A Multi-Agent System Blockchain for a Smart City

André Diogo, Bruno Fernandes, António Silva, José Carlos Faria, José Neves, and Cesar Analide

Department of Informatics
Centro ALGORITMI, University of Minho
Braga, Portugal

Email: a75505@alunos.uminho.pt, bruno.fmf.8@gmail.com, a73827@alunos.uminho.pt, a67638@alunos.uminho.pt, jneves@di.uminho.pt, analide@di.uminho.pt

***Abstract*—In a Smart City context, and specifically targeting public collection of sensor data from arbitrary sources by arbitrary actors, accuracy, reliability and frequency of data may be highly variable. The Blockchain technology allows the management of public immutable ledgers that track the activities of these actors closely and, as such, provides a possible solution to incentivize and empower good actors (those who supply accurate, reliable and frequent data). This paper focuses on mechanisms to provide such incentives, what we call a Proof-of-Confidence (POC), using a view of these actors as intelligent agents, capable of autonomous interaction with the Blockchain. While it is concluded that full security guarantees can not be provided without additional restrictions at the agents' behavior level, our model is used to prove the feasibility of supplying a gamified environment for such agents, with optimizable metrics which favor accurate, reliable and frequent data.**

Keywords-blockchain; smart city; sensor data; multi-agent systems.

I. INTRODUCTION

Smart Cities introduce novel problems related to the management of inordinate amounts of sensor data of various types and origins. From local temperature data to road traffic, the data may be originated from a multitude of data sources, distributed through numerous autonomous and communicating devices, usually referred to as the Internet of Things (IoT). Such data must be stored adequately as it is fundamental for the analysis and development of real physical models.

The distributed management of data originated from the IoT requires a network capable of dealing with changes in the environment. An architecture comprised of regular microservices would be a valid solution to this problem, adapting to changes reactively. It lacks, however, a contextual view of these changes, meaning that the network will act on them based directly on differences in data values, and not on what these differences might mean for the analyzed environment. Intelligent agents add this contextual awareness, as well as autonomy and intelligence in the form of problem solving to achieve greater rewards as result of contributing to the network's maintenance. In fact, these agents are capable of learning, adjusting and optimizing their behaviors in the presence of incentives. The ecosystem of agents, known as a Multi-Agent System (MAS), allows further flexibility in communication and use of established Multi-Agent development platforms [1].

In regard to the network itself, Blockchains enjoy desirable characteristics to organize the collected data due to their immutable and referential nature. They effectively track all identities that inscribe data into the Blockchain using units called as transactions, which couple each identity to the data it published. A public edited ledger keeps its integrity over time, as each block in the chain references its previous, allowing blocks to become increasingly tamper-resistant. However, Blockchains by themselves do not provide incentives to produce accurate, reliable and frequent data that can be used, for example, to produce/optimize machine learning models. This situation deteriorates even further with the huge amount of distributed devices with unknown origins and data sources.

This paper aims to describe a new Blockchain model, whose goal is to enable a gamified environment for a system comprised of a multitude of agents. A system where agents that work towards its intended goal provide good data and allow the potential to identify malicious ones. Hence, this paper is structured as follows, viz. Section II will go over the unique characteristics of the conceived Blockchain, such as how interactions guarantee no data is lost and how data is stored in the Blockchain. Section III will describe how this MAS may interact with the Blockchain, followed by Section IV which describes the developed scoring system: how scores are attributed to the data, how this scoring is calculated and how this scoring achieves the desired goal of providing an optimized metric for accurate, reliable and frequent data. Section V will present a case study for this system in a more restricted environment, denominated SmartHub. Finally, Section VI will present a summarized conclusion of our findings, and outline future work on the proposed system, with special focus on security.

