

Service-oriented Architecture and Cloud Manufacturing

Tomáš Lojka, Marek Bundzel, Iveta Zolotová

Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9/B, 042 00 Košice, Slovakia, e-mails: {tomas.lojka, marek.bundzel, iveta.zolotova}@tuke.sk

Abstract: This paper deals with service oriented architecture (SOA) and cloud technologies in the industry. SOA and cloud technologies have considerable potential to improve industrial production and control. This paper analyzes those systems and proposes an architecture for device integration, data acquisition, data processing and remote control. We designed and implemented SOA for industrial technological routers, allowing device integration in the designed architecture and cloud interoperability. We used a private cloud and designed atomic and composite services. These services empower the architecture with data acquisition, processing and data analytics. The architecture offers a solution for connecting to an industrial network with a remote HMI client and a cloud MES application. We used greenhouse and assembly lines models, a PLC, an eWON, an eFive, a private cloud (Windows Azure Pack), Android, Windows tablet applications, smartphones and desktop computers. We designed and implemented a multi-tenant cloud architecture. In our experiments, we verified our architecture in two simulated plant divisions. Each of them has their own implementation. Our architecture offers better interactivity with suppliers, cooperation with internal systems and can be easily adapted to other extensions of the processes of a plant.

Keywords: Industrial cloud; IoT; SCADA; SOA; MES, Windows Azure

1 Introduction

Control systems dynamically improve and attempt to meet industrial requirements, such as the need for real-time online analysis, realistic representation of industrial processes or obtaining accurate information to control the industrial processes. These requirements raise questions on how to improve the following:

- Integration of devices and data in a plant.
- Implementation of real-time data acquisition and analysis,
- Highly available data.

Flexibility and integration appear to be the main issues. For example, a large number of interconnected monitoring and data processing appliances can cause flexibility and integration problems. For this reason, it is necessary to design an architecture that will solve these problems. Nowadays, the trend for information and control systems (ICS) is to become more agile, heterogeneous, interoperable, open and dynamic [1], [2], [3], [4]. A design and an implementation that ensure the stability of the systems and their functionality in real time appear to be a robust solution. Such a solution should be stable during the current expansion of data flows and, during the expansion of the infrastructure of an entire plant and mainly during the expansion of the number of interconnected devices in it. Service oriented architecture (SOA), the Cloud and the Internet of Things (IoT) can meet the industrial requirements. SOA extends the solution with atomic, device, composite and deployment services. These services support cooperation, offer agility, and operate in a heterogeneous environment [5]. The IoT supports interconnection and the Cloud offers stability and integration. The Cloud can host the representation of the environment where the services are hosted.

In this paper, we focus on the design, implementation and verification of our future solution of an industrial architecture that satisfies the agility, stability and interoperability criteria and works in heterogeneous environments. The architecture consists of Cyber-physical systems, IoT, SOA and the Cloud.

This paper is organized into three main chapters. The first chapter describes the increase in the Supervisory Control and Data Acquisition (SCADA) and Manufacturing Execution System (MES) performance and deals with the SOA. The second chapter deals with the concept definition of the future cloud SCADA and MES architectures. The last chapter presents an evaluation of two use cases.

1.1 Increasing the SCADA and MES Performance

Nowadays, SCADA and MES are parts of the ISA-95 standard. Many industrial information and control systems are based on this standard. The standard has five levels and each level only communicates with the adjacent levels (the pyramid in Figure 1). Level 0 represents physical production processes. Sensors and actuators are a part of the next level that is responsible for monitoring and affecting physical processes. The Control/Interlocking level controls physical processes and it is also labeled as Level 1. The SCADA level is responsible for data acquisition and supervision control. SCADA is also labeled as a system for a remote control and monitoring. It collects data and stores them into databases, offers viewing the current and historical trends, triggers alarms and does statistical data processing. MES is the next layer in the standard and controls the workflow to reach the desired product. It operates with the data stored in the databases and helps make decisions, understand the current situation in the production processes and manage the transformation of materials to the final product. The top layer is Enterprise resource planning (ERP). It covers business activities and deals with an enterprise organization [4].

New IT technologies lead to new approaches in SCADA systems and MES. Cloud computing and services emerged in the last few years. The Cloud offers enough resources and offers robust services for aggregating, processing and analyzing data. It is accessible for collecting and offering data or services anytime and anywhere. It also improves SCADA and MES and other layers with features described by NIST (National Institute of Standards and Technology) [1], [6], [7].

There is some research and implementations of SCADA and MES in the Cloud that use SOA. For example, WebSCADA offers solutions that reallocates SCADA into the Cloud and uses SOA [8]. The ICM-AESOP is a project that reallocates even more than WebSCADA does [4]. It reallocates control, SCADA, MES and ERP into the Cloud and benefits from SOA [4]. Project SOCRADES uses SOA in an embedded device. In general, these solutions and projects focus on SOA-based embedded devices and the reallocation of some ISA-95 levels into the Cloud. The missing part here is a gateway that implements SOA into the existing process level, aggregates data and creates universal access to the Cloud. The SOA gateway can then create a consistent system by offering atomic services for composite services in cloud modules.

With the increasing number of devices, the amount of data grows. This causes problems in human-machine interaction, SCADA and MES. One of the approaches is Machine-to-Machine (M2M) communication, which reduces the human-machine interaction [3], [9]. Another approach is data pre-processing [10]. However, SCADA is supposed to work with larger amounts of data that increase quickly. Nowadays, thousands of industrial devices and appliances can use network communication to transmit data. Therefore, SCADA should change the way of processing such an amount of distributed data and information [1], [2]. MES has the information about the current situation on the plant floor and helps optimize processes on it.

