WHITEPAPER

Building the Connectivity Architecture for the Industrial Internet of Things

Executive Summary

Have you struggled to effectively apply conventional connectivity solutions to your Industrial Internet of Things (IoT) application? Do you feel like you're re-inventing the wheel every time you have to deal with multiple states as you add new applications? Do you end up creating your own custom application-level protocols? Will your connectivity solution be flexible enough to accommodate future requirements, data types, and staffing changes?

The Industrial IoT introduces new requirements for the velocity, variety, and volume of information exchange. Connectivity must be real-time and secure, and it must work over mobile, disconnected, and intermittent links. It must efficiently scale to handle any number of things, each of which may have its own unique requirements for information exchange, such as streaming updates, state replication, alarms, configuration settings, initialization, and commanded intents. These requirements are above and beyond the requirements commonly handled by conventional connectivity solutions designed for static networks.

This white paper describes a connectivity architecture model for the Industrial IoT and shows how the Data Distribution Service (DDS) messaging standard can address the unique requirements of the Industrial IoT. Concepts will be illustrated using a scalable Industrial IoT application architecture that shares real-time data between mobile devices and the cloud and provides browser-based access from thin clients. The optimal use of web-sockets technology is also described in the architecture overview.

Industrial Internet Systems – What's Different?

The Internet originally evolved to connect human beings, regardless of their physical location and computing environments. The current Industrial Internet of Things (IoT), in contrast, is meant to connect devices and systems. There are several important differences with these non-human users of the Industrial IoT:

- They operate in the "real world" where nonstop mode is typical – and where, for example, shutting down systems and power for maintenance or upgrades is not option.
- Failures can trigger severe consequences, including loss of life or property or both.
- They rely on data timeliness; the right answer delivered too late becomes the wrong answer.

The uniqueness of today's Industrial Internet of Things (IoT) can also be viewed in terms of both technical and business requirements. Technically, many criteria for industrial systems are much more stringent, including:

- Scalability, for accommodating growth of volume and variety of devices
- Performance, measured as speed (velocity) and timeliness
- Resilience, or the combination of availability, recovery, and durability
- Security, which can require authentication, authorization, integrity checks, confidentiality rules, and non-repudiation

Equally stringent business requirements include:

- Reliability, which measures the likelihood of failure during the system's lifetime
- Safety, which calls for avoidance of unintended consequences
- Longevity, and the ability to perform incremental upgrades on an ongoing basis
- Diversity, which can be achieved with independent developers and multiple technologies

In light of these significant differences, the Industrial IoT requires unique connectivity technologies designed for intelligent devices. A common pitfall is applying human-driven or request-response architectures and technologies when connected devices and systems call for data-driven or publish-subscribe approaches.

An Open Interoperable Connectivity Architecture

As the Internet continues to evolve and address the unique requirements for industrial devices and systems, designers and standards organizations are fueling the advancement of appropriate connectivity standards. Successfully choosing the most suitable standard starts with an understanding of the architectural role, model, rules, and patterns of connectivity in current industrial systems.

Architecture Role: Reducing Integration Times

Within the Industrial IoT, one of the primary roles of the connectivity architecture is to ensure interoperability and thereby reduce integration time for complex devices and subsystems. Ultimately, the goal is to evolve the connectivity architecture to achieve full plug-and-play compatibility (see Figure 1).

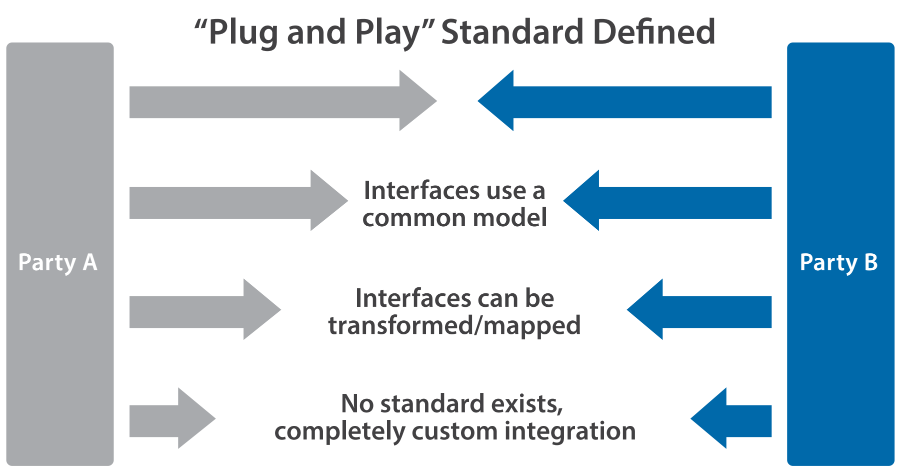

Figure 1. Levels of interoperability within a connectivity architecture

Currently, industry standards for real-time connectivity are focused on mid-level interoperability, or syntactic-level compatibility, where all endpoints and systems use a common data format and syntax.

