TOP 10 CONSIDERATIONS FOR TECHNOLOGY AND BUSINESS TEAMS WHEN IMPLEMENTING OPC UA TECHNOLOGY
Choose the Right Software Development Kit (SDK) for IIoT Connectivity

White Paper
Abstract/Introduction

Growing adoption of the Industrial Internet of Things (IIoT) and Industrie 4.0 is driving requirements for open and secure connectivity between devices (e.g., machine-to-machine) and edge-to-cloud solutions.

Those who deploy the OPC Unified Architecture (UA) will be able to better leverage plant floor to enterprise communications as a vehicle to participate in IIoT applications. The goal for OPC is to be a standard for interoperability for moving information vertically through the enterprise of multi-vendor systems as well as providing interoperability between devices on different industrial networks from different vendors.

Since OPC UA serves as a key data connectivity standard for industrial automation, vendors are seeking to enable their new and existing products with this technology to compete on the IIoT/Industrie 4.0 landscape.

In order to quickly and efficiently implement OPC UA in their products, controls suppliers need an effective software development kit (SDK) to minimize development time and effort, and deliver secure and reliable products.
Table of Contents

Today's Industrial Revolution ................................................................. 4
Need for Improved Connectivity ............................................................... 4
Growth of OPC UA Technology ............................................................... 5
Importance of Developer Tools ............................................................... 5
Choosing the Right SDK ................................................................. 6
  Total Cost of Ownership ................................................................. 6
  Platform Scalability ................................................................. 6
  Ease of Use ................................................................. 7
CPU Utilization ................................................................. 8
Memory Footprint ................................................................. 8
Toolchain Compatibility & Security ............................................................. 8
Language Support ................................................................. 8
Third-party Libraries ................................................................. 9
Future Roadmap ................................................................. 9
Vendor Assistance ................................................................. 9
Partner with Matrikon ................................................................. 10
Today’s Industrial Revolution

The Industrial Internet of Things (IIoT) enables companies to collect, aggregate, and analyze data from sensors to maximize the efficiency of machines and the throughput of an entire operation.

Fig. 1. In a highly competitive global marketplace, industrial organizations are dealing with the evolution of their businesses and operations, where the virtual world of information systems, the physical world of machines and the Internet have become one.

Technological advances have been the impetus for dramatic increases in industrial productivity since the dawn of the Industrial Revolution. The first phase was the mechanization of production using water and steam power. The second phase then introduced mass production with the help of assembly lines and electric power, followed by the third advancement with the use of electronics and information systems to further automate production.

Today, the fourth industrial revolution encompasses the technologies and concepts of the value chain organization. Known as Industrie 4.0 and comprising a set of technology principles set down by the German government, the current phase of innovation relates to the previous three industrial revolutions, each of which heralded a turning point in production and manufacturing strategies. Industrie 4.0 employs the concept of cyber-physical systems designed to link real objects with information-processing, and virtual objects and processes, via information networks – including the Internet.

Need for Improved Connectivity

Now, more than ever, manufacturing firms need to make sense of vast quantities of data having a critical impact on their performance. To support the variety of applications necessary within an industrial facility, information must be delivered with context so it can be understood and used in various ways by a variety of people.
Dubbed the Industrial Internet of Things (IIoT), and in tandem with Industrie 4.0 practices, the latest wave of technological change is bringing unprecedented opportunities to manufacturing businesses. It combines the global reach of the Internet with a new ability to directly control the physical world, including the machines, facilities and infrastructure that define the modern landscape. Adoption of the IIoT is being enabled by the improved availability and affordability of sensors, processors and other technologies that have helped facilitate access to, and capture of real-time information.

In IIoT environments, OPC UA is regarded as a critical standard for ensuring interoperability between a broad set of manufacturing processes and equipment, spanning decades of investment for many companies.

