Towards Trustworthy Financial Reports Using Blockchain

Gianfranco D'Atri

Department of Mathematics and
Computer Science,
Calabria University,
Rende, Cosenza, Italy
Email: gdatri@mat.unical.it

Van Thanh Le, Claus Pahl , Nabil El Ioini

Faculty of Computer Science,
Free University of Bolzano,
Bolzano, Italy
Email: {vanle,cpahl,nelioini}@unibz.it

***Abstract*—The need to develop a system for dealing with the transparency analysis of financial reports has pushed companies to look for possible solutions to store their data in a reliable and trustworthy database, that enables all authorized entities to access and check financial data of their partners. The eXtensible Business Reporting Language (XBRL) is the digital format of financial reports that provides data and rules to perform different analysis following a number of techniques: (1) consistency calculation, (2) rates between debts and interests, (3) checking the Benford's law, (4) financial item value comparison. In this paper, we propose a blockchain based solution where all reports analysis activities and results are recorded into a shared ledger to guarantee their integrity and consistency. Specifically, we have designed and implemented a prototype to validate and store financial statements using Ethereum blockchain. Additionally, we have performed an initial set of tests based on a set of Italian financial reports.**

***Keywords*–Blockchain; XBRL; Financial Reports; DLV; ASP.**

I. Introduction

Financial statements are formal records of the financial activities that companies use to provide an accurate picture of their financial history. Their main purpose is to offer all the necessary data, which allows for an accurate assessment the economic situation of a company and its ability to attract investors.

In the Italian context, financial statements start with business accounting collecting all the relevant financial data, processing and validating its consistency, then generating a standard eXtensible Business Reporting Language (XBRL) format report (i.e., XBRL is a standard digital format for financial reports). The report is then sent to the Chambers of Commerce (a.k.a board of trade, an association or network of businesspeople designed to promote and protect the interests of its members [1]). After a series of checks (e.g., checking consistency between inputs and outputs), the Chamber of Commerce publishes the reports in a publicly accessible domain (i.e., registroimprese.it).

Two of the issues that arise in the current approach for report evaluation is its incompleteness in terms of evaluation method (e.g., checking the format) and its lack of traceability in report updates, which might prove to inconsistency and lack of trust among business organizations, in other words, in many jurisdictions, the reliability and consistency of published data is not yet assured by public bodies. To this end, our goal is to investigate how blockchain can be used to address these limitations to restore trustworthiness in the published financial reports. Our contribution is two fold (i) provide a methodology to automatically evaluate and validate the consistency of the generated reports, *(ii)* use Ethereum smart contract to store financial reports and track all updates that might take place in the future. Additionally, an initial set of experiments is presented to illustrate the cost factor of the proposed approach.

The remaining of this paper is organized as follows: Section II provides background information about the used technologies. Section III discusses the main related work studies connected to our work. Section IV describes the system architecture. Section V presents the implementation details. Section VI experimentally evaluates the cost and performance of our approach and Section VII gives our conclusions.

II. Background

The following section introduces the different technologies used in the definition of the proposed architecture.

A. XBRL

Financial reports contain sensitive data that might have a huge impact on organization's future in terms of investments and collaborations, which mandates careful management and control mechanisms able to capture any inconsistencies or manipulation of the published reports. The first step towards this goal started with the introduction of the eXtensible Business Reporting Language [2], which is the world leading standard for financial reporting. It facilitates inter-organizations communication and enables automatic reports processing and analysis. XBRL relies on XML and XML based schema to define all its constructs. Its structure consists of two main parts:

1) XBRL instance, containing primarily the business facts being reported (see figure 1).

```
<rp:RevenueTotal unitRef="EUR">5000</rp:RevenueTotal>
<rp:CostOfSales unitRef="EUR">3000</rp:CostOfSales>
<rp:GrossProfit unitRef="EUR">2000</rp:GrossProfit>
```

Figure 1. Facts example

2) XBRL taxonomy, a collection of arcs which define metadata about these facts and their relationship with other facts (see figure 2).