1.1.1 Flat and Information-based Architecture

SCADA, MES and basically all ICS seem to be headed towards using SOA services in the future. Therefore, it is important to find the optimal distribution of functionalities between the users, the computing systems and the Cloud solutions. In the next generation of ICS (some functionalities are relocated to a cloud-based side. The traditional ISA-95 standard is oriented to a hierarchical model, whereas the next generation of industrial applications is oriented to use the services exposed by cyber-physical systems or cloud services (Figure 1). The architecture changes into a flat, information-based architecture composed of services. These services are represented as modules, which improves the flexibility of information-based architectures. These changes were defined in the ICM-AESOP project [4].

1.1.2 Services in the Information-based Architecture

The services can divide the architecture into domains with a better interaction and modularity. Another important aspect of ICS based on SOA is robustness and the reinforcement and automation of industrial activities [3].

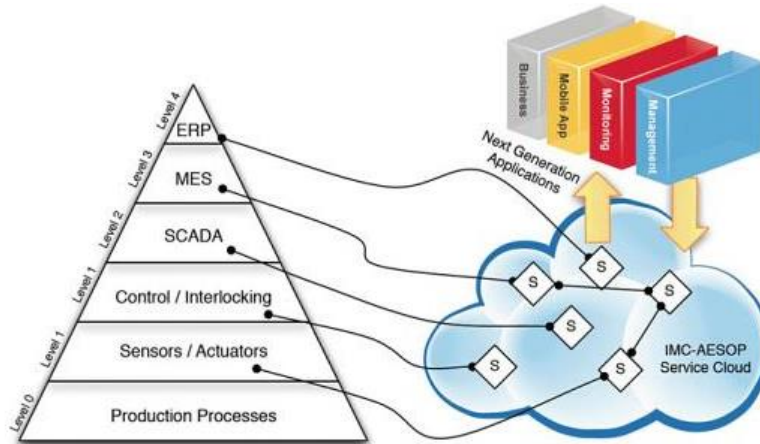


Figure 1

The evolution of ICS - Traditional ISA-95 standard with a flat information-based architecture [4]

We focus on the application of SOA and cloud technologies in ICS, particularly in SCADA and MES parts of ICS. The services increase the functionality and interoperability with devices [11], [12]. We designed, implemented and tested our architecture which makes use of SOA, an industrial gateway/mediator and the Cloud. We created our architecture for service-oriented SCADA systems and MES. This service-based system will support horizontal and vertical connections, such as communication between enterprise systems, plant-floor devices or controllers. This communication is an important part of the Industry 4.0 key paradigms: smart products, smart machines, smart planners, and smart operators [6].

1.2 Service-oriented Architecture in SCADA and MES

SOA enables to improve and automate plant functionalities, and to design, implement and aggregate services. A service represents a primal functional side of the implemented SOA [13]. The plant uses a set of services that communicate with clients or with other services [14].

The reason for using SOA in a plant reflects the base features of SOA [15]:

1. Autonomy
2. Interoperability
3. Weak dependencies

4. Independent implementation
5. Flexibility

A SOA-based industrial solution may consist of components like SCADA, Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), Programmable logic controller (PLC), Distributed control system (DCS), MES, devices, and cloud service integration (Figure 2) [6], [16], [19]. These components represent atomic or composite services which can interact with each other. Devices communicate with highly accessible resources. The services can be implemented in the Cloud. These services enable data analysis and encourage making decisions over larger amounts of data and offer the control and visualization of the data to clients.

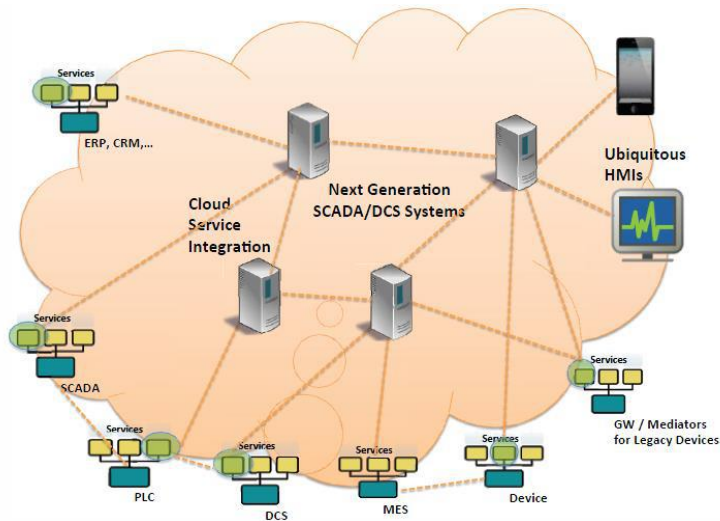


Figure 2

The architecture for SOA in the industry. ERP, CRM, SCADA, PLC, etc. with implemented services. The services communicate with the cloud Service Integration. The Cloud enables communication with HMI clients [3]

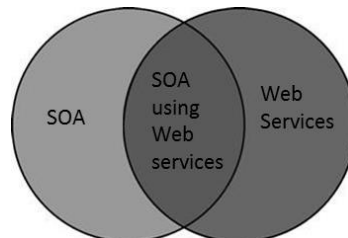


Figure 3

The relation of SOA and Web services [3]

Communication is required in almost all cases. The services are mainly web-based [13]. The relation between SOA and Web Services is described in Figure 3.