II. THE BLOCKCHAIN

The proposed Blockchain is very similar, in structure, to existing cryptocurrency-based public Blockchains. Indeed, it is based on a Proof-of-Work (POW) scheme [2], with some key differences (Figure 1):

 ISBN: 978-1-61208-683-5

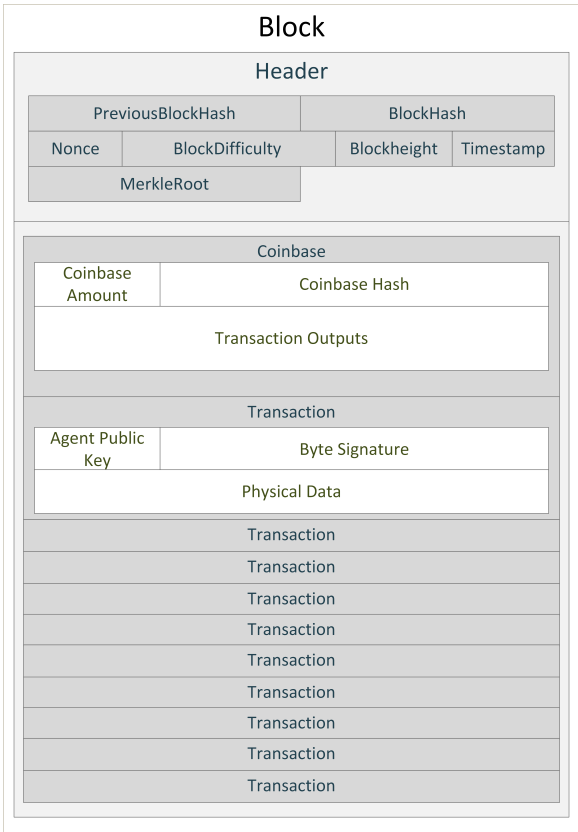

Figure 1. The block structure.

- The unit of transaction is the data collected by an agent from a specific source, tagged with a time-stamp and the agent's identity;
- There is no complex conditional scripting over these transaction;
- The value of a transaction is translated into a score, which makes up the transaction outputs, as opposed to a coin-based system;
- The cumulative score of an agent is monotonically increasing;
- This cumulative score provides a way to gauge the importance of an agent to the system, as frequently participating agents accrue higher scoring;
- Extracting a median score per transaction provides a way to gauge the accuracy of an agent, as accurate agents accrue higher score per transaction.

In the developed Blockchain, a POW schema was chosen in detriment of other schemes for ensuring investment in the Blockchain mainly for its simplicity, the adequacy to potentially resource constrained devices, as well as to avoid different tiers of agents and further implementation complexity. To interact with the Blockchain, the intelligent agents are required to have a cryptographically secure identity based on an asymmetric public and private key-pair. The public identity is the one which is referenced and tracked by the Blockchain, with the data supplied by an agent being signed with its private key, effectively rendering it tamper-proof, as well as establishing irrevocable authorship. The Blockchain is made available to these agents and any other application as a completely stand-alone library.

Advocating for open-source artifacts, all the produced software was published, in GitHub, under a MIT License [3].

A. Architecture

The developed Blockchain (Figure 2) considers three main entities: the agents that participate in building and maintaining the ledger, identified as ledger agents; the agents responsible for supplying ledger agents with processed data, known as slave agents; and data generating devices, like sensors, which may supply data to slave agents or directly to ledger ones. A group of the referred entities, connected between themselves, constitute what is denominated a Smart-Hub (Figure 2a). On the other hand, the MAS Blockchain consists of several hubs and ledger agents that opted to live outside specific hubs (Figure 2b). Each hub may be applied to distinct domains such as government management, road safety and weather forecast, among many others.

B. Data Integrity

To ensure that transactions are not lost, the agent that generates them must keep track of which transactions it issued and are not yet present in a committed block. To ensure that such transactions will populate a block, there must be a periodic diffusion of the transactions that are not yet committed. No restrictions are applied as to the time that transactions are issued, and score is not deducted at any point from a given identified agent. Due to this lack of restrictions over the validity of gathered data, and in contrast to cryptocurrency-based schemes [2] [4], a transaction is never invalidated (as would be the case for insufficient funds or cancellation). As long as the agent stores its transactions, data gathered is guaranteed to never be lost and will eventually be recorded in the Blockchain later in time.

C. Data Storage

Only the ledger agents are required to keep a copy of the ledger, with slave ones being just responsible for gathering data. The data representation is left completely configurable, provided that there is a clear distinction between each type of data. The size of such data must be calculable, as the block size is fixed (as in [2]).

