Distributed Ledger Architecture for Automation, Analytics and Simulation in Industrial Environments

Mauro Isaja. *. John K. Soldatos **

*** Engineering Ingegneria Informatica, v. S. M. della Battaglia 56, Rome, 00185, Italy (e-mail: mauro.isaja@eng.it)**
**** Athens Information Technology, Kifisias Ave. 44, Marousi, 15125, Greece (e-mail: jsol@ait.gr)**

Abstract: The advent of Industry4.0 has given rise to a large number of digital manufacturing systems, which are currently used to digitize industry and transform industrial processes like automation, maintenance and quality control. The present paper introduces a first-of-a-kind reference architecture for developing industrial automation systems based on edge computing and blockchain technologies. It also presents the design of a platform that implements this reference architecture with a view to providing functionalities in three complementary domains, namely automation, production systems' virtualization and data analytics. The presented platform is destined to provide some distinct performance and reliability advantages, based on its edge computing and blockchain foundations.

Keywords: Industrial Automation, Analytics, BlockChain, Edge Computing, Virtualization

1. INTRODUCTION

The on-going digitization of industrial processes is one of the most important technological trends of our time. This digitization entails the deployment of Cyber-Physical Systems (CPS) in industrial plants, as a means of controlling industrial systems and operations in a fully digital manner. In this context, digital automation platforms have emerged, as a means of automating and controlling plant operations using IT (Information Technology) services, rather than relying on traditional OT (Operational Technology).

1.1 Digital Automation Platforms

For over a decade, several initiatives, have introduced factory automation solutions that aim at increasing the flexibility of industrial automation systems based on various technologies like Intelligent Agents (Rodrigues, 2017) and Service Oriented Architectures (SOA) (e.g., (Quintanilla, 2016), (Borangiu 2017)) and have produced proof-of-concept implementations). Although not widely deployed in manufacturing plants yet, the vision of decentralizing the factory automation pyramid is still alive, as this virtualization can make production systems more flexible and agile. This agility is a key to increasing reducing costs and supporting scalable and fast-configurable lines that enable new production models (e.g., mass-customization).

With the advent of the fourth industrial revolution (Industry 4.0) digital automation solutions are revisited in the light of the integration of CPS Systems within cloud computing infrastructures. Cloud-based applications are deployed and used in factories. They leverage the capacity and scalability of the cloud, while fostering supply chain collaboration and virtual manufacturing chains. However, early cloud implementations have revealed limitations of the cloud in terms of efficient bandwidth usage and its ability to support real-time operations such as operations close to the field.

1.2 Edge Computing in Industrial Automation

To alleviate these limitations, edge computing architectures were recently introduced (Satyanarayanan M., 2017)). Edge architectures comprise layers of edge nodes between the field and the cloud, as a means of:

- Saving bandwidth and storage, as edge nodes can filter data streams from the field to dispose non-valuable information for industrial automation.
- Enabling low-latency and proximity processing, since information can be processed close to the field.
- Providing enhanced scalability, since edge computing supports decentralized storage and processing that scale better.
- Supporting shop floor isolation and privacy-friendliness, since edge nodes deployed at the shop floor can be isolated from the rest of the edge network.

These benefits make edge computing suitable for specific classes of use cases in factories, including large scale distributed applications that process streams from numerous distributed devices and nearly real-time applications (e.g., data analysis close to the field or even control CPS systems).

In the last couple of years, several edge computing platforms for industrial automation are being implemented including:

- The edge intelligence testbed by the Industrial Internet Consortium (IIC) - a proof-of-concept implementation of edge computing functionalities on the plant floor.
- Dell-EMC's EdgeX Foundry framework (EdgeX Foundry, 2017), a vendor-neutral open source project hosted by the Linux Foundation.
- Initiatives from major IT vendors, which extend their existing cloud platforms with edge computing capabilities (e.g., Amazon's AWS is extended with Greengrass edge implementation (Amazon GreenGrass,

2017) while Microsoft Azure is extended with IoT Edge (EdgeX Azure, 2017)).

