Transcripts DApp—A Blockchain-Based Solution for Transcript Application

Shrinivas Khedkar, Akhil Powar, Nikhil Powar, Chethan Kille, and Harsh Kansara

Abstract There is a need for a reliable system with an efficient and simplified process for academic transcript application and procurement along with sufficient measures for authentication of issued transcripts and verification of the integrity of transcripts. The work presented in this paper proposes to create a blockchain-based decentralized application which can be used by students for application of transcripts, which can only be approved and issued by the intended institute, and by universities for the verification of the transcripts issued. The proposed system attempts to leverage the immutability and transparency of blockchain and versatility of programming provided by smart contracts to ensure that there is no avenue for alteration or forgery of transcripts.

Keywords Blockchain · Transcripts · DApp

1 Introduction

The current infrastructure for transcripts generation and validation is largely based on physical processes and paperwork. This often requires the presence of student or some representative to physically deliver required documents and/or collect the issued copies of transcripts. There is extra work required by the university receiving the

S. Khedkar · A. Powar · N. Powar (✉) · C. Kille · H. Kansara
Veermata Jijabai Technological Institute, Matunga, Mumbai 400019, India
e-mail: napowar_b15@ce.vjti.ac.in

S. Khedkar
e-mail: sakhedkar@ce.vjti.ac.in

A. Powar
e-mail: aapowar_b15@ce.vjti.ac.in

C. Kille
e-mail: cpkille_b15@ce.vjti.ac.in

H. Kansara
e-mail: hjkansara_b15@ce.vjti.ac.in

D. Patel et al. (eds.), *IC-BCT 2019*, Blockchain Technologies,
https://doi.org/10.1007/978-981-15-4542-9_15

student's transcripts to verify their authenticity. Also there is an added disadvantage of unregulated cost and waiting time to get the transcripts from universities.

There are a few online systems related to application and verification of transcripts. However, many of these are only partly online, and the transcripts obtained are actually physical documents. Completely digitized systems that use PKI place dependency on a third-party. The few systems that do use blockchain are not truly decentralized. They depend on centralized storage, issuers, or centralized credential stores. Opting for storage on the blockchain itself sharply increases the costs associated with any blockchain-based system. Due to absence of a universal log-in/sign-up method, identities on the blockchain-based applications are tied to a particular blockchain address, which greatly reduces mobility for users as they are constrained to keep utilizing the same blockchain address for all transactions. It also leads to a dependency on providers (such as MetaMask, Mist, etc.) to function.

The application of blockchain technology to tweak the current system will ensure transparency and avoid fraudulent information as the system is consensus based and the ledger is public and immutable. A blockchain-based solution will ensure reduced paperwork and hence reduced cost and waiting times for the applicants. Third-party validation will also be eliminated and a trust-less system can be generated. The DApp is also designed to provide true mobility to users by use of identity contracts.

In the rest of this paper, we provide backgrounds of concepts that enable the proposed system, review previous work in related areas of application, provide an overview of the proposed system design, and its working and then lay out possible avenues for future works based on or related to the proposed system.

2 Blockchain

The blockchain technology was formally introduced by Satoshi Nakamoto in their work on Bitcoin [1]. It is an incorruptible digital ledger that records any sort of transaction in a network. Transactions are grouped together and stored as a "block," which is then added to the chain.

Blockchain uses a consensus algorithm to handle addition of new blocks. The algorithm helps the blockchain to ensure a state of consensus and thus maintain the integrity of the blockchain.

Each block in the chain contains the hash of the preceding block. The hash of the new block is calculated by taking into account the transactions included in the block and the hash obtained from the previous block. Thus, any alteration in a block's data will lead to invalidation of that block and all others that had succeeded it. The comparison of hashes thus ensures the validity of the blockchain.

2.1 Smart Contract

A smart contract is a self-executing computer program that is used to perform required tasks when triggered by some predefined conditions [2]. Smart contracts are used to achieve decentralized automation. Smart contracts are publicly visible to those with access to the blockchain.

A smart contract is immutable; in that, once a smart contract is deployed on the blockchain, it cannot be changed. A change in the contract, if absolutely required, must be brought about by creating an entirely new contract.

A smart contract's transactions, including deployment, are recorded in the blockchain similar to any other transaction. Smart contracts may be programmed to carry out any type of general purpose functions.

2.2 Transcripts

A transcript is a record of the courses taken by a student and the grades earned by the student in those courses. These are official documents which can be verified by an authorized entity/person. Generally, universities charge a fee for each copy of the official transcripts. Applicants send their transcripts via post to the admission office of the university in which they seek admission in.

