/DTM technology, see Figure 7, refers to the concept of enabling communication between devices and systems at multiple levels, often involving a hierarchy of communication components within the FDT framework. This approach allows for more complex setups where communication between devices can be structured across several layers, often enhancing flexibility and scalability. Key Aspects of Nested Communication in FDT/DTM:

1. Layered Communication Architecture:
   FDT and DTM support a layered communication model, where communication may be initiated or routed through multiple components. This often involves communication channels that can pass messages or data through intermediary DTMs or gateways to the target devices. The communication process can thus be nested, meaning a DTM could interact with another DTM or device indirectly by routing through several layers.
2. Use of Communication DTMs and Gateway DTMs:
   Nested communication typically involves Comm DTMs and Gateway DTMs, which play a crucial role in managing data traffic between different levels of the network. These DTMs serve as intermediaries, enabling devices to communicate across different communication protocols, or over networks where direct communication is not possible. In many cases, these gateways

manage nested communication flows, translating and routing messages between various systems and devices.

3. Interfacing Multiple Protocols:
   Nested communication is especially useful when dealing with systems that operate on different communication protocols. For example, a Gateway DTM may act as an intermediary between an EtherNet/IP network and a HART device, allowing data to be transmitted across networks with different communication standards. In this way, the nested architecture of DTMs enables seamless integration of heterogeneous devices and systems.

A highly versatile application of DTM technology is the interpreter DTM (iDTM). In general, an iDTM serves as an interpreter for assets from underlying technologies, converting them into a format that can be presented to the user. It encapsulates the FDT API, see Figure 8, allowing frame applications to interact with these assets without the need to understand the underlying details.

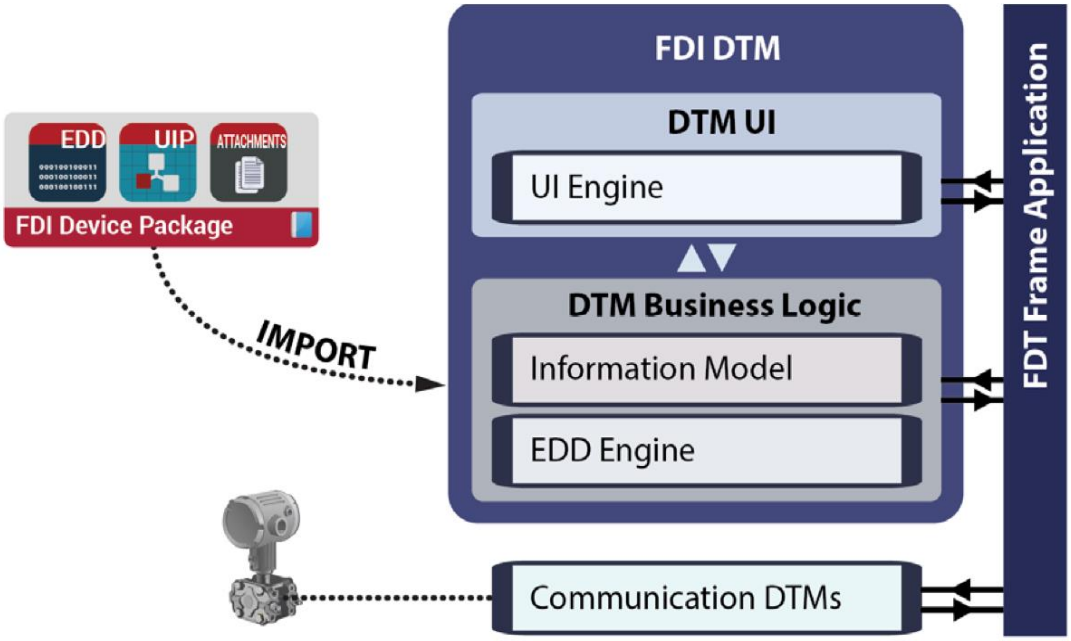


Figure 8 Interpreter DTM architecture - FDI variant illustrated

For example, with EtherNet/IP, the iDTM can handle a simple I/O block with two parameters, two I/O assemblies, and three connections, represented by an EDS file of just over 100 lines. Alternatively, it can manage a more complex energy management device with over 1,000 configuration values and an EDS file exceeding 14,000 lines. In both cases, the device can be fully configured at both the device and system level through a DTM host, without requiring the device vendor to perform any FDT-specific work.

FDI Gap Analysis for EtherNet/IP Applications

The first task for the FCG team working on integrating FDT and FDI technologies is to identify the gaps that need to be addressed in the converged technologies. This includes identifying capabilities that are currently present in FDI but missing in FDT, those that FDI lacks but are available in FDT, and the reasons why neither has become the de facto standard for Factory Automation applications.

