Journal of Data, Information and Management (2021) 3:237–249
https://doi.org/10.1007/s42488-021-00056-0

ORIGINAL ARTICLE

A Blockchain-Based Smart Contract Towards Developing Secured University Examination System

Ashis Kumar Samanta[1] · Bidyut Biman Sarkar[2] · Nabendu Chaki[1]

Received: 10 June 2020 / Accepted: 15 June 2021 / Published online: 16 July 2021

Abstract
A smart contract (SC) is a digital negotiation protocol among two or more anonymous parties without the presence of any trusted intermediaries. It is a self-executing agreement in the form of computer code. Smart contracts run on the blockchain. Thus, the code and the agreement are stored on a distributed public database and cannot be changed. There is a broad range of prospective application scenarios for smart contracts in the digital economy, including financial services, management, healthcare, and the Internet of Things. Ethereum and Hyperledger, are two of the most widely used open-source advance cross-industry blockchain technologies. Major technical challenges such as security, privacy, correctness, and verifiability are among the issues that are yet to be matured and completely resolved for blockchain. In this work, a comprehensive survey on smart contracts is carried out. We also present a case study on a University examination system having a heterogeneous data structure. This deployment brings a comprehensive understanding of the smart contract framework and has been used to find and analyze the gaps in the state of the art in terms of smart contracts.

Keywords Block-chain · Smart contract · Hyper ledger fabric · Heterogeneous data structure · Examination process · Data security

1 Introduction

Applications starting from financial sectors, industrial research, healthcare, agriculture, electronic voting & digitally recorded property assets to regulatory compliance & trading are now aggressively being developed and deployed at a faster pace on a distributed blockchain network.

A smart contract is a digital negotiation protocol among the anonymous parties without the presence of any trusted intermediaries to facilitate the exchange of money, content, property, shares, or anything of value programmatically (computer coded) without any possibility of fraud.

✉ Ashis Kumar Samanta
aksdba@caluniv.ac.in

Bidyut Biman Sarkar
bidyut.biman.sarkar@tict.edu.in

Nabendu Chaki
nabendu@ieee.org

[1] Department of Computer Science and Engineering, University of Calcutta, Kolkata, India

[2] MCA Department, Techno International Newtown, Kolkata, India

A smart contract is one of the techniques of blockchain applications, where different security issues have been considered. Subsequently, the probable solutions to those security concerns are proposed by different researchers (Watanabe et al. 2016; Mori and Miwa 2020). In this article, we have analyzed the security issues for the public and the private blockchain and designed our proposed prototype system to solve the same. Hyperledger Fabric is an open-source framework of blockchain, that supports data privacy hosted by the Linux Foundation (Wang et al. 2019). We use Hyperledger for our application.

In education system, the examination is an instrument to assess the students and rank them following a uniform yardstick. Educational institutions have to maintain assessment records of students for a substantial period of time. These data are not only used for verifying the academic credentials of the students, but also for the purpose of further analysis towards bringing in more effective and fruitful reforms. Thus, it is essential to maintain such data in a secured environment and to make sure that it is not tempered or deleted. Besides, the process of examination should also have the maximum possible transparency and trustworthiness. The volume of such type of data is ever-increasing and are required for rendering different services almost continuously. Besides compliance to

security issues, these services are to be rendered at a low cost. The execution of the services may be done through a smart contract between the educational institution and the students or between the educational institution and any authorized third party. The smart contracts in blockchain technology would secure security, trust and as well as the reduction in time and monetary cost.

This work aims to enact and produce a secure smart contract-based system for an examination system of a large University with a high number of affiliated colleges under it. The system should be such that, it is free from different kinds of vulnerabilities, and with no single point of failure. In addition to cryptographic transaction processing of the smart contract between the receiver and the sender, the application should be well protected against hacking attacks and fraudulent activities.

We strongly believe that a large University needs a system that can provide such a secured and online system to maintain its reputation. On-time delivery of services at the least cost is the call for the day. This is a unique operational challenge. We refrain from disclosure of the name of the University that has been studied for the sake of anonymity. However, the institute that has been considered for the empirical purpose is one of the largest, state-funded, and oldest modern University in India.

In order to demonstrate the security issues, we have extracted and studied around 150 articles from Digital archives such as IEEE, Springer, ACM, and Science Direct. We have reviewed 22 numbers of publications and their security disputes in Block-chain-enabled distributed networks.

The structure of this paper is as follows: we present the literature review on smart contract blockchain technology in section 2. Section 2.1,2.2 and 2.3 represents the application-specific, model-based, and tools designing smart contract respectively. Section 2.4 represents the inference of the literature review of these three types of smart contract. The methodology of the deployment of a smart contract for the University examination system is presented in section 3. Section 3.1 represents the existing data management system and the limitations of the existing system. Section 3.2 represents the proposed system of a smart contract of the University examination data management. Section 3.3 represents the smart contract algorithm of the proposed system. Section 3.4 represents the experimental setup. The system generated output is presented in section 3.5. Cost estimation is presented in section 4. The conclusion and future trends are presented in section 5.

2 Literature review

There has been considerable efforts towards enhancing the security of business processes besides maintaining transparency and tracability using smart contracts. This has been documented in a good number of research articles available on smart contracts. We have divided the related existing literature into three sections:

- *Application-specific Smart Contracts:* In section 2.1, we have presented our study on different applications of the smart contracts that have been proposed to address the security issues. The developers have done using different platforms like the ethereum framework, IoT, game theory, or some other platform.
- *Model-Based Smart Contracts:* In section 2.2, we discuss the different models that have been proposed to maintain the privacy and security of the stored data using different frameworks.
- *Tool Designing Smart Contracts:* In section 2.3, we compare the different types of tools that have been proposed to address the security and privacy of stored data of the smart contract.

It is observed from a systematic study science 2016, that there is an increasing trend in the publication of blockchain-based smart contract articles focusing on experiments, methods, tools, and models. The commonly discussed subjects and explanations in these works are correlated to the security, concealment, privacy, and scalability of blockchain and the programmability of smart contracts.

Based on the detailed study and analysis of the existing research work on smart contracts, we strongly believe that the researchers have contributed quite significantly to mitigate the security issues of smart contracts by utilizing the security and transparency property of smart contracts. Table 1 through Table 3 summarizes the three different types of smart contracts as classified at the beginning of the current section.

2.1 Review on application-specific smart contracts

In Table 1, some of the interesting application-oriented smart contracts proposed for diverse business domains have been presented in brief with our observations and opinion for each of those.

