Building an Industrial IoT Infrastructure with open Source Software for Smart Energy.

Christos Alexakos*, Andreas Komninos†, Christos Anagnostopoulos‡,
George Kalogeras§, Alkiviadis Savvopoulos¶ and Athanasios Kalogeras‖
Industrial Systems Institute
ATHENA R.C.
Patras, Greece
Email: *alexakos@isi.gr, †komninos@isi.gr, ‡anagnostopoulos@isi.gr,
§gkalogeras@isi.gr, ¶savvopoulos@isi.gr, ‖kalogeras@isi.gr

***Abstract*—Internet of Things is the cornerstone of most of the modern technological achievements and one of the biggest sources of data. The petabytes of data generated by telemetry sensors are already used for statistical analysis or the creation of prediction models used in various applications in the area of smart cities, smart building, smart health, smart energy, etc. On the other hand, the expansion of IoT forced the need of more standardized approaches such the ones used in industrial automation. The Industrial Internet of Things (IIoT), as part of Industry 4.0 concept, promotes the cyber physical systems (CPS) as sensors and actuators that will build the modern automation world in and out of the factories. This article studies the IIoT reference architectures and the existing open source IoT platforms for proposing an integrated architecture for installing IIoT infrastructure that can collect and analyze big volume of data, easy and with low cost. The approach is evaluated in a smart building scenario.**

***Index Terms*—IoT architecture, industrial IoT, open source software, smart energy**

I. Introduction

Internet of Things (IoT) technologies provide the ability to establish complex systems that are able to sense, analyze collected information, and respond in various environments ameliorating living standards. Nowadays, more than 1600 IoT projects have been realized word-widely in various sectors such as smart cities, smart energy, smart health, etc. [1]. Exploiting the advantages of IoT technologies, the Industry 4.0 concept promotes the cyber physical systems (CPS) as the key enablers for the Industrial Internet of Things, which is extended beyond the manufacturing environments to everyday automation infrastructure paradigms like smart buildings. On the other hand, a lot of work in open source communities leads to the introduction of scalable and reliable open source platforms that are able to contribute meaningfully to the installation of IoT infrastructure.

This paper presents an approach to build an IIoT infrastructure which is based on the industrial reference architectural model of Industrial Internet Reference Architecture (IIRA). The presented use case concerns the installation of a small-scale living lab for smart energy. The scope is the creation of an innovation ecosystem where various technologies and ideas can be integrated and tested in a real environment [2], [3]. As a demonstrator, an IIoT system is installed using various networks, sensors and open source software at the premises of a research institute.

The paper is structured as follows. Section II presents the major reference architectures for the Industrial Internet of Things. Section III provides a comparison of the most commonly used open-source IoT platforms and section IV refers to the most popular big data analytics platfomrs. Section V presents the proposed installation of the living lab and its association with the the Industrial Internet Reference Architecture (IIRA). Finally, conclusions and future work are presented in Section VI.

II. IIoT Reference Architectures

The Industrial Internet of Things is part of the general IoT evolution. In fact, the impact of the integration of IoT technologies in the manufacturing environment has led to advancements deemed as the Fourth Industrial revolution. Acknowledging the importance of this new and rapidly evolving concept, several initiatives have attempted to define reference architectures, that will standardize the architectural design of IIoT applications. The *Industry 4.0 Platform* and the *Industrial Internet Consortium (IIC)* are two of the mainstream initiatives towards standardization of IIoT systems, supplemented by further initiatives such as *Japan's Society 5.0* and *Made in China 2025*.

The Industry 4.0 Platform is a high-tech strategy of the German Government, promoting computerization in manufacturing. It has developed the Reference Architectural Model Industry 4.0 (RAMI 4.0) [4]. This three-dimensional service-oriented architecture combines all elements in a layer and life

We acknowledge support of this work by the projects "eSmartCity" which is implemented under the "Interreg Mediterranean" program, co-financed by the European Regional Development Fund and "I3T - Innovative Application of Industrial Internet of Things (IIoT) in Smart Environments" (MIS 5002434) which is implemented under the "Action for the Strategic Development on the Research and Technological Sector", funded by the Operational Programme "Competitiveness, Entrepreneurship and Innovation" (NSRF 2014-2020) and co-financed by Greece and the European Union (European Regional Development Fund).

978-1-7281-3345-4/19/$31.00 ©2019 IEEE

cycle model. The three dimensions are Hierarchy, Architecture and Product Life Cycle, which define the functional areas of IIoT applications from the Smart Product to the Connected World concept, the system architecture and the aspects concerning the development and production phases of a product respectively.

