Enabling Financial Reports Transparency and Trustworthiness using Blockchain Technology

Van Thanh Le, Claus Pahl, Nabil El Ioini,

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

Gianfranco D'Atri

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

***Abstract*—Financial report activity standardization becomes more essential in Financial Services and Technology, to facilitate the digitization of the process of communicating and acquiring business information. The eXtensible Business Reporting Language (XBRL) is one of the first steps towards this vision by providing a general digital format of financial reports with different rules and tags. Analyzing XBRL reports allow us to verify the quality and transparency of the data and well as have the full history of the stored transactions. Currently, checking and storing reports are independent for each organization and country, which is less transparent for the public and investors, who might be interested in checking company's records before investing in them. 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 transparency and trustworthy. Specifically, we design and implement a prototype to evaluate and store financial statements using Ethereum blockchain following special metrics. Moreover, we examine different architectural decisions in terms of cost and performance and look at their advantages and disadvantages.**

Keywords–Blockchain; XBRL; Financial Reports; DLV; ASP; On-Chain Computation.

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 of the economic situation of a company and its ability to attract investors.

The precision of financial reports can not be underestimated, any missing numbers or assets in a balance sheet could have a tremendous effect on a business, for instance, a company actually lose their profit because they miss their tax. The accuracy also supports to find mistakes of expenses or internal process early, monthly reports can show the problem to restructure procedures or wrong activities. Moreover, by proposing a report in detail and correctness, businesses get more trust from the community and attract more investors. In a normal balance sheet, there are more than 100 fields needing to be filled that will be a challenge for accountants if they do in the traditional way, so we need a special tool to support the process that will check any missing point, or in other words, to verify the validity of a report, as a suggestion for company managers.

Financial reports contain sensitive data that refer to the company status, following the timeline, the data even affects decision making for future work. For example, a manager wants to examine the importance of an asset to decide whether he will buy more, in case the asset is frequently deprecated every year, his decision will change. However, the owner of the assets could update its historical price in the database to hide the downward trend that makes the storage becomes unreliable. Even the distributed database is stable, high availability and performance but it is still under controlled by authorities or any third parties, thus, in the case, blockchain could bring benefits for this model.

To this end, our goal is to investigate how blockchain can be used to address these limitations to restore trustworthiness in the financial reports. Our contribution is two-fold (i) provide methodologies to automatically evaluate and validate the consistency of the generated reports in case of off-chain and on-chain, *(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 and time factor of the proposed approach.

This paper develops on our previous work [1] by providing i) An off-chain evaluator based iDLV ii) a trustworthy financial report storage that is a baseline for our continuing work.

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

II. REFERENCE SCENARIO AND PROBLEM STATEMENT

A. Scenario

Our reference scenario focuses on the Italian legislation. The financial statements are governed in Italy by Articles 2423. Following the Italian Civil Code, submitting financial reports is mandatory, and it needs to be done through the website of Chamber of Commercial (*webtelemaco.infocamere.it*). Every enterprise must prepare financial statements for each year or two consecutive financial years. These essential reports contain *i)* Balance Sheet, *ii)* Income Statement, *iii)* Cash Flow

Statement, and *iv)* Explanatory note. Depending on the size of the business, we can have different forms of submissions. For example, micro-enterprises are able to skip cash flow statements and explanatory notes. In our scope, we mainly focus on the balance sheet, since it is the most common statement for every company to submit.

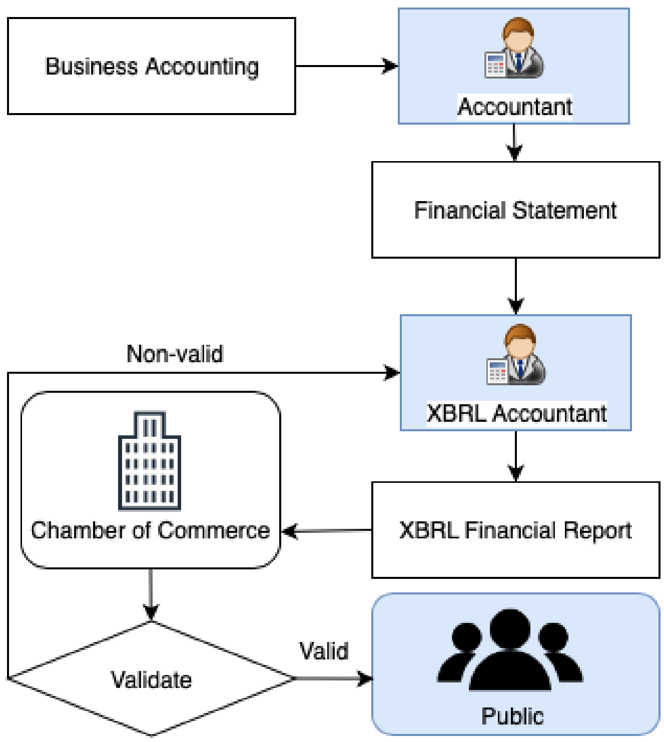

Figure 1. Financial data flow

The use case we refer to is described in Figure 1, main actors and components are:

- Business accounting is the systematic recording, analyzing, interpreting and presenting of financial information.
- An accountant is a person who records business transactions on behalf of an organization, reports on company performance to management, and issues financial statements.
- Financial statements are a collection of reports about an organization's financial results, financial condition, and cash flows.
- XBRL Accountant is a person who works with financial statements and XBRL tools to fill data to XBRL financial reports.
- XBRL is the format for delivering financial reports in an interactive digital format.
- Validating XBRL is the way we check the consistency of XBRL asset values, for example, in Italy, the validator is TEBENI, the tool provided by the Italian Chambers of Commerce.