In Table 1, the number of papers are pointing towards different types of industrial smart contract applications. Network node selection from the XML document is performed for security analysis. It is an application reliant work and not of generic type (Tikhomirov et al. 2018). A few particular types of network security and privacy of the chain are experimented in (Covaci et al. 2018). The statistical concept is used to explain the data security for fraudulent transaction processing using the probability theory on gambling (Watanabe et al. 2016). The three-tier architecture of a smart contract is described and implemented over a public smart voting system and highlighted the data

Table 1 Application-specific Smart Contracts

Paper	*Sources*	*Product/Objectives*	*Comments*
(Tikhomirov et al. 2018)	2018, ACM WETSEB	The smart contract source code written in Solidity is converted into XML format and checks it against X-Path patterns. X-Path uses a path expression to select the node or a list of nodes from an XML document.	Static Analysis of Ethereum Smart Contracts is executed using Solidity code. It is used to remove the simple bugs, which are causing the attack on Smart Contract. The authors also presented an efficient tool for analysis by improving the grammar of patterns. The security mechanism is improved as it supports the debugging of code of Solidity for Etherium.
(Covaci et al. 2018)	2018, IEEE	NECTAR is a Non-Interactive Smart Contract Protocol using blockchain. It focuses on the security, privacy, and correctness issue of the smart contract.	NECTAR is used to validate code in the network without its re-execution. Attacks like man-in-the-middle, malicious tampering, etc. are successfully handled using NECTAR. Other security issues need to be explored. The NECTAR application does not demand any pre-requisite in terms of expertise of the user for implementing a secured smart contract, either public or private.
(Watanabe et al. 2016)	2016, IEEE	Gambler's ruin problem is used to prevent an attacker from monopolizing resources and to keep securing the blockchain.	A persistent gambler who raises his bet to a fixed fraction of bankroll when he wins. He does not reduce it when he loses, will eventually and inevitably go broke, even if he has a positive expected value on each bet. The implementation of Proof of Stake (PoS) prevents the threats of resource monopolizing.
(Tonelli et al. 2019)	2019, IEEE	This work implemented Microservices System with blockchain, Smart contract. It is used to maintain data integrity in smart voting.	The proposed three layer architecture is the key contribution of this work: a.Graphical user interface. b.API-Gateway to connect devices to the system through the API-Gateway. c.Implementation problems for the microservices are solved by smart contracts using the Solidity code.
(Singla et al. 2019)	2019, IEEE	Blockchain Smart Contract distributed Leave Management system is developed using Solidity and Ethereum.	Mobile Phones can be used as an IOT device to access data records. The authors have claimed that the application ensures security, and privacy by saving time, and cost.
(Lindsay 2018)	2018, IEEE	Developed a Smart Contract application for Incentivizing. Mobile phones are used as IoT interfaces.	Sensor-Based Smart City Applications on Ethereum developed to handle smart payments. Proof of Concept (POC) functionality is used here to manage the funds, eligibility of receivers, adding receivers, generating a hash key for secure payment management.
(Mori and Miwa 2020)	2020, Springer	Japan e-Portfolio (JeP.), a University admission system using smart contracts is developed. Different tokens like "Study Document Information (SDI)", "e-Portfolio Information (EPI)", "Qualification Information (QI)" are used.	A class of unique token ERC-721 over Ethereum blockchain is used. Security is taken care by using a public key cryptosystem and common key sharing. Data falsification and loss of information are taken care in this application model to enhance security management.
(Bigi et al. 2015)	2015, Springer	It presents the verification and validation of decentralized smart contract applications. Game theory is used to analyze the strategic behavior among smart contract participants.	Validation of data in a Smart Contract is done using game theory-based applications. A probability that 30% of the dishonest seller will leave the system has been empirically shown.
(Cheng et al. 2019)	2019, IEEE	"Ekiden" is a smart contract system that addresses the gap between the blockchain and its supportive Trusted Execution Environment (TEE).	"Ekiden" maintains high scalable performance securing privacy. It is a deterministic stateful program without loss of data. It has been established by the authors that the latency towards implementing an end to end the smart contract in Ekiden and Ekiden BT is much lower than that of the Etherium. Besides the throughput of Ekiden BT is 2 to 4 times higher than the baseline and upto 2 times higher than the Etherium.
(Zheng et al. 2019)	2019, IEEE	"EtherShare" is an application that shares information in a multiple cloud environment of a blockchain-based Smart Contract. The application uses the Gas (Unit of transaction Mesure) in the Etherium environment.	It is a cost-effective application developed using Solidity and JavaScript over public blockchain with the three-layer architectural framework. The EtherShare application shows that the use of gas in a transaction is directly proportionate to the length of information. However, the cost of the transaction is independent of the length of the information of the transactions.

Table 1 (continued)

Paper	*Sources*	*Product/Objectives*	*Comments*
(Nguyen et al. 2019)	2019, IEEE	An application for farmers risk management from cultivation disasters "NEO" is developed for drought based Insurance	NEO is one of the frameworks of smart contract development. It is free in case of used GAS is less than 10 units. The application helps to estimate the crop insurance and expedite the process using smart contracts by securely avoiding bureaucratic documentation procedures. Besides, the demand and supply of water for irrigation can be managed efficiently.
(Pee et al. 2019)	2019, IEEE	A smart energy trading platform using smart contracts developed on the Ethereum platform using ERC20 token, a popular cryptocurrency for Etherium.	This contract is purely used for energy distribution and receiving of payment. The private blockchain network can be maintained in an energy trading system using the ERC20 token.

integrity of the voting system (Tonelli et al. 2019). Smart Contract on distributed leave management system is implemented on Ethereum. The mobile phone is used as an IOT device for accessing data. No additional security features are described (Singla et al. 2019). A smart contract incentive payment system is developed on Ethereum. The mobile phone is used as an IOT device for accessing data. No additional security features are described (Lindsay 2018). The University admission system is implemented over Ethereum. The security features are described with the help of public-key cryptography and common key sharing (Mori and Miwa 2020). A game theory-based mathematical model is used for the Verification &Validation of decentralized Smart Contract applications. This application access the strategic behavior of the smart contract participants following the Bitcoin protocols (Bigi et al. 2015). The smart contract "Ekiden" is addressing the environmental gap among the trusted execution partners in the blockchain. The product is scalable, and it takes care of the privacy issue of security (Cheng et al. 2019). The cloud-based smart contract EtherShare (Zheng et al. 2019) is developed on a public cloud capable of handling multiple clouds and is claimed as a cost-effective system. This system follows the same architecture and security issues that of (Tonelli et al. 2019). NEO is a community-driven open-source platform, which pulls the core advantages of blockchain technology. Here, the smart contract, "NEO" is used for a farmer's risk management system. It is developed to protect the farmers from cultivation disasters primarily due to draughts. It is free if the GAS is <10. Gas measures the amount of work miners need to do in order to include transactions in a block. If a transaction is processed before the limit is reached, the rest of the gas will be returned to the sending wallet (Nguyen et al. 2019). This contract is used for energy trading among renewable electricity grid architecture, consumers and the Internet of Renewable Energy (IoRE). A flexible and distributed I/O performance evaluation tool for hyper-scale Storage systems. IORE was designed to reduce the time to respond in I/O performance experiments and analysis. This application is developed on Ethereum public blockchain using a combination of Solidity code, JavaScript, and different mathematical and statistical algorithms like the Gamblers Ruin problem (Pee et al. 2019).

The above study highlights application-specific smart contracts on the Ethereum blockchain according to the requirement of the need of the user community, in some cases security of the system is explicit and in most cases, it is implicit and application dependent.