In parallel with Industry 4.0, the Industrial Internet Reference Architecture (IIRA) [5] was developed in the US by the IIC, covering a broader range of possible application sectors than plain manufacturing. Similar to RAMI 4.0, this architecture has a three-dimensional approach which uses however a different perspective, covering the Product Life Cycle, the Industrial Sectors and the Viewpoints. The Industrial Sectors represent the various sectors where IIoT applications can be implemented. The Viewpoints refer to the specific concerns and different perspectives of industry stakeholders. There are 4 viewpoints in the reference architecture, namely:

- The *Business viewpoint* which refers to business-oriented concerns such as business value, expected return on investment, cost of maintenance and product liability;
- The *Usage viewpoint* which is concerned with how an IIoT system realizes the key capabilities identified in the Business viewpoint;
- The *Functional viewpoint* which focuses on IIoT System functional components, structure and interrelation as well as the necessary interfaces and interactions, and, finally;
- The *Implementation viewpoint* which is concerned with the technical representation of an IIoT system and the technologies and system components required.

Thus, the implementation viewpoint implements the activities and functions prescribed by the usage and functional viewpoints. IIoT system implementations follow certain well-established architectural patterns, such as a) Three-tier architecture pattern, b) Gateway-Mediated Edge Connectivity and Management architecture pattern and c) Layered Databus pattern.

The three-tier architecture pattern comprises the edge, platform, and enterprise tiers. These tiers play specific roles in processing the data flows and control flows involved in usage activities. They are connected by means of three networks, as shown in Fig. 1.

- The *edge tier* collects data from the edge nodes, using the proximity network. The architectural characteristics of this tier, including the breadth of distribution, location, governance scope and the nature of the proximity network, vary depending on the specific use cases.
- The *platform tier* receives, processes and forwards control commands from the enterprise tier to the edge tier. It consolidates processes and analyses data flows from the edge tier and other tiers. It provides management functions for devices and assets. It also offers non-domain specific services such as data query and analytics.
- The *enterprise tier* implements domain-specific applications, decision support systems and provides interfaces to end-users including operation specialists. The enterprise tier receives data flows from the edge and platform tier. It also issues control commands to the platform tier and edge tier.

Tiers are connected via different networks:

- The proximity network connects the sensors, actuators, devices, control systems and assets, collectively called edge nodes. It typically connects these edge nodes, as one or more clusters related to a gateway that bridges to other networks.
- The access network enables connectivity for data and control flows between the edge and the platform tiers. It may be a corporate network, or an overlay private network over the public Internet or a 4G/5G network.
- Service network enables connectivity between the services in the platform tier and the enterprise tier, and the services within each tier. It may be an overlay private network over the public Internet or the Internet itself, allowing the enterprise grade of security between end-users and various services.

III. IoT Digital Platforms

A. Open Source IoT Platforms

Production systems are progressively undertaking the digital transformation of their processes, making increasing use of data that has previously been impossible or unprofitable to collect or use. Even though the vision and the will for this digital transformation is a common ground among all the stakeholders, the exact definition of an IoT platform and what it contains and offers, varies according to the context and the vendor. The IoT platform vendor strategies continue to evolve and try to follow the evolution in areas like artificial intelligence and machine learning and simultaneously embrace innovative paradigms, like fog and edge computing. However, in general, IoT platforms must share some common characteristics like interconnecting and managing the IoT endpoints, collecting and processing the data, providing data visualization tools, and providing tools for IoT application development.

The selection of an IoT platform for a project is a difficult task. There are several surveys that compare existing IoT platforms [6]–[8], most of them focusing in performance issues [9], [10] while others specialize in specific application domains, such as smart buildings [11] or smart health [12]. For the scope of this work, comparison of IoT platforms focuses on engineering aspects like easy installation and configuration, scalability and interoperability. Taking into account these requirements, we identified and compared across the four most prevalent open source IoT platforms, namely ThingsBoard[1] , OpenHab[2], DeviceHive[3], and Kaa Project[4]. We performed a thorough comparison on the basis of several key properties which are important for the proper selection of a technological solution. These criteria are presented in Table I.

[1]https://thingsboard.io
[2]https://www.openhab.org
[3]https://devicehive.com
[4]https://www.kaaproject.org

