WHITEPAPER

A Converged Approach to Standards for Industrial Automation

A NEW ARCHITECTURE PROPOSAL TO INTEGRATE OPC UA, DDS & TSN

AUTHORS

Costantino Pipero
Founder & Chief Technologist, Beeond, Inc.

Gerardo Pardo
CTO, Real-Time Innovations (RTI)

Fernando Garcia
Senior Software Engineer, Real-Time Innovations (RTI)

ABSTRACT

Suppliers and end users need a complete solution to address the complexity of future industrial automation systems. These systems require:

- Interoperability to allow devices and independent software applications from multiple suppliers to work together seamlessly
- Extensibility to incorporate future large or intelligent systems
- Performance and flexibility to handle challenging deployments and use cases
- Robustness to guarantee continuity of operation despite partial failures
- Integrity and fine-grained security to protect against cyber attacks
- Widespread support for an industry standard

The industry needs a clear, technically optimal, proven standards approach. This document proposes a new technical architecture to build this future. The design combines the best of the OPC Unified Architecture (OPC UA), Data Distribution Service (DDS), and Time-Sensitive Networking (TSN) standards. It will connect the factory floor to the enterprise, sensors to cloud, and real-time devices to work cells. This proposal aims to define and standardize the architecture to unify the industry.

OPC UA VALUE PROPOSITION AND ISSUES

OPC UA is the prevalent technology in industrial automation for moving data and information from field devices to the cloud. At the fundamental level, OPC UA focuses on creating a distributed object model where end users can integrate devices into working automation systems. OPC UA, along with its companion specifications, provides an environment for understanding and connecting devices. Importantly, it also allows introspection and control of the overall system through the OPC UA "information model." The OPC UA information model allows the information to be organized hierarchically, integrating devices into servers. Servers expose their information (data elements) through services which describe the operations that can be called on them. Clients (e.g., Human Machine Interface [HMI] and other tools) can browse the object model, read/write data elements, invoke methods and configure or manage a system using service calls.

OPC UA offers a powerful "client-server" model for applications (typically clients) to interact with a system. However, the client-server model is not well suited for the real-time information flows needed for real-time control and peer-to-peer communications (e.g. controller-to-controller). To date these real-time flows use multiple field bus protocols that do not interoperate with each other.

The desire to have OPC UA extend to these real-time communications is driving an initiative to create a new OPC UA publish-subscribe extension. This extension would define new publish-subscribe interfaces and UDP-based protocols, which could then be combined with the emerging TSN standard to create a technology with capabilities similar to the traditional field buses.

DDS VALUE PROPOSITION AND ISSUES

DDS is a standard connectivity framework developed by the Object Management Group® (OMG®) that is widely deployed across multiple industries under the Industrial IoT (IIoT) umbrella. At its fundamental level, DDS provides an integration framework for software applications. It is used in thousands of deployed use cases, ranging from large power systems to autonomous vehicles. These systems combine intelligent software with physics-speed sensors and actuators. In these systems, DDS serves as both a control bus and edge-to-cloud connectivity framework.

DDS offers a powerful publish-subscribe databus. The DDS databus creates a secure global data space that provides a common data model to all devices and applications. The databus and common data model decouple applications from one another: Applications interact with the databus rather than with each other; they publish or subscribe to Topics, read or write data object Instances, and so on. DDS provides security at the data layer. Applications need permission to join the global data space and require fine-grained permission to access any Topic or data object they want to read or write. In this way applications are decoupled and secured, allowing for very scalable and elastic information distribution.

DDS also supports service-oriented communication, providing the tools needed to define, discover and invoke remote services.

DDS is an excellent fit for peer-to-peer communications and real-time information flow. It is implemented without any brokers or servers, unlike other publish-subscribe technologies like MQTT or AMQP. It has rich quality of service (QoS) capabilities, enabling each information flow to be tailored to its needs. For example, QoS can be set for reliable and durable for events and alarms, as opposed to best efforts and volatile for periodic sensor readings. Moreover, the functionality and security of DDS is transport-independent. It can therefore be deployed over WANs, multicast-enabled LANs, and TSN networks. Of course, in order to achieve real-time determinism, the transport needs to fulfill its part, which is why the use of DDS with TSN is a sensible choice for critical real-time loops.

DATA AND INFORMATION MODELING IN INDUSTRIAL IOT

Information modeling is key to the interoperability of future Industrial IoT systems. The OPC UA information model is fundamentally object-oriented. Servers contain nodes which can refer to other nodes in the same server or other servers. Nodes can contain data fields that can be remotely accessed as well as methods that can be remotely called. Conceptually, the whole system can be seen as a large graph of objects. Every device, subsystem, or even an entire factory can be represented with nodes in this model. Nodes include extensive meta data that define device configuration and vendor-specific information. This enables device interoperability and generic tools that can view and interact with the system operation.