The works (Tikhomirov et al. 2018; Covaci et al. 2018; Watanabe et al. 2016; Tonelli et al. 2019; Singla et al. 2019; Lindsay 2018; Mori and Miwa 2020; Bigi et al. 2015; Cheng et al. 2019; Zheng et al. 2019; Nguyen et al. 2019; Pee et al. 2019) discussed above on different application areas use different frameworks. Moreover, all of the cases have given priority to the security of data and transactions. Some of the proposals (Tikhomirov et al. 2018; Covaci et al. 2018) aim to enhance the security of smart contracts by debugging the smart contract code itself using Solidity in Etherium framework. The application proposed in the papers (Watanabe et al. 2016; Bigi et al. 2015) described how Game Theory can be implemented in smart contracts to enhance security and efficiency. The papers (Tonelli et al. 2019; Singla et al. 2019; Lindsay 2018; Mori and Miwa 2020; Pee et al. 2019) used the Solidity code for the development of smart contracts using Etherium framework. The transparency and privacy of data are maintained in the respective domain. Authors in (Zheng et al. 2019; Nguyen et al. 2019) handle the GAS calculations of respective transactions in the Etherium framework to make the Smart Contracts more cost-effective.

In short, the study shows that the smart contracts are good to enhance security as well as operational efficiency in diverse application areas. This has inspired us to design a smart contract of the University exam system with a heterogeneous data structure and implement the system of the Hyperledger open-source cross-industry blockchain technology on Linux.

2.2 Review on model-based smart contracts

A few model-based smart contracts on different domains are presented in Table 2. The security of the blockchain has been considered as the primary objective. The efficiency of the blockchain is also evaluated for these models.

A few smart contract models are tabulated in Table 2, (Kosba et al. 2016; Zhang et al. 2016; Juels et al. 2016). Hawk is a framework for privacy-preserving of data in smart contracts known as Zero-Knowledge Processing (ZKP). The transactional cryptography is the responsibility of the Hawk compiler. The code and data sent to the contract are public, whereas the stock of investment having dodging risk is private in nature. The security protection of a dishonest chain manager's activities is unaddressed. The model is also silent about ZKP's safekeeping landscapes, which heads-up for further investigation towards the enhancement of ZKP security features in this model (Kosba et al. 2016).

Town Crier (TC) is a bridge between smart contracts and existing web sites. TC works on top of Ethereum as the universal composability (UC) framework. Besides hardware security, TC supports security services such as attacks from malicious behaviors of the codes, and privacy-preserving protocols. The model supports customized web applications with real-time query processing. TC feeds the data to the smart contract in the form of datagrams, which is accompanied by specific web site data-sources but did not speak about data security in the web-application (Zhang et al. 2016).

CSM is a smart contract criminal activity detection model. It is an extension of Hawk (Kosba et al. 2016) in the following three specific dimensions of the criminal activities.

1. *Leakage/sale of secret documents (*e.g.*, pre-release Hollywood films)*
2. *Theft of private keys (of,* e.g.*, a Certificate Authority signing key)*
3. *"calling-card" crimes (murder, arson,* etc.*)*

Verification and Validation of the application development entirely depend on the model. This smart contract model indicates that new application areas on Ethereum blockchain may further demand application-specific security improvements (Juels et al. 2016).

2.3 Review on tools designed for smart contracts

A few tools designed for the smart contract in the different domains are described and assessed in Table 3 below.

A number of significant smart contract-based blockchain tools are listed in the tabulation in Table 3, (Wang et al. 2019; Luu et al. 2016; Coblenz et al. 2019; Marino and Juels 2016; Chen et al. 2017; Guida and Daniel 2019). The operation of the smart contract is distributed in six layers and the function of each layer is described. The various legal and technical challenges, mainly the possible security loopholes are also heightened in this paper (Wang et al. 2019). 'Oyente', another smart contract auto-auditing symbolic execution tool. The authors claim that the tool has taken care of DAO bugs and also rectified 4 kinds of potential security bugs like; transaction-ordering dependence, timestamp dependence, mishandled exceptions, and reentrancy in Ethereum (Luu et al. 2016). 'OBSIDIAN' is a refined tool of 'Oyente'. Some more security enhancement features are included. It addresses the bugs in the Solidity codes when the transaction is in an inappropriate state of contract (Coblenz et al. 2019). This tool focuses on the legal side of the smart contract during contract modification, and discontinuation using Solidity on the Ethereum Virtual Machine (Marino and Juels 2016). 'GASPER' is another smart contract tool that addresses only identifying gas-costly patterns by analyzing the smart contracts bytecode on Ethereum 2.0 Beacon Chain, which can spontaneously detect gas costly patterns by scrutinizing smart contracts' for 'dead codes', 'opaque predicates', and 'expensive operations in a loop'. It is an improvement of Ethereum on a gas-efficient byte code. It is used to pay miners fees including transactions in their block. It also provides live monitoring and protection for Solidity smart contracts (Chen et al. 2017). Some more tools are developed for reuse of Smart Contracts through service orientation to reduce coding and debugging efforts on Solidity Ethereum (Guida and Daniel 2019).

The execution of a smart contract (SC) involves two parties either to honor the contract (payer and receiver) or to incorporate the penalties in case of dishonoring the contract by either of the participants of the contract. The entire activities are carried out technically, without the presence of the third party. The smart contract executes over the blockchain environment, where data must be secured and trustful. Technically SC reduces the unnecessary delay due to documentation, and payment procedures. It eliminates the cost of the middleman and other miscellaneous costs.

2.4 Inference of the study

We summarise our observations as well as highlight the technical pitfalls of existing Smart Contract implementations as follows:

1. This study segregates the contemporary works into three different blocks, namely application-specific chains, model-based chains, and tool-based chains. In each block, the research works have tried to look for additional security of the chains, which unnecessarily reduce the processing speed and in its usage.
2. None of these existing works involve the handling of unstructured and semistructured data in the blockchain.

Table 2 Model-based Smart Contracts

Paper	*Sources*	*Product/Objectives*	*Comments*
(Kosba et al. 2016)	2016, IEEE	"Hawk" is a formal smart contract model, where cryptography for the respective functionality is automatically incorporated.	Hawk, a framework for privacy-preserving of data in smart contracts. The performance is evaluated by using a SNARK-friendly implementation in the circuits pour, freeze, compute, and naive. The security level used by the native is 80 bit and the cost incurred in standard cryptographic uses has incurred again equal to the 2 to 2.6 times of native implementations.
(Zhang et al. 2016)	2016, ACM	A formal model satisfying its basic security properties in the Universal Composability (UC) framework with the help of Intel Software Guard Extensions (SGX). It is an Ethereum blockchain-based front-end that supports confidentiality.	Town Crier (TC) is a bridge between smart contracts and existing web sites. The model is validated with three applications: asset digitization (CashSettledPut), flight insurance (FlightIns), and normal trading process (SteamTrade). CashSettledPut shows the best throughput and SteamTrade shows the worst throughput with respect to the number of enclaves on a single machine.
(Juels et al. 2016)	2016, ACM	The Ring of Gyges is a "Criminal Smart Contract (CSM)" model developed to detect criminal activities like murder, arson, etc.	The proposed model is used to maintain law and order in the country due to the immutability and distributed nature of blockchain. The model is tested with three different crime like leakage of secrecy (loss of private information, money, documents, etc.), key theft (loss of private hash key value), and calling card crimes (assassination, assault, murder, sabotage, hijacking, kidnapping, denial-of-service attacks, and terrorist attacks).