A company *A* wants to make a financial report for this year, the manager assigns the task to the Chief Accounting Officer (CAO). The CAO then collects business accounting information from accountants as account payable, receivable, expense for wages, etc. After that, a draft version of the financial statement is made, and start preparing for XBRL version, this is a standard digital format for financial reports (XBRL stands for eXtensible Business Reporting Language). The draft is sent to a specialist called XBRL accountant. The specialist will find a suitable taxonomy following their country and business status, for example, the latest XBRL taxonomy version is *PCI_2018-11-04*, followed by [2]. The staff will map and tag financial statement elements to corresponding elements in the chosen taxonomy and use tools like Microsoft Excel or up-to-date tools in [3] to fill the data to make the instance document.

After getting an XBRL report, internal validation is necessary as a pre-check. The validation contains two steps [4]: *i)* validation of the markup and *ii)* validation of calculations. Enhanced Validation and Strict XBRL Validation are two kinds of validating calculations, in Enhanced Validation, it checks in detail child elements of a parent element, if one of them misses or the summary of calculation does not fit for both parent and child elements, an error will be released. Strict one accepts some missing elements. Because there are few companies can fulfill entire fields in taxonomy so, in our investigation, we accept some missing tag like the Strict Validation.

When the validation process finished, the CAO and the company director board will review and give permission to send the XBRL report to the Chambers of Commerce [5]). This is an association or network of entrepreneurs designed to promote and maintain the benefits of its members. The office is also considered as a board of trade that includes groups of businessmen sharing their interests even in the international scope. A chairman will be chosen to negotiate and debate with the government for policies in financial aspects and overall economic environment. In the Chambers of Commerce, there is a brief check for submitted reports before publishing it into the website of the office for the public (e.g *http://www.registroimprese.it/*). In order to access the database on the website, in Italy, it costs around 3 EUR for a report and data researching companies retrieve these data for their analysis and audit.

Independent auditors will examine financial reports after publishing on behalf of investors or customers, they give a composed report containing their opinion about whether the financial statement is fairly stated and comply in all material respects.

In our view, we will follow the scenario to the work of Chamber Of Commerce, auditors can access our public database to investigate, even can refer our evaluation strategies.

B. Problem statement

We are focusing on the processes when XBRL files are created. Evaluating files is a complicated endeavor since it requires the validation of many steps by the local authority. In case the files do not pass the validation process, they need to be corrected, then the whole process needs to be executed again. A standard evaluation system is needed for both companies and governments.

Moreover, the traditional process is not transparent even when the XBRL files are published in a public database. The files can still be updated after years to cover mistakes from the accountants, thus, the database should not be controlled by companies and all changes need be traced.

III. 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 the 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 [6], which is the world-leading standard for financial reporting. It facilitates inter-organization 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 2).

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

Figure 2. Facts example

Each fact has the following components:
Concept name: contains namespace prefix of taxonomy schema like rp:CostOfSales.
Id: defines the unique fact but its optional.
Context Reference: shows the context of fact like working year or location.
Unit refers to unit information of the fact value
Fact value: presents for the value of an asset

