A scalable and flexible Open Source Big Data architecture for small and medium-sized enterprises

Luis Íñiguez[1,2][0000−0003−2615−1993] and Mikel Galar[1][0000−0003−2865−6549]

[1] Institute of Smart Cities
Public University of Navarre, Campus Arrosadia s/n, 31006 Pamplona, Spain
iniguez.71891@e.unavarra.es,mikel.galar@unavarra.es
[2] Karosseriewerke Dresden España luis.iniguez@kwdag.com

Abstract. The advancements of Big Data, Internet of Things and Artificial Intelligence are causing the industrial revolution known as Industry 4.0. For automated factories, adopting the necessary technologies for its implementation involves a series of challenges such as the lack of a proper infrastructure, financial limitations, coordination problems or a low understanding of Industry 4.0 implications. Additionally, many implementations focus on solving specific problems without taking other future or parallel projects into account, leading to continuous restructuring and increased complexity, that is, increasing costs. A lack of a global view when implementing Industry 4.0 solutions can cause difficulties in its adoption, leading to future problems that may be unaffordable for Small and Medium-sized Enterprises (SMEs). Traditional Big Data architectures offer remarkable solutions to complex data issues, but do not cover the complete flow of information that is required in Industry 4.0 applications. Therefore, there is a need to create solutions for the difficulties that this new digital transformation brings to avoid future problems, making it affordable also for SMEs. In this work we propose a flexible and scalable Big Data architecture that is well-suited for SMEs with automated factories, taking the aforementioned difficulties into account.

Keywords: Big data architecture · Industry 4.0 · Edge computing · Industrial internet of things · Open Source Software

1 Introduction

Each industrial revolution has brought changes in the manufacturing process, from the use of steam engines to the development of automated controllers, each revolution has added complexity in exchange for substantial productivity improvements. The latest revolution known as Industry 4.0 focuses on the development of cyber-physical systems [15], which combine improved machine communication through the Internet of Things (IoT) technologies with machine learning algorithms to offer numerous intelligent applications. As a result, typical

industrial software solutions have been improved. Procedures such as predictive maintenance [10] or automatic quality controls [17], now have access to a greater variety of data and better tools to work with, improving the overall factory performance.

Despite of the numerous achievable benefits, implementing industry 4.0 applications brings several challenges [14]. Typical problems are lack of infrastructure, financial limitations or a low awareness of the scope of Industry 4.0. Consequently, existing implementations are scarce or focused on solving specific problems. In the case of SMEs, the problem grows due to the greater financial constraints and their lower degree of technological development [3], which limits future implementations.

Generally, there is a tendency to think of specific solutions to individual problems, creating the necessary infrastructure for each problem. When several solutions must be integrated or it is necessary to add a new solution for a problem that interacts with the previous ones, problems arise because of not having thought globally, implementing the appropriate common infrastructure. This inevitably leads to costly re-implementations due to interoperability needs, to the addition of new processes or to operational changes in the factory.

One way to ease the development of the infrastructure is with Open Source Software (OSS) projects, which offer the latest advances. Proof of this are the multiple OSS initiatives coming from leading high-tech entities [11]. As a result of being widely used, these software projects grow in robustness, quality and stability over time. Additionally, as OSS offers access to the code, third parties can easily collaborate, enhancing the development speed.

Regarding the management of the information within Industry 4.0, it is evident that Big Data technologies play a fundamental role. Data issues like volume, velocity and variability [8] are classic Big Data problems that are also present in Industry 4.0. Therefore, a proper implementation of a Big Data architecture is necessary, although it does not fully cover the whole casuistry in industry, since the heterogeneity of specific industrial and enterprise data sources in a factory must be properly handled to make them available for different applications.

Therefore, there is a need for solutions that are easy to implement yet scalable with a global view of the factories to avoid interoperability problems related to isolated Industry 4.0 project. For this reason, in this work we propose a big data architecture driven by OSS that is affordable for SMEs with automated factories. This architecture allows the development of multiple Industry 4.0 solutions while being adaptable to changes and scalable as the company grows.