3. Most of the blockchain frameworks have their limitations in the customization of the applications. The immutable property must protect the data from tampering. In real-life applications, it is sometimes necessary to edit the data even after proper verification, validation, and authorization.
4. Data replication in the distributed blockchain tends to increase the volume of data in the network. This substantially reduces the blockchain performance. Due to performance reduction, there is a chance of hacking (Cheng et al. 2019).
5. Even though there are different security measures for smart contracts, the transactions and data of smart contracts are always under the threat of cyber terrorism (Kosba et al. 2016).

The technology has its own potential to lower the risk factor of the system, our attempt herein is to use the built-in security of the Hyperledger for access control. The hyperledger also has the flexibility to develop a private blockchain. In case of a university examination system, the privacy, transparency, and security of data must be maintained to render a speedy trustworthy service.

3 Smart contract for university examination system: An use Case

In this section, we have considered a Use-Case with a large, conventional University with several post-graduate departments, and hundreds of affiliated colleges offering undergraduate programs in science, technology, commerce, law, and humanities faculties. We first look at the existing system and note its pitfalls, before presenting our proposal of mitigating some of these using blockchain-based smart contracts.

3.1 Existing data management system and its limitations

We consider a University with about 750 different examinations conducted over 600 examination-centers. The University under consideration is having more than 65 departments and the number of enrolment of students each year is about 1.2 million.

The University deals with different administrative activities like admission, finance, student registration, examination, scholarship processing, etc. The process is intended to deliver the mark sheets, provisional certificates, duplicate certificates, transcripts, migration certificates. Presently the system is entirely manual or semi-automated and so as the security of the system. In the existing system (Fig. 1) users (students, corporate agencies, statistical, and Government agencies) can send a request to the concerned colleges. The college forwards the application to the University. After getting approval from the University, the user is asked to make the payment. The college deposits the fees to the University. The University then takes some time to process and sends the document to the college.

The user has to collect the document from the college. It is a time-bound process.

The high and increasing volume of students and researchers is one of the major challenges of the University to produce error-free results on time with the existing staff strength, which is also getting reduced at regular intervals. The security issues like privacy, authentication, integrity, non-repudiation, and usage of digital signatures. The existing system is always under the security threat from an internal attacker, who may not be authorized for certain things. The storage of data are mainly handled by the database administrator and trusted user(s) at the University level. However, there is a high risk of marks (data) tamparing for a particular examination or re-examination. The marks are received by the university in sealed envelop. If the seal is broken in a undetectable manner and the data is modified, it would raise a question about the credibility of the university system, and would reduce the overall trustworthyness of the services provided by the university. Besides, a University that deals with large volume of students is always on high risk of system fauliure and minimum fault tolerance mechanism with the centralized storage system. The existing centralized data-storage is under the threat of external hacking.

The existing system also suffers from a managerial problem involving human resources, time and expenditure. These data are generally handled by a group of persons specifically assigned. Thus, the absence of any of them can disturb the process. Apart from this, the existing system of data handling is also time taking and is prone to error. Hence, the prevailing process is costly both in terms of time and money. In the existing system, the data is stored in a centralized server with a high risk of system crash and with a high amount of maintenance cost. In order to mitigate the challenges, the limited fund of the University can be exploited efficiently with our proposed blockchain-based secure online solution.

Blockchain would limit the access of the database to the administrators at different clusters only. The objective of the Smart Contract deployment is to establish a common secure data structure to provide uniform reporting irrespective of the user level query either from government, corporate agencies, or specific statistical query. It will reduce the unnecessary documentation time, besides reducing the expenditures on a third party for several operational issues, like the collection of fees, etc. It is not only providing timely solutions but also economical both in terms of infrastructure support and HR point of view. The proposed solution of distributed blockchain network obviously reduces the risk by tracing the point of tampering, if any. The solution also minimizes the risk of single point of server failure. The immutative property of blockchain can resist the tampering of the data in large extent to enhance the quality and trustworthiness of the data.

The large university system that deals with near about 750 number of examinations. All the examinations are not uniformly structured. There is a lot of variation among the number of theoretical and practical papers, courses and modules. There is a variations of the modality of evaluations of the examinations and there final tabulations of result. The implementation of blockchain smart contract can handle the heterogenety of data due to these variations. Therefore, the proposed system can handle different variations in terms of heterogeneous data arguments for the same generic functionality. This requires flexibility of data stroge schema.

3.2 Methodology for the proposed smart contract for university examination

Our proposed system is divided into two parts. In the first part (sec-3.3) we proposed and developed a smart contract to meet our existing challenges. This part contains the contract and the flow of data between the University and different stakeholders to provide a secure, trustworthy, and quality service in the minimum time period. The second part (sec-4) contains the cost analysis and the benefit of the proposed system helps to implement the system.

The user (Fig. 2) directly (students, corporate agencies, statistical, Government agencies, and even college) contact the University for their required document through the website using the valid registration number at their respective category. The necessary document validation, charges for the service, and a payment receipt will be generated. A trigger will be automatically activated to process the smart contract for the respected service. The block will be generated with the serviced transaction. The privacy of the applicants will be maintained by using a private blockchain. The private blockchain is developed using Hyperledger.

3.3 Proposed smart contract algorithm

The Smart contract will work for a generation of mark sheets, certificates, provisional certificates, and the verification demands of records.

Begin

Step 1: The University stores the student data of published results through the software interface into the storage implementing blockchain.

Step 2: The smart contract written in python framework will initiate to generate the detailed terms and conditions.

Step 3: Each term, condition, and requisite fees for the services will be checked by both the user and the University.

Step 4: On acceptance of step-3, the agreements in the form of a smart contract will be executed.

Table 3 Tool designing Smart Contracts

Paper	*Sources*	*Product/Objectives*	*Comments*
(Wang et al. 2019)	2019, IEEE	A research framework for smart contracts based on a novel six-layer architecture is proposed.	Listed technical, and legal challenges along with some typical applications of smart contracts for future references.
(Luu et al. 2016)	2016, ACM SIGS-AC	Investigates the security of running smart contracts based on Ethereum, an open distributed network. The authors claim that the tool has taken care of DAO bugs in Ethereum.	Oyente is a symbolic execution tool. It is developed to find security bugs and counter security attacks. The test evaluation of the implementation of Oyente has reduced the security threats false positive cases is 5.71%, mishandled exceptions is 27.9%.
(Coblenz et al. 2019)	2019, IEEE	Smarter Smart Contract is a tool of smart contract development that defined a tool "OBSIDIAN".	"OBSIDIAN" is a software development tool for the smart contract. This is a refined version of Oyente. This tool is used to develop the smart contract code in domain-specific languages to enhance the security and safety of the data.
(Marino and Juels 2016)	2016, Sprin-ger	Altering and undoing Smart Contracts tool on Ethereum blockchain, which can take care of legal aspects of the smart contract. Code is developed on Solidity, Ethereum.	The primary outcome of this tool is to cancel or modify the smart contract along with the consecutive rights of the involved persons into the contract within the jurisdiction of the law of the respective countries.
(Chen et al. 2017)	2017, IEEE	"GASPER" is a gas-efficient byte code. It is another type of token that is used to pay miners fees including transactions in their block. Ethereum 2.0 Beacon chain is an improvement of Ethereum. It is a new blockchain at the heart of the new Ethereum.	One responsibility of this chain is to allow validators to enter the staking system and build the blockchain instead of the miners. Besides, it stores references for shard states. The implementation of GASPER with 3-pattern gas-costly (Unnecessary code, loop, and additional cost) found that 93.5%, 90.1%, and 80% of the smart contract suffers from this pattern respectively.
(Guida and Daniel 2019)	2019, IEEE	This introduces a model-driven development, which supports the reuse of smart contracts through service orientation. Solidity programming language is used in this work.	The proposed tool provides an interface between the API and internal code, flexibility to reuse the smart contract along with a search engine mechanism for all the engaged persons of the smart contract.

