Monitoring of Production Processes and the Condition of the Production Equipment through the Internet

Jari Halme, Erkki Jantunen, David Hästbacka, Csaba Hegedűs, Pal Varga, Mikael Björkbom, Heikki Mesiä, Rupesh More, Antti Jaatinen, Laurentiu Barna, Pasi Tuominen, Henri Pettinen, Marko Elo, Martin Larrañaga

Abstract—**The decreasing prices of monitoring equipment have vastly increased the opportunities to utilize local data, and data processing for wider global web-based monitoring purposes. The possible amount of data flowing though different levels can be huge. Now the question is how to handle this opportunity in both dynamic and secure way. The paper presents a new concept to manage data for monitoring through the Internet. The concept is based on the use of Arrowhead Framework (AF) and MIMOSA data model, and selected edge, and gateway devices together with cloud computing opportunities. The concept enables the flexible and secure orchestration of run-time data sources and the utilization of computational services for various process and condition monitoring needs.**

I. INTRODUCTION

A dramatic reduction of the prices of the devices capable of industrial data measurement, processing, and communication has led to a situation where data, and information derived from them can be made available globally through the existing networks in a way not seen before [1]. This has enabled the monitoring of the condition of e.g. remote but critical machinery and fleets of machinery e.g. for operation and maintenance management and benchmarking. The key is data management and data interoperability. The question is how to handle the vast amount of data and update the system dynamically based on the needs.

Although many systems can be easily connected to the Internet, there are several aspects that can limit this and steer how this should be done. First, there are applications in which Internet connectivity is not sufficient or the sheer amount of data is too big making it too costly or slow to transfer for cloud processing. Therefore, some processing can take place at the edge level, which further supports data privacy by not moving all the data to the open Internet although an open environment is often seen beneficial. In the production environments there are further requirements on real-time constraints and dependability for which the full Internet-based operation is not always feasible as such.

Nevertheless, the Internet-based connectivity of production systems is still desired to enable, for example, more flexible production system configuration and the use of data throughout the life-cycle of both the goods produced and the production assets themselves. Not all the Internet technologies are suitable as such, and better guarantees for the security, safety and dependability are needed in many production applications. With such domain requirements, a common architecture is needed with agreed practices how the system configuration can meet the set requirements, how the system components and services are bound, and how information is communicated in a secure manner. Moreover, the agreed service definitions, communication semantics and information models further facilitate the cornerstones of an ecosystem model in which more autonomous and flexible operations can take place [2].

In this paper, an architecture is described for the future digital industry to handle global data management and to implement the necessary data connectivity, security, privacy, and other services, for networked global business solutions. Two application areas in the field of lifting and rock-crushing businesses are presented setting the requirements and constraints for the development.

II. NEED FOR MONITORING

A. *Manufacturing monitoring*

In manufacturing, there are typically many steps to assemble a final product. From the lifting business point of view, this means in practice the hoisting of parts to the assembly station. The manufacturing process includes sub processes, each with a set of different devices from different manufacturers, each doing their own task. Previously, the sub processes and devices have

---

Jari Halme is with the VTT Technical Research Centre of Finland, Kemistintie 3, FI-02044 VTT, Finland (phone: +358 50 547 6542; email: jari.halme@vtt.fi).

Erkki Jantunen is with the VTT Technical Research Centre of Finland, Kemistintie 3, FI-02044 VTT, Finland (email: erkki.jantunen@vtt.fi).

David Hästbacka is with Tampere University, Korkeakoulunkatu 1, FI-33720 Tampere, Finland (email: david.hastbacka@tuni.fi).

Csaba Hegedűs is with AITIA International Inc., Budapest, Hungary, (email: hegeduscs@aitia.ai).

Pal Varga is with Budapest University of Technology and Economics, Budapest, Hungary (email: pvarga@tmit.bme.hu).

Mikael Björkbom is with Konecranes, Helsinki area, Finland (email: mikael.bjorkbom@konecranes.com).

Heikki Mesiä is with Konecranes, Helsinki area, Finland (email: heikki.mesia@konecranes.com).

Rupesh More is with Konecranes, Helsinki area, Finland (email: rupesh.more@konecranes.com).

Antti Jaatinen is with Metso Minerals Lokomonkatu 3, Tampere, Finland (email: antti.jaatinen@metso.com).