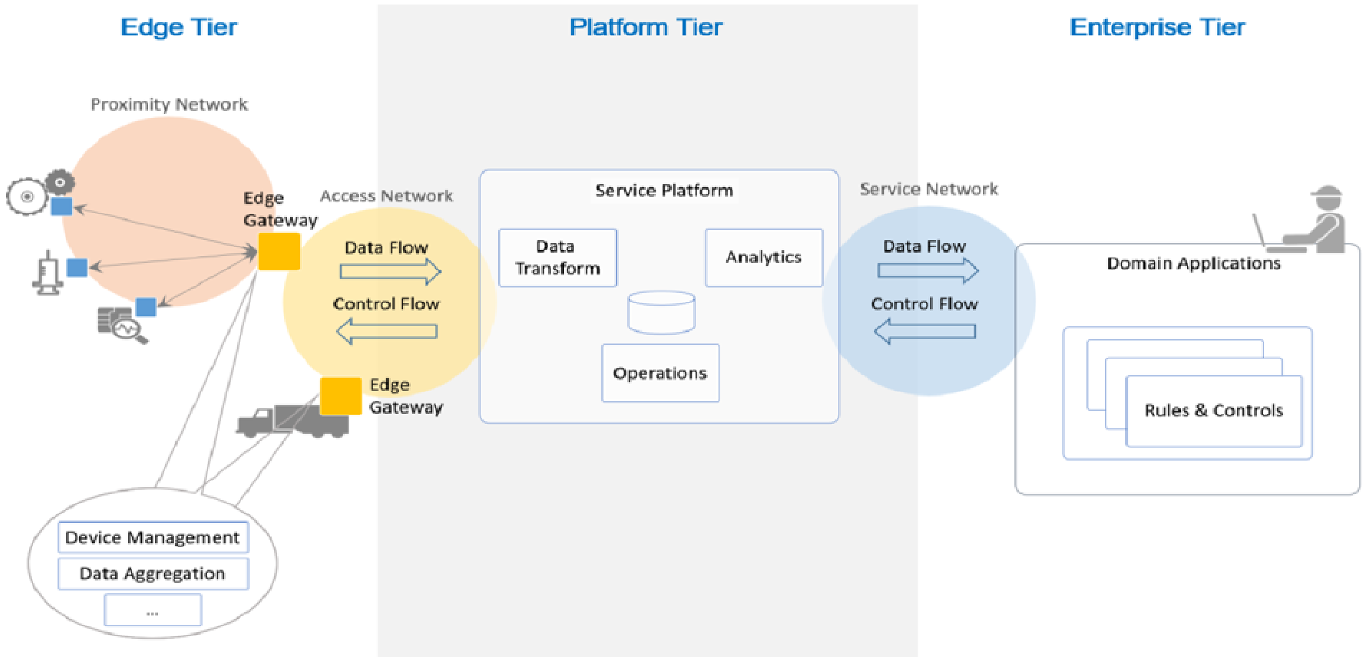

Fig. 1. Three-Tier IIoT System Architecture for IIRA.

TABLE I
IoT Platform Comparison Properties

Property	Description
Technical Characteristics	HW requirements, programming language, OS, installation method
Architecture	Monolithic, microservices etc.
Services	Device management, data visualization, remote command execution etc.
Documentation	User manual, developer manual, API reference etc.
Supported Protocols	MQTT, CoAP, HTTP, REST API, OPCUA etc.
Device Management	Firmware update, role support, device control etc.
Security	HTTP/SSL support, device authentication etc.
Data persistence technologies	SQL, noSQL, Postgress, Cassandra DB etc.
Data Pre-Processing	Support of aggregation functions, filtering, pipeline etc.
Data Visualization	Widgets, Tables etc.
Data Analytics	Connection with data analytics platforms and tools
Scalability	API or SDK for the integration or communication with and other system

B. Comparison results

All four platforms examined cover the basic specifications of a digital platform that would support an industrial internet of things infrastructure. Each of them can interconnect IoT devices and gather data from them. Furthermore, both relational and NoSQL databases are supported to store the collected data from the devices. All platforms under consideration are JAVA-developed and provided under licenses that allow their free use without limitations in various applications. They also all support microservices architectures, allowing its use in fog architectures. Nonetheless, the offerings of each platform varies with respect to each property, so the appropriateness of each platform is strongly dependent on the needs of individual applications.

The Thingsboard [13] platform is the latest open source effort with active presence and support in the open source community. It can be considered as the most complete platform, since its free version satisfies the requirements of most properties, such as device management, collection, preprocessing, editing and graphical presentation of the data. It can be installed easily and it is easy to interconnect a new device since it is accompanied by a gateway that allows the two-way connection of devices with multiple communication protocols and message structure. Furthermore, it supports industrial protocols like the OPC-UA and Modbus. Finally, it provides a complete graphical environment with a decision engine that allows user and device feedback according to rules on data values.

The openHab [14] platform is the most mature work in the field of building automation using the Internet of Things. It is supported by a large community, which also helps in the development of plugins for interfacing with various commercial devices. Moreover, it fully supports all the basic data collection, processing and presentation functions. It lags behind others in installation and customization complexity, and in providing a friendly graphical environment. A multiplicity of communication protocols are supported via the existing plugins, yet industrial protocols are not supported. From a security standpoint, it does not support device authentication and graded access.

The DeviceHive [15] platform is another attempt from the open source community. Its distinguishing feature is the provision of an SDK in several languages which enables the further extension of its functions. It does not support graphical data representation and data analysis for which a third-party application is required. The interconnection of a new device may pose serious difficulties due to the lack of built-in support of many known device interface protocols.