Figure 3 depicts XBRL structure and the relations between the different components.

```
<loc xlink:type="locator"
  xlink:href="taxonomy#rp_RevenueTotal" />
<loc xlink:type="locator"
  xlink:href="taxonomy#rp_CostOfSales" />
<loc xlink:type="locator"
  xlink:href="taxonomy#rp_GrossProfit" />
<calculationArc xlink:type="arc"
  xlink:from="rp_GrossProfit" xlink:to="rp_RevenueTotal"
  weight="1" />
<calculationArc xlink:type="arc"
  xlink:from="rp_GrossProfit" xlink:to="rp_CostOfSales"
  weight="-1" />
```

Figure 2. XBRL Linkbase example

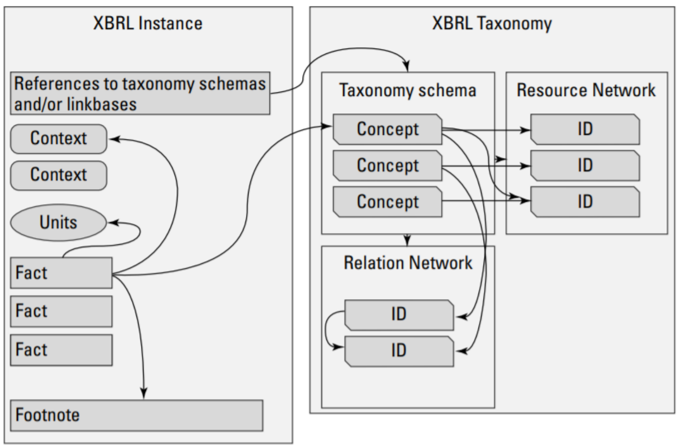

Figure 3. XBRL Structure

B. I-DLV

As the complexity of XBRL structure increases, it could reach a high number of definitions, which makes it impractical to check and validate manually. Thus, a number of tools have been developed to automate the validation process, Answer Set Programming (ASP) is a form of declarative programming oriented towards difficult search problems, highly used in both academia and industry.

The possible use of an ASP language for analyzing XBRL financial reports was explored by Gianfranco d'Atri in [3]. The tokenization and standardization of data supported by the XBRL Consortium allow an extensive and meaningful use of AI techniques to support economic analysis and fraud detection.

I-DLV [4] is a new intelligent grounder of the logic-based Artificial Intelligence system DLV [5], it is an ASP instantiator that natively supports the ASP standard language. Beside ASP features, external computation in I-DLV is achieved by means of external atoms, whose extension is not defined by the semantics within the logic program, but rather is specified by means of externally defined Python programs, the so-called external atom in the rule bodies, which are also one of the most outstanding of I-DLV. Because of these features, in the paper, we applied DLV queries to analyze and absorb valuable knowledge from financial reports.

C. Blockchain

Blockchain is a distributed, decentralized ledger to store transactions and addresses the double-spending problem in a trust-less peer-to-peer network, without the need for a trusted third party or an administrator. Blockchain is maintained by a network of computers called nodes. Whenever a new transaction arrives, the nodes verify its validity and broadcast it to the rest of the network. The main building blocks of a Blockchain are [6]:

- Transactions, which are signed pieces of information created by the participating nodes in the network then broadcast to the rest of the network.
- Blocks, that are collections of transactions that are appended to the blockchain after being validated.
- A blockchain is a ledger of all the created blocks that make up the network.
- The blockchain relies on Public keys to connect the different blocks together (similar to a linked list).
- A consensus mechanism is used to decide which blocks are added to the blockchain.

Generally, there are three types of blockchain platforms: public , consortium, and private [7]. In the public blockchain all participants can execute and validate transactions. In consortium blockchain, the identity of the participants is known, but they do not necessarily trust each other. The network is moderated by one or more participants to keep access under control. Different participants might have different roles. In a private blockchain instead, the whole network is managed by one single organization. In our context, we apply public blockchain to publish financial reports to the public, where all participant could check business working status.

