2017 21st International Conference on Control Systems and Computer Science

Blockchain based distributed control system for Edge Computing

Alexandru Stanciu
National Institute for Research and Development in Informatics
Bucharest, Romania
alex@ici.ro

Abstract—**Edge computing proposes a novel model for providing computational resources close to end devices that are connected to the network. It has numerous applications in Internet of Things, as well as smart grids, healthcare, smart home, etc. This paper presents ongoing research regarding the use of blockchain technology as a platform hierarchical and distributed control systems based on IEC 61499 standard. Hyperledger Fabric was selected as the blockchain solution, where function blocks are to be implemented as smart contracts on a supervisor level. The integration with the edge nodes that perform on the executive level responsible for actual process control is based on a micro-services architecture where Docker containers implement function blocks, and Kubernetes platform is used for orchestrating the execution of containers across the edge resources.**

Keywords—edge computing; IoT; blockchain; smart contracts; hierarchical distributed control system

I. Introduction

The emergence of Internet connected smart devices which can be accessed and controlled remotely via computer networks, has raised the expectations for new and enhanced intelligent computing services.

However, security and privacy of user's data, and specific requirements regarding personal data protection, have challenged the effectiveness of the centralized computing model available on the public cloud infrastructures. A new computing model, with the emphasis on decentralization, and the collaboration of individual work units to achieve a common goal, has been devised for Internet of Things (IoT) applications, and for other domains such as the smart grid, healthcare, connected vehicles, etc.

The main characteristic of this model is that data processing occurs at the edge of the network, as close as possible to the smart device which both produce and consume the data, and the need of powerful computing resources is limited [1].

Edge computing is a new paradigm dedicated to applications that require minimal latency, and which must process large quantity of data that is difficult to transfer over the network. As the cloud services are not suitable for such use cases because of the limited network bandwidth, security and privacy of data, various solutions have been implemented to address computing needs at the edge of the network.

There are similar concepts which have some overlap with edge computing, like mobile cloud computing, which is expected to have an external infrastructure (usually available as cloud services) for data processing and storage for mobile applications.

Another term used interchangeably with edge computing is fog computing. It extends the cloud computing paradigm to the edge of the network, thus enabling a new breed of applications and services. Its main characteristics are: a) low latency and location awareness; b) wide-spread geographical distribution; c) mobility; d) very large number of nodes, e) predominant role of wireless access, f) strong presence of streaming and real time applications, g) heterogeneity. These characteristics make the fog the appropriate platform for a number of critical Internet of Things (IoT) services and applications, namely, connected vehicle, smart grid, smart cities, and, in general, wireless sensors and actuators networks [2].

We have investigated the blockchain technology as a platform for edge computing in order to implement a distributed control system based on IEC 61499 standard.

The rest of the paper is organized as follows. Section II presents an overview of the edge computing architecture. IEC 61499 standard for distributed control system is reviewed in Section III. In Section IV is discussed the blockchain technology, and the current research regarding implementation of function blocks as smart contracts and the integration within the distributed control system. Related work is discussed in Section V, and the conclusion are presented in the final section.

II. Edge computing architecture overview

For many IoT applications which require mobility support, location awareness and low latency, there is a need of a new platform, one which can provide computational resources to both large-scale sensor networks which monitor the environment, as well as intelligent services based on data processing and cloud resources integration.

Edge computing nodes can be seen as members of a decentralized network which provides compute, storage and networking services to end devices.

Since smart devices are usually inadequate in computation power, battery, storage and bandwidth, IoT applications and services are usually backed up by strong server backends, which

DOI 10.1109/CSCS.2017.102

are mostly deployed in the cloud, since cloud computing is considered as a promising solution to deliver services to end users and provide applications with elastic resources at low cost [3].

Edge computing has not replaced the need for cloud computing services. Sometimes, it is necessary to integrate cloud resources in a three-layer architecture which is based on devices, a mesh of edge nodes, and has cloud services on top. It has several properties, such as its close distance to end users.

For latency sensitive applications, the computations must be kept as close to the data as possible. Location-awareness is another important property because intelligent services need to use the context of the user to provide the best experience. For data intensive applications, edge nodes can provide the first steps for data processing, thus limiting the volume of data that should be transferred to the central cloud services.

It can be argued that while the edge nodes provide localization with context and low latency for data processing, the cloud provides a central point of references which can coordinate the edge nodes.