Web services enable interactivity among the processes and the machines in the plant. The communication transmits distributed data and information between the plant devices, appliances or software via the web services [18], [19].

SOA is one of the main focus points of this paper because of its autonomy, interoperability, easy integration and the flexibility of configuration. The services may be used in various ways, thus increasing their reusability. There are two types of services: atomic and composite. Atomic services are well defined and independent of the state of other services. Composite services are a set of atomic or other composite services. These depend on the status of other services comprising the service in question. One of the best service implementations is via web services, which have huge potential in remote control. In our work, we implemented the web services on a traditional, easily programmable eWON industrial router, thus extending its basic-features. We satisfied the interoperability criteria with atomic and composite services.

In other words, the services represent a solution for interactive communication between machines or systems. The following list briefly describes the reasons why service-based SCADA systems and MES can increase the efficiency of manufacturing processes:

- **Cooperation** with mobile devices, sensors and actuators, smartphones and tablets.
- Easy **interoperability** with devices, machines or systems with special human interaction, for example using an intuitive **multi-touch control** on a touch screen tablet or a smartphone.
- **Open integration** with devices, and cloud and third-party applications.

The services have additional features such as security, discovery of other services, data encryption, data management and communication reduction tools. Generally, the services in SCADA and MES can be grouped based on what they do: alarms, control, data management, discovery, Human-Machine Interaction (HMI), life cycle management, diagnostic, safety and process monitoring, resource management, scheduling and performance analysis [4]. However, we mainly focus on data acquisition, alarms, remote control, visualization and performance analysis in this paper.

According to the mentioned problems, we aim to develop a SCADA solution, addressing:

- The use of standards for integration with industrial devices from different vendors and satisfying M2M communication.
- The increase of performance of SCADA systems.

- The implementation of security in SCADA systems.
- The Cloud and SCADA interoperation.

MES provide correct information about the current situation on the plant floor. On top of that, they also offer information on optimization. In this paper, we focus on MES, too. However, we deal with a developed cloud service representation of simple MES functionalities like Overall Equipment Effectiveness (OEE).

2 Concept Definition of the Future Cloud SCADA and MES Architecture

Our concept was developed in two steps and consists of SOA, industrial devices and a private cloud. In the first step, we identified and designed services before designing the architecture. We designed atomic services that:

- Help create composite services.
- Allow for easy maintenance.
- Possess the SCADA functionality with better interaction and robustness.
- Simplify the expansion of the system.

Our concept features services with two types of implementation. The services can be implemented directly on the plant-floor devices and the gateway/mediator, or in the Cloud. According to [4], we specified services and groups for our concept (Table 1). There are many more services described in [4] but we did not use all of them.

Table 1
Services in architecture's concept definition

Service Group	Service role	Description
Alarm	Alarm processing, Alarm configuration	Processes the alarm and creates alarm events for the HMI, and the Control and data management Service groups.
Control	Control	Executes control based on input data and produces control values.
Data management	Data acquisition Data output Logging	Transfers data between sensors, actuators and databases.
Data processing	Filtering/Normalizing Data analytics Event management	Data preprocessing and processing; creates events for the Alarm, Control and Data management groups.
HMI	Human-Machine interaction	Creates a HMI interface for PC and mobile service clients.

Model	Model Model management	Information model for control or optimization of processes.
OEE	Overall Equipment Effectiveness	Counting OEE for a selected production line.

In the next step, we designed an architecture that can improve data acquisition, alarms, remote control and visualization and we also designed a general representation of the architecture (Figure 4). The concept definition of the architecture consists of:

- Cyber-physical systems – representing the technological layer of a plant.
- Internet of Things – representing the interconnection between the Cloud and the Cyber-physical systems.
- The Cloud – representing composite and atomic services, implementing communication, database, SCADA, MES, ERP and Business intelligence services.
- Clients – representing the devices that use the cloud services in the Human-Machine Interaction (HMI).

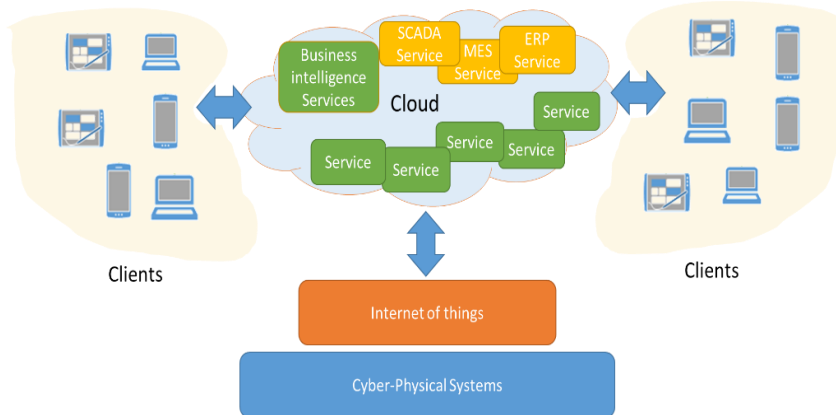


Figure 4

The concept definition of the proposed architecture

We defined the hardware and software resources needed to implement our concept definition of the architecture. The concept consists of a technological layer implemented in the laboratory, a CompactLogix PLC from Allen-Bradley, an eWON industrial router, an eFive industrial firewall, mobile devices, Windows Azure and Google servers. We used a virtual production line model and a greenhouse model to simulate the technological layer. The communication, interoperability and security with Cyber-physical systems are based on a programmable eWON industrial router and an eFive. We extended the eWON with SOA and used the eFive industrial firewall to create a modular and interoperable SOA environment secured by a VPN server. We chose the Windows

Azure cloud platform for the cloud implementation, where we designed and developed the services for connectivity, web services, HMI, simple MES, control, data processing and management, alarms and models.