Two basic assumptions are key for data storage: the first is that data exists at a point in time and the second is that it may be tied to a geographical location. As such, all data is required to be tagged by the agent through a hardware clock with the time at which that data was gathered, either by the sensor or the agent's clock. For geographically significant data, geographic coordinates can be supplied to increase data accuracy, which will impact later calculations.

 ISBN: 978-1-61208-683-5

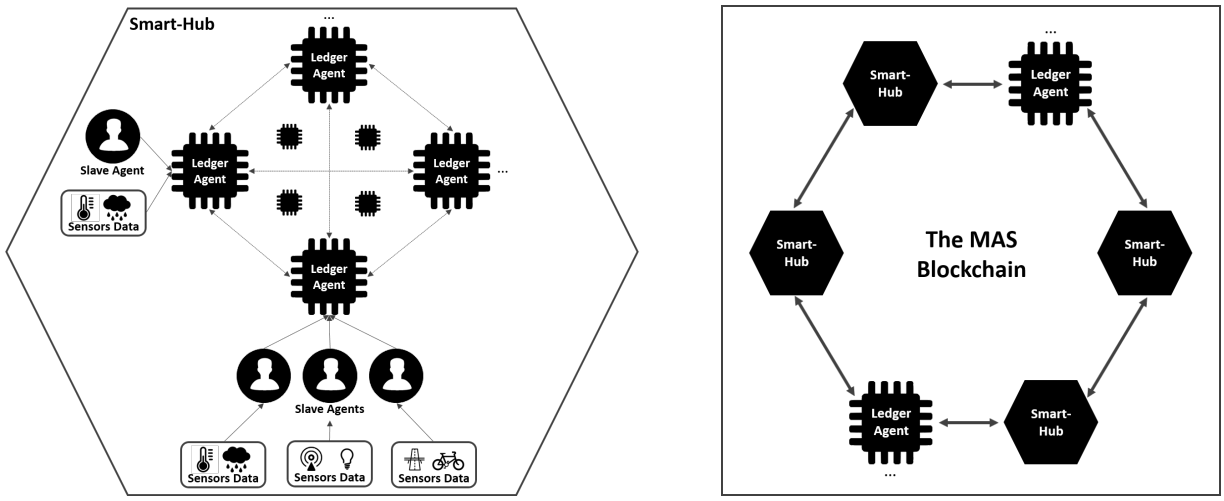

Figure 2. The Blockchain Architecture: (a) a Smart-Hub comprising Ledger and Slave agents; and (b) the MAS Blockchain comprising multi Smart-Hubs and Ledger Agents.

III. Agent Interaction

This section describes the agents' main interactions with the Blockchain (Figure 3), which are synthesized by two main behaviors: block mining and data capture. Two additional behaviors, related to data synchronization, are briefly described. It is worth noting that the Blockchain is completely independent and unaware of the agent's platform.

A. Start-up

Initially, synchronization with other running agents of the Blockchain is made through sequential transfer starting from the latest committed block of the starting agent. The mechanism for peer exchange is left up to the implementation. The agent then begins executing the succeeding behaviors.

B. Mining

To reduce computational load, agents will mine a block only once one is full interleaved with data capture into subsequent blocks. Stricter synchronization must be enforced to avoid long temporary forks, which would result in both large overhead in computation, for block validation, as well as network overhead, transmitting full blocks. Blocks are considered full only when they are populated with enough transactions as to achieve either a fixed amount of ceiling or fill the maximum allocated block size. A fixed amount of transactions help avoid extended delays in generating blocks when only very small transactions are added.

C. Data Capture

Each agent manages its own sources of data, originating in its slave devices, such as electronic sensors or other agents, as well as timings and priorities given to these sources in order to maximize their score. For this main behaviour, through which transactions are generated, an agent chooses one of these pre-configured sources to extract data and generate a transaction. This transaction is inserted to the latest block in construction and must be propagated to other agents with support from its underlying agent platform. These sources of data may be any known Application Programming Interface (API) which allows its masters to poll and extract this data.

