

Using Blockchain in the Internet of Things Coordination

A Smart Contract-based Approach

Radia Belkeziz, Zahi Jarir

Information Systems Engineering Laboratory
Faculty of Sciences- Cadi Ayyad University
Marrakech, Morocco

Abstract—Now-a-days, the Internet of Things (IoT) has generated enormous interest from industry to create distributed and innovative solutions. However, achieving this goal is a tedious task and presents several open challenges as the literature points out. One of the most complex is the IoT coordination service. Unfortunately, most research works give rarely importance to this service in their models or architectures proposals. Wherefore our current contribution deals with this open issue and proposes a solution capable of implementing advanced processes that can be based on orchestration, choreography or both mechanisms. More over and to conduct efficiently both coordination mechanisms when sharing knowledge or tasks between connected objects, we integrate smart contacts to guarantee the modalities of behavior change in the coordination mechanism. Smart contracts are a safe way to decide the coordination mechanism based on the state of the system environment. To prove our approach, we have built a technical architecture based on a multi-agent system to abstract connected objects of IoT systems, blockchain technology, and the frameworks and languages required for collaboration processes such as BPMN, BPEL and BPEL4CHOR. Carbon leakage as a case study is used for experimentation.

Keywords—Internet of things; IoT; Internet of things coordination; blockchain; smart contract; multi-agent systems

I. INTRODUCTION

Internet of things (IoT) defines a network of dedicated physical objects (things) that contain embedded technology to communicate and sense or interact with their internal states or the external environment. Based on such technology the connecting of assets and processes enables the capture of data and events from which an organization can learn behavior and usage, react with preventive action, or augment or transform business processes. Consequently, IoT provides a foundational capability for the creation of a new digital business [1].

However, the development of IoT has encountered challenges that are necessary to meet. Among these challenges, coordination represents the core of any IoT system as it consists of organizing things, objects, information, tasks, functionalities, services, in order to enable them to work together efficiently to attain a required and desired objective. Furthermore, other challenges are unfortunately closely linked to coordination, which makes its achievement more complex. One of the major obstacles facing IoT coordination is the lack

of flexible architecture, and that motivates us to propose our vision to build a suitable architecture [2].

Coordination needs an entire organization to be handled, as it requires understanding and following the building blocks of IoT: identification, data acquisition, data processing, interpretation/semantics, and decision-making. This has guided us towards a system that requires good management of connected objects, and infrastructure to manage data: storage, processing, and analysis, which will lead to appropriate decision-making.

In this paper, we present an architecture based on multi-agent system (MAS) and smart contracts for coordination processes. This architecture consists of three main layers: meta-workflow layer, coordination layer, and object layer. The meta-workflow represents the business part where the main workflow is modeled using BPMN. The coordination layer is made up of three sub-layers: agent abstraction, data management, and coordination mechanism. The agent abstraction sub-layer represents agents that abstract physical objects. As coordination consists in organizing the cooperation of connected objects by sharing knowledge, MAS is most suitable to achieve this purpose as agents rely on each individual's capacities and knowledge. The coordination mechanism sub-layer defines the description of the mechanisms, namely orchestration and choreography is to take on. The data management sub-layer counts the system's policies on gathered data and the smart contracts. Policies are the rules that make it possible to decide on the coordination mechanism according to conditions through the data gathered from the environment. Smart contracts make it possible to ensure coordination mechanisms used and define the actions to undertake.

Generally, coordination is approached from the point of view of services. The actions and interactions of objects are broken down into several independent services. Service-oriented architecture (SOA) has been integrated to achieve this. However, SOA has limitations regarding dynamic multi-service collaboration especially when the collaboration is triggered by complex events [3].

These limitations also affect the ability to adapt to changing business requirements, to the resolution of flexible and distributed collaboration, to the coordination of real-time interaction and collaboration between multiple agents [4].

IoT represents networks formed on a distributed architecture connected objects in which resources are not stored on a central node. These connected objects collaborate and coordinate their behavior [5] even if they have only limited knowledge of their environment or other objects on the network to achieve the system's goal. Coordination mechanisms can change based on coordination policies that allow objects the ability to decide on the coordination mechanism to use.

