Towards Consolidating Industrial Use Cases on a Common Fog Computing Platform

Patrick Denzler*, Jan Ruh†, Marine Kadar‡, Cosmin Avasalcai§ and Wolfgang Kastner*
*Automation Systems Group, TU Wien, Vienna, Austria
Email: patrick.denzler@tuwien.ac.at
†TTTech Computertechnik AG, Vienna, Austria
Email: jan.ruh@tttech.com
‡SYSGO GmbH, Klein-Winternheim, Germany
Email: marine.kadar@sysgo.com
§Distributed Systems Group, TU Wien, Vienna, Austria
Email: c.avasalcai@dsg.tuwien.ac.at
*Automation Systems Group, TU Wien, Vienna, Austria
Email: wolfgang.kastner@tuwien.ac.at

***Abstract*—Converging Information Technology (IT) and Operations Technology (OT) in modern factories remains a challenging task. Several approaches such as Cloud, Fog or Edge computing aim to provide possible solutions for bridging OT that requires strict real-time processing with IT that targets computing functionality. In this context, this paper contributes to ongoing Fog computing research by presenting three industrial use cases with a specific focus on consolidation of functionality. Each use case exemplifies scenarios on how to use the computational resources closer to the edge of the network provided by a Fog Computing Platform (FCP). All use-cases utilize the same proposed FCP, which allows drawing a set of requirements on future FCPs, e.g. *hardware, virtualization, security, communication and resource management*. The central element of the FCP is the Fog Node (FN), built upon commercial off-the-shelf (COTS) multicore processors (MCPs) and virtualization support. Resource management tools, advanced security features and state of the art communication protocols complete the FCP. The paper concludes by outlining future research challenges by comparing the proposed FCP with the identified requirements.**

***Index Terms*—Fog Computing, Industry 4.0, Virtualization, Security, Resource Management, Use Cases**

I. Introduction

Modern factories are complex technical environments built upon software and hardware agents from the domains of information technology (IT) and operations technology (OT). IT and OT in industrial automation form a hierarchy that is still part of the well known automation pyramid [1].

On the lower levels of the automation pyramid, close to the factory floor, we find OT featuring programmable logic controllers (PLCs) and industrial communication systems, such as EtherCat or Profibus [2]. OT must fulfill strict real-time requirements to guarantee a timely processing of sensor values and a safe operation of actuators, valves, and electrical motors of machinery that form complex control loops. A level higher in the hierarchy, the control loops are being monitored by means of Supervisory Control and Data Acquisition (SCADA) systems and managed via human machine interfaces (HMIs) and other industrial applications [3]. In contrast to OT, the HMIs and industrial applications do not have to meet strict real-time requirements. Instead, they have to provide high data throughput paired with computational power, internal connectivity as well as connectivity with external entities, e.g., on the enterprise level of a factory. As a result, they utilize IT, that is commercial off-the-shelf (COTS) multicore processors (MCPs) and standard Ethernet, to connect to PLCs and machinery. The top of the automation pyramid contains plant management, business, and enterprise levels. There we can find traditional servers and desktop PCs that interconnect via standard IT communication systems and reassemble a traditional business IT environment.

The discrepancy between the requirements of IT and OT results in an isolation of the sensors and actuators of machinery on the factory floor from the computational resources and the connectivity available in the management and enterprise levels of a factory. The traditional field busses that are used in OT to connect machinery to PLCs cannot cope with the high rates with which machinery generates operational data. Furthermore, the PLCs often do not provide enough computational power to process and forward machine data to industrial PCs and higher levels of the automation pyramid.

These days, industrial automation is on the urge of a new industrial revolution. Industry 4.0 envisions an architecture that vertically and horizontally integrates a multitude of devices and spans seamlessly from factory floors to the cloud. Over the last years, a multitude of architectures and platforms that help overcome the strict separation of IT and OT in Industry 4.0 have been proposed. They utilize Fog or Edge Computing solutions [4] that place compute nodes between end devices and the cloud in order to bridge the gap between factory floor and cloud.

A common pattern for the application of Fog Computing in Industry 4.0 is the introduction of gateways that either translate between different communication protocols or preprocess acquired data and forward it to the cloud for further data analysis and data visualization. In [5], the authors deploy gateways

per manufacturing cell that instantiate virtual representations of the machinery and devices to enable seamless cooperation of manufacturing cells despite of differing communication protocols. Lucas-Estañ et al. [6] propose a hierarchical architecture for heterogeneous wireless connectivity of cyber-physical production systems (CPPS). A local manager handles the wireless communication of its cell whereas a central orchestrator coordinates the operation of all local managers and the data distribution in order to guarantee data accessibility for all subsystems. Prist et al. [7] make a case for an OSGi based gateway that connects PLCs via wireless technologies to cloud services. The use of OSGi allows for easy integration of new sensors and software modules. The authors also demonstrate wireless data acquisition, cloud based storage, data analysis and visualization by applying their gateway to a real world production line.