Laurentiu Barna is with Wapice, Yliopistonranta 5 FI-65200 Vaasa, Finland (email: laurentiu.barna@wapice.com).

Pasi Tuominen is with Wapice, Yliopistonranta 5 FI-65200 Vaasa, Finland (email: pasi.tuominen@wapice.com).

Henri Pettinen is with CrossControl, Finlaysoninkuja 19-23, Tampere, Finland (email: henri.pettinen@crosscontrol.com).

Marko Elo is with CrossControl, Finlaysoninkuja 19-23, Tampere, Finland (email: marko.elo@crosscontrol.com).

Martin Larrañaga is currently working with the VTT Technical Research Centre of Finland, Kemistintie 3, FI-02044 VTT, Finland (email: martin.larranaga@alumni.mondragon.edu). He is from Mondragon University, Arrasate, Spain.

worked independently, due to the lack of feasible technology to connect the disparate systems. The systems of systems and device abstraction (adapter) approaches, such as the Arrowhead service framework, have enabled the connection and the data gathering from all the devices.

For the monitoring of the system and subsystems in a manufacturing process, the state information and performance metrics need to be gathered. Each device type and make have different set of relevant information. The relevant information is determined by the monitoring needs and can be unique for each manufacturing process. The implementation of the relevant data collection solution can additionally be device specific. This implies that in manufacturing monitoring, the system should enable independent and exchangeable calculation and information communication, depending on the various needs.

For the manufacturing monitoring, it would be beneficial to be able to deploy and manage arbitrary algorithms in a diverse set of devices to enable the different manufacturing monitoring needs for the diverse manufacturing processes. The AF is used to enable the dynamic and loose coupling of monitoring algorithms and system monitoring. The monitoring algorithms are deployed to the different levels of the system: device, local site, and cloud, each doing their part in a systems-of-systems approach.

B. *Condition Monitoring*

Another use case is the vibrating screens; these devices are used to separate material according to the size. They are not very complex or expensive machines, but by nature they are self-destructing due to the large accelerations. The screens are not expected to last for decades, but unexpected failures will cause very large losses. Direct condition monitoring measurements can be made from the screen movement and acceleration levels; the bearings can also be monitored for excessive noise and heat. It is not yet known how well the failures can be predicted in advance, but there are many dangerous operating conditions that can be identified and warned about.

In the process equipment, some kind of condition monitoring can be considered a mandatory function. It is not always obvious what kind of condition monitoring functionality is required for a specific installation and misunderstandings are common. One parameter is the accuracy of detection that is required. A crude scale would be from "something is wrong" to the identifying of the part that has a problem. The other is the time scale; is it enough to detect that a failure has happened, or is the system expected to predict a failure in the future? These targets have a major effect on the engineering of the condition monitoring systems, and the human intuition can be misleading. For example, a predictive system using a digital twin model approach may need only a daily sample of the acceleration of a device, while just detecting that something has failed and preventing additional failure may require a millisecond-level response time and very simple sensors. Another important consideration is the networking of these systems. It is tempting to build systems based on transferring all data to "the cloud" for processing, but in practice this is not feasible, as the costs associated with the running of such data transfer, storage, and analysis become prohibitive for an equipment manufacturer. The issue could be different for a customer that can reap all the benefits from using the analysis results, but for a service provider that collects a subscription fee the equation is difficult.

The research goal for the condition monitoring is to simplify the architecture of the condition monitoring systems so that they would be similar on different levels, even when the underlying physical systems are different. The balance between local processing and cloud computing should be computed in some intelligent way, with adjustments made as the product matures. Usually the calculations can be moved nearer to the sensors and even measurement details can be adjusted when the knowledge of the measured phenomena increases.

C. *Summarising the case needs*

In order to enhance data/information processing and communication between the existing devices in the dynamically changing use case environments, the future digital framework should allow for the independent, dynamic interaction in a balanced way at both device, local site, and cloud levels. The framework should provide both interoperability and integrability to support collaborative procedures when things change, e.g. new devices and services are entering, and others are leaving.

In the next sections, the utilization of AF and Mimosa[1] data model for OSA-CBM (Open System Architecture for Condition-Based Maintenance) are discussed more in detail as a possible option to fulfil the identified requirements.