2.1 Designing an Architecture for SCADA and MES

The architecture is based on SOA and the Cloud. Therefore, we used services described in Table 1. The architecture has the cooperation, interoperability and open integration features from subchapter 1.2. According to the defined concept (Figure 4), we focused on data acquisition, alarms, remote control and visualization. Therefore, we used the Cloud, SOA, the IoT idea and a client application in the design process of the architecture for SCADA and MES. In the proposed architecture (Figure 6), the PLC contains the control logic for assembly lines and greenhouse models. Mobile devices enable remote control and visualization of the technological process. A remote secure connection is made via the industrial VPN server. The mobile clients are smart phones, computers and tablets. The gateway/mediator we used has our own implementation of SOA described in Table 1. The architecture consists of the private cloud and a connection to a public cloud server. The main parts are the gateway/mediator and the private cloud. The implementation of the proposed architecture is in Figure 5.

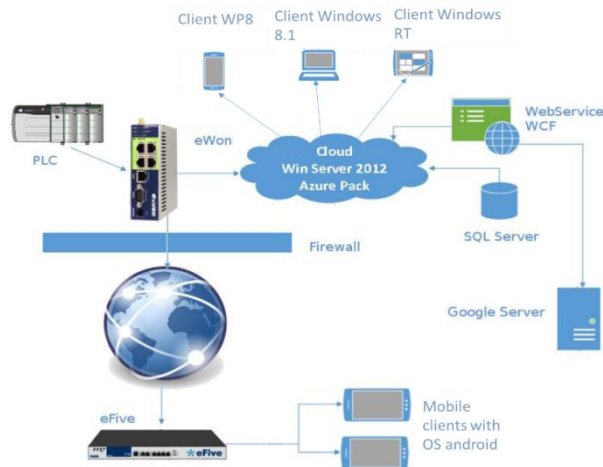


Figure 5

Our designed architecture that consists of the technological layer of ICS, an industrial router eWON, Windows Azure as a cloud solution, a SOA service, an SQL database, a Google server, and mobile Android and Windows clients

2.2 Gateway/Mediator Services in the Designed Architecture

The industrial router does not offer any services in its standard implementation. Our intention was to build the means for a flexible interaction with various industrial devices from various vendors such as Rockwell, Siemens, Omron and others based on SOA. We built a software communication interface with a standardized protocol allowing every device in the factory to be a part of SOA and to be connected to the Cloud (Figure 2). This communication interface can be deployed on devices and it is based on a standardized protocol that is normally used in the Cloud, but not in industrial devices. The interface also improves the connectivity with various clients. Protocols such as HTTP are supported in many development frameworks and even in the most basic frameworks. HTTPS can be used as well, but we did not use it in our experiments because the router has a basic access authentication and we used a VPN tunnel for the connectivity. Another point of our concept definition was to define the format of the messages. The messages should be easily and quickly serializable and should be able to be represented as objects. Suitable formats include XML, JSON, YAML or a new format implementation. XML, JSON and YAML are standardized and can easily be parsed into objects. We decided to use JSON, because of its common popularity in the development community.

We defined a JSON communication protocol and a message format. Then we created a hash table for ID and request recording. We designed and implemented the following services for the hash table:

- Connection test - a simple echo from the service.
- Getting the value of a tag - returns a tag with the structure and the values from the addressed PLC or returns tags directly from the router.
- Setting the value of a tag – sets a value of the addressed tag.
- ACK of alarm – acknowledgement of the alarms.
- Getting the complete state information and the error bits collection.
- Getting the historical data from a specified interval.
- Getting all alarms values.
- Refresh of all tags in the defined tag collection.
- Getting a collection of values from the defined collection.

Every service replies with a response message containing the data for the client, and acknowledgement information or execution errors.

In conclusion, we designed gateway/mediator services and cloud services. The services are atomic and composite. We implemented the services with SCADA, HMI and MES functionalities, but we intend to consider extending them in the future.

2.3 Cloud Services in the Designed Architecture

We created atomic and composite services in a private Windows Azure cloud. The services save data into databases, communicate with our mediator/gateway services and communicate with the Google calendar service. The services can communicate with the gateway/mediator, directly with the PLC, with applications on mobile devices, with desktop application too. We also created atomic MES services and HMI services.

We developed the services in Microsoft Visual Studio and deployed them to the private Cloud. We used a MSSQL service to store the data from the technological layer and developed an OEE service working with the MSSQL service.

Based on Table 1, we implemented the Alarm Control, Data management and processing, HMI and OEE services. The private Cloud is important to form a robust architecture for ICS capable of implementing a commercial ICS solution but also a private or an open ICS solution.

Real-time communication is an important issue in the context of cloud computing [20]. There are approaches using predictive and deterministic behavior of devices that achieve real-time monitoring [4]. Real-time data processing and acquisition are important for control and we addressed this too. Based on our previous research in online data processing, we created specific events that reduce the load on the network and reduce the power consumption of the devices on the network [19]. Event processing will empower future architectures, too [4].