Step 5: Documents will be delivered, SMS will be generated after the receipt of payment.
Step 6: In case of malicious activities, the University has the right to stop the services with necessary legal actions.
Step 7: Exit.

End.

3.4 Experimental setup

The smart contract application for post-examination result data management is developed on Hyperledger Composer a private permissioned blockchain business network tool. The framework for the tool is Hyperledger fabric. Fabric is a Distributed Ledger Technology (DLT), shared among many participants that decentralize its entire network. The experimental setup uses the following hardware and software AMD FX(TM)-8320 Eight-Core processor, 350 GHz with 16GB RAM.

The software to run, Composer and Hyperledger Fabric 2.0, are installed on Ubuntu 16.04 LTS 64 bit Operating Systems. Docker Engine version 19.03.8, Docker-Compose version 1.13.0, Node-js version 8.17.0, npm: v5.x, git: 2.9.x or higher, visual studio code editor version 1.43 and Python 2.7.12 version. Initially, the REST server is installed provided by Hyperledger Composer with a transparent Application Programming Interface (API). This makes it easy to integrate programmatic access to the Blockchain and to connect it to the web or mobile applications. On deployment, smart contract architecture is generated.

The assets need to be transferred among the participants. The communication is done with the help of transactions. In this smart contract, the defined assets are the documents. The participants are the University (service provider) and the service receiver (students, college, Govt Agencies, Corporate Agencies, etc.).

3.5 System generated outputs

All three entities are written in model files (.cto). The code is written in javascript (.js). The access control and validation are done through the (.acl). After the deployment changes in the participant instances of the University Fig. 3a, applicant Fig. 3b, Update Asset Fig. 3c, and the instances of the contract Fig. 3d are created.

Ethereum is a universal, decentralized platform for currency. Coded applications can be built on Ethereum and accessed from anywhere in the world to control the

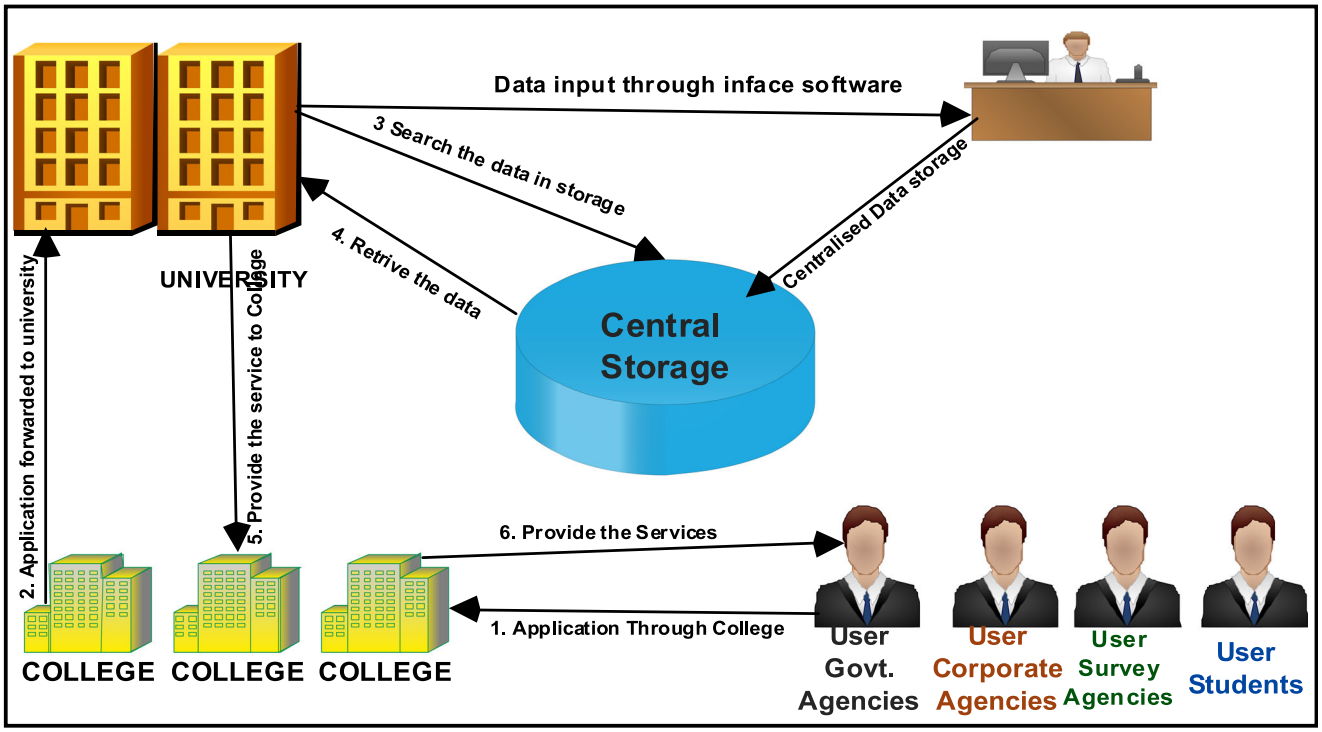

Fig. 1 Existing system of University examination data management

flow of currency. Bitcoin is one of such applications that use blockchain technology to track the proprietorship of digital currency.