D. Synchronization

The inherent problem of consensus in such a distributed system is still present. Therefore, mechanisms similar to those in established cryptocurrencies must be performed by the agents. These are generalized in a synchronization behavior where agents compare and exchange transactions to record in the Blockchain, as well as block headers and full blocks when necessary. This behavior has to be executed concurrently with data capture, as synchronization of known transactions is important to reduce overall computational load. Moreover, block headers are propagated between agents when mining results in a nonce that fulfills the difficulty requirements of the block. When tight synchronization of transactions is ensured, only the block header needs to be propagated, as the same already validated transactions will be present in each individual agent's current block and only the nonce and time-stamp differ in the header.

IV. Scoring - A Proof-of-Confidence Metric

Work has been done in the context of public Blockchains to establish trust between cooperating peers which hold wallets and ledgers in more traditional cryptocurrency-based Blockchains. Such is the case of the NEM Blockchain [4] which relies on a Proof-of-Importance scheme (POI) and maintains peer reputations through the use of the Eigentrust++ [5] algorithm. This approach, however, focuses more on trust between peers based on their interactions than the values they supply in the form of transactions. The scoring approach taken for this paper is thus similar to the POI scheme, as it attributes

 ISBN: 978-1-61208-683-5

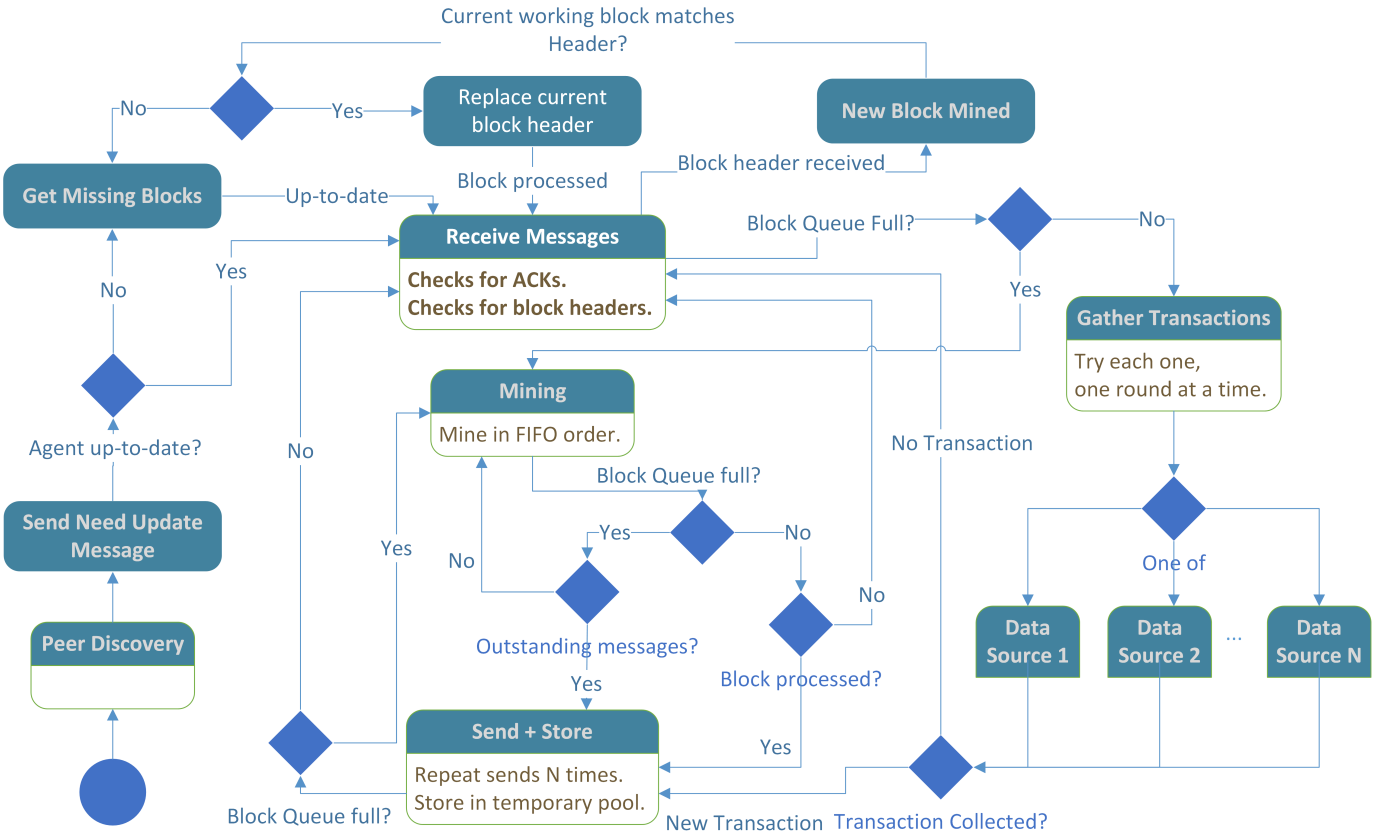

Figure 3. Activity diagram describing the agents' life cycle.

increasing importance to certain agents, but also distributing block mining equitably to all participating agents.

Scoring is the aforementioned optimizable metric and it is calculated by taking into consideration relative measurements. An agent's goal is to maximize this score, enabling a gamified environment where agents compete for the highest score. These scores are attributed to the agents that supply the transactions through the block's special coinbase transaction, and summed to its aggregated total. As usual in cryptocurrency-based Blockchains [2], each transaction is recorded such that it references which transactions were taken into account to produce the addition to the agent's outstanding score and the latest known total score for such agent prior to the new block. Thus, the main difference between scoring and traditional mined cryptocurrency comes from the fact that there is no reliance on a fixed lump-sum [2], but instead the coinbase is incrementally constructed, adding scores to each agent that adds transactions to the block.

A. Scoring Formula

