Chapter 9

A Decentralized Approach to Validating Personal Data Using a Combination of Blockchains and Linked Data

Cristina-Iulia Bucur, Fiorela Ciroku, Tatiana Makhalova, Ettore Rizza, Thiviyan Thanapalasingam, Dalia Varanka, Michael Wolowyk, John Domingue

The objective of this study is to define a model of personal data validation in the context of decentralized systems. The distributed nature of Linked Data, through DBpedia, is integrated with Blockchain data storage in a conceptual model. This model is illustrated through multiple use cases that serve as proofs of concepts. We have constructed a set of rules for validating Linked Data and propose to implement them in smart contracts to implement a decentralised data validator. A part of the conceptual workflow is implemented through a web interface using Open BlockChain and DBpedia Spotlight.

The current state of the World Wide Web is exposed to several issues caused by the over-centralisation of data: too few organisations yield too much power through their control of often private data. The Facebook and Cambridge Analytica scandal [84] is a recent example. A related problem is that the data poor, citizens who suffer from insufficient data and a lack of control over it, are thus denied bank accounts, credit histories, and other facets of their identity that cause them to suffer financial hardship. Over 60 UK citizens of the Windrush Generation have been erroneously deported because of a lack of citizenship data they once had that was lost due to government reorganization [7]. Such reliance on central entities for validation means that consumers are passing control over their privacy and authenticity of personal information.

This study examines the relation between decentralized data validation expressed through the integration of blockchain back-end data storage and Linked Data (LD). Decentralization is used to mean that no central authority has control over data and operations on this data. Blockchain technologies conform to the concept of decentralization: data is controlled and owned by the players in a neutral space/platform. Blockchains are useful for secure persistence, immutability, tracking and tracing all changes. Their main advantages is assessing the validity of how the data is used and keeping track of its usage. One of their disadvantage is that the technique involves no indexing. Thus blockchains have issues with search solutions. We propose to put an LD layer over blockchain. The LD is needed when storing data and data can be heterogeneous. An LD layer might help with querying, reasoning and to add semantics to the data.

This study addresses the following broad research questions:

- How does the concept of validity change in the context of a decentralized web?
- What does a decentralized approach to data validation look like?
- What benefits would accrue from a decentralized technology that supports validation in the context of LD?

The hypothesis of this research work examines whether Blockchains can provide a mechanism that can respond as a decentralised authentication platform to these questions. Blockchain is a distributed, public ledger that grows constantly as records of information exchange (transactions) are sequentially added to it in blocks [79].

The problem of Linked Data (LD) validity is that even though LD access is decentralized, its publication is centralized. The data production is not transparent. The validity must be trusted based on the authoritative institution publishing the data whose signature appears as an International Resource Identifier (IRI). IRIs are dereferencable, and can thus be dependably accessed with permissions or publicly. The data is typically not encrypted. Unauthorized access can allow the modification of information. The use of the data is not or is difficult to document and the processing of it is not transparent. For example, a matchmaking site would produce a record on the Blockchain every time they process a user profile, making the user aware of how their user profile is used in. Furthermore, it provides a safer storage of information as it distributes them over a (large) network of computers, making it more resilient to data loss or corruption.

In computer science, data validation is generally considered as a process that ensures the delivery of clean and clear data to the programs, applications and services using it" [6]. Beyond this definition that focuses on formal aspects of the data, the concept is also used in information science or data journalism as

”the process of cross-checking the original data and obtaining further data from sources in order to enrich the available information.” [57]. The term validation is also used in the blockchain context to describe the technical process that ensures that a transaction is validated by the network. In the context of this paper, data are said valid if one can assign to them a certain degree of trust and quality based on the validation of an authority or peer. It has two important aspects: validating data and how that data is used.

The motivation behind can be attributed to a number of well-known cases of misuse of trust by authorities who are in charge of centralised systems [66, 77, 97]. The decentralised nature of Linked Data means that it is also prone to the aforementioned vulnerabilities. W3Cs Verifiable Claims Group aims to make the process of expressing and exchanging credentials that have been verified by a third party easier and more secure on the Web [93]. This guideline would enable one to prove their claims, such as age for purchasing alcohol or creditworthiness, without having to share any private data that will eventually be stored in a centralised platform.

Our work is to present a solution for validating information on the Linked Data by leveraging the power of Blockchain technology. The selected data, DBpedia, is one of the recommended datasets by the summer school organizers (Dbpedia.com; ISWS 2018). The report presents a working demo that acts as a proof-of-concept and finally, we conclude the report by discussing the work required to maintain the sustainable growth of the network.