Conceptually, each device or system brings its own OPC UA object model along, realized as one or more OPC UA servers. A system integrator hooks clients to the OPC UA servers at deployment or runtime. It is expected that client applications will need to discover and connect to the servers explicitly, find the information they need, and interact with them.

DDS also contains powerful information modeling capabilities. However, instead of using an object-oriented model, it uses a data-centric model where data and services are global concepts that are not tightly coupled to a specific device. Therefore, because all data and services are available on the databus, the only information needed by the application is the domain ID, the topic (or service) name, and the key that identifies the specific data-object (or service instantiation). In this model, applications are not expected to connect to any servers or specific nodes. They just send their request to the databus, which takes care of discovering applications that are connected to it and getting information to the right place(s). This means discovery is automatic; data and services can flow to multiple locations; applications can join, leave or change locations and IP addresses; and so on. The databus automatically manages the correct flows.

So which model is best? Object-oriented client-server or data-centric publish-subscribe? Naturally, each of these models has its sweet spot.

- The object-oriented model is a good match for device configuration where each vendor can define and expose its own information model. The object-oriented model is also useful as a way to provide a simple consolidated interface to a more complex subsystem (e.g., an external machine or plant interface). This model fits systems whose topology remains fairly static during operation.
- The data-centric model is a good match for loosely-coupled systems that need to operate more autonomously. It can handle changes of topology and system reconfiguration without impacting the running applications. It is optimal for publish-subscribe and real-time control through its direct serverless communication for critical data flows.

Complex systems need both approaches. By their very nature, complex systems have many components and flows. They have devices, real-time critical loops, subsystems with fairly static components as well as interactions that are more ad-hoc and dynamic. For example, the components bolted into a robot will be static during deployment, but in the case of mobile robots, their internal behavior and their interactions with external robots and agents may change dynamically.

The industry needs a common integrated solution. The natural instinct is to extend one of these models to cover its gaps. However, this is not the best solution:

- The current OPC UA pub-sub extension approach to extend the OPC UA object-oriented client-server model with “bolted on” publish-subscribe concepts cannot deliver the performance or scale necessary for complex environments: the lack of data-centric databus will result in limited facilities that do not scale.
- Likewise, extending the DDS databus “decoupled” data-centric model of pub-sub Topics and Services to “bolt on” a more tightly-coupled object-oriented model would not provide the rich feature set currently available in OPC UA.

By defining a DDS data model that matches the OPC UA information model, DDS can talk to OPC UA client-server systems. The OMG has defined an OPC UA/DDS Gateway Specification that presents an initial mapping of OPC UA and DDS. With this, IIoT systems gain the necessary ability to integrate devices with fieldbus-like performance, while including the data modeling capabilities that are required for larger system integration.

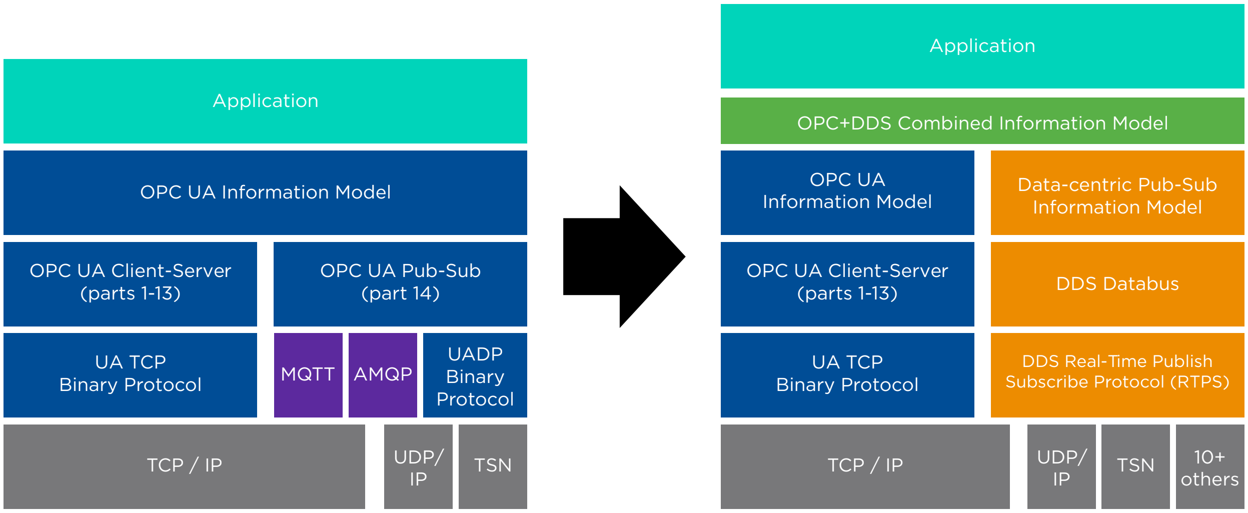

Figure 1: The diagram to the right presents an alternative to the OPC UA pub-sub model through a proven, extensible DDS design. It will scale to thousands of nodes, eliminate dependence on servers, provide flexible physical layer implementation and enable fine-grained security. On the right, the OPC UA object-oriented model is merged with the DDS data-centric model. With this design, both client-server and pub-sub will be based on years of field-proven technologies and standards with industry-leading capabilities.