The scoring is calculated via relative measurements, namely ratios between data values and time-stamps, possibly restricted to a configurable geographic radius (for geographically tagged data). In order to ascribe higher scores to data displaying desirable traits such as accuracy, reliability and frequency, a contract was established.

1) Contract:

- Any data that populates the Blockchain must, at least, allow for comparison against a previous transaction of the same type. As such, data must be identifiable by type or category (such as temperature, humidity or traffic data) and this category must allow some type of partial ordering;
- The comparison must result in a ratio of the previous value and the current value. These values depend on the type of data being measured.

2) Relative Calculations: Due to the nature of a POW metric, and the fact we consider arbitrary agents, the presence of monotonic clocks across all possible agents can not be guaranteed and, so, it is possible to commit data that is older than one would expect. As such, the insertion of data that is older than some existing data of the same category in a small geographical radius is likely to happen. The approach taken in this paper to rank data is one that is simple, although requiring complex querying over the Blockchain. Indeed, it comes from the observation that physical data tends toward homogeneity for reduced time-frames: taking, for example, temperature readings, it is highly probable that within a small region and considering a time-frame of a few seconds, temperature readings will not differ significantly. This assumption allows the protocol used for synchronization between peers to

 ISBN: 978-1-61208-683-5

avoid additional resource usage and complexity accrued by employing time synchronization approaches, such as NEM's Blockchain [4] time synchronization protocol, based on [6] a more classic approach taking from [7]. In addition, instead of maintaining a complex scoring system, such as PageRank-based algorithms [8]), a simple short-term memory formula was developed to ease resource usage in score attribution.

3) Formula: The formula to calculate the score of a given transaction is broken into three different components, and is predicated on a comparison between the transaction to be added, noted with subscript a and the transaction closest in time to it (as measured by the time-stamp) present in the Blockchain or in the currently constructed block, of the same type, noted with subscript c:

- The first component extracts a relative measurement between the two time-stamps, noted as δ_t:

$$\delta_t = \frac{|t_c - t_a|}{t_c} \tag{1}$$

- The second component extracts a relative measurement between the values present in the data, noted δ_v, for each value i or j in the set of V_c and V_a values present in each:

$$\delta_v = \frac{|\sum v_{ci} - \sum v_{aj}|}{\sum v_{ci}}, i \in V_c, j \in V_a \tag{2}$$

- The final component is a small additive base component, which guarantees a base incentive to provide data, noted as $base$.

