

Article

Technical Interoperability for Machine Connectivity on the Shop Floor

Magnus Åkerman ^{1,*}, Johan Stahre ¹, Ulrika Engström ², Ola Angelsmark ², Daniel McGillivray ², Torgny Holmberg ², Maja Barring ¹, Camilla Lundgren ¹, Martin Friis ³, and Åsa Fast-Berglund ¹

¹ Industrial and Materials Science, Chalmers University of Technology

² Ericsson Research, Ericsson

³ Manufacturing Development Centre, SKF

* Correspondence: magnus.akerman@chalmers.se; Tel.: +46-(0)729-636518

Abstract: This paper presents technical solutions to increase Industry 4.0 maturity. Within the “5G-Enabled Manufacturing” project, a 5G network has been deployed at the shop floor to enable fast and scalable connectivity. This network was used to connect a grinding machine to a private cloud to enable visibility and transparency of the production data, which is the basis for Industry 4.0 and smart manufacturing. Results indicate a present lack of commercially available product independent solutions for interconnection and data transfer despite the availability of open standards and well-documented demonstrator projects. The solution is described and discussed regarding technical interoperability, focusing on the system layout, communication standards, and open systems. From the discussion, it is derived that manufacturing end-users are in need to expand and further internalize knowledge on future information and communication technologies to reduce their dependency on equipment and technology providers.

Keywords: interoperability; IoT; connectivity; Industry 4.0; smart manufacturing.

1. Introduction

The next generation cellular wireless technologies, 5G, will enable a scalable all-in-one connectivity platform that supports connectivity of ubiquitous equipment, enabling a fourth industrial revolution (Industry 4.0). The technology only partially exists today, as requirements are being refined and formalized into the open 3GPP international standard that is set for the near future [1]. However, connectivity infrastructure is only one piece of the puzzle towards the goal to achieve highly flexible fully interconnected manufacturing system. As suggested by Vernadat [2], the ability for systems to interconnect can be divided into technical, semantic, and organizational levels, that are incorporated in the holistic concept of interoperability. Technical interoperability includes connectivity but also deals with syntactical and architectural aspects. Interoperability has been an important part of manufacturing system design since software systems were first introduced to aid resource management during the Computer Integrated Manufacturing (CIM) era during the 1990s. Since then there has been a rapid development of information technologies, e.g. Cloud Computing; Big Data; Internet of Things (IoT); Internet of Services (IoS); and Cyber-Physical Systems (CPS) [3-7]. Such technologies need to be further adopted by the industry in order to build smart manufacturing systems, which emphasizes the importance of interoperability [8, 9].

The work presented has been conducted within the research project “5G-Enabled Manufacturing” (5GEM). The project is a collaboration between a large manufacturing company (SKF), a large telecommunication system provider (Ericsson), and academia (Chalmers University of Technology). The project investigates the general question of how 5G networks can be utilized in a production system. The key question is how the manufacturing industry can enhance the speed and quality of the application of new communication technology to improve manufacturing performance.

The aim of this paper is to present challenges and solutions regarding technical interoperability in an industrial setting. A real-world case of connecting a grinding machine to a cloud infrastructure using the current state of the art cellular technology is described and presented in detail. The results are discussed focusing on technical interoperability, divided into the three areas: system layout, communication standards, and open systems, beyond fast and scalable connectivity to increase Industry 4.0 maturity.

2. Background

2.1 Industrial digitization and Industry 4.0 maturity

A key issue for manufacturing enterprises is to ensure that the application of new technologies adds value to their core business. A key goal of Industry 4.0 is to create cyber-physical production systems (CPPS) from which data-driven autonomous systems can emerge. The Industry 4.0 Maturity Index model [10] is a way to measure how far an enterprise has reached towards this goal. The model sets several prerequisites needed for an industrial system to take advantage of new capabilities. To reach these different steps, many things need to be in place. **Figure 1** shows a high-level list of what technological solutions, or paradigms, that can be loosely connected with each step. The first step, computerization, is to have a digital system as opposed to a purely mechanical one, a development that once triggered the third industrial revolution during the 1950's. The second step is to connect the various digital systems. The academic discussion about the complexity of such interconnections intensified during the 1980's, during the Computer Integrated Manufacturing (CIM) era, which leads to multiple frameworks of enterprise integration and interoperability [2, 11].

The third step is named visibility, which means to have a digital model of the factory. Data from machines and sensors need to be collected and stored continuously so that the current state of the production system is always known and based on facts. Collecting all the relevant data in a complex manufacturing environment requires an efficient and scalable information system. A solution is to utilize shared resource pooling and Cloud Computing technologies [12]. According to the National Institute of Standards and Technology (NIST) cloud computing 'is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction' [3].