1.3 FAR-EDGE Vision: Blockchains for Industry

This paper introduces the Reference Architecture (RA) and platform design of the H2020 EC co-funded FAR-EDGE project, whose vision is to research and provide a proof-of-concept implementation of an edge-computing platform for factory automation, that will indicate the potential to increase automation flexibility and lower automation costs, without compromising production time and quality. The FAR-EDGE RA is aligned to the IIC RA, while exploiting concepts from other RAs and standards such as the OpenFog RA and RAMI 4.0 (Reference Architecture Model Industrie 4.0) (Platform Industry 4.0. Smart Manufacturing, RAMI4.0 (2016)). The project is providing one of the first reference implementations of edge computing for factory automation, similar to the ones listed above.

FAR-EDGE is unique, since it investigates the applicability of disruptive key enabling technologies such as: distributed ledger technology (DLT) and smart contracts – better known under the collective label of *Blockchain*. DLT, while being thoroughly tested in areas like digital currencies (e.g., Bitcoin), has never been applied before to industrial systems. FAR-EDGE aims to demonstrate how a pool of specific ledger services built on a generic DLT platform can enable decentralized factory automation in an effective, reliable, scalable and secure way. Ledger services in FAR-EDGE will be responsible for sharing process state and enforcing business rules across the computing nodes of a distributed system, thus permitting virtual automation and analytics processes that span multiple nodes. This is among the project's unique research contributions, which essentially sets it apart from other edge computing efforts worldwide.

The rest of the paper is structured as follows: Section 2 presents and overview of the FAR-EDGE RA, while Section 3 discusses the detailed design of the FAR-EDGE automation platform which is aligned to the RA. Section 4 concludes the paper.

2. FAR-EDGE REFERENCE ARCHITECTURE

2.1 Reference Architecture Scope

The FAR-EDGE Reference Architecture (RA) is the conceptual framework that drives the design and the implementation of the FAR-EDGE Platform. It is described from two architectural viewpoints: functional and structural. Figure 1 provides an overall architecture representation with all elements.

2.2 Functional Viewpoint

According to the RA, the functionality of a factory automation platform can be decomposed into three high-level **Functional Domains**: Automation, Analytics and Simulation, and four **Crosscutting (XC) Functions**: Management, Security, Digital Models and Field Abstraction & Data Routing. These domains map into similar IIRA concepts, even though the scope of the IIRA is much wider.

Functional Domains and XC Functions are orthogonal to **structural tiers**: the implementation of a given functionality may – but is not required to – span multiple Tiers, so that in the overall architecture representation Functional Domains appear as vertical lanes drawn across horizontal layers. A description of the functional domains follows.

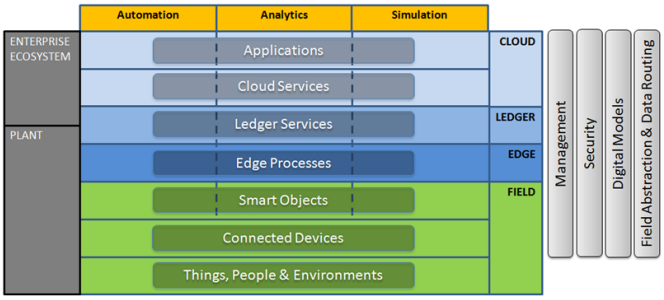

Fig. 1. FAR-EDGE RA Overview

Automation Domain: Includes functionalities supporting automated control and automated configuration of physical production processes. The latter is the enabler of plug-and-play factory equipment, which allows faster and less expensive adjustments of the production process. The Automation domain requires a bidirectional monitoring and control communication channel with the Field, typically with low bandwidth but very strict timing requirements. In some advanced scenarios, Automation is controlled – to some extent – by the results of Analytics and/or Simulation. The Automation domain partially maps to the Control domain of the IIRA. The main difference is that Control is also responsible for decoupling the real word from the digital world, as it includes the functionality for Field communication, entity abstraction, asset management, and modelling. In the FAR-EDGE RA, instead, the Automation domain is only focused on its main role, while auxiliary concerns are dealt with by XC Functions.

Analytics Domain: Includes functionalities for gathering and processing Field data for a better understanding of production processes – i.e., a factory-focused business intelligence. This requires a high-bandwidth Field communication channel. The Analytics domain provides intelligence to its users, not necessarily limited to humans or vertical applications (e.g., a predictive maintenance solution): The Automation and Simulation domains, when properly configured, can both make direct use of the outcome of data analysis algorithms.