9.1 Resources

In our experiments we use the semantic annotation system DBpedia Spotlight. It allows for semantic queries in order to perform a range of NLP tasks. The tools can be assessed through a web application, as well as using a web Application Programming Interface (API).

- The DBpedia knowledge base [65] is the result of both collaborative and automated work that aims to extract of Wikipedia structured information in order to make them freely available on the Web, link them to other knowledge bases and allow them to be queried by computers [13].

- DBpedia Spotlight [30] is a service of Named Entity Linking based on DBpedia that looks for about 3.5M things of unknown or about 320 known types in text and tries to link them to their global unique identifiers in DBpedia. The system uses context elements extracted from Wikipedia and keyword similarity measures to perform disambiguation. It can be downloaded and installed locally or queried with open APIs in ten languages. There are a variety of Linked Data that can be utilised to further evaluate the viability of our framework.

- We use Open Blockchain implemented by the Knowledge Media Institute to interact with a blockchain through four sets of API: User API, Store

API, Util API and IPFS API. The first set of commands provides an authentication a user and managing its account. The sets of command Store API and Util API allow for fully interaction with the blockchain, including the requests for smart contracts stored in a blockchain and their hashes, registration of a new instance of the RDF store contract. IPFS API provides an assess to an IPFS storage.

9.2 Proposed Approach

Smart-Contract Response Principle Smart contract is an immutable self-executable code containing agreements that must be respected. For smart contracts, a set of rules is formulated as an executable code and the compliance with the rules is verified on the nodes. User have access to smart contracts. Since we deal with two different types of users (trusted and untrusted) we propose to use two different validation models in our framework. The obtained responses (decisions) can be processed in two different ways (w.r.t. the type of users) in order to get consensus-based response. We have constructed a general model that can be adapted for two different type of users.

The basis for the final decision is the majority vote. Let us consider how the majority vote model can be applied for the blockchain-based validation. On a query we get an infinite sequences of responses $r1, r_2, \dots, r_k$, (one response for one claim). The claims can either be accepted or rejected, i.e. $r \in D, D = 0, 1$, where 0 / 1 corresponds to reject / accept responses, respectively. To take the final decision, we define the function $f : D \to D$:

$$f(r_1, \dots, r_n) = [0.5 + \frac{\sum_i = 1, ..., n r_i - 0.5}{n}] \tag{9.1}$$

where $n \in N$ is a number of responses that are required for taking the final decision and [.] is the floor function, i.e., it takes as input a real number and gives as output the greatest integer less than or equal to this number. The function takes n first responses and returns 0 / 1 in case where the final decision is to accept or reject, respectively.

Weak Validation Model When trusted users access smart contracts we use a weak validation model (for example, we consider a model where to get a Schengen Visa it is sufficient to obtain approvement or rejection only from one country). In this model, n is a fixed value since all the responses are obtained from the reliable sources. In the simplest case, where $n = 1$, to return the final decision the function takes the first received answer. Since responses are trusted their number is supposed to be small.

Strong Validation Model When untrusted users access smart contracts, we require more strict approvement rules. In other words, we require strong validation for untrusted responses (for example,. when citizen assess to smart contracts the obtained responses should be verified carefully). We assume that the most of the users are trusted (or at least more than a half). In that case, the first model can be used when n is large, i.e., to take a final decision a lot of responses are needed to be received. The weakness of the application when the model for untrusted users is the following. As the number of required responses should be large, to get the final decision can take a lot of time. We propose to use the difference-based model, where the final decision is taken when the number of accept or reject answer exceeds a chosen value, i.e., n is not fixed in advance, the number of responses that is needed to be received depends on the difference in the number of obtained accept and reject responses:

$$n = argmin_{m \in N}[| \sum_{i=1,..,m} I(r_i = 0) - I(r_i = 1) |> Q]$$

Where $I(.)$ is an indicator function, it takes 0 / 1, when the condition in the brackets is false / true, respectively. Value n is the minimal value when the difference between the number of accept and reject responses exceed the chosen threshold Q.

Example 4. Let us consider how the majority vote models work in practice.

1. Case 1: n is fixed. Let $n = 7$, i.e., to take a final decision 7 responses are needed to be obtained. Assume the sequence of responses is 0101110.

   $$f(0, 1, 0, 1, 1, 1, 0) = [0.5 + \frac{4 - 0.5}{7}] = 1$$

   Thus, the final decision is "accept".

2. Case 2: n is not fixed and depends on the difference of the obtained responses. Let $Q = 2$. The and responses received are summarized in Table 9.1.