FDI and FDT share some overlapping functionalities; however, they primarily serve different purposes, as each was designed to address distinct use cases. While both technologies offer valuable integration capabilities, they also present certain limitations - particularly in the case of FDI - when used as a device integration solution for EtherNet/IP. These challenges arise due to the inherent architecture of typical EtherNet/IP systems, which involve multiple layers of sub-networks, requiring nested communication, and the specific workflows that engineers follow during system commissioning in factory automation environments.

To address these challenges, the following section outlines key use cases that will be important for the FCG working group. Those channellings and use-cases presented are the gaps that needs to be fulfilled. This FCG working group is tasked with developing a converged integration technology that merges the strengths of both FDT and FDI. The goal is to create a unified solution that can be effectively applied across process, hybrid, and discrete manufacturing industries.

Additionally, this section provides an overview of the core differences between FDT and FDI, highlighting their distinct areas of focus. While these technologies have traditionally been utilized for different integration needs, they have both played a crucial role in asset management, demonstrating their value in industrial automation.

The team has focused on the most common use case in factory automation, which involves a plant-wide network with multiple controllers, devices, and I/O systems. Some controllers are provided as part of packaged equipment with their own isolated networks, while the software environment itself is a multi-vendor solution.

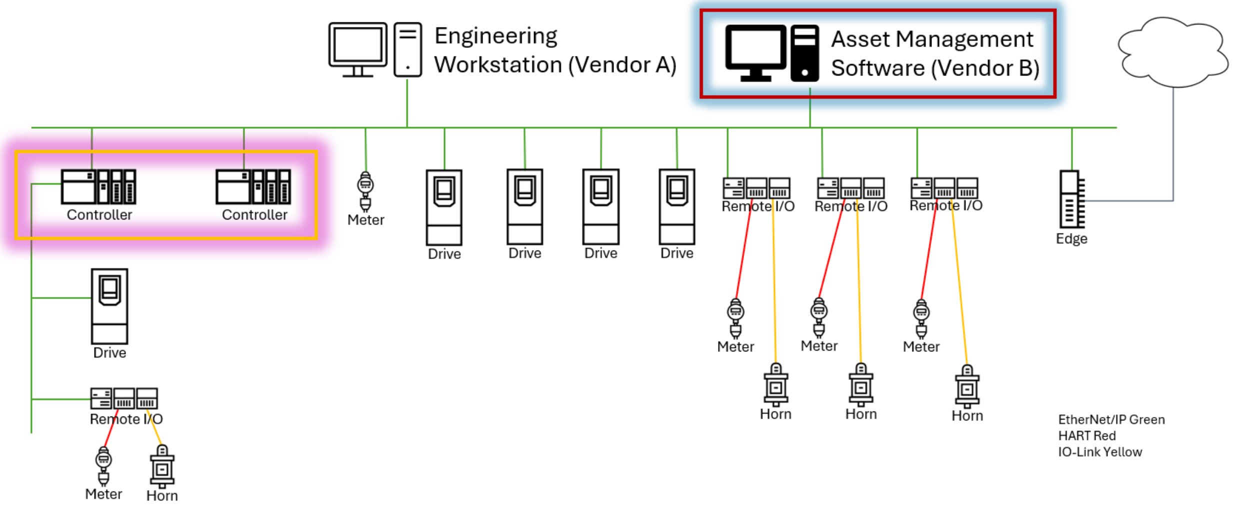
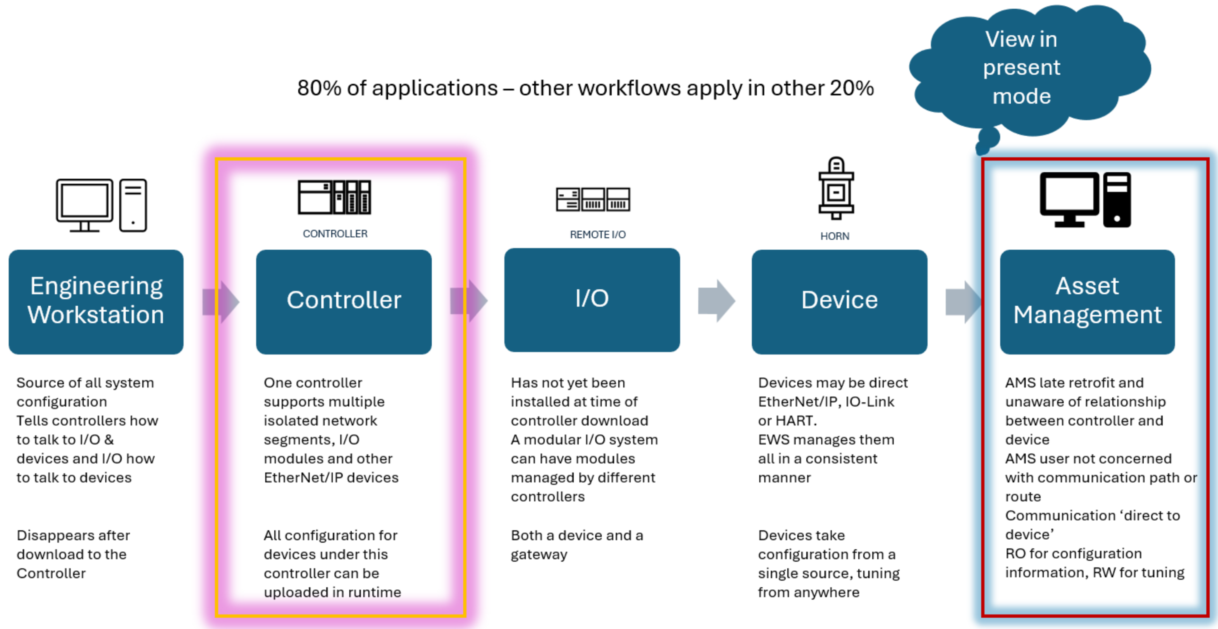