I4.0 maturity index four is called transparency, which means to make sense of the digital models to understand the current situation. A central aspect of this is to use algorithms for data analysis in Big Data applications. A popular definition of Big Data includes the four important V's of data and comes from an IDC report 2011 [13]. "Big data technologies describe a new generation of technologies and architectures, designed to economically extract value from very large volumes of a wide variety of data, by enabling high-velocity capture, discovery, and/or analysis."

In the fifth step, predictive capacity, real-time data and analyzed information needs are linked, aggregated, simulated etc. into useful information which can be used for decision-making. This aligns with a certain aspect of IoT that focus on a layered architecture and the use of middleware [5]. Encapsulating functionality in well-defined services enables and simplifies data sharing between new and legacy systems with IoS [6, 14].

The sixth and last step in the Industry 4.0 Maturity Index model is called adaptability. It is the end goal of both Industry 4.0 and IoT, which is also referred to as smart manufacturing or smart industry [15, 16]. A Cyber-Physical Production System (CPPS) is an adaptable system with the capability to sense its environment and dynamically react to it. Since this reaction need to function at different time scales, depending on which part of the system that needs to adapt, there is a need for programming and modeling languages with such support [7].

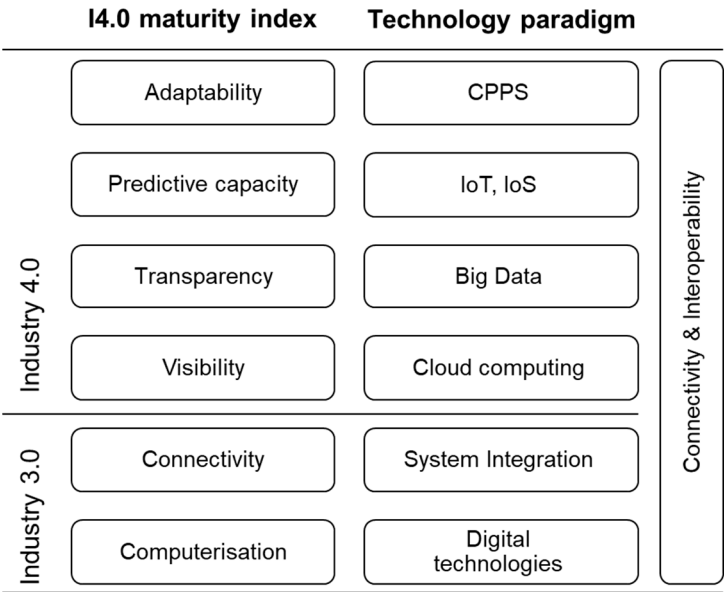


Figure 1. A high-level illustration of what technological solutions, or paradigms, that can be loosely connected with each step in the Industry 4.0 maturity index model.

2.2 Technical interoperability and enabling technologies

Interoperability issues relate to technical, semantic, or organizational aspects [2]. Technical interoperability aspects regard the data exchange between different information technologies at the physical and syntactical level. When considering data for CPS in manufacturing it is important to differentiate between configuration data and run-time data [7]. Configuration data is generated during the design of the system and describes the physical parts of the system. Run-time data is generated during the operational phase and describes the status of the manufacturing process. Optimal synergies are achieved when these data types can share the same models, which require usage of industrial standards for both design and operations. The automation engineering standard Automation ML (AML) and communication standard OPC Unified Architecture (OPC UA) are examples of standards that can be semantically linked [17].

Another core consideration when interconnecting systems is the layout, meaning how each entity relates to others in terms of computation and decision making. The classical control system is hierarchical and deterministic, operating with centralized decision making. In such systems, data only flows vertically between child and parent nodes. The IoT paradigm assumes a decentralized approach where data can flow horizontally. Such systems are event-driven and non-deterministic where a more complex system emerge, which needs new methods to be managed [7]. A practical approach to achieve CPS in manufacturing is to create a semi-hierarchical approach with emphasis on modularization. An example is a solution by Wang and Haghighi [18], promoting a holonic (modularized) system where each Holon is controlled by a software agent (autonomous unit).

Approaches to the layout problem, developed outside of the manufacturing domain, include various platforms and middleware for IoT applications that have gained traction lately. IoT platforms, also called frameworks or middleware, are systems that simplifies the integration of things (e.g. sensors, machines, and equipment) through decoupling to enable IoT applications. There are three types of IoT systems: service-based, cloud-based, and actor-based [19]. Service-based systems are centralized heavy-weight generic platforms often deployed in a cloud environment [20]. Cloud-based systems are application specific services often tied to a specific product. Actor-based middleware systems are flexible and scalable since they support deployment in different layers. Examples of actor-based middleware are the Ptolemy framework, Node-RED, and Calvin [21-23]. Actors are reusable function blocks that can easily be deployed on different hardware. Some platforms also allow migration of actors to where and when they are needed as long as the system

supports the needed requirements [24, 25]. This allows computations closer to the network edge, thus allowing for a more decentralized system, known as edge-networks or fog computing [26].

