IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 7 July 2022, accepted 20 July 2022, date of publication 27 July 2022, date of current version 2 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3194264

RESEARCH ARTICLE

An Architecture Combining Blockchain, Docker and Cloud Storage for Improving Digital Processes in Cloud Manufacturing

GAETANO VOLPE, AGOSTINO MARCELLO MANGINI, (Member, IEEE), AND MARIA PIA FANTI, (Fellow, IEEE)
Department of Electrical and Information Engineering, Polytechnic University of Bari, 70125 Bari, Italy
Corresponding author: Gaetano Volpe (gaetano.volpe@poliba.it)

ABSTRACT The Blockchain has been given great attention in recent literature among emerging technologies in software architectures. More specifically, when verifiable transactions between untrusted parties are concerned in a safe and reliable environment, its peculiar decentralized and tamper-proof structure makes it suitable for a vast class of business domains, such as Cloud Manufacturing, which is a new paradigm in the industry based on cloud technologies. However, the stiffness of existing solutions, that are unable to provide and implement heterogeneous services in a Cloud environment, emphasizes the need of a standard framework to overcome this limit and improve collaboration. Firstly, this paper introduces a Blockchain based platform designed with Smart Contracts for improving digital processes in a manufacturing environment. The primary contribution is the integration of two popular cloud technologies within the Blockchain: Docker, a scalable platform to run applications in lightweight environments, and Cloud Storage. Each process available in the platform requires input files and produces output files by using cloud storage as a repository and it is delivered by the owner as a self-contained Docker image, whose digest is safely stored in the chain. Secondly, with the purpose of selecting the fastest node for each new process instance required by consumers, we introduce a task assignment problem based on a deep learning approach and past metrics. The proposed platform is applied to a real-world industrial case study regarding ophthalmic lenses manufacturing and the optimization of lens surface calculation.

INDEX TERMS Cloud manufacturing, blockchains, software architecture, distributed ledger, smart contracts, computer architecture.

I. INTRODUCTION

Cloud Computing significantly changed the way industries do their business today [1]. In 2020, enterprise spending on cloud infrastructure services amounted to almost 130 billion U.S. dollars, while in 2011 it was only 3.5 [2]. In manufacturing industry, the adoption of cloud technologies lead to the new paradigm of *Cloud Manufacturing* [3], which enables companies to provide and receive services over Internet via an intelligent service-oriented architecture.

The associate editor coordinating the review of this manuscript and approving it for publication was Nitin Gupta.

Decentralization is a key factor both in Cloud Computing and Cloud Manufacturing as distributed architectures improve services reliability, scalability and performance when compared to traditional on-premise centralized approaches [3]. Similarly, security is also a key concern in these contexts as data elaborated in the cloud may be compromised, altered or stolen and the integrity of services could be severely affected [4].

Blockchain, as a secure ledger for storing transactions, plays a strategic role in modern distributed architectures and the recent literature has shown a great interest about this revolutionary technology over the last decade. A systematic review provided in [5], shows that Blockchain based application have been adopted in many business domains: from popular financial applications, such as cryptocurrencies, to completely different fields, such as *Business and Industry*, *IoT*, *Governance*, *Education*, *Health*, and *Privacy & Security*.

Smart Contracts are typically the core component on which most of these applications are based. A Smart Contract is essentially an algorithm that is coded and executed in the Blockchain itself with the purpose of regulating a specific agreement between parties participating in transactions. A comprehensive summary of smart contracts and their most popular applications is provided in [6].

Among all business domains, a specific attention in recent literature has been given to the application of Blockchain in *Business & Industry* [5]. In detail, as described in the taxonomy provided in [7], a well structured class of Industry 4.0 applications are already implemented. Moreover, the authors describe the adoption of Smart Contracts in fields such as *Internet-Of-Things*, *Supply Chain*, *Manufacturing* and *Energy*. However, in [8], the challenges and the current status of Blockchain in manufacturing are surveyed and the relative research paths for the near future are suggested by the authors.

Some specific solutions for *Engineering and Manufacturing* are detailed in [9]. The reviewed applications are designed for specific purposes such as increasing the efficiency of the manufacturing process, protecting data validity and enhancing communications inside and outside the organization.

The Blockchain, as a tool that could be employed by all the management stakeholders of a product life-cycle is argued in [10] by describing different purposes, such as making deals and sharing information about products. In their work, the authors review the state-of-the-art literature in the field from both the manufacturing system perspective and product life-cycle management perspective. In general, when sustainable manufacturing in Industry 4.0 is concerned, the Blockchain is considered as a concrete enabler for current Enterprise Resource Planning and Manufacturing Execution Systems solutions.

As an example of such application, in [11], a decentralized task execution model for Industrial-Internet-of-Things (IIoT) environments named *ManuChain* is suggested. The proposed architecture has a dual layer architecture and it is based on a permissioned Blockchain and specifically designed Smart Contracts. More recently, in [12], the authors introduce a Blockchain based security service architecture for Cloud Manufacturing environments, that facilitates authentication and privacy protection in decentralized 5G IIoT contexts.

Collaborative *Business Process Management* (BPM) in Blockchain domain is discussed in [13]. In their systematic review, the authors show that, among others, papers about *process implementation & execution* and *process modeling* phases are the most representative.

In this context, a common research topic is the translation in Smart Contracts of processes defined according to the well known Business Process Model and Notation (BPMN) [14]. In [15], an open source BPM system named *Caterpillar*, is suggested. The solution includes a BPMN to Smart Contract compiler to perform the translation in automatic fashion.

Concerning the Cloud Manufacturing, the existing Blockchain based solutions already satisfy some of the most critical requirements such as decentralization and security. Furthermore, compared with other distributed architectures, the Blockchain is known to have a greater performance when those features are concerned. In addition, in several business domains, a Blockchain peculiar feature such as the immutability of transactions clearly enhances their credibility while, at the same time, the unnecessity of trusted third parties makes easier to verify contents and improves the reliability of a given platform [5].

Moreover, in [16], the advantages of using Blockchain in such environments and, specifically, for Additive Manufacturing and 3D printing is noticeable. In that respect, the authors observe that preserving the intellectual property to secure CAD models is a critical issue in such context and that the Blockchain technology can be efficiently implemented to license the models and secure the whole manufacturing process. In addition, in [17], a concrete example of a Blockchain and Smart Contract based architecture for Additive Manufacturing is proposed with the specific purpose of enforcing agreements between design and 3D printing companies and preserve the intellectual property of design files.