Architecture Model: Connectivity Gateways

A connectivity standard that delivers syntactic-level interoperability facilitates the introduction of connectivity gateways (see Figure 2) to address the diversity of devices in modern systems. These gateways serve multiple purposes, including the support of external systems and devices that rely on other connectivity technologies. Gateways can also be used to create hierarchical architectures and to group various endpoints and devices into subsystems.

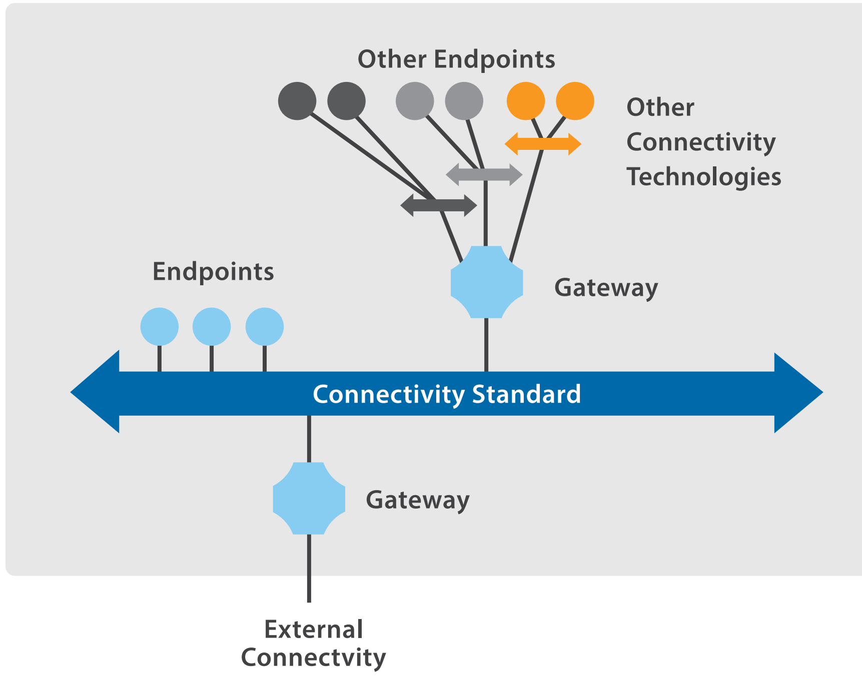

Figure 2. Connectivity gateways

Data-Centricity Rule: Decoupling Apps from Data

Unlike human-driven environments, industrial systems operate autonomously and therefore call for a data-driven architecture. This shift can be compared to the historical development of databases. By decoupling data from applications, databases gave application developers much greater flexibility for evolving modular, independent applications, and they fostered innovation and standards in the application programming interface (API).

Within the Industrial IoT, data-centric communications can similarly promote interoperability, scalability, and ease of integration. The concept of a data bus allows the possibility of decoupling data from application logic so application components interact with data and not directly with each other. The data bus can independently optimize the delivery of data in motion, and can also be more effectively managed and scaled separately from the application components.

Data Flow Patterns: Fundamental Building Blocks

In conventional enterprise IT environments, the data architecture deals with events, transactions, queries, and jobs. The Industrial IoT, which is made up of a broad range of devices, differs greatly from this human-driven environment. The fundamental building blocks of the Industrial IoT include streams of data, commands, status (or state) information, and configuration changes.

Note that the key activity triggers within conventional environments involve human requests or responses (decisions). In the Industrial IoT, activity is triggered by data or state changes that exist and happen autonomously.

Connectivity Architecture Realization: DDS

The Data Distribution Service (DDS) standard uniquely addresses all of these architectural considerations for the Industrial IoT. Published and managed by the Object Management Group (OMG), the world's largest systems software standards organization, DDS provides an open industry standard for data-centric connectivity (see Figure 3) with more than 12 implementations available today.

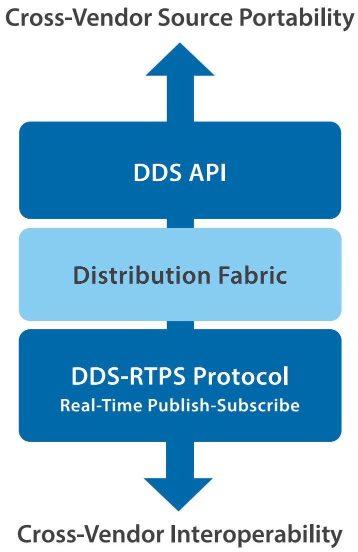

Figure 3. DDS Connectivity Model

DDS provides an API for applications, enabling portability across languages and implementation vendors. More importantly, an extensible wire protocol means that DDS enables backwards compatibility; new components can be added without requiring changes to legacy equipment.