2.3 Decentralized Storage

The blockchain is not meant for data storage as storing large documents will be very expensive. It is a public ledger meant to record transactions which are inexpensive to store as compared to conventional files, owing to their fixed, limited size. Storing files on the blockchain would require us to store them as transactions. Large files would be spread over several transactions, which may span multiple blocks.

Interplanetary File System (IPFS) [3] is a peer-to-peer file-sharing system. It uses a distributed hash table (DHT) so that the data is spread across a network of computers, and efficiently coordinated to enable efficient access and look-up between nodes. The main advantages of DHT are decentralization, fault tolerance, and ability to be scaled. Nodes do not require central coordination; the system can function reliably even when nodes fail or leave the network, and DHT can scale to accommodate millions of nodes. Together these features result in a system that is generally more resilient than client–server structures.

To maintain the integrity, the data blocks are stored as a Merkle DAG. This is done by organizing data blocks using hash functions. It is simply a function that takes an input and calculates a unique alphanumeric string(hash) corresponding with

that input. All content on IPFS can be uniquely identified, since each data block has a unique hash. Plus the data is tamper-resistant because to alter it would change the hash.

2.4 Decentralized Application (DApp)

Decentralized application is an application that has its back-end code running on a decentralized peer-to-peer network. A truly decentralized application includes decentralized storage. DApps, by definition, do not have a single point of failure.

3 Related Work

Traditionally, applications involving document transfer and application process have been implemented using public key infrastructure [4]. These methods have employed centralized certificate authorities to authenticate both ends. As such, they are vulnerable to the problems of a trust-based system including third parties [5].

Transcript application using blockchain aims to address all drawbacks faced by earlier methods. Sony Global Education has one such implementation which uses Hyperledger Fabric as the underlying blockchain for managing transcripts and scores [6]. However, all documents must adhere to one of the predefined structures for validity.

Learning Machine, in collaboration with MIT Media Laboratory, has created the Blockcerts tool-set, which is an open infrastructure for creating, issuing, viewing, and verifying blockchain-based certificates [7]. This can be used for purposes beyond education, with professional document sharing and managing capabilities. This tool-set does not allow for changes in the issued certificates. Any errors in the certificate thus result in revocation of the original certificate and issuing of a new one.

Smart Cert is an initiative based on the Blockcerts tool-set, which are digital certificates registered on a blockchain [8]. These certificates are signed using cryptography and can be shared. With Smart Cert, the certificates are not stored in a decentralized storage, only their hashes are. The certificates themselves are stored in a centralized storage.

C Verification's solution is a blockchain-based recruitment and background verification platform [9]. It allows the user to store verified testimonies of their professional achievements and share it with all potential employers. However, C Verification is still a third-party between the issuer and consumer of the references, which is not desirable in our scenario. Also, C Verification charges a fee from its users for every action, retaining 30–100% of it depending on the use case.

BC Diploma provides a platform to certify diplomas by associating Ethereum technology with cryptography [10]. BC Diploma is a DApp that can be used by institutions of higher education to issue their degrees on Ethereum. Although it is

fully decentralized, the diplomas are stored on the chain in text format and in fixed sizes, which requires standardization of diplomas. Such standardization does not exist and forcing current diplomas into constraints would result in loss of information.

A paper published by researchers at Southern Taiwan University of Science and Technology [11] describes an application that manages and verifies digital certificates. In this, the authors have proposed a Web-based application to store digital certificates which are then validated using serial numbers issued by the system or using QR codes. The universities themselves fill out student details while uploading the digital certificates, which can then be shared by the student with other universities or with employers.

4 Transcripts DApp

The following sub-sections describe the proposed system design. Figure 1 depicts the architecture of the proposed system.

4.1 Framework of the System

The system goes through the following steps:

1. The admin adds the academic institute to the institute list in the 'Entity List' smart contract. This is usually done by the admin after they have performed some sort of authentication and verification of the institute.
2. The student logs into the system using their sub-domains assigned. For first-time log-in, a sub-domain is created and assigned to the student. The student is hereafter represented entirely by their ID contract. Further details about this step can be found in the next section.
3. The student creates a new application, or proceeds directly to step 5. By creating a new application, a new 'Transcript Application' contract is created. Further details about this step can be found in the next section.
4. The institute then approves the application of the student. Further details about this step can be found in the next section.
5. The student then views the application status. If application has been approved, student may access the transcript document directly though it or may choose to download it.
6. The student may also share visibility of their transcript with others.