Sequence no. of responses	Response Value	Comments on the final decision
1	0	Q = 1, the decision cannot be taken
2	1	Q = 0, the decision cannot be taken
3	0	Q = 1, the decision cannot be taken
4	0	Q = 2, the decision can be taken, n = 4, $f(0, 1, 0, 0) = [0.5 + \frac{1-0.5}{4}] = 1$

Table 9.1: The principle of decision making for non-fixed number of responses. Q is the difference in the number of responses.

The proposed models have the limited-response drawbacks. It means that in cases where only a few responses can be obtained, the response time for the final decision might be great. To avoid the time-lost problem, Q might be non-fixed in advance and a limit on maximal response time T is fixed. In that case, the requirement of the final decision can be relaxed to get the final result within the chosen dataframes.

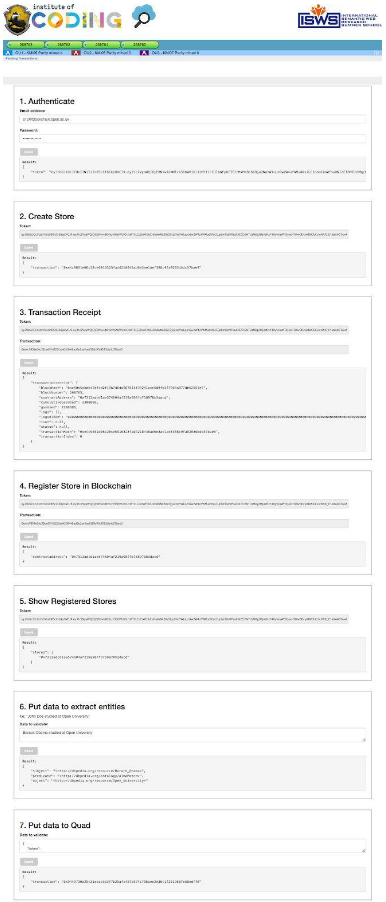
Proof of Concept In our proof of concept we developed an application to test the Open Blockchain [4](2018a) infrastructure and API together with a Linked Open Data dataset. A brief demo can be found here: `https://hufflepuff-iswc.github.io`. The application has the complete workflow needed to store the data on blockchain and link it with linked open data functionalities. The Screenshot of the application is provided in the Figure 1 and description of each step is listed below:

- In order to store the information in the blockchain the user should create an account and register himself with his credentials. In contradiction to the standard authentication methods - the users credentials are stored in encrypted decentralized way in the blockchain. After successful login the user gets an authentication token, which is then used for authorization of the next requests.
- In the next step the user has to create a new instance of the RDF store to put his data in using his authentication token. After the store is created it is put to a block and transaction number is returned back. Each transaction and block creation are visualised on top of the page.
- For the mining of a block some time is required. The user can check the status of the block mining by requesting the block receipt.
- Using authentication token and the transaction number the RDF store has to be registered in the blockchain. By registering the store the smart contract is created and the address of this contract is returned back to the user.
- At any time the user can check the RDF stores, which are associated with his account.
- In the next step the user has to load the file or data which has to be stored in the blockchain. The file/data is then automatically splitted to the validatable statements and semantic information in form of RDF triples is extracted from them.
- Finally, the extracted RDF data is stored in a transaction in the blockchain.

The user can choose which statements he wants to validate and select the trusted authorities suggestions provided by the system. Using the semantic information and fulfilling it with the contact information from the open sources the system will inform the authorities about the validation request. The validation request with the request status is stored in the users profile and can independently of other information be shared with the third parties.

In order validate the stored data the trusted authority signs the verifiable information using his private key. The data and the signature are put together to the blockchain application.

By the verification of validation the organisation sends a request with the document to the system. The system retrieves the stored sentences, extracted


268763
268762
268761
268760
OU1 - KMi05 Parity mined 4
OU2 - KMi06 Parity mined 0
OU3 - KMi07 Parity mined 0
Pending Transactions
Submit

1. Authenticate

Email address: s12@blockchain.open.ac.uk

Password: ••••••••••••••

```
Result:
{
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyaWQiOjE0NiwidXNlcm5hbWUiOiJzMTIiLCJlbWFpbCI6InMxMkBibG9ja2NoYWluLm9wZW4uYWMudWsiLCJpbml0aWFsaXNlZCI6MTUzMDg3
}
```

2. Create Store

Token: [illegible]

```
Result:
{
    "transaction": "0xe4c9851e06c28ce6916323fad4218448ad6e3ae1aef388c9fa9283dbdc57bae3"
}
```

3. Transaction Receipt

Token: [illegible]

