

# On-Chain Global Maintenance Services\*

Alessandro Bellini<sup>3</sup>, Antonio Bonifacio<sup>3</sup>, Salvatore Esposito De Falco<sup>2</sup>,  
Simone Naldini<sup>3</sup>, Francesco Pacileo<sup>2</sup>, Diego Pennino<sup>1</sup>, Maurizio Pizzonia<sup>1</sup>,  
Domenico Sardanelli<sup>2</sup>, Andrea Vitaletti<sup>2</sup>, Pietro Vito<sup>2</sup> and Marco Zecchini<sup>2,\*</sup>

<sup>1</sup>Università degli Studi Roma Tre, Dipartimento di Ingegneria, Sezione Informatica e Automazione, Via della Vasca Navale 79, 00146 Rome, Italy;

<sup>2</sup>Sapienza Università di Roma, Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Via Ariosto 25, 00185 Rome, Italy;

<sup>3</sup>Mathema, Via Torricoda 29, 50142 Florence, Italy;

## Abstract

Facility management deals with all activities that are not core business for a company and are consequently outsourced to specialized companies. Maintenance is a fundamental activity in facility management and it is often handled by *Global Maintenance Services* (GMS) where some maintenance activities are delegated by the company to service providers and are remunerated according to measurable results expressed as Key Performance Indicators. In this context, it would be desirable to have information systems trustable by all involved actors. In this paper, we discuss the design of a blockchain solution capable to support a GMS on-chain. We first introduce the GMS concept and how it is related to the Principal-Agent relationship, then we show a reference architecture to implement GMS on-chain. We discuss a use case of on-chain GMS in a hospital showing how smart contracts and oracles can be used in this context. We present the advantages of this approach and we discuss the open problems for realizing a proof-of-concept.

## Keywords

Maintenance, Blockchain, Oracle, Smart Contract, Sensors

---

DLT 2022: 4th Distributed Ledger Technology Workshop, June 20, 2022, Rome, Italy

\*This research was partially funded by Sapienza Ateneo Research grant “La disintermediazione della Pubblica Amministrazione: il ruolo della tecnologia blockchain e le sue implicazioni nei processi e nei ruoli della PA”. This research was partially funded by Italian Minister of Economic Development (MISE) grant “ReASSET: Piattaforma Maas (Maintenance as a Service) Blockchain-IoT per la gestione operativa in tempo reale degli asset tecnico-impianistic”. This research was partially funded by POR FESR LAZIO 2014 – 2020, call for “Gruppi di ricerca 2020”. Det. n. G04052 of Apr. 4th, 2019, under the “LazioChain” project, CUP F85F21001550009 - POR project code A0375E0116.

\*Corresponding author.

✉ salvatore.espositodefalco@uniroma1.it (S. E. D. Falco); francesco.pacileo@uniroma1.it (F. Pacileo); pennino@ing.uniroma3.it (D. Pennino); pizzonia@ing.uniroma3.it (M. Pizzonia); domenico.sardanelli@uniroma1.it (D. Sardanelli); vitaletti@diag.uniroma1.it (A. Vitaletti); piro.vito@uniroma1.it (P. Vito); zecchini@diag.uniroma1.it (M. Zecchini)

ORCID 0000-0001-5339-4531 (D. Pennino); 0000-0001-8758-3437 (M. Pizzonia); 0000-0003-1074-5068 (A. Vitaletti); 0000-0002-2280-9543 (M. Zecchini)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

## 1. Introduction

*Facility Management* (FM) deals with managing the facilities, namely all assets, both tangible and intangible, that support a company's core business making the life of occupants of residential buildings, shops, offices or factories more pleasant and safe. Within FM, a Global Maintenance Service (GMS), is a form of outsourcing contract specifically related to maintenance and based on measurable results. Through a GMS contract, a *client*, or *principal*, entrusts a series of activities aimed at the maintenance of the facilities to a single *primary service provider*, or *agent*, for a well defined period of time. The following elements are relevant to this paper for a GMS contract.

- The contract is based on results. The remuneration is a function of a series of Key Performance Indicators (KPIs) through which it is possible to measure the quality, efficiency and effectiveness of the performed activities.
- There is a working group made up of representatives of the client and the primary service provider, whose function is to ensure the correct start and execution of the project, with particular regard to the implementation of integrated management tools.
- The primary service provider appoints a single manager, with respect to which the client can refer as the sole interlocutor and who has responsibility for the activity of all the personnel involved in the performance of the services covered by the contract. The primary service provider can delegate some activities to *secondary service providers*.