This three-tier model for edge computing is necessary especially for data intensive applications by reducing the data transfer and storage requirements. Data processing in the edge nodes is thus extremely important for applications that produce huge volumes of data, and which should provide real-time response to end users.

Another interesting application domain for edge computing is the smart grid. In order to measure and control the distribution network, smart meters are deployed on the field. There is a central system to control the grid infrastructure which gathers and analyses the data received, and sends commands to adjust any modifications with respect to supply and demand of resources.

With edge computing, this central automation system, can be integrated with a network of intelligent devices, in a hierarchical control structure. This covers the geographical distribution of the power grid and provides advanced services based on integration between the central and local resources. The central control structure on the higher level has a strategic role, whereas the local control systems on the lower level are responsible for the real-time operation of field resources.

This type of role separation in a hierarchical distributed control system is used in the context of edge computing model as a criterion for the selection of the level on which its components are deployed. For example, as presented in Fig. 1, the principal components that are involved in strategy decision are placed on the higher level, and are implemented as smart contracts in a blockchain provided as a cloud service. The rest of the components are part of the lower level of the hierarchical distributed control system, and are implemented on the edge nodes (which are close to the processes to be controlled).

We will review in the next section the IEC 61499 standard for distributed control systems, and will present our ongoing work related to the usage of blockchain technology in order to execute a critical part of distributed control algorithms in a secure environment.

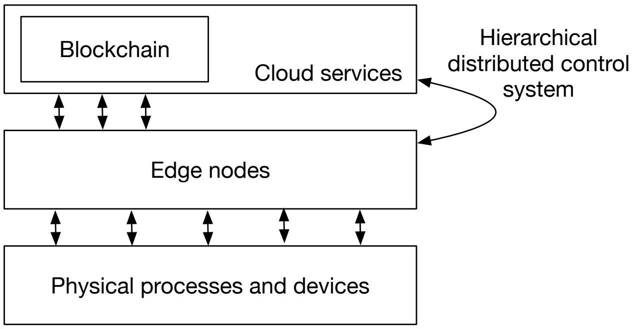
Blockchain
Cloud services
Hierarchical distributed control system
Edge nodes
Physical processes and devices

Fig. 1. Hierarchical distributed control system model for edge computing

III. IEC 61499 STANDARD FOR DISTRIBUTED CONTROL SYSTEMS

The IEC 61499 standard outlines a generic model for distributed control systems, defining the architecture and the compliance requirements for software tools.

The promising application areas of IEC 61499 include flexible material handling systems, in particular airport baggage handling, flexible reconfigurable manufacturing automation, intelligent power distribution networks and smart grid, as well as the wide range of embedded networked systems.

There were various attempts to create distributed control systems, first by using Programmable Logic Controllers (PLCs) to centrally process collected data from field area networks. After that, it was experimented the integration of PLCs via networks, and then, by genuinely distributed automation development, where the intelligence is designed from the very beginning as decentralized and embedded into software components, which can be freely distributed across networked hardware devices [4].

The IEC 61499 standard has been designed to support distributed automation control systems. The main component is the function block (FB), which has been designed like a process abstraction which is used in distributed computing systems - an independent computational activity with its own set of variables (context) and communication with other processes via messages.

Function blocks which are the atomic units of execution in IEC 61499 based systems, consists of two parts, a function block interface and an execution control chart that operates over a set of events and variables. The execution of a FB entails accepting inputs from its interface, processing the inputs using the algorithms selected by the execution control chart, and emitting outputs.

Typically, an algorithm consists of loops, branching and update statements, which are used to consume inputs and generate outputs. The IEC 61499 standard allows algorithms to be specified in a variety of implementation-dependent languages.

Another function block type is the Service Interface Function Block (SIFB). This represents the interface to low level services provided by the operating system or hardware of the embedded device, such as:

- Graphical User Interface (GUI) elements such as a input fields and controls,
- Communication services,
- Interfaces to hardware devices.

Service interface function blocks (SIFB) can be considered as device drivers that connect the external environment with function block applications. IEC 61499 compliant software tools and their associated runtime packages can provide a large selection of GUI and communications SIFBs.

Device and resource models are used in the IEC 61499 standard to emulate the physical components (e.g. controllers, sensors, actuators, etc.) as logical elements of the process automation system. A conceptual model of the main entities and the relationship between them is presented in Fig. 2.