2) XBRL taxonomy, a collection of arcs, which define metadata about these facts and their relationship with other facts (see Figure 3 as example of calculation linkbase).

```
<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 3. XBRL Linkbase example

Totally, a taxonomy schema has five types of linkbase:
Label linkbase: provides human-readable strings for concepts, multiple languages are also supported here for each language.
Reference Linkbase: is intended to contain relationships between concepts and references to authoritative statements.
Calculation Linkbase: contains mathematical relationships among numeric items.
Definition Linkbase: associates concepts with other concepts using a variety of arc roles to express relations between concepts in taxonomies
Presentation Linkbase: contains hierarchical presentation relationships among concepts

Figure 4 depicts XBRL structure and the relations between the different components.

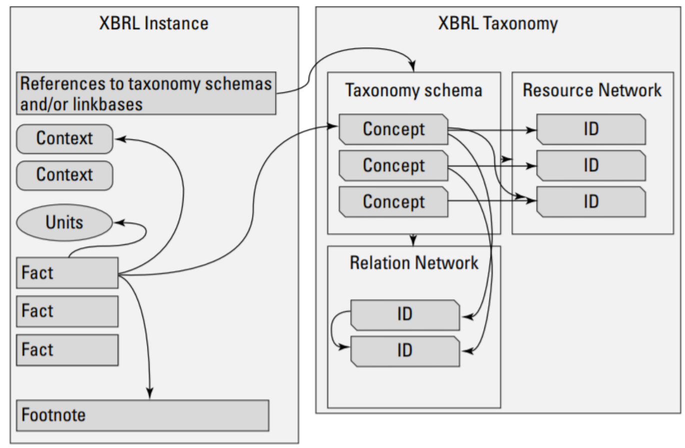

Figure 4. XBRL Structure

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, several tools have been developed to automate the validation process, Answer Set Programming (ASP) [7] is a form of declarative programming oriented towards difficult search problems, highly used in both academia and industry. ASP programs consist of rules by the form:

$$< head >: - < body > .$$

The rules are called facts, the symbol $:-$ means if, if the body is true, the head will exist. $< body >$ includes $b1 \vee b2 \vee .. \vee b3$, and $< head >$ are $h1 \wedge h2 \wedge \ldots \wedge h3$, so when all of b value is true, one of h will be chosen to do other computations.

The possible use of an ASP language for analyzing XBRL financial reports was explored by Gianfranco d'Atri in [8]. The tokenization and standardization of data supported by the XBRL Consortium allow extensive and meaningful use of AI techniques to support economic analysis and fraud detection.

I-DLV [9] is a new intelligent grounder of the logic-based Artificial Intelligence system DLV [10], 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 technology

1) Distributed Ledge Technology: Distributed Ledger Technology (DLT) is an innovative data structure relying on a decentralized and distributed peer-to-peer network to exchange data and a consensus protocol to keep data consistent. The main goal of DLT is to solve the double spending problem and provide a single source of truth with no trustworthy or centralized authority.

2) Blockchain: Blockchain [11] is a distributed ledger technology, built out of a linked list of boxes called blocks, linked together by hash codes. Each block references the previous block by including its has in its header. The blocks contain transactions that represent the information managed by the network (e.g., financial transactions, identity transactions ...). The Bitcoin blockchain is considered to be the first blockchain implementation, however, today, there more than a hundred implementation with different flavors and target different domains.

The main building blocks of a blockchain are [12]:

- Transactions, which are signed pieces of information created by the participating nodes in the network and then broadcast to the rest of the network. Every transaction must be verified by nodes before recorded into a public ledger, the verification node needs to ensure that the spenders own the crypto-currency via the digital signature, and has sufficient amount of currency in their account.
- 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 (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 [13]. 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 participants could check business working status.

3) Ethereum: Ethereum [14] is a general purpose blockchain platform that enables the deployment of distributed applications. The main feature of Ethereum is the introduction of smart contracts, which are computer code residing in the blockchain and which gets executed once certain conditions are met. Smart contracts enable the development of decentralized autonomous organization (DAO), that uses smart contracts as functions to enforce governance mechanisms.

Since Ethereum is a public permissionless platform, it relies on mining to generate the next blocks of the chain. The transactions need to pay a fee in Gas [15], which is a unit that measures how much work of a node for an action or a task.

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 trustworthy publishing and updating.

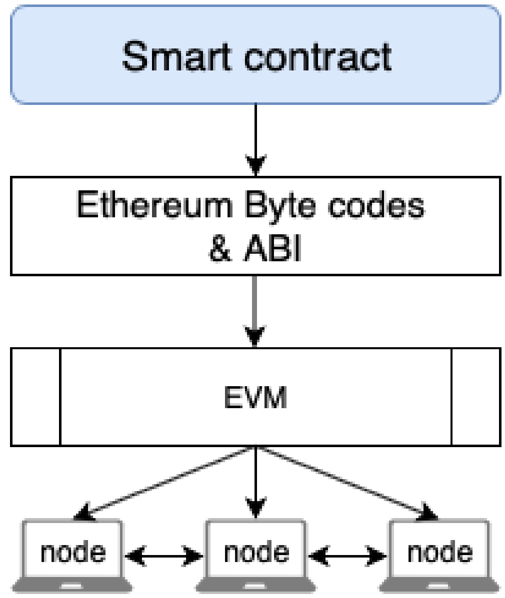

Figure 5. Smart contract execution

Figure 6 depicts an overview of the proposed architectures. It is divided into three main components: XBRL Reader, XBRL Evaluator, and XBRL Storage but in different approaches with blockchain integration that guarantees various levels of traceability.

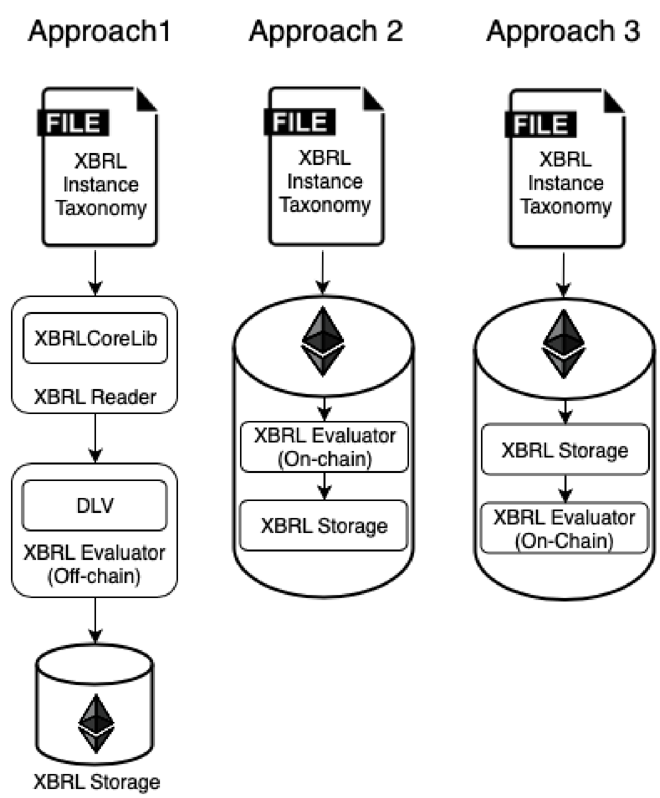

Figure 6. Alternative architectural designs

A. Components

We will start with each main component to understand its roles in approaches.

1) 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.

2) XBRL Evaluator: Facts and arcs from the first step are evaluated in the module, only in the first approach (see Figure 6 - Approach 1), the part is separated with DLV solver, in the others, the Evaluator is injected inside blockchain as a function. We define the needed aspects to investigate in a financial report here:

- 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 pay for the warehouse cost.