It must be noted that for geographically tagged data, a configurable and reasonable radius must also be defined, in which the transaction closest in time must be restricted to, in relation to the transaction to be added. These three components are then composed in order to emphasize and prioritize either the δ_t, such that frequency and reliability of the data is valued more highly; the δ_v, such that stricter homogeneity of values are valued more highly; and the base component, which incentives data collection regardless of quality and which, as to not overshadow the previous components, should be strictly smaller, preferably by, at least, one order of magnitude.

B. Data Guarantees

It should be stressed out that although no guarantees are given that malicious agents are barred from cluttering the Blockchain with arbitrary data, such agents would be easily flagged due to their naturally low scoring, either through their cumulative score or their median score per transaction. Since this scoring system effectively maintains a short term memory, as it always compares transaction data to that which is closest in time to it, although abnormal events (considering for example a fire close to a temperature sensor) result in initially poorly scored readings, subsequent readings will score higher by being compared to the previous low scoring reading.

These four key properties provide what we call a Proof-of-Confidence (POC) since this Blockchain allows for both data and agents to be ranked according to some measure of confidence, in low resource environments.

V. SMART-HUB

A simplified model of a smart city was used to develop a prototype implementation of the described system, as well as develop the main scoring formula, in which five fixed different categories of data where considered, all geographically tagged:

- Temperature data;
- Humidity data;
- Luminosity data;
- Noise data;
- Other data.

A strict categorization approach was followed in which this data not only follows the previously defined hierarchy, but is divided into classes according to its type, and in conformance with the established scoring contract.

A composition formula, developed and used in this case study, is one that values δ_t over δ_v with a very small base component, effectively prioritizing a dense timeline of data.

$$\frac{(\delta_t \cdot base_t)^2 \cdot \frac{\delta_v \cdot base_v}{2} + base}{divisor} \tag{3}$$

where,

$$base_t = 5,\ base_v = 2,\ base = \frac{1}{3},\ divisor = 50000 \tag{4}$$

The restriction radius of geographically tagged data considered for this case study is based on a percentage deviation lower than 0.1% of the transactions geographical coordinates, as it centers around a very small scoped region, of the scale of a factory or warehouse.

The entire system was developed from the ground up to be able to run on a single Java Virtual Machine (JVM) instance, enforcing the previously referred separation of the Blockchain as a library and implementing an agent as an application which uses this library using the JADE development framework [9]. This agent is designed to integrate with other agents of its kind to form a MAS and allow for easy configuration of data sources (configured via a documented file whose structure is not yet stabilized). The inclusion of other data was added for additional flexibility of the Blockchain, allowing arbitrary data to be inserted, albeit this data being an order of magnitude less valuable, by increasing its divisor (4) and bypassing the delta calculations and contract entirely, effectively supplying the maximum ratio of one for both deltas. There were no restrictions to geographical coordinates of the data for increased simplicity, though very tight bounds were assumed for the radius considered for calculations, as previously referred.

A block size of two megabytes was considered, as it was found that this particular Blockchain requires both tighter synchronization than the classical ones and a big increase in computational load in verifying block integrity. This considerably large block size allows for a big bulk of transactions to

 ISBN: 978-1-61208-683-5

be supplied into the Blockchain at a time to compensate for the extra resource usage due to these characteristics. To provide long-term storage support, as well as complex querying capabilities, use of an embedded database management system was planned.

Throughout the development process, certain characteristics of the Blockchain were observed and conclusions extrapolated over certain key aspects, two of them, already mentioned above:

A. Computational Load

It was observed that the use of the POW scheme, coupled with the restrictions of these IoT devices, lead towards potentially undesirable computational loads, via uninterrupted mining and validation of network transmitted blocks, thus the restriction of mining only to completely full blocks.

Of note as well is the high cost in validating blocks transmitted through the network, due to the need for comparisons between potentially distant transactions in the chain, in order to recalculate scores and ensure they are correct. In order to do so, a multitude of costly queries may have to be run against the Blockchain, and as such, tighter synchronization is a must. However, in data scarcity environments, this can prove to slow down block throughput considerably. The trade-off was made to ensure less computational load, as such a general purpose Blockchain lends itself more naturally to long-term analysis, leaving more urgent applications to be solved by more local context solutions.