The sole focus on gateway functionality already mitigates vertical isolation and improves data accessibility for new industrial applications utilizing cloud services. However, in many industries, there is a trend towards consolidating functionality that before has been executed on dedicated hardware. For example, in the automotive industry, virtualization of in-car domain controllers yields overall development and production cost reduction by enabling easier integration of new functionality while maintaining support for legacy components and cutting down the amount of wiring due to the reduction of dedicated electronic control units. With the advent of Fog Computing, consolidation of functionality is also feasible in industrial automation.

In [8], the authors propose a novel hypervisor based PLC architecture that partitions the hardware platform into four virtual machines (VMs). The architecture allows the integration of classical PLC functionality whilst extending the capabilities offering a digital twin and cloud connectivity. Thereby, each VM takes over a different functionality, namely, local resource management, operation of PLCs, executing the control logic of machines that interfaces with SCADA infrastructure, bridging machine simulations and their software and hardware counterparts, and interfacing with external, cloud-hosted services.

We note a lack of generally applicable use case studies, whose insights can be easily applied to new scenarios in industrial automation. As a result, we present three real-world industrial use cases and derive requirements on a Fog Computing Platform (FCP) that implement these use cases. The FCP is an open platform composed of diverse actors, ranging from cloud services, over Fog Nodes (FN), to factory equipment, such as PLCs. A FN is a virtualized device that bridges the gap between the factory floor and cloud services by taking over functionality from the domains of IT and OT. Thereby it follows the paradigm of consolidation of functionality, effectively converging IT and OT onto a single device. The proposed FCP can be implemented and easily adapted to new use cases since it utilizes COTS hardware and state-of-the-art technologies in the area of real-time virtualization, security, and resource management. However, we identify open problems and research challenges that if tackled could maximize the benefits of Fog Computing for industrial automation resulting in cost reduction, shorter time to market, and even new business opportunities [9].

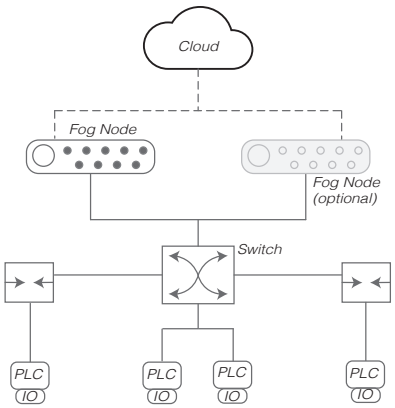

Fig. 1: FCP and network design applied to all use cases

Section II presents three industrial use-cases to deploy the FCP. Section III infers necessary requirements, for the FCP design developed in Section IV. Finally, we discuss relevant research challenges in Section V and Section VI concludes the paper.

II. Industrial Use Cases

In this section, we present three real-world industrial use cases (UCs), i.e., *UC 1: Converging System Infrastructure*, *UC 2: Accessing and Using Real-Time Machine Data*, and *UC 3: Deploying Machine Software Updates*. The UCs aim at consolidating functionality formerly provided by several components onto a FCP. The FCP network topology containing two FNs (one optional), Ethernet switches connecting either PLCs or machine IOs, and cloud services is illustrated in Figure 1, whereas Figure 8 shows the architecture of the FNs themselves. Section IV provides further details about the proposed generic FCP developed in FORA [10]. For accommodating the outline of requirements and open research challenges in later sections, each use case refers to the same FCP with the same FN architecture and network topology.

A. UC 1: Converging System Infrastructure

The legacy industrial network, as shown in Figure 2, is built upon a supervisory control and data acquisition (SCADA) system commonly used in OT. In SCADA systems, several industrial PCs and PLCs control the production processes and allow the configuration and monitoring of the connected machinery [3]. On the IT level, MQTT and OPC Classic [2] middleware transfer data to relevant clients and servers. A dedicated server stores the collected data in a time-series database for further analysis and client access. A gateway allows remote access to the network for maintenance activities and data transfer. The shop floor network uses EtherCat to ensure real-time communication and timely execution of control loops.

In *UC 1*, the main objective is to consolidate current functionality on a FCP and add further features such as firewalls,

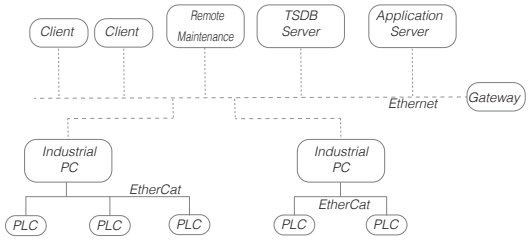

Fig. 2: Overview UC 1 original SCADA network.

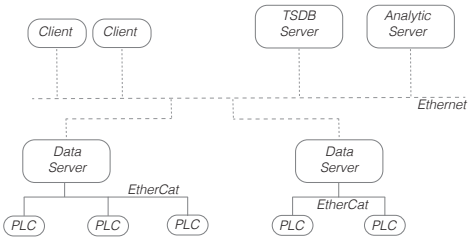

Fig. 4: Overview UC 2 original network with limited analytic capabilities.

novel industrial applications, and resource management. On a network and component level, FNs and cloud services replace the industrial PCs and dedicated servers. OPC UA [2]takes over the functionality of MQTT and OPC Classic, whereas standard Ethernet supporting the time-sensitive networking (TSN) [2] standards replaces EtherCat on the factory floor network, as illustrated in Figure 1.