First issued in 1996, the OPC Foundation’s data connectivity standard is increasingly being used by automation suppliers and original equipment manufacturers (OEMs) to enable IIoT connectivity in their products. The OPC standard allows for secure and reliable exchange of data across manufacturing and other enterprises. The OPC Unified Architecture (OPC UA) is a platform-independent, service-oriented architecture that integrates all the functionality of the original OPC specifications into a single, flexible framework. OPC UA extends the capabilities of the Classic OPC model by improving upon security and employing standard Internet technologies.

OPC UA is recognized as an enabling technology for the IIoT and Industrie 4.0, supporting multi-vendor, multi-platform interoperability for moving data and information from the embedded world to the enterprise. The standard is built on an information model that provides structure and context to information at its source, which is critical to have responsive systems. By adopting OPC UA, automation vendors get the best in open data connectivity today and in the future.

**Importance of Developer Tools**

The choice of tools to be used in the software development process can literally make or break a project. Once the target environment and programming language(s) is chosen, and the requirements and end goals are understood, the next task is to select the tools that will be used throughout the process. It’s...
When considering the high performance standards set for products used in the industrial realm, IIoT projects present a challenge for even the most seasoned development teams.

Important to be aware of the types of developer resources that are available for use, the benefits each can provide as well as the implications for using them.

A well-functioning software development kit (SDK) delivers a quality user experience for both the end-user and the engineers who integrate it. Selecting an SDK represents a substantial commitment in terms of initial adoption, implementation, and ultimately maintenance. These activities are costly, and if the most appropriate toolkit is not chosen, the result likely means significant additional work, rework, or worse – serious system limitations.

In the automation market, suppliers require an effective SDK solution for deploying IIoT connectivity across their product lines. This includes the robust, user-friendly tools necessary to embed OPC UA functionality in their device or microchip to quickly move products to market. Vendor development teams seek a fully scalable SDK that allows them to easily interconnect industrial software systems, regardless of the platform, operating system or size.

Choosing the Right SDK

Whether you are a tool builder or an application developer – if your software needs to access automation data, you will want to implement OPC UA connectivity to ensure your system can access data via the world’s most popular, standards-based, open connectivity solution. The question is: how to best do this?

Discrete and process industry manufacturers, commercial customers, and automation OEMs choosing an OPC UA SDK should be mindful of the following key considerations:

1. Total Cost of Ownership

Most customers choose to buy an OPC UA SDK rather than develop it in-house due to a lack of internal expertise, high development and maintenance costs, the challenges of keeping up with constantly evolving standards and specifications, and the need to reduce OPC UA implementation time in their application and accelerate time-to-market.

Not all commercially available SDKs are the same. It is important to ensure the SDK vendor has implemented the common functionality necessary to enable OPC UA in most user applications, offers base functionality and functions, implements security handling, provides application programming interface (API) wrappers for high abstraction languages, and makes available use case examples.

Ease of use is incredibly important when choosing an SDK. Some technology providers choose to sell their commercially available SDKs for very low cost, but rely heavily on consulting revenue when they engage with customers. Customers are forced to pay for these additional services since the SDK does not include all common OPC UA functionality and simple-to-use APIs to integrate with existing applications. When calculating the total cost of ownership, customers should take both the consulting services cost and SDK upfront procurement cost into account.

2. Platform Scalability

An SDK’s scalability should go beyond being hardware platform agnostic and operating system (OS) independent. Most vendors have developed multiple stock-keeping units (SKUs) of products to address specific applications. This makes it impossible to enable OPC UA in all new or existing products, from discrete sensors and actuators, PLCs, remote terminal units (RTUs) and distributed control systems (DCSs), all the way to high-performance servers in a data center, using one scalable SDK. Developers have to learn
multiple SDK codes to work across different platforms. At the same time, SDK vendors will face an insurmountable challenge to keep up with maintenance and enhancements to all SKUs. Managing the product lifecycle becomes a problem and customers do not always get the timely support they need during development.