Since coordination service manages the behavior of IoT systems through the sharing of data and knowledge, the execution of tasks, etc. between its components, it becomes essential to conduct efficiently the behavior state of the whole IoT systems by supervising the execution of coordination mechanisms. Wherefore we have focused on integrating blockchain technology by using smart contracts with coordination policies due to its benefits in various fields. Blockchain is a technology that enables autonomous operation driven by rational decisions over participants. This ability to organize the flow of interactions reliable between -unknown-participants makes the blockchain a powerful instrument for coordination [6]. It decentralizes trust in a consensual manner, through peer interactions, and strengthens coordination. With the technology of smart contracts, objects can coordinate their behaviors to approve or reject changes in the coordination mechanism. Smart contracts stored on blockchains are self-executing contractual states that are not controlled by anybody, so they can be trusted. This can improve the impartiality and efficiency of policy management.

Blockchain, therefore, comes with the solution to the limitations of traditional coordination among IoT objects.

In our approach, smart contracts take a part in the coordination process as they define the coordination mechanism and the actions to undertake according to the environment's changes that agents take awareness of.

The remainder of the paper is structured as follows: Section 2 presents the related work along with a discussion according to the coordination process. Section 3 introduces our proposed IoT coordination architecture, whereas Section 4 highlights its behavior. Section 5 describes carbon leakage as a case study for experimentation. Section 6 presents the technical architecture used for the approach's implementation. Section 7 exposes the conclusion and future work.

II. RELATED WORK

The paper [7] presents a large-scale blockchain-based storage system, called Sapphire for data analytics in the internet of things. In the paper, authors have pointed out that autonomous coordination is required in a decentralized IoT solution, and the different IoT devices that interact with each other have varying levels of trust depending on the rules of engagement that they are operating via constraints. IoT devices are equipped with smart contracts that achieve contractual agreements with other devices. However, the use of smart contracts for the coordination of objects and their organization in the distributed network does not focus on the coordination in itself but its main goal is to achieve operational security.

An architecture for managing heterogeneous IoT is presented in [8]. It is mentioned that blockchain enables decentralized coordination to overcome IoT inherent challenges. The system is implemented in layers 3, 4 and 5 of the IoTWF reference model. The architecture adopts a hierarchical design that consists of having much core IoT participating in a public blockchain and used to realize monitoring, roll-backs, achieve coordination, and exchange data. The achievement of coordination is not further explained in the paper even though authors supported the importance of blockchain in the decentralized coordination of connected objects.

The paper [9] presents the importance of integrating IoT and Blockchain into the processes of a university campus. They started by introducing the three general architectural models of the three concepts: the university's network architecture, IoT, and distributed blockchain. Then, they established the university campus' architecture integrated with IoT to identify the security weaknesses of the IoT. And finally, they displayed where the integration of blockchain can handle the vulnerabilities and perform all the security needs an IoT system requires. In the paper, it is mentioned that the blockchain layer allows coordination between the different devices but does not shed more light on how the coordination has been approached.

In [10], an architecture pattern for trusted orchestration management is presented. The authors presented the behavioral and structural properties of the architecture pattern. They started by introducing the trusted orchestration management principles and requirements based only on security. The actions for this trusted orchestration management is the deployment and execution of the components along with the logging and identity management of all orchestration-related actions. Here, the authors included blockchain to manage security and smart contracts to define orchestration decisions. The use of blockchain and smart contracts in this paper was only from a security (identity, origin, non-repudiation) point of view to ensure trust.

Authors in [11] considered the integration of IoT, blockchain, and Building Information Modeling for the design of a smart building as the safety of people, security of data, and efficiency of construction are keys to smart building foundations. In this paper, it was highlighted that integration of blockchain and IoT with Building Information Modeling (BIM) is a great way of overcoming data sharing between IoT objects and managing data, but they didn't give further details as they focused only on security issues.