NEW OPC UA - DDS ARCHITECTURE PROPOSAL

There is an opportunity to unify the industry around a proven technical solution that combines the best of OPC UA and DDS. OPC UA and DDS can co-exist and interoperate – and they already do. The OMG OPC UA/DDS Gateway Specification provides a mapping of the information models that allows pure OPC UA and DDS applications to exchange information with pure DDS applications. The intent of this white paper is to expand beyond this gateway specification with a more comprehensive integration proposal.

OPC UA needs tested, highly-capable publish-subscribe technology. Therefore, we propose to implement the OPC UA information model in DDS. This provides an alternative to the current OPC UA pub-sub approach, replacing it with a DDS-based publish-subscribe design. The result will bring proven software and control integration together with industrial automation expertise. It will support multiple physical network types, allowing systems to transparently use Ethernet, TSN, and different transport technologies.

A new Blend Software Development Kit (SDK) for this system would implement both DDS publish-subscribe and OPC UA client-server (see Figure 1). OPC UA client-server applications would use OPC UA APIs and the OPC UA TCP binary protocol. Publish-subscribe applications would use the DDS API on top of the RTPS protocol. Both would implement a common type system and security model to ease interoperability.

This architecture could be deployed in two different ways:

- Using separate SDKs for OPC UA and DDS in combination with Gateway nodes
- Using the Blend (OPC UA + DDS) SDK

Applications that need both client-server and publish-subscribe would either use:

1. The Blend SDK that allows individual nodes to do both OPC UA client-server and DDS publish-subscribe, or
2. Existing separate OPC UA and DDS SDKs in combination with OPC UA/DDS Gateway nodes. Gateway nodes pass data between client-server and publish-subscribe applications. They also proxy important system definition information on both sides, including discovery and security.

A typical distributed system would contain a mix of these nodes. The ability to support nodes that act as pure OPC UA clients or servers will remain important. These nodes can be implemented with the existing OPC UA SDKs and will have a smaller footprint. Likewise, nodes that primarily use publish-subscribe for communication could benefit from using a pure DDS SDK. Going forward, as the need for more capable and autonomous systems grows, more and more systems are expected to be built using the Blend SDK. There will be more Converged nodes and fewer pure OPC UA or pure DDS nodes. Once this transition is complete, the need for the Gateway will go away.

Full discovery across OPC UA and DDS works, even for systems that use a pure OPC UA or pure DDS SDK. DDS topics will be available in the address space of the OPC UA servers embedded in gateway nodes. Likewise, DDS participants will show their represented OPC UA services and clients in the gateway nodes. The converged nodes will be visible by both pure OPC UA and pure DDS nodes.

Initially, security will require separate configuration of the client-server and publish-subscribe access rules. However, both client-server and publish-subscribe can tie to the same certificate authorities and share the same credentials (e.g. signed identity certificates). A combined security model can be developed as part of the Blend SDK.