Figure 3 presents an overview of the FN configuration and cloud components required to accommodate the intended consolidation on a FN as depicted in Figure 8. On the cloud level, graphical user interfaces (GUIs) allow remote maintenance activities of network components. The Data Distribution Service (DDS) middleware [2], provides the possibility to transfer and receive data to and from remote locations, whereby the application marketplace provides applications for simple deployment and extension.

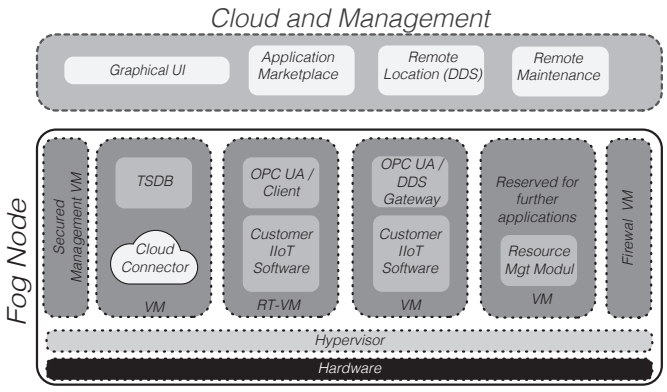

Fig. 3: Fog Node configuration and cloud components for UC 1

The primary FN accommodates most of the previous functionalities encapsulated in dedicated VMs. In more detail, the FN hosts one real-time VM running an OPC-UA client and a specific IIoT application to ensure real-time communication with the field devices (PLC). The received data is stored in a local time-series database before being forwarded to the cloud for further processing. In order to enable efficient use of the FN resources, each FN comes with a resource management module that automatically handles application deployment requests given by the user in the cloud. The resource management module is described in [11]. The desired application is installed directly from the cloud marketplace, depending on the available resources on the FN. As there is a redundant FN in this use-case, the resource management module can dynamically migrate applications to the second FN if the first FN is fully utilized or fails.

A special component in *UC 1* is the OPC-UA/DDS gateway. It connects the local OPC UA middleware with DDS. The OPC-UA/DDS gateway is installed on a dedicated VM and channeled through a Firewall VM, ensuring secure data transfer between the local network and remote locations. The secured management VM on the FN protects the system from unauthorized access and allows maintenance from the could.

In summary, the consolidation of the SCADA functionality into a Fog environment shows a reduction of hardware (cost) and system integration simplification. Furthermore, the new setup allows for improved data access while maintaining a high degree of security.

B. UC 2: Accessing and Using Real-Time Machine Data

In the second use case, sensors and actuators installed in machines on a shop floor generate thousands of data points per second while, at the same time, each work piece accumulates several hundreds of megabytes of raw data. In the original factory floor setup, as depicted in Figure 4, PLCs connect directly via EtherCat to machinery. Neither the PLCs nor the network can handle the vast amounts of data generated by the sensors and actuators. The industrial PCs (Data Servers), that could provide the computational power to preprocess and forward this high amount of data, lack the required connectivity and flexibility to host such applications. As a result, the data analysis for predictive maintenance or process optimization being performed in the two dedicated servers boils down to the bare minimum.

As in the previous use-case, the main goal is to consolidate functionality and implement additional features on the FCP as shown in Figure 1. The focus, however, lies on data analysis for predictive maintenance in order to minimize machine downtime, for quality control, or general optimization of the production process. Additionally, the FCP should provide local data analysis and the possibility to spawn specific analysis tasks during runtime. To demonstrate the ability of the FCP to adjust to legacy networks, the original EtherCat protocol remains in place.

The mapping of this use case to the FCP is depicted in Figure 5. The FN's RT-VM contains CODESYS, a controller development system, that allows receiving machine data directly from the PLC's IOs over EtherCat in real-time. A UDP publisher writes the data to the local time-