An IoT-Blockchain fusion model is presented in [12]. It integrates four layers: perceptual layer, network layer, blockchain layer, and application layer. Smart contracts are responsible for realizing transactions among the device. They are built on the blockchain which provides security and takes on the IoT data storage along with external distributed storage solutions. The devices' interactions and exchange of information are not mentioned in the paper. The use of blockchain was mainly for its trustful infrastructure.

In [13], a decentralized trust framework, called IoT Passport, for collaborative IoT based on blockchain technology is presented. It is composed of three layers: the Perception Layer, the Network Layer, and the Application Layer. Smart contracts are used to program common rules used to issue passports to each device, in addition to other rules about collaboration, authorization attributes, rewards, etc. Blockchain is used for its trust mechanisms and access control approaches.

III. DISCUSSION

The integration of blockchain with IoT is gaining momentum in research. However, in the works presented, this integration essentially presents only the security aspect and does not display the importance of integrating blockchain for coordination. In a matter of fact, blockchain cannot be used only for purposes namely security, trust, value transfer, but effectively for coordination... Coordination is a complex challenge that is linked to several others. In previous work, we established that to achieve coordination, we must take into account other challenges related to it namely discovery, heterogeneity, availability, context-awareness, decision-making [2], and which allows respecting the coordination process identification => data acquisition => data processing => Interpretation => decision-making [14].

1) *Identification*: gives a clear and unique identity to each device in the system. There are different identifying methods like EPC (electronic product codes), uCode, an identification number system. Addressing also can assist to uniquely identify objects within the network. It includes IPv4 and IPv6/6LoWPAN.

2) *Data acquisition*: means gathering data (sensing) from objects in the network. These objects can be sensors or actuators.

3) *Data processing*: when data is gathered from the devices, it needs to be processed. There are different methods of processing like classification, storing, calculation, etc. that are used to extract meaningful information from the data.

4) *Interpretation*: Information issued from data processing needs to be interpreted to facilitate decision-making. This interpretation can be done through different methods: policies, cloud-based analysis, machine learning technics, deep learning, neural networks, etc.

5) *Decision-making*: The support of appropriate decision-making is the interpretation phase. Decision-making dwells on the turn that the system can take through the actions to be taken. It is the result of extracting knowledge from interpreted information.

Table I presents the classification of related work according to the coordination process.

Due to blockchain's distributed aspect and with the use of multi-agents, coordination can be established despite all the challenges and issues linked to it. Indeed, sharing information and data is more accessible and cooperation among different parties is enabled and with the use of smart contracts, participants' behaviors are coordinated. This motivated us to propose an IoT architecture that integrates blockchain. It is presented in the following sections.

TABLE I. IOT COORDINATION PROCESS CLASSIFICATION

	Quanqing X. et al (2018)	Tseng L. et al (2020)	Villegas-Ch W. (2020)	Pahl C. et al (2018)	Lokshina I.V. et al (2019)	Gong X. et al (2020)	Tang B. et al (2019)
Identification	-	Private blockchain	-	Devices register to a local network	-	Unique address in the blockchain	IoT passport
Data Acquisition	Smart cities, smart grid, smart home, smart building	IoT devices	IoT devices	IoT devices	Smart building	Smart devices	IoT devices
Data processing	Data classification, custom processing	Analytics procedures in the cloud	Big data layer	cloud	-	Network layer	-
Interpretation	-	-	Big data layer	cloud	-	-	-
Decision-making	-	-	-	-	BIM	-	-

IV. IOT COORDINATION ARCHITECTURE

The following architecture is defined in agreement with the coordination process already presented. It consists of three layers described in the following and it is depicted in Fig. 1.

1) *Meta-workflow layer*: This layer holds the meta-workflow. It represents the static modeling of the global process which, in a common environment, pursues a normal execution.

