Benchmarking the efficiency of RDF-based access for blockchain environments

Juan Cano-Benito Andrea Cimmino Raúl García-Castro
Ontology Engineering Group
Universidad Politécnica de Madrid
E-mail: {jcano,cimmino,rgarcia}@fi.upm.es

Abstract

Blockchain and knowledge graphs are technologies that have become pervasive in several domains where web services have been developed relying on them. The inmutability of the data offered by the blockchain together with the capabilities of the knowledge graph when consuming data, enables web services to provide richer functionalities. Literature has explored the benefits of combining both qualitatively, and only a few works have exposed quantitatively the feasibility of combining these technologies. In particular, as far as we know no work reports the cost of storing knowledge graphs serialized in RDF into blockchains, or analyses alternatives such as virtualisers that transform on the fly data from different formats into RDF. In this paper we present an empirical analysis of the cost of storing into a blockchain in comparison with storing JSON, and the benefits when solving SPARQL queries by reading directly the RDF or using a virtualiser fed with RDF. For the sake of our experiments, we rely on different sensors that store their data into two blockchains, on top of which we perform our analysis.

Keywords: *Blockchain, Knowledge Graph, Semantic Web, RDF, RDF Virtualisation*

1 Introduction

Nowadays blockchain has become a pervasive technology in a wide range of sectors [1]. The reason is due to the fact that it allows to store data ensuring its inmutability [14]. The data stored into a blockchain may be expressed in any format and under any model. As a result, a large number of services have decided to publish knowledge graphs (KGs) relying on blockchain to store their data [21].

Blockchain has many implementations, such as Ethereum, Bitcoin, or Hyperledger Fabric. These implementations often associate a cost to the amount of data that peers write in the chain. As a result, the same data written in the chain with a verbose format have a higher cost to be paid by a peer, in comparison with having the data represented with a simpler format.

The cost of writing becomes especially relevant when blockchain is storing KGs since their data format is Resource Description Framework (RDF), which is known to be verbose. Therefore, although a KG stored in a blockchain has clear benefits when consuming data due to the RDF, this format will entail a higher cost in comparison with other lighter formats, it has also an expected higher cost. There is an ever-growing number of proposals that store a KG in a blockchain, but there is a lack of knowledge about how suitable this approach is and if other alternatives could work better.

In this paper a case study is presented in which an empirical analysis is performed in order to establish the benefits and costs of storing a KG in a blockchain. In addition, a virtualisation approach that generates virtual RDF from data expressed in JavaScript Object Notation (JSON) that is stored in a blockchain is considered. The scope of this paper is establishing how costly is storing RDF instead of JSON, and if a virtualisation approach is a better alternative that directly storing RDF in the blockchain.

The case study is contextualised in a simulated research laboratory that counts with 15 light bulb sensors, an occupancy sensor, and a temperature sensor. The sensors send data to an agent that writes such data into two different Ethereum blockchains. In one of them, data is written as plain JSON, whereas on the other one, data is expressed in RDF using the VICINITY ontology [7]. The analysis consists in measuring how costly is storing RDF and JSON in terms of gas, and how effective is querying the data is querying either data.

The analysis carried out aims at exploring the following research questions:

- H1: What has a higher cost when writing data in the blockchain, RDF or JSON?
- H2: What is faster when reading from the blockchain, RDF or JSON?

DOI reference number: 10.18293/SEKE2020-104

- H3: Considering a virtualiser that transforms on the fly JSON data into RDF. What is faster to query, RDF or virtual RDF?

The rest of the article is organised as follows: Section 2 reports proposals in the literature combining these technologies; Section 3 introduces concepts used across the paper; Section 4 presents the architecture followed in our experimental analysis; Section 5 explains how the experimentation was carried out and reports its results; and, finally, Section 6 recaps our conclusions and main findings.

2 Related Work

The approach of storing the RDF data of a KG in a blockchain has been addressed mainly from a theoretical point of view without reporting any quantitative analysis [4, 6, 8, 10, 13, 16, 24, 26, 27]. Although different proposals provide a preliminary qualitative analysis [15, 17, 22, 23], most of the works describe specific applications that have stored their KGs in a blockchain without analysing the efficiency of this decision over other alternatives [9, 11, 2, 25].