Data are originated in the technological layer or they are produced by the clients in our solution. Data exchange is done by an event-based communication. This way, only important data are sent to the Cloud, thus decreasing the amount of communication calls, communication cost and increasing the bandwidth of the network. We select the relevant information during data processing and information discovery from the raw sensor data. The discovered information is then classified based on the communication topic. According to the topic, an event is created containing factual information, the priority and statistical information derived from the sensor data. We used Tsallis entropy to process statistical information. The results of the statistical process are then classified into topic and event priority classes. New classification results trigger events. This processing is implemented on technological devices such as our gateway/mediator where we implemented SOA.

The cloud connectivity is based on services while a service is a client for other services. This helps create a modular system consisting of atomic and composite services. The device algorithm runs in the background, monitors data and decides when to create events. Event handlers process data and decrease the communication cost thanks to the used algorithm which reduces the number of messages with insufficient information value by 49. We created two cases to demonstrate the Cloud connectivity. In the first case, the Cloud is directly

connected to the PLC via the cloud service. In the second case, the Cloud is connected to the PLC via our SOA solution implemented on the gateway/mediator.

2.4 Description of the Devices and Technologies Used

The most important technological parts are the eWON and Microsoft Azure.

An **eWON** router is a specialized programmable industrial router containing integrated protocols for communication with PLCs from Rockwell, Schneider, Siemens, OMRON, etc. This router can also be connected to the eFive VPN server and can use the public connectivity service Talk2M hosted in the Cloud for remote management. We chose eWON because it is programable and offers connectors to industrial networks. It is a suitable hardware device for our designed architecture, because it is built for the industry. However, we created our own private remote management solution without using Talk2M. The router has other features like HTML visualization, sending alarms via SMS messages and e-mails, monitoring and controlling production processes, storing historical values and defined tags as well as setting thresholds to trigger the alarm status of these tags and other features. The main feature is a web server with HMI applications. This feature is common in the solution described in [17]. The most important features are the ability to be integrated with various vendors and the ability for its functionality to be extended with JAVA.

An **eFive VPN server** provides a VPN tunnel to a corporate network. The combination of an eFive and an eWON allows a central SCADA system to be created on the internet network and even the geographic locations of the individual components are insignificant. The server contains a native embedded firewall and can manage the routing of the internal corporate network. It also provides data logging in case of failure or communication issues. The eFive was designed to work with eWON routers, which is why we decided to use an eFive to guarantee secure access to our services implemented on an eWON.

Microsoft Azure Pack lets its users build a private cloud infrastructure. It supports web services and SQL databases. The Azure Pack is comparable to Microsoft Azure, but has fewer features (Figure 6). However, the Azure Pack has public “self-service” and “multi-tenant” features and our intention is to use the “multi-tenant” feature. The plant’s IT support can create its own Azure subscription for each division of the plant. Each division is a client and can have sufficient resources easily manageable by their IT support (“self-service”). This solution has huge potential for the plants where the possession of their own private cloud and internal management of data, software and hardware is required. We chose Azure because many automation projects design their software for Microsoft products.

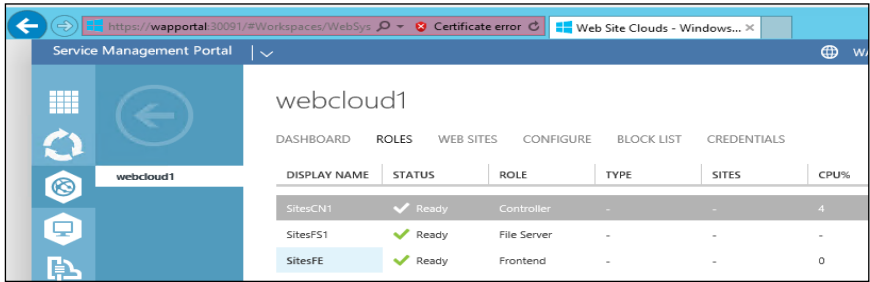


Figure 6
The Windows Azure pack portal of a private cloud

2.5 Implementation of the Designed Architecture

We deployed the Windows Azure Pack across seven virtual machine instances using HyperV, which was the minimal configuration. The Windows Azure private cloud offered a flexible database and SOA collecting data from the PLC via the technological router. The router has limited storage for historical data. Therefore, we used the cloud database for data storage.

The integration with Google servers is another feature of the designed architecture. We designed and developed a private cloud service to write important data to the Google server. After that, these data are accessible to other users such as plant suppliers. In our solution, we used Google cloud to simulate the interactivity with some cooperating industrial companies outside of the plant’s private zone.

In the Cloud, we implemented a service for mobile devices. The service supports the tag management of the simulated assembly lines controlled by the PLC, and it also collects, stores and processes data from the PLC. It is accessible to mobile devices via an interface we designed for them that allows them to interact with the service and gives them the ability to send information on production quality, alarms or tag values.

The communication with the eWON was designed with a secure communication channel using OpenSSL. This security feature ensures a secure communication with the PLC-controlled technological processes. Our additional contribution is an implementation of SOA in an eWon industrial programmable router which does not come with any SOA functionality. Therefore, we have designed and implemented the SOA functionality at the beginning. Having done that, the eWON can be easily integrated and its services can be used by applications running on mobile devices, Android applications behind the VPN server and outside of the plant’s private zone.