The KAA [16] platform is supported by a large group of developers but lacks recent updates. The platform supports specific data collection and storage functions. Data processing and graphical visualization are only supported by the commercial version.

Each platform has positives and negatives aspects, according to which it can be considered satisfactory depending on the specifications of each application use-case. In the context of an industrial environment, in which it is necessary to integrate various heterogeneous industrial network devices of things, and where there is also a need for bidirectional communication with the devices, the platforms must support multiple protocols and interact with devices in an untroubled manner. In addition, the degree of support by the community is a decisive factor that must be taken into account. From this perspective, the ThingsBoard and openHab platforms seem to be ahead from the rest of the competition.

IV. Data Analytic Platforms

A. Overview of modern data analytics

The industrial IoT is characterised strongly by the hallmark 3Vs of Big Data, as described in the classical definition of Big Data by Laney [17]: *Volume, Variety and Velocity*. Labrinidis and Javadish [18] define the process of managing big data as consistent of two distinct levels, namely *Data Management* (where data is acquired and subsequently pre-processed, e.g. for the purposes of cleaning, transforming, integrating, aggregating etc., and *Analytics* (where data is modelled and analysed, leading to interpretations and knowledge). As explained by Tsai et al. [19], the process of data analysis (or data analytics) utilizes the outputs of the data management process as input, in order to find hidden patterns, rules and information from the data. For this purpose, a range of statistical techniques is often used to mine data, though more recently, other analytics methods, such as supervised and unsupervised machine learning techniques, have begun to rapidly gain ground, used either exclusively, or in conjunction with statistical processing techniques. Big Data analytics are typically classified as *Descriptive, Predictive* and *Prescriptive analytics* [20]. In the context of the IoT, analytics can also be applied in real-time (i.e. upon the incoming stream of data), or off-line (on stored data).

Although data analytics can be applied for a wide range of purposes, Gandomi and Halder [21] outline some of the most common uses for multimodal data analytics. In text analytics, typical uses are information extraction (structured data and relationships from unstructured texts), summarization (from single documents or collections), question answering (responses to questions posed in natural language) and sentiment analysis. Another type of use case is audio analytics, either applied to sounds or speech, in order to extract structured information (e.g. audio features to help recognise and classify sound, or extract semantics and sentiment from speech). Video analytics refers to the extraction of information from static images or video streams (e.g. image classification and content description, detection of presence). Social network analytics includes several techniques found in text and multimedia analytics (content-based analytics), but also focuses on the relationships between users and entities, since social networks are typically represented as graph structures (e.g. community detection, social influence and link prediction). Finally, predictive analytics is a modern trend that attempts to provide predictions on the future values of certain data, by generating data models trained on historical data. Common examples of these could be predicting a user's interest in particular products, forecasting the electricity demands of a region, or inferring the likely remaining time-to-failure of a hardware component.

B. Open Source Big Data Platforms

To address the challenges of Big Data in the IoT, the concept of scalability is fundamental. In this respect, there are two major approaches, namely to horizontally scale a system (i.e. add more servers to increase storage capacity and distribute computational loads), or to vertically scale a system (i.e. add more hardware resources to existing servers, such as more RAM, processors or GPUs).

Whether one choses horizontal or vertical scalability, in the open-source world there are few product offerings that combine all the necessary components for Big Data analytics (storage, pre-processing, analysis) in one single package. Instead, most solutions based on open-source software, depend on a range of base components that manage data storage, querying and program execution, upon which further components for data analytics can be installed and executed. This allows system developers to adopt a mix-and-match approach, depending on their needs and available expertise. In the next section, we present the most well-known such components.

C. Base components

The most well-known Big Data platform is Apache Hadoop [22]. This consists of a distributed file system (HDFS) for storage across a large number of nodes, and a resource management layer (YARN) which schedules jobs across the node cluster. The MapReduce programming model allows parallel execution of data queries, which can be simplified by using other components such as Apache Hive or Apache Pig, which offer SQL-like capabilities to programmers.

Hadoop is well-suited to batch processing of large datasets, however it suffers from a speed disadvantage, since at every processing step, the results are written to and read from hard drive storage. Apache Spark[5] is a more recent platform for big data management, which relies on the storage of immutable datasets in RAM, instead of hard drives. Its operational concepts are very similar to Hadoop, however it offers a tremendous speed advantage, at the cost of more expensive hardware requirements [23]. Spark's speed improvements make it suitable for batch processing, as well as, to some extent, stream processing.

Finally, we mention here two horizontally scalable database platforms, which offer NoSQL database storage and querying components. The most well-known examples are MongoDB and Apache Cassandra, both of which offer rapid performance in querying and fault tolerance through distributed database storage. Extending the suite of options, Neo4j is perhaps the most commonly used graph database, allowing the representation of semantically linked data.