The majority of proposals address how semantic web and blockchain technologies could work jointly in order to enhance their benefits without providing any analysis of its feasibility [6, 8, 10, 16, 27]. Some proposals report a qualitatively analysis of how some specific domains could benefit from using these two technologies together. For instance, for chemistry [26], smart cities [24], publications [13], or government [4] domains.

Several proposals provide a quantitative analysis of the combination of these two technologies. Ruta et al. [22, 23] performed an analysis over the discovery of Internet of Things (IoT) resources whose meta-descriptions were stored in a blockchain using RDF. They reported discovery and query processing time over the RDF involved in such task. However, the results have not enough granularity to establish only the reading time of the RDF, nor they provide a comparison with other alternatives.

Le-Tuan et al. [17] presented a scenario of a small network of lightweight nodes. Each node processes 1 billion triples, but those triples are not stored in the blockchain that contains instead a hash pointing to an RDF online documents. Therefore, although the proposal reports the time for writing and querying data, these results do not involve directly the blockchain. As a result, the cost of writing is neither analysed or reported.

Ibañez et al. [15] studied the verbosity of RDF expressing data. They reported the number of bytes that different serialisations of RDF have when expressing the same data. In addition, authors considered the same information compressed with different algorithms. However, RDF was not stored in any blockchain, nor any cost was reported.

As a conclusion, the literature currently lacks to determine the benefits of storing KGs inside a blockchain from the point of view of the cost of writing RDF instead of other serialisations, e.g., JSON. Additionally, no work has explored alternatives like using RDF virtualisers in order to have the benefits of RDF when consuming data while storing in the chain less-verbose formats like JSON.

3 Background

Most of the concepts on top of which this paper is build are well-known, namely: RDF [5], the SPARQL Protocol and RDF Query Language (SPARQL) [12], JSON [3], and blockchain [20]. Nevertheless, others concepts are not terms widely known and, therefore, in this section they are defined.

Transaction: is the name of the operation that writes or stores some data inside a blockchain. Depending on the data size that is been written, it requires more or less space in one block. As a result, if transactions require more space than the one available in a block, they will be written in more than one block.

Usually, a transaction has a virtual cost since it requires a certain amount of computing power. As a result, performing a transaction has an associated cost in public blockchains and, depending on the implementation, it may have different names; for example, for Ethereum it is called Gas [28].

Software agent Autonomous actions in a tailored-domain environment can be done [29]. The means of the actions performed by an agent have as goal to meet a set of design requirements. A system with two or more agents is known as Multi-Agent System.

In the context of this paper, a proactive agent with simple reflexes based on condition-action is used.

RDF Virtualisation is a technique used in the semantic data integration context [18]. Usually, it refers to a piece of software connected to a data source and with a set of translation rules called mappings. These techniques are able to translate on the fly data from heterogeneous formats and models into RDF expressed according to a specific ontology, allowing to solve SPARQL queries over such data.

Virtual RDF is the one generated as an RDF virtualisation technique [18]. It receives such name due to the fact that the RDF is not stored anywhere and is consumed as produced; unless a software agent stores it somewhere.

4 Experimental Architecture

The scope of this paper is to provide an empirical analysis of how suitable is to store RDF inside a blockchain, due to the cost that it entails. Alternatively, storing JSON and using a virtualiser could bring the same benefits without the drawbacks of the former approach.

The scenario presented in Figure 1 has been endowed in order to perform the desired analysis. The scenario consists of a set of sensors which data is stored inside a blockchain using the JSON format. Besides, the same data is stored using RDF inside another blockchain. Then, relying on this infrastructure, a set of tests have been performed to find answers to the research questions reported in the introduction.

Next, the different components of the architecture are explained:

IoT Infrastructure: the sensors within the architecture are 15 light bulb sensors, 1 temperature sensor and 1 occupancy sensor. These devices send their data to the *IoT Collector* that forwards such information to the *Agent JSON Writer* and to the *Agent RDF Parser*. The data is reported by the sensors in JSON format.

Agent RDF Parser: this agent receives the JSON data from the *IoT Infrastructure* and using a fixed RDF template injects such data into the template using JSONPath expressions. Then, it forwards the instantiated RDF template to the *Agent RDF Writer*.

