White Paper

Industrial Internet of Things

ExxonMobil and Intel Collaborate to Transform the Automation & Control of Onshore Oil & Gas Wells and Surface Facilities

The Universal Wellpad Controller (UWC), developed in collaboration with ExxonMobil, leverages Off-the-Shelf (OTS) hardware and open source middleware software to provide an innovative solution that brings cost-effective, interoperable automation and control to onshore O&G operations.

Authors

Pat Moore
Sensors, Automation and Robotics Advisor
ExxonMobil Upstream Research Company

Tao Cheng, Ph.D.
Technology Development Lead
ExxonMobil Upstream Research Company

Vani Desai
IoT Solutions Architect
Intel Corporation

Dean Samara-Rubio, Ph.D
Product Line Manager for Energy Sector
Intel Corporation

Louis Desroches
Global Lead - Process Industries
Intel Corporation

Don Nuyen
Account Executive
Intel Corporation

Table of Contents

> "Control systems are rapidly moving from the monolithic, closed and proprietary solutions of the past to an open and interoperable model. This is igniting a new era of innovation for the oil and gas industry. Modern virtual machines and containers built on a flexible and scalable microservices architecture will enable this transition."
>
> - Brian McCarson, Vice President, Industrial Solutions Division, Intel

I. Introduction

Like many capital-intensive industries where production assets, once deployed, may be in operation for decades, the selection of a technology solution in the oil and gas industry is not one that is taken lightly, as the organization is relying on this solution for the considerable commercial life of the asset. Therefore, for the sake of reliable service, the industry in the past has been selecting large system providers for their automation and control solutions. However, the vast majority of these systems are proprietary, which necessarily results in "vendor lock-in," tying the operator to a specific vendor for a considerable amount of time, leading to issues of limited innovation, compatibility, and technology obsolescence.

This obsolescence and the rapidly evolving needs of the energy industry are driving operators to carefully explore new approaches to mitigate possible vendor lock-in and provide themselves with the agility to adapt, innovate faster, and compete more effectively in a dynamic market.

More specifically, oil and gas companies are actively looking at transforming the automation and control solutions by first breaking the hardware and software inter-dependency inherent to closed proprietary control systems, then leveraging advances in IT computing to create an open-architecture, industry-supported interoperable platform adapted to the oilfield operational environment with special system requirements such as the following:

- certificate for appropriate area classification (e.g., ATEX/C1D2)
- extended operational temperature

- sealed and fanless designs
- extended lifetime
- cyber security
- ruggedized power supply
- connectivity to field instruments and to industrial data management systems.

In addition, oil and gas companies are transforming their IT/OT technology infrastructure to adopt new advances in hardware and software security, application deployment methods (containers, micro-service architecture, virtualization, open APIs/interfaces), analytics capabilities such as artificial intelligence (AI), machine learning (ML), deep learning (DL) and federated learning – just to name of few.

Moving to such new architectures is often a process, so Intel is offering oil and gas organizations the means to jump start their transformation with new classes of solutions that give them a foundation upon which to build a more agile future.

One of the first physical realizations of an open interoperable control platform is the Universal Wellpad Controller (UWC), which is a powerful edge control device for well and surface facilities as well as an example of how Intel and its partners are driving innovation in the oil and gas industry.

II. Introducing the UWC

Developed in collaboration with ExxonMobil, the Universal Wellpad Controller (UWC) enables oil and gas organizations to manage and scale their field automation more efficiently and cost-effectively. The UWC is used to monitor and control onshore production wells and surface production facilities. It integrates ruggedized OTS hardware with an open architecture and reference open source software (UWC middleware) and application components from multiple vendors into an interoperable, process control platform (Figure 1). The extensibility of the platform and the edge analytics features of the middleware give operators and integrators the ability to insert near-real-time analytics at the wellpad, allowing process optimization and enhanced operational efficiency.

The UWC is an open and interoperable solution that not only replaces proprietary, single-vendor, monolithic Remote Terminal Units (RTUs), but serves as a flexible powerful controller that can support oil and gas operators monitoring and control needs through all phases of the wellhead production cycle, from free-flowing high production to secondary or tertiary recovery production.

Unlike conventional proprietary controllers, the controls of the UWC are deployed as containerized software installed onto OTS hardware. This allows multiple control functions to be instantiated and consolidated on a single UWC unit. It also provides a platform that can be updated and have new functions/applications added over time.