IoT systems are natural aggregators of data, often including data analytics capabilities for Big Data applications. When data volumes grow the system requires technologies that can distribute computing and storage capabilities and send a large amount of data between the different nodes. There are several open-source projects that can enable these systems, such as Hadoop, Spark, and Kafka [27-29] for distributed computing and data streaming. For data storage, the NoSQL databases, which are more flexible than the traditional relational databases, is a key technology [4]. The term NoSQL, which should be read as “not only SQL”, incorporates characteristics like non-relational, distributed, open-source, and horizontally scalable [30].

Frameworks for CPS and IoT often include a middle layer that connects objects and applications through services [31-33]. This is called a Service Oriented Architecture (SOA), which principle is to connect business functions (services) from providers to consumers by using service locators or service brokers [34]. A definition of a service, therefore, need to account for business and technical perspectives as well as the consumer and provider perspectives but from the technical perspective, a service is an encapsulation of functionality [35]. SOA is often used synonymously with Web-services but this is only one implementation. Web services often comply with the Representational State Transfer (REST) style that implies stateless interactions and a hierarchical resource representation [36]. SOA is also supported by the above-mentioned communication protocol for industrial automation, OPC UA. New communication protocols have also been developed to support ubiquitous and sometimes limited hardware, such as Message Queuing Telemetry Transport (MQTT) and Constrained Application Protocol (CoAP) [37, 38].

To interconnect all the ubiquitous equipment that is assumed in the IoT paradigm requires a fast, robust, and scalable connectivity platform. The next generation open telecommunication standard of cellular wireless technologies, 5G, is estimated to be commercially available around 2020. As in previous cellular telecom standards, 5G is developed within the 3G Partnership Project (3GPP) framework, building on components from the 4G/LTE, the prevalent telecom standard. Some of the key characteristics of 5G are higher data rates, shorter delays, and high reliability [39]. Requirements integrated into the new standards reflect the needs of tomorrow, i.e. “a fully mobile and connected society” [40]. Ongoing standardization efforts have defined several scenarios and connected them with sets of requirements [1]. These requirements need several solutions, most of which already partially exist [41].

3. Method and project description

The methodology applied to this work is the design science approach. Design science research centers around artificial artifacts, designed and developed with the researchers deeply involved. This methodology has been successfully used in information systems research where an artifact can be software, hardware, a method, a theory etc. [42]. The crucial aspects of design science for information systems research are illustrated in the information systems research framework developed by Hevner, et al. [43]. An artifact is designed and evaluated in a design/test cycle. The design/test cycle gets its relevance in an iterative process of adapting to real-life business needs and evaluating the results in the intended environment, which in this case is the shop floor. The design/test cycle is built on theories from the common knowledge base, and new learnings from the design/test cycle can often result in new theories that should be communicated to contribute to new knowledge and expand the knowledge base, this is called the rigor cycle.

As mentioned above, the project (5GEM) investigates how the next generation cellular networks, 5G, can support manufacturing industries in implementing smart manufacturing systems. Part of the investigations was to deploy a cellular network on the shop floor, connect a grinding machine, collect data from the grinding process, and to utilize that data in real-time and analytics applications. The results from the applications provide the system feedback loop that can interconnect with an automated system, e.g. controlling the machine directly, or be displayed to manufacturing operators through a mobile support system (mobile application) already in operation at the shop-floor. Beyond

the fast and scalable connectivity that the infrastructure provided, this required software and hardware designed to collect and transfer the data between source and utilization, which are the relevant artifacts for this research. To connect a producing grinding machine to a cloud network seems like a straightforward task. However, there are many things to consider regarding the technical aspects. First, there is a question of what data that should be collected, then there is the problem of how to collect that data, then that data needs to be transferred. Solutions for the latter are the relevant artifacts that are further described in the results chapter. **Figure 2** shows how these artifacts relate to the project goal and infrastructure.

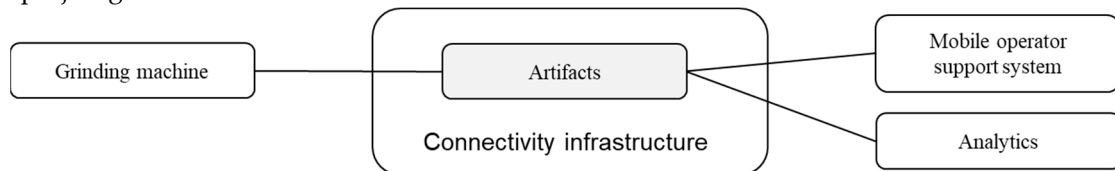


Figure 2. The artifacts described in the results section are the enabling technologies connecting the grinding machine to the data consumers on top of the basic connectivity infrastructure.