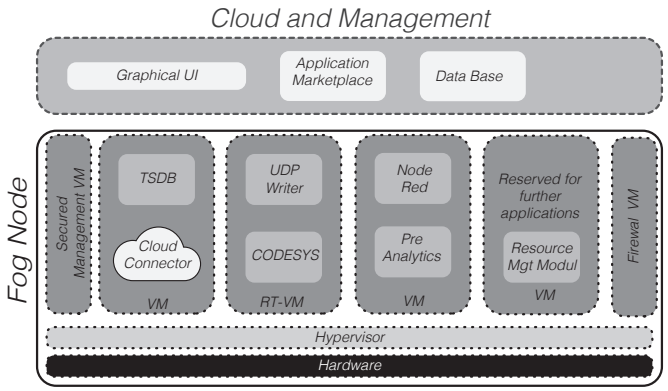

Fig. 5: Fog Node configuration and cloud components for UC 2

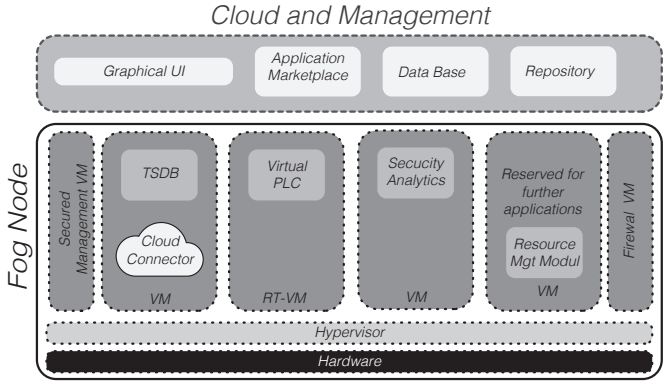

Fig. 7: Fog Node configuration and cloud components for UC 3

series database where analytic tools can access it. Due to the available computational resources, the FN allows an efficient preprocessing of the data before it is transferred to the cloud for big data analysis. Additionally, clients can execute simple data analysis tasks by using Node Red running in a dedicated VM and deploy their applications by using the same deployment mechanism introduced in UC 1. The overhead induced of running applications in a VM in the fog layer should be comparable to the overheads induced in the cloud [12]

In summary, the consolidation of legacy systems onto the FCP enables on-site data analysis and transfer of vast amounts of machine data to the cloud. The possibility to deploy analytic tools on the FNs enhances the capacity to investigate various viewpoints on data without disruptive system changes.

C. UC 3: Deploying Machine Software Updates

UC 3 aims at deploying updates to the shop floor requiring small or no downtime at all. Traditionally, when updating field devices with new programming, an operator is required to update each PLC manually what causes the production process to stop. In some cases, faults in the new code could even cause further disruptions. Additionally, older systems, as shown in Figure 6, with industrial PCs connected to the PLCs via a fieldbus, are not considered secure anymore [13].

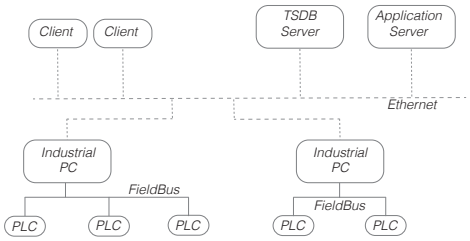

Fig. 6: Overview UC 3 original network with a Fieldbus

It is unique to UC 3, that the FN allows running a virtualized software PLC in the RT-VM that connects via a fieldbus directly to the IOs located in the machines. The setup as shown in Figure 7 allows changing and testing new code in a cloud repository and subsequently deploying it to the specific virtual PLC on the FN. With deploying new code directly from the cloud, the deployment requires only minimal stop times of the production process and minimizes possible faults in the code due to simulation testing of configurations and code in the cloud [8]. A security monitor application installed in an isolated VM protects the network and the FNs, by detecting unauthorized activities.

In summary, using a FN as a platform to host virtualized PLCs for control tasks and deploy code updates automatically from the cloud shows a reduction of stop times and an improved maintainability of the deployed code. The integrated security features of the FN's secure VM and isolated monitoring protect the FN and the network from unauthorized access.

III. Requirements

To enable consolidation of functionality as presented in the aforementioned use cases, the FCP must satisfy a set of requirements. We identify and discuss the requirements associated with each use case, namely, requirements regarding the FN hardware specification, virtualization technology, FN security, FN communication, and resource management.

A. Fog Node Hardware Specification

The FN aims to bring together OT and IT in a single device. Therefore, it must provide horizontal connectivity to PLCs and machinery on the factory floor as well as other FNs. Furthermore, FNs act as the link between the shop floor and the enterprise and business levels of a factory thus they must provide means for effective vertical communication in order to facilitate the utilization of cloud services in Industry 4.0.

Besides horizontal and vertical connectivity, converging OT and IT poses strong requirements on computational power and the timeliness of computations. This mixed-criticality requires the FN to come with a multicore processor that allows parallel execution of functionality from the IT and OT domain. At the same time, the FN must guarantee strict isolation of mixed-critical functions. Therefore, we require the hardware platform to support current virtualization extensions that allow for a partitioning of hardware resources such as processor, main memory, and networking.

B. Virtualization Technology

We require the FN to takeover functionality from the OT as well as the IT domain. As a result of this mixed-criticality, the FN must provide strict isolation between different functionalities. To this end, a hypervisor partitions the hardware platform into isolated VMs. For example, SYSGO's commercial PikeOS [14] real-time hypervisor provides a mixed-criticality runtime environment leveraging strong partitioning properties and timing guarantees to build certified modular systems. Notably, it can run a safety-critical ARINC653 guest OS together with a virtualized Linux OS on the same platform into isolated VMs. However, at the moment, PikeOS does not support middleware features, such as memory bandwidth and cache partitioning, as necessary in the use-cases. Our hypervisor implementation must fully utilize the hardware platforms capabilities to implement the required horizontal and vertical connectivity of the FN.