By employing an open platform like UWC at its production sites, oil and gas operators now have an automation platform that is easily updated and completely interoperable with other new technologies and protocols, including monitoring and analytics applications and new types of sensors and sensor networks. By replacing multiple single-function devices with a single UWC and integrating the latest software, companies can streamline operations and perform analytics at the edge to reduce network overloads and accelerate insights.

The UWC also enables oil and gas companies to deploy the right software for the specific needs of their individual wellpad deployments, either employing their own custom applications, third-party software solutions, and/or open source software applications. Each UWC can host the precise mix of software applications that is needed depending on the requirements of the specific deployment. What's more, a single UWC can be used to control multiple wells at a production location, further increasing productivity.

The full benefits of the Universal Wellpad Controller include the abilities to:

- **Mitigate possible vendor "lock-in" and protect against obsolescence** through the use of OTS hardware
- **Innovate more quickly** by leveraging an open source software model and containerized microservice architecture
- **Consolidate and simplify deployments** by replacing multiple single-function devices with a single Universal Wellpad Controller with preferred software

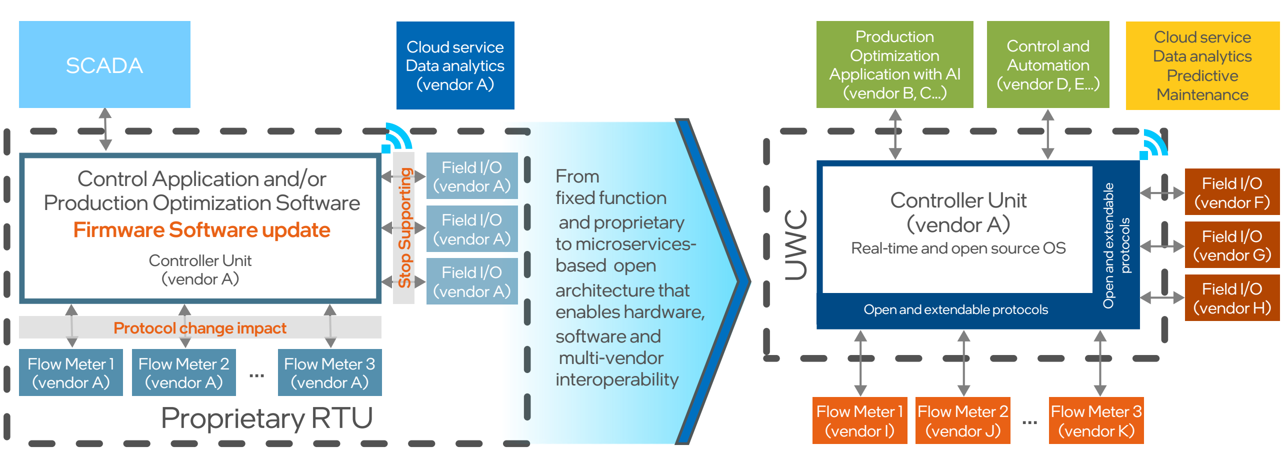

Figure 1. Industry transformation of traditional RTU-based proprietary technology to loosely coupled architecture with multi-vendor interoperability. The proprietary platform, from one "vendor A" shown on the left side, will be transformed into microservices-based control application over OTS hardware on the right side, delivering the protocol and multi-vendor interoperability, flexibility, scalability, manageability, etc. the O&G companies desire.