The Analytics domain matches perfectly the Information domain of the IIRA, except that the latter is receiving data from the Field through mediation of Control functionalities.

Simulation Domain: Includes functionalities for simulating the behaviour of physical production processes for the purpose of optimization or of testing what/if scenarios at minimal cost and without essential impact on regular shop activities. Simulation requires digital models of plants and processes to be in-sync with the real world objects they represent. For instance, a machine model assumes a given value of electric power / energy consumption, but the actual values will diverge as the real machine wears down. To

detect this gap and correct the model accordingly, raw data from the Field (direct) or complex analysis algorithms (from Analytics) can be used. Model synchronization functionality is not part of the Simulation domain, which acts only as a consumer of the Digital Models XC Functions.

Crosscutting Functions: Their implementation tends to be pervasive, affecting several Functional Domains and Tiers. Specifically, they include:

Management: Low-level functions for monitoring and commissioning/decommissioning of individual system modules. They partially correspond to IIRA's Operations functional domain, with the exclusion of its more high-level functions like diagnostics, prognostics, and optimization.

Security: Functions securing the system against the unruly behaviour of its user and of connected systems, including digital identity management and authentication, access control policy management and enforcement, communication and data encryption. They partially correspond to the Trustworthiness subset of System Characteristics from IIRA.

Digital Models: Functions for the management of digital models and their synchronization with the real-world entities that they represent. Digital modes are a shared asset, used as the basis for automated configuration, simulation and field abstraction. They correspond to the Modelling and Asset Management layers of IIRA's Control functional domain.

Field Abstraction & Data Routing: Functions ensuring the connectivity of business logic (i.e. of the Functional Domains) to the Field, abstracting away the technical details – like device discovery and communication protocols. Data routing refers to the capability of establishing direct producer-consumer channels on demand, optimized for unidirectional massive data streaming – e.g., for feeding Analytics. They correspond to the Communication and Entity Abstraction layers of IIRA's Control functional domain.

2.3 Structural Viewpoints

The FAR-EDGE RA uses two classes of concepts for describing the structure of a system: Scopes and Tiers.

Scopes define a coarse mapping of system elements to either the factory - Plant Scope - or the broader world of corporate IT - Enterprise Ecosystem Scope. Examples of elements in Plant Scope are machinery, Field devices, workstations, SCADA and MES systems, and any software running in the factory data centre.

Tiers are more detailed and technical-oriented. As opposed to scopes, they provide more insight into the relationship between system components. They are described in following paragraphs.

The **Field Tier** lies on the bottom layer of the FAR-EDGE RA and is populated by Edge Nodes (EN): any kind of device connected to the digital world on one side and to the real world to the other. ENs may have embedded intelligence (e.g., a smart machine) or not (e.g., an IoT sensor or actuator); the FAR-EDGE RA honours this difference: Smart Objects are ENs with on board computing capabilities, Connected Devices are those without. The Smart Object is where local control logic runs: it's a semi-autonomous entity that does not need to interact too frequently with the upper layers of the system.

The Field is also populated by entities of the real world – i.e., physical elements of production processes not directly connected to the network, and as such are not considered as ENs: Things, People and Environments. These are represented by some form of EN wrapper. For example, temperature (Environment) is measured by an IoT sensor (Connected Device), a worker's proximity (People) to a physical checkpoint location is published by an RFID wearable and detected by an RFID Gate (Connected Device), and a conveyor belt (Thing) is operated by a PLC (Smart Object).

The Field Tier is in Plant Scope. Individual ENs are connected to the digital world in the upper Tiers either directly by means of the shop floor's LAN, or indirectly through some special-purpose local network. From the RAMI 4.0 perspective, the FAR-EDGE Field Tier corresponds to the Field Device and Control Device levels on the Hierarchy axis (IEC-62264/IEC-61512), while the entities there contained are positioned across the Asset and Integration Layers.