2) *Coordination layer*

This layer represents the key of our architecture. It includes the following components:

- Policies: representing rules that will disrupt the behavior of the system. They are classified into two categories: environmental policies, which represent conditions on the data collected from the environment, and coordination policies which are established according to environmental policies and which determine the coordination mechanism to be adopted.
- Agent abstraction: Connected objects are abstracted using MAS. Each abstracting agent has access to the abstracted object components (i.e sensing and actuating components, computation, communication, and storage components). Thus, the sensed data are processed at this level to extract needed information. The choice to use agents at this level was decided as coordination falls under the distributed resolution of problems which consists in organizing the cooperation of connected objects by relying on each individual's capacities and sharing knowledge.

Agents are defined by their roles (or actions) and their knowledge of the environment.

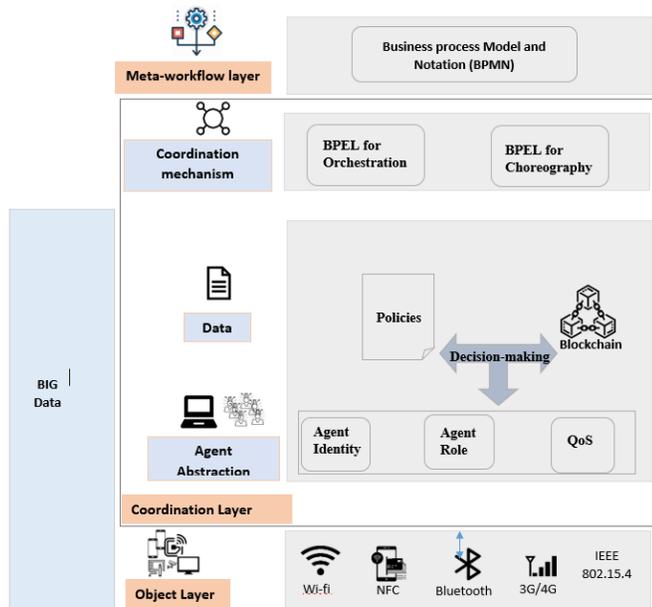


Fig. 1. Proposed Architecture.

Knowledge of the environment is the part of the environment's general states that the agent knows about. It takes into consideration one or several environmental policies.

Roles are the actions that the agent takes when operating:

- Data-collection: the agent interrogates its environment for collecting data.
- Data-storage: the agent stores information or collected data.
- Data-processing: The agent processes the collected data to extract information.
- Data-interpretation: interpretation of processed data.

• Data:

a) *Data storage*: During system operation, data is collected and stored for the purpose of analysis and monitoring in real-time.

b) *Blockchain*: Smart contracts: smart contracts group together the terms and conditions for the change of the coordination mechanism and also the agents involved in the coordination and who consent to that change.

Depending on the policies, the behavior of the system may change. The contract checks that everything is in order according to the environment and decides on which action(s) to undertake. Every call to the contract is a transaction saved on the blockchain.

- Coordination mechanisms: orchestration and choreography represent the coordination mechanisms that are used mostly. Usually, either orchestration or choreography is used as a coordinating mechanism. The goal is to combine the advantages of both to better manage the cooperation of objects in performing actions, especially critical ones.

CASE1: Orchestration: the whole organization is managed from the point of view of a single agent. The agent chosen to orchestrate the application's operation is the agent that acts as the system's environment (agent coupled with an actuator).

CASE2: Choreography: the organization is managed from a global perspective. Each agent is responsible for their behavior and works together with other agents to fulfill the overall goal of the application.

CASE3: Switch between orchestration and choreography: in this case, during the operation of the system, the coordination mechanism can change and switch from orchestration to choreography and vice versa depending on the agent's policies and coordination policies, according to the smart contracts.

CASE4: Both orchestration and choreography: throughout the system's operating, one or several agents can be engaged in choreography while at the same time their actions are orchestrated. Both mechanisms are used in parallel and the agent(s) enrolled have different tasks to manage.

3) *Object layer*: The object layer represents the various connected objects of the system. Different connection modes can be used, namely Wi-fi, Bluetooth, 3G/4G, NFC.

An object has its own mechanical and/or electrical function. For its connectivity either, the object is directly designed connectable or it exists already and the connectivity is added afterward.