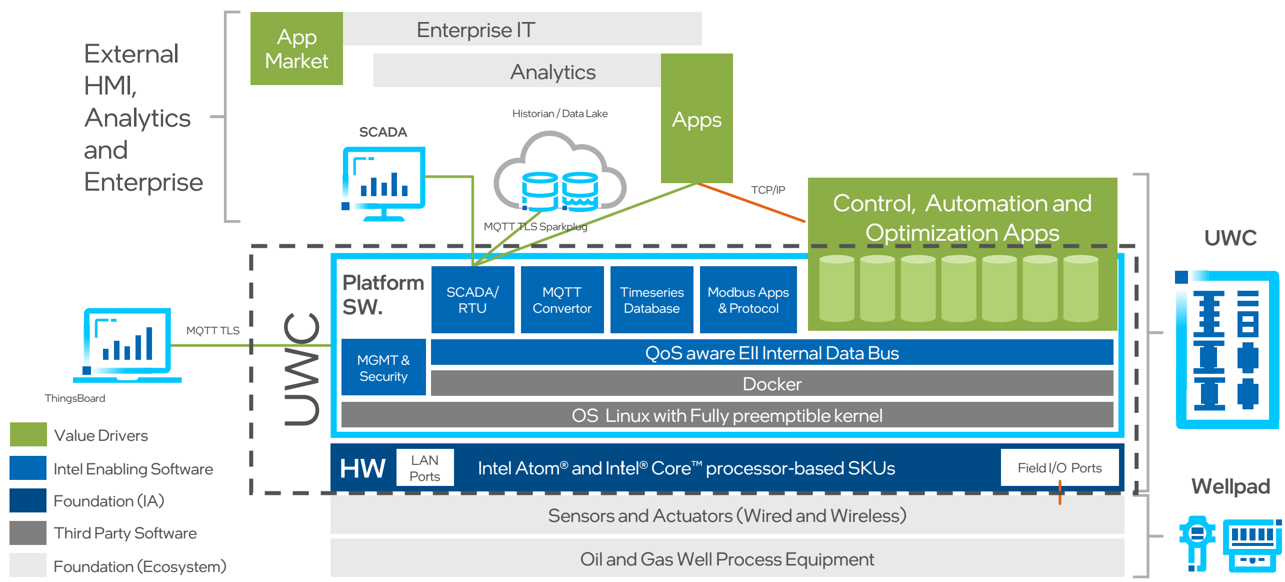

Figure 2. UWC solution diagram and its integration within a wellpad system

- **Reduce expenses** throughout the production lifecycle, including the costs of maintenance, support, equipment storage, and training, through active production and automation asset management, which mitigates field obsolescence challenges.
- **Increase flexibility** with scalable edge compute, from low-cost solutions powered by Intel Atom® processors to more powerful Intel® Core™ processor-based devices.

III. Oil and Gas Organizations Are Shifting to Open, Interoperable Solutions

In order to better understand the benefits of the UWC, one must understand the broader shift occurring not just in the oil and gas industry, but with enterprise technology architectures around the world. Today, the systems used to define most oil and gas deployments consist of monolithic software stacks, specialized hardware systems, and proprietary technology that are not under the control or management of oil production firms. The "inner workings" of these solutions are not visible to the end user, and oil and gas companies are at the whim of technology vendors in terms of the pace of innovation. Similarly, even though many standards exist for the oil and gas industry, most of these systems are incapable of fully interoperating with solutions from other vendors.

The UWC enables oil and gas operators to shift their technology architectures from proprietary, legacy systems to an interoperable open environment that consolidates existing workloads as software-defined applications that run on standard computing platforms. Applying open standards and open source software to the oil and gas industry will lay the foundation for this dramatic shift from workloads using proprietary interfaces and running on fixed-function hardware to interoperable and portable workloads running on open source software and OTS hardware. The UWC is the first of a new generation of solutions that are laying the groundwork for this transformation.

By applying these architectures, oil and gas organizations will be able to:

- Leverage additional economies of scale created by standards and open source software projects
- Access a much larger pool of software vendors from which to source potential solutions
- Implement new innovations more rapidly, as well as upgrade from one generation of technology to the next with minimized effort and risk
- Consolidate existing systems and reduce the total cost of ownership (TCO) with regard to capital expenditures (CapEx) and streamline and reduce operating expenditures (OpEx).

The UWC was designed specifically with this shift in mind. Figure 2 above provides a detailed diagram of the modular components, many of which are based on open source solutions. This architecture leverages container-based applications and a microservice architecture to not just fulfill its core purpose, but to provide a platform for additional "value driver" applications that provide myriad functions depending on the needs of the deployment.

IV. Modernizing Process Control Systems Using Containers and Microservice Architectures

The UWC's design is based on container technology and a modular, microservice architecture. This approach has been a driving force for innovation across almost every industry around the world. A recent report from industry analyst firm Forrester Research found that container usage is also rising with 86 percent of IT leaders prioritizing their use for more applications.[1] In fact, according to IDC, by 2023, over 500 million digital apps and services will be developed and deployed using cloud-native approaches like containers and microservices. For context, that is the same number of apps developed in total over the last 40 years.[2]

The majority of systems used by the oil and gas industry today are made up of a monolithic application architecture.