Figure 9 Typical system architecture

This use case is where FDT is most frequently utilized in factory automation, see Figure 9, with asset management tools almost always relying on DTMs for device presentation and communication infrastructure support. FDT is also often employed for device configuration within engineering tools.

A key focus in this case has been on the configuration workflow, particularly regarding the packaged control system and isolated network, consisting of a controller, drive, I/O system, meter and horn, shown on the left side of Figure 10 .



Source of all system configuration
Tells controllers how to talk to I/O & devices and I/O how to talk to devices

Disappears after download to the Controller

One controller supports multiple isolated network segments, I/O modules and other EtherNet/IP devices

All configuration for devices under this controller can be uploaded in runtime

Has not yet been installed at time of controller download
A modular I/O system can have modules managed by different controllers

Both a device and a gateway

Devices may be direct EtherNet/IP, IO-Link or HART.
EWS manages them all in a consistent manner

Devices take configuration from a single source, tuning from anywhere

AMS late retrofit and unaware of relationship between controller and device
AMS user not concerned with communication path or route
Communication ‘direct to device’
RO for configuration information, RW for tuning

Figure 10 Typical configuration workflow

The lifecycle of a device within a control system configuration begins even before the physical device or controller is installed—or perhaps even manufactured. An engineer, well-versed in the application, selects the sensors and actuators that the controller needs to interact with, along with the device configuration, particularly those settings that impact the control system’s performance. Device configuration represents only a small part of the overall control system operation. Typically, an engineering tool aims to provide a unified, consolidated view of the entire system configuration, rather than having different configuration elements scattered across multiple files and assets.

Once the system configuration is complete, it is downloaded to a controller for testing and commissioning. In most cases today, this involves using a simulated controller and simulated devices. For machine builders supplying packaged equipment, this step may involve testing the configuration on a "golden reference" machine design that remains on their premises, with all other machines being identical copies of this reference design.

After completing virtual and golden reference commissioning, the engineering workstation and engineer are no longer needed. The machine proceeds into serial production, with the design often remaining unchanged for years.

Though outside the scope of both ODVA and FCG, an essential part of the workflow involves copying the application into a new controller. This allows machine-builder business models in which a machine design goes into serial and mass production rather than each instance of a machine being a new design. Later, the I/O modules and EtherNet/IP devices appear, but they do so in a random order and at random intervals.

Below is an outline and detailed description of the current snapshot of integration technology options offered by FDI and FDT. Additionally, it includes a breakdown of the common functionalities expected from any integration technology option. To enhance clarity, the functionalities have been categorized into two main sections, Asset management functionality and other functionality. By structuring the information in this way, we aim to provide a clear and comprehensive view of the available integration technologies and their respective functionalities.

FDT Functionality with a Focus on Asset Management

- Through Interpreter DTM, the system can interface with any device utilizing EtherNet/IP, HART, and IO-Link, ensuring broad compatibility.
- The Interpreter DTM leverages the same source asset data as Engineering Workstations (EWS), supporting EDS, IODD, and EDD files for seamless integration.
- Comm DTMs facilitate connectivity across multiple network layers, including:
    - Backbone Ethernet for high-level system communication
    - Machine-local Ethernet for localized network interactions
    - HART I/O and IO-Link I/O for direct device communication
- Vendors have the capability to develop and provide custom DTMs tailored to their specific infrastructure needs, ensuring adaptability and extensibility.

FDT Functionality Enhancements

- Eliminating Windows dependency would be a major improvement, enabling business logic programming in diverse languages such as C, Python, and C#, thus expanding development flexibility and cross-platform compatibility.