3.1. Connectivity infrastructure

The project infrastructure is distributed over three Swedish cities: Gothenburg, Lund, and Stockholm. See **Figure 3** for an overview of the setup. In Gothenburg, the SKF factory is connected to Ericsson’s facilities through a radio-link. The Gothenburg site is connected to Ericsson’s core network and linked to facilities in Lund which is a data center that hosts a private cloud in terms of processing and storage. The gathered data is also distributed to the data analytics site in Stockholm. On the shop floor, the closest thing to a 5G network was deployed, which is a dedicated LTE network with “5G enabling technologies”. Devices are connected to the network through either embedded modem (e.g. LTE enabled mobile phones) or using external USB modems.

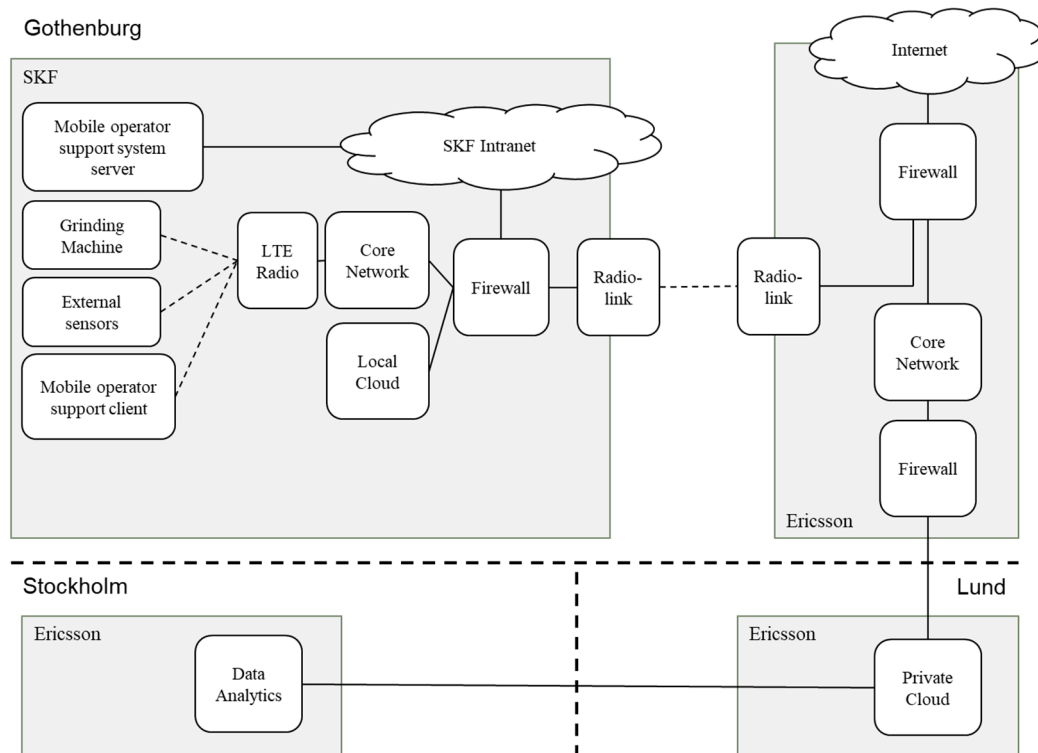


Figure 3. Summary of the project connectivity infrastructure.

3.2. Mobile Operator Support System

SKF has a specialized operator support system with an iOS client [44]. This system can take advantage of a 5G connection in two ways. The first way is that the iOS hardware can directly connect to a 5G system, which enables the user to send and receive data faster. This is useful when e.g. streaming video instructions. The second way is to interconnect the operator support system with the real-time data collected from the grinding machine and the possible results that come from the data analysis.

4. Results

Data from the grinding process is collected through four separate systems: the grinding machine onboard computer, a vibration system, IO-Link sensors, and an embedded sensor. While it is possible, to some extent, to connect the sensors to the machine PLC and transfer all the data from the onboard computer, that solution was opted out because it would require too much change in the machine, a task requiring a very limited resource. Each system supports different communication standards and/or methods, which made the overall system more complicated. To simplify this, the actor-based IoT middleware Calvin was used for aligning the data transfer. Calvin is a distributed software system, which allows its function blocks to run on different hardware while they communicate seamlessly with each other. The reference implementation of the Calvin platform, Calvin-base [23], runs on most Linux systems, which means that Calvin can run in hardware in the local cloud as well as in the data center.