This work is organized as follows. First, Section 2 describes the state of the art on Big Data architectures. Section 3 lists the components needed by the architecture with different OSS solutions. Section 4 points out the requirements for a Big Data architecture oriented towards automated factories in SME. Section 5 presents the proposed solution. Last, Section 6 presents the conclusions and future work.

2 Big Data architectures

In this section we recall well-known Big Data architectures and some practical adaptations found in the literature.

Lambda architecture: Lambda architecture is a widely known big data architecture that combines batch and real-time views to offer data consistency [13]. It is composed of a batch layer to manage a master dataset and precompute batch views, a serving layer to serve batch views, and a speed layer to serve real-time data of not yet processed by batch jobs. Consequently, queries are made against batch and real-time views to ensure data consistency with low latency. This is a complex architecture because each data pipeline needs its own code that must be kept synchronized to bring data consistency when queries are done in both pipelines.

Kappa architecture: Kappa architecture is developed as an alternative due to the complexity of lambda architecture [6]. Kappa architecture brings a simpler format where batch and real-time tasks are done by stream processing technologies. It mainly relays in data messaging technologies to store incoming data to be used as streams. In this architecture, data is treated as a stream and have a short live-time. Consequently, this architecture is limited to certain use cases such as real-time analytical scenarios.

Liquid architecture: Liquid architecture is created as a data integration stack to provide low latency data access [2]. It is composed of two layers, a processing layer and a messaging layer. The processing layer is used to execute Extract, Transform and Load (ETL) jobs, guarantee service, provide low latency results and enable incremental data processing. The messaging layer is used to store high-volume data with high availability and access to data through metadata. The key feature of liquid is decoupling consumers and producers, increasing flexibility between the different use cases.

The exposed architectures focus on solving important problems within Big Data, but they do not cover the whole scenario in specific applications. As a result, different Big Data architectures are being implemented within the industry and adapted according to specific use cases. Thus, in the literature we can find implementations such as Nadal et al. [16], which adapts the lambda architecture adding semantic information to different use cases. Otherwise, Vouros et al. [20] develops an architecture similar to Kappa and Liquid to perform continuous stream processing jobs to provide responses at different time rates depending on the different application requirements, all of them in the heterogeneous framework of mobility analytics. An example of a specific application in Industry 4.0 is presented by Santos et al. [19], where the architecture is implemented in a multinational enterprise (MNE). In this work, aspects such as the heterogeneity of data sources, metadata management or different type of data applications are taken into account. However, it does not provide a mechanism to extract data from shopfloor devices. In contrast, in our proposal we focus specially on how to gather data from industrial machines and offer it to different applications with a scalable implementation. To do so, our proposal is inspired by the mentioned

architectures adapting them to Industry 4.0 casuistry in SMEs with automated factories.

3 Architecture components and OSS solutions

An Industry 4.0 infrastructure supporting multiple applications is complex. It must be able to collect information from different data sources, process and distribute them between different services at a reasonable time. This involves several components that must interoperate. Additionally, other services such as security, data quality and fast transmission of results must be ensured. There are many software solutions including cloud providers that offer solutions to these components, but we focus on OSS solutions since they are globally accessible, avoiding vendors lock-in. In this section we recall the necessary components to create a Industry 4.0 infrastructure with OSS alternatives.

Industrial communication protocols: Industry 4.0 requires an effective communication with the different components in a factory. There is a legacy of the past industrial stage that has led to low-level communication protocols such as MODBUS, PROFIBUS or PROFINET being used in devices such as Programmable Logic Controllers (PLC) or industrial machinery.

Over time, both communication protocols and PLCs have evolved to optimize processes and integrate higher-level software elements. However, today there is a gap between low-level industrial communication and high-level communication services. There are developments in the literature that have tried to unify these two worlds [1,9], however, the direction that is currently being taken is different.

At a sensory level, IoT technologies have made light and reliable communication between devices with different services possible. Thus, technologies such as Message Queuing Telemetry Transport (MQTT) are being used as communication protocols. However, many of the industrial device sensors do not have this technology and communication is oriented towards the PLC via legacy communication protocols.