Agent RDF/JSON Writer: although the architecture counts with two different agents for this task, conceptually they perform the same function. Both receive a document and write it in the blockchain. The *Agent RDF Writer* stores the received RDF documents in the RDF blockchain serialised as Turtle, and the *Agent JSON Writer* stores the JSON documents received in the JSON blockchain.

Blockchain: in the architecture, data (either in RDF or in JSON) is stored in a different Ethereum blockchain. Their functionality is the same since the blockchain is agnostic to the data format.

Agent JSON/RDF Reader: the architecture counts with two different agents for this tasks that perform the same function. These agents read the information within their respectively blockchain. As a parameter, they can receive the number of transactions to be read, providing as a result the collection of documents stored in those transactions.

Virtualiser: this component in the architecture is implemented with a software called Helio[1]. It reads a number of transactions from the blockchain and, relying on a set of translation rules, generates an RDF document with all the JSON documents stored in the RDF blockchain, i.e., the *Virtual RDF*. The virtual RDF is generated so it is exactly the same of the one provided by the *Agent RDF Reader*.

SPARQL Agent: this agent receives a SPARQL query and returns the query result. Depending on how it is configured, it relies on the *Virtual RDF* or on the RDF output by the *Agent RDF Reader* to answer the query.

The goal of this architecture is to provide a playground were different measurements can be taken. First, the gas consumption when storing the RDF or the JSON documents, relying on the *Agent Writers*. Secondly, the time that takes reading RDF and JSON documents from the blockchain, relying on the *Agent Readers*. Third, the time that takes answering a query with the data stored in a set of transactions when such RDF is provided by the *Agent RDF Reader* or the *Virtualisation* component.

As a result, by performing these measurements, the research questions introduced in Section 1 will be validated, analysing the feasibility of storing RDF or JSON directly on the chain, and using a virtualiser to obtain the RDF benefits.

[1]https://helio.linkeddata.es/

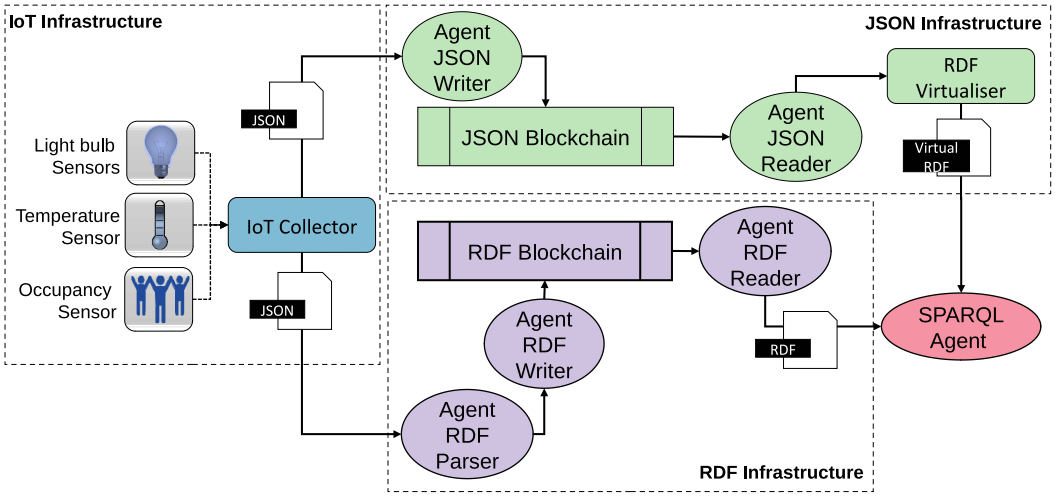

Figure 1. Experimental architecture

5 Experimental Analysis

The different experiments designed to address the three research questions formulated in this paper are reported in this section. First, the gas spent when storing RDF and JSON is measured in order to validate the first research question. Second, the time required to read from the JSON and the RDF blockchains, respectively, is measured in order to validate the second research question. Finally, the time taken to perform the same query relying on the one provided by the virtual RDF and the one provided by the RDF blockchain were measured to validate the third research question.

The best effort was done in order to prevent information loss due to the bespoke characteristics of each format. Therefore, the JSON and the RDF documents stored in both blockchains contain the same information. During the experiments we compared the results considering the same amount of transactions, namely: 2,000, 4,000, 6,000, 8,000, 10,000, 12,000, and 14,000.