Monolithic applications are packaged onto hardware predetermined for peak loads. The main problem with this development approach is the tight coupling between components (hardware/software), such that it cannot be easily upgraded and updated. For example, a simple software update can have negative side effects that could result in revalidation of the entire hardware and software stack.

In contrast, a microservice architecture is an architectural style that structures an application as a collection of individual services that are easily maintained and tested, but also deployable on their own. The microservice architecture enables rapid, frequent, and reliable delivery of large, complex applications, which makes it very attractive to enterprises of all kinds, as this provides significant agility and rapid innovation. The microservice architecture emphasizes that the application should be composed of small services so that each microservice implements a single function and is deployed individually. In addition, each microservice has a well-defined interface so that other microservices can communicate and share data. Each microservice isolates potential problems within that single microservice, so if there is an issue in a microservice, it only impacts that individual service. The other microservices can continue to handle requests for control and automation.

Most microservice architectures make extensive use of containers. A container is a type of software that is highly standardized and operates as a single unit that packages up all the necessary code for an application so it can move from one computing environment to another and still run reliably. Containers are lightweight in the sense that they do not require much computing power. They are also "independent" in that they are modular in nature and can be combined to work easily with other containers. Similar to the concept of a "virtual machine," a software-based instantiation of a computer, a container is essentially a "virtual application" - an executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings. Containerized software will always run the same, regardless of the infrastructure. The UWC is able to provide the benefits it does due to its foundational microservice architecture. The UWC provides a set of essential middleware services for experts in the field to create microservice-based solutions. Features include data collection and control with Modbus, prioritized data exchange pathways for process control data, connectors for device management, SCADA, and back-end data systems. The scalable hardware and software framework enables the addition of new protocols, data analysis, and other capabilities to meet specific application needs. With a UWC, the control and optimization of multiple wells can be consolidated on one single edge compute node.

The process control and monitoring applications are latency-sensitive and require soft-real-time[3] capabilities. The UWC middleware architecture ensures that control applications are not disturbed by other, non-real-time microservices (application) on the system. The microservices architecture keeps all different subsystems in separate containers that ensure the required amount of isolation and control over the involved system resources. For example, the SCADA RTU microservice interfacing with SCADA Headend and Modbus microservice responsible for field device data ingest/control are isolated and assigned appropriate priority for a system resource usage. The SCADA RTU microservice reporting data to SCADA Headend are running at lower system priority compared to an artificial lift application performing control operations using Modbus microservice interface. The containers provide options to schedule threads with real-time priority when underlying Linux-based operating sytem supports preemptive scheduler (PREEMPT_RT patch[4]); UWC microservices architecture includes this feature of containers for prioritizing compute resources.

Another important aspect is how microservices communicate with each other; especially for process control, the QoS (Quality of Service) must be guaranteed to achieve soft-real-time capabilities. The UWC middleware includes EII (Edge Insights for Industrial) internal data bus (ZeroMQ based) that guarantees QoS and efficient data transfer with minimum latency in IPC mode. The UWC microservices communicates over EII internal bus using publish-subscribe messaging pattern and policy-based data traffic prioritization.

This modular software architecture enables end users to easily control prioritization of data traffic, which in turn enables soft real-time[3] control for multiple applications. The application developer can configure prioritized data traffic based on system operations such as read sensor value, write to actuators, or continuously poll sensor value. The vendor applications and microservices can request a prioritized data path using a well-defined publish-subscribe interface of the microservice.

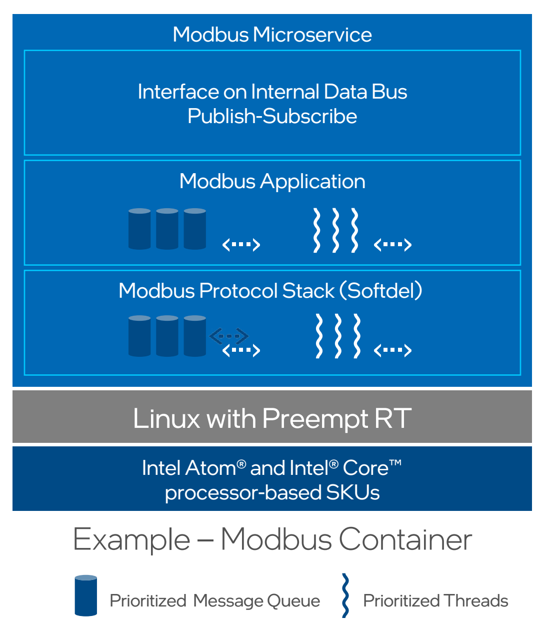

Figure 3. An example of the UWC's modular, container-based microservice architecture