The last important part of the designed architecture is the PLC. We implemented a PLC program to control a greenhouse and simulate production lines. We control

the greenhouse temperature, lighting and humidity. The Rockwell PLC is connected to the eWON industrial router via Ethernet and uses the ABLogix protocol to communicate with it.

2.6 MES and HMI Mobile Clients

We created two types of clients. These client applications are consumers of the cloud services or consumers of the gateway/mediator services:

- An Android HMI SOA client application that monitors and controls the greenhouse model. The client uses the gateway/propagator services (implemented SOA in an eWON) to communicate with the PLC.
- A Windows tablet client that monitors the performance of assembly lines. This application is a MES client application and uses services implemented in the Azure Cloud.

The Android mobile device and the HMI Android application are shown in Figure 7. The tested controlled environment represents the green house controlled by the PLC.

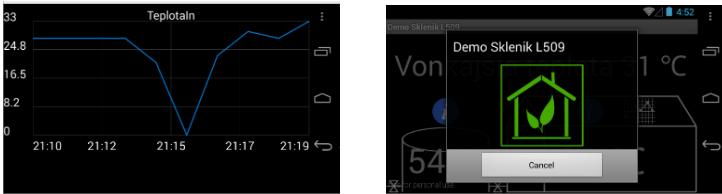


Figure 7

Data from the eWON technological router. Data were sent to the mobile device via a VPN and a SOA service implemented on the eWON industrial router

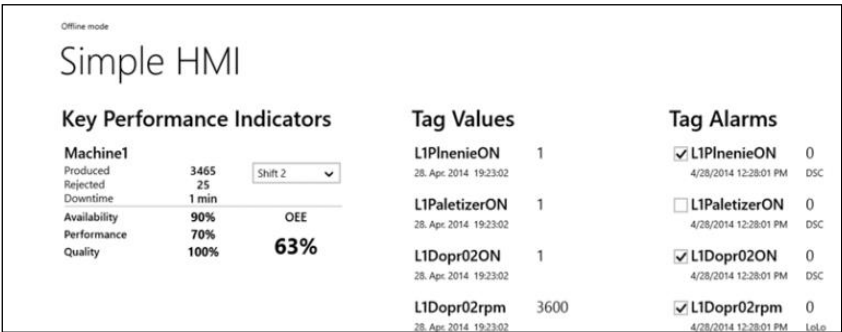


Figure 8

The main screen of the Machine1 of the production line. Data were delivered to the tablet MES Windows store application via the composite MES service

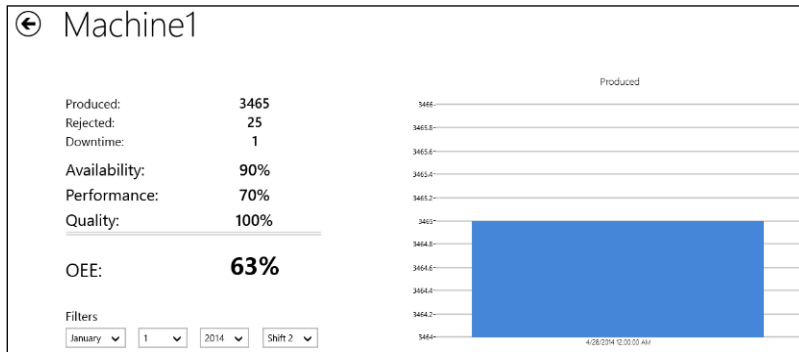


Figure 9

The Machine1's detailed view of the production line. Data were delivered to the Windows store application via the composite MES service

We developed the Windows client application as a Windows store application, too. The application enables managers to monitor the assembly lines, to set the KPI or the OEE. The application screens are shown in Figure 8 and Figure 9.

3 Testing and Evaluation of the Designed Architecture

We implemented our proposed architecture in laboratory conditions already. In the future, we plan to cooperate with an industrial company to implement and test this solution in real-world conditions. We took measurements in the case studies performed in our laboratory.

We verified our architecture in two case studies (Figure 10). Each case represents one division of a plant. The first is responsible for the greenhouse and the second for the assembly lines. Both divisions use PLCs, the Cloud and SOA.

The greenhouse uses a PLC and is directly connected to the Windows Azure Pack. The assembly lines use an eWON router to integrate the data from their PLC and to communicate using SOA. We created the services in Azure. The Windows Azure Pack enables communication with the services on the eWON router and interoperability with the Google server.

Firstly, we performed a qualitative comparison of the divisions using the attributes described in Table 2.