All these tests have been carried out on a computer with the following characteristics: intel i7 6700k, 32 Gb RAM and 1Tb SSD.

Finally, all the times reported as box plots are measured in seconds reporting the results of executing 10 times each experiment. The test performed to establish if the results have statistically significant differences is the well-known Iman–Davenport test [19], with a confidence level of 95%. This test outputs a p-value; if this value is below 0.05 it means there are no statistically differences between the results, i.e., they can be considered the same.

5.1 Gas consumed storing RDF vs JSON

In this experiment RDF and JSON documents were stored in different blockchains. Both documents contained the same information; however, data expressed in RDF required around 6,000 characters, whereas JSON data required approximately 550 characters to encode the same information. Figure 2 depicts the gas consumed storing sets of RDF and JSON documents containing equivalent information.

As it can be observed, storing data in RDF requires for each transaction, on average, an amount of gas that is 10 times more than the one required by the information serialised in JSON. In this case there is no need of applying any statistical test since the magnitude of such difference makes results clear.

5.2 Time required to read transactions

In this experiment we measured the time that took reading a set of transactions from the JSON and the RDF blockchains, respectively. In addition, the time for the JSON data to be sent to the Helio virtualiser is included in the results. Figure 3 depicts the results of this experiment.

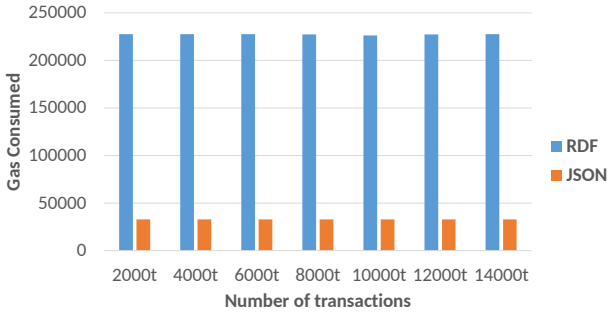

Figure 2. Gas consumed by RDF and JSON

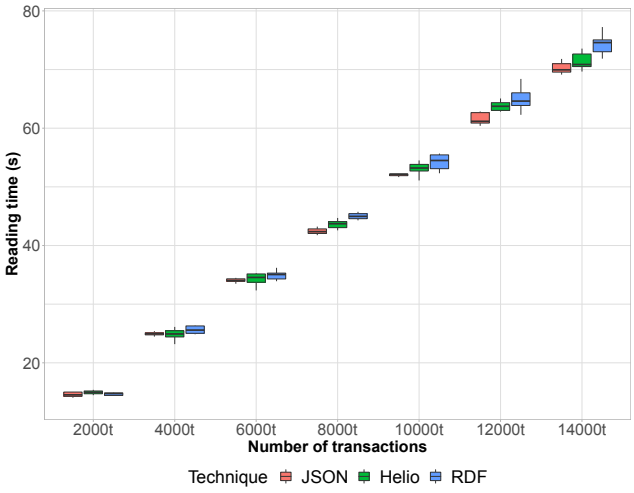

Figure 3. Time for reading the transactions

As it can be observed, the reading times for the three cases are close enough. The statistical test applied over their average values in order to ensure their statistical equivalence returns the following p-values: between JSON and Helio is 0.13, between JSON and RDF is 2.66×10^{-4}, and between Helio and RDF is 0.02×10^{-4}. With this p-values, it can be concluded that reading times are statistically equivalent between JSON and Helio. Instead, between JSON and RDF, and Helio with RDF, there is a statistical difference. As a result, reading JSON, and optionally feeding the Helio virtualiser, is faster than just reading the RDF.

5.3 Issuing SPARQL queries

In this experiment the time that took reading the blockchain plus the time that takes solving a SPARQL query was measured. The query issued asked about all the

known data in the blockchain. Figure 4 depicts the results of this experiment.