Currently, more than a dozen DDS implementations have propagated the standard into hundreds of system designs in healthcare, transportation, communications, energy, defense, and other industries.

The success of DDS in the Industrial IoT stems from the standard's ability to connect everything everywhere with a shared data model and open data bus. Seamless data sharing can be achieved with DDS regardless of proximity, platform, language, physical network, transport protocol, and network topology.

DDS can act as the core connectivity model within large, complex systems. For maximum interoperability between subsystems, the DDS Routing Service introduces a gateway capability. This feature makes it possible to organize data bus hierarchies, and it enables bridging or mediation between different data models, protocols, or security domains. The DDS Routing Service can also isolate and filter content, which provides the ability to scrub sensitive information.

The data-centric model, combined with the flexible gateway functionality, makes DDS an ideal realization of a connectivity architecture for today's Industrial IoT. The leading DDS implementations provide all the fundamental building blocks for this architecture and create a model for seamless data sharing at all levels, from sensors and edge devices to the cloud.

Building the Connected Architecture

A Generic Use Case

Many domains utilize a common connectivity use case. Figure 4 illustrates "generic" connectivity, where a user has both local and remote access to a variety of intelligent connected devices and systems.

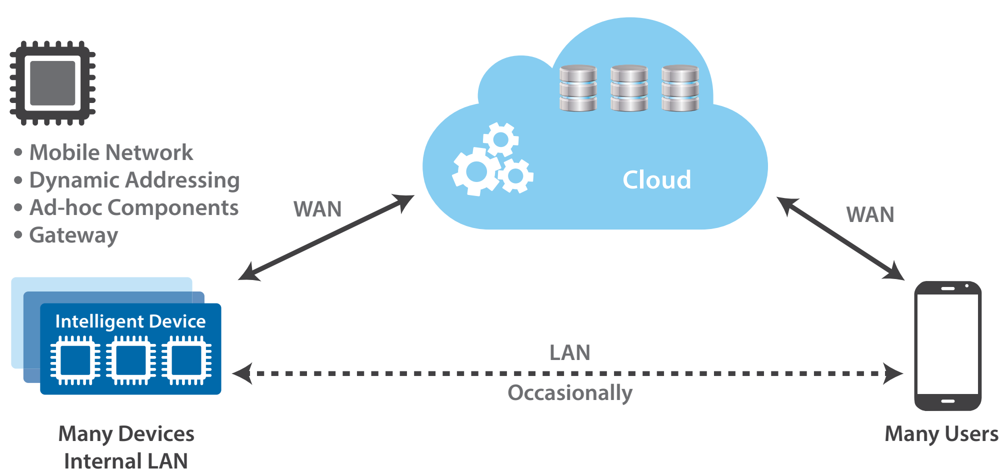

Figure 4. Generic connectivity use case (devices, LAN/cloud, and many users)

Within a connected home, the devices can include smart thermostats, lighting controllers, security cameras, and more. Residential environments have access to relatively stable LAN networks and addressing technologies to support ad hoc components. Service providers offer gateways to the cloud, and thereby make it easy for many users to remotely tap into their connected home systems.

The same basic model applies in the energy industry. The users in this case are the operators overseeing energy turbines at multiple locations. These smart devices and subsystems can be connected over stable LANs and WANs to allow local or remote operations management.

In healthcare environments, the connectivity model encompasses the smart devices at the patients' bedsides as well as laboratory equipment, with doctors and clinicians given access to data via the cloud or onsite LANs. In this and other domains, mobile networking and dynamic addressing may be introduced for more flexible user and device connectivity options.

Architecture Mapping

The generic use case highlights common connectivity scenarios, which can each be evaluated in terms of the ideal connectivity technology. Based on location and function, the right connectivity technology must be chosen for:

- Smart devices (endpoints)
- Device-to-cloud connections
- Connectivity within the cloud
- User connectivity (cloud-to-user)

Between the devices and the cloud (WAN connections), DDS provides an ideal solution with:

- Stateful interactions
  - o Intelligent connections/disconnects, and the ability to resend only relevant data upon reconnection
  - o Intelligence built into the bus, without application overhead
- Many data-flow patterns, for meeting current and future requirements
- Publish-subscribe architecture style that is data-driven
- Scalability, performance, resilience, and security

Inside the endpoint devices themselves, DDS has been applied broadly for the same reasons listed above for device-to-cloud connections. Additionally, DDS makes it possible to design smart devices that operate very reliably and meet safety and longevity requirements in industries such as healthcare and automotive. DDS also supports diversity of transports and platforms within a system, as previously discussed in terms of gateway capabilities and routing services.