However, it is often required that modern platforms satisfy other important requirements in the context of collaboration. In particular, they need to be standardized and flexible in order to be easily implemented. As suggested in [1] and [3], most of the existing solutions about the development of virtualization and servitization technology are still weak and, consequently, fail to fit those requirements.

This paper focuses on particular manufacturing processes that produce pieces in small lots or in single unit, such as ophthalmic lenses or 3D printed products. Such types of manufacturing procedures require specific digital processes, computational intensive tasks and raise standardization, cybersecurity and intellectual property preservation issues.

Cloud solutions and standardized frameworks can enable different actors to effectively provide, update and consume any kind of process delivered as a service. The absence of such standard framework in current related literature makes it hard to flexibly integrate a generic software logic in such environments and impacts on servitization process.

Motivated by this open issue, this work proposes a decentralized, distributed, cybersecure and collaborative platform to improve digital processes in Cloud Manufacturing environments. The proposed platform is based on Ethereum [18], which is one of the most popular Blockchain implementations, and Smart Contracts. In addition, different actors such as *Process Owner*, *Process Consumer* and *Process Runner* cooperate in the platform to handle process life cycle and effectively execute digital business processes.

Differently from previous solutions, such as [17], that consider only Blockchain as enabling technology, the main contribution of the paper is using Docker, Cloud Storage and

Deep Learning in the Manufacturing Sector, integrated with Blockchain, in order to solve the problem of standardizing the processes, ensuring flexibility and security, preserving intellectual property, and enabling scalability. Such requirements are addressed by the following technologies in an innovative approach.

First, we integrate in the proposed Blockchain platform two popular cloud technologies: *Docker* [19] and *Cloud Storage* [20]. Docker is a container-based platform to run applications in isolated environments from lightweight and self-contained images and it has great advantages in terms of performance, scalability, and portability over traditional virtual machines [21]. While Docker in a typical Blockchain architecture is only employed to run the components of the Blockchain itself, in the proposed platform, the process logic is delivered to the runners right in terms of a Docker image, which includes both the application and its dependencies. To the best of our knowledge, it is the first time that Docker is used in Cloud Manufacturing and in a Blockchain environment in this specific role to provide a flexible and distributed computational environment.

Second, each Docker image is identified by a unique *digest* that is stored and updated in the chain by the *Process Owner* through a Smart Contract. Since the Blockchain is tamper-proof in nature, by storing the *digest* in the chain, consumers and runners are always guaranteed to use the latest version of the process as deployed by the *Process Owner*, thus preserving the integrity of the execution. Moreover, in order to keep the Blockchain not too heavy, we rely on traditional Cloud Storage platforms to store and retrieve instances related input and output files.

Third, since in Cloud Manufacturing task scheduling plays a fundamental role [22] and the allocation of physical resources such CPU and memory in a shared Cloud Computing environment is generally variable, we introduce a task assignment problem with a *deep learning* approach to predict process run-time that is based on past Docker containers metrics. In this way, given an instance request in a specified time band, we are able to choose the fastest runner.

The proposed blockchain platform including Docker and Cloud storage is implemented in a real case study of ophthalmic lenses manufacturing. Thanks to the flexibility of Docker technology, we show that the proposed platform combines the great advantages of existing Blockchain based solutions in Cloud Manufacturing in terms of reliability, performance, and cybersecurity with a completely different service deployment approach. In fact, in this environment any process that can be packed in a Docker image can be easily and safely implemented. Hence, a more effective collaboration between owners, runners and consumers can be achieved with no integration overhead for new processes.

This paper is an extension of the previous paper [23] where the proposed platform was described in an embryonic form. In this work, we provide a detailed description of the platform architecture and large implementation details about the case study regarding ophthalmic lenses manufacturing. In particular, the task assignment problem and the related deep learning approach are described by focusing on the lens surface calculation step. Moreover, we discuss the specifications of the process implementation on the proposed platform highlighting the related benefits in terms of cybersecurity, reliability and performance compared to the traditional solutions.

The remainder of this paper is organized as follows: Section II introduces some basic concepts related to Blockchain, Smart Contracts, Docker, Cloud Storage and Digital Processes. In Section III, we introduce the platform architecture describing in detail its principal components, use cases, smart contracts and client applications involved. In section IV, we present a real case study regarding ophthalmic lenses manufacturing. Finally, section V concludes the paper.

II. BACKGROUND

A. BLOCKCHAIN AND SMART CONTRACTS

Blockchain is essentially a ledger whose architecture is distributed rather than centralized. Its main purpose is to record transactions, signed at least by the issuer, and allow their verification by all nodes joining the peer-to-peer network, without the need of a central authority [5].

As the name itself suggests, the ledger is made by a chain of blocks, in which each block stores a certain number of transactions. The chain is guaranteed to be *tamper-proof* as each block is connected to the previous one by including its Secure Hash Algorithm (SHA) *SHA-256* in the header [24].

As the order in which the blocks are connected is relevant for the transactions history, all nodes must reach a sort of agreement whenever a new block has to be added to the chain. For this purpose, many different *consensus* mechanisms have been studied in recent years [25].

In a Blockchain, there is also the possibility to enforce a specific agreement between two or more parties during a transaction. In this case, one or more *Smart Contracts* are employed [26]. A Smart Contract is a small algorithm coded in the Blockchain in languages such as *Solidity* [27] and *Vyper* [28] that is executed when specific trggering conditions are met.

B. DOCKERS, CONTAINERS AND CLOUD STORAGES

Docker is a client-server platform on which multiple applications can be run in isolated environments called *containers* on the same host. Containers are different from regular virtual machines in the sense that they do not need an hypervisor, but they leverage the kernel of the host operating system. However, a full life-cycle management is provided within the platform, including actions such as create, start, stop and destroy and features such as collecting metrics.

Each container is run on the top of a Docker *image*, which is a template built according to a list of instructions included in a *Dockerfile*. Each of those instructions is a single *layer* that is added to the image filesystem. For each image a unique fingerprint can be indicated by calculating a single *SHA-256 hash* that considers all of its layers.

Docker images are typically stored in a *registry*, which is essentially a repository that allows users to pull or push images according to their permissions and to store multiple versions of a single image identified by different *tags*.

Cloud Storage is one of the most popular cloud technologies and it is offered by some of the major providers, such as *Amazon* and Microsoft by the *S3*[1] service and the *Azure Blob Storage*[2] platform respectively. It is essentially a data storage service [20] that enables users to store their files in the Cloud.

C. DIGITAL PROCESSES AND TIME PREDICTION ALGORITHM

In this work, a digital process is defined as a software task that is typically made of a computationally intensive algorithm. The task requires one or more input files and produces a certain number of output files to be used in the following steps of a manufacturing process.