In our case, Ethereum [8] is the best candidate, since it is an open source blockchain platform that enables developers to build and deploy decentralized applications. The platform runs smart contracts, a computer protocol running on top of a blockchain, performing as a contract. Smart contracts can include data structures and function calls that are executed in a centralized fashion. This guarantees the fact that the contract execution will persist on the chain.

III. Related Work

Providing trustworthy financial data is a challenging endeavor. Over the years different tools have been developed to analyze the financial information generated by companies to in order to check its consistency and integrity. However, since most of the proposed tools rely on third party organizations, issues related to trustworthiness and privacy still need to be solved.

Recently blockchain has found applications in different domains including IoT [9] [10], finance [11], health care [12] and others. In the literature, a number of studies considered the implication of blockchain on financial services and accounting. Byström [13] argues that blockchain can help corporate accounting in many ways, especially in terms of trustworthiness in accounting information and data availability in a timely manner. In [14], the authors discuss how blockchain can be an enabler technology for accounting ecosystem auditing and transparency. In [15], Colgren discusses the advantages that blockchain can bring to companies by allowing a fast and public access to companies financial statements. In [11], Bussmann has given a more general overview on the potential disruption of blockchain on the Fintech market. For banking services, Ye Guo [16] suggests that blockchain is able to

 ISBN: 978-1-61208-703-0

replace the banking industry as external and internal issues like economic deceleration and increasing credit risk and non-performing assets. Thus, blockchain could synchronize and verify financial transactions to eliminate the problems of subsequent reconciliation. Applying blockchain as a storage, Sven Helmer et al. built MongoDB database functions into Ethereum in [17], that separates the driver and database to reduce the cost transactions. The main goal of their approach is to keep all data on-chain.

In terms of tools related to XBRL, a number of tools are in use, however, they are not able to guarantee the long term trustworthiness of the reports on produced. With regard to analysis of financial report in XBRL format, Arelle [18] is an open source platform for XBRL financial reports format analysis. Users can view the structure of a document and use features with a GUI. Arelle provides many services that can be integrated with other technologies. Altova [19] is also well-known based on the XML development. With the help of Altova, users can present XBRL maps and relationships inside, including facts, context and arcs. These tools have their own evaluation tools but just check with basic concept even with some specific documents, so the result is not consistent. Moreover, considering the transparent characteristics of financial documents, we need a better approach that guarantees transparency of the whole validation process.

IV. System Architecture

The goal of the proposed architecture is to provide an end to end solution that leverages different technologies for managing financial reports and a trustworthy publishing and updating.

Figure 4 depicts an overview of the proposed architecture. It is divided into three main components: XBRL Reader, XBRL Evaluator and XBRL Storage.

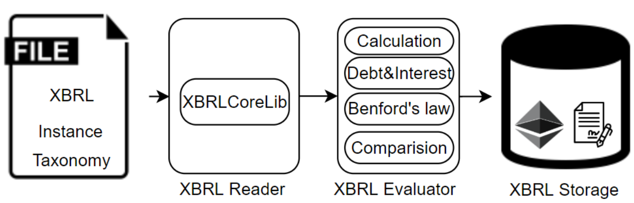

Figure 4. Architecture

A. XBRL Reader

XBRLReader is responsible for validating the XBRL formatting by checking that all the schema is fully described. It takes as input an XBRL Instance that contains facts and a link to the taxonomies to be used.

The output of XBRLReader is a list of facts and arcs that are given to the XBRL Evaluator.