III. ARCHITECTURE BASED ON ARROWHEAD FRAMEWORK

The AF has been developed as an architectural service platform option for the future digital industry. It is an open source framework for service-oriented systems that also take into account features required in the cyber-physical systems such as the QoS (Quality of Service), security, and Internet connectivity. There are naturally various other frameworks available for the Internet of Things (IoT) [3]. However, AF can take care of the need of various stakeholders dynamically, and its architecture can handle the specifics of the system of systems. The framework is under development in the ECSEL project Productive4.0 in which also the previously mentioned use cases are developed.

A. *Arrowhead Framework*

The AF[2] has three fundamental concepts that are aiding to digital industry applications. Since the automation and industrial environments are always local and usually closed from direct access, the appropriate security measures must be kept while digitalizing as well. This invokes the definition of Local Clouds that are at least network-wide but usually physically segregated networks too.

---

[1] MIMOSA is a not-for-profit industry trade association developing and encouraging adoption of open physical asset lifecycle management http://www.mimosa.org/2017/07/26/what-is-mimosa/

[2] The arrowhead framework wiki: https://forge.soa4d.org/plugins/mediawiki/wiki/arrowhead-f/index.php/Main_Page

Secondly, since the automation and industrial IoT face serious interoperability and integrability issues on the design level, the AF relies on the service-oriented thinking (Service Oriented Architectures, SOA) [4] to aid and guide the developments of the automation and industrial IoT systems. This, in short, defines the services and service consumption as an abstraction of the Internet Protocol (IP) based network communication for a specific purpose. In here, there are services as such, and there can be multiple service providers and service consumers communicating based on the definition (contract) of the given service (Figure 1).

Moreover, a given communication link may not be established by the hardwiring of certain resource instances together, but rather it relies on the loose coupling and late binding design patterns of the SOA. This is, in short, giving the IoT systems the dynamic re-configurability to be able to interconnect systems on demand in runtime.

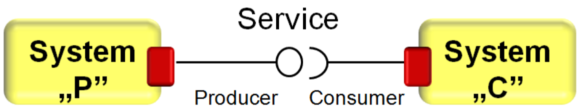

Figure 1. A consumer system consuming a service from a provider system

Finally, these automation and industrial local clouds must be governed, so that the involved systems (both cyber and physical) and their operations are corresponding to the general targets of the larger ecosystem, i.e. the system of systems. Arrowhead solves this by introducing the core systems that realize governance through implementing central resources to facilitate the loose coupling and late (runtime) bindings between the application systems (Figure 2).

Within the AF the application systems utilize the mandatory core systems (Service Registry, Orchestrator and Authorization System) to make the interconnection with other systems in the desired way. Further details on the core interactions with the AF modules can be found in [5, 6].

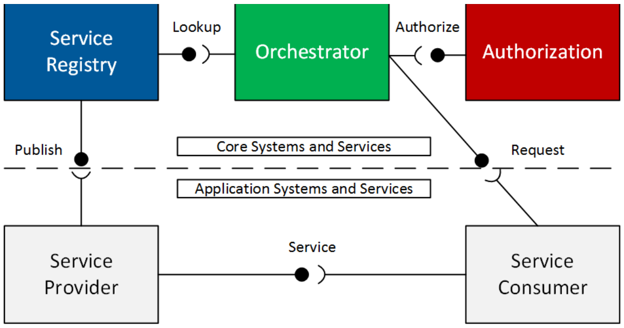

Figure 2. Arrowhead core interactions

With these governing, mandatory core systems and additional supporting core system, the AF further aids the SOA-based IoT system of systems by providing a secure and convenient way of interconnecting systems even from different Local Clouds [7, 8]. This capability is essentially an Arrowhead-managed secure IP tunnelling service between networks residing behind firewalls, Network Address Translation (NAT) systems, just like almost all industrial IP networks.

B. *Production monitoring and analytics concept*

The production assets need to be IoT devices or equipped with data acquisition devices. The following figure (Figure 3) describes the conceptual architecture of how the device data flows via a gateway further to the cloud. Parallel to the gateway there is another pipeline in the concept for providing the development data independent of the actual gateway. This parallel pipeline can be used to gather data that is used to develop analytics models. These analytics models will then be deployed both to the gateway and cloud. In a production setting the data acquisition pipeline could be embedded into the gateway itself.

The analytics models are deployed as services and during run-time there is a need to orchestrate their use as well as manage the access rights. The general idea is, depending on the need, to dynamically orchestrate the flow of data into the analytics services executed at different levels.