The public chain is secured as the data is immutable and accessed by most of the nodes in the chain. The volume and the size of the network in public blockchain become so vast that the efficiency of transactions could be quite poor. In the

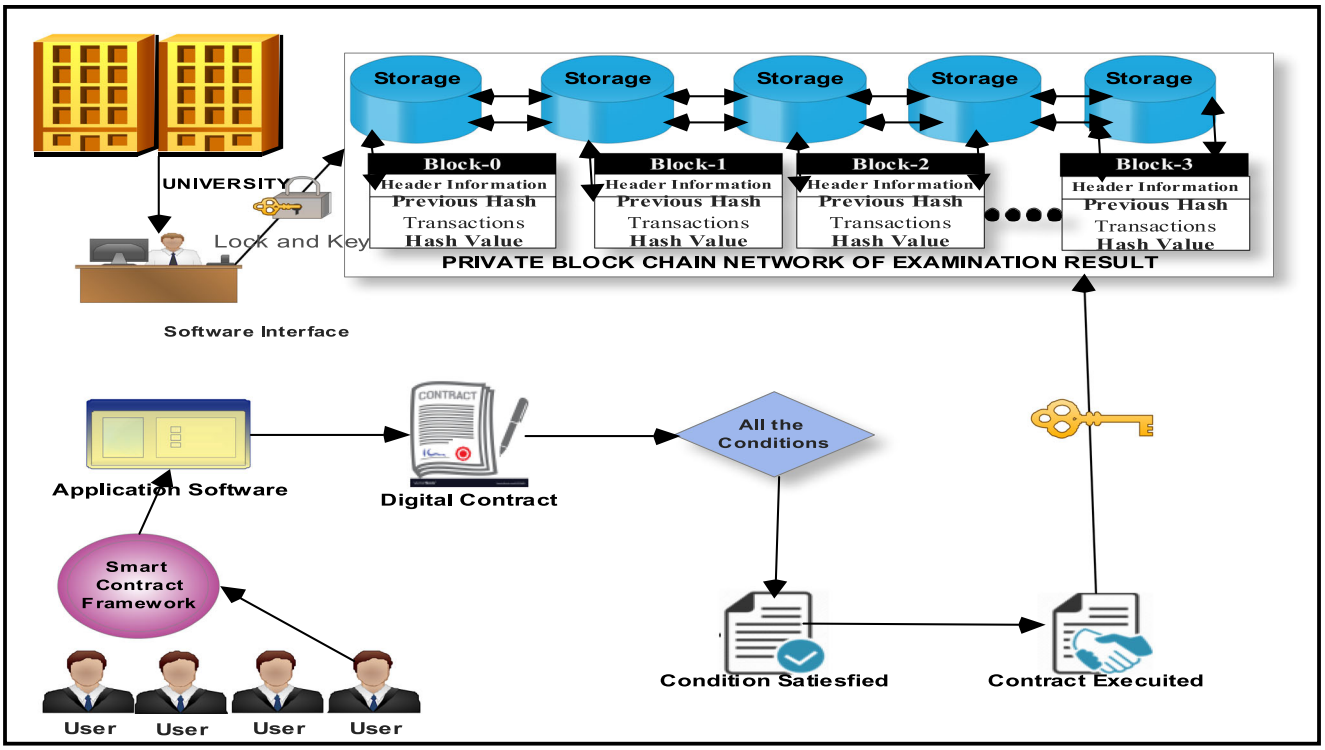

Fig. 2 Smart contract implementation of University examination of data management

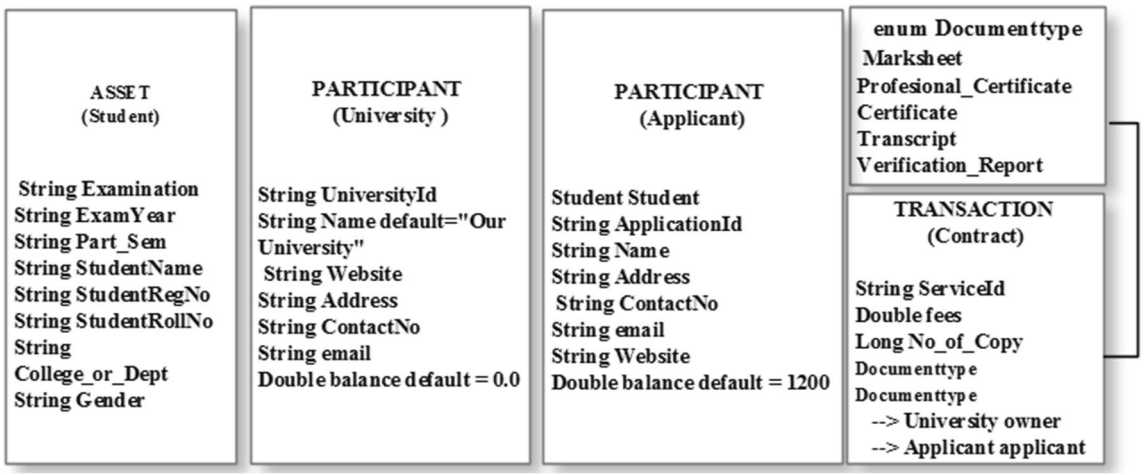

Fig. 3 Design Architecture of University smart contract

case of a private blockchain, efficiency can be increased by restricting access. The available frameworks to develop the Smart Contract are quite limited and yet to be matured. Frameworks used by most of the researchers to develop the Smart Contract are Ethereum, Bitcoin, and NEO.

In the Ethereum framework, Solidity is used to develop a smart contract. There is a constant effort to analyze the code of smart contracts and debugging the code in security, functional, operational, and development areas where the contract was under the threat (Tikhomirov et al. 2018; Luu et al. 2016; Coblenz et al. 2019; Marino and Juels 2016). Various programming tools have been developed to enhance the security, privacy, correctness, and authorization of the data of smart contracts (Covaci et al. 2018; Kosba et al. 2016; Zhang et al. 2016; Chen et al. 2017). The "Gambler's Ruin Problem" and "Noncooperative" game theory have been tried to generate the security model (Watanabe et al. 2016; Bigi et al. 2015) for smart contracts. The "Gambler's Ruin Problem" is used to defend the dishonest miner from monopolizing the resources under the condition of proof of stake. Clarifications are still not quite clear on how the iterative Gambler's Ruin Problem Markova chain fits well to handle the security issues where the dissent minor will fail to write into the node or will succeed to break the security of the smart contract (Watanabe et al. 2016).

Figure 3 a and b Add Participant, c Update Asset, d Transaction.

Finally, the transaction is submitted and added to the smart contract block in the blockchain. The admin can view all the details of the assets, participants, and transactions in the blockchain of the Hyperledger Composer Fig. 4.

The processes will be triggered for service requesters after validation with digital signatures and payment of applicable fees. This mechanism follows according to the Uniform Electronic Transactions Act (UETA) (https://www.stalawfirm.com/en/blogs/view/enforceability-of-smart-contracts-in-india.html, n.d.).

Many real-time applications of the smart contract have been developed to interface with IoT devices along with the web interface (Tonelli et al. 2019; Singla et al. 2019; Mori and Miwa 2020). Besides having several advantages, as the smart contract as a technology is still in the developing stages, it is also subject to different threats. By providing developers with the tools to build decentralized applications, Ethereum is liable for running the programming code (Mori and Miwa 2020).

4 Computation of cost

The computing cost for the hyper ledger is interesting. The term free gas network refers to a network when the gas price is zero. A network with a gas price of zero is also known as a zero gas network or no gas network. The gas cost of the Hyperledger composer is zero.

We consider the standard market price of the hyper ledger provided by Azure Blockchain (Azure Blockchain Service pricing PREVIEW n.d.), a fully-managed, multi-ledger solution. It charges for storage of the data in the ledger as well as associated ledger logs. There is no upfront cost, no termination fees, and pay only for what is used and is presented in Table 4. Here it is considered 1 USD = 80 INR.