B. Memory Footprint

Having a considerably large block size, as well as having very few restrictions on the types of data that can potentially be supported, leaves a memory footprint potentially unacceptable for the more embedded devices, making the ledger maintaining agents more adequate to devices in a management role, such as data aggregating devices.

C. Network Bandwidth

Due to aforementioned large block size, it might be unfeasible for very restricted devices in terms of network bandwidth to participate in the Blockchain.

VI. Conclusion and Future Work

From the recorded observations, we can concluded that the conceived Blockchain model is best suited for applications that are non time-critical, but, instead, favor data management, storage and long-term analysis. It is also concluded from the conceived scoring system that the developed Blockchain is able to provide incentives for agents to supply data that tends towards accuracy, reliability and frequency. However, no strong guarantees can be provided in such a public context, due to the potential presence of malicious agents. We note an interesting property of our system, which motivates the focus of future work: as we value frequent and reliable data, agents and data sources which fail to produce such data can be identified and, if necessary, discarded. These thresholds require more definition and tests. Future work will also focus on identifying malicious agents, signaling abnormal data, as well as identifying classes of composition formulas, such as (3), best suited for prioritization of each desired characteristic of the data. Thus, the task to ensure truly accurate, trustworthy data, if so required, is left to the implementation of data gathering mechanisms used by each agent.

Three further extensions to this Blockchain model are also proposed in an attempt to close the gap between accuracy malicious activity:

- We propose either the integration of a configurable alert system directly in the Blockchain on block committal or on the agent level, themselves monitoring Blockchain activity;
- A special temporary blacklist transaction could be included that references a fixed number of irregular instances of transactions supplied by a given agent, effectively disallowing these agents to contribute to the Blockchain for a fixed number of blocks;
- A system of captive scoring could also be implemented, in which transactions by new identities are only committed to a block after ones' identity has been recognized to have produced sufficient scoring.

Acknowledgment

This work has been supported by COMPETE: POCI-01-0145-FEDER-007043 and FCT – Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC/00319/2013, being partially supported by a Portuguese doctoral grant, SFRH/BD/130125/2017, issued by FCT in Portugal.

References

[1] W. Li, T. Logenthiran, and W. Woo, "Intelligent multi-agent system for smart home energy management," in *IEEE Innovative Smart Grid Technologies - Asia (ISGT ASIA) Bangkok, Thailand*, 2015, pp. 1–6, doi: 10.1109/ISGT-Asia.2015.7386985.

[2] "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008, URL: https://bitcoin.org/bitcoin.pdf [retrieved: October, 2018].

[3] "Prototype Implementation Repository," 2018, URL: https://github.com/Seriyin/mas-blockchain-main [retrieved: October, 2018].

[4] "NEM: Technical Reference," 2018, URL: https://www.nem.io/wp-content/themes/nem/files/NEM_techRef.pdf [retrieved: October, 2018].

[5] X. Fany, L. Liu, M. Li, and Z. Su, "Eigentrust++: Attack resilient trust management." 2012, URL: https://www.cc.gatech.edu/~lingliu/papers/2012/XinxinFan-EigenTrust++.pdf [retrieved: October, 2018].

[6] S. Scipioni, "Algorithms and Services for Peer-to-Peer Internal Clock Synchronization," 2009, phD Thesis URL: https://www.dis.uniroma1.it/~dottoratoii/media/students/documents/thesis_scipioni.pdf [retrieved: October, 2018].

[7] L. Lamport and P. M. Melliar-Smith, "Byzantine clock synchronization." in *Third Annual ACM Symposium on Principles of Distributed Computing*, 1984, pp. 68–74, URL: http://lass.cs.umass.edu/~shenoy/courses/summer04/readings/Lamport_52_byz_clock.pdf.

[8] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web." 1998, technical report, Stanford Digital Library, Technologies Project URL: http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf [retrieved: October, 2018].

[9] K. Chmiel, D. Tomiak, M. Gawinecki, P. Karczmarek, M. Szymczak, and M. Paprzycki, "Testing the efficiency of JADE agent platform," in *Third International Symposium on Parallel and Distributed Computing/Third International Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Networks*, 2004, pp. 49–66, doi: 10.1109/ISPDC.2004.49.

 ISBN: 978-1-61208-683-5