The monitoring analytic concept architecture is strengthened by the concept of MIMOSA, which is a data model definition that allows data models that are easy to fit into the relational object and information management models required by Condition-based Maintenance (CBM) [9]. The MIMOSA definitionss are compliant with and form the informative reference to the published ISO 13374-1 standard for machinery diagnostic systems [10].

C. *Utilizing Arrowhead*

Besides providing a service-oriented approach with all its benefits, utilizing the AF in this scenario has further advantages. First, the local cloud approach enables to apply different security methods at the required levels for communication within the local cloud and for the inter-cloud scenarios. Second, Arrowhead provides the Service Registry, Orchestration and Authorization services by default, hence the dynamic service orchestration is given. This is very beneficial for cases when the elements of the ecosystem change dynamically, when a new system is introduced, or when a redundant system takes over the service providing role from a failing one. Third, the AF provides additional services, such as the Event Handling, QoS monitoring, or Inter-Cloud servicing, which eases the development and integration efforts during deployment.

The mandatory core services of the local Arrowhead cloud can be hosted by any device that is able to guarantee the required processing resources. While even a Raspberry Pi can be such a host, in the current scenario these Arrowhead core services are hosted on a server machine that is part of the scenario anyway, hosting the data handling functionalities, among others. The illustration of the local cloud scenario supported by edge devices is shown in Figure 4.

IV. TECHNICAL ARCHITECTURE AND IMPLEMENTATION

The conceptual architecture outlined in the previous section is implemented based on the latest Arrowhead version (4.0) using the existing hardware and software components.

A. *Gateway software stack*

In the use cases, the device architecture consisted of two Raspberry Pi 3 (one edge device with a GrovePi+ board and one getaway). The communication between the two Raspberry Pi´s is wireless. Optionally, other edge devices can be connected based on the needs e.g. for the vibration and load data.

Node-RED (Flow-based programming for the IoT) is working at the edge level. The MIMOSA data model is used at the edge, gateway and cloud levels. Moreover, the AF is working in the gateway level (Figure 4). The Node-RED makes it easy to read digital and analog (10 bits through the GrovePi+ board) signals. If the Node-RED detects that there has been a change in any of the digital inputs new asset_events and asset_event_num_data are created for each digital signal, and a new meas_event with mevent_num_data is created for each analogue signal in the local MIMOSA database.

Once the values are saved in the local MIMOSA database, the Node-RED saves the values also in the database of the gateway Raspberry Pi. Once the data is in the MIMOSA database of the Gateway device the AF has been used to securely storage new data in the MIMOSA database in the Azure cloud. The provider service provides the new data of the MIMOSA database to the AF. Then the consumer service consumes this data and stores it in the MIMOSA database of the Azure cloud using a REST interface (Representational State Transfer).

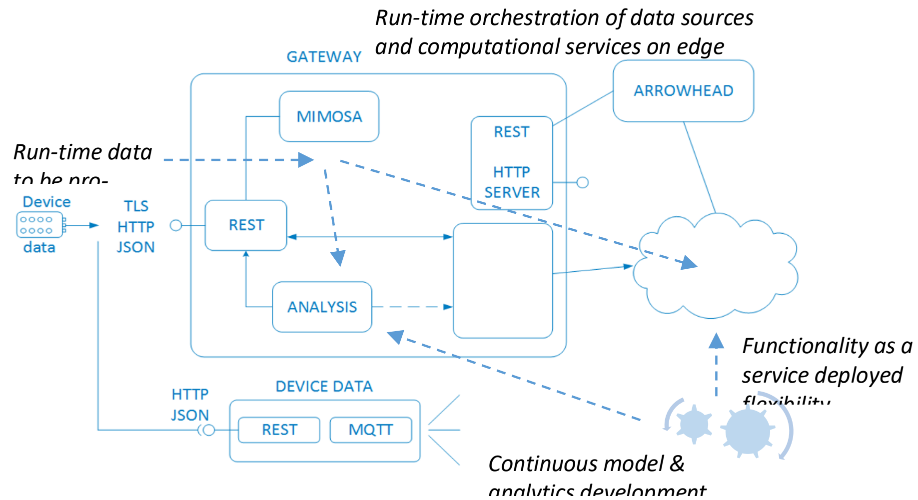

Figure 3. Overview of the Arrowhead-compliant Process- and Condition-Monitoring Scenario