The **Edge Tier** is in Plant Scope, located above the Field Tier and below the Cloud Tier and is the core of the FAR-EDGE RA. It hosts those parts of Functional Domains and XC Functions that can leverage the edge computing model. The Edge Tier is populated by Edge Gateways (EG) i.e. computing devices that act as a digital world gateway to the real world. Strategically positioned close to physical systems, the EG can execute Edge Processes: time- and bandwidth-critical functionality having local scope. For instance, the orchestration of a complex physical process that is monitored and operated by a number of sensors, actuators (Connected Devices) and embedded controllers (Smart Objects); or the real-time analysis of a huge volume of live data that is streamed from a nearby Field source.

Deploying computing power and data storage in close proximity to where it is actually used is a standard best practice, which helps reducing network latency. However, this applies in cases where the scope of individual subsystems is narrow (e.g., a single work station). If instead the critical functionality applies to a wider scenario (e.g., an entire plant or enterprise), it must be either deployed at a higher level (e.g., the Cloud) – thus losing all benefits of proximity – or run as multiple parallel instances, each focused on its own narrow scope. In the latter case, new problems arise: keeping global variables in-sync across all local instances of a given process, reaching a consensus among local instances on a global truth, collecting aggregated results from independent copies of a data analytics algorithm, etc. FAR-EDGE's innovative approach lies in the introduction of a specific system layer – ***the Ledger Tier*** – that implements such mechanisms.

The **Ledger Tier** – a unique approach in FAR EDGE- does not correspond to any physical deployment environment, and the entities it "contains" are abstract. Such entities are Ledger Services, implementing decentralized business logic as smart contracts on top of a distributed ledger. These are transaction-oriented: each service call, which needs to modify the shared state of a system, must be evaluated and approved by Peer Nodes before taking effect. Similarly, to "regular"

services, Ledger Services are implemented as executable code; however, they are not actually executed on any specific computing node: Instead, each service call is executed in parallel by all Peer Nodes that happen to be online at the moment, which then need to reach a consensus on its validity. Most importantly, even the executable code of Ledger Services can be deployed and updated online by means of a distributed ledger transaction, just like any other state change.

Ledger Services implement the part of Functional Domains and/or XC Functions that enable the edge computing model, through providing support for their Edge Service counterpart: i.e., the Analytics Functional Domain may define a local analytics function (Edge Service) to be executed in parallel on several EGs, and also a corresponding service call (Ledger Service) that will be invoked from the former each time new or updated local results become available, so that all results can converge into an aggregated data set. In this case, aggregation logic is included in the Ledger Service. The Ledger Tier lays across the Plant and the Enterprise Ecosystem Scopes, as it can provide support to any Tier.

The **Cloud Tier** is the top layer of the FAR-EDGE RA. "Traditionally" populated by Cloud Servers (CS) running the parts of the business logic of Functional Domains and XC Functions that benefit from having the widest of scopes over production processes, and can deal with the downside of being physically deployed far away from them. This includes planning, monitoring and management of entire factories, enterprises and supply chains (e.g., MES, ERP and SCM systems). CSs implement specialized functions that are provided as individual API calls to Applications, which instead "package" a wider set of related operations relevant to some higher-level goal. The Cloud Tier is in Enterprise Ecosystem scope. Cloud Services and Applications are visible from all Tiers. Most frequently the Cloud Tier corresponds to one or more corporate data centres (private cloud), ensuring that the entire system is fully under the control of its owner.

3. FAR-EDGE PLATFORM DESIGN

In this paragraph we provide design details for each individual Tier of the FAR-EDGE platform, which adheres to the RA presented in the previous section.

3.1 Field Tier

The Field Tier is the realm of Edge Nodes (EN), which connect the real world to the digital world of IT – and vice versa. We focus on ENs that have enough computing capabilities on board to be active actors in a CPS i.e. intelligent devices that are conveniently called "Smart Objects", ranging from a simple PLC to a fully-autonomous smart plant. Typically, Smart Objects are controller boards with some significant processing power and a good network connection, but lack any (usable) local storage. The Platform defines two Components that can be run on such a device, namely Policy Decision Point (PDP) and Ledger Clients.

The **Policy Decision Point (PDP)** Component implements parts of the Security XC Functions related to the enforcement of access policies. It supports Smart Objects in assessing incoming calls against security policies that are defined and maintained centrally, approving their execution only if they are legitimate.