The different connected systems are summarized in **Table 1**. The grinding machine computer and the vibration system are connected through the OPC UA standard, which they support to various degrees. The external sensors support the IO-Link protocol, which allows data extraction directly over TCP/IP. However, to simplify the connection to the Calvin actor, another gateway was developed that translates the TCP stream and share the data as a RESTful web service. This translation software was developed using the Play framework [45] and the raw data could be mapped using the IO-link data sheets. The embedded sensor is a temperature sensor directly connected to a Raspberry Pi computer. Since Calvin can run on Raspbian, the Linux distribution supported by the Raspberry Pi, a Calvin actor can communicate directly with the embedded sensor.

Table 1. The different connected system from which the data is collected.

Data source	Description	Protocol
Machine onboard computer	The machine computer can provide most of the important data related to the grinding process.	OPC UA
Vibration system	The external vibration system consists of vibration sensors that are mounted on the machine, it sends an aggregated version of the vibration data.	OPC UA
IO-Link sensors	IO-Link sensors communicate (using IO-Link) with a gateway, called IO-Link master, from which it is possible to retrieve sensor data over TCP/IP. A second gateway translates the data and makes it available as a web service.	REST API
Embedded Sensor	Temperature sensor connected to a Raspberry Pi computer.	N/A

To connect the systems to Calvin, support for each communication or device was needed. This was achieved by implementing a plugin which handles all communication with the device or service in the platform abstraction layer, called *calvinsys*. The Calvin application, visualized in **Figure 4**, is running in a Calvin system spanning three instances – one on a Raspberry Pi located near the machine, one in the local cloud in the factory, and one running in the Ericsson Research data center. Letting the platform handle all specifics of the external service gives the illusion of the service being

part of the platform itself. The function blocks, or actors, are distributed on the three calvinsys instances and the communication between them is seamless, meaning that the physical separation of the calvinsys instances is invisible to the function blocks.

When the data has been transferred to the cloud through the Calvin application, it is distributed to three different data consumers: the database, the analytics application, and the mobile operator support system. The operator support system already utilizes the publish/subscribe protocol MQTT as the default communication protocol. After deciding on the structure for the topics for the MQTT messages, a Calvin function block was implemented that publish the needed datatypes. The data is also streamed to the data analytics center with the help of Kafka [29], which can also fetch stored information from the database if needed. The database, called MongoDB, is a general-purpose NoSQL database. Here the data is stored with value, timestamp, and origin of the data as key components.

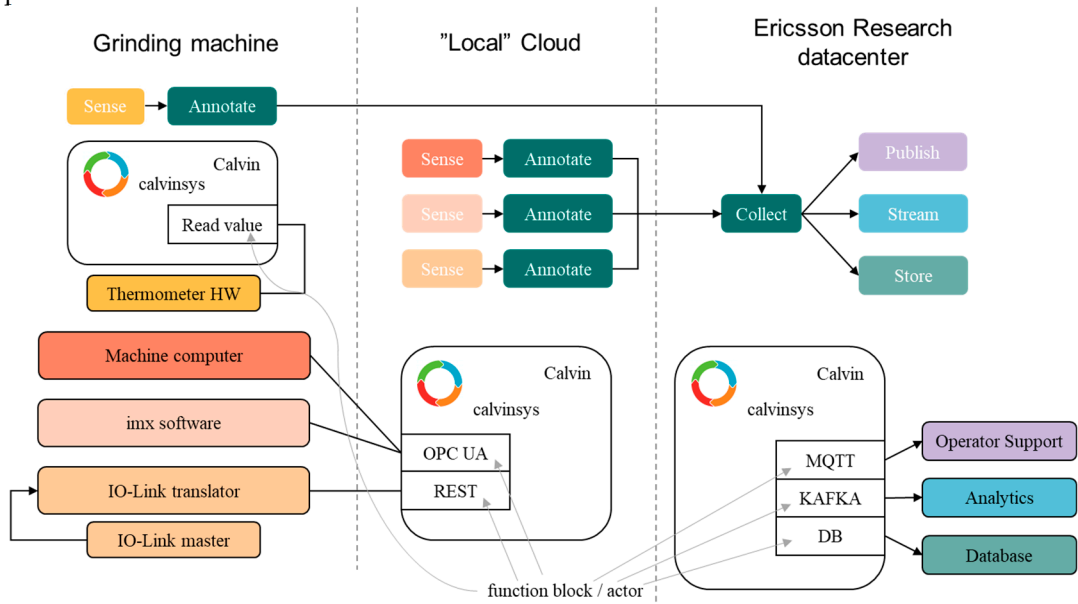


Figure 4. The Calvin application is distributed over three physical locations. One in the Raspberry Pi located by the machine that reads the thermometer values, one in the local cloud in the factory reading data from the machine and external sensors, and finally in the central private cloud to distribute and store the data.