These objects perform functionalities like querying their environment and collecting sensor data (sensors) and/or receive instructions to act (actuators). These functionalities require a source of energy that represents one of the great challenges of connected objects, both to guarantee the longest possible service performance without maintenance, and to guarantee environmental respect despite the proliferation of energy-intensive connected objects.

Physical objects are abstracted into agents. This abstraction is done through the description of these objects. Each of them has properties that can be described.

4) *Big data*: This module is an important part of any IoT system. It enables us to process a large amount of data on a real-time basis and storing them using different storage technologies. It helps to improve decision-making by analyzing gathered data on the system and the environment. For now, we didn't focus on this part as there are several solutions to data analysis for IoT. We believe we have to mention this part as it represents an important element of the development of any IoT system and we will go into details about this subject in our future work.

V. ARCHITECTURE'S BEHAVIOR

A. Meta-Workflow Layer

A meta-workflow designates a high-level process that defines the overall execution of the system or application, including the tasks to be performed in a normal environment and the overall objective to be achieved by the system are identified. The meta-workflow is defined in BPMN. Then, the workflow modeled in BPMN is translated into BPEL. This transformation makes it possible to have an executable process.

As the coordination has two mechanisms, orchestration and choreography, the BPEL makes it possible to describe both. Depending on the meta-workflow, the use cases for orchestration or choreography are written in BPEL and BPEL4CHOR.

B. Coordination

In its normal functioning, the coordination mechanism is set on either of the cases presented before. When the system encounters a delicate or critical case, depending on established policies, the coordination mechanism may change. This change and the actions that follow result from the call to the smart contract which represents the transactions that are stored on the blockchain. Agents communicate with each other using MQTT. Some require data from other agents to continue their work or provide services. At the slightest change in the environment that requires special attention, the agent or agents that handle the situation inform the coordinated agents to

confirm the change of the coordination mechanism following the smart contracts. They ensure that the conditions of the coordination policies are verified and then the transaction is done.

C. Policies

Coordination actors can be numerous and the existence of constraints limits the extension of the scope of coordination. Determining the coordination changes is necessary for the proper functioning of the system because, without a meticulous definition, coordination can conflict with the specific objectives of the objects. This is where the coordination policies come into force. They define the changes in the coordination mechanism through conditions that are evaluated from the data collected by connected objects.

D. Smart Contracts

Smart contracts are the system that will ensure the terms and conditions of the change in the coordination mechanism. For a given application, the contract guarantees that the coordination policies are respected for the two coordination mechanisms. It provides, based on environment data the coordination mechanism to apply and the actions to undertake under the change that has occurred.

E. Operating Coordination Algorithms

Operating algorithms for the Orchestration process and Choreography process are shown in Fig. 2 and Fig. 3.

In the case of an Orchestration

1. Orchestrator loads the smart contract
2. Orchestrator calls remotely the smart contract's methods through its instance
 - 2.1. The Contract's method:
 - 2.1.1. Evaluates the alert
 - 2.1.2. returns the result on the coordination mechanism and the action(s) to undertake
 - 2.1.3. IF the coordination mechanism changes:
 - 2.1.3.1. Orchestrator informs the coordinated agents
 - 2.1.3.2. Do choreography algorithm
 - 2.1.4. Else Continue on Orchestration algorithm

Fig. 2. Coordination Algorithm in Case of Orchestration.

In the case of a choreography

1. For each x in $\{j, k, \dots, p\}$
 - 1.1. Agent A_x loads the smart contract
 - 1.1.1. A_x calls remotely the smart contract's method through its instance
 - 1.1.2. Contract's method:
 - 1.1.2.1. Evaluates the alert
 - 1.1.2.2 Returns the result on the coordination mechanism and the action(s) to undertake
 - 1.1.3. IF the coordination mechanism changes:
 - Do Orchestration algorithm
 - 1.1.4. Else Continue on Choreography algorithm

Fig. 3. Coordination Algorithm in Case of Choreography.