A device model is the functional definition of a physical component in a larger distributed system. Each device may contain some inherent behaviour owing to its physical subcomponents. In order to manage the complexity of devices, the concept of resource models is used. A device may contain zero or more resources encapsulating independent function or tasks.

A resource represents an independent task executing on a device. Such tasks are segregated from each other in such a way that a particular system resource (e.g. a sensor or an actuator) may only be accessed and operated upon by a single resource. Due to the absence of shared variables, resources and devices communicate using communication function blocks in order to perform the coordination between tasks.

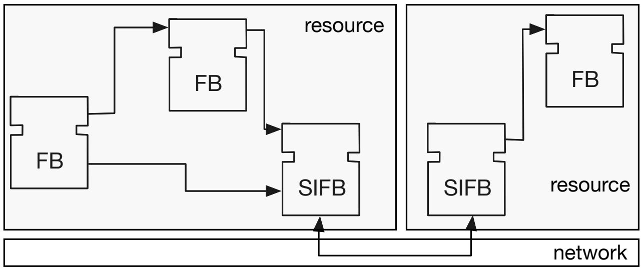

Fig. 2. A basic overview of IEC 61499 distributed control systems

Each device is capable of performing a set of different tasks that coordinate by means of a communication network and, thus, constitute a distributed system [5].

IV. Using Blockchain beyond Bitcoin transactions

A blockchain is essentially a distributed database of records, or public ledger of all transactions or digital events that have been executed and shared among participating parties. Each transaction in the public ledger is verified by consensus of a majority of the participants in the system. Once entered, information can never be erased. The blockchain contains a certain and verifiable record of every single transaction ever made. Bitcoin, the decentralized peer-to-peer digital currency, is the most popular example that uses blockchain technology [6].

Blockchains allow us to have a distributed peer-to-peer network where non-trusting members can interact with each other without a trusted intermediary, in a verifiable manner.

Within the blockchain context, smart contracts are scripts stored on the blockchain. (They can be thought of as roughly analogous to stored procedures in relational database management systems). Since they reside on the chain, they have a unique address.

A smart contract can be triggered by addressing a transaction to it. It then executes independently and automatically in a prescribed manner on every node in the network, according to the data that was included in the triggering transaction. (This implies that every node in a smart contract enabled blockchain is running a virtual machine (VM), and that the blockchain network acts as a distributed VM) [7].

A. Hyperledger Fabric

Hyperledger Fabric (*github.com/hyperledger/fabric*) is an implementation of a distributed ledger platform for running smart contracts, leveraging familiar and proven technologies, with a modular architecture allowing pluggable implementations of various functions. It is one of multiple projects currently in incubation under the Hyperledger Project.

The distributed ledger protocol of the fabric is run by peers. The fabric distinguishes between two kinds of peers: A validating peer is a node on the network responsible for running consensus, validating transactions, and maintaining the ledger. On the other hand, a non-validating peer is a node that functions as a proxy to connect clients (issuing transactions) to validating peers. A non-validating peer does not execute transactions but it may verify them.

The Hyperledger Fabric is a permissioned blockchain platform aimed at business use. It is open-source and based on standards, runs user-defined smart contracts, supports strong security and identity features, and uses a modular architecture with pluggable consensus protocols [8].

B. Experimental testbed specifications

The use case that is the subject of our investigation on the topic of distributed control systems is related to the PID control of dynamic systems.

The three-tier edge computing model can be used to design the components of the PID controller on the lower level, and include on the higher-level elements that are responsible for the selection of algorithms and methods dedicated to the fine tuning of the controller parameters.

In order to experiment the execution of a distributed control system based on IEC 61499 standard on edge computing nodes an adapted environment from the Calculos project [9] was proposed. The aim of Calculos was to design and implement an open platform for cloud services which can provide modeling and optimization with complex algorithms standardized as function blocks, according to the IEC 61499 model, in order to be compatible, and usable in any control system.

The high-level architecture of the distributed control system is presented in Fig. 3. As Function Blocks are implemented in

Docker containers, Kubernetes platform is used for orchestrating the execution of containers across the edge nodes.

Each FB has its control algorithms and execution logic implemented in a Python program which is packaged together with its dependencies in a container image. FBs communicate (events and data) using a publish/subscribe mechanism via a broker which is based on Redis message broker.

IEC 61499 resources which can contain several FBs are implemented as Pods - a Kubernetes specific abstraction dedicated to a group of linked containers, where inside a Pod, each container can access any other containers.