Each day, the application explained above collects and transmits about one million data entries, most of them from the onboard machine computer. The data analytics experts use these entries to test and evaluate new algorithms and approaches. At the shop floor, the operators can see and act on a few of these data entries and remote monitoring and control can now easily be expanded upon.

5. Discussion

The described artifacts have been implemented and tested in a real-life production system at SKF and have been proven to enable several steps towards CPPS. Scalable connectivity is achieved through 5G for both stationary and mobile equipment. By exploiting Calvin as a common connectivity platform together with open industry standards for data communication and storage, relevant information is made visible in a cloud environment. Kafka and MongoDB are used to stream aggregated data to the analytics center that can apply algorithms and increase system transparency.

Since Calvin also aligns the different type of connected systems, both locally in the factory and centrally in the private cloud, it would also simplify the process of connecting more machines and equipment, a prerequisite for more predictive capacity. Furthermore, the mobile operator support system allows manufacturing operators to access real-time data and to share important information.

There are many ways to connect a machine on the shop floor and some of the described enabling technologies and application has been tested before [46, 47]. However, when practically collecting

data from a machine at the shop floor there seem to be no obvious solution that fits a generic machine setup. Machine suppliers are adding their own support for data collection but then there is the issue of data ownership and supporting a multitude of different monitoring systems. The solution described here is one generic way to acquire data from machines or equipment.

5.1. System layout

One important aspect regarding system layout is choosing between the different approaches when it is possible to get the data in different ways. The machine onboard computer is the most important system since it is the source of several datatypes that cannot be collected otherwise. Getting this data proved to be a challenge because the computer had to be reprogrammed to gain access to the desired data. This programming task directly, or indirectly, interfered with production since both the machine and the engineer are valuable resources needed to run the manufacturing process. If choosing e.g. IO-Link instead the solution is logically decoupled from the physical process.

Calvin is used for aligning all connections to the grinding machine. The result is identical data collection applications over different connections and with high reusability. Another advantage is the ability to run the function blocks, or actors, distributed in the local cloud. To have the platform hosted inside the factory allow the system to function in isolation for a period, meaning it can seamlessly store data to be transmitted later if the connectivity to the data center suddenly disappears [19]. The Calvin application is created by defining the data flows, which is also the approach in other IoT middleware. To create these data flows can be easier for people without experience in generic programming languages, which should also be considered when choosing solution [19]. When scaling the application, platforms like Calvin can also aid with the decision of software deployment using the required-based approach [48]. This means that the function blocks can only run where the correct requirement is met e.g. that there is a specific sensor present.

5.2. Communication standards

The plugin that needed to be developed in this situation was support for OPC-UA. While it was straightforward to use existing open source solutions for the actual communication, there were some issues with slight variations in the implementations that had to be considered. Upgrading software in an industrial setting is not always an option, the machine in question is crucial to the manufacturing process. This means that the adaptability of the IoT platform is important.

Two systems support OPC UA as the communication protocol, the grinding machine onboard computer and the IMX (vibration system), which is a natural choice since it is an open, platform-independent system that supports SOA and can enable other future synergies [17, 49]. A limitation of OPC UA that was evident during this project is the difference in the various implementations that is due to them being self-certified.

With the IO-link enabled sensors it is possible to extract the data directly bypassing the machine computer. That avoids meddling with the machine program but it can also miss out on the possibility of semantic alignment that adopting the OPC UA standard can enable. In this case, a separate translation software was developed which required some work but modern software frameworks simplified this task [45]. The solution presented here is a stand-alone REST API supporting the few sensors that were mounted to the machine. Such implementation could be made to support almost all IO-link sensors by connecting it to the common device library service that 2017-07-24 had more than 80% of all devices documented [19, 50]. Since the presented application was developed, the company IFM have released gateways with an IoT connection (**Figure 5**) which basically works similarly to the REST API that was developed for this application, with a web service exposing the sensor data as a generic JSON (JavaScript Object Notation) document.



Figure 5. IoT enabled IO-Link gateway to the right, the old one used for this application to the left.

MQTT is used to send real-time data to the mobile operator support system. The data is sent using the JSON structure, which is the de-facto standard for Web services and aligns well with the document database solution. There is little to say about that other than it shows the benefit of utilizing well-defined open standards.