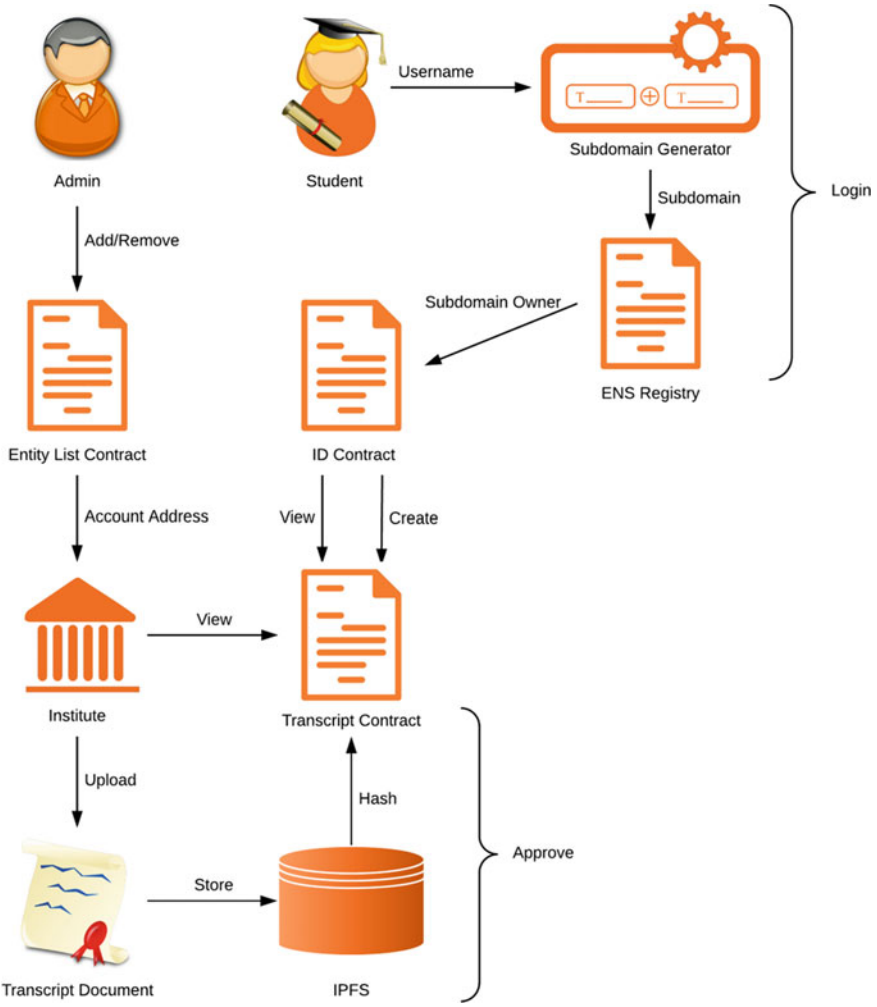

Fig. 1 Architecture of transcripts DApp

4.2 Description of Major Functions

The implemented prototype performs the following major functions:

Log-in and Sign-up
The implemented prototype uses a universal log-in and sign-up procedure as described in Ethereum Improvement Proposal EIP-1078 [12]. The DApp controls a high-level domain name based on the Ethereum Name Service. The username is

used to create a unique sub-domain for each user, which is the only input required from the user for log-in. The created sub-domain is assigned to an ID contract, as defined in the Ethereum Improvement Proposal EIP-725 [13]. After the initial creation of sub-domains, the user may log-in using the sub-domain name and using a wallet provider that is registered as a level-1 (Admin) key of the ID contract.

The process of universal log-in and sign-up is as follows:

1. A user who wishes to log-in provides a username, which is prefixed to the domain name to derive a sub-domain.
2. The DApp checks availability of entered sub-domain name with the ENS registrar on the current Ethereum network.
3. If sub-domain name is taken, skip to step 6. Else, if sub-domain is available, the DApp prompts user to verify entered name, or create a new sub-domain
4. For creating a new sub-domain, the DApp creates a new ID contract for the user and assigns the current wallet being used by user as the Admin Key (level-1 key) of the ID contract.
5. The DApp registers the desired sub-domain and transfers its ownership to the newly created ID contract.
6. If user attempts to log into a previously registered sub-domain, the DApp checks if the key is present in the associated ID contract.
7. If the key is already present in the ID contract, the DApp concurs that the user is verified and successfully logs them in. Else, the DApp asks for a new transaction for entering the new key as an Encryption Key (level-4 key).
8. By measures provided in the ID contract, addition of keys can be done successfully only when the transaction is signed by an Admin Key (level-1).
9. The DApp provides a QR code or a URL for the user to sign the transaction using their admin key.
10. After success of the transaction, the user is logged into the DApp.