B. XBRL Evaluator

Facts and arcs from the first step are evaluated following these aspects:

- Calculation consistency will check each value of facts, even if the value is aggregated from other asset's values like the example $GrossProfit = RevenueTotal - CostOfSales$, we will compare the result of $RevenueTotal - CostOfSales$ and $GrossProfit$ value with a threshold, the check applies for all the assets in the report, this kind of check also shows the errors inside reports where the difference between the actual value and the calculated value is greater than the threshold.
- The rate between interest and debt: a financial report normally shows data in 2 consecutive years, it could calculate changes of $interest/debt$ ratio during the years, if the index is too high, an alert is crucial for the company because it could be a potential sign for bankruptcy.
- Financial item comparison: From many reports in a year, we also compare financial item values among businesses to find, for example, the company has the highest revenue, or even filter companies do not have cost of warehouse.
- Benford's law checking: Benford's law [20] is an observation about the frequency distribution of leading digits in real-life data sets. The law states that a set of numbers is said to satisfy Benford's law if the leading first digit d ($d \in 1, .., 9$) occurs with probability (see figure 5):
$P(d) = \log_{10}(d+1) - \log_{10}(d) = \log_{10}(\frac{d+1}{d})$

d	$P(d)$	Relative size of $P(d)$
1	30.1%	
2	17.6%	
3	12.5%	
4	9.7%	
5	7.9%	
6	6.7%	
7	5.8%	
8	5.1%	
9	4.6%	

Figure 5. Benford's law for the first digit

The complicated formula are explained in [21] about stock prices example with distributions. Benford's law could check the whole data set or each financial reports.

We note that the evaluation process can result in valid reports meaning that they satisfy all the pre-defined evaluation criteria or invalid reports that violate one or more requirements. At this point, it is up to the report owner to decide whether to publish the report or not. We also note that if invalid reports are published, they can be updated subsequently (e.g., add more information) to a valid state.

C. XBRL Storage

Storing financial data in a trusted location is a necessity to keep data safe and to be able to trace all the updates occurring over time. The main pieces of data of interest in our scenario are the financial facts and arcs. Blockchain is used as the back-end storage where each fact and arc are stored in separate transactions. Once transactions are validated (i.e., added to the blockchain), the data becomes available to the users of the network who can view them, and any updates can be traced.

D. Use cases

To illustrate the interaction between the different components, we have defined a set of use cases addressed by the proposed architecture. All scenarios assume that the user has a company registered in the system, the user then chooses an XBRL file and the evaluator shows four possible outcomes. Fig. 6 depicts a sequence diagram that covers most of the scenarios.