VI. USE CASE STUDY

Smart home devices operate together, automating actions and sharing consumer data. This provides homeowners convenience, comfort, and energy efficiency. These functions require coordination among devices, thus the smart home was chosen to explain more the operating of our architecture. From the smart home use case, the application of a carbon leak is chosen to illustrate the functioning of the architecture.

We have three agents: CarbonAgent that senses the level of the carbon rate in the environment; oxygenAgent senses the oxygen rate in the environment; windowAgent that acts on the environment (action of opening or closing).

When the operating coordination mechanism is orchestration, the agent windowAgent is the orchestrator. The choice was made as it listens to events emitted by the other agents that capture environmental data and acts based on them. The operating of the application is as follows:

1) Carbon Agent

```
1- CarbonAgent.getCarbonValue()
   CarbonAgent.Save(CarbonAgent.getCarbonValue())
2- CarbonAgent.getCarbonPolicy()
3-if
(CarbonAgent.getCarbonValue().comparteTo(CarbonAgent.ge
tCarbonPolicy) ==1)
   CarbonAgent.Send(Alert)
4-if CarbonAgent.receive(decision)
   CarbonAgent.doDecisionAction()
   else Continue
```

2) Oxygen Agent

```
1- OxygenAgent.getOxygenValue()
   OxygenAgent.Save(OxygenAgent.getOxygenValue())
2- OxygenAgent.getOxygenPolicy()
3-if
(OxygenAgent.getOxygenValue().comparteTo(OxygenAgent.g
etOxygenPolicy) ==1)
   OxygenAgent.Send(Alert)
4-if OxygenAgent.receive(decision)
   OxygenAgent.doDecisionAction()
   else Continue
```

3) Window Agent

```
1-String address=WindowAgent.DeployContract(credentials)
2-Contract contract =
WindowAgent.LoadContract(address, credentials)
3-if (Receive
(CarbonAgent.Alert|| OxgenAgent.Alert))
   contract.Method()
   WindowAgent.receiveDecision()
   WindowAgent.send(decision)
   WindowAgent.doDecisionAction()
   else
   WindowAgent.close()
```

When choreography is of use in the scenario, the operating algorithm for each agent is as follows:

1) Carbon Agent

```
1-String address =
CarbonAgent.DeployContract(credentials)
2-Contract contract =
CarbonAgent.LoadContract(address, credentials)
3- CarbonAgent.getCarbonValue()
4-CarbonAgent.getCarbonPolicy()
5-if
(CarbonAgent.getCarbonValue().comparteTo(CarbonAgent.ge
tCarbonPolicy) ==1)
   contract.Method()
   CarbonAgent.receiveDecision()
   case1:
   CarbonAgent.send(Alert) to WindowAgent
   case2:
   Keep sensing (CarbonAgent.getCarbonValue())
```

2) Oxygen Agent

```
1-String address=
OxygenAgent.DeployContract(credentials)
2-Contract contract=
OxygenAgent.LoadContract(address, credentials)
3- OxygenAgent.getOxygenValue()
4-OxygenAgent.getOxygenPolicy()
5-if
(OxygenAgent.getOxygenValue().comparteTo(OxygenAgent.g
etOxygenPolicy) ==1)
   contract.Method()
   case1:
   OxygenAgent.send(Alert) to WindowAgent
   case2:
   Keep sensing (OxygenAgent.getOxygenValue())
```

3) Window Agent

```
1-String address=
WindowAgent.DeployContract(credentials)
2-Contract contract =
WindowAgent.LoadContract(address, credentials)
3-if
(Receive(CarbonAgent.alert || OxygenAgent.alert ))
   WindowAgent.open()
   else
   WindowAgent.close()
   contract.Method()
   WindowAgent.receiveDecision()
```

Here are the operating for each orchestration and choreography approach to coordination. Based on the data gathered from the environment, the smart contract decides which scenario to apply. The different cases that could be encountered are those presented in Section 3, coordination mechanisms.

VII. TECHNICAL ARCHITECTURE

This section presents the technical architecture that highlights the technologies used to implement the approach. It is depicted in Fig. 4.