5.3. Open systems

Without a doubt, the open-source solutions were crucial for the implementations in this project. This is true regarding both the data management infrastructure and end-point solutions. The open systems for process distribution, data streaming, and data storage are industry standard and developed and well tested through the emergence of large and well known social platforms [51]. For systems like Calvin, the open-source approach is important since existing and future implementations benefit from a large community that can share content. Using Raspberry Pi's or other low-cost microcomputers with attached hardware modules are excellent tools for testing and verifying IoT solutions. However, to take advantage of these systems requires own responsibility and overall system knowledge. Traditional industrial systems are hierarchical, centralized, and built for robustness. This leads to stand-alone systems since each subsystem are built fully deterministic and with a well-defined scope. Decentralized systems with more horizontal connections are more flexible and complex [7].

6. Conclusions

A solution has been implemented in a real-life industrial setting and discussed regarding technical interoperability, focusing on the system layout, communication standards, and open systems. The findings are tied to the key question of improving manufacturing performance by tying the technical implementation to the Industry 4.0 maturity Index model. From the discussion, it can be derived that manufacturing enterprises need to internalize the knowledge of how IT systems can be utilized to be able to connect their existing and new equipment while not being too reliant on technology providers. The example of IoT enabled IO-Link sensors also shows the fast development of available equipment but that this development will most likely align with existing efforts if current standards and state of the art solutions are followed.

Acknowledgments: The authors gratefully acknowledge the financial support from the national, Digital Industrial Pilot Programme operated by the Swedish Agency for Innovation Systems (VINNOVA).

References

1. "3GPP TS 22.261: Service requirements for the 5G system; Stage 1," Release 15 ed, 2017.
2. F. B. Vernadat, "Technical, semantic and organizational issues of enterprise interoperability and networking," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 13, pp. 728-733, 2010.
3. P. Mell and T. Grance, "The NIST Definition of Cloud Computing - Recommendations of the National Institute of Standards and Technology," 1047-6210, 2011.
4. M. Chen, S. Mao, and Y. Liu, "Big Data: A survey," *Mobile Netw Appl*, vol. 19, pp. 171-209, 2014.
5. L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, pp. 2787-2805, 10/28/ 2010.
6. C. Schroth and T. Janner, "Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services," *IT professional*, vol. 9, 2007.
7. L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, *et al.*, "Cyber-physical systems in manufacturing," *CIRP Annals - Manufacturing Technology*, vol. 65, pp. 621-641, 2016.
8. M. Hermann, T. Pentek, and B. Otto, "Design Principles for Industrie 4.0 Scenarios," in *49th Hawaii International Conference on System Sciences (HICSS)*, Hawaii, 2016.
9. S. Mittal, M. A. Khan, D. Romero, and T. Wuest, "Smart manufacturing: Characteristics, technologies and enabling factors," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 2017.
10. G. Schuh, R. Anderl, J. Gausemeier, M. t. Hompe, and W. Wahlster, "Industrie 4.0 Maturity Index: Managing the Digital Transformation of Companies," acatech STUDY, Munich2017.
11. D. Chen and N. Daclin, "Framework for enterprise interoperability," *Proc. of IFAC Workshop EI2N*, pp. 77-88, 2006.
12. Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *J Internet Serv Appl*, pp. 78-18, 2010.
13. J. Gantz and D. Reinsel, "Extracting Value from Chaos," 2011.
14. N. Papakostas, J. O'Connor, and G. Byrne, "Internet of things technologies in manufacturing: Application areas, challenges and outlook," pp. 126-131, 2016.
15. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, pp. 1645-1660, 9// 2013.
16. K.-D. Thoben, S. Wiesner, and T. Wuest, "'Industrie 4.0" and Smart Manufacturing – A Review of Research Issues and Application Examples," *Int. J. of Automation Technology*, vol. 11, pp. 4-16, 2017.
17. R. Henßen and M. Schleipen, "Interoperability between OPC UA and AutomationML," *The 8th International Conference on Digital Enterprise Technology*, 2014.
18. L. Wang and A. Haghighi, "Combined strength of holons, agents and function blocks in cyber-physical systems," *Journal of Manufacturing Systems*, vol. 40, pp. 25-34, 2016.
19. A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and Q. Z. Sheng, "IoT Middleware: A Survey on Issues and Enabling Technologies," *IEEE Internet of Things Journal*, vol. 4, pp. 1-20, 2017.
20. H. Derhamy, J. Eliasson, J. Delsing, and P. Priller, "A Survey of Commercial Frameworks for the Internet of Things," presented at the 20th Conference on Emerging Technologies & Factory Automation (ETFA), 2015.
21. (2017-11-22). *Ptolemy II*. Available: <https://ptolemy.eecs.berkeley.edu/ptolemyII/>
22. IBM Emerging Technologies. *Node-RED*. Available: <http://nodered.org/>