In addition, we refer to a *Time Prediction Algorithm* as a function to predict the amount of time needed to complete tasks on a specified environment that we call *Process Runner*. For this purpose, a deep learning approach based on an *Artificial Neural Network* (ANN) is introduced. The proposed ANN is trained by considering the past metrics collected from process runners after each task execution.

III. THE PROPOSED PLATFORM STRUCTURE

In this section, we introduce a novel practical approach for implementing BPM in the Blockchain by leveraging on the features of two of the most popular cloud technologies: *Docker* and *Cloud Storage*. This platform is specifically designed for manufacturing environments, nevertheless other implementations are viable.

Based on Dumas BPM life-cycle [29], we suggest the use of Docker in *process execution* phase, supposed that the process to be executed is a digital process. Moreover, we store process inputs and outputs in a traditional cloud storage and we use Blockchain and Smart Contracts for *process implementation and monitoring*. We also introduce a basic task assignment problem based on execution time prediction performed through an ANN trained with past process runs metrics.

The proposed platform architecture is described in Fig. 1. The core of the platform, in the central part of the scheme, is a *Consortium Blockchain* built on *Ethereum*. It implements two main use cases: *Create Process* and *Consume Process*, that represent the process life-cycle which is orchestrated by the *Process Smart Contract*.

Given the consortium nature of the Blockchain, we decided to use *Proof-of-Authority* consensus algorithm [30]. More in detail, *Create Process* is related to the creation of a new process and *Consume Process* describes a single instance life-cycle.

In addition, four client applications are also provided for the purpose of interacting with the Blockchain: the *Process Client Application* that implements the main use cases, the *Process Runner Application* that interacts with the Docker based process runners, the *Permission Granting Application* that listens for events on the *Process Smart Contract* and orchestrates the permissions on the Cloud Storage and, finally, the *Data Mining Application* that collects metrics about past instances and updates the ANN weights on a dedicated *Oracle Smart Contract*.

A. COMPONENTS AND USE CASES

In this subsection, the main components of the platform are described in detail. We identify five main entities in the platform related to the process life-cycle:

1) **Process Owner**: It represents the entity that created the process core logic and coded the related algorithm. An owner usually delivers a process to the platform that can be requested by multiple consumers.
   This entity is involved in the following use case:

   > **Create Process**: It is the action of creating the process in the chain. First, the *Process Owner* packs the algorithm files in a Docker image along with all dependencies, then it sends the image to a Docker registry and finally executes the *CreateProcess* function.

2) **Process Consumer**: It represents the entity that needs to perform an instance of the digital process for its business purposes.
3) **Process Runner**: It provides the computational resources to execute the process instance by running the related image in a container.
4) **Permissions Granting Application** (PGA): this component assigns and revokes permissions on files in the Cloud Storage.
   These last three components are involved in the following use case:

   > **Consume Process**: First, the *Process Consumer* requests a new instance of the process by invoking the *ProcessInstance* function, then a *Process Runner* is assigned by the Smart Contract and the instance is finally executed. Both the *Process Consumer* and the *Process Runner* need to interact with the Cloud Storage for storing and retrieving instance inputs and outputs, therefore the PGA plays its role by assigning and revoking the related permissions.

5) **Data Mining Algorithm**: It is the external component that collects past instances metrics from the chain and trains the execution time prediction ANN for each registered runner and for each known process on a regular time basis. At the end of each training, it pushes

[1] https://aws.amazon.com/s3/
[2] https://azure.microsoft.com/en-us/services/storage/blobs/

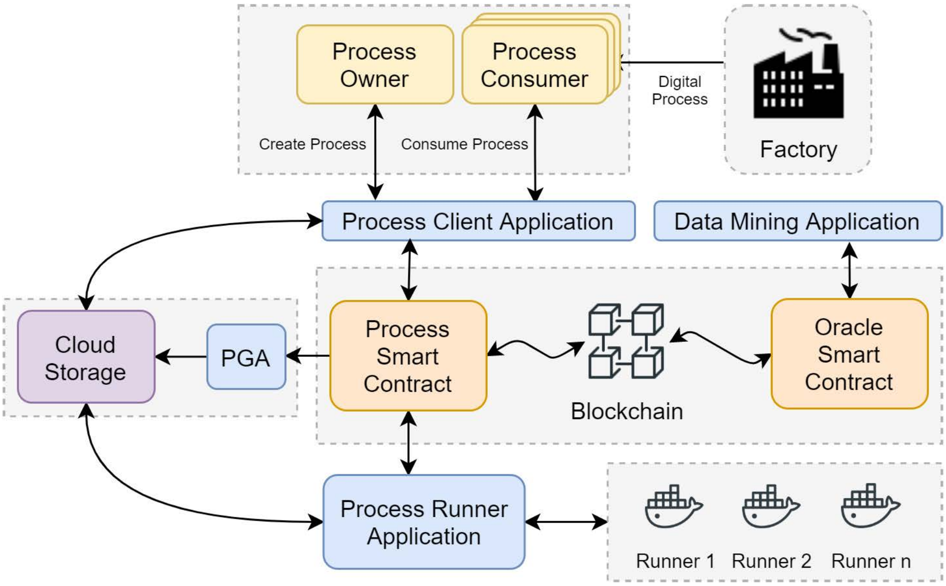

FIGURE 1. **Platform architecture & use cases.**

updated weights for each runner to the *Oracle Smart Contract* in the Blockchain.

B. SMART CONTRACTS

The whole process life-cycle relies on two Smart Contracts developed in Solidity programming language:

1) PROCESS SMART CONTRACT

It stores data of processes, runners and instances. In addition, it provides *CreateProcess* and *ProcessInstance* functions that, in the proposed platform, represent the two main transactions of the Blockchain. As shown in Listing 1, the Process model includes the relevant Docker image name (optionally with its tag) and the docker SHA-256 *digest* that ensures the integrity of the execution, effectively encoding the process logic in the Smart Contract and in the Blockchain. Consequently, when the Process Owner invokes the *CreateProcess* function in the Smart Contract, it creates a new unique process identified by the related Docker image *digest*. The *Process Runner* is therefore allowed to use only that image when it is assigned a new instance. Should it retain a different version in its local repository, it must preliminary pull the required image from the remote registry before executing the instance.