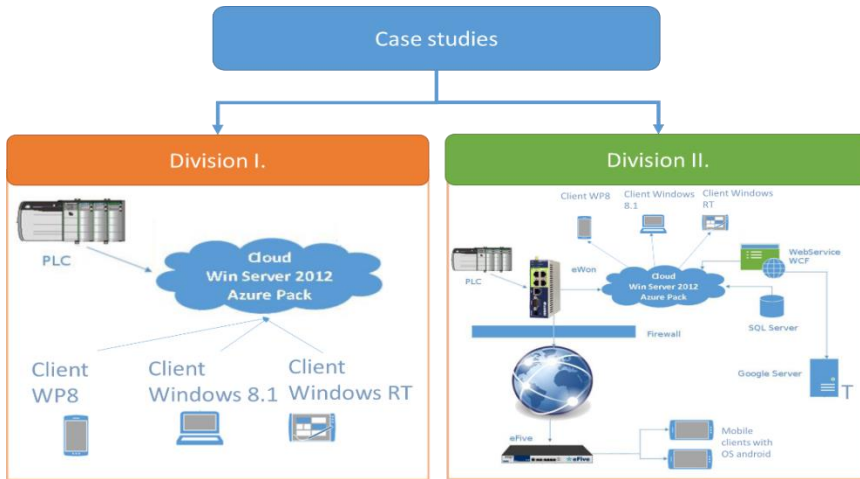


Figure 10

Two case studies used to test our architecture

Table 2

Comparison of the designed architecture implemented by two divisions

Attribute	Division I	Division II	Comment
Data accessibility	Periodic data reading (every 250 ms).	eWON enabled better accessibility with SOA.	SOA interacted with the cloud service only when data was changed.
Client interaction	The private cloud stored production data and had no public connection.	The SOA in the eWON interacts with local SOA clients and with remote clients through the firewall.	SOA enabled connection to the data accessible to the VPN client and did not allow connection to the sensitive data stored in the cloud.
HMI/MES	Direct connection to the PLC, efficiently reads/writes data, problem to integrate another device.	SOA makes it easier to interact with various industrial devices.	The Cloud and SOA provide great device and data integration for data acquisition and control.
Integration	Important to implement protocols for every special industrial device.	Thanks to the eWON and SOA, the cloud has access to the industrial devices of different types using various protocols.	The implementation of SOA-based gateways is a great advancement.
Interaction with third party companies/suppliers	The Cloud can integrate and analyze data faster, provides data to the local clients.	With the help of a shared Google calendar, and a VPN and SOA connection directly to the devices, the suppliers can immediately use the services, etc.	SOA increased the interoperability with third party companies and suppliers.

According to Table 2, the conclusion is that Division I. has faster data processing and reaction with the designed attributes. However, Division II offers easier integration of plant-floor devices and better scalability and modularity in lower levels. Therefore, Division II has bigger potential as a future modular, cooperative, agile, and heterogeneous plant architecture.

We defined our desired improvements in chapter 1.2. According to them, we created an open and an easily-integratable architecture. We chose the programmable router to satisfy any further requirements for the M2M communication. The better integrability of our infrastructure increased the SCADA performance. The common industry solution uses a local OPC or an aggregating point that resends data to the Cloud. This solution requires specialized implementation in various operating system too. In our architecture, the integration was improved because of our composite cloud services that uses atomic services from the field level. From another point of the view, SCADA and MES inherit beneficial features from the Cloud and SOA such as data availability, fault tolerance, disaster recovery and security.

3.1 Evaluation of Quality of Services

To evaluate our architecture, we defined the quality of services and their usability as the evaluation of the architecture in our experimental part. In our realized divisions, we measured the number of requests and their response time. We wanted to evaluate the quality and reusability of the services. Therefore, we designed an evaluation rule (1) which defines the quality of services. We used our own evaluation rule, which only depends on the response time of the services, general errors and internal errors of the services.

$$\theta = \frac{\sum_{i=0}^N K_i}{(N * (t_T - t_0))} * \frac{n_r}{(n_r + n_e + n_0)} \quad (1)$$

We defined K as the load factor for atomic services. We identified it as the reciprocal value of the time between accepting the client's call and the service's response to it. N represents the number of called services that are executed in the sample interval set by the user. The $t_T - t_0$ represents the selected time interval, and n_r represents the number of the executed services that finished with an OK state, n_e is the count of requests that resulted in an error state, and n_0 represents the number of services which were not executed due to service errors.

We used the formula for measuring the quality of our implemented services in a simulation to empirically evaluate the solutions in Division I. and Division II. We were not able to simulate every situation in the laboratory, therefore we created a simulation process. This process consists of 8 requests that may receive three types of answers: 0 - Error with the information about the error, 1- OK, 2 - The

service did not answer within 1000 ms. The tests consisted of ten cases and each case consisted of eight calls.

The quality of the services is large in cases 3, 5 and 10 (Figure 11). The time of the response was low in these three cases and the states 0 (error) or 2 (the service did not answer within 1000 ms) did not occur. There was one state 0 and state 2 in case 9. Therefore, the quality value is the lowest.

The quality value of services in Division II. is improved when states 0 and 2 do not appear (case 5). However, if state 0 appears at least once in Division II, then the quality value of Division I is superior (case 6) because of its better error recovery. The after-error recovery will be addressed in further research.

This situation was only simulated in our laboratory implementation and may behave differently in real-world implementations.



Figure 11

The test results of the quality of our services (Eq. (1))

A different behavior of the systems in the simulated environment is noticeable. However, we cannot yet strictly decide which division represents a better solution for real-world conditions. Both solutions are stable for industrial divisions and can improve the interoperability, data processing, HMI and control in the industry. However, there is still room for further research and safety tests should be done to verify our solution in real-world conditions.

Conclusions

We proposed an architecture that improves the data accessibility, integration and interoperability in the industry. One of our goals was to improve the SCADA and HMI interoperability. The architecture we designed and tested makes use of a private cloud and SOA. We implemented our own design of SOA on an eWON

industrial router that improves the interactivity with the integrated industrial devices, the clients and the Cloud. We also created atomic and component services to improve the SCADA and HMI interactivity. In the Cloud, we also implemented atomic services for MES clients working with data stored by another service in the cloud database. Thanks to the Windows Azure Pack, our architecture offers multi-tenancy for individual plant divisions. We tested two architectures and analyzed the results. The tests were performed on two plant divisions. According to the obtained results, we confirmed that using SOA for integration is more suitable than using a simple cloud solution without the service we designed and implemented on the eWON industrial router. The final architecture creates a universal access to the Cloud and benefits from an easy connection to the existing process level. The SOA gateway creates a consistent system by offering atomic services for our composite services in the Cloud modules.