23. (2015, 2017-12-15). *Calvin - Lets Things Talk to Things*. Available: <https://www.github.com/EricssonResearch/calvin-base>
24. E. Latronico, E. A. Lee, M. Lohstroh, C. Shaver, A. Wasicek, and M. Weber, "A Vision of Swarmlets," *IEEE Internet Computing*, vol. 19, pp. 20-28, 2015.
25. P. Persson and O. Angelsmark, "Calvin – Merging Cloud and IoT," *Procedia Computer Science*, vol. 52, pp. 210-217, 2015.
26. M. Yannuzzi, R. Milito, R. Serral-Gracià, D. Montero, and M. Nemirovsky, "Key ingredients in an IoT recipe: Fog Computing, Cloud computing, and more Fog Computing," presented at the 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2014.
27. (2017-11-05). *Apache Hadoop*. Available: <http://hadoop.apache.org/>
28. (2017-12-18). *Apache Spark: Lightning-fast cluster computing*. Available: <http://spark.apache.org/>
29. (2017-11-29). *Apache Kafka: A distributed streaming platform*. Available: <https://kafka.apache.org/>
30. (2017-12-18). *NoSQL*. Available: <http://nosql-database.org/>
31. L. Xu, W. He, and S. Li, "Internet of Things in Industries: A Survey," *IEEE Transactions on Industrial Informatics*, vol. PP, pp. 1-11, 2014.
32. A. Botta, W. d. Donato, V. Persico, and A. Pescapé, "Integration of Cloud computing and Internet of Things: A survey," *Future Generation Computer Systems*, vol. 56, pp. 684-700, 2015.
33. X. F. Liu, M. R. Shahriar, A. Sunny, L. S. N., M. C., and L. Hu, "Cyber-physical manufacturing cloud: Architecture, virtualization, communication, and testbed," *Journal of Manufacturing Systems*, vol. 43, 2017.
34. M. Endrei, *Patterns: service-oriented architecture and web services* vol. 1st. Poughkeepsie, N.Y.: IBM Corp., International Technical Support, 2004.
35. R. Perrey and M. Lycett, "Service-Oriented Architecture," presented at the Applications and the Internet Workshops, 2003.
36. R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," Doctor of Philosophy, University of California, IRVINE, 2000.
37. IEC/ISO 20922. *Information technology -- Message Queuing Telemetry Transport (MQTT) v3.1.1*. Available: http://www.iso.org/iso/catalogue_detail.htm?csnumber=69466
38. Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," 2014.
39. "5G radio access – capabilities and technologies," Uen 284 23-3204, April 2016 2016.
40. "NGMN 5G Whitepaper," 2015.
41. I. F. Akyildiz, S. Nie, S.-C. Lin, and M. Chandrasekaran, "5G roadmap: 10 key enabling technologies," *Computer Networks*, vol. 106, pp. 17-48, 2016.
42. A. R. Hevner and S. Chatterjee, *Design research in information systems: theory and practice* vol. 22. London;New York;: Springer, 2010.
43. A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quarterly*, vol. 28, pp. 75-105, 2004.
44. M. Åkerman, Å. Fast-Berglund, M. Karlsson, and J. Stahre, "Introducing Customized ICT for Operators in Manufacturing," *Procedia CIRP*, vol. 41, pp. 490-495, 2016.
45. (2018-02-28). *play*. Available: <https://www.playframework.com/>
46. J. Schlehtendahl, M. Keinert, F. Kretschmer, A. Lechler, and A. Verl, "Making existing production systems Industry 4.0-ready," *Production Engineering*, vol. 9, pp. 143-148, 2014.

- 430 47. D. Mourtzis, E. Vlachou, and N. Milas, "Industrial Big Data as a Result of IoT Adoption in Manufacturing,"
431 *Procedia CIRP*, vol. 55, pp. 290-295, 2016.
- 432 48. O. Angelsmark and P. Persson, "Requirement-Based Deployment of Applications in Calvin," *Interoperability*
433 *and Open-Source Solutions for the Internet of Things: Second International Workshop*, vol. 10218, pp. 72-87, 2017.
- 434 49. (2017-11-21). *Unified Architecture*. Available: <https://opcfoundation.org/about/opc-technologies/opc-ua/>
435 50. (2017-08-30). *What is IO-Link?* Available: [http://www.io-link.com/en/Technology/what_is_IO-](http://www.io-link.com/en/Technology/what_is_IO-Link.php?thisID=76)
436 [Link.php?thisID=76](http://www.io-link.com/en/Technology/what_is_IO-Link.php?thisID=76)
- 437 51. R. Sumbaly, Jay Kreps, and S. Shah, "The big data ecosystem at linkedin," *Proceedings of the 2013 ACM*
438 *SIGMOD International Conference on Management of Data*, pp. 1125-1134, 2013.