The *ProcessInstance* function is depicted in the *Unified Modeling Language* (UML) [31] activity diagram in Fig. 2. When the consumer requests a new process instance, it stores the inputs in its area on the Cloud Storage. At the same time, the *ProcessInstance* function selects the fastest *Process Runner* by predicting the execution time for each registered available runner based on the weights stored in the *Oracle Smart Contract*. Since each runner is set for events listening on the Smart Contract, it becomes aware of the assignment and decides to accept or refuse the task by executing an auxiliary function. If it accepts, it is immediately granted the needed permissions on the cloud Storage by the PGA to retrieve the instance inputs. Once the inputs are available, the process instance is executed in a Docker container with the requested image. At the end of the execution, the Smart Contract is notified and the outputs are stored on the Cloud Storage to be finally retrieved by the consumer. On the contrary, if the first assigned runner refuses, the Smart Contract is notified and next fastest runners are attempted until one of them accepts or there are no more available runners. In the latter case, the instance ends and the consumer is notified so that it can try again at a later time.

2) ORACLE SMART CONTRACT

The *Oracle Smart Contract* is used to store the calculated weights of each ANN trained on past instance metrics. The weights are updated on a regular time basis. The whole process is implemented in the *Data Mining Application* described in Section III-C.

C. CLIENT APPLICATIONS

The interaction of the components with the Blockchain and the Smart Contracts is carried out by using client applications that invoke functions in the Smart Contract and listen to the chain events.

In this work, we introduce four applications:

1) **Process Client Application**: it implements *Create Process* and *Consume Process* use cases and interacts with the Cloud Storage to store instance inputs and retrieve instance outputs. It could be also connected with third-party applications to exchange files and data.

```
pragma solidity 0.8.1;

contract Process {
    enum ProcessInstanceState
    {Placed, Assigned, Completed, Refused}
    uint public totalProcesses;
    uint public totalProcessInstances;
    Process[] public processes;
    ProcessInstance[]
    public processInstances;

    struct Account {
        int256 balance;
        bool accountType;
    }
    struct Process {
        string dockerImage;
        // Docker SHA-256 Hash
        bytes dockerId;
        uint clickFee;
        address owner;
        ProcessRunner[] runners;
    }
    struct ProcessRunner {
        address runner
    }
    struct ProcessInstance {
        uint processId;
        address runner;
        address consumer;
        uint executionTime;
        uint executionFee;
        uint totalFee;
        bytes inputId;
        bytes outputId;
        ProcessInstanceState state;
    }
    function createProcess(...)
    function processInstance(...)
    ...
```

Listing 1. Solidity code snippet for process smart contract.

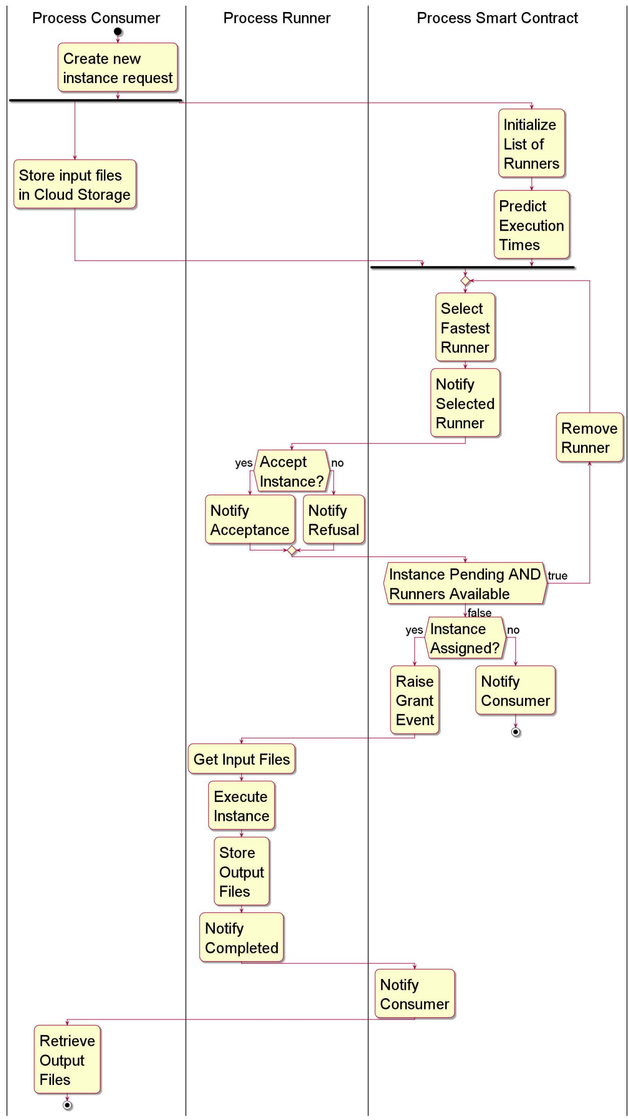

FIGURE 2. ProcessInstance function UML activity diagram.

2) **Process Runner Application**: it implements *Consume Process* use case and interacts with the Docker host to pull requested images from the registry and run the instance container for process execution. It also interacts with the Cloud Storage to retrieve instance inputs and store instance outputs.
3) **Permission Granting Application - PGA**: it listens for chain events and assigns and revokes permission on instance inputs and outputs for consumers and runners.
4) **Data Mining Application**: it collects past instances metrics by querying the *Process Smart Contract* on a regular time basis. Each observation in the training dataset includes the day of the week and the time band (hour of day) as input, while the actual running time in seconds is collected as output. Those features are depicted in Table 1. After building the dataset, a single multilayer perceptron ANN (MLP) for each runner and for each process is trained by using standard back-propagation. MLP is a class of feedforward artificial neural network composed of multiple layers of perceptrons with a threshold activation and it is widely used for regression problems where, given a set of inputs, a real-valued quantity is to be predicted. After training, the calculated weights are pushed to the *Oracle Smart Contract* to be used for future instances running time prediction. In this way, although a machine learning approach is adopted, the task assignment process is deterministic and can be easily integrated in a Blockchain based architecture.

IV. CASE STUDY: OPHTHALMIC LENSES MANUFACTURING

In this section, we present an implementation of the proposed architecture in ophthalmic lenses manufacturing. Indeed, the manufacture of ophthalmic lenses is a typical production of single, specific pieces, each one made on the basis of a

TABLE 1. Time prediction ANN dataset details.

Feature	Domain	Description
dayOfWeek	$1 \leq x \geq 7, x \in \mathbb{Z}$	Day of the week, Sun-Sat
hourOfDay	$0 \leq x \geq 23, x \in \mathbb{Z}$	Hour in which instance has been requested
runningTime	$\mathbb{R}_{>0}$	Actual execution time (Output)

unique medical prescription. Moreover, the lenses production requires high amount of data and complex optimization. On the other hand, cybersecurity issues must be considered to secure the ownership of the lenses design and implementation software. For this reason, the selected case study reflects the requirements to apply the proposed technology and allows us to enlighten the benefits of the Blockchain architecture.