- Benford's law checking: Benford's law [16] 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 7):
$P(d) = \log_{10}(d+1) - \log_{10}(d) = \log_{10}(\frac{d+1}{d})$
The complicated formula is explained in [17] about stock prices example with distributions. Benford's law could check the whole data set or each financial report.

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.

3) 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 backend 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	$P(d)$	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 7. Benford's law for the first digit [16]

B. Structural design

We propose an architecture that relies on two main perspectives i) off-chain for a standalone computation to analyze financial data ii) on-chain for enabling public execution in blockchain for possible functions. We consider three typologies (see Figure 6) demonstrating possible models with pros and cons for our financial data controller.

1) Approach 1: the financial reports are read by the XBRL reader, where all facts and arcs are extracted, then the Evaluator runs iDLV solver to analyze them and generate the results. The report owners can review the results before publishing them in the blockchain. Afterward, the owner can still update the inserted reports, and customers or the public can track the changes via blockchain logs or historical data.

- Advantages: the off-chain evaluator reduces validation time, especially when we are dealing with big data analysis with many financial reports at once.
- Disadvantages: The evaluation is executed as a black box and only the results are published. All the calculations are hidden, which might affect the level of trust from the public about the results.

2) Approach 2: facts and arcs also are results from the Reader, and directly they are evaluated with on-chain functions that are composed in a smart contract, after finishing the computation, if the owner is satisfied with the results, they can push it into blockchain. Any changes are accepted but still under the evaluation and can backtrack.

- Advantages: the Evaluator is implemented on the blockchain, therefore, everyone can verify its logic. To update the evaluation function, we only extend the current smart contract.
- Disadvantages: Sending a whole financial report (around 2MB) at once to a view function requires more time and it might cause the local node to crash

(e.g., the view function contains an infinite loop), which leads eventually to out_of_gas error.

Because of the negatives, approach 2 will not be implemented for testing.

3) Approach 3: this approach reverses the data storage and evaluation phases in comparison to approach 2. This allows data to be stored in advance without affecting the evaluation time.

- Advantages: data can be stored on-chain much prior to the evaluation process. Therefore, the evaluator can access the data without having to wait for it to load.
- Disadvantages: The computation runs on local nodes for evaluation that makes the performance becomes so slow for the whole financial report.

Generally, the positive and negative aspects of each approach are showed in Table I.

TABLE I. APPROACH QUALIFIES

Strategies	*Performance*	*Cost*	*Trust*	*Feasible ?*
Approach 1	high	high	low	yes
Approach 2	low	low	high	no
Approach 3	low	low	high	yes

C. Working 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. 8 depicts a sequence diagram that covers most of the scenarios.

- If all aspects are satisfied (valid), the user publishes the data into the blockchain.
- If one of the 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 subsections. 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 [18] with the following configuration in Table II.

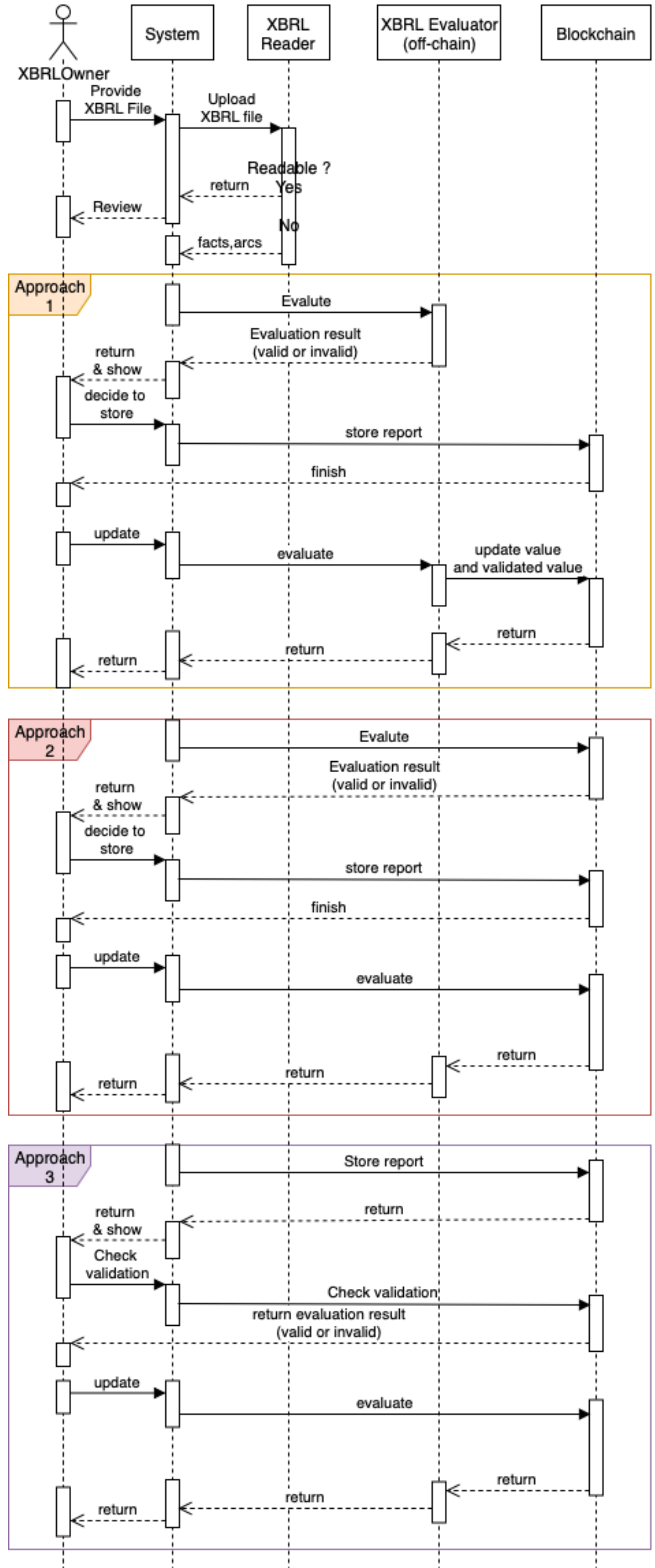

Figure 8. Report evaluation sequence diagram

The full implementation of the proposed approach can be found in our Github repository [19]. The two chosen approaches share XBRL Reader and XBRL Storage as a common part that we will describe in advance, only the evaluator for each solution are different from process and input.

TABLE II. NETWORK CONFIGURATION

Properties	*Value*
Original network	Ethereum
Difficulty	0x90000
Gas limit	0x2fefd8
Running nodes	4
Network speed	53 Mbps in download and upload

A. XBRL Reader

XBRL Reader uses XBRLCore [20], 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 9.

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

Figure 9. Facts and Arcs example

B. XBRL Storage

Financial data from the evaluator are published into blockchain via web3js and built smart contract. web3.js [21] is the Ethereum compatible JavaScript API, which implements the Generic JSON RPC specification, which is a collection of libraries, which allow you to interact with a local or remote Ethereum node, using an HTTP or IPC connection. Smart contract will make the skeleton to store data of a report, a company has many reports, each report has its own facts and arcs (see Figure 10).

Functions facilitate users to fill data into the structure (see Figure 11). As explained about the gas limitation above, adding each fact and arc one by one will reduce the burden on the network.

C. XBRL Evaluator

1) Approach 1: 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 12), and calculation.py includes utility functions such as real numbers operations and list functions (see Figure 13). 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. With queries, we can verify one or many documents at once with all defined metrics.