For institutes and admins, the procedure for registration differs. They are required to contact the current admins at the provided correspondence points. After performing the requisite authentication and verification of their identities, the admins themselves add new institutes and admins. The list of current admins and institutes is stored using a smart contract. The admins also hold the power to remove institutes found guilty of misconduct.

The process of registration and log-in for academic institutes is as follows:

1. The representative of the institute contacts the administrators regarding registration of the institute on the DApp.
2. The administrators, after performing some requisite manual verification, request the Ethereum account address which would be used to represent the institute.
3. The administrator adds the institute as a name–address pair into a Entity List contract.

4. The Entity List contract contains the names and address of all the institutes registered with the DApp.
5. The Entity List contract has built-in checks to only allowing existing administrators, the permission to add new institutes, or remove existing institutes.
6. The institute may then log-in using their registered account address.

Create Transcript Application

The DApp creates a new smart contract for each new application made by a student. The contract stores the details provided by the student and also identifies the institute that is responsible for approving the application. The DApp determines the status of application using the presence or absence of the hash value.

The DApp keeps track of all the created applications using a contract that stores the list of applications. One list is maintained for each user, which stores both unapproved and approved applications.

The process of creating a new transcript application is as follows:

1. The student logs into the DApp using the previously mentioned procedure.
2. The student chooses to create a new application and fills in the required details in an application form.
3. The student may select any institute they desire to obtain their transcripts from, known as the provider, from the list of institutes registered with the DApp.
4. When the student submits the application form, the DApp validates all fields to check for proper input formats for all fields.
5. The application may or may not require a fee for processing, as per the requirements of the institute.
6. The DApp creates a new Transcript Application contract using the data received from the student.
7. The Transcript Application contract has a field which determines the institute that can approve the application.
8. The Transcript Application contract also has a field that stores the hash value of the uploaded transcript obtained from IPFS. A non-existent hash value denotes that the application is yet to be approved.
9. The DApp stores the value of the newly deployed Transcript Application contract in the Transcript List contract.
10. The Transcript List contract has a list of all associated contracts for each user. For students, it stores the addresses of all the Transcript Application contracts that have been created by them. For institutes, it stores the addresses of all Transcript Application contracts that have been created with them as the authorized provider.

Approve Transcript Application
The smart contract is designed such that it can be approved only by the designated institute. The institute merely uploads the transcript document using the DApp. The process of storing the document on IPFS, storing the obtained IPFS hash in the contract, and fetching of the document using the stored IPFS hash are performed by the DApp behind the scenes. Once the file is uploaded and the hash in the smart contract is set, the DApp considers the application as approved.

The process of approving a transcript application and uploading the transcript is as follows:

1. The institute logs into the DApp using their registered account address.
2. The DApp displays a list of all transcripts that mention the concerned institute as the provider using the data stored in the Transcript List contract.
3. The institute may view the details of the application and the applicant as fetched from the Transcript Application contract.
4. The institute may perform some requisite verification to determine whether the applicant mentioned qualifies for the desired transcripts.
5. Once verification is complete, the institute prepares a transcript which may be completely digital or may have physical copies. Either way, the transcripts must be converted into PDF file format for uploading.
6. When the institute uploads the contract, the DApp connects to a remote IPFS node to upload the file onto IPFS.
7. The IPFS upload method returns a hash which is stored in the Transcript Application contract.
8. Due to built-in checks, the Transcript Application contract only permits the institute specified by the user to set the field denoting the IPFS hash of the transcript.
9. The Transcript Application is thus considered as approved and the stored transcript may be accessed by the student.
10. The IPFS hash uniquely determines the stored transcript, and the immutability of blockchain ensures that the hash value once set in the Transcript Application contract cannot be changed by any other means.

This procedure uses the properties of both smart contracts and IPFS to ensure the integrity and authenticity of stored transcripts.