Transaction: 0xe4c9851e06c28ce6916323fad4218448ad6e3ae1aef388c9fa9283dbdc57bae3

```
Result:
{
    "transactionreceipt": {
        "blockHash": "0xe50a5abdb42bfcd2f18b7d6da987814f39241cce6d0fb347964a877db62553e5",
        "blockNumber": 268763,
        "contractAddress": "0x7313adcd1ae574604af219a984fb758970b3dacd",
        "cumulativeGasUsed": 2308886,
        "gasUsed": 2308886,
        "logs": [],
        "logsBloom": "0x0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
        "root": null,
        "status": null,
        "transactionHash": "0xe4c9851e06c28ce6916323fad4218448ad6e3ae1aef388c9fa9283dbdc57bae3",
        "transactionIndex": 0
    }
}
```

4. Register Store in Blockchain

Token: [illegible]

Transaction: 0xe4c9851e06c28ce6916323fad4218448ad6e3ae1aef388c9fa9283dbdc57bae3

```
Result:
{
    "contractaddress": "0x7313adcd1ae574604af219a984fb758970b3dacd"
}
```

5. Show Registered Stores

Token: [illegible]

```
Result:
{
    "stores": [
        "0x7313adcd1ae574604af219a984fb758970b3dacd"
    ]
}
```

6. Put data to extract entities

F.e: "John Doe studied at Open University"

Data to validate: Barack Obama studied at Open University

```
Result:
{
    "subject": "<http://dbpedia.org/resource/Barack_Obama>",
    "predicate": "<http://dbpedia.org/ontology/almaMater>",
    "object": "<http://dbpedia.org/resource/Open_University>"
}
```

7. Put data to Quad

Data to validate:

```
{
  "token":
```

```
Result:
{
    "transaction": "0x9449730a25c31ebcb3b3f7b25afc4b7b57fc706aee2e36c14251968fc60edff8"
}
```

Figure 9.1: Screenshot of the proof of concept application

RDF triples together with the signature of the trusted authorities and if the information could be validated successfully, puts a validation badge for each statement.

The prototype consists of multiple components which can be seen in the architecture overview in Figure 9.2. The user makes a HTTP request to the API, where he uploads the document that should be validated by the system (1). The Named Entity Recognition system extracts the semantic entities using natural language processing techniques (2). The entities are represented as RDF triples and combined together with information from the Linked Open Data cloud (4) put to the Open Blockchain network (5.1). In the network, the document and the RDF triples are stored in an InterPlanetary File System (IPFS) distributed file storage network and the retrieved hash is stored in the blockchain transaction(5.2) (`http://ipfs.io`). This information is then stored

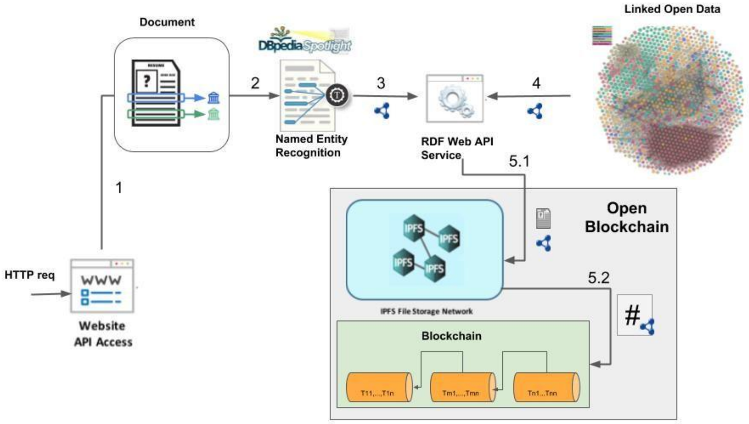


Figure 9.2: Architecture Overview. Adapted from Domingue, J. (2018) Blockchains and Decentralised Semantic Web Pill, ISWS 2018 Summer School, Bertinoro, Italy.

for validation.

Use Cases

The proposed distributed validation approach can be used in multiple different use cases.

Blockchain Dating The first suggested use case is storing dating data on a blockchain. In this case, the semantic triples from all personal dating-relevant data (e.g, interests, age, ex-partners, etc.) are extracted, encrypted, and put in IPFS. The retrieved hashes are stored on blockchain. Permissions are defined to allow and describe how and what parts of this personal data can be used by different services or dating websites. The description of data usage and permissions is written in a separate smart contract on the blockchain that is signed with each individual dating service provider. This ensures that the owner of the data is in full control of which platform uses what parts of data and how that data is used. Validation of the user data can be done by the peers in the blockchain network that have interacted with the user. As there is no trusted authority that can officially validate all the personal information like interests or events attended, the peers will (in)validate the presented information about the user. A trust system can be used to strengthen the validation system.