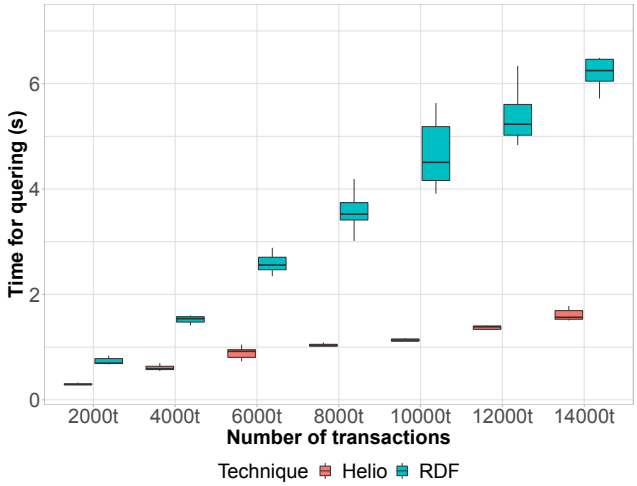

Figure 4. Time for querying all the data in the blockchain

At the light of these results, using the virtualiser Helio allows to solve the SPARQL queries faster than just aggregating RDF documents from the *Agent RDF Reader*. The difference is due to the fact that the translation is fast and produces a whole RDF document; instead, the *Agent RDF Reader* needs to aggregate all the documents into a single one before solving the query. This behaviour explains the linear growth of the results in the chart.

The statistical test outputs a p-value of 0.02; therefore, it can be concluded that there are no statistical differences and, thus, using a virtualiser in this context is the same than reading and querying the RDF directly.

6 Conclusions

This article presents an empirical study that aims at answering the research questions proposed in section 1. These questions revolve around if storing RDF in a blockchain is efficient, and if alternatives exist in order to keep the benefits of RDF but avoiding its drawbacks. The experimental results led to the following answers:

RQ1: at the light of the results reported in subsection 5.1 we can conclude that writing RDF is more than 10 times more expensive than writing JSON.

RQ2: results from sub-section 5.2 advocate that reading JSON is faster reading than RDF; even feeding with the read data a virtualiser is faster than reading RDF.

RQ3: sub-section 5.3 proofs that querying the virtualiser is faster than reading and querying the RDF from the blockchain.

As a conclusion of our empirical analysis, storing RDF in a blockchain brings clear benefits for consuming data, e.g., been able to query semantic data, use standardized models or bring the benefits of link-data. However, RDF has some drawbacks: i) reading the data from the blockchain takes more time than reading the same data in other format like JSON, and also, ii) writing RDF in a blockchain has an elevated cost in terms of gas.

As a result, in this paper a virtualiser to translate on the fly JSON into RDF was analysed. The experimental results achieved proof that using a virtualiser under the studied circumstances is more efficient than using RDF. It has the same benefits, but none of its drawbacks.

In the future, this analysis will be extended considering SPARQL query rewriting techniques, which could be even more efficient than using virtualisers. Also other parameters will be studied, such as scalability and memory usage.

7 Acknowledgements

This paper was written in the context of the DELTA European project, and thus has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 773960.

References

[1] J. Al-Jaroodi and N. Mohamed. Blockchain in industries: A survey. *IEEE Access*, 7:36500–36515, 2019.

[2] A. Banerjee and K. Joshi. Link before you share: Managing privacy policies through blockchain. In *2017 IEEE International Conference on Big Data*, pages 4438–4447. IEEE, 2017.

[3] L. Bassett. *Introduction to JavaScript object notation: a to-the-point guide to JSON*. O'Reilly Media, Inc., 2015.

[4] T. Beris and M. Koubarakis. Modeling and preserving Greek government decisions using semantic web technologies and permissionless blockchains. In *European Semantic Web Conference*, pages 81–96. Springer, 2018.

[5] D. Brickley, R. V Guha, and B. McBride. RDF schema 1.1. *W3C recommendation*, 25, 2014.

[6] J. Cano-Benito, A. Cimmino, and R. García-Castro. Towards blockchain and semantic web. In *International Conference on Business Information Systems*, pages 220–231. Springer, 2019.

[7] A. Cimmino, V. Oravec, F. Serena, P. Kostelnik, M. Poveda-Villalón, A. Tryferidis, R. García-Castro,

S. Vanya, D. Tzovaras, and C. Grimm. VICINITY: IoT semantic interoperability based on the Web of Things. In *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 241–247. IEEE, 2019.

[8] J. de Kruijff and H. Weigand. Understanding the blockchain using enterprise ontology. In *International Conference on Advanced Information Systems Engineering*, pages 29–43. Springer, 2017.