5 Smart Contract Pseudo-Code

The pseudo-codes of the smart contracts utilized in the DApp are described below:

Transcript Application Contract

```
constructor (address _owner, address _provider,
    string _name, string _id, string _courseName,
    int _startYear, int _completionYear) {
  transcriptHash = "Not set";
  transcriptOwner = _owner;
  providingAuthority = _provider;
  name = _name;
  id = _id;
  courseName = _courseName;
  courseStartYear = _startYear;
  courseCompletionYear = _completionYear;
}

function setTranscriptHash (string s) returns (string) {
  if(msg.sender != providingAuthority) {
    return "Error";
  }
  transcriptHash = s;
  return "Success";
}
```

Transcript List Contract

```
mapping(address => address[]) transcriptList;

function addTranscript (address student, address
    college, address transcriptAddress) {
  transcriptList[student].add(transcriptAddress);
  transcriptList[college].add(transcriptAddress);
}

function getTranscripts (address account) returns
    (address[]) {
  return transcriptList[account];
}

function removeTranscript (address transcriptAddress) {
  foreach(entity in transcriptList) {
    entity.remove(transcriptAddress);
  }
}
```

Entity List Contract

```
function addAdmin (address addr) returns (string) {
  if(!isAdmin(msg.sender)) {
    return "Error";
  }
  admins.add(addr);
  return "Success";
}

function addProvidingAuthority (string memory name,
    address addr) returns (string) {
  if (!isAdmin(msg.sender)) {
    return "Error";
  }
  providingAuthorities.add(ProvidingAuthority(name,
      addr));
  return "Success";
}

function removeProvidingAuthority (address addr)
    returns (string) {
  if (!isAdmin(msg.sender)) {
    return "Error";
  }
  providingAuthorities.remove(addr);
  return "Success";
}
```

6 Costs and Scaling

The costs associated with various types of transactions in the DApp are shown in Table 1. These transaction costs were determined through the Ropsten test network based on the value of Ether at the time of writing.

Apart from the costs mentioned in Table 1, institutes may also choose to charge some fees for the transcripts, which can be paid during the submission of application.

The scaling of the DApp is directly tied to the rate of transactions on the Ethereum network. As of writing, Ethereum supports a rate of about 15 transactions per second and a block time of about 10–20 s.

Table 1 Transaction costs

Role	Action	Approximate cost-ether (USD)
User	Registration	0.000311 (0.04)
	New transcript application	0.000291 (0.03)
Institute	Uploading transcript	0.000022 (0.00)
Administrator	Add new institute	0.000021 (0.00)
	Add new administrator	0.000015 (0.00)
	Remove institute	0.000012 (0.00)

7 Conclusions and Future Work

We conclude that the proposed DApp has significant benefits in terms of costs, time required and convenience. The DApp is fully decentralized, which eliminates any dependency on third-party verification agencies. Using smart contracts, we ensure that only the authorized institute can issue transcripts to students. The hash of uploaded transcripts uniquely determines the transcript file on IPFS, thus ensuring that the file cannot be tampered with.

The starting point for future work would be working on a concrete proof of concept which delivers a decentralized trust-less system as proposed in the design while providing at least the same level of security as provided by current system of physical transcripts.

Currently, the proposed system is limited to handle academic transcripts. Possible areas of extension of the proposed system would be design of similar systems to handle other documents such as certificates and performance reports or integration of all of the above in a single system.

References

1. Nakamoto S (2009) Bitcoin: a peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf
2. Buterin V and others (2014) A next-generation smart contract and decentralized application platform
3. IPFS, https://ipfs.io/ipfs/QmR7GSQM93Cx5eAg6a6yRzNde1FQv7uL6X1o4k7zrJa3LX/ipfs.draft3.pdf. Accessed 10 December 2018
4. Maurer U (1996) Modelling a public-key infrastructure. In: European symposium on research in computer security. Springer, pp 325–350
5. Ellison C, Schneier B (2000) Ten risks of PKI: what you're not being told about public key infrastructure. Comput Security J 16(1):1–7 (2000)
6. SGE Education blockchain, https://blockchain.sonyged.com/. Accessed 10 December 2018
7. BlockCerts, https://www.blockcerts.org/about.html. Accessed 10 December 2018
8. Blockchain Imperative for Educational Certificates, https://gosmartchain.com/whitepaper/SmartChainUniversity-whitepaper.pdf. Accessed 11 December 2018

9. Blockchain-based recruitment and background verification platform, https://cverification.com/src/docs/Cverification_Whitepaper.pdf. Accessed 11 December 2018
10. BCDiploma, https://www.bcdiploma.com/ico/img/BCD-WhitePaper_last.pdf. Accessed 11 December 2018
11. Cheng J, Lee N, Chi C, Chen Y (2018) Blockchain and smart contract for digital certificate. In: 2018 IEEE international conference on applied system invention (ICASI), pp 1046–1051 (April 2018)
12. Van de Sande A (2018) EIP 1078: Universal login/signup using ENS sub-domains. https://eips.ethereum.org/EIPS/eip-1078. Accessed 10 December 2018
13. VogelSteller F (2018) EIP 725: Proxy Identity. https://eips.ethereum.org/EIPS/eip-725. Accessed 10 December 2018