The process of lens manufacturing is typically made of five steps in sequence: *Calculation*, in which the lens surface to be machined is computed, *Surfacing*, *Polishing*, *Coating* and, finally, *Edging*.

In this work, we focus on *Calculation*, which is a pure digital process. A typical lens calculation software is called *Lens Design System* (LDS).

A. LENS DESIGN SYSTEM

A LDS takes the patient's ophthalmic prescription as input, which is typically called a *job*, and performs a mathematical calculation that produces the data to be sent to a CNC machine to realize a surface on the back of a semi-finished lens. We depicted the process using BPMN notation [14] as shown in Fig. 3.

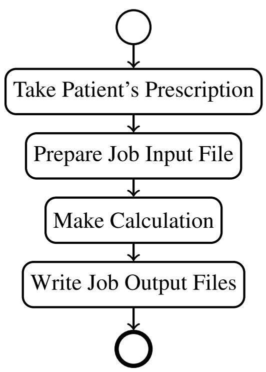

FIGURE 3. Lens design system process.

In the eye care industry the exchange of data between machines and systems is regulated by a standard published by the *Vision Council of America*, an US based organization that includes almost all the relevant players. The name of this standard is *Data Communication Standard* and the current version is 3.12 [32].

Listing 2 shows a typical job input file including prescription data along with other relevant manufacturing parameters.

Each parameter is identified by a record label followed by a record separator, which is invariably the "=" sign,

```
JOB=1111
DO=B
REQ=LDS
LNAM=LensDesignName;LensDesignName
SPH=+1.00;+1.00
CYL=+0.0;+0.00
AX=0.0;0.0
ADD=2.0;2.0
CRIB=60;60
LIND=1.5;1.5
FRNT=5.00;5.00
```

Listing 2. LDS job input file.

TABLE 2. DCS record labels description.

Record Label	Description
LNAM	Name of lens style
SPH	Rx Sphere power (diopters)
CYL	Rx cylinder power (diopters)
AX	Prescribed cylinder axis
ADD	Addition power for multifocal lenses (diopters)
CRIB	Lens diameter (mm)
LIND	Refraction index of lens material
FRNT	Blank true front curve for power calculations (diopters)
MINEDG	Minimum edge thickness of finished lens cut to shape (mm)
GBASEX	Generator base curve (diopters)
GCROSX	Generator cross curve (diopters)

```
JOB=1111
DO=B
REQ=LMS
LNAM=LensDesignName;LensDesignName
CRIB=60;60
LIND=1.5;1.5
GBASEX=-4.43;-4.43
GCROSX=-4.60;-4.60
```

Listing 3. LMS job output file.

and then by actual field values for right and left eyes, separated by a semicolon ";". Table 2 shows the meaning of the most relevant labels included in the presented case study.

The type of lens surface to be calculated is set by the LNAM label value and in some cases, such as for progressive lenses, it heavily affects the calculation time. In fact, while conventional single vision lenses have a spherical or toric shape and can be easily computed, when it comes to progressive lenses, a differential geometry problem has to be solved to compute the resulting *freeform* surface. These problems are often computationally intensive and thus require several seconds of processing.

In the case of conventional geometric shapes the output file, which is called the LMS file, simply includes the radius of the sphere or of the torus. On the contrary, in case of freeform surfaces, an additional file is needed which describes the whole surface through a square matrix and it's called the SDF file. In Listing 3 and 4 an example of both files is provided.

```
REQ=SDF
JOB=1111
SURFMT=1;R;B;85;85;84;84
...
ZZ=16.688;16.176;15.682;15.206; ...
ZZ=16.310;15.800;15.308;14.833; ...
...
```

Listing 4. SDF job matrix file - 84 × 84 mm - step 1mm.

B. IMPLEMENTATION

This subsection proposes an implementation of a LDS system based on the presented architecture.

Firstly, we identify the actors of the case study as follows: *Process Owner* is the Lens Designer, *Process Consumer* is the Lens Manufacturer and *Process Runner* is the Surface Calculator.

In this case study, a Lens Designer implements its calculation algorithm with the programming language of his choice and packs the compiled application along with all libraries and dependencies in a Docker image to be pushed to a container registry.

Although the *Process Smart Contract* encodes the calculation logic by including the Docker *digest* in the model, it is in charge only of the process flow, therefore no particular changes are needed to support the specific case study.

Consequently, the implementation of *CreateProcess* and *ConsumeProcess* functions for respectively adding new lens design and requesting new calculation is straightforward.

Moreover, even the implementation of client applications does not require any particular change from proposed architecture as we only need to collect input and output files both on client and runner edges.

However, the Process Client Application needs to be connected to the lens manufacturer's information system to allow the CNC machine to retrieve the calculated surface.

We use *Amazon S3* service as our cloud storage, therefore both consumers and runners need to have an account on those platforms to store and retrieve files.

C. RUNNING TIME PREDICTION ANN

Data Mining Application collects metrics from past lens calculation instances and trains a model for each runner and for each process to predict the running time of a single lens design instance on the runner at a specified time.

In this particular context, calculation time is pretty stable for a design unless other simultaneous processes are running on the node and the available resources, such as cpu and memory, must be shared.

In order to simulate a real environment and evaluate the efficacy of the prediction approach, we use one *c5d.large* node on Amazon Web Services equipped with two *Intel Xeon 3.6Ghz* vCPUs and 4GB of RAM. In addition, we choose a progressive lens design and we run manually a set of 9143 calculations to get a reference value of running time in seconds in both *busy* and *normal* state to evaluate the performance of the sample node under different load conditions. In the first *busy* case, namely when there are multiple intensive processes running at the same time on the node, the calculation mean time is 15s with a standard deviation of 1.0s. Conversely, in the second *normal* case, when only one process is running at a time, the mean of calculation time is 7s with a standard deviation of 0.5s. We use calculated values and two normal distributions to build multiple random datasets of 200 samples/hour, from 8:00 to 20:00, from Monday to Sunday, collecting a total of 18200 samples.

In particular, for each day of the week, after randomly picking a two hours period for busy state condition, we build a first normal distribution:

$$X_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$$

where X_1 is a random variable that represents the calculation time in busy state condition with mean value $\mu_1 = 15s$ and standard deviation $\sigma_1 = 1s$.