Figure 1 summarizes and clarifies the relations between the parties discussed so far.

Examples of the employment of GMS contract for the maintenance of facilities include the following.

**Lighting.** Energy supply and ordinary and extraordinary maintenance of related systems.

**Real estate assets.** Their ordinary and extraordinary maintenance, plant maintenance, cleaning and surveillance services.

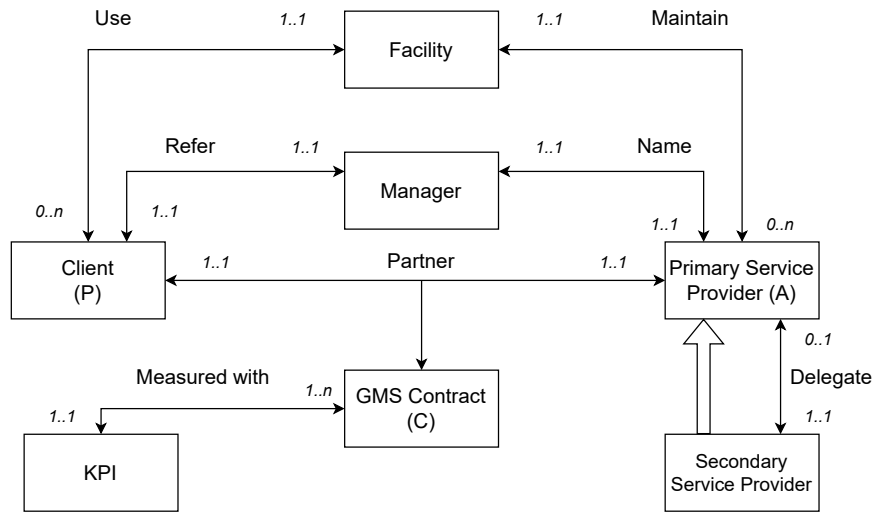
**Green.** Paving, cleaning, cutting of the grass, refurbishment of green areas.

**Heat.** Ensure the heating and air conditioning system including the supply of fuel, gas and electricity.

GMS can be modelled as a relationship between a Principal (P) and an Agent (A) [4], where the principal appoints the agent to act on its behalf for the maintenance of its facilities. According to the GMS, the relationship is governed by a contract (C) based on results measurable by suitable KPIs.

Usually, the client pays the provider either on the basis of measurements declared by the provider or by performing measurements by themselves. In the first approach, the client must trust the provider. In the second approach, the costs of the client for autonomously performing the measurements might be too high with respect to the benefits of the outsourcing approach.

In this paper, we propose an architecture based on blockchain and IoT technologies to address this problem. We also provide details for a sample use case of this approach encompassing oracles to acquire measurements from IoT devices into the blockchain. Our sample use case



**Figure 1:** Diagram of the relations between parties

is taken from a real tender for heat maintenance related to an Italian hospital. We provide examples of smart contract code to realize that use case.

This paper is structured as follows. In Section 2, we provide background notions about blockchain technologies and how they are able to access off-chain data through oracles. In Section 3, we describe the architecture of a blockchain-based GMS and how it benefits from this technology. Finally, Section 4 draws the conclusions of the paper and provides some discussion about open problems.

## 2. Blockchain Background

A *blockchain* is a type of *Distributed Ledger Technology* (DLT) where transactions, new records to the ledger, are recorded according to an immutable order obtained by means of *cryptographic hash functions* that chain the blocks in which transactions are recorded. Unlike a centralized database, a blockchain is decentralized, namely there is no need for a central authority or intermediary for processing, validating, and/or authenticating transactions. A blockchain is typically managed by a set of autonomous *nodes* that collectively create a peer-to-peer (p2p) network adhering to a protocol for inter-node communication and validating new blocks. Nodes do not trust each other and malicious nodes are tolerated, within certain limits that depends on the consensus algorithm. The most common blockchains can be abstracted as a key-value database. For example, in a blockchain implementing a cryptocurrency, keys are *addresses* (or *accounts*), while values are the balances of their *wallets*. In this scenario, a transaction is an operation that transfer some cryptocurrency from a wallet to another. For efficiency reasons, transactions are not confirmed one-by-one but aggregated into *blocks*. Transactions are confirmed when a new block is *created* (or *mined*). The mining of a new block requires to verify that all transactions of the block, considered in the chosen order, comply to certain

*consensus rules* (that depends also on the application domain). When a new block is mined by a node of the network, all the other peers verify that it respects the consensus rules in a process called *validation*. The *consensus* (for more details, see [16, 17, 8]) is the decentralized process by which a block is finally stored in the ledger.