PLCs, being key elements of automation and a communication junction point, have undergone further evolution. The PLC can be used for extracting information from the elements in an automated process. However, the communication protocols used by PLCs vary with the manufacturer. Projects such as Apache PLC4X proposes an OSS solution to communicate with these devices, offering the possibility of communicating with the PLCs of the main manufacturers.

Additionally, traditional communication between PLC and Supervisory Control And Data Acquisition (SCADA) devices were built via OLE for Process Control (OPC) communication, whereas nowadays OPC Unified Architecture (OPC UA) protocol is being adopted. OPC UA has entered in the Indutry 4.0 as a solution to handle the heterogeneity of devices. Its implementation is growing in acceptance among PLC and industrial machinery manufacturers. An OSS implementation of OPC UA is Eclipse Milo.

Message brokers: Due to the fact that industry 4.0 requires communication with a high number of devices, the intercommunication between devices and

services can be chaotic and inefficient. Message brokers simplify and centralize the flow of information in a single service to reduce both the system complexity and communication overload. Within the OSS alternatives, we highlight RabbitMQ and Apache Kafka [7]. On the one hand, RabbitMQ allows connecting with multiple protocols such as MQTT, Advanced Message Queuing Protocol (AMQP), Simple Text Oriented Messaging Protocol (STOMP) or even WebSockets offering complex routing with different ways of messaging such as point to point, publish-subscribe and request reply. On the other hand, Apache Kafka provides a flexible way for distributing events and intermediate data across many applications. It is specially oriented to work with Big Data tools and works better with large batches of data rather than multiple small messages. In IoT we can find lightweight brokers such as Eclipse Mosquitto, which is a widely used OSS broker due to its capacity to work either with low resource devices or large servers.

Data Flow: ETL tasks are typical when data is involved. When there are several data sources, as in industry, not only the data from the devices has to be handled, but also several applications such as Enterprise Resource Planning (ERP), Manufacturing Execution System (MES) and Computerized Maintenance Management System (CMMS), which work in parallel and are integrated in the factory. Therefore, an information flow component helps to control the ETL processes to be performed. Within the OSS world we can find Apache Nifi, which has connectors available to work with various data sources, in addition to having the ability to work in a distributed manner, which makes it a scalable solution. In smaller dimensions, we have solutions such as Node-red, Eclipse Kura and Apache Minifi (a lightwieght implementation of Nifi), which are more oriented towards IoT and Industry 4.0, since they can be deployed in small devices. In combination, distributed data flow solutions can be used in large servers while lightweight solutions can be deployed in smaller devices.

Big Data processing frameworks: For data processing in Big Data environments there are two approaches, batch and stream processing. The main OSS tool for batch processing is Apache Spark. Spark is a cluster computing engine designed to be fast and general purpose. It extends the popular MapReduce model to efficiently support more types of computations. Apart from batch processing, data stream processing platforms and frameworks have increased its use in the recent years due to the necessity of quick or immediate response to events in different areas such as logistics, industry or finance. Besides from Spark, the most widely used frameworks for this purpose are Apache Storm, Apache Flink and Apache Samza.

Data Storage: Massive storage is one of Big Data key features. Big Data storage system can save large volumes of data while providing incremental scalability and data replication to avoid information loss due to hardware failure. A widely used Big Data storage option is Hadoop Distributed File System (HDFS). As a proof of its impact, HDFS has allowed the development of multiple database solutions in the Big Data landscape either by building on top of it like HBase or following its own philosophy like Apache Cassandra. Additionally, other solutions

oriented to analytical tasks have been developed, this is the case of distributed databases such as Apache Kudu or Apache Druid.

4 Architecture requirements

In this section we explore the needs to be covered by a Big Data infrastructure for SMEs. We are going to focus on the cases of SME and their transformation to industry 4.0 [12] because there are major differences in its adoption compared to MNE.

Communication with industrial elements. As described in Section 3, the infrastructure must be able to communicate with a wide variety of industrial components using different communication protocols.

Application centered. Industry 4.0 hosts numerous applications, each one requiring specific data at different time constraints. Consequently, the architecture must offer the required data as quick as possible to different applications.