C. Fog Node Security

The Fog Node must be a trustable platform able to resist threats. Potential adversaries could raise attacks to steal confidential information or modify programs and data to compromise business information, security, or safety of people and equipment. An attack can be initiated from different sources:

- **External Agent:** Since the device is an open system with networking capability, it is a direct entry point to the factory for an adversary. A malicious actor could attempt hijacking the device interface: e.g. distributed deny of service could compromise Fog Node availability.
- **Third-Party Service Provider:** A hidden threat could lurk in a third-party software, for spying or compromising other user-level software at runtime.
- **Malicious Operator:** The threat could also be initiated from the factory floor, by a malicious operator accessing the hardware to get access to confidential information.

From these attack scenarios, we define the security asset in the FN as the confidentiality, integrity, and availability properties for FN components and for the data flow transiting through the device. To ensure integrity and confidentiality, the FN requires encryption support and secure storage. A robust system design must protect the user-level software enclaves to avoid interference exploits.

D. Fog Node Communication

An essential requirement for the FCP is that it accommodates multiple communication technologies and protocols. Especially, converging IT and OT requires new deterministic networks communication [15] such as the TSN standards. Supporting TSN in all components of the FCP is an essential requirement. Similarly important is the support of legacy protocols, e.g., traditional fieldbusses or industrial Ethernet solutions. As shown in UC 2 and UC 3, there is a need for brownfield implementations.

Another communication-related requirement is the support of legacy and state of the art protocols. MQTT and OPC Classic are still widely used in industry and novel protocols and middleware such as OPC UA or DDS, have become the standard for new installations. Moreover, specific gateway applications hosted on FNs allow the bridging of various protocols and middleware [16].

E. Resource Management

In a FCP, in which nodes are placed at different locations and the total computational resources is divided between multiple FNs, collaboration represents the core characteristic to enable deployment and execution of different applications closer to the edge of the network. A collaboration between nodes is obtained with the help of novel resource management techniques as indicated by the resource management module presented in UC 1 and UC 3. However, these techniques comes with a set of requirements that must be fulfilled by the FCP.

We identify and discuss three distinct core requirements needed by the resource management module:

- **FN Monitoring Capabilities:** The resource management module aims at deploying IoT applications on a FCP composed of multiple distributed FNs that share a communication link; applications that have stringent requirements such as low latency and increased privacy and security. As a result, the monitoring capabilities of a FCP represents a prime requirement. To correctly deploy IoT applications, we must be able to know, at deployment time, some characteristics of the platform like latency and bandwidth; both equally important for checking the satisfiability of a deployment.
- **Resource Management Core Requirements:** A core requirement for a FCP is represented by the existence of specific tools that enable the correct functionality of the resource management module. In our case, the deployment technique used is based on Satisfiability Modulo Theories (SMT) [17] to ensure that satisfiable mappings are obtained.
- **FN Communication Requirements:** In a FCP, horizontal communication of FNs is essential for deployment of the components of distributed applications.

IV. Proposed Fog Computing Platform

The convergence of IT and OT will be supported by consolidating functionality of PLCs, industrial PCs, and gateways on a FCP. Every function in the domain of IT and OT consists of three dimensions that have to be addressed: processing, horizontal connectivity, and possibly vertical connectivity. Each dimension comes with a set of requirements that we have derived from the use cases in the previous section. We will now discuss the capabilities of a FCP that are needed to fulfill these requirements. Figure 8 provides an overview of the FCP hardware and software stack.

A. Hardware Platform and Virtualization

In the dimension of processing, the FCP allows consolidation of functionality on a FN by utilizing COTS MCPs that allow for parallel processing. The FCP has to ensure safe and secure isolation of functionality that share a hardware

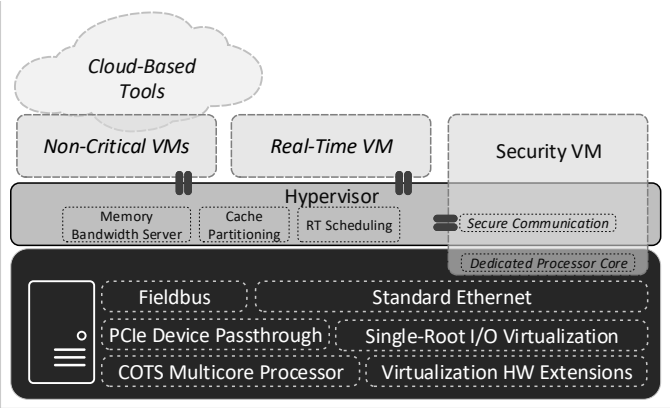

Fig. 8: Generic Fog Node architecture that is applicable for all discussed use cases.

platform. Therefore, we utilize a hypervisor to encapsulate each functionality in a separate VM enforcing strict isolation of co-located VMs by means of partitioning resources such as the processor, memory subsystem, and networking devices.

The hypervisor partitions the processor either spatially or temporally. In case of spatial partitioning, the hypervisor dedicates one or more processor cores exclusively to a single VM. This is particularly needed with regard to critical VMs with very strict safety and security requirements. The strict spatial separation of distinct critical domains guarantees basic security properties such as confidentiality, integrity, and availability, given well-defined secure communication channels between critical VMs and the rest of the system [18] as depicted by the *Security VM* in Figure 8. In case of less critical VMs, the hypervisor can partition the processor temporally by scheduling VMs on remaining, not spatially partitioned, processor cores. Thereby, the selected scheduling algorithm fundamentally impacts the performance properties, such as timeliness of execution and throughput, of the VMs. The scheduling of VMs on processor cores by the hypervisor and the scheduling of the guest operating system (OS) inside the VM form a scheduling hierarchy. Research in the field of hierarchical scheduling has shown that real-time performance with state-of-the-art hypervisors is feasible [19]. However, timing interference caused by the main memory subsystem can degrade the real-time performance.