DDS has also made inroads in the cloud. Here, the requirements span a broader range and give rise to a mixture of connectivity options. DDS can support this connectivity diversity, and it can also promote longevity of cloud solutions.

In contrast, other technologies make more sense for the user-to-cloud WAN connections (see Figure 5). At this point in the connectivity model, traditional web technologies such as web sockets and HTTP meet the human-centric requirements with:

- Stateless interactions
- Single data-flow pattern (query)
- Request-response architecture style that is human-driven
- Established scalability and security infrastructure
- Forgiving performance and resilience (including easy-to-restart connections and applications)
- Ubiquitous access from mobile devices or thin clients

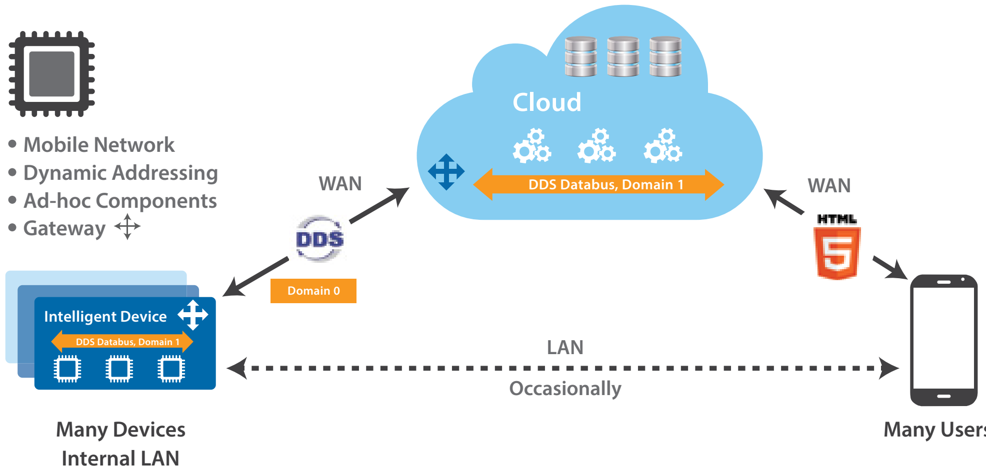

Figure 5. Mapping the right technologies into the connectivity architecture

Deployment Flexibility

DDS domains make it easy to isolate subsystems with individual data communication planes. Besides facilitating security rules with logical separation, domains also make it possible to tailor endpoint discovery rules and activity levels and significantly reduce network bandwidth and CPU/memory overhead over gateway connections. As shown in the previous diagram, for example, DDS domains can be defined with:

- Domain 0 on the WAN connections. Within Domain 0, discovery can be limited to detection of the gateway endpoints and routing services. (These gateways act as proxies for the endpoint devices in their realms.)
- Domain 1 encompassing the devices and the cloud. Full device discovery can be carried out on the device and cloud buses.

DDS also supports a choice of transports, including UDP, TCP, shared memory, OpenSSL (TLS/SSL, DTLS), and low-bandwidth connections. For example, in the generic use case, the DDS connectivity between devices and the cloud can utilize DDS over TCP. Typically, transport guidelines are different for:

- LANs: Use UDP (with multicast, if available). This applies within a cloud or for application-to-application communications.
- WANs: For device-to-cloud communications, use TCP (with TLS).

DDS is being adopted for this last category to provide remote access to any DDS data bus. DDS can manage state for seamless data-sharing and switching between cellular and Wi-Fi networks. State is managed independently of the network mobility and switching, and DDS Quality of Service (QoS) can introduce resilient rules for distributing and managing state information.

Finally, for cloud-to-human communications (mobile user endpoint devices or thin clients), you can use traditional web sockets and HTTP(s) (over TCP).

For an online demonstration of remote access from web applications, download a free interactive Web Shapes demo.

To Learn More

RTI Connext DDS, the leading DDS implementation, is at the forefront of the Industrial Internet revolution:

- Leading implementation of DDS
- C, C++, C#/.NET, and Java APIs
- Tools to monitor, debug, test, visualize, and prototype distributed applications and systems
- Adapters to integrate with existing applications and IT systems

For more information about RTI Connext DDS Pro package for Windows and Linux, and how to design smarter and more secure control systems, visit the RTI website.

To get started with DDS, download a free trial of the RTI Connext DDS solution.

Your systems. Working as one

CORPORATE HEADQUARTERS
232 E. Java Drive
Sunnyvale, CA 94089

Tel: +1 (408) 990-7400
Fax: +1 (408) 990-7402
info@rti.com

www.rti.com