FDI Functionality with a Focus on Asset Management

- Nested communication support is currently unavailable, limiting complex multi-layered interactions.
- The ability to support multi-vendor I/O is dependent on AMS suppliers, reducing flexibility in heterogeneous environments.
- EDD is a lossy translation of EDS, potentially leading to incomplete or inaccurate device data representation.
- FDI provides limited assistance for the majority (80%) of common workflows because it lacks awareness of I/O configurations and controller requirements, making it less effective for comprehensive asset management.

Comparison of FDI and FDT in Asset Management

- FDT offers a more efficient approach to AMS support, while FDI's reliance on EDD adds complexity and maintenance overhead.
- FDI requires double implementation efforts, with both UIP (User Interface Plug-in) and Business Logic (BL) needing separate development—unless integrated through an FDT iDTM, which consolidates these elements.
- FDI lacks clear separation between data, business logic, and UI, making development and maintenance more cumbersome.

Motivations for converged solution

When examining EDS, FDI, and FDT it's evident why any given stakeholder may find two of these technologies inadequate for their needs. Each was developed with distinct use cases in mind, which means each offers unique functionality that the others lack, or features that are only shared by one. For a large portion of business logic and presentation, ODVA has historically designated these responsibilities as vendor-specific, falling outside the scope of standardization.

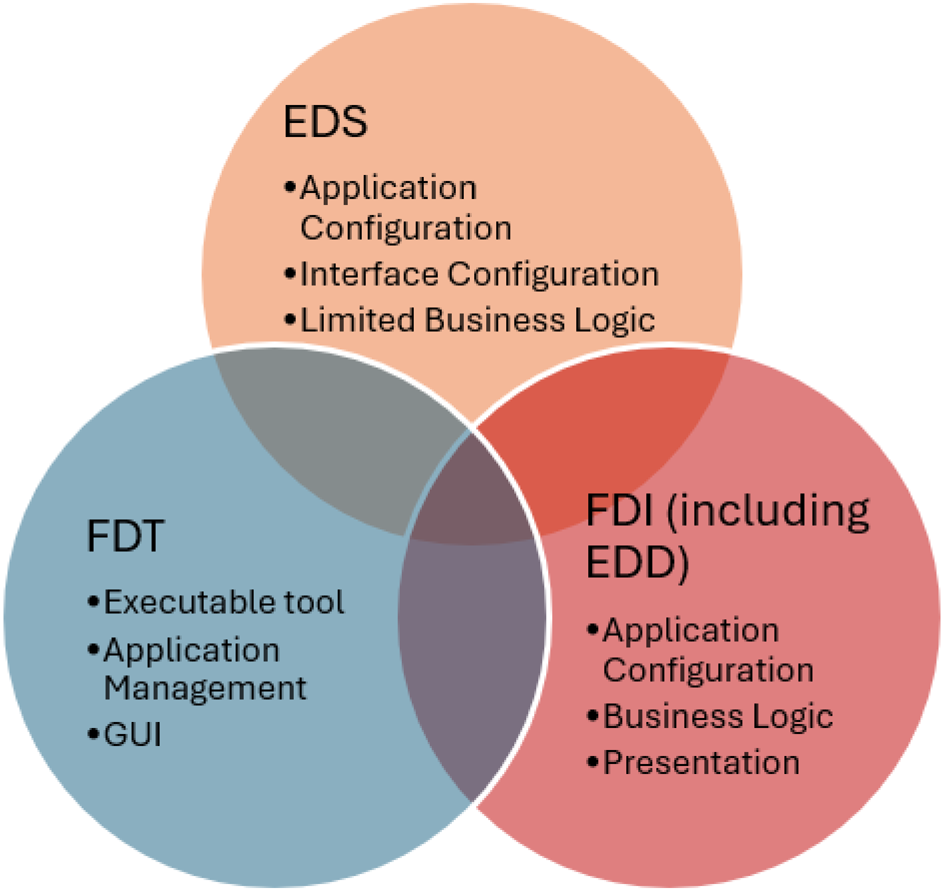


Figure 11 EDS, FDI, and FDT core features

Figure 11 offers a visual depiction of the core features provided by EDS, FDI, and FDT, highlighting the unique capabilities of each technology. The following description provides a concise breakdown of these features in text form. Both the list and the visual representation effectively illustrate the differences and distinctions between the three device integration technologies, showcasing how each one addresses specific needs and functionalities in the integration landscape. This comparison emphasizes the varying approaches and capabilities of EDS, FDI, and FDT, offering insight into their roles within modern automation and device management systems.