The **Ledger Clients Component** is a software library that can be used by Smart Object logic to interact with Ledger Services in an easy way. It allows Smart Object logic to make outgoing calls to specific Ledger Services. The library contains a dedicated client for each Ledger Service. Each Ledger Client Component maps to all three Functional Domains of the RA, and to the Management and Digital Models XC Functions.

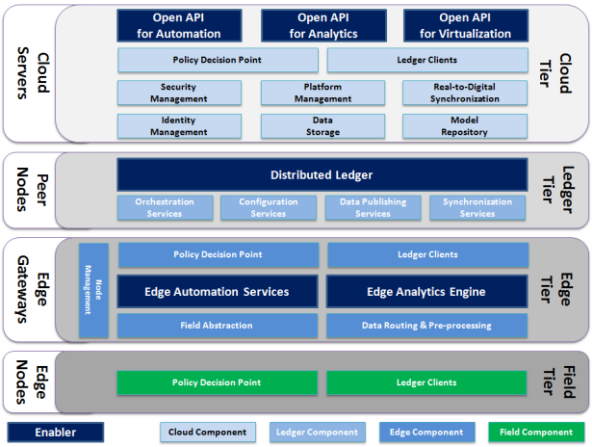

Fig. 2. FAR-EDGE Platform Design

3.2 Edge Tier

At the Platform level, edge computing is supported by seven Components, two of them with Enabler status: Edge Automation Services, Edge Analytics Engine, Field Abstraction, Data Routing & Pre-processing, Policy Decision Point, Ledger Clients and Node Management. All Components are meant to be deployed on Edge Gateway (EG) machines. Specifically:

The **Edge Automation Services Enabler** implements the core part of the Automation Domain, as it defines a runtime environment for local automation workflows. The environment provides core services like security and a local registry of Field device endpoints. Such services and their interfaces comply with documented FAR-EDGE Platform standards to allow edge automation workflows developed on a given Platform to be easily migrated to another.

The **Edge Analytics Engine Enabler** implements the core part of the Analytics Domain, as it defines a runtime environment for local data analysis algorithms. The environment includes both an execution engine – i.e., a generic container for running data processing logic – and southbound/northbound interfaces for extracting raw Field data and sending processed results to the Platform, respectively.

The **Field Abstraction Component** implements part of the Field Abstraction & Data Routing XC Functions. Its role is to decouple the Field from the upper layers of the Platform, hiding the complexity of different communication protocols and data models. It only deals with events, commands and data messages, exchanged in both directions with a minimum of latency; it does not deal with data streams or any kind of massive data flow, as such concerns belong to Data Routing.

The on-going FAR-EDGE implementation supports the OPC UA and the MQTT field communication protocols.

The **Data Routing & Pre-processing Component** implements part of the Field Abstraction & Data Routing XC Functions. Its role is to forward data generated by Field sources to the appropriate processor in the most efficient way, while hiding technical complexities to both sides of the channel and also allowing for some basic pre-processing of data – e.g., filtering, in order to save network resources, when the final destination of the information flow is in the Cloud. As opposed to Field Abstraction, this task is less concerned with time constraints and more with data throughput.

The **Policy Decision Point (PDP) Component** of the Edge Tier implements those parts of the Security XC Functions related to the enforcement of security policies. It is not constrained by low-end computing resources, as is the case with the Field Tier version. Its users are those service points exposed by Components and Enablers running on EGs.

The **Ledger Clients Component** is a software library that can be used by application logic running on EGs – e.g., automation workflows or data analysis algorithms – to interact with Ledger Services in an easy way, ignoring the technicalities of establishing a network connection and managing transactions. It provides the same functionality as the Component with the same name in the Field Tier, but with the additional capability of subscribing to ledger events. As for the Field-level Component, a dedicated client for each Ledger Service is provided. For this reason, this Component maps to all three Functional Domains of the RA.

The **Node Management Component** implements some technical hooks – e.g., service and/or subscription endpoints – that allow for the remote management of EGs. This functionality implements the Management XC Functions at the Edge level and is entirely implementation-dependent.