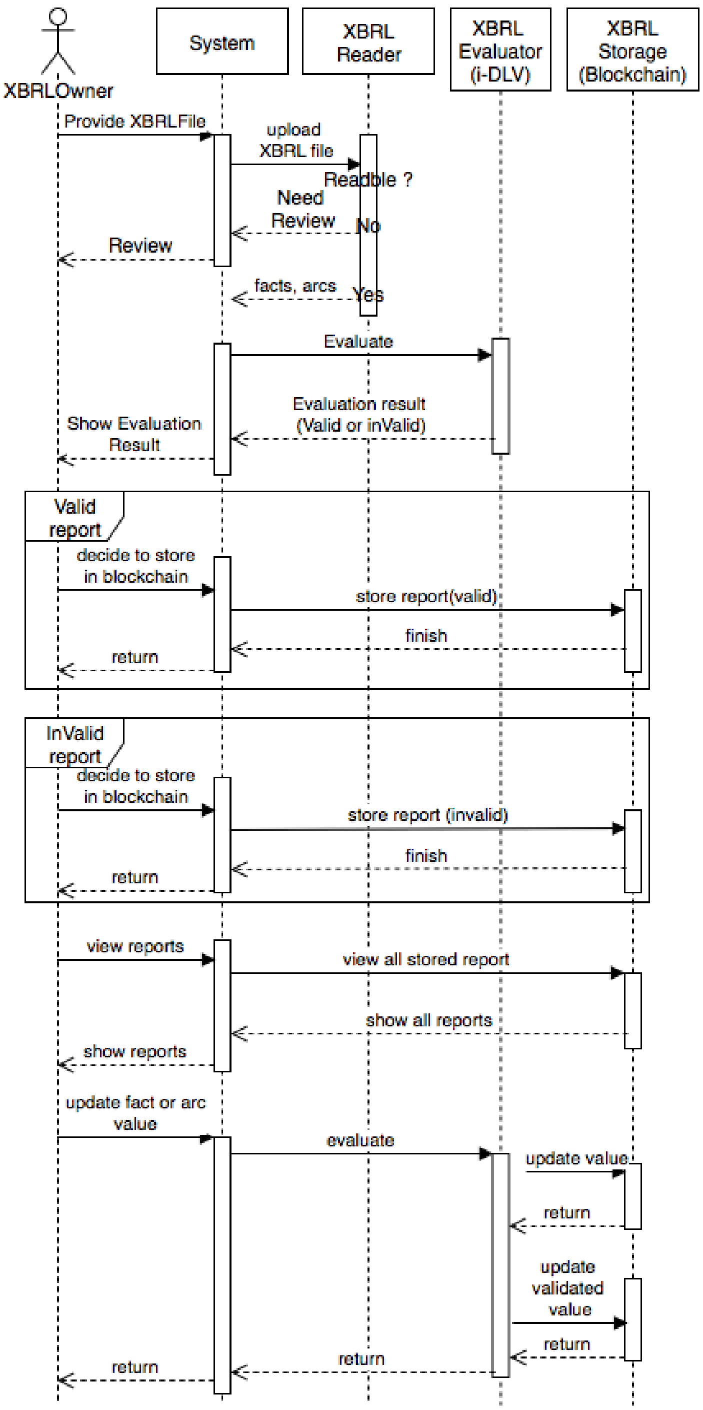

Figure 6. Report evaluation sequence diagram

- If all aspects are satisfied (valid), the user publishes the data into the blockchain.
- If one of evaluation criteria is violated, the user is advised to review the report and submit it later.
- If one of the evaluation criteria is violated, the user can still publish it into the blockchain but it will be flagged as invalid.
- Invalid reports already in the blockchain can be updated by their owners (e.g., update report values). The evaluator will check them again, if the updated report is accepted, the flag will change to valid. We note that if valid reports are updated with incorrect values they will be also flagged invalid.
- Other users or any third party organizations could view and evaluate any reports.

V. IMPLEMENTATION

The implementation of the proposed approach is conducted using a three layer architecture. Each of the layers is detailed in the following sub sections. The current implementation is a standalone application that interacts with the blockchain network. For the Etherum network, we rely on Blockchain network instance deployed at the University of Calabria, Italy called Unical coin [22] with the following configuration: (difficult:y ”0x90000”, gasLimit : ”0x2fefd8”, running nodes: 4). The full implementation of the proposed approach can be found in our Github repository [23].

A. XBRL Reader

XBRL Reader uses XBRLCore [24], a library to read and extract data. It receives as input an XBRL file and extracts all the relevant information for the validation process, which include both XBRL instances and XBRL taxonomies (arcs) according to the XBRL 2.1 Specification. XBRLCore also has it own validation but it does not fit to the newest taxonomy (for example with group of item). For example, facts: $RevenueTotal : 5000EUR$, and $CostOfSales : 3000, GrossProfit : 2000$ and arcs: $GrossProfit = RevenueTotal - CostOfSales$, could be presented as figure 7

```
fact(revenueTotal, "5000", eur).
fact(costOfSales, "3000", eur).
fact(grossProfit, "2000", eur).
arc(grossProfit, costOfSales, "-1").
arc(grossProfit, revenueTotal, "1").
```