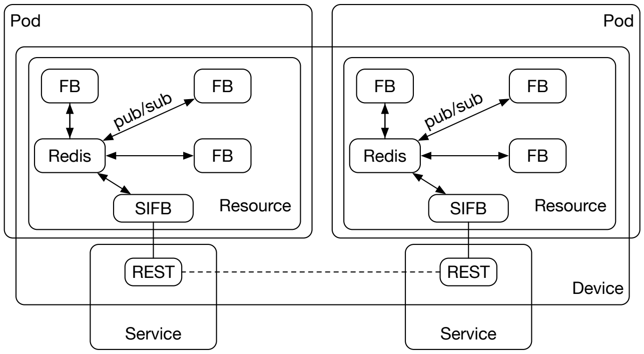

Fig. 3. High level architecture of IEC 61499 components deployed as Docker containers with Kubernetes platform

The process interface and the user interface of the IEC 61499 control system are designed as REST APIs implemented by Service Interface Function Blocks (SIFB). An IEC 61499 Device contains one or more Resources which should be able to communicate among each other via SIFBs.

Kubernetes allows the definition of services as objects which expose Pods resources. These Kubernetes services provide the REST interface to Resources, and they are the basis for a micro-service architecture. Each Resource implements a specific functionality of the distributed control system, and all the Resources are loosely coupled in a message based architecture.

The Docker containers are deployed on edge nodes in addition to Hyperledger Fabric validating nodes. This architecture emulates the three tier edge computing model: physical devices and processes, edge nodes and cloud services. The blockchain is deployed on the top level to ensure that transactions are secured and properly validated. The blockchain can execute smart contracts which are implemented in Go language, and thus the function blocks are designed as smart contracts that enforce the user decisions concerning high-level strategies employed in the process automation system.

One very important aspect which must be considered is that smart contracts cannot access outside data, such as remote APIs or services. In order to be able to achieve consensus among blockchain nodes, the execution must be deterministic.

The workaround this problem is based on special decentralized blockchain application which are used to push data into the blockchain, so that other smart contracts can consume it, and transfer data to the outside world. These special applications are the equivalent of the SIFB in the IEC 61499 standard.

We have evaluated the performance of Hyperledger Fabric v0.6 regarding the number of transactions (invoke and query transaction types) executed per second (tps). For this benchmarking, we have used the following Google Cloud Platform resources:

- *n1-standard-4*, a machine type with 4 vCPUs and 15 GB memory, which was able to execute 186 tps (query transactions) and 291 tps (invoke transactions).
- *n1-standard-2*, a machine type with 2 vCPUs and 7.5 GB memory, which executed 170 tps (query transactions) and 255 tps (invoke transactions).

The performance benchmarking was performed with hyperledger-py tool against a cluster of 4 validating nodes that use Practical Byzantine Fault Tolerance consensus.

These numbers show that higher level of the distributed control system which is based on the blockchain technology has a clear limitation regarding the load that can be effectively processed in real time. However, as the role of this layer is to monitor and supervise the lower level, it is reasonable to expect that number of function block that are executed as smart contracts to be a small fraction of the total number of the function blocks that compose the automation system.

In addition, as the strategic decisions are based on the evaluation of expected performance and related costs, the importance of a secure medium to keep and execute these transactions is underlined by the possibility to automatically enforce operational rules specified by users, and encoded in smart contracts.

V. Related work

Moving IoT components from the cloud onto edge hosts helps in reducing overall network traffic and thus minimizes latency. However, provisioning IoT services on the IoT edge devices presents new challenges regarding system design and maintenance. One possible approach is the use of software defined IoT components in the form of virtual IoT resources. This, in turn, allows exposing the thing/device layer and the core IoT service layer as collections of micro services that can be distributed to a broad range of hosts. In [10] was investigated the concept of software-defined IoT components called virtual resources and the use permission-based blockchains as a means for distribution.

A new secure, private, and lightweight architecture for IoT, based on blockchain technology that eliminates the overhead while maintaining most of its security and privacy benefits was investigated on a smart home application as a representative case study for broader IoT applications. The proposed architecture was hierarchical, and consists of smart homes, an overlay network and cloud storages coordinating data transactions with blockchain to provide privacy and security [11].

An approach for developing a campus-wide sensor network using commodity single board computers (Raspberry Pi) was presented in [12]. Edge computation is made of per-node event triggers, that are defined and monitored on each sensor device directly. Further compute tasks might include local aggregation of data such as map/reduce or per-node user queries executing on-device.