3.3 Ledger Tier

The Ledger Tier is where the full potential of edge computing is unleashed. It allows for truly distributed process logic to work in use cases that require some level of coordination – as is true for most industrial scenarios – without any centralized service being in charge of it. It employs the distributed ledger and smart contract patterns - both key elements of Blockchain technologies. We stress the term "logical" as the FAR-EDGE Platform architecture does not mandate any specific deployment choice for Peer Nodes (PN) – i.e., the physical server machines that implement both the distributed ledger and the smart contracts. These can be installed in environments belonging to the Cloud or Edge Tiers, possibly even to the Field in some cases. PNs run the Distributed Ledger Enabler and the Orchestration Services, Configuration Services, Data Publishing Services and Synchronization Services Components.

The **Distributed Ledger Enabler** is the foundation of the Ledger Tier. It provides a transactional ledger that is replicated on all PNs and automatically kept in-sync across all instances, using peer-to-peer communication only. The ledger stores and maintains the shared state of distributed processes: global variables and even complex data structures that are read and written concurrently by business logic running on any Tier of the FAR-EDGE Platform – typically on EGs. Such state is secured by cryptographic techniques, so that records in the ledger cannot be modified or removed once created, unless there is a system-wide unanimous consensus on doing that. A majority consensus of PNs is also required for the creation of every new record, which constitutes an atomic transaction. The logic for assessing the validity of transactions is itself defined by consensus of all stakeholders: it is a smart contract – i.e., executable code that implements a deterministic algorithm – deployed on the ledger. This means that the coordination of distributed processes is not only decentralized, but also dynamically (re)definable according to changing needs.

Note that distributed ledgers are not a good fit for cases involving storage of massive amounts of data or run of very complex, long-running business logic. The ideal clients perform frequent reads and infrequent writes of the shared state, the latter based on fast and simple logic.

The **Orchestration Services Component** includes all the smart contracts that support the Automation Domain functionalities related to distributed automation workflows. These smart contracts are scenario- or even use case-specific, as they implement ad-hoc business logic and ad-hoc shared data models. The common trait is that they expose one or more service endpoints for distributed processes to call on need. Typically, clients will be local automation workflows executed by EGs, and possibly Cloud-based applications that need to exercise control over them.

The **Configuration Services Component** includes all the smart contracts supporting the Automation Domain functionalities related to self-adjustment and reconfiguration, and those related to the management of a global asset/service registry in the Management XC Functions. All smart contracts are scenario- or even use case-specific, as they implement ad-hoc business logic and may be based on ad-hoc Plant and CPS models. They expose one or more service endpoints for clients to call on need.

The **Data Publishing Services Component** includes all the smart contracts that support the Analytics Domain functionalities and depends on the Distributed Ledger Enabler. These smart contracts are scenario- or even use case-specific, as they are used to collect/aggregate results from ad-hoc distributed analytics processes running on EGs. They expose service endpoints for clients to call on need i.e. the availability of new or updated local results on any EG, that must be contributed to the global view in order to keep it current. For instance, a standard set of Key Performance Indicators (KPI) may be measured locally on each EG, so that raw Field data is not transmitted on the network beyond the narrow boundaries of a plant's LAN. Each time a local KPI value changes, this change is submitted to the relevant Data Publishing service endpoint and the underlying smart contract will compute the new value of the corresponding global KPI.

The **Synchronization Services Component** includes all the smart contracts that support the Digital Models XC Functions. These smart contracts are scenario- or even use case-specific, as they are used to keep digital models in-synch with changes in the real world they represent. As for all Ledger Services, they expose one or more service endpoints for clients to call upon changes in the real world.

3.4 Cloud Tier

The Cloud Tier includes a total of eleven components, three of them having **Enabler status**. They are classified in: (i) **Public Cloud Services**, which are accessible through three Open APIs, namely the Open API for Automation, the Open API for Analytics and the Open API for Virtualization; (ii) **Internal Cloud Services**, including Identity Management, Model Repository and Data Storage; and (iii) **Platform Cloud Tools**, including security Management, PDP, Ledger Clients, Real-to-Digital Synchronization and Platform Management components. An explanatory analysis of the above follows:

The **Open API for Automation Enabler** implements all the public cloud service endpoints belonging to the FAR-EDGE Automation Domain, accessed over the Web through HTTPS. It enables access to Edge Automation Services, Orchestration Services and Configuration Services.