Figure 7. Facts and Arcs example

B. XBRL Evaluator

XBRL Evaluator stores facts and arcs together with the queries in a query file to examine indices in the reports, also report where there is the error by i-DLV by calling from Java Runtime:

```
idlv xbrlFile.dlv calculation.py
```

xbrlFile.dlv includes the list of facts and arcs, queries (see figure 8), and calculation.py includes utility functions such as real numbers operations and list functions (see figure 9). After running the command above, it prints ”invalidDocument” if the data is not correct otherwise it prints ”validDocument”. The code computes the assets’ values by *i)* choosing each fact and its relation (arc) *ii)* multiple weight with asset value of each arc, and *iii)* sum these values to get expected asset value to compare with the actual value from fact. If they are not equal, $checkFact$ returns $false$ and $isValidDocument$ is also $false$, in other words, the document is not valid, otherwise, it is accepted.

```
chooseArc(F1, F2, V) :- fact(F2,V2,U), arc(F1,F2,W),
    &times(F1,V2,W;V).
invalidFact(F) :- chooseArc(F, _, V), &checkFact(F,V
    ;"False").
invalidDocument :- &checkDocument(;"False").
validDocument :- &checkDocument(;"True").
```

Figure 8. Query example

```
listFacts = {};
isValidDocument = True
def times(F, X, Y):
    fx = float(X)
    fy = float(Y)
    if F not in listFacts:
        listFacts[F] = 0
    listFacts[F] += fx * fy
    return str(fx * fy)
def checkFact(F,X):
    fx = float(X)
    if fx == listFacts[F] :
        return True
    else :
        isValidDocument = False
        return False
def checkDocument():
    return isValidDocument
```

Figure 9. Calulcation.py example

C. XBRL Storage

Financial data from evaluator are published into blockchain via web3js and built smart contract. Smart contract will make the skeleton to store data of a report, a company has many reports, each reports has its own facts and arcs (see figure 10).

```
struct Fact {
    string concept;
    string context;
    string value;
    string unit;
    string factgroup;
}
struct Arc {
    string conceptFrom;
    string conceptTo;
    string weight;
    string calLinkBase;
}
struct Report {
    string reportId;
    string date;
    string validated;
    Fact[] facts;
    Arc[] arcs;
}
struct Company {
    address companyAddress;
    string companyName;
    Report[] reports;
}
Company[] public companies;
```

Figure 10. Companies structure

Functions facilitate users to fill data into the structure (see figure 11).

```
function ownCompany();
function ownReport( reportId);
function registerNewCompany();
function getCompany();
function addReport(report);
function addFact(fact);
function addArc(arc);
function updateFact(fact);
function updateArc(arc);
```

Figure 11. Functions

VI. Evaluation

Two important aspects to evaluate when considering blockchain based solutions are cost and performance. Running computations onchain might result to be costly and impractical in many scenarios. The cost is associated with smart contracts execution and transactions recording and it is generally determined by two parameters: the amount of gas used by the execution of contract and the gasPrice associated with the transaction. The first one depends on the needed computation to perform the task, since every instruction executed by the Ethereum Virtual Machine has a certain gas cost. The second instead represents the cost in Ether of one gas unit, which depends on the blockchain network state when the transaction is performed. The general rule is that when a high number of transactions are pending, those with higher gasPrice have higher probability of being executed by a miner and be therefore added to the chain. In terms of performance since increasing the number of transactions increase the application latency.

A. Cost evaluation

We tested our system using 200 valid XBRL files, 22 invalid files (valid in calculation consistency) provided by different business providers and are annual financial reports. The tests consider all the implemented functions of the smart contract. These tests have been run on a test blockchain network and can be reproduced by calling a set of REST endpoints. Endpoint return the amount of gas consumed by while executing transactions. The amount of gas used is multiplied by the gasPrice to obtain the costs in Ether. The Ethereum to Euro conversion factor to these prices allows to compute the monetary cost. Table I presents the cost of executing the various contract functions.