D. Data analytics components

Most of the open-source solutions in data analytics offer bolt-on compatibility with base components such as Hadoop, and are typically offered alongside a more feature-rich paid version. Many implementations refer to "data analytics" in the context of visualizing data and generating reports. While these techniques offer valuable services, we focus on those implementations which include machine-learning algorithm integrations, aligning with the data analytics definition in [21]. For the purpose of this work, we selected three machine learning supporting open-source platforms for Big Data analytics. These offerings are the Knowage[6], the H2O[7] and KNIME[8]. A summary of the features found in each offering is shown in Table II. Notably, the Apache Spark platform also offers its own suite of machine-learning tools (MLlib).

TABLE II
IoT Platforms Comparison Characteristics

Software Platform	Base support	Feature set	Data processing	ML algorithms
Knowage	Spark	Limited	Yes	Yes
H2O	Hadoop, Spark	Full	No	Yes
KNIME	Hadoop, Spark	Full	Yes	Yes

Further from the above, data analytics solutions developers may base their solutions on popular programming languages (such as R, Python) which include many pre-compiled libraries for statistical and machine learning processing. These also include libraries for connecting to popular base components such as Spark and Hadoop, making data integration easy. Software libraries for common programming languages (e.g. Weka for the Java language) can also be employed to this end.

[5]https://spark.apache.org/
[6]https://www.knowage-suite.com
[7]https://www.h2o.ai
[8]https://www.knime.com

E. Complete platforms

In this section, we discuss platforms that offer a complete suite of features for data analytics, integrating storage, processing, querying and analysis tools. Apache Spark [24] is considered the next step up from Hadoop and it also provides a scheduler, query optimizer and execution engine, which can run either standalone, or on top of an existing Hadoop installation (replacing Hadoops YARN). As mentioned, one advantage of Spark is that it comes not just with data query capabilities, but a full machine-learning library, which contains classification, regression, clustering and other ML algorithms, optimized to run under Sparks parallel execution environment.

We also note here the HPCC[9] platform, which is a complete big data storage and analysis platform, similar to Spark, offering a suite of tools for data pre-processing (Thor), data querying (Roxie), job scheduling and automation and finally processing via distributed execution machine learning algorithms, over commodity computing clusters. HPCC claims superior performance compared to Spark and Hadoop, since it supports three types of parallelism: Data (division and processing), Pipeline (two operations on the same dataset simultaneously), and System (parallel execution of independent operations).

V. A Living Lab for Energy Efficiency

A well-known source of energy inefficiency in a building reality is human behaviour. For example, while opening a window for one minute can be seen as a decision of negligible impact from the building user point of view, the actual impact of a simple action like this, depending on the external environmental conditions, could be drastic in terms of thermal dispersion and thus energy costs. It is then clear that any expression of the simplest user freedom in a building can impact considerably on the overall building energy consumption.

Following the concept of living labs, an IIoT infrastructure has been installed at our research institute's premises (Industrial Systems Institutes - ISI), in order to examine new approaches to the energy efficiency based on automation and human factors. The open source platform Thingsboard has been chosen as the central IoT platform that implements an IIRA driven architecture.

A. IoT Devices and Networking

ISI occupies three different independent offices at the building of Patras Science Park[10]. Due to building administration constraints, we were not able to install smart energy meters at the central energy infrastructure, therefore our energy use monitoring installation comprises of smart plugs and power switches.

Both commercial solutions and hardware open source custom projects are used as IoT sensors in our installation. For the smart plugs, the products of the MEAZON SA[11], a Greek