The main memory subsystem includes the main memory itself, last level caches (LLCs) that are shared between processor cores, and the bus that connects the processor and the main memory. The hypervisor spatially partitions the main memory as well as the LLCs between VMs. Cache partitioning mechanisms have proven to drastically improve timing determinism of virtualized COTS MCP platforms [20]. Furthermore, the hypervisor mitigates timing interference stemming from the main memory bus by temporally partitioning the bus utilizing a memory bandwidth server [21].

B. Connectivity

In order to enable horizontal connectivity of consolidated functionality, the FN implements a field bus for communication with machinery and a standard Ethernet network interface card (NIC) for communication with other FNs and PLCs on the factory floor. The fieldbus is passed through to a dedicated real-time VM. The NIC and the FN's chipset on the other hand support single-root I/O virtualization (SR-IOV) that allows sharing a PCI Express device between multiple VMs via so-called virtual functions whereas the hardware supports the hypervisor implementing spatial partitioning of the virtual functions of respective VMs. SR-IOV virtualization also ensures high throughput vertical connectivity of the VMs running on the FN [22]. We utilize the TSN Ethernet standards [15] to enable horizontal real-time communication with PLCs and other FNs.

C. Resource Management

Besides the ability to combine IT and OT, the proposed FCP brings one more important feature that enables the distribution of new applications on the factory floor making the manufacturing process more efficient, i.e., resource management. Resource management enables the deployment of applications on Fog Nodes aiming to optimize the usage of available resources. Furthermore, it can minimize the downtime of the factory as well as the maintenance cost by allowing to perform updates at runtime. To achieve such functionality, all Fog Nodes run a local resource management module that empowers them to collaborate by sharing resources to achieve a common goal, i.e., to execute the application and satisfy application requirements. Apart from the resource management module found on each Fog Node, the FCP provides additional modules to enable users to deploy applications, i.e., GUI and application marketplace. The GUI allows an operator to interact with the FCP, giving the possibility to choose the application that is being deployed as well as monitoring the already installed applications. The marketplace contains all applications that are available for deployment.

Finally, when combining all presented modules, there are three simple steps to transparently deploy an application from the marketplace to the FCP. First, the user connects to the GUI and selects an application to be deployed from the marketplace. Once the application is chosen, the user chooses a preferred FN to deploy it on. Lastly, the chosen FN becomes the application coordinator using the resource management module to ensure the application's correct functionality. For example, in case of a high load on the FN the coordinator might migrate the application to a suited neighboring FN.

V. Research Challenges

The FCP introduces multiple advantages when implemented in a smart factory by bridging IT and OT. However, there are multiple research challenges that must be solved before being adopted by different industries. In this section, we discuss these challenges from the perspective of different FCP components, i.e., device communication, virtualization, Fog Node security, and resource management.

A. Virtualization

The proposed FCP relies on a hypervisor that utilizes state-of-the-art mechanisms, such as cache partitioning and memory bandwidth servers, to minimize timing interference of co-located VMs. However, with the progression of the standardization of the TSN standards more and more applications for those standards will emerge and eventually find their way to industrial automation, due to their easy applicability and interoperability. In particular the TSN Qbv standard [23] brings enhancement for scheduled traffic via standard Ethernet, similar to already existing time-triggered communication protocols, such as TTEthernet or CAN, yet the current virtualization stack lacks general support for time-triggered communication. A simple solution that still enables time-triggered communication of VMs is statically assigning VMs to processor cores of a FN so they can execute and send, receive, and process packets in accordance to the global network schedule. However, statically assigning resources limit the flexibility of the FCP and often result in underutilization of computational resource. For example, a simple control loop that requires time-triggered communication might not fully utilize its exclusively assigned processor core.

As a result, we require a hypervisor that can schedule the execution of VMs according to the global network schedule. This raises a series of research problems:

- **Synchronized Time**: A distributed control system on the factory floor only meets its global network schedule due to the use of a common time base that is being established using a clock synchronization protocol, such as IEEE 1588-2008 or IEEE 802.1AS as defined by the TSN standards. As a result, the hypervisor being in control of every aspect of the execution of its VMs must have access to the common time base. Research is needed in how to enable the hypervisor to access and use a common time base.
- **Time-Triggered Hypervisor:** The hypervisor must dispatch its VMs in accordance to the global network schedule. An event-driven approach, in which the hypervisor enables a VM when a scheduled network packet arrives, could lead to unpredictable attack vectors since an intruder could introduce malicious scheduled traffic that tampers with the scheduling of the hypervisor. Therefore, we claim the need for a time-triggered hypervisor in industrial automation similar to ARINC 653 hypervisors in avionics [24].
- **Dynamic Allocation of Spare Resources:** The generation of global communication schedules requires estimates of the worst-case execution times (WCETs) of all components. Obtaining a WCET estimations for functions running on a MCP are known to be difficult [25]. Even if mechanisms such as cache partitioning or memory bandwidth servers reduce timing non-determinism, WCET estimates usually succeed average case execution times. Therefore, most of the time the virtualized FN will be underutilized mitigating the benefits of dispatching VMs in accordance to the global network schedule. As a result, there is research needed in how to utilize spare execution time resulting from overestimating the WCET and how to make the hypervisor's timing even more deterministic so WCET estimation becomes more accurate.