Figure 3 shows an example of Modbus Microservice- how the UWC leverages containers and encapsulates Modbus protocol stack (from Softdel Systems[5]) and Modbus application. Modbus microservice interfaces with the internal EII bus and exposes a publish-subscribe (Pub-Sub)-based interface to other microservices on the edge compute node. Modbus microservice application and stack both are multi-threaded with prioritized threads. The prioritized threads are scheduled based on their priority relative to other runnable threads. The Modbus microservice includes prioritized message queues that mean the message with higher priority

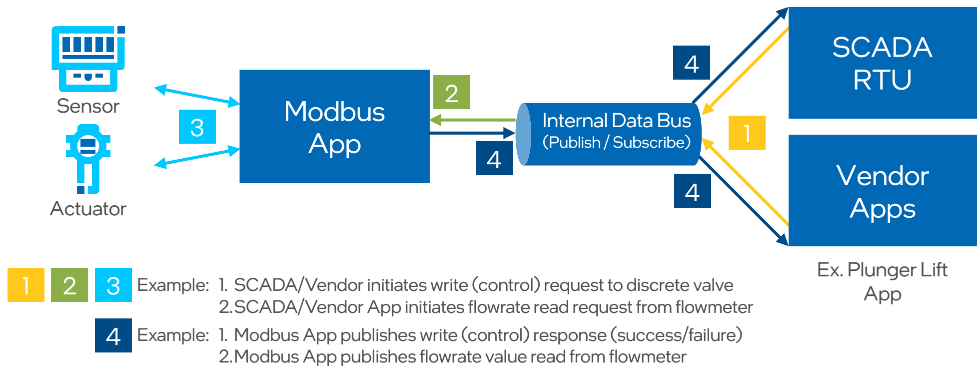

Figure 4. Request/Response-based operation on internal data bus between microservices

will be in front of the queue for processing versus other normal priority messages. The Modbus service is explained as an example; similarly, other microservices from UWC software middleware are decoupled, individually deployable, include publish-subscribe interface and prioritized data pathways.

Publish-Subscribe messaging pattern is an asynchronous messaging service that decouples services that produce events from services that process events. Within the Pub-Sub messaging framework, one microservice sends the messages and others subscribe to them. The sender microservices do not need to know who will receive the messages or how many receivers are there. The UWC middleware includes the publish-subscribe messaging pattern for communication that decouples the control application from underlying control protocols/transport and improves interoperability. Once the publish-subscribe message formats are defined, the control applications can be developed independently from each other as well as the underlying transport protocol. In the future, transport protocols can be updated without having to change the control applications. Figure 4 shows the example of Request/Response-based operation on EII internal data bus. The vendor app or SCADA microservice uses the Pub-Sub interface to read the value of acquired data from the sensor and actuators, such as reading the flowrate from the flowmeter of a wellpad. The production optimization app could actuate the sensor using the same Pub-Sub interface. In the future, the app could use different protocol/physical interfaces to control the valve without updating the Pub-Sub-based interface for the application.

Figure 5 shows an example of asynchronous communication with multiple microservices. The Modbus microservice does data acquisition from the sensors and/or actuator on a wellpad (such as the flowrate from flowmeter) and publishes results on the Internal EII data bus. Multiple microservices can subscribe to this acquired data. For example, a SCADA headend might need an update on line pressure, or a production optimization microservice might need to know the line pressure, flow, etc. in order to actuate the artificial lift discrete valve. Line pressure values may also need to be stored into historical time-series databases for further analytics. The Pub-Sub model enables the scalability of exchange of data and the dynamic addition of various apps via microservices.

The UWC middleware framework includes device management that can OTA (Over The Air) add or update microservices from the cloud or via an on-premises-based interface. The field technician can connect to the UWC edge node over MQTT (formerly MQ Telemetry Transport) TLS to securely update running microservices or even add a new application. The device management framework is designed to be modular and flexible, ensuring scalability of the solution across preferred Cloud Service Providers and on-prem-based deployment.

The UWC solution itself runs on Intel Atom processors, but due to its modular architecture, it can scale from Intel Atom processors to Intel Core processors with no changes to the UWC middleware framework. There is even OTS hardware available with C1D2 (Class 1 Division 2) certification with Intel Atom and the Intel Core product line.