Then, we build a second normal distribution:

$$X_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$$

where X_2 is a random variable that represents the calculation time in normal state condition with mean value $\mu_2 = 7s$ and standard deviation $\sigma_2 = 0.5s$.

Finally, we randomly sample 200 values/hour either from X_1 or X_2 according to the corresponding state. Since we use a normal distribution, almost 99.7% of the sampled values fall in the interval $[\mu - 3\sigma, \mu + 3\sigma]$.

After generating the dataset, we build the ANN described in Table 1. Since a multilayer perceptron network requires numerical values as input, *One-Hot* encoding technique [33] is used to transform *day of week* and *hour of day* features in a single vector of only 0 and 1 for each sample. The resulting network is depicted in Fig. 4 and consists of 20 neurons in the input layer, 21 neurons in the hidden layer and a single neuron in the output layer which represents the calculation time.

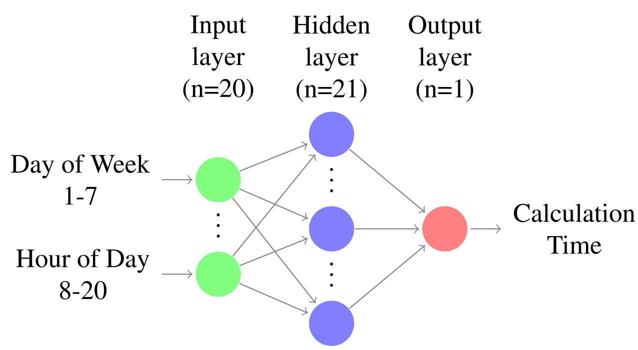

FIGURE 4. MLP network infrastructure.

We split the original dataset into 80% training and 20% test sets. Given that we deal with a regression problem, the network is trained with Mean Squared Error (MSE) as loss function to minimize the average squared difference between the collected calculation time values and the predicted values, whereas Rectified Linear Unit (ReLU), which is nowadays the default choice for many types of neural networks as it often achieves better performance than other methods [34], is used as activation function. Lastly, *Adam* algorithm [35] is used as optimizer. The training process converges to MSE=0.38, for both training and test sets, in less than 10 epochs, as depicted in Fig. 5.

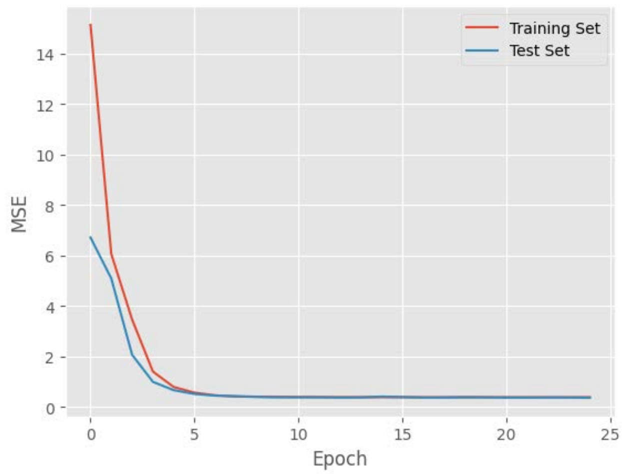

FIGURE 5. Training Epochs vs. MSE.

The residual MSE=0.38 yields to RMSE=0.61s that allows us to predict the calculation time with sufficient accuracy for the purpose of this platform. In order to clearly show the effectiveness of the network after training, we predict the calculation time of the last 100 samples of the test set and we plot both actual and predicted values in Fig. 6. In particular, Fig. 6 shows that the proposed platform is able to fully capture the time band in which a specific node is busy and, consequently, to select an alternative free node to calculate a specific lens design instance in the least possible time.

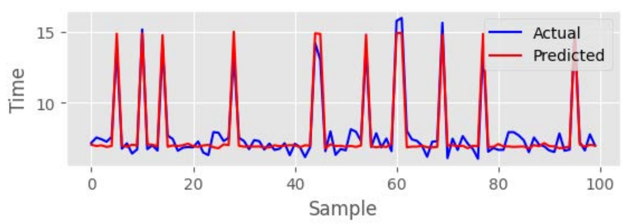

FIGURE 6. Effectiveness of the time prediction ANN.

D. DISCUSSION

In this subsection we compare the proposed BPM platform with two existing LDS design software: IOT[3] that is a monolithic software package to be installed in a lens manufacturer calculation server; ProCrea[4] that is a client-server architecture providing calculation service through a centralized platform.

[3]https://www.iot.es/
[4]https://www.procreatech.com/

The mentioned solutions have the following limitations.

- **Lack of flexibility:** The two vendors provide ten different lens designs for single vision and progressive lenses. The availability of new designs for the lens manufacturer only depends on their updates. It is not easy for a lens designer to make a new design available to the market without relying on one of these legacy solutions.
- **The instances are executed one by one in a queue:** therefore, it is not possible to parallelize calculations. Huge queues require a long time to be processed and product lead time is inevitably affected.
- **Lack of reliability:** in the case of IOT, any fault of the local server would result in a production stop; in the case of ProCrea, the fault of the centralized platform would prevent on-premise clients to make new calculations.
- **Lack of security:** The details and the accounting of calculated jobs are stored either in the local on-premise server or in the centralized platform database. There's no guarantee that these data cannot be damaged, lost or altered and therefore the integrity of economic transactions could be affected.
- **Constrained requirements:** ProCrea client software is *Java* based and it requires a specific version of *Java Runtime Environment*. IOT, which is *Microsoft .NET* based, makes calculations locally; thus, calculation time strictly depends on the amount of CPU and memory on the on-premise server.

By implementing a LDS system in the proposed Blockchain platform, the following improvements are achieved.

1) COLLABORATION

We provide a shared and decentralized platform on which multiple Lens Designers can make their algorithms available in form of self-contained Docker images. Consequently, computational resources providers can register as calculators and provide their service. The manufacturer uses always the same client for all lens designs. The integration of new or updated designs is therefore easier, as it does not require any structural change or any significant reconfiguration of the environment.

2) PERFORMANCE

The provided solution is distributed and fault tolerant. Indeed, the manufacturer can run multiple calculations, that are executed simultaneously on different runners. Hence, the performances in term of computational time and reliability improve.

3) CYBERSECURITY

Since the Blockchain is tamper-proof, the economic transactions of the fees charged to the manufacturer and paid to lens designers and surface calculators cannot be altered. The use of Ethereum platform gives the additional option to finalize transactions right on the chain by implementing a dedicated ERC-20 token [36].