EDS core features:

- Application Configuration: The ability to set up and manage the configuration of applications within a system, ensuring compatibility and efficient integration.
- Interface Configuration: Configuring the interfaces between devices and software systems, enabling seamless communication and interaction.
- Limited Business Logic: Basic business logic functionality to support fundamental operations within the system, though with more limited capabilities compared to other technologies.

FDI (including EDD) core features:

- Application Configuration: Similar to EDS, FDI also supports the configuration of applications, but with more advanced features to accommodate complex device integration scenarios.
- Business Logic: FDI offers a higher level of business logic, allowing for deeper, more intricate workflows and decision-making processes that support complex industrial automation tasks.
- Presentation: A robust presentation layer that visualizes the data and configurations in a user-friendly interface, making it easier for engineers to interact with the system.

FDT core features:

- Executable Tool: A powerful tool that enables the execution of system tasks, offering flexibility and control over the automation process.
- Application Management: Comprehensive management of applications, supporting installation, configuration, and maintenance of software and device drivers across different platforms.
- GUI (Graphical User Interface): An intuitive and user-friendly graphical interface that simplifies interaction with the system, making it easier to manage devices, monitor performance, and troubleshoot issues.

However, from the user’s perspective, these differences become a challenge in a multi-vendor environment. Users care about all the various use cases, and it’s essential to recognize the need for commonality wherever possible. By harmonizing these technologies, we can simplify the tool vendor’s experience and help them to simplify their users’ experience throughout the entire device lifecycle. After all, the device itself remains unchanged over time, and it is crucial that it provides a single asset compatible with all tools, regardless of the underlying technology.

This issue becomes increasingly critical as cybersecurity risks to control systems grow, and as the maintenance and revision cycles for devices continue to shorten. With the rise of more frequent updates and faster deployment of devices, traditional security approaches no longer suffice. Control systems, which were once isolated and offered a much smaller attack surface, are now more interconnected and exposed to potential threats, increasing the cybersecurity attack surface. In this rapidly evolving landscape, our control systems need to be more prepared for modern security practices such as DevSecOps and compliance with industry standards like IEC 62443. DevSecOps, which integrates security into the entire lifecycle from development to decommissioning and operational processes from the very beginning, is essential for ensuring the ongoing protection of systems from cradle to grave. However, many organizations struggle to adopt these practices, leaving gaps in security that can be exploited.

Moreover, many of the technologies we rely on today were developed long before the advent of Google or the smartphone era. As the boundaries between IT and OT continue to blur, it's time to shift our focus. Instead of just improving communications, we must begin to think about information processing. The world is moving towards cloud-native software, and we must prepare for that transformation, embracing new approaches that are flexible, scalable, and capable of meeting the demands of a digital-first future.

A device can be brought to market with multiple communication interfaces, making it applicable across various market segments and compatible with any host, all while maintaining a single integration interface for both presentation and business logic. This means that a host engineering tool can integrate and manage a device without being concerned about its underlying communication protocol. It’s important to note that this doesn’t imply all hosts support all protocols, or that all protocols yield identical results, but rather that the common functionalities are handled in a unified way.

As an ODVA community, we must approach our role in industrial automation with humility. Today, very few devices are designed for a single communication protocol. Multi-protocol devices, whose business logic is defined and exposed via web-based user interfaces, function the same way regardless of the communication protocol. Just as users expect seamless integration of all devices into their host systems, device vendors seek to have their devices integrated into any host using a single asset, ideally leveraging common elements, irrespective of the communication protocol used.

Technology SDOs ensure interoperability through conformance testing, ensuring that devices and hosts work together for both communication and integration. This does not diminish the importance of interoperability - ODVA and EtherNet/IP have earned a solid reputation for promoting interoperability through conformance testing. However, as we move forward, there may be a need to extend this focus to integration to better meet the evolving needs of users, addressing both current and future requirements.

The success of the SIC will be measured by the ability to achieve seamless device and system commissioning, ensuring cybersecurity requirements are met, while providing full lifecycle management and decommissioning within a fully integrated network. All of this must be accomplished without sacrificing ease of use.

The automation industry stands to gain significantly from the efforts of the SIC. As the industry faces an increasing skills gap and a shortage of expertise, our systems must become easier to use. Through collaboration, we can simplify the technology, creating the potential for greater harmonization and lowering barriers for industry adoption.

The vision is for a single association to manage and drive the harmonization of integration technologies, providing a foundation for new industry applications, such as tools, services, second-path access, and Asset Administration Shells (AAS). All leading controller companies must support this new integration technology in their engineering tools.

At the end of the day, users don’t care about the technology or the tech stacks behind it - they simply want these complexities to be invisible. The task, as a development community, is to focus on creating a simplified user experience and developing technologies that foster ease of use throughout the device lifecycle. If this makes the end users' tasks easier, that’s the true measure of success.