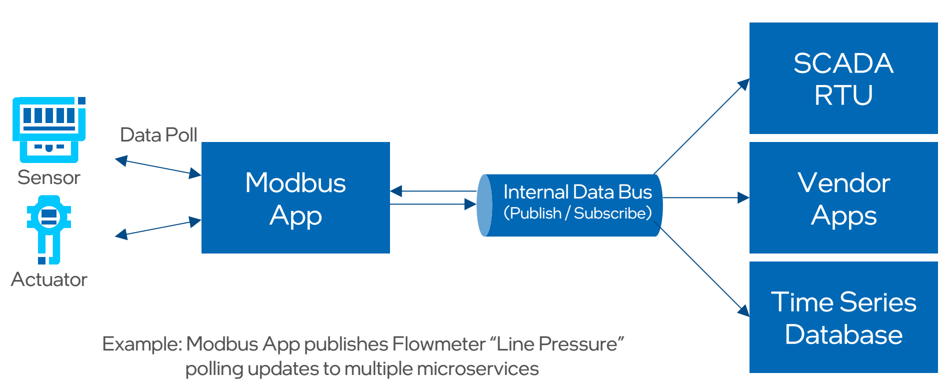

Figure 5. Asynchronous data communication on internal data bus between microservices

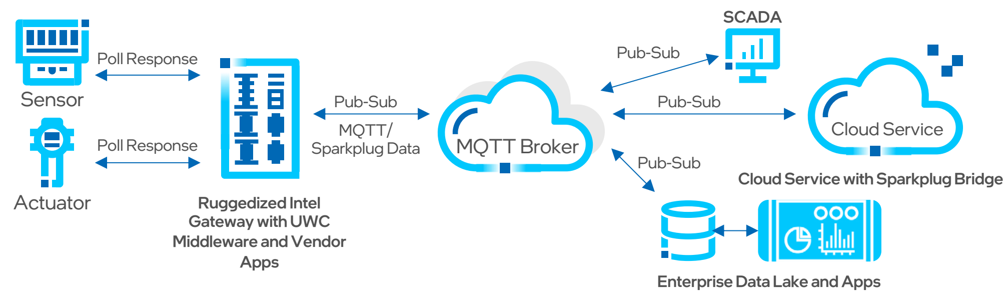

Figure 6. UWC infrastructure readiness

V. Bridging the IT/OT Gap with Sparkplug

Part of what makes the UWC platform so powerful as a platform is its ability to serve as a bridge between operational technology (OT) frameworks governing wellhead production and Information Technology (IT) architectures that generally house the management, monitoring, and analytics solutions governing business applications and metrics. Historically, OT and IT networks operated in separate spheres and had little to no interaction with one another. A big reason was there was no mechanism, nor common language, for the systems in OT and IT to "speak" with one another.

However, in recent years, the advent of trends like digital transformation has enabled the convergence of OT and IT systems. A big part of this has been the rise of containers and microservices, but also the development of messaging protocols that enabled these disparate systems to communicate. For industrial systems, a protocol known as MQTT has become the dominant messaging protocol for industrial systems across multiple markets. MQTT is a lightweight, publish-subscribe network protocol that allows for multiple data consumers. MQTT enables companies to ingest data and share it throughout both IT and OT systems, offering a critical mechanism for sharing information across the entire network.

Despite its powerful functionality, the industry recognized MQTT needed something more to provide value to the data it transmitted across OT and IT networks. As a result, a coalition of organizations created the open source Sparkplug specification. Sparkplug is a specification that defines how to use MQTT in a mission-critical, real-time OT environment. It is an open source software specification that is governed by the Eclipse Foundation, an organization tasked specifically with providing vendor-neutral governance over Sparkplug and other open source solutions. This is similar to the role of the Linux Foundation in relation to the governance of the Linux kernel.

Eclipse Sparkplug provides MQTT clients with a framework to integrate and understand the purpose of the industrial data being transmitted. The specification defines the namespace, defines state, and defines the nature of the payload. This process enables true interoperability between industrial applications, so anyone can make use of the framework for MQTT interoperability. Many device manufacturers are supporting Sparkplug, which means it is built in natively on the device on the OT floor.