Scalable. Unlike most big MNE, factories from SME start the digital transformation at a lower technological development [5]. Thus, creating a complete Industry 4.0 infrastructure given that MNEs are still developing their own infrastructures is unrealistic. Therefore, the infrastructure must grow with the factory as it incorporates and develops technology to add new processes or industrial machinery.

Flexible. Industry is focused on optimizing processes, therefore, both the different processes of a factory and the configuration of the different elements that compose it are subject to changes. In accordance, the infrastructure must be flexible to adapt quickly to changes occurring in the factory.

Financially reasonable. A major challenge for several SMEs is financial constraints. Digital transformation into Industry 4.0 is expensive since it involves different developments. An excessive implementation cost can be an expense that does not justify the benefits.

Secure. Security is a sensitive issue within the industry. Many processes involve automated machinery that, if not properly secured, can cause serious damage. At the same time, a security breach can lead to the theft of intellectual property [18]. Therefore, the infrastructure must keep security in mind.

5 Proposed solution

The proposed architecture is divided into three layers. The first layer is focused on the communication with the different elements of a factory and it is made up of a series of agents. The second layer is oriented towards the intercommunication and preprocessing, being composed of a broker, a data flow system and a processing framework for ETL. The third layer is the Big Data layer, in which all the data collected from the factory will be stored in order to carry out future analytical and artificial intelligence work. Figure 1 summarizes the technology stack of the proposed architecture.

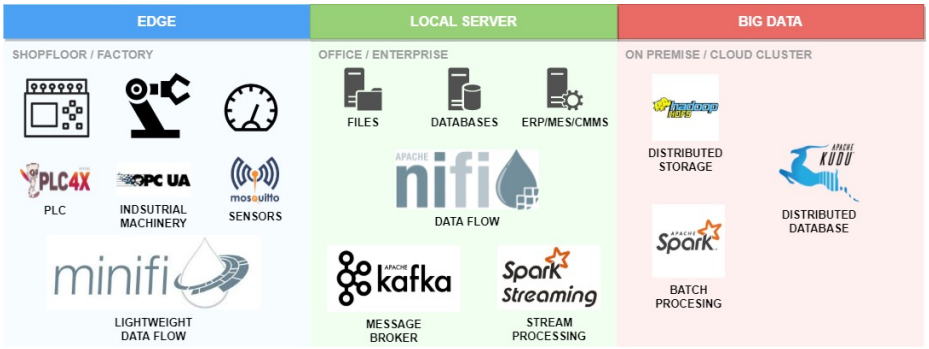

Fig. 1. Technological stack for the infrastructure

5.1 Edge layer

The task of the Edge layer is the communication with the different industrial devices within the factory. The way of achieving it is via multiple agents with the ability to connect to sensors, PLCs and industrial devices installed inside the factory network. Thus, as showed in Figure 1, each agent will use a lightweight data flow system such as Node-Red, Eclipse Kura or Apache Minifi in combination with different connectors (e.g., Eclipse Milo for machines with OPC UA, Eclipse Mosquitto to gather data from devices with MQTT communication and Apache PLC4X to work with PLCs). Additionally, the agents will have the ability to communicate with the broker included in the next layer, generating a separation between the manufacturing zone and the enterprise zone while optimizing the information load on the network. The same agents, having processing capacity, can perform lightweight preprocessing and add meta-information about the process to the data. The number of agents and their arrangement will vary depending on each factory. As a result, a way of extracting industrial data in a scalable way using OSS alternatives is achieved.