But we must also make our own work easier. If we end up duplicating effort, creating technologies that take significantly longer to implement without clear benefits, we will have failed. As vendors, we too face a skills gap and need to ensure our technology makes it easier to recruit and train skilled workers. The implementation of new integration technologies should not create additional burdens for device vendors. For example, a vendor whose device currently requires just 100 lines of EDS code should not feel that the new integration technology will require a disproportionate amount of additional work.

Ultimately, the true measure of success is adoption. We need to develop technology that host, and engineering tool vendors want to incorporate into their solutions because it helps them deliver greater value to their users with less effort.

Long-term vision

The long-term goals outlined in this section reflect an early consensus that has been developed as a result of the work carried out by the SIC over time the group has existed. These objectives are a product of ongoing collaboration and dialogue among stakeholders, and as such, they represent a shared vision that has evolved through collective input. However, it is important to note that, like any project or initiative involving diverse stakeholders, these goals may evolve as new insights emerge and further discussions take place. The flexibility to adapt and refine these goals is inherent in the process, and the final outcomes may shift as more information comes to light.

Also, the opportunity to actively engage in shaping the direction of the project is available exclusively to those who are directly involved in the collaborative efforts. Stakeholders who participate in the ongoing work have the ability to influence and contribute to the shaping of these goals, ensuring that the process remains dynamic and inclusive.

First and foremost, it is essential to have a unified package that works seamlessly across both factory automation and process automation. This package should be versatile enough to cater to the unique requirements of both sectors, while maintaining the flexibility needed to address the diverse range of devices, systems, and protocols used in these environments. In today’s rapidly evolving industrial landscape, the boundaries between factory automation and process automation are becoming increasingly blurred. Both sectors are integrating more advanced technologies, from IoT and cloud computing to artificial intelligence and machine learning, and as a result, there is a growing need for a single, comprehensive solution that can operate across both domains.

One of the key areas of alignment is the necessity to separate the different layers of the device integration stack, allowing software tools to interact with each layer independently. This structure enables flexibility—simple devices that only handle data can function without requiring business logic or a user interface, while more complex devices, such as modular I/O systems with customizable modules and intricate design rules, can be fully characterized and managed. A critical feature is the ability to operate business logic headlessly, meaning it can run without a user interface. The integration layers can be broken down as follows:

1. Data-only devices:
   Devices that provide only data (parameters) without any added logic or interface.
2. Data + Business Logic (B/L) devices:
   Devices that include both data and the necessary business logic.

3. Data + B/L + User Interface devices:
   Devices that integrate data, business logic, and a user interface for user interaction.
4. Data + B/L + UI + Dynamic devices:
   More sophisticated devices, such as modular I/O systems or multivariable devices, which feature dynamic configurations and complex design options.

The Common Industrial Protocol (CIP) family of protocols inherently supports nested or bridged-and-routed communications. However, when adding protocols like IO-Link or HART, a secondary communications driver becomes necessary. This capability, available in FDT but not in FDI, must be preserved to ensure continued flexibility and support for different communication technologies.

One of the strengths of the FCG is its provision of common components that simplify development, especially for host suppliers. It is essential that the tools optimized for protocols like HART and PROFIBUS/PROFINET are extended to natively support CIP. By doing so, the development process can remain consistent, efficient, and adaptable across a range of protocols.

Ultimately, this brings us to the importance of conformance testing. FCG and ODVA must collaborate to ensure that both hosts and devices meet a common minimum standard for conformance, regardless of the specific technology they support. This collaboration is vital for ensuring seamless integration and interoperability across a wide variety of devices and systems.

Rockwell Automation is proposing an API-First approach, see Figure 12, that focuses on flexibility and interoperability across different platforms and device models. Rather than being tied to a specific communication technology or device model, this approach emphasizes the interaction between the host application and device packages via well-defined APIs. These APIs allow for seamless integration between the host application and various components, enabling different deployment options, all of which can be containerized for scalability and efficiency.