The MQTT/Sparkplug SCADA RTU microservice is included to provide interoperability to the leading SCADA systems and cloud infrastructure and to achieve OT-IT interoperability. This is a key feature of the UWC, as this enables this solution to be deployed anywhere and essentially interoperate with almost any solution that the end user requires. The result is enormous flexibility and adaptability to almost any oil and gas environment. Figure 6 provides a detailed view into how UWC leverages Sparkplug to interoperate across a SCADA Network and cloud service providers.

During operations, the well conditions change rapidly and can indicate everything from minor problems to actual emergencies that require an immediate shut down. In an actual emergency situation, the operator monitoring the well needs to be immediately alerted. With the UWC's modern architecture, alerts can be sent to multiple locations, including the SCADA system, the operators' smartphone or other mobile device, as well as data logs in the cloud or on-premises. The alerts are published from the UWC edge node on scalable MQTT broker systems.

VI. Laying the Foundation for AI and Model-based Control Optimization Applications

With its container-based microservice architecture, open source middleware framework and integration of open source specifications, the UWC provides oil and gas end users with a powerful platform to quickly update their wellpad productivity. Just as importantly, its modular architecture and integration of open source specifications like Sparkplug make the UWC a flexible platform that enables oil and gas companies to build AI/ML and analytics-based solutions that have the potential to drive their businesses to new levels of production. These AI/ML solutions can deliver accurate data analysis very quickly based on well-established algorithms leveraging the enormous data set collected over time. This process can help field engineers to understand complex problems such as selecting the optimal type of artificial lift system and appropriate time of application.

The nature of its open source architecture enables the UWC to serve as a critical "translation" point, gathering critical data and metrics from multiple wellpads and then converting this data into a format to be consumed by powerful AI-based applications in more remote IT-based systems. Its flexible architecture enables oil and gas entities to easily upgrade applications on the UWC itself while also leveraging its analytics application of choice in the data center. These companies can also leverage rapid innovations created by the open source community around MQTT/Sparkplug, allowing them to take advantage of the expertise of the larger industrial community.

VII. Conclusion

As global energy markets around the world continue their journey of digital transformation, the oil and gas industry is recognizing the benefits of open and interoperable software-based systems. This new class of solutions built on OTS hardware with open source software based on containerized microservices architecture supporting soft-real-time capability and open specifications like Sparkplug is poised to enable a simpler, more productive, and more cost-effective industry where innovation is commonplace. Success in this new market is dependent on the individual oil and gas operator's embrace of this new model. Only a truly interoperable and open architecture approach will allow for the consolidation of existing oil and gas workloads as software-defined applications running on standard advanced edge compute platforms like the UWC.

Beyond laying the foundation for a new era of remote process control productivity, this open model will also enable the development of new technologies from AI and analytics to new sensor network models that will further advance the oil and gas industry. Flexibility, reduced costs, rapid innovation and increased productivity are all a result of this approach. Thanks to the joint development of Intel and ExxonMobil, the UWC is just the first example of the many innovations that will be driven by adopting a microservice architecture and an open source model.

Learn More

- **https://www.intel.com/content/www/us/en/internet-of-things/industrial-iot/overview.html**
- UWC
  https://www.intel.com/content/www/us/en/energy/digital-oilfield.html

  https://software.intel.com/content/www/us/en/develop/articles/universal-wellpad-controller.html
- Edge Insights for Industrial

  https://software.intel.com/content/www/us/en/develop/topics/iot/edge-solutions/industrial-recipes.html

  https://www.intel.com/content/www/us/en/internet-of-things/industrial-iot/edge-insights-industrial.html
- Sparkplug Working Group **https://sparkplug.eclipse.org/**

[1] Cloud Container Adoption in the Enterprise, Forrester Research (June 2020) - https://ecm.capitalone.com/WCM/tech/cloud-container-adoption-in-the-enterprise-report-capital-one.pdf

[2] IDC FutureScape Outlines the Impact "Digital Supremacy" Will Have on Enterprise Transformation and the IT Industry (October 2019)- https://www.idc.com/getdoc.jsp?containerId=prUS45613519

[3] The PREEMPT_RT patch enhances Linux OS for real-time

[4] Soft real-time system, even if the system fails to meet the deadline, possibly more than once the system is not considered to have failed, and missing the deadline is also acceptable as long as it occurs rarely and not repeatedly. The UWC applications control cycles executed in the range of 100ms to 1s.

[5] Modbus Protocol Stack provided by Softdel Systems https://www.softdel.com/

0421/VD/MIM/PDF 346591-001US