One way to address this problem is to choose a truly scalable SDK. OPC UA toolkits should work equally well in small, embedded environments and large, enterprise-based applications. This scalability makes the SDK a “one-stop-shop” solution for companies that want to OPC UA-enable multiple product lines ranging from embedded to enterprise applications. There are significant benefits to a single, fully scalable toolkit that allows users to quickly and easily interconnect industrial software systems, regardless of the platform, operating system or size. Ideally, the SDK should employ a robust and reliable design built from embedded first principles to maximize product uptime.

3. Ease of Use

By partnering with a knowledgeable SDK provider, you don’t have to be an OPC UA expert to take advantage of the standard’s powerful functionality. The SDK should incorporate abstraction methods employing simple objects, thus eliminating the need for in-depth knowledge of the OPC UA specification. The SDK should logically organize tasks in an intuitive manner for software developers, and utilize a common and consistent approach to simplify deployment from application to application. The SDK should also provide easy integration using APIs. In this way, users can enjoy easy customization with access to low-level OPC UA functions. Instead of having to master the nuances of multiple development products, they can learn one code base and then apply it across all of their systems and devices.

SDKs implementing a drop-in “OPC UA server/client-in-a-box” design provide a way to launch OPC UA-enabled products faster and
Considering that the world is moving in a digital direction, and that industries are increasingly connecting hardware and software to realize new value propositions, all enterprises must become hyper-agile if they are to compete.

with the fewest changes. Prototype development is reduced to days – not weeks or months.

4. CPU Utilization

Irrespective of implementing OPC UA technology in a Greenfield or Brownfield project, the cost of bill of materials (BOM) is of major concern. Many designers would like to reuse the BOM without moving to higher cost hardware (e.g., using ARM Cortex-M4 instead of ARM Cortex-M7). They may also want to target low-cost microcontrollers and use less central processing unit (CPU) resources on their embedded processor. This is why an OPC UA SDK should be written and optimized from first principles for embedded systems so the application can still perform a significant amount of work in a single thread in the case where multi-thread is not available. On an enterprise-class system, these same principles lead to significantly enhanced performance.

Customers should choose an SDK able to execute tasks in an OS or real-time operating system (RTOS), or in a bare metal environment on a microcontroller. This approach facilitates monitoring of a large number of process variables by multiple concurrently connected UA clients.

Fig. 4: Developers seek a software toolkit with superior performance characteristics allowing them to target low-cost microcontrollers or use less CPU resources on embedded processors.

5. Memory Footprint

A truly scalable OPC UA SDK should support all profiles – nano, micro, embedded and standard – an essential requirement for OPC UA implementation, especially in resource-constrained applications. The benefit is that developers can use the same API across all processor sizes and operating systems. The use of a small RAM and Flash footprint makes it possible to target low-cost microcontrollers or use less memory resources on an embedded processor. This memory model also does away with the need for system heap enabling superior robustness. The SDK can be optimized for minimum RAM and Flash utilization, or for large data sets and multiple concurrent client connections.

6. Toolchain Compatibility & Security

It is important to choose an OPC UA SDK that is designed to support a broad range of platforms and toolchains. Ideally, the SDK should run on 32-bit and 64-bit architectures, and employ distributed as obfuscated source code freeing vendors to use any hardware platform they want since they will compile the SDK for that platform. This is unlike “binary” SDK distributions, which only provide compiled versions of an SDK for specific platforms. An SDK without platform or OS dependencies can be compiled for and run on all CPUs and microprocessor units (MPUs) that meet the system requirements.

Developers purchasing an OPC UA SDK should also consider the toolkit’s support for commonly used communication transport protocols, encoding, and security modes. This includes support for opc.tcp transport and binary encoding, as well as sign and encrypt up to the latest standards. Additionally, they should ensure that their device is Ethernet TCP-enabled.