The present cost incurred by the University to provide the services, where applications are received online and the service is provided manually are presented in Table 5. The test data is collected from the University administration as a pilot for this development project. The first time service cost of mark sheet and certificates are included in examination fees. The unit rate of stationery and printing cost (in USD) of mark sheet, provisional certificate, transcripts, and verification report is 0.125. The unit rate of stationery and printing cost (in USD) of certificates are 0.2 and 0.125 respectively. The postal cost per unit (in USD) of mark sheet, provisional certificate, and the certificate is 0.50 and the postal cost per unit (in USD)

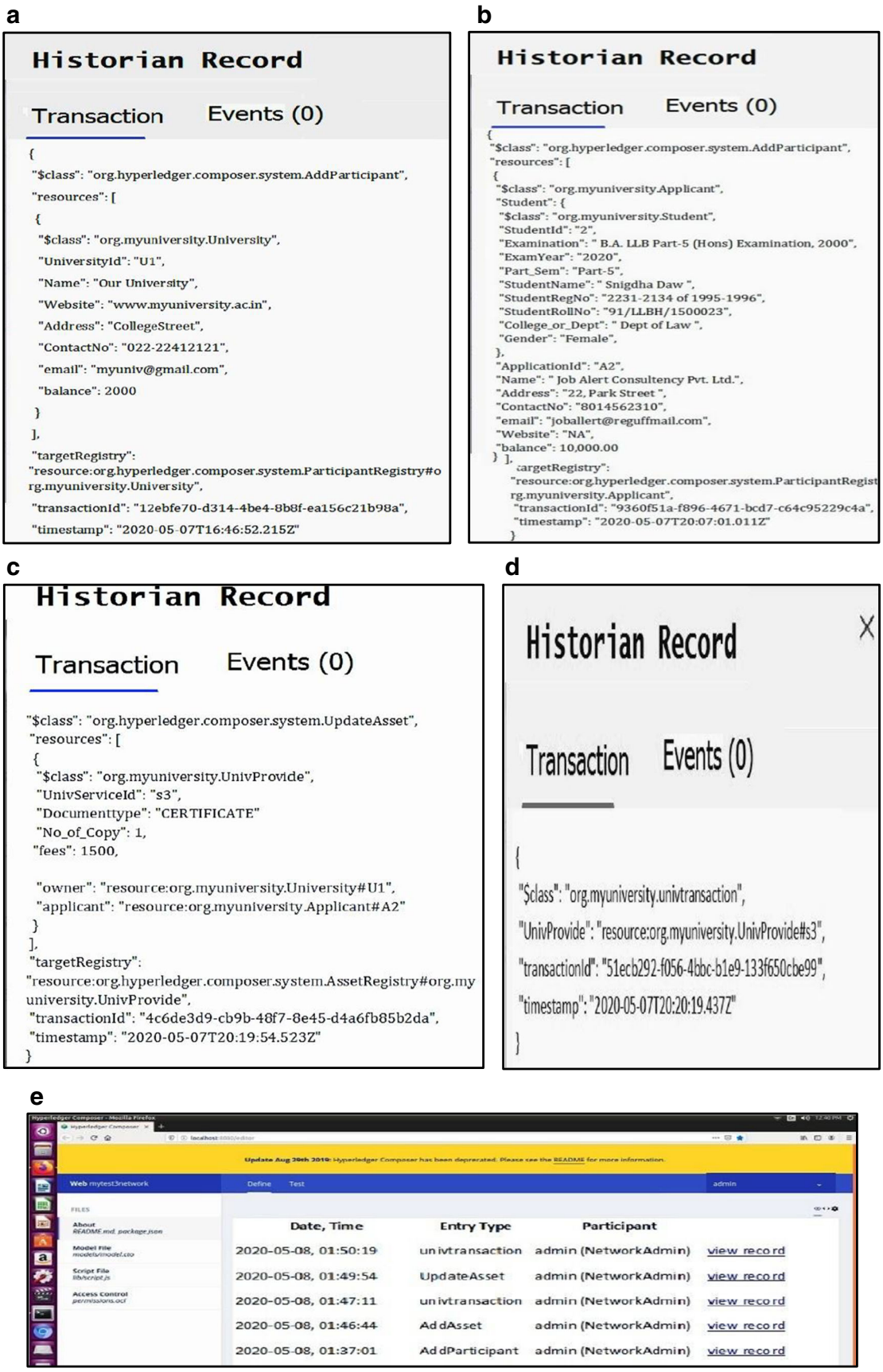

Fig. 4 All Transactions of Hyper-ledger Composer

Table 4 Azure Services.

Sl. No.	*Member*	*Node/ Member*	*Storage Space*	*Node Cost ($)*	*Storage Cost($)*	*Total Cost ($)*
1	2	2	10GB	180.94	1.65	183 (Approx.)
2	1	2	250GB	575.38	30.98	606 (Approx.)
3*	2	2	10GB	180.94	1.65	183 (Approx.)

** Blockchain data manager costs $ 4.96/month in Sl. No.-3 in Table 4*

of transcript and verification report is 0.75. The fees collection rate per unit student for mark sheet, provisional certificate, transcripts, and verification report are 12.5, 10.0, 20.0, 25.0 and 12.5 respectively.

On average, the University yearly provides 2 mark sheets and 1 certificate to 360,000 students. On implementation of secured smart contract, the cost would be.

= $ (2 * 360000 * $ .75 + 1 * 360000 * $ .825).

= $ (540,000 + 297,000).

= $ 837,000.

Plus cost towards node maintenance, system up keeping, storage, and service = $ 125,000.

Thus, the University can save a minimum fixed cost of 0.125 million USD besides saving additional substantial costs due to overhead. The smart contract demands the additional cost of 0.05 million USD for storage only.

A tangible cost is also involved in salary and other payments made to the assigned human resources. This has not been shown explicitly in the cost calculations. Apart from this, the introduction of blockchain would also reduce the intangible cost by reducing the possibility of error, tampering, and thereby making such services more efficient and prompt due to the distributed nature of the network. Besides, this will also reduce the security concerns for data and would increase fault tolerance by eliminating the requirement of centralized system data handling. In the context of the COVID-19 pandemic, it will be particularly helpful in reducing activities that could potentially violet different norms being followed to ensure social distancing.

5 Conclusions

Providing service has emerged as even greater a challenge in this pandemic of COVID-19 throughout the world. Most of the educational institutes are closed or have limited access since March 2020. In this situation, work from home, and teaching-learning processes are practiced using the internet. This situation demands efficient and secure solutions for the dissemination of knowledge in a large, distributed environment. The work presented in this article could indeed be extremely significant in this pandemic of COVID-19 to serve a large community of location-independent students by exploiting the blockchain framework towards developing secured, low-cost digital applications.

In this paper, we have analyzed the various domains of smart contracts and security issues. We have analyzed and established the necessity and advantages of a smart contract with a case study of the University result for data management. The cost analysis is also an effort for future decisions.

In future, we shall try to augment other services into the smart contract to extract more advantages from the system. The development of a generalized cloud-based meta-tool is also one of our feature development plans. However, it is to consider that the future development of smart contracts will not be restricted within the existing frameworks and should support the following:

1. Distribution of the database in different clusters reduces the load of the nodes of the network. It will increase the

Table 5 Existing Cost of Service.