B. Fog Node Security

Partitioning based security protects the system against system interfaces hijack attempts. However, some intrusions comply with system enclaves but exploit authorized channels to gain illegitimate access. For example, return oriented programming attacks [26] exploit program vulnerabilities by reusing code to hijack a program control flow to eventually run shell code on the target. Even though the Fog Node design is well separated, some software modules can possibly communicate. For example, two applications run on the node, a real-time critical server interfacing with the OT and an IT client performing analysis on OT data; an adversary could try to manipulate the client interface to raise deny of service on the server. Therefore, the integrity of user-level software's execution is a necessary asset to protect. Intrusion detection is an active field of research, particularly developed in IT; e.g. [27]. The deployment of such security in a mixed-criticality context has to our knowledge not been investigated yet.

C. Device Communication

Building an FCP environment, combining OT and IT and consolidating functionality on single components such as Fog Nodes, require a well-organized network communication. The proposed FCP solves most of these challenges by offering a wide choice of connectivity. However, open hurdles remain especially concerning legacy systems using vendor-specific protocols. The advance of TSN and OPC UA and the joint effort in the industry to create standards points towards the right direction, yet, the need for gateway applications between different protocols will prevail as older systems will continue operating for several years [16]. Accordingly, gateways need to accommodate the FCP's flexibility and scalability.

The configuration of a gateway requires a large amount of knowledge about the configuration of the respective middlewares. This manual configuration reduces the possibility of automatic distribution and the scalability of the gateway applications in data-intensive installations. Therefore, research to simplify or automate the configuration process needs attention from the industry as well as from academia.

D. Resource Management

Moving computational resources closer to the edge of the network stands at the core of the Fog Computing paradigm; however, this is not a trivial task. Hosting successfully applications on a FCP requires novel resource management techniques that enable the efficient utilization of all available resources found at the edge. According to [28], the resource management can be classified into five different categories, i.e., resource estimation helps to estimate how many resources an application's component requires, resource discovery aims at

discovering available resources at the edge, resource allocation finds a satisfiable mapping of components to Fog Nodes such that the application requirements are met, resource sharing ensures Fog Node collaboration and resource optimization combines all previously mentioned categories to optimize the utilization of available resources.

Each category has its challenges to overcome, however, in the context of smart manufacturing some of these categories are less important. For example, in a manufacturing scenario, the resource estimation, resource discovery, and resource sharing categories are more easily applied since the network is controlled by the same organization. As a result, the focus is on providing a resource allocation technique to ensure the correct functionality of both the deployed application as well as the existent running applications.

VI. Conclusion

The paper presents three industrial use cases with a specific focus on the consolidation of IT/OT functionality by using a Fog Computing Platform (FCP) that provides resources closer to the edge of the network. Each use case exemplifies scenarios found in the industry. The comparison of the derived use case requirements with a proposed FCP architecture revealed open research challenges in the areas of *hardware, virtualization, security, communication and resource management*. Further studies include research for solving the identified challenges and investigating possible key performance indicators (KPIs) for evaluating the effectiveness of OT/IT consolidation projects. Please visit http://www.fora-etn.eu/ for more details.

Acknowledgment

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 764785, FORA—Fog Computing for Robotics and Industrial Automation. Special thanks goes to, Dr. Wilfried Steiner, Prof. Schahram Dustdar, Dr. Sergey Tverdyshev and Prof. Gerhard Fohler

References

[1] T. J. Williams, "The purdue enterprise reference architecture," in *Proceedings of the JSPE/IFIP TC5/WG5.3 Workshop on the Design of Information Infrastructure Systems for Manufacturing*, ser. DIISM '93. NLD: North-Holland Publishing Co., 1993, p. 43–64.

[2] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, March 2017.

[3] S. A. Boyer, *Scada: Supervisory Control And Data Acquisition*, 4th ed. Research Triangle Park, NC, USA: International Society of Automation, 2009.

[4] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, ser. MCC '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 13–16.

[5] A. Ismail and W. Kastner, "A middleware architecture for vertical integration," in *2016 1st International Workshop on Cyber-Physical Production Systems (CPPS)*, April 2016, pp. 1–4.

[6] M. C. Lucas-Estañ, T. P. Raptis, M. Sepulcre, A. Passarella, C. Regueiro, and O. Lazaro, "A software defined hierarchical communication and data management architecture for industry 4.0," in *2018 14th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, Feb 2018, pp. 37–44.