Distributed Career Validation Another possible use case is a distributed career validation system. The system should store and verify education, skill, and career information for individuals. The qualification documents are stored in distributed secure way and due to the qualities of blockchain can not be changed and will never disappear. The system saves the business resources and effort for recruiting and validation of the job applications. The authorities in this case are universities, online schools, and previous employees.

Splitting a document such as a curriculum vitae (CV) in small easily verifiable pieces of information and fulfilling the missing information using semantic inferences can help authorities, such as universities, former employers, etc., easily prove the validity of the information the candidate has provided. And the new employer can trust that the authority proved the data provided by the candidate and it was validated.

Blockchain Democracy Blockchain-based authentication systems provide a more secure mechanism than conventional identity tools since they remove the intermediaries and as they are decentralized, the records are retrievable, even after cases of disaster. In order to achieve a successful transition between a centralized government to a decentralized one, the data in all the official databases needs to be transferred on the blockchain. Whenever new data is to be added in the blockchain, the smart contract regulates the process of validation as a governmental official will confirm or not the truthness of the data.

In the case of e-Estonia, the citizens are can identify themselves in a secure way and every transaction can be approved and stored on the blockchain. The communication between different departments of the government is shortened in time, which makes the institutions more efficient. In the case that a citizen needs a certificate from the government, they identify themselves in the system and send the request to an institution. The employees of the institution (miners) are competing for the task and the first that completes the task is rewarded inside the blockchain. As soon as the task is done, it is stored in the system and can be accessed by the citizens.

9.3 Related Work

Zyskind and others (2015) defined a protocol that turns a blockchain into an automated access-control manager without need to trust a third party. Their work use blockchain storage to construct a personal data management platform focused on privacy. However, the protocol does not use LD and has not been implemented. To the best of our knowledge, there has been no follow-up to this work.

Previous work on validating Linked Open Data with blockchains includes several researches at the Open University [5](Open BlockChain 2018b). Allan Third et al. [96], for instance, compares four approaches to Linked Data/Blockchain verification with the use of triple fragments.

Third & Domingue (2017) have implemented a semantic index to the Ethereum blockchain platform to expose distributed ledger data as LD. Their system indexes both blocks and transactions by using the BLONDiE ontology, and maps smart contracts to the Minimal Service Model ontology. Their proof of concept is presented as a first steps towards connecting smart contracts with Semantic

Web Services. This paper as well as the previous one focuses on the technological aspects of blockchain and does not describe case studies related to privacy issues on the Web.

Sharples & Domingue [90] propose a permanent distributed record of intellectual effort and associated reputational reward, based on the blockchain. In this context, Blockchain is used as a reputation management system, both as a proof of intellectual work as an intellectual currency. This proposal, however, concerns only educational records, while ours aims is to address a wider variety of private data.

9.4 Conclusion and Discussion

In the present work, we propose a novel approach for validating LD using the Blockchain technology. We achieved this by constructing a set of rules that describes two validation models that can be encoded inside smart contracts. The advantages of using Blockchain technology with Linked Data for distributed data validation are: 1) The user maintains full control over their data and how this data is used (i.e. no third party stores any personal information), 2) Sensitive data is stored in a distributed and secure manner that minimises the risk of data loss or data theft, 3) The data is immutable and therefore a complete history of the changes can be retrieved at any time, 4) RDF stores can be used for indexing and for searching for specific triples in Linked Data; 5) Using LD, information can be enriched with semantic inferences; 6) Using smart contracts means that the validation rules on the decentralised system are reinforced forever.

However, the framework presented in the paper has a few limitations: 1) It is vulnerable to all weaknesses that the Blockchain technology suffers from (e.g. smaller networks are vulnerable to 51% attack); 2) It requires a certain degree of trust in government organisations for maintaining accurate information about the data (i.e. garbage-in-garbage-out), and 3) In our formalisation we proposed to use a time-independent smart contract consensus model (where the parameters of the function that produces the final response are fixed). The model suffers from a time-loss problem in time-lag cases. This model can be further improved by defining time-dependent parameters that ensure obtaining a response in the defined time-frames.

Building a decentralized system that uses blockchain technology to support the validation of LD opens up the possibility for secure data storage, control and ownership. It enables a trusted, secure, distributed data validation and share the only explicitly required information with the third parties. In the future work, we plan implement the validation and verification workflow described in our approach and to improve the limitations mentioned above.