7. Language Support

Developing an OPC UA SDK in different languages while maintaining scalability and superior quality is a difficult task. Vendors who
have released multiple SKUs of their SDK supporting different languages are already finding it challenging to make incremental improvements to their products as new specifications like MQTT/AMQP Pub/Sub and UDP are being released. Experience has shown that C++ is the optimal language for use in developing an SDK. Conversely, wrapping native code in C, Java, .NET, or Python is a tried and tested technology and is not technically challenging. If a customer needs to incorporate the SDK in any language other than native C++, the SDK supplier can provide an appropriate wrapper service or other assistance.

8. Third-Party Libraries

Third-party libraries are another crucial consideration for developers undertaking OPC UA implementation. Most companies already have a preferred library version for their products and applications, so they usually like to stick to what they know. For this reason, leading technology providers typically offer wrappers for standard crypto libraries. They also offer use-case examples, manuals, and API reference for using other supplied wrappers such as OpenSSL, NanoSSL, mBedTLS, TinyXML2, and LibXML2.

9. Future Roadmap

It may seem obvious, but when choosing an OPC UA SDK vendor, it is important to know if they are financially stable and have the expertise necessary to support their customers’ needs over the long term. Since there are on-going developments around SDKs, and the OPC Foundation is always releasing new specifications like AMQP Pub/Sub, UDP, and in the near future TSN, it is critical to choose a partner who can keep up with the new features and enhancements. The vendor should maintain a market-backed technology roadmap and be committed to delivering SDK solutions addressing all of the new features.

10. Vendor Assistance

Aside from cost, performance and reliability issues, it is essential to work with an OPC UA vendor that provides high-quality customer support. A good vendor should have a dedicated support team available to help resolve any issues that may arise. This will ensure that the implementation process is smooth and that any problems can be addressed quickly, minimizing downtime and maximizing efficiency.
In addition to minimizing production costs, the Matrikon Flex OPC UA SDK requires the least amount of memory in the industry and runs as efficiently as possible to leave sufficient CPU resources for correct device functionality.

SDK vendor who is dedicated to building close relationships with customers to best address their business and technical needs. The right vendor will have proven industry experience and know-how, and provide a local presence on a global scale.

Partner with Matrikon

For over 20 years, Matrikon has been the world’s leading data connectivity supplier providing solutions for every major control system and application on the market. With successfull Classic OPC and OPC UA installations and industry-leading live-support around the world, Matrikon solutions are recognized for enabling universal access and seamless connectivity across the enterprise – independent of the devices, applications or manufacturer selection.

Fig. 6. For over 20 years, Matrikon has been the world’s leading data connectivity supplier to the automation industry.

Leading automation system and device suppliers choose the Matrikon Flex OPC UA SDK to easily and seamlessly embed OPC UA into their products. This versatile toolkit has a low memory requirement, but at the same time is highly performant. It enables developers to use the same API across all processor sizes and operating systems.

The Flex SDK is ideal for companies that aren’t experts on OPC UA, but want to offer it in their products, which is becoming increasingly popular with the growth of the IIoT. The toolkit quickly and easily enables any application, regardless of size, with OPC UA. It is intended for use by developers who need to implement native data connectivity that is:

- Based on a secure open standard
- Preserves rich data context
- Hardware platform-independent
- Operating system-agnostic
- Scalable for use in embedded and personal computer (PC) environments
- Flexible enough to facilitate communications between devices and between applications on the shop floor, office premises, and/or the enterprise cloud

Unlike other OPC UA SDKs that require developers to use a separate toolkit when implementing products on different platforms, the Flex SDK is the only solution developers need to use, maintain and update for all their products. This is an efficient and cost-effective answer to deploying IIoT connectivity across a product portfolio – and to ultimately take products to market sooner.
For More Information

Learn more about how Matrikon’s Flex OPC UA SDK can improve your performance, visit www.matrikonopc.com/sdk or contact your Account Manager, Distributor or System Integrator.

Matrikon

#1800, 10405 Jasper Avenue
Edmonton, Alberta, Canada T5J 3N4

www.matrikonopc.com