5.2 Local Server Layer

The objective of this layer is twofold: 1) to intercommunicate the Edge layer with different applications in the company and other external applications; 2) to perform a fast pre-processing of the data obtained to create a common data schema. For this purpose, we consider Apache Kafka as a distributed message broker, Apache Nifi as a data flow system and Apache Spark Streaming as a stream processing framework. The distributed broker allows separating physical industrial machines from out of the shopfloor elements as well as reducing and simplifying the flow of information. The data flow system allows access to additional data that is outside the shopfloor network (e.g., data files or databases from enterprise systems such as ERP, MES or CMMS) and sending them to the broker. Additionally, the data flow system is capable of moving data files, relieving the broker from doing this task. Finally, there is a preprocessing system

aimed to create a common schema. A key part is to offer processed data as fast as possible to enable quick responses. The data flow system may have ETL capabilities but as showed in Section 2, the combination of a distributed message system as Kafka with a stream processing framework enables fast data to applications that work at low frequencies and interoperatibility between applications. Additionally, these systems, although designed to work in a distributed manner, have the ability to work in a non-distributed manner. Consequently, the layer is scalable both in the sense of growing as its use increases, as well as being easily reproducible in other factories following the same common scheme. Additionally, it is flexible to serve several different applications, add new ones or modify them without affecting others.

5.3 Big Data Layer

Finally, this layer is dedicated to the mass storage and batch processing of the data received from the previous layer through the broker as shown in Figure 2. This layer will employ HDFS as a distributed file system, a distributed database with analytical capabilities such as Apache Kudu and Apache Spark for batch data processing framework. Thus, in this layer large processing jobs taking more time can be carried out such as analytical activities or machine learning developments. Unlike the previous one, this layer must be totally distributed since it usually grows rapidly, leading to the need for larger storage and computation requirements from the initial stages. Thus, its location can be in a private cluster in one of the factories or in a public cluster in the cloud depending on the needs of the company.

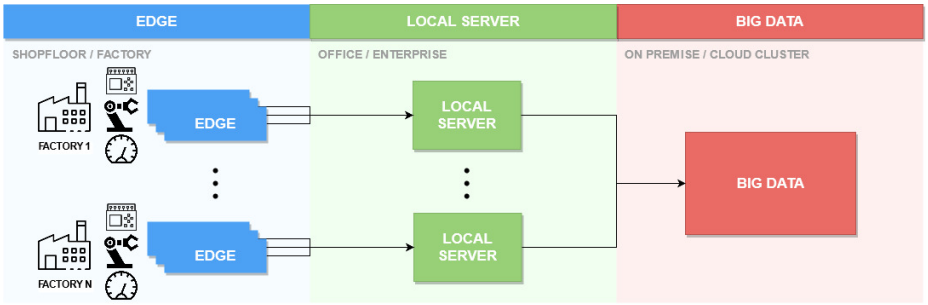

Fig. 2. Simplified architecture representation

5.4 Results

With the three layers described above, a flexible and scalable infrastructure is achieved with a flow of information from the factory processes to a big data

cluster. It should be noted that this architecture allows to serve other applications, for example, it is possible to develop a monitoring and alarm generation application locally with the data obtained in the local server layer, but it is also possible to generate a model using deep learning techniques trained in the cloud. The main advantage of this architecture is the scalability either for adding new elements in the factory or for extending it to other factories within the company. Nevertheless, there are still difficulties that are not solved with this architecture. On the one hand, there is a need for qualified personnel to develop the infrastructure, which need to know the situation of the factory's infrastructure in terms of networks, communication and machine configuration. On the other hand, using Kafka as a central messaging broker creates a critical central point of failure and implies that a change in the data schema may affect other applications, requiring additional software to handle schema versions to avoid updates affecting third-party applications

6 Conclusions and Future work

We have presented a Big Data architecture implemented with OSS solutions for SMEs bringing flexibility of deployment and design while ensuring scalability as the factory or the use cases grow. For SMEs, introducing elements of IoT, Big Data and machine learning is a significant challenge that can lead to create isolated developments and consequently, expensive reimplementations. With this architecture, the problem is diluted as it offers development independence from different parties while using a common data model for the entire company. Thus, it favors the technological development of the factory in a gradual and sustainable way. As future work, there are certain aspects that have not been addressed but can be considered to create a more robust system: IoT communication for long distance devices, other industry standards such as Reference Architecture Model Industry 4.0 (RAMI 4.0) [4], metadata management software for data schema governance or business intelligence tools for enhancing analytical tasks. Additionally, as future work, we will recreate the exposed architecture in a SME to test its capabilities.