4) COSTS

Since the cloud storage service has typically low fares, a long-term storage is guaranteed for input and output files. This feature also prevents data loss on manufacturer's side. In addition, when the Blockchain is implemented in *Consortium* fashion, such as in the proposed platform, the cost of executing transactions in Ethereum is controllable and can be even zeroed.

However, every Blockchain based solution has some limitations that need to be investigated. For example, in terms of scalability, the number of transactions per second is strictly dependent on the consensus algorithm that is implemented in the system. In that respect, since the adopted Blockchain is implemented in *Consortium* fashion, the consensus algorithm is scalable and does not require the time consuming operations of mining-based algorithms. In addition, from the transactional privacy perspective, although the core of the intellectual property is in the Docker images and in the file stored in the Cloud Storage area, the content of transactions is visible to all users who have access to the Blockchain. Hence, the implementation of a complex Blockchain architecture such as *HyperLedger Fabric*, which can limit the user access to the transactions, would benefit the proposed system.

V. CONCLUSION

The Blockchain is a consolidated paradigm shift in software architectures. The increasing interest in this technology over the last decade has shown that it can be adopted not only for financial applications such as cryptocurrencies, but also in industry environments and specifically in Cloud Manufacturing to increase the efficiency of production process and take the benefits of a decentralized approach and a tamper-proof ledger.

This work proposes a novel platform for improving digital processes in an Ethereum based Blockchain enviroment in which we integrate two of the most popular cloud technologies: Docker and Cloud Storage. In detail, Docker, which in a typical Blockchain architecture is usually employed to run the software components of the Blokchain, in this platform runs process instances using input files stored in Cloud Storage in a distributed and flexible computational platform. We identify and describe the roles of *Process Owner*, *Process Consumer* and *Process Runner*, whose differences and cooperation are the fundamentals of the proposed platform.

Process life-cycle is implemented in the main Smart Contract, while the individual process core logic is packed in a Docker image enabling any generic host to perform an heterogeneous set of tasks. A simple task assignment problem, implemented through an ANN approach, is also introduced for the purpose of improving the performance by decreasing instances execution times.

Finally, we provide a case study regarding the implementation of this platform in Ophthalmic Lenses Manufacturing environment and specifically in Lens Design Systems. Two results are enlightened. First, we show that no significant changes are needed on the industrial process to implement the new platform. Second, we underline the benefits of moving from a legacy static and centralized approach to a flexible decentralized, distributed, secure and collaborative platform on which multiple lens designers can provide their calculation algorithms while lens manufacturers can run different calculations simultaneously on registered calculators.

In future work, we plan to implement this architecture in different blockchain platforms such as *Hyperledger Fabric*, to improve the task assignment problem and deepen some aspects related to transactional privacy and intellectual property preservation.

REFERENCES

[1] X. Xu, "From cloud computing to cloud manufacturing," *Robot. Comput.-Integr. Manuf.*, vol. 28, no. 1, pp. 75–86, Feb. 2012.

[2] (May 2021). *Global Cloud and Data Center Spending 2020 | Statista*. Accessed: May 7, 2021. [Online]. Available: https://www.statista.com/statistics/1114926/enterprise-spending-cloud-and-data-centers

[3] M. K. Lim, W. Xiong, and Z. Lei, "Theory, supporting technology and application analysis of cloud manufacturing: A systematic and comprehensive literature review," *Ind. Manage. Data Syst.*, vol. 120, no. 8, pp. 1585–1614, Jul. 2020.

[4] C. Esposito, A. Castiglione, B. Martini, and K.-K.-R. Choo, "Cloud manufacturing: Security, privacy, and forensic concerns," *IEEE Cloud Comput.*, vol. 3, no. 4, pp. 16–22, Jul. 2016.

[5] F. Casino, T. Dasaklis, and C. Patsakis, "A systematic literature review of blockchain-based applications: Current status, classification and open issues," *Telematics Inform.*, vol. 36, pp. 55–81, Mar. 2018.

[6] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F.-Y. Wang, "Blockchain-enabled smart contracts: Architecture, applications, and future trends," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 11, pp. 2266–2277, Nov. 2019.

[7] U. Bodkhe, S. Tanwar, K. Parekh, P. Khanpara, S. Tyagi, N. Kumar, and M. Alazab, "Blockchain for industry 4.0: A comprehensive review," *IEEE Access*, vol. 8, pp. 79764–79800, 2020.

[8] J. Leng, S. Ye, M. Zhou, J. L. Zhao, Q. Liu, W. Guo, W. Cao, and L. Fu, "Blockchain-secured smart manufacturing in industry 4.0: A survey," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 1, pp. 237–252, Jan. 2020.

[9] J. E. Kasten, "Engineering and manufacturing on the blockchain: A systematic review," *IEEE Eng. Manag. Rev.*, vol. 48, no. 1, pp. 31–47, Mar. 2020.

[10] J. Leng, G. Ruan, P. Jiang, K. Xu, Q. Liu, X. Zhou, and C. Liu, "Blockchain-empowered sustainable manufacturing and product lifecycle management in industry 4.0: A survey," *Renew. Sustain. Energy Rev.*, vol. 132, Oct. 2020, Art. no. 110112.

[11] J. Leng, D. Yan, Q. Liu, K. Xu, J. Zhao, R. Shi, L. Wei, D. Zhang, and X. Chen, "ManuChain: Combining permissioned blockchain with a holistic optimization model as bi-level intelligence for smart manufacturing," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 1, pp. 182–192, Jan. 2020.

[12] T. Hewa, A. Braeken, M. Liyanage, and M. Ylianttila, "Fog computing and blockchain-based security service architecture for 5G industrial IoT-enabled cloud manufacturing," *IEEE Trans. Ind. Informat.*, vol. 18, no. 10, pp. 7174–7185, Oct. 2022.

[13] J. A. Garcia-Garcia, N. Sanchez-Gomez, D. Lizcano, M. J. Escalona, and T. Wojdynski, "Using blockchain to improve collaborative business process management: Systematic literature review," *IEEE Access*, vol. 8, pp. 142312–142336, 2020.

[14] T. Allweyer, *BPMN 2.0: Introduction to the Standard for Business Process Modeling*. Norderstedt, Germany: Books on Demand, 2010.

[15] O. López-Pintado, L. García-Bañuelos, M. Dumas, and I. Weber, "Caterpillar: A blockchain-based business process management system," in *Proc. BPM*, 2017, pp. 1–5.