Acknowledgement

This research was funded by the FEI Grants 2015 FEI-2015-10 project (70%) and KEGA - 001TUKE-4/2015 (30%). This project was carried out with the help of Jozef Sroka and Ladislav Pataki.

References

- [1] J. Rhoton, *Cloud Computing Explained: Implementation Handbook for Enterprises*, London: Recursive Press, ISBN 0-9563556-0-9, 2010
- [2] P. Mell, T. Grance, *The NIST Definition of Cloud Computing*, Gaithersburg: National Institute of Standards and Technology, 2011
- [3] S. Karnouskos, A. W. Colombo, "Architecting the Next Generation of Service-based SCADA/DCS System of Systems", *IECON 2011 - 37th Annual Conference on IE EE Industrial Electronics Society*, Melbourne: IEEE, 2011, ISBN 978- 1 - 61284- 969 – 0, 2011, pp. 359-364
- [4] A. W. Colombo, et al., "Industrial Cloud-based Cyber-Physical Systems", *The IMC-AESOP Approach*, 2014
- [5] B. Dániel, P. Korondi, G. Sziebig, T. Thomessen, "Evaluation of Flexible Graphical User Interface for Intuitive Human Robot Interactions". *Acta Polytechnica Hungarica*, 11(1), 2014, pp. 135-151
- [6] B. Lydon, "The 4th Industrial Revolution, Industry 4.0", *Unfolding at Hannover Messe 2014*, Automation.com, 2014
- [7] W. Stolarz, M. Woda, "A Tenant-based Resource Allocation Model Application in a Public Cloud", *Acta Polytechnica Hungarica* 12.7 (2015)
- [8] A. Soetedjo, et al. Web-SCADA for Monitoring and Controlling Hybrid Wind-PV Power System. TELKOMNIKA (Telecommunication Computing Electronics and Control), [S.l.], Vol. 12, No. 2, pp. 305-314, Jun. 2014, ISSN 2302-9293

- [9] K. Breiner, D. Görlich, O. Maschino, G. Meixner, D. Zühlke, “Run-Time Adaptation of a Universal User Interface for Ambient Intelligent Production Environments”, *Human-Computer Interaction. Interacting in Various Application Domains*, Springer Berlin Heidelberg, 2009, pp. 663-672
- [10] A. Kovarova, M. Konopka, L. Sekerak, P. Navrat, “Visualising Software Developers' Activity Logs to Facilitate Explorative Analysis”, *Acta Polytechnica Hungarica*, 13.2, 2016, pp. 159-178
- [11] T. Lojka, M. Bundzel, I. Zolotová, “Industrial Gateway for Data Acquisition and Remote Control”, *Acta Electrotechnica et Informatica*. Vol. 15/2 (2015), s. 43-48. - ISSN 1335-8243, 2015
- [12] S. Weerawarana, et al., *Web Services Platform Architecture: SOAP, WSDL, WS-policy, WS-addressing, WS-BPEL, WS-reliable messaging and more*. Prentice Hall PTR, 2005
- [13] D. K. Barry, D. Dick, *Web Services, Service - Oriented Architectures, and Cloud Computing*. Morgan Kaufmann, ISBN 978 - 0-12398 - 357 - 2, 2013, p. 248
- [14] P. Spiess, et al., “SOA-based Integration of the Internet of Things in Enterprise Services”, *Web Services*, 2009. ICWS 2009. IEEE International Conference, IEEE (2009, 2009), pp. 968-975
- [15] D. Booth, et al., *Web Services Architecture*, W3C® (MIT, ERCIM, Keio) 2004
- [16] I. Zolotová, M. Bundzel, T. Lojka, “Industry IoT Gateway for Cloud Connectivity”, *Advances in Production Management Systems: Innovative Production Management Towards Sustainable Growth*, Springer International Publishing, 2015, pp. 59-66
- [17] O. Bingol, et al. “Web-based Smart Home Automation”, PLC-controlled Implementation. *Acta Polytechnica Hungarica* 11.3, 2014, pp. 51-63
- [18] B. Kulvatunyou, H. Cho and Y. J. Son, “A Semantic Web Service Framework to Support Intelligent Distributed Manufacturing”, *International Journal of Knowledge-based and Intelligent Engineering Systems* 9.2 (2005), ISSN: 1327-2314, 2005, pp. 107-127
- [19] T. Lojka, I. Zolotová, “Online Data Stream Mining in Distributed Sensor Network”, *WSEAS Transactions on Circuits and Systems*. Vol. 13 (2014), ISSN 1109-2734, 2014, pp. 412-421
- [20] J. Lee, B. Bagheri, H. A. Kao, “A Cyber-Physical Systems Architecture for Industry 4.0-based Manufacturing Systems”, *Manuf Lett*, 2015, pp. 18-23
- [21] L. Wang, X. Xu, “Advances and Challenges in Cloud Manufacturing”, *Journal of Manufacturing Science and Engineering*, 137 (4), 2015