There are two types of blockchains that can be categorized according to the access in reading and writing to the content of the ledger and to the access in participating to the consensus. In public *blockchains*, everyone can read the content of the ledger and propose new transactions that, if successfully validated by the consensus, will be eventually stored in the ledger. On the contrary, in *private* blockchains, users are authenticated and access control allows or denies each user operation as occurs for access control of regular information systems. Similarly, in *permissionless* blockchain every user can participate in the consensus, while in *permissioned* one the participation in the consensus is allowed only to specific users.

While initially blockchain has been primarily conceived to implement cryptocurrency trading, it can now be adopted to realize general-purpose applications through the use of *smart contracts*. They consist of pieces of code that are executed as part of a transaction. In simple terms, in these cases, the blockchain implements a global decentralized virtual machine and smart contracts are the programs running on it.

Smart contracts can process only data that are stored in the blockchain. However, in the GMS use case that we consider in this paper, there is the need of accessing off-chain data. This is possible using an *Oracle* (for more details, see [6]). Oracles are components that allow a blockchain, or a smart contract, to get inputs from outside the blockchain through regular blockchain transactions. There are several oracle services providing APIs to allow smart contracts to access external data. Examples include Chainlink [1], Provable [3], BandChain [5], and Tellor [2].

### 3. GMS On-Chain

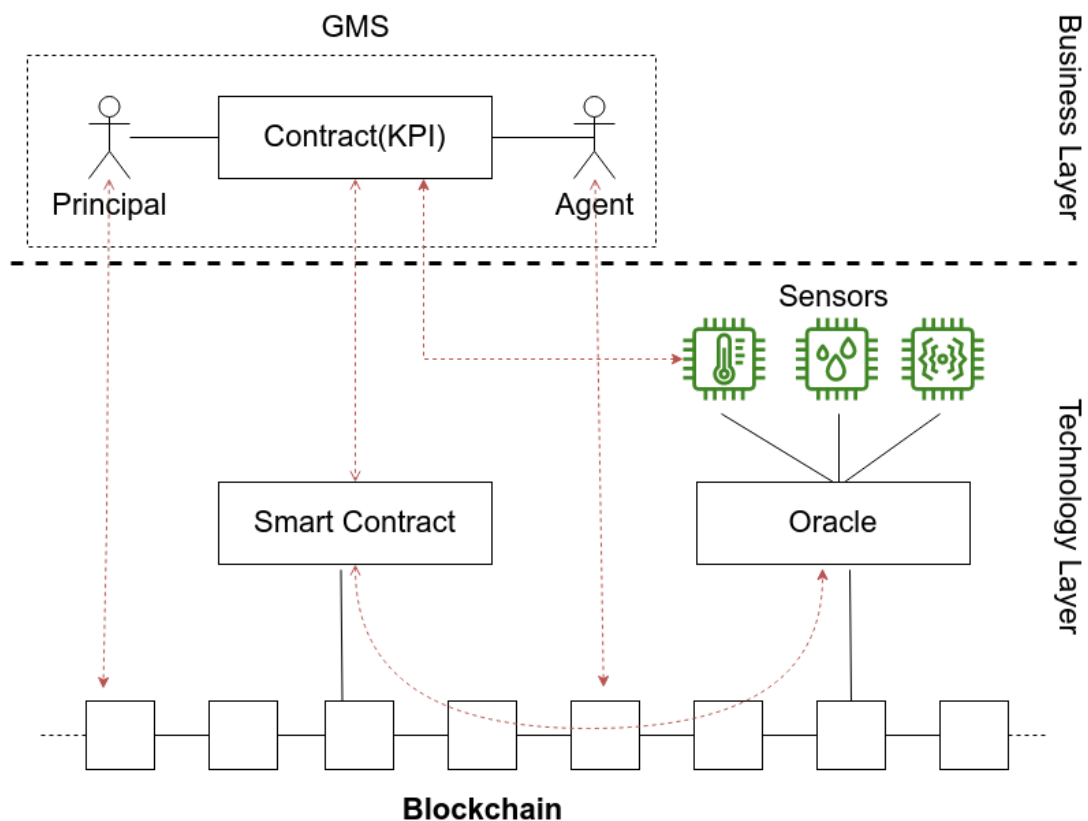
We have seen that GMS can be modeled as a P/A relationship (see Figure 1). In such a relationship, the agent acts on behalf of the principal and should not have a conflict of interest in carrying out its task. An agent may act in its own best interests and in a way that is contrary to the best interests of the principal, generating the so-called *P/A Problem*. This problem typically arises when P has into enough information to directly ensure that A is always acting in P's best interest.

The transparency, immutability, traceability and algorithmic governance offered by *Blockchain* technologies can contribute to mitigate the P/A Problem [10], reducing (or even eliminating) the asymmetry of information and thus facilitating the creation of a genuine net value.

The employment of the blockchain, allows us to envision new models of governance, where trust between the actors is substituted by A and P relying on the consensus within the P2P blockchain infrastructure, i.e. relying on a community rather than on the trust in individual actors. In this perspective, the natural different interests of P/A, at least economically-wise, as well as the participation of different providers competing in the market, are guarantees to the achievement of a real consensus among the parties, even in less open infrastructure such as the permissioned blockchains.

In particular, the Blockchain can provide:

- algorithmic governance autonomously managed by smart contracts capable to implement decentralized decision-making processes providing the highest guarantee of impartiality to all the involved stakeholders;
- a transparent and immutable bidding process to select the service providers (i.e. Agents);
- a minimization of the information coordination costs on a shared infrastructure, making the organization's data accessible to new customers and suppliers;
- a reduction of verification costs, namely costs involved in verifying the transactions between Principal and Agent.
- a reduction of intermediation costs, i.e. the costs due to the certification activities by a third party, external to the contractors.



**Figure 2:** At the Business layer, the relationship between the Principal and Agent modeling the GMS, is regulated by a contract based on performance measures defined by suitable KPIs. The Business components are mapped into their technical counterparts in the Technical Layer. Principal and Agents are two entities on the Blockchain. The remuneration of the Agent (i.e. a transaction from the Principal to the Agent) is governed by a Smart Contract according to specific sensors' observations measuring the KPIs. To access sensors' data, the Smart Contract interacts with an Oracle.

### 3.1. Modelling the P/A Relationship On-Chain

The architecture of the on-chain GMS modeled as P/A relationship is represented in Figure 2. Here we assume that both the Principal and the Agent are entities on-chain identified by an address. Note that we currently assume the pseudoanonymity sufficient to carry on the economic transaction behind the GMS contract, however, while this is technically more convenient, we have not yet properly investigated all the complexity of managing identity on-chain [12] in particular when norms, laws, and regulations must be satisfied.

In the following, we will use Solidity code sketches to illustrate the structure and main components of the necessary smart contracts. Smart contracts of the following examples refer to the hospital heating use-case described in Section 3.2. The GMS contract is translated into a smart contract (see Listing 1). The function *payAgent*, at Line 23, performs the payment if the KPI are satisfied. In this case, this occurs when the level of CO emission measured by a sensor is below a given threshold (see line 27). To access the data of the IoT sensors [13], the smart contract interacts with an Oracle [7] as sketched in Listing 2. In this case, we use the Provable Thing Oracle [3], that provides access to off-chain data to a number of Blockchain Technologies, including Ethereum, EOS, R3 Corda and Hyperledger Fabric.

In some cases, primary service providers can take advantage of sub furniture provided by secondary service providers. Also in this case, a smart contract can be employed to manage this relationship as a P/A one as shown in Listing 3.

### 3.2. Use-Case: GMS for an Hospital

In this section, we show how the components of Figure 2 can be mapped to a real tender specifications document [11] which defines the modalities through which the Bianchi Melacrino Morelli Hospital in Reggio Calabria, Italy, intends to entrust the ordinary and extraordinary maintenance service of the buildings, the technical plants, and the furniture to a primary service provider for three years.

Art. 6 of [11], details the reference maintenance plan, and provides a number of sheets that list all the maintenance operations that the primary service provider has to perform. For the sake of simplicity, we simply consider the sheet in Table 1.

Maintenance guide N. 05		
Plant or Facility type	Operations performed by the maintenance service	Cyclicity
thermomechanical plants	Combustion control according to Legislative Decree 152/06	Annual

**Table 1**

Maintenance operations carried by the primary service provider

The sheet mandates that every year, the combustion of thermo-mechanical systems must be checked according to the D.L.gs 152/06 [14] regulation.

**Principal and Agent** The Principal is the hospital company. The Agent is the primary service provider taking care of the maintenance of the hospital facilities as foreseen in the GMS according to the specification provided [11] and possibly taking advantage of secondary service providers.

**Measurable Contract** Art. 286 of D.Lgs 152/06 [14] defines the threshold values and the measurement modalities to check the emissions. In the following, we summarize the most relevant elements for the considered use case.

- The atmospheric emissions of civil thermal plants with nominal thermal power above the threshold value must comply with the limit values set out in part III of Annex IX to part five of D.Lgs 152/06 (see Table 2).
- The emission values of the plants must be checked at least annually by the person in charge of the operation and maintenance of the plant during normal inspection and maintenance operations. The measured values, with the indication of the relative dates, of the measurement methods used and of the person who carried out the measurement, must be attached to the plant logbook.
- For the purposes of sampling, analysis and assessment of emissions from thermal plants referred to in paragraph 1, the methods provided for in part III of Annex IX (see Table 2) are applied.
- The installer verifies compliance with the emission limit values.

	Installed Electrical Rated Output (MW)			
	[1] > 0,15 ≤ 3	> 3 ≤ 6	> 6 ≤ 20	> 20
Total dust	100mg/Nm <sup>3</sup>	30mg/Nm <sup>3</sup>	30mg/Nm <sup>3</sup>	30mg/Nm <sup>3</sup>
Total organic carbon (TOC)	n.a.	n.a.	30mg/Nm <sup>3</sup>	20mg/Nm <sup>3</sup> 10mg/Nm <sup>3</sup>
Carbon monoxide (CO)	350mg/Nm <sup>3</sup>	300mg/Nm <sup>3</sup>	250mg/Nm <sup>3</sup> 150mg/Nm <sup>3</sup>	200 100mg/Nm <sup>3</sup>
Nitrogen oxides (expressed in NO <sub>2</sub> )	500mg/Nm <sup>3</sup>	500mg/Nm <sup>3</sup>	400mg/Nm <sup>3</sup> 300mg/Nm <sup>3</sup>	400mg/Nm <sup>3</sup> 200mg/Nm <sup>3</sup>
Sulphur oxide (expressed in SO <sub>2</sub> )	200mg/Nm <sup>3</sup>	200mg/Nm <sup>3</sup>	200mg/Nm <sup>3</sup>	200mg/Nm <sup>3</sup>
[1] For plants with a rated thermal input equal to or greater than 0.0035 MW and no greater than 0.15 MW, it is applied an emission value for total dust of 200				

**Table 2**

An example of requirements and KPIs for a tender. Values shown in this table are taken from the Italian regulation [14] (see text), which applies to the the tender considered in Section 3.2.

According to the GMS, the smart contract will (a) verify the satisfaction of the requirements for the emissions according to the measured KPIs and the limit defined in [14] and (b) perform the payments to the service providers.

**KPI and sensors** The KPI to measure the satisfaction of the contract terms is clearly defined in Table 2. As an example, if we consider the Total Suspended Particles and a heating system with nominal installed power between 3 and 6 MW, the threshold is 30mg/Nm<sup>3</sup>.

The measurements of the satisfaction of the KPI are provided by the sensors of Table 2, namely Total Suspended Particles, Total Organic Carbon, Carbon Monoxide, Nitrogen Oxides, Sulfur Oxides. Sensors should sample the environment with accuracy and a sampling period defined by the regulation.

**Smart Contract examples.** Listing 1 illustrates how Principal and Agent can manage their collaboration with a smart contract. “PayCOContract” accesses external data through the contract oracle in Listing 2. Finally, “Subcontract” in Listing 3, demonstrates how a primary service provider can delegate a secondary service provider on a blockchain through a specific smart contract.

```

1  pragma solidity ^0.4.22;
2  import "./ExampleContract_CO_Oracle.sol";
3
4  contract PayCOContract {
5
6      ExampleContract_CO_ORACLE private OracleContract;
7      address public P;
8      address public A;
9      uint public CO_Threshold;
10     uint public lastPayment;
11     uint public payment;
12     uint public INTERVAL = 10;
13
14     constructor (address _OracleContract, address _P, address _A, uint _CO, uint _payment) {
15         OracleContract = ExampleContract_CO_ORACLE(_OracleContract);
16         P = _P;
17         A = _A;
18         CO_Threshold = _CO;
19         payment = _payment;
20         lastPayment = block.number;
21     }
22
23     function payAgent() payable {
24         require(block.number > lastPayment + INTERVAL);
25         require(msg.value == payment);
26         require(msg.sender == P);
27         if (stringToUint(OracleContract.CO) < CO_Threshold) revert();
28
29         A.transfer(msg.value);
30     }
31 }
32 }

```

Listing 1: A sketch of a smart contract to execute the payment from the Principal to Agent if specific conditions are met. For the sake of brevity “stringToUint” function is omitted but it casts a string into an unsigned integer in Solidity.

```

1  pragma solidity ^0.4.22;
2  import "github.com/provable-things/ethereum-api/provableAPI_0.4.25.sol";
3
4  contract ExampleContract_CO_ORACLE is usingProvable {
5
6      string public CO;
7      event LogConstructorInitiated(string nextStep);
8      event LogCOUpdated(string CO);
9      event LogNewProvableQuery(string description);
10 }

```



```

11 function ExampleContract() payable {
12     LogConstructorInitiated("Constructor was initiated. Call 'updateCO()' to send the
        Provable Query.");
13 }
14
15 function __callback(bytes32 myid, string result) {
16     if (msg.sender != provable_cbAddress()) revert();
17     CO = result;
18     LogCOUpdated(result);
19 }
20
21 function updateCO() payable {
22     if (provable_getPrice("URL") > this.balance) {
23         LogNewProvableQuery("Provable query was NOT sent, please add some ETH to cover
            for the query fee");
24     } else {
25         LogNewProvableQuery("Provable query was sent, standing by for the answer..");
26         provable_query("URL", "json(https://api.sensor.it).CO");
27     }
28 }
29 }

```

Listing 2: A sketch of a smart contract to get data from a CO sensor by Provable Things Oracles. Provable Things provide oracles for a number of Blockchain Technologies, including Ethereum, EOS, R3 Corda and Hyperledger Fabric

```

1 pragma solidity ^0.4.22;
2 import "./PayCOContract.sol";
3
4 contract Subcontractor is PayCOContract {
5     address public Secondary;
6     uint public SecondaryPayment;
7
8     constructor (address _OracleContract, address _P, address _A, uint _CO, uint _payment,
        address _S, uint _secondary_payment) PayCOContract(_OracleContract, _P, _A, _CO,
        _payment) {
9         Secondary = _S;
10        SecondaryPayment = _secondary_payment;
11    }
12
13    function paySecondary() public {
14        require(block.number > lastPayment + INTERVAL);
15        require(msg.value == SecondaryPayment);
16        require(msg.sender == A);
17        if (stringToUint(this.OracleContract.CO) < CO_Threshold) revert();
18
19        Secondary.transfer(msg.value);
20    }

```

Listing 3: A sketch of a smart contract to execute the payment from the Primary Service (A) to Secondary Service provider if specific conditions are met. Note that, since this is a special case of contract where the agent delegates to another entity the management of some facility, Subcontract inherits the main contract "PayCOContract"

## 4. Conclusions

In this paper, we discuss the design of a blockchain solution capable to support the Global Maintenance Service on-chain, the implementation of the Principal/Agent relationship and how modeling of the GMS on-chain provides several advantages. The transparency of Blockchain can eliminate the asymmetry of information and consequently, it reduces (or even eliminate) the P/A problem and allows a transparent and immutable bidding process for the selection of Agents. The algorithmic governance autonomously managed by smart contracts is capable to implement decentralized decision-making processes providing the highest guarantee of impartiality to all the involved stakeholders and the shared blockchain infrastructure allows us to minimize the information coordination, verification and intermediation costs.

All these arguments encourage us to proceed in this investigation, however, a number of relevant questions still need to be properly handled.

The selection of the most suitable blockchain technology is the first relevant issue. Public/permissionless blockchains provide the highest guarantees but could be difficult to be implemented in an industrial context where some information is necessarily sensitive and private. However, the natural different interests of Principal and Agents, at least economically-wise, as well as the participation of different providers competing in the market, are guarantees to the achievement of a real consensus among the parties, even in less open infrastructure such as the permissioned blockchains.

The employment of the blockchain, allows us to envision new models of governance, where trust to individuals is overcome by consensus from a community. However, it is not yet clear what are the implications in legal terms of this new governance in particular in terms of accountability. More in general, the applicability of the algorithmic governance provided by the blockchain should be better investigated in view of current laws, norms and regulations [9].

Finally, the concept of identity on-chain should be better explored, also in view of the Self-Sovereign Identity [15] concept.

## References

- [1] Blockchain Oracles for Hybrid Smart Contracts | Chainlink. <https://chain.link/> [Online; accessed 18. Gen. 2022].
- [2] Tellor. <https://tellor.io/> [Online; accessed 18. Gen. 2022].
- [3] Provable - blockchain oracle service, enabling data-rich smart contracts, 2019. <https://provable.xyz> [Online; accessed 26. Jan. 2022].

- [4] Agency, February 2022. <https://www.law.cornell.edu/wex/agency>, [Online; accessed 28. Apr. 2022].
- [5] Band Protocol - Cross-Chain Data Oracle, 2022. <https://bandprotocol.com/bandchain> [Online; accessed 26. Jan. 2022].
- [6] Hamda Al-Breiki, Muhammad Habib Ur Rehman, Khaled Salah, and Davor Svetinovic. Trustworthy blockchain oracles: review, comparison, and open research challenges. *IEEE Access*, 8:85675–85685, 2020.
- [7] Giulio Caldarelli. Understanding the blockchain oracle problem: A call for action. *Information*, 11(11), 2020.
- [8] Md Sadek Ferdous, Mohammad Javed Morshed Chowdhury, and Mohammad A. Hoque. A survey of consensus algorithms in public blockchain systems for crypto-currencies. *Journal of Network and Computer Applications*, 182:103035, 2021.
- [9] Philipp Hacker, Ioannis Lianos, Georgios Dimitropoulos, and Stefan Eich, editors. *Regulating Blockchain: Techno-Social and Legal Challenges*. Oxford University Press, Oxford, 2019.
- [10] Wulf A Kaal. Blockchain solutions for agency problems in corporate governance. In *Information for Efficient Decision Making: Big Data, Blockchain and Relevance*, pages 313–329. World Scientific, 2021.
- [11] Azienda ospedaliera Bianchi Malacrino di Reggio Calabria. Servizio di manutenzione degli immobili e degli impianti dell'azienda ospedaliera bianchi malacrino di reggio calabria. [https://ospedaler.it/files/old/CSA\\_manut\\_%201\\_%2014\\_.pdf](https://ospedaler.it/files/old/CSA_manut_%201_%2014_.pdf)[Online; accessed April 2022].
- [12] Diego Pennino, Maurizio Pizzonia, Andrea Vitaletti, and Marco Zecchini. Efficient certification of endpoint control on blockchain. *IEEE Access*, 9:133309–133334, 2021.
- [13] Diego Pennino, Maurizio Pizzonia, Andrea Vitaletti, and Marco Zecchini. Blockchain as iot economy enabler: A review of architectural aspects. *Journal of Sensor and Actuator Networks*, 11(2), 2022.
- [14] Gazzetta Ufficiale. Decreto legislativo 3 aprile 2006, n. 152. "norme in materia ambientale" pubblicato nella gazzetta ufficiale n. 88 del 14 aprile 2006 - supplemento ordinario n. 96. <https://web.camera.it/parlam/leggi/deleghe/06152dl5.htm>[Online; accessed April 2022].
- [15] Fennie Wang and Primavera De Filippi. Self-sovereign identity in a globalized world: Credentials-based identity systems as a driver for economic inclusion. *Frontiers in Blockchain*, 2, 2020.
- [16] Wenbo Wang, Dinh Thai Hoang, Peizhao Hu, Zehui Xiong, Dusit Niyato, Ping Wang, Yonggang Wen, and Dong In Kim. A survey on consensus mechanisms and mining strategy management in blockchain networks. *Ieee Access*, 7:22328–22370, 2019.
- [17] Huanliang Xiong, Muxi Chen, Canghai Wu, Yingding Zhao, and Wenlong Yi. Research on progress of blockchain consensus algorithm: A review on recent progress of blockchain consensus algorithms. *Future Internet*, 14(2):47, 2022.