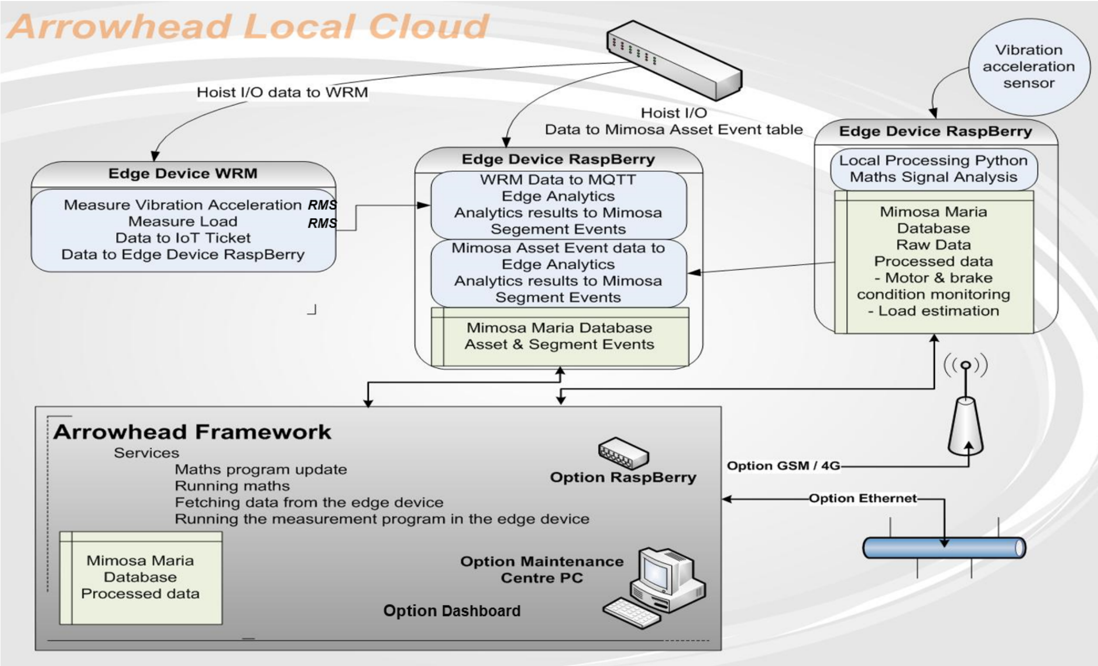

Figure 4. Arrowhead local cloud embodiment for the hoist operation monitoring

B. *IoT Ticket*

IoT-Ticket (iot-ticket.com) is an Internet of Things (IoT) platform and ecosystem for data acquisition, visualization, and storage, asset control and monitoring, data analytics and advanced reporting. Adopting a large collection of relevant industrial standards and communication protocols, the IoT-Ticket integrates with other systems and devices (see Figure 5). The platform offers the following tools and services:

- IoT-Ticket Server – the core component handling the configuration and deployment of functionality throughout the system with analytics capabilities.
- IoT-Ticket Dashboard – the Interface Designer and Dataflow Editor allowing to interact with the remote devices and to get status updates from the enterprise level to sites, assets, and data nodes.
- IoT-Ticket Security Gateway - establishing connections to remote sites through the certificate-based VPN tunnels and allowing to define user specific access to components and functionality.
- IoT-Ticket Report Viewer and Editor - for designing, creating, and exporting reports with dynamically generated content.
- WRM® 247+ - robust and customizable IoT device for remote management, measurement, and control with wired and wireless connectivity, digital and analog I/O channels, and built-in sensors.

Within the scope of Productive4.0 the IoT-Ticket platform and WRM® 247+ devices are integrated locally and at cloud level with the Arrowhead (4.0) platform to demonstrate and test its feasibility to the edge (Figure 4) and cloud computing, for production monitoring and control. For the chosen industrial demonstrator use case, WRM® 247+ devices will become autonomous in terms of data services offered by automatically registering themselves to the AF and providing data to IoT-Ticket servers and the local REST APIs (Application Programming Interface) services discovered from the framework. Within the same use case, the IoT-Ticket will offer analytics services to the framework and will allow to dynamically and automatically monitor, register and unregister services to the Arrowhead platform from the cloud data (e.g. IoT-Ticket data nodes).