TABLE I. Costs of smart contract functions execution

Function	*Ether cost (GWei)*	*Euro cost (€)*	*Avg Time (ms)*
registerNewCompany	0,00032	0,059	7022
addFact	0,01	1,83	7579
addArc	0,01	1,83	7579
addReport	0,0012	0,22	11705
updateFact	0,01	1,83	7579
updateArc	0,01	1,83	7579
updateValidatedValue	0.0012	0,22	12325

We note that on average an annual report contains around 129 facts and 122 arcs (251 transactions) which would cost approximately 2.5124 ETH (183 EUR at 9 November 2018 followed by [25]).

B. Performance evaluation

In terms of performance, we simulated the main scenario used in our approach, that is the process of publishing reports (addReport, addFact, addArc, updateValidatedValue). Figure 12 shows the average execution time for the whole process. The x axis represents the total number of facts and arcs as used in the process.

The results depicted show that the execution time is linear relative to the number of transactions. However, there are other factors that affect the execution time, mainly the variation of gas price which affects what transactions will be picked by the miners first and the size of the network (i.e., how fast the transactions are broad casted).

VII. Conclusion and Future Work

In this paper, we have presented the design and a prototype implementation of a blockchain based financial reports ledger. The main goal of the proposed approach is to increase trust

 ISBN: 978-1-61208-703-0

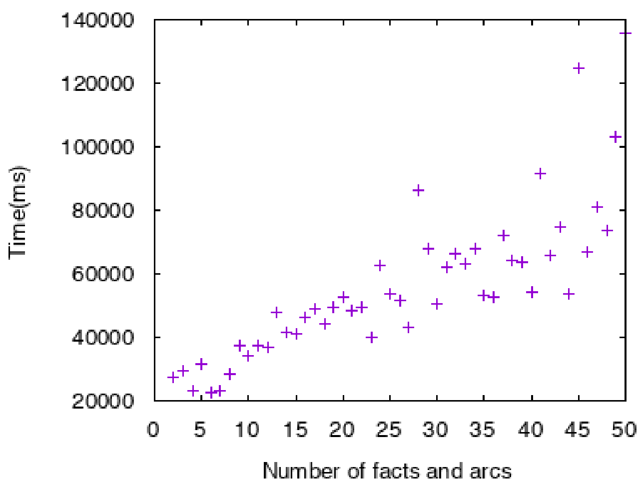

Figure 12. Main scenario average execution time.

and transparency in published financial reports, which can have great impact on inter-organizational transactions.

Using ASP in the first prototype makes flexible and easy to maintain. However in the next version, we will move all computations into be on-chain, and compressing technologies are also considered to reduce transaction weight, so the execution time can be reduced.

Although the study is limited to the Italian context and does not provide a cross analysis with other systems, the goal here is to shed some light on the great potential of using distributed ledger technologies in financial reports validation, storage and traceability. The proposed approach has been applied to the niche area of financial reports, but the same approach may have much wider applications in numerous contexts.

For future work, we are investigating the automatic correction of invalid XBRL documents such as typing mistakes and facts missing value. Moreover, financial statements should be based on the cash flow statements from organization to organization. When we have all data flow, we can provide end to end trustworthiness and reliability.

REFERENCES

[1] Investopedia, "Chamber of Commerce," 2018, URL: https://www.investopedia.com/terms/c/chamber-of-commerce.asp [accessed: 2019-04-10].

[2] XBRLOrganization, "An Introduction to XBRL," 2001, URL: https://www.xbrl.org/the-standard/what/an-introduction-to-xbrl/ [accessed: 2019-04-10].

[3] G. D'Atri, "Logic-based consistency checking of xrbl instances," IJACT, vol. 3–6, 2014, pp. 126–131.

[4] J.Zangari, "idlv," 2018, URL: https://github.com/DeMaCS-UNICAL/I-DLV/wiki [accessed: 2019-04-10].

[5] W. T. Adrian, M. Alviano, F. Calimeri, B. Cuteri, and e. a. Dodaro, "The asp system dlv: Advancements and applications," KI-Künstliche Intelligenz, 2018, pp. 1–3.