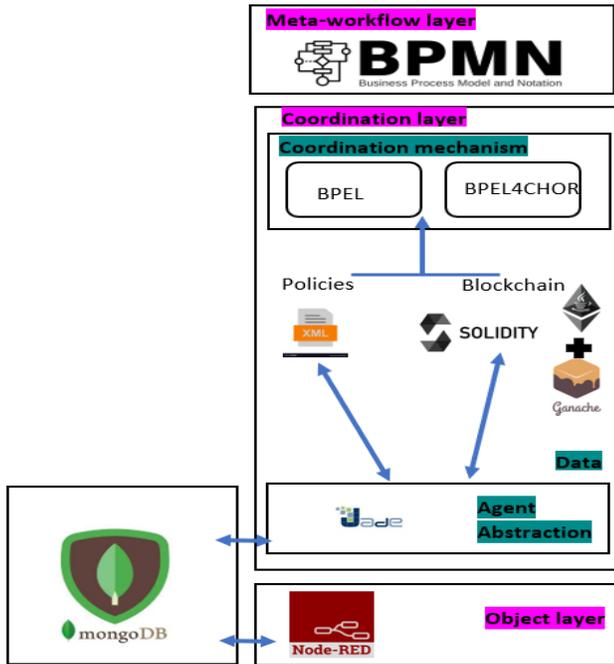


Fig. 4. Technical Architecture

A. Technologies of Implementation

1) *Meta-workflow*: BPMN: Business Process Model and Notation, is a business process modeling method to describe the value chains and business activities in the form of a graphic representation.

2) *Coordination*:

- **Mechanism**: Translation BPMN to BPEL/BPEL4CHOR. There are several solutions to transform BPMN into BPEL in the literature namely Oracle Business Process Analysis Suite/Oracle Business Process Management Suite; VIATRA2; graph transformation.
- **Policies**: XML.
- **Smart Contracts**: solidity. The contract written in solidity is compiled using solc which generates bin and abi files, and then, a java version of the contract is generated using web3j through the bin and abi files.
- **Blockchain**: ganache. It is a tool that enables you to set up a personal ethereum blockchain. All transactions are viewed and tracked there.
- **Agent abstraction**: Agents are developed in JADE (Java Agent Development Framework). The communication between agents is based on events and handled through MQTT. For this implementation, HiveMQ cloud is used.

3) *Objects*: Smart objects are handled using Node-red. Their profiles are stored in a mongo Data Base.

4) *Big data*: In addition to the objects' profiles, mongoDB is used to save data that agents gather from the environment for further analysis.

B. Description of the Implementation

This paragraph shows some implementation screenshots.

At the meta-workflow level, the functioning of the system is modeled with BPMN as shown in Fig. 5 and Fig. 6. Visually depicting system processes helps link the intent of the process to its implementation while serving to improve efficiency and accommodate new circumstances. BPMN, therefore, allows a better understanding of all the steps of a business process to achieve a more efficient process producing a quality result.

One of the advantages of BPMN is that it leads to the creation of documents in XML which are fundamental to the execution of the process, especially in BPEL. In BPEL, individual activities are called Partner Link exposed as web services that can be used by other BPEL processes. BPEL provides basic activities and functions as well as the exception and transaction handling, also to model control flow between activities there are structured elements such as loops or conditional commands.

Fig. 7 shows an example of the used BPEL code to describe an orchestration business process, whereas Fig. 8 displays an example of code to describe a choreography process.

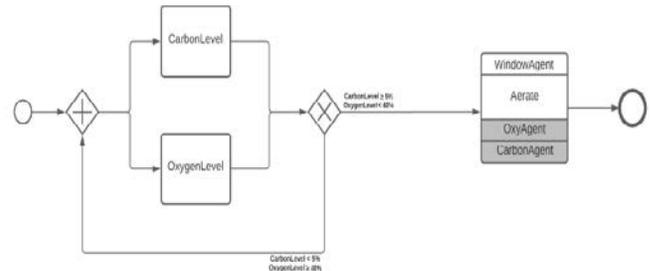


Fig. 5. Meta-Workflow in BPMN.