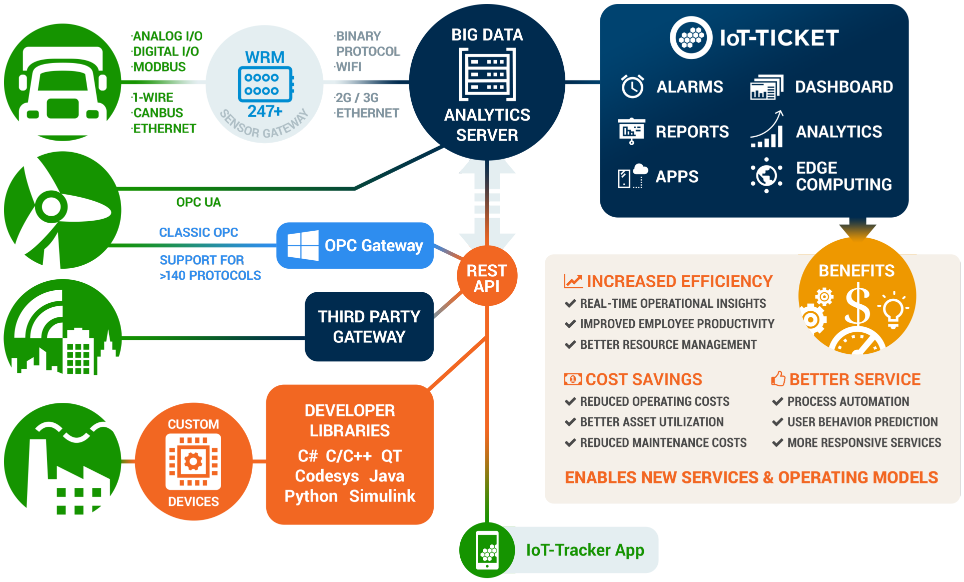

Figure 5. IoT Ticket cloud

C. *Solution as Gateway and Local HMI*

CrossControl's display computers combine processing power and displays in one package e.g. controls, video monitoring, and wireless communication can be performed by a single on-board display computer. Open architecture and open interfaces foster scalability and enable application creation by any party.

In Productive4.0 use cases for process and condition monitoring, the on-board display can serve as a gateway device (Figure 4), which gathers data from the various edge devices, processes the data, and dispatches it to the offsite systems. Additionally, the display can run as a local dashboard, visualizing selected telemetry and state values of the monitored machine and its work cycle phase. In both cases the display can be implemented as Arrowhead compliant so it can act as a local Arrowhead cloud, making it flexible e.g. to add new edge nodes into the corresponding environment.

In these targeted use cases, the device architecture consists of an onboard rugged display computer (CCpilot VS) and a communication module (CrossLink AI)The communication module supports a selection of wireless technologies, enabling connectivity with on-site peers and enterprise systems. The SDK and runtime library, called as LinX (Figure 6), provides the tools and protocols for applications operating in on-board data collection, monitoring, visualization, logging, and exchange with the cloud.

V. BENEFITS

It is expected that the number of data processing and communication points will be increasing rapidly. This IoT boom means also that the number of possible weak points is increasing. Thus, authentication, authorization and high cyber security is relevant through the whole system of the interconnected data points. The AF provides tools and assurance services for the information security of multiparty environments against cyberattacks, and prevents unauthorized use of digital data.

In order to generate relevant advisories for phenomena that are planned to be diagnosed/predicted in the production processes and equipment, high data reliability and the accuracy of the analytic results is needed. The presented architecture corresponds to a Mantis reference model [11], and it enables appropriate processing for monitoring at different levels (edge, gateway, cloud) starting close to the data source and then carrying out refinement at the other levels. It is possible to apply accurate, high speed, and real-time processing capability at the local level and modular data refinement processes with extracted features combining data/information from the relevant other sources at the other levels to increase the analysis reliability. It also helps with the balancing between the local and cloud computing.

Linking different components together and making use of data from multilevel sources improve the building up and orchestrating of the new dynamically evolving systems. By following the approach presented in the paper, specification for the application services carrying information vital for the processes can be done in a unified way through the core systems and services. This empowers development of new services and business opportunities. The ecosystem of similar architectures can be linked both at the local and cloud levels. The created interoperability makes it possible to expand and multiply the existing systems when needed by keeping the common orchestration manageable.