A framework supporting context-aware sensing, computing and communication capabilities into industrial applications was introduced in [13]. The solution provides context information sensing and processing, an access mechanism for interfacing sensor networks with IoT and cloud services and a four-level architecture to perform industrial process control that includes the modules of a context-aware control platform.

In [14] was presented an evaluation of Docker as an edge computing platform based on four requirements: deployment and termination; resource & service management; fault tolerance and caching. Due to its small footprint, good performance and fast deployments, it was assessed that Docker could be a viable Edge computing platform.

VI. Conclusions

Edge computing proposes a novel model for providing computational resources close to end devices that are connected to the network. It has numerous applications in Internet of Things, as well as smart grids, healthcare, smart home, etc. One important feature of edge computing is that the centralized computing model based on cloud services is augmented with a decentralized network of nodes that create an intermediate layer between the sensors and devices and cloud services. For data intensive applications, it has the benefit of reduced data transfers and increased responsiveness.

For many IoT applications, a distributed automation system can be implemented as a hierarchical structure with two tiers, with the higher level performing supervision and strategic decisions, and the lower level having direct control of devices and processes.

We have investigated the IEC 61499 standard for distributed control systems, and have presented the ongoing research regarding the implementation of function blocks as smart contracts executed by the blockchain on a supervision level, as well as the integration with the edge nodes that perform the executive level responsible for process control.

Acknowledgment

This work was partly supported by the Romanian National Research Programme PNII project “Cloud Architecture for an open Library of Complex re-Usable Logical function blocks for Optimized Systems – CALCULOS”, and by Nucleu project PN 16090401 “Evaluation and experimentation of Internet of Things platforms”.

References

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal,* vol. 3, no. 5, pp. 637-646, 2016.

[2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, 2012, pp. 13-16: ACM.

[3] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *Hot Topics in Web Systems and Technologies (HotWeb), 2015 Third IEEE Workshop on*, 2015, pp. 73-78: IEEE.

[4] V. Vyatkin, "IEC 61499 as enabler of distributed and intelligent automation: State-of-the-art review," *IEEE Transactions on Industrial Informatics,* vol. 7, no. 4, pp. 768-781, 2011.

[5] L. H. Yoong, P. S. Roop, Z. E. Bhatti, and M. M. Kuo, *Model-Driven Design Using IEC 61499*. Springer, 2015.

[6] M. Crosby, P. Pattanayak, S. Verma, and V. Kalyanaraman, "Blockchain technology: Beyond bitcoin," *Applied Innovation,* vol. 2, pp. 6-10, 2016.

[7] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," *IEEE Access,* vol. 4, pp. 2292-2303, 2016.

[8] C. Cachin, "Architecture of the Hyperledger blockchain fabric," in *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, 2016.

[9] O. Chenaru, G. Florea, A. Stanciu, V. Sima, D. Popescu, and R. Dobrescu, "Modeling Complex Industrial Systems Using Cloud Services," in *Control Systems and Computer Science (CSCS), 2015 20th International Conference on*, 2015, pp. 565-571: IEEE.

[10] M. Samaniego and R. Deters, "Using Blockchain to push Software-Defined IoT Components onto Edge Hosts," in *Proceedings of the International Conference on Big Data and Advanced Wireless Technologies*, 2016, p. 58: ACM.

[11] A. Dorri, S. S. Kanhere, and R. Jurdak, "Blockchain in internet of things: Challenges and Solutions," *arXiv preprint arXiv:1608.05187,* 2016.

[12] K. Hentschel, D. Jacob, J. Singer, and M. Chalmers, "Supersensors: Raspberry Pi Devices for Smart Campus Infrastructure," in *Future Internet of Things and Cloud (FiCloud), 2016 IEEE 4th International Conference on*, 2016, pp. 58-62: IEEE.

[13] D. Merezeanu, G. Vasilescu, and R. Dobrescu, "Context-aware Control Platform for Sensor Network Integration in IoT and Cloud," *STUDIES IN INFORMATICS AND CONTROL,* vol. 25, no. 4, pp. 489-498, 2016.

[14] B. I. Ismail *et al.*, "Evaluation of docker as edge computing platform," in *Open Systems (ICOS), 2015 IEEE Confernece on*, 2015, pp. 130-135: IEEE.