```
struct Fact {
    string concept;
    string context;
    int value;
    string unit;
    string factgroup;
}
struct Arc {
    string conceptFrom;
    string conceptTo;
    int 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

```
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. Storage functions

```
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 12. Query example

2) Approach 3: XBRL Evaluator will integrate with the XBRL Storage as a view function in blockchain (see Figure 14). View functions do not cost any fee in execution but the amount of gas is still calculated and is still limited in one block, this is also a method of Ethereum to prevent infinite loop. To evaluate a financial report, we need to scan the fact and arc list with two for and four if that is cumbersome for a local node, therefore, we break one big function into two smaller function calls. $calActualFactValue$ finds related arcs and the check it with $checkArc$ before returning each chosen fact actual value, and then we can compare it with the real

```
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 13. Calulcation.py example

value in the dataset.

The result of the evaluation is not necessarily published into the blockchain since anyone can verify via their local nodes. Thus we can save more energy and Ether cost for the transaction of validated values.

```
function checkArc(
    uint companyIndex,
    uint reportIndex,
    uint findex,
    uint aindex) public view returns (
        int rs
    ) {
}
function calActualFactValue(
    uint companyIndex,
    uint reportIndex,
    uint findex) public view returns (
        int realvalue, int rs
    ) {
}
```

Figure 14. On chain evaluator functions

VI. EVALUATION

A. System configuration

We demonstrate the testing environment:

- Processing core: Intel Core i7-4710HQ (2.50 GHz, 6 MB L3 Cache).
- Chipset: Mobile Intel HM86 Express Chipset.
- RAM: 8.00 GB.
- Operating system: Ubuntu 16.04.4 LTS
- Hard Disk Drive: Solid State Drive (SSD) 1 TB Toshiba
- Application: Java web project maven eclipse

Two important aspects to evaluate when considering blockchain based solutions are cost and performance, we also evaluate defined financial metrics to review the accuracy of our methods.

B. 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 calculation is performed by the XBRL Evaluator, which implements the required mechanisms to check the soundness and completeness of the given files. 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 returns the amount of gas consumed 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 computing the monetary cost. Table IV presents the cost of executing the various contract functions.

TABLE III. 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 61 arcs (192 transactions) which would cost approximately 0.74 ETH (118.86 EUR at 28 August 2019 followed by [22]).

C. Time evaluation

In terms of time execution for XBRL Storage, we simulated the scenario used in our approaches, that is the process of publishing reports (addReport, addFact, addArc, updateValidatedValue). Figure 15 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 for storage is linear relative to the number of transactions. However, other factors 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 broadcasted).

Considering the time execution in evaluation, each strategy performs differences as shown in Figure 16. The off-chain one uses iDLV solver so the time is much faster than the other, the maximum time evaluation in the case is only around 300 ms in contrast of on-chain calculator, that is more than 30000 ms (30 seconds) and the value will increase following the growth of a number of facts and arcs. In a real report with around 129 facts and 61 arcs (192 transactions), blockchain takes around 17031ms (17 seconds) while it is only 5516ms (5 seconds) in iDLV.

When executing transactions, we set the gas price as default at 20 gwei, but following [23], execution time could change

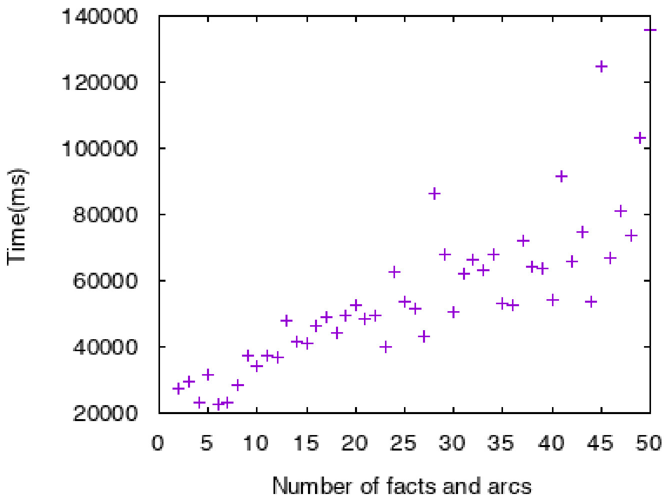

Figure 15. Storage execution time.

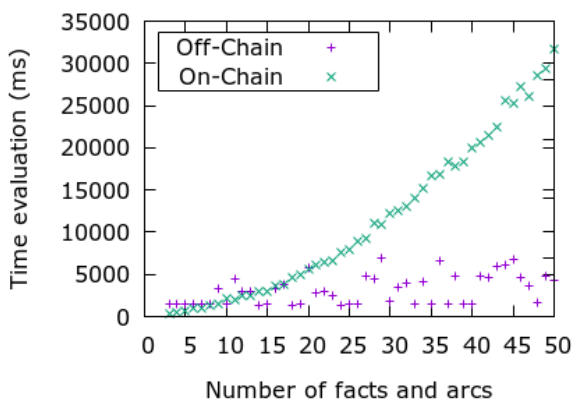

Figure 16. Evaluation time.

based on the gas, the suitable gas price is 43 gwei. We are using the private Ethereum chain so doing experiences with gas price changes are not feasible in the environment, and we will leave the question for future work when we can test the price in the real network.

D. Metric evaluation

As discussed about the four possible metrics we need to investigate, our dataset contains 120 financial reports in XBRL format, in the context of 2000 and 2015. Following the service provider, in these reports, there are:

- 100 valid files: the reports are accepted in any aspect to publish to a public database.
- 20 invalid files: the 10 reports do not meet the calculation consistency requirement and the other 10 reports miss important fields.

With the data set, we tested our algorithms that result in:

- Calculation consistency: the consistency is calculated by comparing the expected value with its actual value of all facts in reports. As the example above, the expected value is determined by: $ExpectedValue = \sum subFactValue * arcWeight$ Since building financial report is complicated, that is cumbersome to correct all values, so we accept the fact if the difference between its expected and actual value is smaller than a threshold, we set a default threshold in our model is 0.5, and the difference is calculated by: $valueDifference = \frac{expectedValue - actualValue}{actualValue}$