[9] M. Demir, O. Turetken, and A. Ferwom. Blockchain and IoT for delivery assurance on supply chain (BIDAS). In *2019 IEEE International Conference on Big Data*, pages 5213–5222. IEEE, 2019.

[10] M. English, S. Auer, and J. Domingue. Blockchain technologies & the semantic web: A framework for symbiotic development. In *Computer Science Conference for University of Bonn Students*, pages 47–61, 2016.

[11] D. Graux, G. Sejdiu, H. Jabeen, J. Lehmann, D. Sui, D. Muhs, and J. Pfeffer. Profiting from kitties on ethereum: Leveraging blockchain RDF data with SANSA. SEMANTiCS Conference, 2018.

[12] S. Harris, A. Seaborne, and E. Prud'hommeaux. SPARQL 1.1 query language. *W3C recommendation*, 21(10):778, 2013.

[13] M. R Hoffman, L. Ibáñez, H. Fryer, and E. Simperl. Smart papers: Dynamic publications on the blockchain. In *European Semantic Web Conference*, pages 304–318. Springer, 2018.

[14] F. Hofmann, S. Wurster, E. Ron, and M. Böhmecke-Schwafert. The immutability concept of blockchains and benefits of early standardization. In *2017 ITU Kaleidoscope: Challenges for a Data-Driven Society (ITU K)*, pages 1–8. IEEE, 2017.

[15] Luis Daniel Ibáñez, Huw Fryer, and Elena Paslaru Bontas Simperl. Attaching semantic metadata to cryptocurrency transactions. In *DeSemWeb@ISWC*, 2017.

[16] Henry Kim, Marek Laskowski, and Ning Nan. A first step in the co-evolution of blockchain and ontologies: Towards engineering an ontology of governance at the blockchain protocol level. *SSRN Electronic Journal*, 2018.

[17] A. Le-Tuan, D. Hingu, M. Hauswirth, and D. Le-Phuoc. Incorporating blockchain into RDF store at the lightweight edge devices. In *International Conference on Semantic Systems*, pages 369–375. Springer, 2019.

[18] M. Lefrançois, A. Zimmermann, and N. Bakerally. A SPARQL extension for generating RDF from heterogeneous formats. In *European Semantic Web Conference*, pages 35–50. Springer, 2017.

[19] D. G Pereira, A. Afonso, and F. Medeiros. Overview of friedman's test and post-hoc analysis. *Communications in Statistics-Simulation and Computation*, 44(10):2636–2653, 2015.

[20] M. Pilkington. Blockchain technology: principles and applications. In *Research handbook on digital transformations*. Edward Elgar Publishing, 2016.

[21] S. Porru, A. Pinna, M. Marchesi, and R. Tonelli. Blockchain-oriented software engineering: challenges and new directions. In *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, pages 169–171. IEEE, 2017.

[22] M. Ruta, F. Scioscia, S. Ieva, G. Capurso, and E. Di Sciascio. Semantic blockchain to improve scalability in the Internet of Things. *Open Journal of Internet Of Things (OJIOT)*, 3(1):46–61, 2017.

[23] M. Ruta, F. Scioscia, S. Ieva, G. Capurso, and E. Di Sciascio. Supply chain object discovery with semantic-enhanced blockchain. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, pages 1–2, 2017.

[24] M. Ruta, F. Scioscia, S. Ieva, G. Capurso, G. Loseto, F. Gramegna, A. Pinto, and E. Di Sciascio. Semantic-enhanced blockchain technology for smart cities and communities. In *3rd Italian Conference on ICT for Smart Cities & Communities (I-CiTies 2017)*, 2017.

[25] M. Sicilia and A. Visvizi. Blockchain and OECD data repositories: opportunities and policymaking implications. *Library hi tech Journal*, 2019.

[26] J. J Sikorski, J. Haughton, and M. Kraft. Blockchain technology in the chemical industry: Machine-to-machine electricity market. *Applied Energy*, 195:234–246, 2017.

[27] H. Ugarte. A more pragmatic Web 3.0: Linked blockchain data. 2017.

[28] G. Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 2014.

[29] M. Wooldridge and N. Jennings. Intelligent agents: Theory and practice. *The knowledge engineering review*, 10(2):115–152, 1995.