[9]https://hpccsystems.com
[10]http://www.psp.org.gr
[11]https://meazon.com

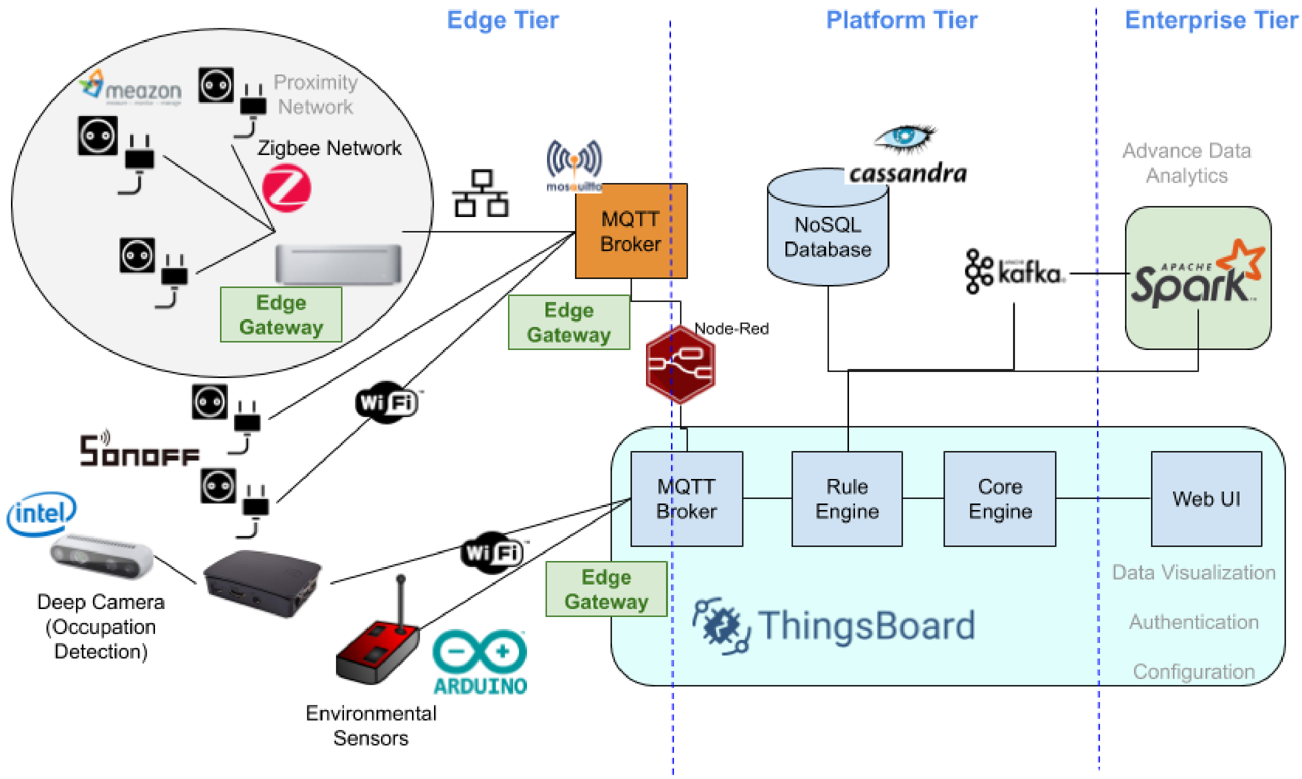

Fig. 2. IIRA-based Three-Tier IIoT System Architecture at the ISI living lab.

company with an open telemetry protocol and Sonoff[12], a series of products from the Chinese company ITEAD with open source firmware, were used. For monitoring the environmental parameters (temperature, sensitivity, light, etc.) and room occupation (motion detection), custom multi-sensor prototype devices based on Arduino boards were used.

For the networking of IoT sensors and the aggregation of the telemetry, the existing WiFi and Ethernet network infrastructure was leveraged. Furthermore, in each room a ZigBee gateway was installed, creating a proximity network for the MEAZON smart plugs to eventually communicate with the central IoT platform. Furthermore, Intel RealSense[13] depth cameras have been installed in some of the room identifying human presence and counting the number of people that co-exist in the monitored area. The depth cameras are connected with a embedded system implemented using the Raspberry Pi 3 Model B+[14] platform. It's role is to processes the image from the camera and extract the required information and to sent them to the IoT platform.

This setup simulates the heterogeneity of devices and communication protocols encountered in the IIoT, including a range of commercial-off-the-shelf (COTS) devices and custom-built hardware, communicating over a range of diverse wireless and wired network infrastructures. We use the MQTT protocol over all network connections to manage latency and reliability issues that emerge through the heterogeneity. The device abstractions afforded through the ThingsBoard platform allow for the easy integration of any type of device, as long as it can support two-way communication via MQTT.

B. IoT Architecture

The proposed architecture follows the three-tier model suggested by IIRA. Thingsboard platforms services were used for the implementation of both platform and business layer. Fig. 2 shows the functional components that are used for installing the proposed living lab.

At the edge tier, the IoT devices are communicating through the WiFi, Ethernet and ZigBee networks to a central server where the ThingsBoard Platform is used. The devices are communicating using both MQTT protocol. At the central server (Platform tier), two MQTT brokers are installed. One is used by the Thingsboard platform as the endpoint of input and output of data.The second broker is an Eclipse Mosquito broker running on a NodeRed service, which is used for communicating with devices that do not support the message content defined by the IoT Platform, thus affording additional abstraction capabilities to allow the integration of COTS devices. For IoT devices with programmable firmware, we implemented the exchange of data with platforms native MQTT. This data contains messages for sending telemetry as well as receiving commands from the platform (i.e. turn off a

[12]https://sonoff.itead.cc/en
[13]https://sonoff.itead.cc/en
[14]https://www.intelrealsense.com/

Fig. 3. Screenshot of the overall dashboard with Living Labs power metrics.

plug). For the rest of the devices that follow their own message structure, we developed an extensible data transformation pipeline management software (NodeRed broker service), which is used as gateway that translates the from and to the devices messages to ThingsBoard's messages structure. The platform tier consists of the modules of Thingsboard platform. The in-built Rule Engine is used for Data Transformation and Analytics, where the incoming and outgoing data is filtered or transformed following a rule-based logic. The transformed data is forwarded to the ThingsBoard core module for storage to the local database, leveraging the platform's in-built Kafka stream management services. A Cassandra No SQL database is used for storing the telemetry and device configuration data. Additionally, ThingsBoards core module provides assets management and user authorization and authentication services as part of the operations capability defined by IIRA.