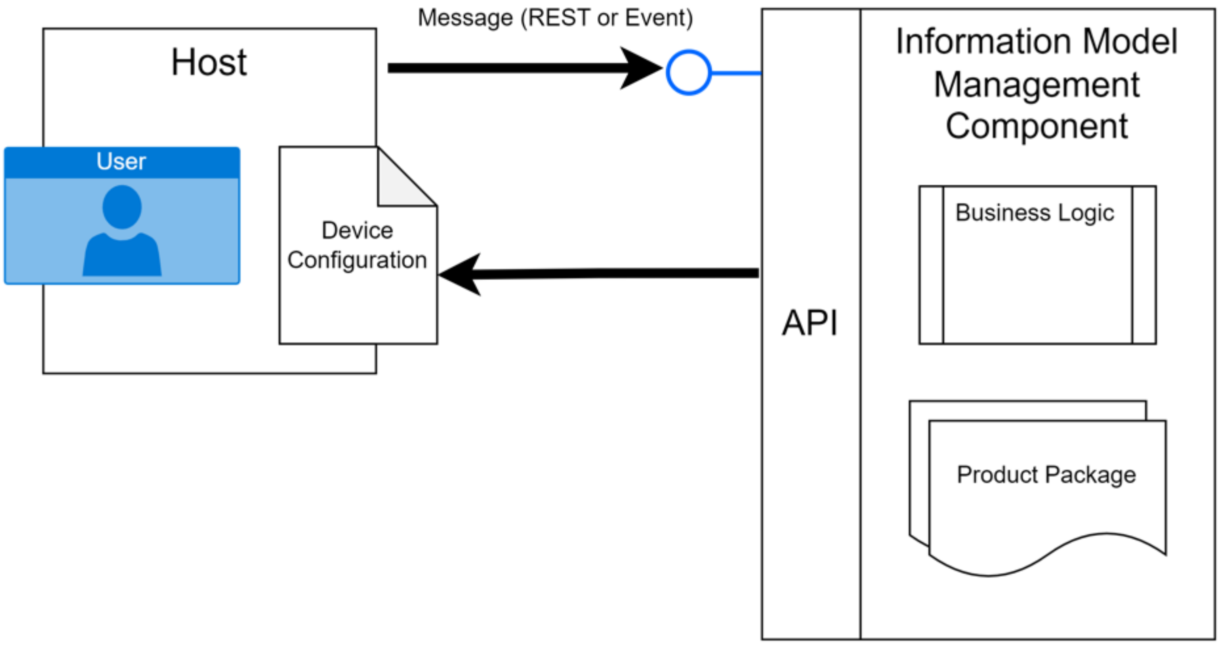


Figure 12 Well defined APIs in an API-First approach

This API-First architecture is designed to support highly adaptable deployment patterns, ranging from cloud-native solutions to mobile apps and desktop applications. By breaking away from the operating system dependencies that have historically limited integration technologies—including those in ODVA—this new approach aims to offer unprecedented flexibility. The host application, which interacts with the APIs, can take a variety of forms, such as:

- A webpage
- A mobile application
- A desktop application
- A compiler

The system's Information Model Management Component operates independently and can be self-contained in multiple deployment formats, such as:

- A container
- An executable file
- A website
- Serverless functions
- WebAssembly

This architecture not only decouples the core components from specific hardware and operating systems but also provides robust support for modern deployment environments. The use of APIs and containerization allows Rockwell Automation to future-proof its solutions, enabling greater adaptability across different industries and use cases. By fostering this level of flexibility, Rockwell Automation is positioning itself to address the growing demands of modern industrial automation, where agility and scalability are critical.