7 Acknowledgment

This work was supported in part by the Navarre Department of University, Innovation and Digital Transformation to industrial doctorates 2020, expedient 0011-1408-2020-000006 and the collaboration between the Public University of Navarre and Karosseriewerke Dresden España, S.L.U.

References

1. Bellagente, P., Ferrari, P., Flammini, A., Rinaldi, S., Sisinni, E.: Enabling PROFINET devices to work in IoT: Characterization and requirements. In: Conference Record - IEEE Instrumentation and Measurement Technology Conference (2016)

2. Fernandez, R.C., Pietzuch, P., Kreps, J., Narkhede, N., Rao, J., Koshy, J., Lin, D., Riccomini, C., Wang, G.: Liquid: Unifying nearline and offline big data integration. CIDR 2015 - 7th Biennial Conference on Innovative Data Systems Research (2015)
3. Frank, A.G., Dalenogare, L.S., Ayala, N.F.: Industry 4.0 technologies: Implementation patterns in manufacturing companies. International Journal of Production Economics (2019)
4. Hankel, M., Rexroth, B.: The reference architectural model industrie 4.0 (rami 4.0). ZVEI (2015)
5. Horváth, D., Szabó, R.Z.: Driving forces and barriers of Industry 4.0: Do multinational and small and medium-sized companies have equal opportunities? Technological Forecasting and Social Change (2019)
6. Kreps, J.: Questioning the lambda architecture (2014), https://www.oreilly.com/radar/questioning-the-lambda-architecture/
7. Kreps, J., Corp, L., Narkhede, N., Rao, J.: Kafka: a Distributed Messaging System for Log Processing (2011)
8. Laney, D., et al.: 3d data management: Controlling data volume, velocity and variety. META group research note (2001)
9. Langmann, R., Rojas-Pena, L.F.: A PLC as an industry 4.0 component. Proceedings of 2016 13th International Conference on Remote Engineering and Virtual Instrumentation, REV 2016 (2016)
10. Li, Z., Wang, Y., Wang, K.S.: Intelligent predictive maintenance for fault diagnosis and prognosis in machine centers: Industry 4.0 scenario. Advances in Manufacturing (2017)
11. López, L., Costal, D., Ayala, C.P., Franch, X., Annosi, M.C., Glott, R., Haaland, K.: Adoption of OSS components: A goal-oriented approach. Data and Knowledge Engineering (2015)
12. Luthra, S., Kumar Mangla, S.: Evaluating challenges to Industry 4.0 initiatives for supply chain sustainability in emerging economies. Process Safety and Environmental Protection (2018)
13. Marz, N.: How to beat the cap theorem (2011), http://nathanmarz.com/blog/how-to-beat-the-cap-theorem.html
14. Moeuf, A., Lamouri, S., Pellerin, R., Tamayo-Giraldo, S., Tobon-Valencia, E., Eburdy, R.: Identification of critical success factors, risks and opportunities of Industry 4.0 in SMEs. International Journal of Production Research (2020)
15. Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W., Ueda, K.: Cyber-physical systems in manufacturing. CIRP Annals (2016)
16. Nadal, S., Herrero, V., Romero, O., Abelló, A., Franch, X., Vansummeren, S., Valerio, D.: A software reference architecture for semantic-aware Big Data systems. Information and Software Technology (2017)
17. Oliff, H., Liu, Y.: Towards Industry 4.0 Utilizing Data-Mining Techniques: A Case Study on Quality Improvement. In: Procedia CIRP (2017)
18. Pereira, T., Barreto, L., Amaral, A.: Network and information security challenges within industry 4.0 paradigm. Procedia Manufacturing (2017)
19. Santos, M.Y., Oliveira e Sá, J., Andrade, C., Vale Lima, F., Costa, E., Costa, C., Martinho, B., Galvão, J.: A Big Data system supporting Bosch Braga Industry 4.0 strategy. International Journal of Information Management (2017)
20. Vouros, G., Glenis, A., Doulkeridis, C.: The Delta Big Data Architecture for Mobility Analytics. In: 2020 IEEE Sixth International Conference on Big Data Computing Service and Applications (2020)