[6] C. Cachin, "Architecture of the hyperledger blockchain fabric," 2016, workshop on Distributed Cryptocurrencies and Consensus Ledgers.

[7] N. El Ioini and C. Pahl, A Review of Distributed Ledger Technologies: Confederated International Conferences: CoopIS, C&TC, and ODBASE 2018, Valletta, Malta, October 22-26, 2018, Proceedings, Part II, 10 2018, pp. 277–288.

[8] EthereumFoundation, "Ethereum," 2013, URL: https://www.ethereum.org/ [accessed: 2019-04-10].

[9] C. Pahl, N. El Ioini, S. Helmer, and B. Lee, "An architecture pattern for trusted orchestration in iot edge clouds," in Fog and Mobile Edge Computing (FMEC), 2018 Third International Conference on. IEEE, 2018, pp. 63–70.

[10] C. Pahl, N. EL Ioini, and S. Helmer, "A Decision Framework for Blockchain Platforms for IoT and Edge Computing," Proceedings of the 3rd International Conference on Internet of Things, Big Data and Security, no. IoTBDS, 2018, pp. 105–113.

[11] O. Bussmann, "The future of finance: Fintech, tech disruption, and orchestrating innovation," in Equity Markets in Transition. Springer, 2017, pp. 473–486.

[12] M. Mettler, "Blockchain technology in healthcare: The revolution starts here," in e-Health Networking, Applications and Services (Healthcom), 2016 IEEE 18th International Conference on. IEEE, 2016, pp. 1–3.

[13] H. Byström, "Blockchains, real-time accounting and the future of credit risk modeling," Lund University, Department of Economics, 2016.

[14] J. Dai and M. A. Vasarhelyi, "Toward blockchain-based accounting and assurance," Journal of Information Systems, vol. 31, no. 3, 2017, pp. 5–21.

[15] T. D. Colgren, "Xbrl, blockchain, and new technologies," Strategic Finance, vol. 99, no. 7, 2018, pp. 62–63.

[16] Y. Guo and C. Liang, "Blockchain application and outlook in the banking industry," Financial Innovation, vol. 2, no. 1, 2016, p. 24. [Online]. Available: https://doi.org/10.1186/s40854-016-0034-9

[17] S. Helmer, M. Roggia, N. El Ioini, and C. Pahl, "Ethernitydb – integrating database functionality into a blockchain," 09 2018, pp. 37–44, aDBIS.

[18] H. Fischer and D. Mueller, "Open source & xbrl: the arelle® project," in 2011 Kansas University XBRL Conference, 2011, pp. 29–30.

[19] Altova, "XBRL Development Tools," 2018, URL: https://www.altova.com/xbrl-tools [accessed: 2019-04-10].

[20] W. contributors, "Benford's law — Wikipedia, The Free Encyclopedia," 2019, URL: https://en.wikipedia.org/w/index.php?title=Benford%27s_law&oldid=885125880 [accessed: 2019-04-10].

[21] L. Pietronero, E. Tosatti, V. Tosatti, and A. Vespignani, "Explaining the uneven distribution of numbers in nature: the laws of benford and zipf," Physica A: Statistical Mechanics and its Applications, vol. 293, 2001, pp. 297–304.

[22] UnicalCoinTeam, "Unicalcoin," 2017, URL: https://github.com/marcuzzu/UnicalCoin [accessed: 2019-04-10].

[23] V. T. Le, "Trustable system for XBRL," 2001, URL: https://github.com/levanthanh3005/TrustableSystem-for-XBRL [accessed: 2019-04-10].

[24] Y.seki, "XBRL Core," 2006, URL: https://sourceforge.net/projects/xbrlcore/ [accessed: 2019-04-10].

[25] S. ltd, "Currencio — Cryptocurrency Converter," 2001, URL: https://currencio.co [accessed: 2019-04-10].