Finally, at the Enterprise tier, a web interface allows the creation of dashboards with visualization of the telemetry data in interactive charts. An example is shown in Fig. 3.

C. Big Data Analytics

The ThingsBoard platform offers the ability to produce outgoing datastreams to external applications, through the integration module with Apache Kafka[15] service. This can be configured through the platform's rule engine, effectively setting-up a data forwarding mechanism to third party applications. In addition, it is possible to configure access to the NoSQL database for external applications so that data can be obtained in bulk for batch processing. As a result, it is relatively straightforward to implement an external processing pipeline for big data analytics, in a number of configurations.

For this, we implement an Apache Spark instance, which can be used to perform various data analytics functions. As an example, Spark's MLlib can be used to configure, train and deploy machine-learning based applications, which can offer classification-based services, such as anomalous event detection on the monitored IIoT system (e.g. abnormal operating conditions, inappropriate system states during rule operation). Another type of use is applications that offer regression-based predictive services, for example, forecasting energy demands, recommending device (plug) state configurations based on external current and forecasted weather conditions, etc. To this end, data from the ThingsBoard platform needs to be processed both in batch mode (large, historical datasets) and stream mode (incoming streaming data).

- *Batch processing using ThingsBoard storage:* Spark applications can read directly from the Cassandra database, obtaining subsets of the data which can be used to train machine learning models on the Spark instance. Models can be updated off-line at regular intervals, using recent data stored in the ThingsBoard platform. Long-term forecasts can be made using these historical data values only, and they can be invoked on-demand or automatically produced at regular intervals.
- *Batch processing using Spark storage:* For batch ML model training and prediction, another approach is to replicate incoming data streams by feeding them, through the Kafka service, into independent storage on the Spark instance. This can allow the periodic overwrites of data, keeping local subset copies on the Spark instance without the overhead of querying and retrieving directly from the ThingsBoard platform storage. This enhances data integrity through increased security and replication.
- *Stream processing:* Spark can receive data directly from Kafka and perform classification and regression tasks on the data as it arrives, using stored models built earlier through batch processing. The results of these predictions can be fed back to the ThingsBoard platform, again using the Kafka service, in order to drive rules in the Rule Engine, or allow the display of real-time forecasting on the Enterprise Tier web UI.

VI. Conclusions and Future Work

The proposed work presents a practical paradigm for applying an IIoT reference architecture in our case the IIRA utilising open source software in combination with industrial products that can been found in the market. As a use case, a small scale Living Lab was installed at the premises of Industrial Systems Institute. Through this use case, we demonstrate that the implementation of reference IIoT architectures using open-source software is not only feasible, but in fact a highly scalable and flexible approach to IIoT implementation, combining custom and COTS hardware. As future work, an extension of this Living Lab is scheduled, mainly in the Enterprise Layer, where the additional platform for big data analytics including Apache Spark will be deployed in a cloud computing infrastructure. This platform will enable the users to proceed with complex data analysis tasks and use the results to create new rules which will feed the IoT platforms rule engine. More importantly, we are looking at distributing the data analytics intelligence capabilities to reduce the reliance on single-points of failure and increase the robustness, responsiveness and scalability of the system. For this, we are investigating the implementation of multi-tier fog-based analytics, in the form of a hierarchy of individual AI-enabled boards at the edge

[15]https://kafka.apache.org

(e.g. Google Coral or Raspberry Pis with AI hats working at the room level), local analytics using Spark on Raspberry Pi nano-clusters (e.g. at floor level), and global analytics using the cloud Apache Spark configuration described in this paper (e.g. at building level).

References

[1] I. Analytics, "The top 10 iot segments in 2018–based on 1,600 real iot projects," *Website: https://iot-analytics. com/top-10-iot-segments-2018-real-iot-projects*, 2018.

[2] M. Gascó, "Living labs: Implementing open innovation in the public sector," *Government Information Quarterly*, vol. 34, no. 1, pp. 90–98, 2017.

[3] C. Bloch and M. M. Bugge, "Public sector innovationfrom theory to measurement," *Structural change and economic dynamics*, vol. 27, pp. 133–145, 2013.