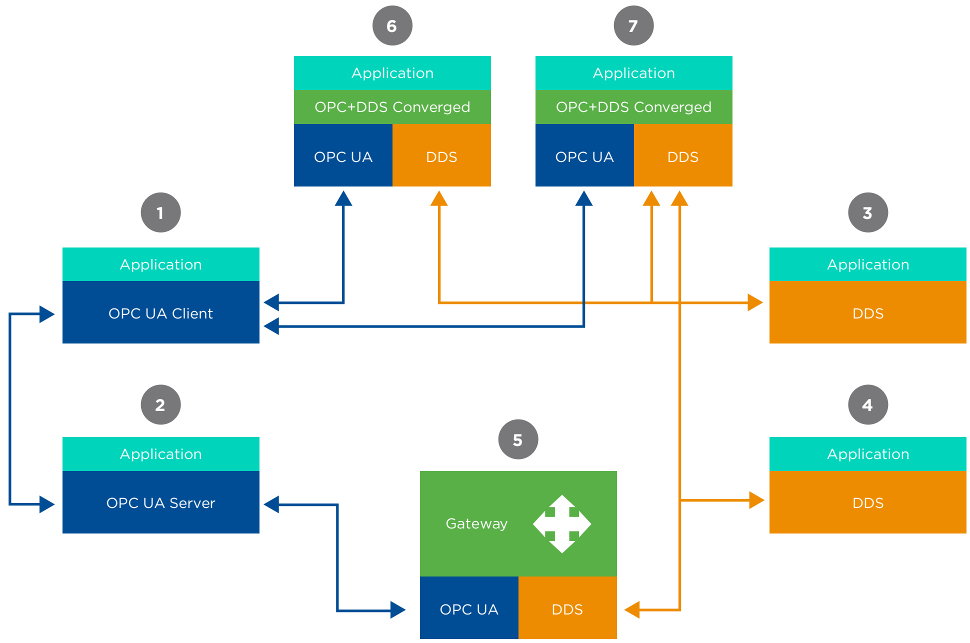

Figure 2 The proposed Blend SDK will support 4 types of nodes: UA clients (1); UA servers (2); DDS nodes (3, 4); Gateway nodes (5); and Converged nodes (6, 7). Converged nodes use the Blend SDK that allows creating application nodes that use both patterns (OPC UA and DDS). Gateways allow systems that use a pure OPC UA SDK to interoperate with those that use a pure DDS SDK. Gateways are only needed to integrate systems that were not designed to use both client-server and publish-subscribe from the beginning.

IMPLEMENTATION PLAN

Combining the proven OPC UA object-oriented information model with the DDS data-centric publish-subscribe model provides seamless, secure, reliable access to all system information and supports the needed information exchange patterns. It allows industrial organizations to leverage the domain experience of DDS in transportation, medical, energy, and military applications.

This proposal presents a way to quickly deliver the necessary standards and technology to deploy products and solutions. The OPC UA/DDS Gateway Specification provides a good starting point for this approach. This standard already provides mappings for the type systems and the functional behavior of the OPC UA Gateway. The next step would be to start developing the Blend SDK.

The bottom line is that the industry cannot afford to wait a decade or more for the necessary technologies to evolve and mature. This proposal of a new, integrated architecture addresses the complex requirements of industrial automation with timeless durability and complete flexibility. The well-developed technologies of OPC UA and DDS can be combined, with the effort going into integration rather than ground-up redesign. It will deliver rapid deployment options to suppliers and end users, whether building hardware or software products.

End users expect easy integration of data models into their complete systems. Integrating OPC UA and DDS provides standard information models across the industrial landscape. It also offers complete extensibility to add a custom data model. Long term, industrial automation end users will achieve vendor independence, an installed base of products and services, and multiple platform options. IIoT end users in other industries will likewise benefit from seamless device integration and information models.

NEXT STEPS

This document is a proposal for industry review and comments. Our goal is to build a coalition of key vendors, industry consortia and standards bodies to support this approach. We seek your support for this effort.

To comment on this paper, please contact Gerardo Pardo (gerardo@rti.com); Fernando Garcia (fgarcia@rti.com); or Costantino Pipero (costantino.pipero@beeond.net).

ABOUT BEEOND

Beeond, Inc. helps both automation vendors and users transition their products and systems to the Industrial Internet of Things (IIoT) using the OPC Unified Architecture (UA) Standard. Our OPC UA consulting, training and software development services are designed to guide and support our customers through the complete technology adoption lifecycle. We follow a Five Step IIoT Adoption Process that is well organized and practical, so our customers realize value quickly and cost-effectively. Unlike traditional software development companies, we focus only on OPC UA and its implementation in edge computing, factory floor and cloud-based systems. Our experience and expertise reduces development cost and lowers project risk for our customers. For more information, visit our website at www.beeond.net.

ABOUT RTI

Real-Time Innovations (RTI) is the Industrial Internet of Things (IIoT) connectivity company. The RTI Connext® Databus is a software framework that shares information in real time, making applications work together as one, integrated system. It connects across field, fog and cloud. Its reliability, security, performance and scalability are proven in the most demanding industrial systems. Deployed systems include medical devices and imaging; wind, hydro and solar power; autonomous planes, trains and cars; traffic control; Oil and Gas; robotics, ships, and defense.

RTI lives at the intersection of functional artificial intelligence and pervasive networking℠.

RTI is the largest vendor of products based on the Object Management Group (OMG) Data Distribution Service™ (DDS) standard. RTI is privately held and headquartered in Sunnyvale, Calif.

Download a free 30-day trial of the latest, fully-functional Connext DDS software today: https://www.rti.com/downloads.

Beeond
1822-6 S Glenburnie Rd, Suite 213
New Bern, NC 28562

beeond.net
company/beeond
beeond

rti
232 E. Java Drive,
Sunnyvale, CA 94089

rti.com
rti_software
rtisoftware
company/rti
connextpodcast
rti_software