The **Open API for Analytics Enabler** implements all the public cloud service endpoints belonging to the Analytics Domain. These endpoints allow external systems to interact with the distributed data analysis logic deployed on the Edge Tier (Edge Analytics Engine) and on the Ledger Tier (Data Publishing Services).

The **Open API for Virtualization Enabler** implements the public cloud service endpoints, belonging to the Simulation Domain, related to the integration of external simulation tools. It also implements the public cloud service endpoints belonging to the Digital Models XC Functions.

The **Identity Management (IDM) Component** implements, the Security XC Functions that relate to digital identities. It is in charge of exposing single-sign-on service endpoints for other Platform services.

The **Model Repository Component** implements those low-level parts of the Digital Models XC Functions that are not exposed to external systems.

The **Data Storage Component** implements generic persistence for the purpose of internal sharing of data.

The **Security Management Component** implements those parts of the Security XC Functions related to the administration of access policies.

The **Policy Decision Point Component** implements those parts of the Security XC Functions that relate to the enforcement of access policies.

The **Ledger Clients Component** is used by the three Open API Enablers to access Ledger Services.

The **Real-to-Digital Synchronization Component** implements a specific subset of Digital Models XC Functions related to the Simulation Domain i.e. keeping digital models in-sync with changes happening in the real world.

The **Platform Management Component** implements the Cloud part of the Management XC Functions as a platform cloud tool.

4. CONCLUSION AND OUTLOOK

Earlier paragraphs provided a comprehensive overview of the FAR-EDGE RA, which outlines the building blocks required for the developing digital automation platforms that leverage the capabilities of edge computing and blockchains for industry. Moreover, the paper has also outlined the design of a platform that adheres to the FAR-EDGE RA, including details about the various components that it comprises and the ways they interact between them. The implementation of this platform is on-going and major parts are already made available through an open source license in GitHub (under https://github.com/far-edge/). Early releases of the edge analytics, data routing components and ledger services are already available as open source software components. Readers are advised to treat these implementations that realized the indicated platform design as work in progress. We expect our reference implementation to provide a sound basis for further research and experimentation on edge computing and blockchains for digital automation in industry, while at the same time boosting relevant pilots and implementations.

5. ACKNOWLEDGEMENTS

This work has been carried out in the scope of the FAR-EDGE project (H2020-703094). The authors acknowledge help and contributions from all partners of the project.

REFERENCES

Amazon GreenGrass (2017), https://aws.amazon.com/greengrass/

Gilchrist A. (2016) IIoT Reference Architecture. In: Industry 4.0. Apress, Berkeley, CA.

Borangiu, T., Silisteanu, A., Raileanu, S., Voinescu, I. (2017). Service Orientation of Environment Control Processes, Exploring Service Science IESS17, Lecture Notes in Business Information Processes vol. 279, p. 383-396, doi: 10.1007/978-3-319-56925-3_30.

EdgeX Azure (2017), https://azure.microsoft.com/en-us/services/iot-edge/

EdgeX Foundry (2017), https://www.edgexfoundry.org/

Industrial Internet Consortium (IIC) (2017) *Why We Build Testbeds: First Results* http://www.iiconsortium.org/pdf/Why_we_build_testbeds-first_results_091917.pdf

Platform Industry 4.0. Smart Manufacturing (2016) – Reference Architecture Model Industry 4.0 (RAMI4.0). 1–35, (2016)

Quintanilla, F.G., Cardin, O., L'Anton, A., Castagna, P. (2016). A modeling framework for manufacturing services in Service-oriented Holonic Manufacturing Systems, Engineering Applications of Artificial Intelligence, Volume 55, October 2016, Pages 26-36, doi: 10.1016/j.engappai.2016.06.004

Rodrigues, N., Leitao, P., Oliveira, E. (2017). Dynamic Service Reconfiguration with Multi-agent Systems, Service Orientation in Holonic and Multi-Agent Manufacturing, Studies in Computational Intelligence 694, p. 307-318, Springer, doi: 10.1007/978-3-319-51100-9-27.

Satyanarayanan M. (2017) The emergence of edge computing *Computer (Long. Beach. Calif)*, 50 (2017), pp. 30-39