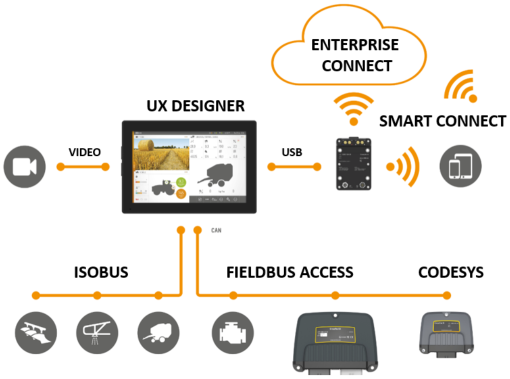

Figure 6. LinX application platform

VI. CONCLUSION

In this paper, a new concept is presented to manage the communication and data to information processing requirements at the factory floor for the future digital industry needs. The core concept is based on the utilization of the AF and MIMOSA data model. From the attached local edge devices, data is processed through the gateway to the cloud in a flexible way. The main process related data analytics is done at the gateway level where the application system services (provider & consumer) are published and requested through the AF mandatory core systems (Service registry, Orchestration and Authorization). The MIMOSA data model is used at both the edge, gateway, and cloud levels, and it is interacting with the data, analytics and services. The MIMOSA data model is also suitable for the process event management. The presented concept enables the secure and flexible orchestration of the run-time data sources and the utilization of computational services. It is also a valid scenario for the various process and condition monitoring needs and is suitable for the continuous analytics and model development in the constantly evolving operational environments on the factory floor.

ACKNOWLEDGMENT

Research leading to these results has received funding from the EU ECSEL Joint Undertaking under grant agreement n◦ 737459 (project Productive4.0) and from the partners' national programmes/funding authorities (Business Finland).

REFERENCES

[1] E. Jantunen, G. Di Orio, F. Larrinaga, M. Becker, M. Albano & P. Malo, "A Framework for Maintenance 4.0", 14 Jun 2018, Proceedings of the 10th IMA International Conference on Modelling in Industrial Maintenance and Reliability. Manchester, UK: Institute of Mathematics IMA
[2] D. Hästbacka, E. Jantunen, M. Karaila, & L. Barna, "Service-based condition monitoring for cloud-enabled maintenance operations", 22 Dec 2016, Industrial Electronics Society IECON 2016 - 42nd Annual Conference of the IEEE Institute of Electrical and Electronic Engineers IEEE, p. 5289-5295
[3] H. Derhamy, J. Eliasson, J. Delsing, P. Priller, A survey of commercial frameworks for the Internet of things. In:IEEE International Conference on Emerging Technologies and Factory Automation(EFTA), 2015.
[4] M. Bell, Introduction to Service-Oriented Modeling. Wiley and Sons, 2008.
[5] C. Hegedűs, D. Kozma, G. Soós, and P. Varga, "Enhancements of the arrowhead framework to refine inter-cloud service interactions", in Conference: The 42nd Annual Conference of IEEE Industrial Electronics Society (IECON), 2016.
[6] F. Blomstedt, L. L. Ferreira, M. Klisics, C. Chrysoulas, I. M. de Soria, B. Morin, A. Zabasta, J. Eliasson, M. Johansson, and P. Varga, "The arrowhead approach for SOA application development and documentation", in IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society, 2014.
[7] P. Varga and C. Hegedűs, "Service interaction through gateways for inter-cloud collaboration within the arrowhead framework", in 5th International Conference on Wireless Communications, Vehicular Technology, Information Theory, Aerospace and Electronic Systems (VITAE), 2015
[8] C. Hegedűs, P. Varga, and A. Frankó, "Secure and trusted inter-cloud communications in the arrowhead framework", in 2018 IEEE Industrial Cyber-Physical Systems (ICPS), 2018.
[9] E. Jantunen, G. di Orio, C. Hegedus, P. Varga, I. Moldovan, F. Larrinaga, M. Becker, M.Albano, P. Maló, "Maintenance 4.0 World of Integrated Information", in Interoperability for Enterprise Systems and Applications, I-ESA'18, Berlin, Germany, 2018.
[10] "ISO 13374-1:2003 condition monitoring and diagnostics of machines – data processing, communication and presentation – part 1: General guidelines", 2003.
[11] C. Hegedus, P. Varga, and I. Moldován. "The MANTIS architecture for proactive maintenance", 2018 5th International Conference on Control, Decision and Information Technologies (CoDIT). IEEE, 2018.