  With the configuration, we have 83 per 120 files are correct, that means the service provider decision and our outcomes are similar for each file. The other 37 files have differences in qualifying since our strategy does not check some special essential fields that need advice from experts, moreover, they also have a middle man to fix unaware mistake to make sure the reports are clean before publication.
- The rate of debt and interest: we compared the differences between the rates from two consecutive years and a threshold that is fixed by 0.5, and all of the reports have the higher values, thus the rate threshold needs to be concerned to re-defined.
- Financial aspect comparison: by retrieving all values from our reports, we have a big database, to be explicit, in the context of 2016, there are more than 4000 financial items that need to be reviewed. *itcc_ci_totalecreditiversosociversamentiancoradovuti* is the total receivables from shareholders for payments must be shown to highlight both the portion that has already been called by the company, and the portion still to be called, we choose the item as an example, and the result showed the item is missed in 88 files, is zero in 23 files and has values in only 14 files.
- Benford's law: we checked all first digits in reports, choose two big groups in each report as $conto\_economico$ and $stato\_patrimoniale$, and also review the law suitability in numbers of each report. The law suitable comparison is made by $scarto = \sqrt{\frac{\sum_{i=1}^{9}(P_i - B_i)^2}{9}}$. The $scarto$ will be calculated in Table IV following each first digits from 9 to 1, the negative and zero value are also countable, the table also presents the differences between the actual percentage in the dataset and the benford's law.

TABLE IV. Costs of smart contract functions execution.

Number	*Count*	*Percentage* (P_i)	*Benford's law*(B_j)	Differences
9	750	4.72	4.6	0.12
8	806	5.08	5.1	-0.02
7	873	5.5	5.8	-0.3
6	985	6.21	6.7	-0.49
5	1303	8.21	7.9	0.31
4	1469	9.25	9.7	-0.45
3	2001	12.61	12.5	0.11
2	2820	17.76	17.6	0.16
1	4867	30.66	30.1	0.56
Negative	1731			
Zero	4945			
			Scarto	0.33

For all financial documents, the $scarto$ value is 0.33 as Table IV, but for the two groups, the $scarto$ of $conto\_economico$ is 0.96 and $stato\_patrimoniale$ is 1.33 that are compared with the stardard law in Figure 17.

We qualified Benford's law in each report, the result is

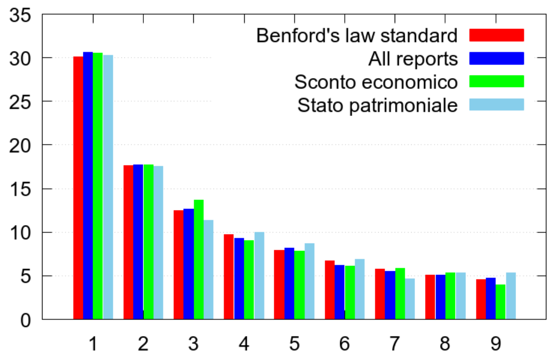

Figure 17. Compare first digits of groups with Benford's law

even worse when the maximum *scarto* for one report is 1193.66 and the minimum one is 45.64.

The result from the larger amount of data gets closer to the law, even with the whole dataset, it is only 0.33 in differences but get higher with smaller dataset, we can experience that the Benford's law could check the data of a company in case the financial report is exported every month, and the checking can run after several years.

VII. 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 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 [24] [25], finance [26], health care [27], smart mobility [28] and others. In the literature, a number of studies considered the implication of blockchain on financial services and accounting. Byström [29] 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 [30], the authors discuss how blockchain can be an enabler technology for accounting ecosystem auditing and transparency. In [31], Colgren discusses the advantages that blockchain can bring to companies by allowing fast and public access to companies financial statements. In [26], Bussmann has given a more general overview of the potential disruption of blockchain on the Fintech market.

For banking services, Ye Guo [32] suggests that blockchain is able to 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. Q.K.Nguyen [33] indicated that blockchain could potentially reshape the economy, but the banking industry requires high speed of transaction execution with high scalability that is still a limitation of blockchain. VAT Fraud detection and prevention is a recommendation from [34] that uses access keys to get the authorization when purchasing cross border products. In [35], Reverse factoring and dynamic discounting are two approaches that could get benefits from blockchain, which becomes a financier to guarantee the future payment or holding the cash from buyers to optimize the working capital.

Applying blockchain as storage, Sven Helmer et al. built MongoDB database functions into Ethereum in [36], that separates the driver and database to reduce the cost transactions. The main goal of their approach is to keep all data on-chain. Shafagh et al. in [37] proposes a solution for blockchain auditable storage of IoT data by two layers of control plane and data plane, in the architecture, we can define access rights based on blockchain and control data from distributed off-chain storage. A searching function is applied with distributed hash table (DHT), nevertheless their strategy is not implemented yet and they also did not specify, which tools are used. Another approaches for public storage, [38] storj extends the function to enable people to rent other free storage and bandwidth, based on blockchain framework, it uses encryption, file sharing to store files on a peer-to-peer network. IPFS [39] stands for Interplanetary File System, it works on HTTP protocol and uses DHT to store the data following content-addressing technique that set a permanent link for any uploaded content, content-addressing allows us to verify the data too and any users in the network can access the content by its address. The system now is still unstable and has a lot of bugs, access content from other computers still take more time than traditional storage, we can expect IPFS future release with more stable versions.