[7] M. Prist, A. Monteriù, A. Freddi, E. Pallotta, P. Cicconi, F. Giuggioloni, E. Caizer, C. Verdini, and S. Longhi, "Cyber-physical manufacturing systems for industry 4.0: Architectural approach and pilot case," in *2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0 IoT)*, June 2019, pp. 219–224.

[8] M. Azarmipour, H. Elfaham, C. Gries, and U. Epple, "Plc 4.0: A control system for industry 4.0," in *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, vol. 1, Oct 2019, pp. 5513–5518.

[9] D. Kiel, C. Arnold, and K.-I. Voigt, "The influence of the Industrial Internet of Things on business models of established manufacturing companies – A business level perspective," *Technovation*, vol. 68, no. C, pp. 4–19, 2017.

[10] FORA, "FORA European Training Network http://www.fora-etn.eu/."

[11] C. Avasalcai, C. Tsigkanos, and S. Dustdar, "Decentralized resource auctioning for latency-sensitive edge computing," in *IEEE International Conference on Edge Computing (EDGE)*, 2019.

[12] L. Chen, S. Patel, H. Shen, and Z. Zhou, "Profiling and understanding virtualization overhead in cloud," in *2015 44th International Conference on Parallel Processing*, 2015, pp. 31–40.

[13] M. Gutiérrez, A. Ademaj, W. Steiner, R. Dobrin, and S. Punnekkat, "Self-configuration of IEEE 802.1 TSN networks," *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, pp. 1–8, 2018.

[14] SYSGO, "https://www.sysgo.com/products/pikeos-hypervisor/."

[15] W. Steiner and S. Poledna, "Fog computing as enabler for the industrial internet of things," *e & i Elektrotechnik und Informationstechnik*, vol. 133, no. 7, pp. 310–314, Nov. 2016.

[16] R. Joshi, P. Didier, J. Jimenez, and T. Carey, "The Industrial Internet of Things Volume G5: Connectivity Framework," Tech. Rep., 2018.

[17] C. Barrett and C. Tinelli, *Satisfiability Modulo Theories*. Cham: Springer International Publishing, 2018, pp. 305–343.

[18] J. Rushby, "The design and verification of secure systems," in *Eighth ACM Symposium on Operating System Principles*, 1981.

[19] S. Xi, M. Xu, C. Lu, L. T. X. Phan, C. D. Gill, O. Sokolsky, and I. Lee, "Real-time multi-core virtual machine scheduling in xen," in *2014 International Conference on Embedded Software, EMSOFT 2014, New Delhi, India, October 12-17, 2014*, 2014, pp. 27:1–27:10.

[20] M. Xu, L. T. X. Phan, X. Phan, H. Choi, and I. Lee, "vcat: Dynamic cache management using CAT virtualization," in *2017 IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS 2017, Pittsburg, PA, USA, April 18-21, 2017*, 2017, pp. 211–222.

[21] H. Yun, G. Yao, R. Pellizzoni, M. Caccamo, and L. Sha, "Memguard: Memory bandwidth reservation system for efficient performance isolation in multi-core platforms," in *19th IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS 2013, Philadelphia, PA, USA, April 9-11, 2013*, 2013, pp. 55–64.

[22] D. Muench, O. Isfort, K. Mueller, M. Paulitsch, and A. Herkersdorf, "Hardware-based i/o virtualization for mixed criticality real-time systems using pcie sr-iov," in *2013 IEEE 16th International Conference on Computational Science and Engineering*, Dec 2013, pp. 706–713.

[23] "Ieee standard for local and metropolitan area networks – bridges and bridged networks - amendment 25: Enhancements for scheduled traffic," *IEEE Std 802.1Qbv-2015 (Amendment to IEEE Std 802.1Q-2014 as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd-2015, and IEEE Std 802.1Q-2014/Cor 1-2015)*, pp. 1–57, March 2016.

[24] M. Masmano, I. Ripoll, A. Crespo, and J. J. Metge, "Xtratum: a hypervisor for safety critical embedded systems," in *In: Proceedings of the 11th Real-Time Linux Workshop*, 2009.

[25] R. Wilhelm, J. Engblom, A. Ermedahl, N. Holsti, S. Thesing, G. Bernat, C. Ferdinand, R. Heckmann, T. Mitra, F. Mueller, I. Puaut, P. Puschner, J. Staschulat, and P. Stenstrom, "The worst-case execution-time problem - overview of methods and survey of tools." *ACM Trans. Embedded Comput. Syst.*, vol. 7, 01 2008.

[26] R. Roemer, E. Buchanan, H. Shacham, and S. Savage, "Return-oriented programming: Systems, languages, and applications," in *CM Transactions on Information and System Security - TISSEC*, vol. 15, 03 2012, pp. 1–34.

[27] S. Jose, D. Malathi, B. Reddy, and D. Jayaseeli, "A survey on anomaly based host intrusion detection system," in *Journal of Physics: Conference Series*, vol. 1000, apr 2018, p. 012049.

[28] K. Toczé and S. Nadjm-Tehrani, "A taxonomy for management and optimization of multiple resources in edge computing," *CoRR*, vol. abs/1801.05610, 2018.