[16] T. Ghimire, A. Joshi, S. Sen, C. Kapruan, U. Chadha, and S. K. Selvaraj, "Blockchain in additive manufacturing processes: Recent trends & its future possibilities," *Mater. Today, Proc.*, vol. 50, pp. 2170–2180, Jan. 2022.
[17] A. Haridas, A. A. Samad, D. Vysakh, and V. Pathari, "A blockchain-based platform for smart contracts and intellectual property protection for the additive manufacturing industry," in *Proc. IEEE Int. Conf. Signal Process., Informat., Commun. Energy Syst. (SPICES)*, Mar. 2022, pp. 223–230.
[18] (Jan. 2021). *Ethereum Development Documentation*. Accessed: Jan. 21, 2021. [Online]. Available: https://ethereum.org/en/developers/docs
[19] (Jan. 2021). *Docker*. Accessed: Jan. 22, 2021. [Online]. Available: https://docs.docker.com/get-started/overview
[20] W. Zeng, Y. Zhao, K. Ou, and W. Song, "Research on cloud storage architecture and key technologies," in *Proc. 2nd Int. Conf. Interact. Sci. Inf. Technol., Culture Hum. (ICIS)*, New York, NY, USA, 2009, pp. 1044–1048.
[21] C. Anderson, "Docker [software engineering]," *IEEE Softw.*, vol. 32, no. 3, p. 102, Apr. 2015.
[22] A. Halty, R. Sánchez, V. Vázquez, V. Viana, P. Piñeyro, and D. A. Rossit, "Scheduling in cloud manufacturing systems: Recent systematic literature review," *Math. Biosci. Eng.*, vol. 17, no. 6, pp. 7378–7397, 2020.
[23] G. Volpe, A. M. Mangini, and M. P. Fanti, "An architecture for digital processes in manufacturing with blockchain, Docker and cloud storage," in *Proc. IEEE 17th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2021, pp. 39–44.
[24] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.
[25] G.-T. Nguyen and K. Kim, "A survey about consensus algorithms used in blockchain," *J. Inf. Process. Syst.*, vol. 14, no. 1, pp. 101–128, Feb. 2018.
[26] R. Koulu, "Blockchains and online dispute resolution: Smart contracts as an alternative to enforcement," *SCRIPTed-J. Law, Technol. Soc.*, vol. 13, no. 1, pp. 40–69, 2016.
[27] (Jan. 2021). *Solidity Documentation*. Accessed: Feb. 22, 2021. [Online]. Available: https://docs.soliditylang.org/en/v0.8.1
[28] (Feb. 2021). *Vyper Documentation*. Accessed: Feb. 22, 2021. [Online]. Available: https://vyper.readthedocs.io/en/stable
[29] M. Dumas, M. La Rosa, J. Mendling, and H. A. Reijers, *Fundamentals of Business Process Management*. Berlin, Germany: Springer, 2013.
[30] (Feb. 2021). I. Barinov, V. Baranov, and P. Khahulin. *POA Network Whitepaper*. Accessed: Feb. 21, 2021. [Online]. Available: https://github.com/poanetwork/wiki/wiki/POA-Network-Whitepaper
[31] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*. White Plains, NY, USA: Longman, 1999.
[32] (Aug. 2019). *DCS—Data Communication Standard, V.3.12*. Accessed: May 10, 2021. [Online]. Available: https://www.thevisioncouncil.org/sites/default/files/DCS-v3_12_FINAL_v2.pdf
[33] D. Harris and S. Harris, *Digital Design and Computer Architecture*, 2nd ed. Oxford, U.K.: Morgan Kaufmann, 2012.
[34] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14rh Int. Conf. Artif. Intell. Statist.*, vol. 15, G. Gordon, D. Dunson, and M. Dudík, Eds. Fort Lauderdale, FL, USA, Apr. 2011, pp. 315–323.
[35] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.
[36] (Feb. 2021). *ERC-20 Token Standard | Ethereum.org*. Accessed: Feb. 11, 2021. [Online]. Available: https://ethereum.org/en/developers/docs/standards/tokens/erc-20

GAETANO VOLPE received the M.Sc. degree in computer and automation engineering from the eCampus University of Novedrate, Italy, in 2020. He is currently pursuing the Ph.D. degree with the Laboratory of Control and Automation, Polytechnic University of Bari. He also works as an Independent Consultant for private companies in the field of cybersecurity, IT infrastructures, and software architectures. His research interests include cybersecurity topics, blockchains in manufacturing environments, discrete-event systems, and Petri nets.

AGOSTINO MARCELLO MANGINI (Member, IEEE) received the Laurea degree in electronics engineering and the Ph.D. degree in electrical engineering from the Polytechnic University of Bari, Bari, Italy, in 2003 and 2008, respectively. He has been a Visiting Scholar with the University of Zaragoza, Zaragoza, Spain. He is currently an Associate Professor with the Department of Electrical and Information Engineering, Polytechnic University of Bari. He has authored or coauthored over 90 printed publications. His current research interests include modeling, simulation, and control of discrete-event systems, Petri nets, supply chains and urban traffic networks, distribution and internal logistics, management of hazardous materials, management of drug distribution systems, and healthcare systems. He was on the Program Committees of the 2007–2015 IEEE International SMC Conference on Systems, Man, and Cybernetics and the 2009 IFACWorkshop on Dependable Control of Discrete Systems. He was on the Editorial Board of the 2017 IEEE Conference on Automation Science and Engineering.

MARIA PIA FANTI (Fellow, IEEE) received the Laurea degree in electronic engineering from the University of Pisa, Pisa, Italy, in 1983. Since 1983, she has been with the Department of Electrical and Information Engineering, Polytechnic University of Bari, Italy, where she is currently a Full Professor of system and control engineering and the Chair of the Laboratory of Automation and Control. She was a Visiting Researcher at the Rensselaer Polytechnic Institute, Troy, NY, USA, in 1999. Her research interest includes management and modeling of complex systems, such as transportation, logistics and manufacturing systems, discrete event systems, Petri nets, consensus protocols, and fault detection. She has published more than 315 articles and two textbooks on her research topics. She was a Member At Large of the Board of Governors of the IEEE Systems, Man, and Cybernetics Society. She is also a member of the AdCom of the IEEE Robotics and Automaton Society. She is also the Chair of the Technical Committee on Automation in Logistics of the IEEE Robotics and Automation Society. She was the General Chair of the 2011 IEEE Conference on Automation Science and Engineering, the 2017 IEEE International Conference on Service Operations and Logistics, and Informatics, and the 2019 IEEE Systems, Man, and Cybernetics Conference. She was a Senior Editor of the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING. She is an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS.