Using Blockchains to Implement Distributed Measuring Systems

Wilson S. Melo Jr, Alysson Bessani, Nuno Neves, Altair Santin and Luiz F. R. C. Carmo

***Abstract*—In recent years, measuring instruments have become quite complex due to the integration of embedded systems and software components and the increasing aggregation of new features. Consequently, metrological regulation and control require more efforts from notified bodies, becoming slower and more expensive. In this work, we evaluate the use of blockchains as a resource to overcome such challenges. We start with a conceptual model for implementing measuring instruments in a distributed blockchain-based architecture, and compare it with traditional measuring instruments and distributed measuring models discussed in previous works. We also made a security analysis, demonstrating that blockchains-based measuring systems can impact the way measuring instruments are used in consumer relations while improving security and simplifying metrological regulation and control. We implement a vehicle speed measuring system using the Hyperledger Fabric blockchain platform. We evaluate the security and performance of our blockchain-based measuring system by executing tests with data from real speed meter sensors. The results are promising and validate the feasibility of our idea. Finally, we point out the main challenges related to our approach, suggesting alternatives and potential issues to be addressed by future works.**

I. Introduction

Measurement instruments (MI) are used in many application domains including industry, commerce, energy, transportation, health care and environment protection [1]. In Europe alone, MI are responsible for an annual turnover of more than 500 billion Euros [2]. In developing countries, the demand for MI has increased substantially due to the adoption of technologies and methods well established in developed countries [1]. MI also can be seen as fundamental building blocks for new technologies such as internet of things and cyber physical systems (e.g., smart grids) [1], [2], [3], [4], [5], [6].

MI are nowadays quite complex, since they are strongly based on embedded systems and are often connected and accessible by the Internet [2], [3]. This kind of scenario might expose MI to security gaps that can be explored with malicious intent [4], [5]. Legal metrology is responsible for promoting MI metrological assurance, establishing security requirements and technical activities such as type approval, verification and metrological supervision [1]. However, the increasing complexity of MI affects such activities substantially. Type approval requires more effort while verification can involve use cases which are hard to reproduce inside labs. In turn, metrological supervision becomes difficult due to the diversity of MI models, their geographic spread and the limited resources owned by regulatory agencies.

We work with the hypothesis that the aforementioned difficulties should be overcome with alternative approaches that simplify MI design while employing strategies to decentralize metrological supervision. Such idea finds many aspects in common with a new trendy technology: *blockchains* [7]. A blockchain can be described as a distributed data structure which assures information integrity and authenticity while providing a platform for executing self-enforced software procedures, called *smart contracts* [8]. Blockchain solutions have been very successful in financial applications (e.g., Bitcoin and Ethereum), inspiring its use in different applications and knowledge areas [7], [8]. More recently, a few works have proposed blockchain applications in legal metrology, which include decentralized audit, mechanisms for software loading, Public Key infrastructure (PKI) for MI manufacturers and Distributed Measuring Systems (DMS) [9], [10].

In this paper we discuss how blockchains can improve measuring applications, evaluating two main aspects: *distributed measuring* (DM) and *decentralized surveillance*. This paper extends the ideas presented in our previous work [10]. We start from preliminary concepts already consolidated in Legal Metrology about MI regulation and control. Then we explore ideas related to the integration of MI in DMS, proposing a blockchain-based model. Such aspects result in an innovative concept that dissociates the measurement service from the measurement quantity while it improves MI security and makes metrological assurance simpler and less expensive.

Our main contributions can be summarized as follows:

- We introduce the idea of DM using blockchains and describe its advantages when compared to traditional MI and other DM models. To the best of our knowledge, this is the first paper to describe a blockchains-based DMS.
- We propose an architectural model for implementing our idea, showing that MI and blockchains enable a new business model where the measuring process is an independent service, reducing conflicts of interest.
- We present a security analysis, showing that our model improves MI security since it constrains the attacker capabilities, thus simplifying MI regulation and control.
- We develop a practical case study using the Hyperledger Fabric [11] platform. We create a blockchain network that implements legally relevant software using smart contracts for measuring vehicle speed. We also present results that demonstrate the feasibility of our idea.

W. Melo Jr. and Luiz F. R. C. Carmo are with the National Institute of Metrology, Quality and Technology, Duque de Caxias, RJ, Brazil; and with the Federal University of Rio de Janeiro, Brazil.

A. Bessani and N. Neves are with LaSIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal.

A. Santin is with Pontifical Catholic University of Parana, Curitiba, PR, Brazil.

- We point out challenges that shall be addressed in future works using blockchains in measuring applications.

II. Background

A. Legal metrology and MI reliability

Legal metrology embraces MI regulation and control. It is crucial to assure the correctness of measurements [1] and regulate consumer relations [6]. Usually, legal metrology regulations are defined by government agencies or international committees. Regulation directives traditionally establish a set of requirements and activities [1]. The main ones are related to legal control of MI and type approval, which can include documentation and code inspection, validation and verification; and metrological supervision, including quality, market and field surveillance. These activities are usually executed by notified bodies[1] that are designated to assert MI conformity [2], [6].

Legal metrology activities related to electronic and software controlled MI can demand more complex procedures and specialized knowledge. Usually, regulation adopts security requirements and good practices from well-known technical standards [2], [5], [12]. The *OIML D 31(E)* document [13] and *WELMEC Software Guide 7.2* [14] are probably the more widespread standards for software-controlled MI design, deployment and inspection.

The majority of security issues with MI arise from parties seeking undue economic advantages. A classical example occurs in the commerce of measured goods where vendors and consumers have conflicting interests [1]. Malicious vendors can try to maximize profits while malicious consumers can try to minimize prices by frauding measurements. Measurement frauds against MI (such as scales, energy meters and fuel pumps) are very common in developing countries [3], [12]. Attacks can also intend to steal sensitive information and intellectual property [3], [6]. In some cases they even threaten people's physical integrity (e.g., tampering measurements related to medical procedures) [4].

B. Distributed measuring

Distributed measuring (DM), where components are connected through a network, is well studied. Boccardo et al. [4] describes a strategy to simplify MI type approval and supervision activities related to medical MI. Their proposal consists in signing sensing raw data of a sphygmomanometer immediately after analog-to-digital (AD) conversion. Although it is not a DM case, this approach suggests that part of the measurement computing can be done externally to the sphygmomanometer hardware core due to the use of a digital signature to check sensing data integrity and authenticity. Peters et al. [5] describes a MI security framework using virtual machines to separate *legally relevant* (LR) and *non-legally relevant* (NLR) software.[2] The authors propose different virtual machines to execute LR and NLR functions and define secure interfaces for communicating among them. This approach is presented as an alternative to improve security and reduce MI complexity. Additionally, it enables virtualization using different hardware cores and consequently allows the implementation of *Distributed Measuring Systems* (DMS). Lastly, a DM architecture using cloud computing is discussed by Oppermann et al. [6]. The authors present advantages related to IT infrastructure cost-savings and the possibility of MI manufacturers to offer modern interconnected devices and features. They also present a comprehensive example of how to integrate energy meters in a DMS. In contrast, they also point out issues related to communication security, data management, and reliability. Roughly speaking, they assert that the following challenges need to be addressed:

- DM instruments must be as secure as their classical counterparts.
- Large amounts of data will be accumulated in distributed repositories, requiring proper treatment.
- If distributed service providers are considered untrustworthy, then data security is very difficult to assure.

C. Blockchains

Blockchain is an emerging technology which has caught the attention of stakeholders in different industry segments. Initially associated with crypto-currency markets due to Bitcoin popularity [7], blockchain-based architectures have been proposed for a wide set of application areas, including sensor networks, internet of things, smart cities, among others [8].

Conceptually, a blockchain can be regarded as a distributed append-only data structure (designated as *ledger*) which is replicated and shared among a set of network peers [8]. This structure consists of a sequence of blocks where block n is cryptographically linked to the block $n-1$ using a hash function. Consequently, block n cannot be changed without also modifying all subsequent blocks $n+i, ..., n+k$ [15]. Being a decentralized model, blockchains availability does not depend on third parties, which can greatly save costs. In turn, integrity and availability are ensured by consensus among the peers, preventing the whole chain from being modified and requiring an agreement about any block to be appended to the ledger [15], [16]. Blockchain platforms can be classified as *permissionless*, in which anybody can join and participate in the network consensus, or *permissioned*, in which consensus is achieved by a set of known and identifiable peers [16]. Usually, permissioned blockchains consensus protocols expend less computational resources and can reach better transaction latency and throughput.

A blockchain can store virtually any digital asset, from data to self-executing scripts, usually defined as *smart contracts*. This makes blockchains not only a data storage solution but also a complete distributed platform for proper and distributed automated workflow [8]. Once smart contracts are executed at every network peer in an independent and automatic manner, software integrity is achieved from blockchains integrity as a whole.

[1] Notified bodies are public or private parties organized for verifying MI.

[2] OIML D 31(E) and WELMEC 7.2 use LR to designate any component which can affect measuring final results, while NLR cannot do that.

III. Defining MI Security Scope

In this work, we want to evaluate the security level of different measuring systems models and point out the advantages and drawbacks of these models. We are especially interested in MI reliability and the required effort for providing MI metrological assurance. Metrological requirements and activities are very particular for different MI classes. However, a simple set of requirements and activities are representative of most software-controlled MI concerning security properties. From that point of view, we define our MI security scope based on a generic attack model and its respective metrological assurance framework. We described both in this section.

A. Attack model

We consider a simple attack model that can be built from MI use cases, according to OIML D 31(E) and WELMEC 7.2 guides. MI are targeted by malicious entities trying to get undue economic advantages by tampering with measurements. Basically, the attacker capability consists of changing the MI expected behavior, tampering any LR component and compromising the reliability of the measurements.

We assume the attacker could be any entity with access to the MI components or sensitive features, at any moment of its lifecycle. Attackers can be manufacturers, vendors, clients, and other entities. Malicious manufacturer staff (e.g., a malicious programmer) can inject software vulnerabilities and backdoors, ”selling” them to other potential attackers. Once an MI is deployed, vendors and clients can have access to its resources, exploring eventual failures and misbehaviors or changing sensitive parameters related to MI accuracy. Furthermore, modern MI usually provide interfaces for loading software updates and upgrades. Such features can be explored by malicious vendors and clients for loading tampered LR software or for modifying critical MI parameters.

Conversely, we establish that an attacker cannot compromise tamper-proof hardware devices, neither cryptographic primitives and communication protocols from algorithms recognized as secure. Also, we also define that an attacker cannot take part in collusion attacks with more than a fraction of peers that integrates the network. The exact value of this fraction depends on the blockchain implementation [16].

B. Basic Metrological Assurance Framework (BMAF)

We assume the existence of a Basic Metrological Assurance Framework (BMAF) tailored to implement MI regulation and control, which works as a countermeasure to the previously described attack model. Such BMAF gives a minimal set of requirements and activities, which is very realistic since its statements can be found in regulation directives implemented in several countries [2], [3], [4], [12].

Our BMAF sets the following protection requirements:

- **R1**: MI have reliable physical sealing to protect physical components such as sensors and electronic circuits;
- **R2**: MI implement acceptable mechanisms for LR software identification and integrity checking by notified bodies during MI supervision;
- **R3**: MI implement security mechanisms for LR software loading that accept only software modules signed by manufacturers and responsible notification bodies.

In turn, BMAF also establishes the following control and supervision activities:

- **A1**: MI hardware and software detailed analysis, LR software source code inspection and conformity assessment regarding the MI protection requirements;
- **A2**: MI validation and verification of all relevant MI use cases identified during type approval;
- **A3**: MI supervision by periodic inspection in both manufacturing site and application field. Activities must include MI seal verification and LR software identification and integrity check.

IV. Blockchains in Measuring Systems

In this section, we compare three different measuring system models: the traditional MI, a cloud-based measuring system and our blockchain-based measuring system model (Figure 1). For each model we describe the relevant supervision activities, using different sized icons to represent the expected magnitude of effort and cost associated with it.

A. Traditional MI

Traditional MI can be seen as dedicated computers calculating measurements of a physical quantity (e.g., size, weight, speed). They include sensors for interfacing with the physical world and AD converters for gathering data, besides other LR and NLR components, which are usually software modules. Sensors and AD converters are also LR components, being usually immutable hardware components (Figure 1-A).

Although LR and NLR software separation is a well-known concept, many MI manufacturers do not adopt such a practice. The claimed reasons are costs, computational resource restrictions or the existence of legacy software. However, despite their complexity, traditional MI software modules are usually monolithic systems [17]. This fact affects metrological assurance activities substantially, making MI regulation and control more expensive and complex, due to the following aspects:

- Type approval can demand MI hardware and software evaluation and check against a set of integrity requirements. Since LR and NLR are usually tightly coupled, notified bodies leading type approval need to evaluate and attest the compliance of all software modules. In some cases, LR software source code must be inspected for assuring their correctness.
- Software validation and verification can become more difficult due to the diversity of MI use cases, many of them being hard to reproduce out of the real measurement environment.
- Metrological supervision requires notified bodies to have sufficient staff to proceed with MI surveillance activities in both manufacturing and the field. Although physical seals can be helpful to protect physical components, they are ineffective for protecting software components.

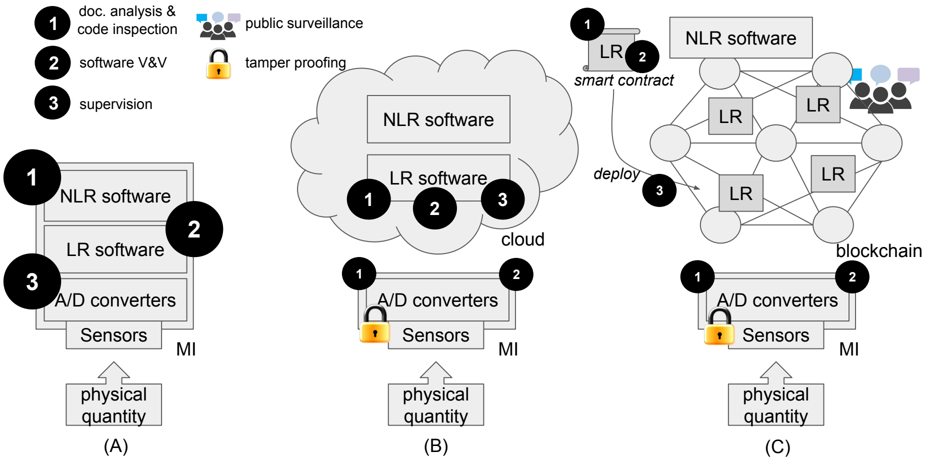

Fig. 1. Comparing measuring models: (A) Traditional MI; (B) Cloud Measuring System; (C) Blockchain Measurement System.

Due to their complexity, the activities also demand a highly qualified professional profile, complementary checking, and greater supervision staff proficiency. These factors contribute to make MI regulation and control a very expensive and time-consuming process.

B. Cloud-based Measuring System

When LR and NLR components are properly separated into independent modules, one could run these modules in different devices connected by well-defined interfaces. Such architecture leads to a DMS. For evaluating its properties, we take a cloud computing MI model inspired by Oppermann et al. [6]. In this model, LR and NLR software are running as cloud services, outside of MI physical set (Figure 1-B). We assume MI communicates with the cloud services using a secure channel (e.g., TLS) and that side channel attacks are infeasible.

When traditional and cloud models are compared, one can observe that the distributed architecture simplifies MI devices. MI practically do not include software components anymore once both LR and NLR software modules are running in the cloud. In practice, MI is now set up as a blend of sensors and AD converters. A communication interface allows the MI to send sensing raw data to the cloud measuring system. Basically, the MI could be designed only based on hardware components (e.g., smart sensors, cryptographic chips), although one should consider that some simple software could be necessary. In any case, a significant amount of software is moved from the MI to the cloud, being provided as a service. Consequently, LR and NLR software can be scaled accordingly to the demand.

A DMS easily enables software separation. This happens because LR and NLR software do not run over a monolithic platform anymore. That encourages manufacturers to decouple LR and NLR software, which consequently makes LR software less complex. Such aspect also impacts LR software type approval efforts, saving time and costs associated with documentation analysis, code inspection, and testing.

C. Blockchain-based Measuring System

Now we introduce the *blockchain model* (Figure 1-C). We consider that MI generate and store reliable measurements of physical quantities while managing the interests of different involved parties (e.g., consumption relations). Thus, measurements can be seen as *transactions* whose values must be protected against tampering and accidental changes [2]. Such aspects make DM a typical use case for blockchains applications.

As a first and intuitive insight, we devise a distributed ledger storing reliable measurement transactions which can be checked by any involved party. Also, the blockchain would support the execution of LR (and even NLR) software using smart contracts, which process information from sensors and generate a consolidated measurement value. The integrity of measurements and LR software (as smart contracts) is preserved by the blockchain inherent properties [18]. The ledger accounting enables the management of cumulative consumption transactions, such as energy and gas metering. If a financial blockchain platform is used, it can integrate billing and payment functions. That is an interesting additional resource when compared to the cloud model.

We can glimpse a practical implementation of such DMS in the following example related to energy measuring. Smart meters can be designed in a straightforward way: a tamper-proof hardware with only (1) voltage and current sensors; and (2) a module able to sign sensors' raw data and send them as a blockchain transaction. In the blockchain network, the peers invoke a smart contract that implements all remaining LR computation (e.g., signal processing, noise reduction, values integration). The blockchain ledger stores the final measurement. In turn, one obtains the cumulative energy consumption by querying each meter's stored measurements.

TABLE I
TRADITIONAL MI, CLOUD MODEL AND BLOCKCHAIN MODEL SECURITY ANALISYS SUMMARY.

	Trad. MI	Cloud Model	Blockchain Model
R1	Required	Required (tamper proofing).	Required (tamper proofing).
R2	Required	Required only for LR software in the cloud.	Unnecessary, LR smart contracts have integrity enforced due to blockchain properties.
R3	Required	Required only for LR software in the cloud.	Unnecessary, LR smart contracts are signed by notified bodies and checked by blockchain peers on deployment.
A1	Necessary	Necessary, but the evaluation of LR software in the cloud is expected to be easier than MI embedded software.	Necessary, but the evaluation of LR smart contracts is easier than the other models.
A2	Necessary	Necessary, but LR software use cases are reduced and its V&V can be performed without the need of field tests.	Necessary, but LR software use cases are reduced and its V&V can be performed without the need of field tests.
A3	Necessary	Partially necessary, since periodical inspections take place only in data centers where cloud servers are hosted.	Unnecessary, LR smart contracts have integrity enforced due to blockchain properties.

There is a crucial difference between blockchain and cloud models: the liability of the distributed services. In most use cases, MI belong to one of the parties interested in the measurement computing result. Energy and fuel are typical examples where vendors of goods own the MI. In the cloud model, one can expect that an interested party will hold the cloud measuring services. On the other hand, the blockchain is a truly decentralized architecture, being held potentially by several parties. Thus, one can expect that a blockchain model will require the contribution of different parties interested in the measurement activities, and consequently it will need to be designed following a different philosophy.

In the blockchain model, we devise measuring as *a service offered by someone without any interest in the measured quantity*. This idea is remarkably distinct to the traditional scenario where a vendor provides MI for measuring and is rewarded proportionally to the measurement. This idea fits very well in the blockchain model. Smart contracts can be used for computing measurements based on sensing information. However, they are coded by different parties that do not have conflicts of interest related to the measured quantity. Whatever the measurement result is, these parties shall be rewarded by a pre-set value. That motivates new players to provide better measuring algorithms. Additionally, this strategy creates incentives for keeping the blockchain network since that becomes profitable. This concept also breaks the traditional way MI are used in consumer relations, creating a new market for players who want to offer computing services for measuring.

The blockchain model also enables a set of complementary activities involving MI market and field surveillance that can be done by checking measurements inserted in the distributed ledger. Besides notified bodies, any entity representing society interests, consumers, goods providers, among others, can take part in additional supervision activities. We call that *public surveillance*. Such efforts can include smart contracts for generating redundant measurements for counter-proofing, or statistical analyses against the ledger looking for fraud evidence or patterns, for instance.

A last important aspect is the intrinsic blockchain robustness against attacks and failures. Since blockchains make extensive use of cryptography in both transactions and storing, information reaches a high level of protection regarding authenticity and integrity assurance. The known security attacks that can compromise a blockchain network are related to collusion among the stakeholders that participate in the consensus decision [19]. However, as more organizations take part in the consensus, more expensive and unfeasible these attacks become. Thus blockchains can provide a secure mechanism for assuring the legal liability and trustworthiness of instruments and measurements.

V. SECURITY ANALYSIS

In this section, we present a security analysis by comparing the measuring models discussed in the previous section. We consider the attacks and the metrological assurance framework BMAF described previously. We demonstrate how the traditional MI and DMS impact BMAF requirements and activities. Table I depicts such analysis.

Initially, we evaluate traditional MI security. In this scenario, one should note that BMAF requirements and activities are necessary to prevent attacks. As already discussed at Section IV-A, the activities of control and supervision of traditional MI involve a substantial effort. Documentation analysis, code inspection, and software validation and verification need to be done on all components and software modules. In turn, supervision also requires experienced surveillance technicians to implement inspection and software integrity checks.

When the cloud model is analyzed, one can notice that MI become simpler because LR and NLR software are now running in the cloud. Additionally, such situation reduces the capabilities of a typical attacker (e.g., consumers do not have physical access to MI software interfaces anymore). The BMAF protection requirements are still necessary, however requirements R2 and R3 are applied on the LR software implemented in the cloud. Supervision activities are also impacted, requiring fewer efforts to be executed. In A1, the document analysis and the code inspection of LR software running in the cloud are expected to require less effort than embedded software evaluation. Activity A2 is also made simpler once LR software tests can now be performed using interface stubs, without the need of real MI physical environment. Similarly, A3 also becomes less expensive because LR software identification and integrity check are executed against cloud servers, which are far fewer than the deployed MI. Finally, field surveillance for checking MI physical seals can also be eliminated. If we assume simplified MI as immutable instruments, they can be conceived as tamper-proof devices. That approach could eliminate the need for verifying MI seals as it implies that the MI will be permanently damaged and any attack trying to explore such vulnerability will not succeed.

Lastly, we evaluate the blockchain model. In addition to presenting the same characteristics of the cloud model, the blockchain security properties also affect BMAF requirements and activities. LR software is now a smart contract whose the deployment rules can be enforced for requiring developers and notified bodies attestation, something that automatically satisfies R3 and makes its regulation unnecessary. Once deployed, LR software is distributed among the peers, and it cannot be changed anymore. Blockchain peers cannot execute a different smart contract code. Otherwise, blockchain security assumptions will be violated. In consequence, R2 also becomes unnecessary. Regarding the activities, although A1 and A2 are still necessary, they should become much simpler when compared to the other models. This happens because the structure of smart contracts significantly limits the complexity resulting from having different technologies, software components and programming languages while imposing software separation. Finally, A3 becomes unnecessary in a blockchain network for the same reasons as R2.

We conclude that while DM already reduces attackers capabilities, such reduction is more accentuated in the blockchain model. Once LR software is produced by players who are exempted from conflicts of interest, many activities related to the assurance of software correctness and integrity are made simpler or even unnecessary. The blockchain security properties play an important role in this context.

VI. Case Study

A. *Speed meters and the case study scenario*

In this section, we develop an experiment to demonstrate the feasibility of our proposal. It consists of a vehicle speed DMS using blockchains (Figure 2). These meters are efficient solutions for estimating vehicle speed on public roads, generating traffic statistics and enforcing speed limits for drivers [20], [21]. The meter detects each vehicle, determines its speed and captures one or more pictures identifying the vehicle's license plate when necessary. The measurement and the license plate image constitute the legally relevant record, which is expected to be reliable and protected against frauds.

We choose the city of Sao Paulo, in Brazil, as a case study. Sao Paulo has a vehicular fleet with more than 8 million vehicles and about one thousand speed meters deployed along its roads [22]. Furthermore, over the last two years, we have proceeded with formal type approval of speed meters in the Brazilian National Institute of Metrology, Quality and Technology (Inmetro[3]). This experience provides valuable information that supports the analysis in this section.

Developing countries are essentially guided for legal metrology policies related to fraud detection and avoidance [1], [3], [12]. In the majority of cases, these countries adopt restrictive legal metrology activities, which include detailed type approval processes and intensive metrological inspection, especially field surveillance. That is the Brazilian reality regarding vehicle speed meters. These instruments are under restrictive regulation and control directives, which associates them to a WELMEC 7.2 class-D risk level [14]. In this aspect, the blockchain-based DMS can introduce promising advantages, as it simplifies the metrological assurance framework activities, while preserving many of the advantages from a DMS in terms of performance and costs saving.

B. *Conceiving a vehicle speed meter DMS*

In Brazil, vehicle speed meters are usually built as *Type-U instruments*. WELMEC 7.2 [14] defines this classification as instruments that run their software in universal computer hardware. This happens because speed meters nowadays have an extensive list of requirements and aggregate a large number of NLR features. Consequently, their software usually is quite complex and hard to evaluate and test. Besides, speed meters manufacturers complain about opening their solutions for inspection due to intellectual property issues. That is the case of the speed meters evaluated in Inmetro. So they are representative cases of the traditional MI model described in this work.

Another aspect is that speed meter owners usually deploy their equipment at far places along roads spread over large geographic areas. That increases the difficulty of regular inspection activities and consequently increases costs associated with metrological surveillance. Thus, all those aspects make speed meters strong candidates for solutions that help to separate LR and NLR software.

We implement such a solution creating a blockchain network for distributed measuring. We integrate the speed meter's LR features in a simple tamper-proof hardware that uses two inductive sensors to capture the vehicle's magnetic profile [21]. After detecting a vehicle, this hardware uses a private key to sign the sensor's raw data and sends that to a blockchain-based DMS. The blockchain executes the LR software as a smart contract and computes the vehicle speed. The vehicle detection event also triggers any other device used for providing complementary evidence (e.g., a camera that captures the vehicle license plate). Furthermore, manufacturers may aggregate any other module necessary for implementing NLR functionalities or even use the blockchain to do that. Their decision does not affect the legal metrology activities once NLR features are not under regulation.

C. *Architecture using Hyperledger Fabric*

Our prototype uses Hyperledger Fabric [11] as a permissioned blockchain, where the peers cooperate to store measurements and execute LR software. Fabric is an open source blockchain platform that includes two concepts that are very helpful for implementing our idea: *endorsers* and *security policies*.

Endorsers are peers that effectively execute smart contracts, which are called *chaincodes* in Fabric. The way endorsers operate has significant implications concerning intellectual property and performance. First, a manufacturer needs to reveal her LR software only to peers contracted to execute her measuring chaincode as a service, and to the notified body

[3] http://www.inmetro.gov.br

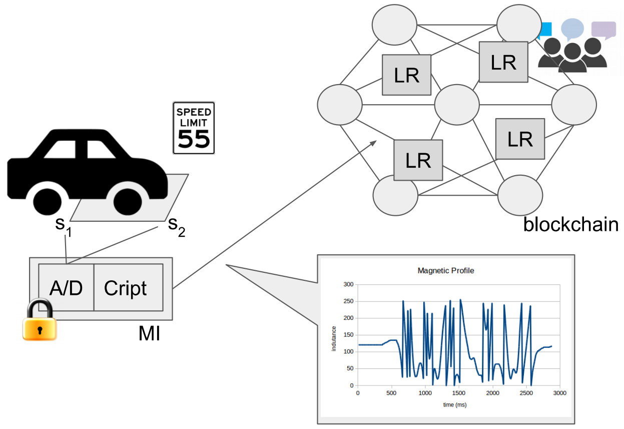

Fig. 2. The blockchain-based vehicle speed DMS.

responsible for approving it. Second, a manufacturer can size his solution performance by providing as many endorser peers as necessary, resulting in a scalable architecture.

Security policies can set the way the network validates the measurement provided by endorsers. These policies can also work as a protection mechanism against collusion attacks and implement metrological surveillance. Blockchain networks are constituted by peers that belong to different stakeholders or organizations. Although they do not need to trust each other, each peer continually verifies other peers behavior. Security policies define rules for such monitoring. They can specify, for instance, which organizations must take part in a transaction endorsement (i.e., a measurement calculation). The more organizations are involved, the more expensive it becomes to carry out a collusion attack. Such monitoring activities are a sort of metrological surveillance since different organizations are continually checking the measurements resulting from a chaincode execution.

Since we resort to a permissioned blockchain, the consensus protocol plays an important role in our experiment. Fabric refers to consensus as an *orderer service*. We perform our experiment with two different types of orderer services: the *solo orderer* and the *Byzantine Fault-Tolerant (BFT) orderer*. The solo orderer service is native from Fabric distribution, and it is very practical to implement tests and develop proof of concept prototypes. However, regarding security, the solo orderer cannot be considered a suitable solution because it implies that consensus comes from only one organization. In turn, the BFT orderer [15] is fully replicated for tolerating Byzantine failures. Thus, one can configure the BFT orderer with several replicas, with different organizations controlling each one of them. Such approach employs a decentralized consensus provided by the BFT-SMaRt replication library [23], thus providing security against collusion attacks.

D. *Describing MI regulation and control activities*

We set up our vehicle speed DMS as illustrated in Figure 3. Inmetro and LaSIGE represent two independent organizations with distinct functions. Inmetro is a notified body responsible for regulating and controlling such instruments. LaSIGE provides computational resources for executing LR software from speed meters.

The speed meter manufacturer implements LR software as a chaincode. We create a chaincode written in Go that analyses raw data from both sensors and finds the moment when the vehicle activates each sensor. Once the samples are taken in regular periods, and we know the distance between the sensors, it becomes trivial to determine the vehicle speed. When the meter also enforces speed limits, one or more images from the vehicle's license plate can be necessary. Such requirement brings some concerns about privacy. Although the image is part of the legally relevant information, it is not necessary for determining the vehicle speed. So one can use different approaches to avoid problems with privacy. One idea consists of encrypting the images using asymmetric cryptography before sending them to the blockchain. One can do that using the public key of the legal authority responsible for issuing traffic tickets. A better alternative is to send only the image digital signature to the blockchain. The image is kept in a private data storage, and the blockchain can attest its integrity and authenticity whenever necessary. Such approach improves performance and eliminates privacy concerns. We consider this approach in our solution.

Our experiment assumes the implementation of legal metrology activities as follows. Firstly, the notified body proceeds with the MI type approval. He does that by evaluating the MI device (i.e., inspecting sensors, cryptographic and communication features) and the LR source code (i.e., the Fabric chaincode). After, the notified body executes the applicable tests for assuring all MI LR functionalities. Once MI hardware and software are approved, the notified body is responsible for instantiating the chaincode in the blockchain.

In Fabric, a chaincode instantiation includes the notification of endorsers that will execute that chaincode. To do that, the notified body inserts into the blockchain a new transaction comprised by the chaincode fingerprint (i.e., the software image hash) and its respective security policy. Thus all the peers in the network know how to validate the chaincode execution by checking the applicable security policies. After instantiation, the chaincode fingerprint becomes immutable, following the intrinsic blockchain properties. Any future update in the chaincode will require a new instance of it, which means to create a new chaincode version.

After Inmetro instantiates the LR chaincode, any MI owner can contract computing services from the LaSIGE organization for executing the respective LR software. In practice, LaSIGE provides endorser peers. One must note that LaSIGE is only one possible organization that can offer such a service. Although we have only two organizations in our experiment, a real scenario can include several independent organizations. Each one of them could have endorser peers offering measuring services. Once the MI owner finds an available endorser peer, she needs to install the approved LR chaincode. That enables any MI in the field to generate transactions, ask endorsers to determine the vehicle speed, and store the vehicle speed legally relevant information in the blockchain ledger.

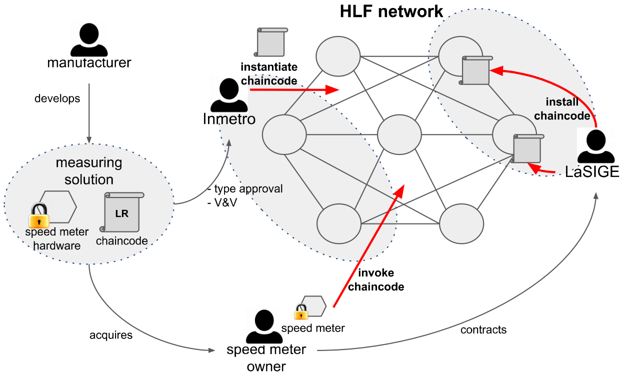

Fig. 3. Vehicle speed DMS solution scheme.

E. Security analysis

We verify how our implementation offers countermeasures against common attacks associated with the capabilities described at the Section III-A. These countermeasures match the properties already discussed in Section V. In the following we describe some attacks and how our blockchain-based DMS deals with them.

1) An attacker tries to compromise the speed meter hardware integrity: We conceive the speed meter hardware as a tamper-proof device. Consequently, any attempt at violating seals or stealing a private key shall destroy the speed meter hardware, forcing its replacement and exposing the attacker.

2) An attacker tries to install a malicious chaincode in an endorser peer: The blockchain ledger contains the fingerprint of every instantiated chaincode, and legitimate endorsers automatically check the LR chaincode integrity on deploy time. If a malicious speed meter owner tries to install a modified LR chaincode version, the endorser refuses such software. Furthermore, the endorser appends a transaction registry in the blockchain. That can be useful in audits for detecting attacks against software integrity in the future.

3) An attacker can collude with an organization for loading malicious chaincode in endorser peers: Security policies assure protection against collusion attacks. For instance, suppose that an attacker colludes with LaSIGE, making compromised peers accept a modified LR chaincode. One can avoid such attack by enforcing security policies defining that at least N peers of different organizations must endorse the chaincode execution. Thus, a collusion attack including only the LaSIGE organization will not succeed. The attacker needs to compromise more organizations, which makes the attack too expensive and, consequently, unfeasible.

4) An attacker has success in colluding with enough peers for injecting fraudulent measurements in the blockchains: One should remember that, in the proposed DMS, organizations compute measurements as an independent service (i.e., they do not have any advantage or reward regarding the measurement result). Such business model discourages collusion and makes these attacks disadvantageous. Even so, assuming that an attacker succeeds in compromising a sufficiently large number of peers for endorsing a malicious transaction, one can expose such fraud by auditing the measurement record. Since Inmetro keeps any approved LR chaincode and the blockchain records all information used in any transaction, Inmetro can re-execute the LR chaincode and compare the obtained measurements. The sensor raw data can easily have its integrity verified by using the MI public key. Regarding the MI private key integrity, we already discussed this issue in the first attack described in this section.

VII. Performance Issues

An essential step in our experiment is the speed meter DMS performance evaluation. We try to estimate the blockchain peers behavior without considering network communication issues. Essentially, we are interested in two main aspects:

- The throughput and latency within each endorser peer. This subject is important because it helps to estimate how many peers a speed meter owner will need, considering a specific demand.
- The throughput of a simple blockchain network configuration. We test a high number of transactions against a network consisting of only two peers (one of them as an endorser) and two different types of consensus service.

A. Demand goals

We estimate the experiment demand based on vehicles traffic real data. According to a technical report from the Sao Paulo's Traffic Engineering Company (CET-SP) [22], there was in 2016 approximately one thousand speed meters spread along the city. The same report analyses the vehicles flow on the main roads in the city and points out an average of 2,772 vehicles/hour (or 0.76 vehicles/sec) during rush hour. Assuming such demand for a total of 1 thousand speed meters, we need a blockchain network able to process something around 800 tps (transactions per second). Androulaki et al. [11] benchmarks Fabric performance in something between 2,000 to 3,000 tps. However, they consider specific scenarios with proper customizations for evaluating particular performance issues. Our experiment does not employ any specific customization. We use Fabric just as provided by its developers, in an ordinary hardware infrastructure available in any datacenter. The objective is to evaluate the results that a speed meter solution owner can obtain by using Fabric.

B. Test environment setup

Our blockchain network environment consists of 3 nodes from a Dell PowerEdge R410 cluster. Each node has two CPUs Intel Xeon Processor E5520 with 2.27 GHz and 32 GB of RAM. Fabric standard distribution version 1.1[4] runs over *docker containers*, so that each physical node can host several peers. For the sake of simplicity, we use three nodes for allocating the orderer service, the LaSIGE peers, and the Inmetro peers, respectively.

Regarding the orderer service, we test two different scenarios. The first one is the native Fabric's solo orderer service, which is provided primarily for testing. The second scenario uses the orderer service developed by Sousa et al. [15], which tolerates Byzantine failures.

[4] http://hyperledger-fabric.readthedocs.io/en/release-1.1

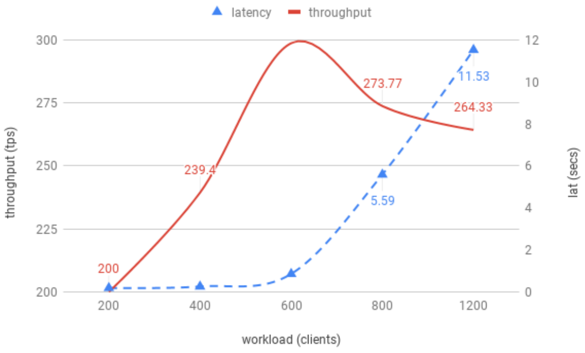

Fig. 4. Throughput and latency using native Fabric solo orderer.

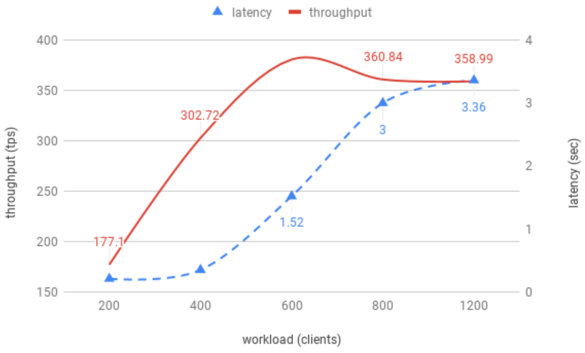

Fig. 5. Throughput and latency using BFT orderer with four replicas.

The client transactions load is mimicked using four nodes from a Dell PowerEdge R300 cluster, each one with an Intel Xeon Processor L5410 with 2.33 GHz and 8 GB of RAM. The transactions simulation uses real data from speed meters developed by the company Perkons[5] SA, which kindly granted a dataset for this experiment.

C. Test methodology

The performance tests execute as follows. Each physical machine creates the respective Fabric docker containers (peers or clients). Client containers are responsible for generating transactions. We use Fabric as an *off-the-shelf solution*, without any customization. Each client instance corresponds to a container process that sends transactions to the blockchain. We try to produce a maximum workload by increasing the number of clients.

When a client instance is created, it selects a vector of bytes containing the sensors raw data from the dataset mentioned above, for each transaction. Clients use the Fabric protocol to invoke an LR chaincode and send such data as an argument to an endorser peer from LaSIGE organization. The endorser peer receives the vector of bytes, calculates the vehicle speed, and returns an endorsed package with the respective measurement. The client uses such package for composing the complete transaction and sends it to the orderer service responsible for generating the ledger blocks and disseminate them to the other peers. The client also keeps records of timestamps and the elapsed time for completing the transaction. Such information gives the throughput and latency of the system.

D. Performance test results

Figures 4 and 5 depict our tests results for the system using the solo orderer and the BFT orderer with 4 replicas, respectively. With the solo orderer service, throughput and latency reach a better trade-off around 300 tps and 1 second, respectively. The workload necessary to get such results corresponds to 600 simultaneous clients. With a higher workload, performance degrades substantially. We reach a throughput of around 260 tps and a high latency of 12 seconds when testing a workload from 1,200 clients. This high latency can be explained by the transactions queuing in the solo orderer.

The BFT orderer service with four replicas performs better. It reaches the best trade-off with a throughput of 380 tps and a latency in 1.6 seconds with the same workload of 600 simultaneous clients. The BFT orderer also keeps throughput stable at about 360 tps even with a workload of 1,200 clients, presenting only a slight increase of 1 second in latency. The BFT orderer optimizes latency by discarding the exceeding of transactions after reaching its max throughput. However, clients need to control refused transactions, creating their queue and resending the transaction again after some time.

The BFT orderer service also includes an important aspect. It enables a truly distributed consensus service, once each replica belongs to a different organization. Although the number of replicas impacts performance [23], it aggregates security by preventing collusion attacks.

One can observe that our results point out a difficulty in dealing with the estimated peak demand of 800 tps. However, we understand that the blockchain can absorb such demand along the day since the number of transactions goes down after the rush hour. We conclude that Fabric performs satisfactorily for implementing a speed meter DMS. Furthermore, if necessary, one can even reach a better performance by customizing Fabric features, something indeed feasible once the platform is an open source software product.

VIII. Quantitative Assessment

In this section, we provide some quantitative assessment regarding the adoption of the blockchain-based DMS model, when compared with traditional MI. We do that by evaluating advantages and drawbacks related to both technologies (Table II). The discussions present in the section are not exhaustive. Actually, they are preliminary results of a risk analysis study in progress at the moment. However, we believe that the discussed aspects are useful for providing assessment information to people interested in implementing a blockchain-based DMS.

A. Software vulnerabilities

There is a direct relation between the size of a software product and the number of software defects. Alhazmi et al.

[5] http://www.perkons.com

TABLE II
ADVANTAGES AND DRAWBACKS OF THE TRADITIONAL MI AND OUR FABRIC DMS IMPLEMENTATION.

Evaluated aspect	**Trad. MI**	**Fabric DMS**
Expected LR software defects (D/kLOC)	18	2
Code inspection effort (men/month)	0.5	0.033
Costs with hardware[6](U$)	500,000	5,000
Connectivity dependency	None	Very high
Estimative of availability	Fair	Very high

[24] estimate that the ratio of remaining vulnerabilities to the total number of software defects is often in the range of 15%. Considering that, we estimate how much the amount of LR software can impact the security of traditional MI and blockchain-based DMS. In our analysis, we adopt the average ratio of approximately 6 D/kLOC (defects by thousands of lines of code) reported by Carrozza et al. [25]. In our experience with speed meters type approval at Inmetro, we found that LR software average size is around of 3,000 LOCs. This software size statistically suggests the existence of about 18 software defects and, consequently, a high probability of having a vulnerability. Such LR software size is a consequence of the strong coupling between LR and NLR modules. When one considers only the software effectively used in measuring, the LR software size can be remarkably reduced. We confirm that in our speed meter DMS implementation. Its LR software consists of a Go language chaincode that requires no more than 200 LOCs. Such size points out an estimate of no more than two remaining software defects, which also reduces the chances of potential vulnerabilities.

B. Code inspection efforts

The LR software size also affects the efforts required by MI type approval, especially code inspection activities. We evaluate such impact by using the studies of Ebert and Jones [26] about the quality of embedded software. Their work reports an average production rate of 60 FP (Function Points) by men/month in code inspection. Since the majority of the manufacturers implement their software in C language, we adopt the FP/kLOC conversion rate in the QSM Function Point Languages Table [7] of approximately 100 LOCs per FP. Considering the LR software sizes estimated previously, we have an expected code inspection effort of 0.5 men/month in the traditional MI model against 0.033 men/month in the blockchain-based DMS.

C. Costs with hardware

Other important aspect concerns the costs associated with the Type-U hardware adopted by speed meters manufacturers. In Brazil, due to the high temperatures associated with the tropical climate, manufacturers need to build their meters using specific motherboards and components. Such hardware easily exceeds U$ 500, something that makes the Brazilian traditional speed meters a quite expensive product. In our experiment, we consider the deploy of 1,000 meters, which correspond to the number of such devices in Sao Paulo. With the traditional MI model, that implies a direct cost of approximately U$ 500,000 only with the Type-U hardware. In our implementation using Fabric, we succeeded in executing the LR software of the same number of MI in only three nodes of a Dell PowerEdge R410 cluster, which represents a cost with hardware that does not exceed U$ 5,000. We cannot directly compare both scenarios because manufacturers avail the Type-U hardware deployed in the field to provide NLR features. However, we understand that the NLR software can also be provided by independent remote services. Furthermore, the required hardware becomes less expensive once remote services run from data centers where the environment do not present the same inclement weather found in the field. Thus cost saving is evident in the adoption of a distributed and decentralized solution.

D. Connectivity dependency and demand

Connectivity is a critical requirement in the blockchain-based DMS. The simple MI hardware needs to send sensing information to the blockchain on every vehicle detection. If the device loses connectivity, the information must be discarded or stored in temporary memory. Such scenarios can require a more sophisticated MI hardware (e.g., additional memory for temporary data and a state machine to deal with connectivity restrictions) or even compromise MI availability. On the other hand, the traditional MI includes enough computational resources to manage information when there is no connectivity. They can even operate offline for several days without any problem related to information loss.

Concerning the demand by connectivity, we can state that both solutions are similar. Although the traditional MI can send information in batch mode, the expected amount of information is practically the same as in the blockchain-based DMS. Furthermore, despite the extensive use of cryptography in a blockchain application, that does not represent a significant overhead regarding the amount of propagated information. Roughly speaking, the use of connectivity resources depends on the number of detected vehicles and not from the adopted model.

E. Service reliability and availability

We evaluate the reliability and availability of the services provided by both speed meter models by comparing properties of a centralized and a distributed system. In Reliability Theory, the MTBF (Mean Time Between Failures) and the MTTR (Mean Time to Repair) are traditional measures of reliability and availability of a system [27].

Traditional MI are centralized solutions and can easily become a single point of failure. Although the same happens with the simple MI hardware in the blockchain-based DMS, one knows that complex systems fail more often than simpler ones. So we can state that the $MTBF_h$ of the simple MI hardware used int the blockchain-based DMS is expected to be higher than the $MTBF_H$ of traditional MI. Regarding the peers providing LR software execution, we claim that blockchains are based on distributed trust instead of a single point of trust. That means the blockchain fails only if multiple

[6] The analysis do not include the hardware required for NLR software.

[7] http://www.qsm.com/resources/function-point-languages-table

stakeholders collude against the system. Our implementation uses redundant peers (or replicas) with Byzantine consensus, which tolerates F faults for $N = 3F + 1$ nodes [15]. Every replica has its own $MTBF_R$. The blockchain-based DMS total MTBF needs to consider that the replicas can present parallel faults, while the simple MI hardware and the set of replicas affect each other in serial faults. In turn, availability estimative depends primarily on the MTTR [27]. Consequently, if the organizations integrating the blockchain consensus can restore any faulty replica in an interval time $T \leq MTBF_R * F$, they can assure the service availability.

At the moment we write this paper, we do not have enough quantitative information for estimating the MTBF and MTTR of each solution model. However, we can state two crucial aspects:

- The blockchain-based DMS is expected to present a higher MTBF due to its simpler hardware in the field and because it does not have a single point of failure at the blockchain.
- The blockchain-based DMS is expected to present a better availability ratio due to its inherent fault tolerance.

Considering this discussion, we estimate the traditional MI model as presenting a fair availability level, while the blockchain-based DMS is expected to offer high availability.

IX. Challenges Ahead

Although blockchains-based DMS is a promising approach, several challenges need to be addressed for their use. Some of them become very clear after we proceed with our practical experiment. We highlight the following main issues that should be addressed in future works:

• **The measurement Big Data:** MI usually manipulate a high amount of data. In a large-scale scenario (e.g., energy meters in a smart grid), MI can update their measurements faster, generating lots of transactions. A network connecting millions of meters may generate a transaction load unfeasible to be processed by existing blockchains. In our tests with Fabric, for instance, we reach a max throughput of 380 tps, although Androulaki et al. [11] points out a performance more than of 2,000 tps in their benchmark. However, even such performance may not be enough to meet the demand for measurements on a smart grid, for instance. In such scenarios, solutions can require different workarounds and creative alternatives. We recall the use of endorsers, an idea that we explored with success in our experiment. Besides, one can try to use aggregated measurements for reducing transactions in a blockchain. Also, one can try smarter MI for determining transactions on demand.

• **Measuring and privacy:** Measurements assigned to a specific person allow to infer information about her habits and lifestyle. In a blockchain with a public ledger, this problem becomes more serious. One needs to establish an acceptable trade-off between privacy and efficiency. In our experiment, we faced such a problem with the vehicle license plate image and solved it by storing such information outside of the blockchain. However, depending on the application scenario, privacy can require more sophisticated mechanisms for protecting or obfuscating identities, such as pseudonyms or identity protection layers. Permissioned blockchains constitute a suitable alternative once they contemplate an access control layer built into blockchain nodes [16]. One can also constrain access policies in such a manner that they satisfy privacy rules and restrictions.

• **Communication issues:** Although we consider MI as connected devices, communication can be a problem in applications demanding real-time decisions. That is a restriction for any DMS over asynchronous networks. Thus blockchain-based measuring is not appropriate for all MI applications. Furthermore, attacks targeting communication (e.g., DDoS) represent an additional risk, although decentralized systems such as blockchains are more resilient to such attacks than conventional cloud architectures.

• **Oracles authentication:** External information providers are usually called *oracles* in blockchain architectures. In the described model, MI sensors can be seen as oracles since they are responsible for providing information from the physical world. Even though sensors are small components which can be protected using physical seals, sensors authentication can be necessary to assure measuring reliability.

X. Conclusion

In this paper we discussed how blockchains can be used to support DMS. Due to their intrinsic security properties, blockchains can improve MI metrological assurance by imposing restrictions against potential attacks while reducing technical efforts related to regulation and control activities. We demonstrated those properties by implementing a vehicle speed meter DMS using Hyperledger Fabric. Our results were consistent, and they support the feasibility of our proposal. However, despite its promising application, blockchains pose several challenges that need to be faced. The main ones are related to the amount of data, privacy, communication and oracles authentication. Future work shall include a complete risk analysis of our blockchain-based model to develop new strategies for addressing the challenges discussed here.

Acknowledgment

The authors thank Perkons SA for kindly providing the speed meters dataset used in the experiments of this work.

This work was partially sponsored by the by the Coordination for the Improvement of Higher Education Personnel (CAPES), grant 99999.008512/2014-0; by the EU-BR SecureCloud project (MCTI/RNP 3rd Coordinated Call); and by FCT through the LASIGE Research Unit (UID/CEC/00408/2013) and the IRCoC project (PTDC/EEI-SCR/6970/2014).

References

[1] B. A. Rodrigues Filho and R. F. Gonçalves, “Legal metrology, the economy and society: A systematic literature review,” *Measurement*, vol. 69, pp. 155–163, 2015.

[2] M. Esche and F. Thiel, “Software Risk Assessment for Measuring Instruments in Legal Metrology,” in *Proceedings of the Federated Conference on Computer Science and Information Systems*, vol. 5, 2015, pp. 1113–1123. [Online]. Available: https://fedcsis.org/proceedings/2015/drp/127.html

[3] S. Camara, R. C. S. Machado, and L. F. Carmo, “A Consumption Authenticator Based Mechanism for Time-of-Use Smart Meter Measurements Verification,” *Applied Mechanics and Materials*, vol. 241-244, no. February, pp. 218–222, 2012. [Online]. Available: http://www.scientific.net/AMM.241-244.218

[4] D. R. Boccardo, R. C. S. Machado, S. Camara, C. B. Prado, W. S. Melo Jr., L. C. Ribeiro, and L. F. Carmo, “Software validation of medical instruments,” *2014 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, no. October 2015, pp. 1–4, 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6860090/

[5] D. Peters, F. Thiel, M. Peter, and J.-P. Seifert, “A secure software framework for Measuring Instruments in legal metrology,” *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, pp. 1596–1601, 2015. [Online]. Available: http://ieeexplore.ieee.org/document/7151517/

[6] A. Oppermann, F. G. Toro, F. Thiel, and J.-P. Seifert, “Secure Cloud Computing: Reference Architecture for Measuring Instrument under Legal Control,” *Security and Privacy*, vol. e18, pp. 1–26, 2018.

[7] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[8] K. Christidis and M. Devetsikiotis, “Blockchains and Smart Contracts for the Internet of Things,” *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[9] D. Peters, J. Wetzlich, F. Thiel, and J.-p. Seifert, “Blockchain Applications for Legal Metrology,” in *IEEE International Instrumentation and Measurement Technology Conference*, Houston, Texas, USA, 2018, p. 6.

[10] W. S. Melo Jr., L. F. Carmo, A. Bessani, N. Neves, and A. Santin, “How Blockchains can improve Measuring Instruments Regulation and Control,” in *IEEE International Instrumentation and Measurement Technology Conference*, Houston, Texas, USA, 2018, p. 6. [Online]. Available: https://ieeexplore.ieee.org/document/8409724/

[11] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, and J. Yellick, “Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains,” in *Proceedings of the Thirteenth EuroSys Conference*, Porto, Portugal, 2018.

[12] H. Luchsinger, C. Cajica, M. Maldonado, and I. Castelazo, “Are Gas Pumps Measuring Up? The Mexican Experience,” *NCSLI Measure*, vol. 3, no. 2, pp. 62–68, 2008.

[13] International Organization of Legal Metrology (OIML), “OIML D 31, Editon 2008: General requirements for software controlled measuring instruments,” p. 53, 2008.

[14] European Cooperation in Legal Metrology (WELMEC), “WELMEC 7.2, 2015: Software Guide,” pp. 1–114, 2015.

[15] J. Sousa, A. Bessani, and M. Vukolić, “A Byzantine Fault-Tolerant Ordering Service for the Hyperledger Fabric Blockchain Platform,” in *48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2018.

[16] M. Vukolić, “The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9591, pp. 112–125, 2016.

[17] V. Abreu, A. Santin, A. Xavier, A. Lando, A. Witkovski, R. Ribeiro, M. Stihler, V. Zambenedetti, and I. Chueiri, “A Smart Meter and Smart House Integrated to an IdM and Key-based Scheme for Providing Integral Security for a Smart Grid ICT,” *Mobile Networks and Applications*, pp. 1–15, 2017.

[18] Z. Zheng, S. Xie, H.-N. Dai, and H. Wang, “Blockchain Challenges and Opportunities : A Survey,” *International Journal of Web and Grid Services*, pp. 1–24, 2017. [Online]. Available: http://inpluslab.sysu.edu.cn/files/blockchain/blockchain.pdf

[19] G. O. Karame and E. Androulaki, *Bitcoin and blockchain security*. Artech House, 2016.

[20] Y. K. Ki and D. K. Baik, “Model for accurate speed measurement using double-loop detectors,” *IEEE Transactions on Vehicular Technology*, vol. 55, no. 4, pp. 1094–1101, 2006.

[21] S. S. M. Ali, B. George, L. Vanajakshi, and J. Venkatraman, “A Multiple Loop Vehicle Detection System for Heterogeneous and Lane-less Traffic,” *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 5, pp. 1413–1417, 2011.

[22] Companhia de Engenharia de Tráfego - CET, “Pesquisa de monitoramento da mobilidade: mobilidade no sistema viário principal: volume e velocidade - 2015,” Tech. Rep., 2017.

[23] A. Bessani, J. Sousa, and E. E. P. Alchieri, “State machine replication for the masses with BFT-SMaRt,” in *44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2014, pp. 355–362.

[24] O. H. Alhazmi, Y. K. Malaiya, and I. Ray, “Measuring, analyzing and predicting security vulnerabilities in software systems,” *Computers and Security*, vol. 26, no. 3, pp. 219–228, 2007.

[25] G. Carrozza, R. Pietrantuono, and S. Russo, “Defect analysis in mission-critical software systems: a detailed investigation,” *Journal of Software: Evolution and Process*, vol. 27, pp. 22–49, 2015.

[26] C. Ebert and C. Jones, “Embedded software: Facts, figures, and future,” *Computer*, vol. 42, no. 4, pp. 42–52, 2009.

[27] I. Koren and C. M. Krishna, *Fault-tolerant systems*. Elsevier, 2010.

Wilson S. Melo Jr. is a Researcher at the Brazilian National Institute of Metrology, Quality, and Technology (Inmetro). He holds a Ph.D. in Computer Sciences from the Federal University of Rio de Janeiro (UFRJ). He has more than 20 years of experience with software development and testing projects. His main expertise regards software for industrial applications, especially solutions related to measurement, control, patterns recognizing, and cybersecurity. More information about him can be found at https://www.researchgate.net/profile/Wilson_Melo_Junior.

Alysson Bessani is an Associate Professor of the Faculty of Sciences of the University of Lisboa, Portugal, and a member of LASIGE research unit. He holds a Ph.D. in Electrical Engineering from UFSC (Brazil) and was a visiting professor in Carnegie Mellow University (2010) and a visiting researcher in Microsoft Research Cambridge (2014). He is the co-author of more than 100 peer-reviewed publications on dependability, security, Byzantine fault tolerance, and cloud. More information about him can be found at http://www.di.fc.ul.pt/~bessani.

Nuno Neves is Professor at the Department of Computer Science, Faculty of Sciences of the University of Lisboa. He leads the Navigators research group and he is on the scientific board of the LASIGE research unit. His main research interests are in security and dependability aspects of distributed systems. Currently, he is investigator in several national and EU projects, such as SEAL and uPVN. His work has been recognized in several occasions, for example with the IBM Scientific Prize and the William C. Carter award. He is on the editorial board of the International Journal of Critical Computer-Based Systems. More information about him can be found at http://www.di.fc.ul.pt/~nuno.

Altair Olivo Santin received the BS degree in Computer Engineering from the PUCPR in 1992, the MSc degree from UTFPR in 1996, and the PhD degree from UFSC in 2004. He is a full professor of Graduate Program in Computer Science (PPGIa) and head of Security & Privacy Lab (SecPLab) at PUCPR. He is a member of the IEEE, ACM, and the Brazilian Computer Society.

Luiz F. Rust C. Carmo received a Ph.D. degree on Computer Science in 1994, from the LAAS/CNRS, Toulouse III France. Presently, he is a Senior Specialist in Computer Sciences of the Brazilian Institute of Metrology, Technology and Quality (Inmetro), General Coordinator of the Education Center. He is an active lecturer of both the Doctoral programs in Computer Sciences of UFRJ and in Metrology of Inmetro. His research interests include information security, and embedded systems.