[4] M. Hankel and B. Rexroth, "The reference architectural model industrie 4.0 (rami 4.0)," *ZVEI, April*, 2015.

[5] S.-W. Lin, B. Miller, J. Durand, R. Joshi, P. Didier, A. Chigani, R. Torenbeek, D. Duggal, R. Martin, G. Bleakley *et al.*, "Industrial internet reference architecture," *Industrial Internet Consortium (IIC), Tech. Rep*, 2015.

[6] T. G. Ferrari, A. Hinze, and J. Bowen, "An iot for everyone: fact or fiction?" in *Proceedings of the 32nd International BCS Human Computer Interaction Conference*. BCS Learning & Development Ltd., 2018, p. 97.

[7] M. Kim, J. Lee, and J. Jeong, "Open source based industrial iot platforms for smart factory: Concept, comparison and challenges," in *International Conference on Computational Science and Its Applications*. Springer, 2019, pp. 105–120.

[8] J. Guth, U. Breitenbücher, M. Falkenthal, P. Fremantle, O. Kopp, F. Leymann, and L. Reinfurt, "A detailed analysis of iot platform architectures: concepts, similarities, and differences," in *Internet of Everything*. Springer, 2018, pp. 81–101.

[9] M. A. da Cruz, J. J. Rodrigues, A. K. Sangaiah, J. Al-Muhtadi, and V. Korotaev, "Performance evaluation of iot middleware," *Journal of Network and Computer Applications*, vol. 109, pp. 53–65, 2018.

[10] A. A. Ismail, H. S. Hamza, and A. M. Kotb, "Performance evaluation of open source iot platforms," in *2018 IEEE Global Conference on Internet of Things (GCIoT)*. IEEE, 2018, pp. 1–5.

[11] R. L. Dumitru, "Iot platforms: Analysis for building projects." *Informatica Economica*, vol. 21, no. 2, 2017.

[12] T. Kadarina and R. Priambodo, "Monitoring heart rate and spo2 using thingsboard iot platform for mother and child preventive healthcare," in *IOP Conference Series: Materials Science and Engineering*, vol. 453, no. 1. IOP Publishing, 2018, p. 012028.

[13] L. T. De Paolis, V. De Luca, and R. Paiano, "Sensor data collection and analytics with thingsboard and spark streaming," in *2018 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS)*. IEEE, 2018, pp. 1–6.

[14] K. Kreuzer *et al.*, "Openhab-empowering the smart home," *Openhab. org, Tech. Rep.*, 2013.

[15] A. Parchitelli and E. Di Sciascio, "Reflective internet of things middleware-enabled a predictive real-time waste monitoring system," in *Web Engineering: 18th International Conference, ICWE 2018, Cáceres, Spain, June 5-8, 2018, Proceedings*, vol. 10845. Springer, 2018, p. 375.

[16] G. S. Aquino, C. A. Silva, M. Itamir Filho, D. R. Pinheiro, P. H. Lopes, C. A. Barreto, A. P. Silva, R. O. Silva, T. L. Souza, and T. M. Damasceno, "Queuewe: An iot-based solution for queue monitoring," in *International Conference on Computational Science and Its Applications*. Springer, 2017, pp. 232–246.

[17] D. Laney, "3d data management: Controlling data volume, velocity and variety," *META group research note*, vol. 6, no. 70, p. 1, 2001.

[18] A. Labrinidis and H. V. Jagadish, "Challenges and opportunities with big data," *Proceedings of the VLDB Endowment*, vol. 5, no. 12, pp. 2032–2033, 2012.

[19] C.-W. Tsai, C.-F. Lai, H.-C. Chao, and A. V. Vasilakos, "Big data analytics: a survey," *Journal of Big data*, vol. 2, no. 1, p. 21, 2015.

[20] E. Ahmed, I. Yaqoob, I. A. T. Hashem, I. Khan, A. I. A. Ahmed, M. Imran, and A. V. Vasilakos, "The role of big data analytics in internet of things," *Computer Networks*, vol. 129, pp. 459–471, 2017.

[21] A. Gandomi and M. Haider, "Beyond the hype: Big data concepts, methods, and analytics," *International journal of information management*, vol. 35, no. 2, pp. 137–144, 2015.

[22] M. R. Ghazi and D. Gangodkar, "Hadoop, mapreduce and hdfs: a developers perspective," *Procedia Computer Science*, vol. 48, pp. 45–50, 2015.

[23] A. Siddiqa, A. Karim, and A. Gani, "Big data storage technologies: a survey," *Frontiers of Information Technology & Electronic Engineering*, vol. 18, no. 8, pp. 1040–1070, Aug. 2017. [Online]. Available: https://doi.org/10.1631/FITEE.1500441

[24] A. G. Shoro and T. R. Soomro, "Big data analysis: Apache spark perspective," *Global Journal of Computer Science and Technology*, 2015.