We found few articles discussing about on-chain computation, Xu et al via [40] showed the comparison of on and off chain solutions for data storage, computation, even based on different blockchain models. Keeping data inside blockchain is more expensive but this is a one-time cost for permanent storage. Current blockchain technologies as Ethereum and Bitcoin are suitable with simple computation, Digital Asset Modeling Language (DAML) [41] is a new coming solution for complex computing, this is a programming language for financial institutions. DAML is designed to solve the issues of the agreement without revealing its content, the model is optimized for a private environment and the current version can interact with Hyperledger [42]. Our purpose is to make a public and trust network for the community, so DAML is not reasonable for our adaption.

Considering big data in blockchain, there are many promise use case presented in [43] about blockchain application for big data like IoT or personal data area, Karafiloski et al. showed current application as examples to extend for data analysis. L. Yue et al. in [44] proposed a big data blockchain model with several levels of data access and collection right, the smart contract in the model will automatically encapsulate predefined states and conversion rules, trigger executions. Mystiko [45] is quite comprehensive to build a whole blockchain network from scratch with up-to-date and high level technologies like cassandra for storage, kafka for communication. The high scalability, availability and elastic search enable it to be big data friendly. However, the project is implemented for a private chain that is suitable for multi party solutions, while we need a public chain for high level of trust and public verification, thus, applied strategies in Mystiko could be considered for our next extended version.

In terms of tools related to XBRL, several tools are in use, however, they are not able to guarantee the long term trustworthiness of the reports on produced. With regard to the analysis of financial reports in XBRL format, Arelle [46] 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 [47] 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 concepts 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.

VIII. 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 approaches is to increase trust and transparency in published financial reports, which can have a great impact on inter-organizational transactions. From the side of authorities, they do not have to wait for the long process of validation or storing reports, companies will submit and verify it via the system automatically, and for investors or any users want to check reports or changes in facts, they only follow logs to reveal differences.

Using ASP as a core computation in the first approach makes flexible and easy to maintain but get less trust since the public can not read clearly the full code after deployment, another strategy with on-chain computation get higher trust but take more time in evaluation (around 100 times in comparison with the off-chain version), the trade-off seems more acceptable with the public. Combination the two methods could be a solution in our future work when the companies can review and fix mistakes with a draft of report evaluation by ASP and the customers only need the on-chain function to check the validity without off-chain solver investigation.

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.

Additionally, since we are currently working in the context of smart mobility infrastructure in the 5G CARMEN project [48], which focuses on crossing organizational boundaries use cases. Services such as cross border insurances and multimedia content access require trustworthy collaboration among different countries to generate and manage payments. This has a great effect on the cash flow and therefore, financial reports of international companies. Our work in this paper can contribute to trust management by providing a standard and trustworthy mechanism for financial transaction validation.

References

[1] G. D'Atri, V. T. Le, C. Pahl, and N. El Ioini, "Towards trustworthy financial reports using blockchain," in Proceedings of The Tenth International Conference on Cloud Computing, GRIDs, and Virtualization (CLOUD COMPUTING 2019), Venice, Italy, 2019, pp. 37–42, ISBN: 978-1-61208-703-0.

[2] Agenzia per l'Italia digitale, "XBRL - Standard formato elettronico editabile per la presentazione dei bilanci," 2019, URL: https://www.agid.gov.it/it/dati/formati-aperti/xbrl-standard-formato-elettronico-editabile [accessed: 2019-12-04].

[3] XBRL International, "Tools and Services," 2019, URL: https://www.xbrl.org/the-standard/how/tools-and-services/ [accessed: 2019-12-04].

[4] M. E. Phillips, T. E. Bahmanziari, and R. G. Colvard, "Six Steps to XBRL," 2008, URL: https://www.journalofaccountancy.com/issues/2008/feb/sixstepstoxbrl.html [accessed: 2019-12-04].

[5] Investopedia, "Chamber of Commerce," 2018, URL: https://www.investopedia.com/terms/c/chamber-of-commerce.asp [accessed: 2019-12-04].

[6] XBRL Organization, "An Introduction to XBRL," 2001, URL: https://www.xbrl.org/the-standard/what/an-introduction-to-xbrl/ [accessed: 2019-12-04].

[7] Wikipedia contributor, "Answer set programming," URL: https://en.wikipedia.org/w/index.php?title=Answer_set_programming&oldid=898338706 [accessed: 2019-12-04].

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

[9] J. Zangari, "idlv," 2018, URL: https://github.com/DeMaCS-UNICAL/I-DLV/wiki [accessed: 2019-12-04].