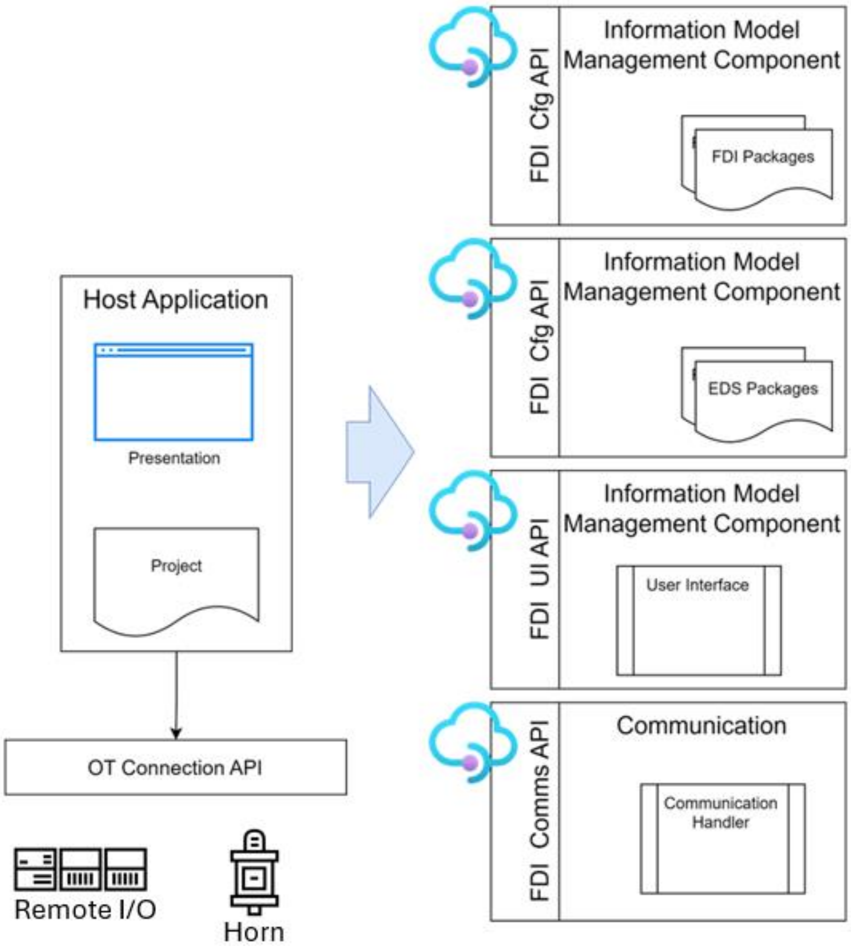


Figure 13 API-First approach put in context

In the context of an API-First approach, the solution is designed not as a monolithic application but as a collection of independent services that all implement the same API. This architecture offers significant flexibility, scalability, and security, as outlined in Figure 13 and the following summary:

- Stateless: The system does not maintain state between interactions, ensuring that each request is independent and can be processed in isolation.
  - Instances of Devices Aren't Saved: Device states or instances are not stored, which allows for a more dynamic, flexible system that can easily scale and adapt to different environments.
  - Same Input Produces the Same Output: Consistency is a key feature, as providing the same input will always result in the same output, ensuring predictability and reliability.
  - Enables Scalability: The stateless nature and standardization of APIs allow the system to scale easily, handling everything from a single device to large networks of hundreds of plants without issues.
- Zero Trust: Security is paramount, and a zero-trust architecture ensures that no entity is inherently trusted. All communication must be authenticated, reducing the risk of unauthorized access.

    - Credentials Required at All Interfaces: Every interface within the system requires proper authentication, ensuring that only authorized users or devices can interact with the services.

- Platform Independent: The API-First approach allows for compatibility with various platforms, making it versatile and adaptable to different operating environments.
    - Different Services Per Platform, Same API: While the underlying services may differ across platforms (e.g., mobile, desktop, cloud), the API remains consistent, ensuring seamless interoperability and integration.
- No Direct Connection to the OT Network: This ensures that the system remains secure by avoiding direct exposure to the operational technology (OT) network, thereby minimizing potential vulnerabilities.
- Deployment Flexibility: The solution supports multiple deployment options, technologies, and platforms, offering versatility in how it is implemented and maintained.
    - Technologies and platforms: The solution supports multiple options, technologies, and platforms, and is self-contained in various formats such as containers, executables, websites, serverless functions, or WebAssembly. It is platform-independent, meaning it can run on any environment without being tied to a specific deployment location. Whether deployed as a webpage, mobile app, desktop app, or compiler, the host application remains flexible and adaptable to the platform in use.
    - Versionable: The APIs and services can be versioned, allowing for backward compatibility and smoother transitions during updates or migrations.

By defining a comprehensive set of APIs, fieldbus organizations can take responsibility for their own device packages. For example, the FCG can provide containers to manage FDI packages, while ODVA containers can handle EDS packages. This approach allows device vendors to create containers that are tailored to their specific needs and in the programming language of their choice. It also enables communications to be sandboxed, keeping device and host operations secure and isolated.

The stateless operation ensures that the system can scale seamlessly, whether managing a single device or supporting operations across hundreds of plants. The zero-trust architecture promotes cyber-resiliency by requiring authentication at every step, reducing the risk of unauthorized access. Additionally, revision management can follow modern DevSecOps best practices, enabling efficient version control, continuous integration, and ensuring that security is integrated throughout the development lifecycle.

This API-First approach provides a robust, scalable, and secure foundation for managing industrial automation and device integration, offering the flexibility needed to meet the demands of modern manufacturing environments.

****

The ideas, opinions, and recommendations expressed herein are intended to describe concepts of the author(s) for the possible use of ODVA technologies and do not reflect the ideas, opinions, and recommendation of ODVA per se. Because ODVA technologies may be applied in many diverse situations and in conjunction with products and systems from multiple vendors, the reader and those responsible for specifying ODVA networks must determine for themselves the suitability and the suitability of ideas, opinions, and recommendations expressed herein for intended use. Copyright ©2025 ODVA, Inc. All rights reserved. For permission to reproduce excerpts of this material, with appropriate attribution to the author(s), please contact ODVA on: TEL +1 734-975-8840 FAX +1 734-922-0027 EMAIL odva@odva.org WEB www.odva.org. CIP, Common Industrial Protocol, CIP Energy, CIP Motion, CIP Safety, CIP Sync, CIP Security, CompoNet, ControlNet, DeviceNet, and EtherNet/IP are trademarks of ODVA, Inc. All other trademarks are property of their respective owners.