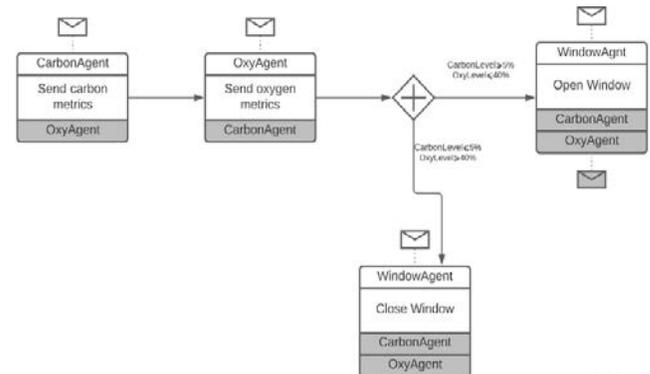


Fig. 6. Detail of the Choreography.

We plan as a future work to focus on scalability problem resolution, and the use of big data benefits to optimize the decision-making on the coordination mechanisms.

REFERENCES

- [1] M. Hung, "Leading the IoT" Gartner Research 2017.
- [2] R. Belkeziz and Z Jarir, "An Overview of the IoT Coordination Challenge" on International Journal of Service Science, Management, Engineering, and Technology., vol. 11, 2020.
- [3] B. Cheng, M. Wang, S. Zhao, Z. Zhai, D. Zhu and J. Chen, "Situation-Aware Dynamic Service Coordination in an IoT Environment" in IEEE/ACM TRANSACTIONS ON NETWORKING, vol. 25, August 2017.
- [4] P. Leitão "Toward self-organized Service-Oriented Multi-agents systems" in Studies in Computational Intelligence, vol 472, 2013.
- [5] K. Seung Woo, K. Mingoo, P. Jong-Il, "IoT delegate, Smart Home Framework for Heterogeneous IoT Service Collaboration" in KSII Transactions on Internet and Information Systems, vol. 10, 2016.
- [6] G. Ciatto, S. Mariani, A. Maffi, A. Omicini, "Blockchain-Based Coordination: Assessing the Expressive Power of Smart Contracts" in information, vol. 11, 2020.
- [7] Q. Xu, K.M.M. Aung, Y. Zhu and K.L. Yong "A Blockchain-Based Storage System for Data Analytics in the Internet of Things" in New Advances in the Internet of Things. Studies in Computational Intelligence, vol 715, 2018.
- [8] L. Tseng, L. Wong, S. Otoum, M. Aloqaily and J.B. Othman "Blockchain for Managing Heterogeneous Internet of Things: A Perspective Architecture," in IEEE Network, vol.34, 2020.
- [9] W. Villegas-Ch, X. Palacios-Pacheco and M. Román-Cañizares, "Integration of IoT and Blockchain to in the Processes of a University Campus," in Sustainability vol. 12, 2020.
- [10] C. Pahl, N. El Loini, S. Helmer, B. Lee, "An Architecture Pattern for Trusted Orchestration in IoT Edge Clouds" in Third International Conference on Fog and Mobile Edge Computing, 2018.
- [11] I. V. Lokshina, M. Gregus, W. L. Thomas, "Application of Integrated Building Information Modeling, IoT and Blockchain Technologies in System Design of a Smart Building" in Procedia Computer Science, vol. 160, pp.497-502, 2019.
- [12] X. Gong, E. Liu and R. Wang, "Blockchain-Based IoT Application Using Smart Contracts: Case Study of M2M Autonomous Trading" in 5th International Conference on Computer and Communication Systems, 2020.
- [13] B. Tang, H. Kang, J. Fan, Q. Li and R. Sandhu "IoT Passport: A Blockchain-Based Trust Framework for Collaborative Internet-of-Things," in Proceedings of the 24th ACM Symposium on Access Control Models and Technologies, 2019.
- [14] R. Belkeziz, Z. Jarir and I. EL Kassmi "Multi-agent Reinforcement learning to IoT Coordination," in Journal of Physics: Conference Series, vol. 1743, 2021.