[10] W. T. Adrian, M. Alviano, F. Calimeri, B. Cuteri et al., "The asp system dlv: Advancements and applications," KI - Künstliche Intelligenz, vol. 32, no. 2, 2018, pp. 177–179. [Online]. Available: https://doi.org/10.1007/s13218-018-0533-0

[11] M. Crosby, P. Pattanayak, S. Verma, and V. Kalyanaraman, "Blockchain technology: Beyond bitcoin," Applied Innovation, vol. 2, no. 6-10, 2016, p. 71.

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

[13] N. El Ioini and C. Pahl, "A review of distributed ledger technologies," in On the Move to Meaningful Internet Systems. OTM 2018 Conferences. Springer International Publishing, 2018, pp. 277–288, ISBN: 978-3-030-02671-4.

[14] Ethereum Foundation, "Ethereum," 2013, URL: https://www.ethereum.org/ [accessed: 2019-12-04].

[15] G. Wood, "Ethereum yellow paper," 2014, URL: https://github.com/ethereum/yellowpaper [accessed: 2019-12-04].

[16] Wikipedia contributors, "Benford's law — Wikipedia, The Free Encyclopedia," 2019, URL: https://en.wikipedia.org/w/index.php?title=Benford%27s_law&oldid=912469307 [accessed: 2019-12-04].

[17] 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.

[18] Unical Coin Team, "Unicalcoin," 2017, URL: https://github.com/marcuzzu/UnicalCoin [accessed: 2019-12-04].

[19] V. T. Le, "Trustable system for XBRL," 2001, URL: https://github.com/levanthanh3005/TrustableSystem-for-XBRL [accessed: 2019-12-04].

[20] Y.seki, "XBRL Core," 2006, URL: https://sourceforge.net/projects/xbrlcore/ [accessed: 2019-12-04].

[21] Ethereum Foundation, "web3.js - Ethereum JavaScript API — web3.js 1.0.0 documentation," 2019, URL: https://web3js.readthedocs.io/en/v1.2.1/ [accessed: 2019-12-04].

[22] Softo ltd, "Currencio — Cryptocurrency Converter," 2001, URL: https://fcurrencio.co [accessed: 2019-12-04].

[23] GitBook, "ETH Gas Station API," 2019, URL: https://docs.ethgasstation.info [accessed: 2019-12-04].

[24] 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.

[25] C. Pahl, N. El Ioini, and S. Helmer, "A Decision Framework for Blockchain Platforms for IoT and Edge Computing," in Proceedings of the 3rd International Conference on Internet of Things, Big Data and Security, no. IoTBDS, 2018, pp. 105–113, DOI: 10.5220/0006688601050113.

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

[27] 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.

[28] V. T. Le, C. Pahl, and N. El Ioini, "Blockchain based service continuity in mobile edge computing," in The 6th IEEE International Conference on Internet of Things: Systems, Management and Security. IEEE, 2019.

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

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

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

[32] Y. Guo and C. Liang, "Blockchain application and outlook in the banking industry," Financial Innovation, vol. 2, no. 1, 2016, p. 24, URL: https://doi.org/10.1186/s40854-016-0034-9 [accessed: 2019-11-11].

[33] Q. K. Nguyen, "Blockchain - a financial technology for future sustainable development," in 2016 3rd International Conference on Green Technology and Sustainable Development (GTSD), 2016, pp. 51–54, DOI: 10.1109/GTSD.2016.22.

[34] R. Ainsworth and A. Shact, "Blockchain (distributed ledger technology) solves vat fraud," SSRN Electronic Journal, 01 2016, DOI: 10.2139/ssrn.2853428.

[35] Y. Omran, M. Henke, R. Heines, and E. Hofmann, "Blockchain-driven supply chain finance: Towards a conceptual framework from a buyer perspective." IPSERA 2017 - Budapest - Balatonfüred., 04 2017.

[36] S. Helmer, M. Roggia, N. El Ioini, and C. Pahl, "Ethernitydb – integrating database functionality into a blockchain," in ADBIS, 09 2018, pp. 37–44, ISBN: 978-3-030-00062-2.

[37] H. Shafagh, L. Burkhalter, A. Hithnawi, and S. Duquennoy, "Towards blockchain-based auditable storage and sharing of iot data," in Proceedings of the 2017 on Cloud Computing Security Workshop. ACM, 2017, pp. 45–50.

[38] Storj Labs, "Storj : A Decentralized Cloud Storage Network Framework," Tech. Rep., 2018, URL: https://storj.io/storjv3.pdf [accessed: 2019-12-04].

[39] Protocol Labs, "IPFS Document," Tech. Rep., 2019, URL: https://docs.ipfs.io [accessed: 2019-12-04].

[40] X. Xu, I. Weber, M. Staples et al., "A taxonomy of blockchain-based systems for architecture design," in 2017 IEEE International Conference on Software Architecture (ICSA). IEEE, 2017, pp. 243–252, DOI: 10.1109/ICSA.2017.33.

[41] D. Asset, "Introduction of DAML," 2019, URL: https://docs.daml.com/daml/intro/0_Intro.html [accessed: 2019-12-04].

[42] T. L. Foundation, "Hyperledger," 2019, URL: https://hyperledger.github.io [accessed: 2019-12-04].

[43] E. Karafiloski and A. Mishev, "Blockchain solutions for big data challenges: A literature review," in IEEE EUROCON 2017 -17th International Conference on Smart Technologies, July 2017, pp. 763–768, DOI: 10.1109/EUROCON.2017.8011213.

[44] L. Yue, H. Junqin, Q. Shengzhi, and W. Ruijin, "Big data model of security sharing based on blockchain," in 2017 3rd International Conference on Big Data Computing and Communications (BIGCOM), Aug 2017, pp. 117–121, DOI: 10.1109/BIGCOM.2017.31.

[45] E. Bandara, W. K. Ng, K. De Zoysa, N. Fernando, S. Tharaka, P. Maurakirinathan, and N. Jayasuriya, "Mystiko—blockchain meets big data," in 2018 IEEE International Conference on Big Data (Big Data), Dec 2018, pp. 3024–3032, DOI: 10.1109/BigData.2018.8622341.

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

[47] Altova, "XBRL Development Tools," 2018, URL: https://www.altova.com/xbrl-tools [accessed: 2019-12-04].

[48] 5G-Carmen, "5G for Connected and Automated Road Mobility in the European UnioN," 2018, URL: www.5gcarmen.eu [accessed: 2019-12-04].