Sl.	*Document*	*Annual (No.) A*	*Stationary cost ($) B= Rate x A*	*Printing Cost ($) C= Rate x A*	*Postal Cost ($) D= Rate x A*	*Total Cost/ Year ($) E= B+ C+ D*	*Total Cost/ month ($) F= E/12*	*Fees Collection ($) G= Rate x A*
1	Mark sheet	11,000	1375	$1375	5500	8250	688	137,500
2	Provisional Certificate	35,000	4375	4375	17,500	26,250	2188	350,000
3	Certificate	15,000	3000	1875	7500	12,375	1031	300,000
4	Transcript	30,000	3750	3750	22,500	30,000	2500	750,000
5	Verification Report	40,000	5000	5000	30,000	40,000	3333	500,000
6	*TOTAL*	*131,000*	*17,500*	*16,375*	*83,000*	*116,875*	*9740*	*2,037,500*

efficiency and scalability of the entire blockchain network.

2. With due authorization, authentication, validation, and verification the blockchain should be editable. The data of the participants and the assets of private blockchain can be edited. Private blockchains allow some editing of data till a transaction is committed. While it allows some flexibility, the fundamental trustworthiness of blockchains in terms of tracebility of transactions remains uncompromised.
3. We plan to enhance in a way such that the blockchain should support a sizable quantity of structured and semi-structured data. However, this itself would require a considerable amount of effort and has to handle several research issues.
4. We like to enhance the security of the chain by incorporating a game theory-based approach to differentiate the honest and dishonest participants and filter.

References

Tikhomirov S, Voskresenskaya E et al (2018) SmartCheck: static analysis of Ethereum smart contracts, 1st international workshop on emerging trends in software engineering for Blockchain (WETSEB), Gothenburg, Sweden, pp. 9–16

Covaci A, Madeo S et al (2018) NECTAR: non-interactive smart contract protocol using Blockchain technology. 1st international workshop on emerging trends in software engineering for Blockchain (WETSEB), Gothenburg, Sweden, pp. 17-24

Watanabe H, Fujimura S et al (2016) Blockchain contract: securing a blockchain applied to smart contracts. IEEE international conference on consumer electronics (ICCE), Las Vegas, pp. 467-468. DOI: https://doi.org/10.1109/ICCE.2016.7430693

Tonelli R, Lunesu MI et al (2019) Implementing a microservices system with Blockchain smart contracts. International workshop on Blockchain oriented software engineering (IWBOSE), Hangzhou, China, pp. 22-31. DOI: https://doi.org/10.1109/IWBOSE.2019.8666520

Singla V, Malav IK et al (2019) Develop leave application using Blockchain smart contract. 11th international conference on communication systems & networks (COMSNETS), Bengaluru, India, pp. 547-549. DOI: https://doi.org/10.1109/COMSNETS.2019.8711422

Lindsay J (2018) Smart Contracts for Incentivizing Sensor Based Mobile Smart City Applications. International Smart Cities Conference (ISC2), Kansas City, MO, USA, pp. 1–4. DOI: https://doi.org/10.1109/ISC2.2018.8656959

Mori K, Miwa H (2020) Digital University admission application system with study documents using smart contracts on Blockchain. The 11th International Conference on Intelligent Networking and Collaborative Systems 1035:172–180. https://doi.org/10.1007/978-3-030-29035-1

Bigi G, Bracciali A et al (2015) Validation of decentralized smart contracts through game theory and formal methods. In Programming Languages with Applications to Biology and Security 9465:142–161. https://doi.org/10.1007/978-3-319-25527-9_11

Cheng R, Zhang F et al (2019) Ekiden: a platform for confidentiality-preserving, trustworthy, and performant smart contracts. European symposium on security and privacy (EuroS&P), Stockholm, Sweden, pp. 185-200. DOI: https://doi.org/10.1109/EuroSP.2019.00023

Zheng P, Zheng Z et al (2019) EtherShare: Share Information in Joint Cloud Environment Using Blockchain-Based Smart Contracts. International Conference on Service-Oriented System Engineering (SOSE), San Francisco, pp. 233–2335. DOI: https://doi.org/10.1109/SOSE.2019.00040

Nguyen TQ, Das AK, Tran LT et al (2019) NEO smart contract for drought-based insurance. Canadian conference of electrical and computer engineering (CCECE), Edmonton, AB, Canada, pp. 1-4. DOI: https://doi.org/10.1109/CCECE.2019.8861573

Pee SJ, Kang ES et al (2019) Blockchain based smart energy trading platform using smart contract. International conference on artificial intelligence in information and communication (ICAIIC), Okinawa, Japan, pp. 322-325. DOI: https://doi.org/10.1109/ICAIIC.2019.8668978

Kosba A, Miller A et al (2016) Hawk: the Blockchain model of cryptography and privacy-preserving smart contracts. Symposium on security and privacy (SP), San Jose, pp. 839-858. DOI: https://doi.org/10.1109/SP.2016.55

Zhang F, Cecchetti E et al (2016) Town crier: an authenticated data feed for smart contracts. The ACM SIGSAC conference on computer and communications security, pp. 270-282. https://doi.org/10.1145/2976749.2978326

Juels A, Kosba A et al (2016) The ring of gyges: investigating the future of criminal smart contracts. The ACM SIGSAC Conference on Computer and Communications Security, pp:283–295. https://doi.org/10.1145/2976749.2978362

Wang S, Ouyang L, Yuan Y, Ni X, Han X, Wang FY (2019) Blockchain-enabled smart contracts: architecture, applications, and future trends. Transactions on Systems, Man, and Cybernetics: Systems 49(11): 2266–2277. https://doi.org/10.1109/TSMC.2019.2895123

Luu L, Chu D et al (2016) Making smart contracts smarter. The ACM SIGSAC Conference on Computer and Communications Security, pp.:254–269. https://doi.org/10.1145/2976749.2978309

Coblenz M, Sunshine J et al (2019) Smarter smart contract development tools. The IEEE/ACM 2nd international workshop on emerging trends in software engineering for Blockchain (WETSEB), Montreal, QC, Canada, pp. 48-51. Doi: https://doi.org/10.1109/WETSEB.2019.00013

Marino B, Juels A (2016) Setting standards for altering and undoing smart contracts. In: Alferes J, Bertossi L, Governatori G, Fodor P, Roman D (eds) Rule Technologies Research, Tools, and Applications, vol. 9718, pp. 151–166. https://doi.org/10.1007/978-3-319-42019-6_10

Chen T, Li X et al (2017) Under-optimized smart contracts devour your money, the 24th international conference on software analysis, evolution and reengineering (SANER), pp. 442-446. DOI: https://doi.org/10.1109/SANER.2017.7884650

Guida L and Daniel F (2019) Supporting Reuse of Smart Contracts through Service Orientation and Assisted Developme. International Conference on Decentralized Applications and Infrastructures (DAPPCON), Newark, CA, pp. 59–68. DOI: https://doi.org/10.1109/DAPPCON.2019.00017

"Azure Blockchain Service pricing PREVIEW" (n.d.), https://azure.microsoft.com/en-in/pricing/details/blockchain-service/. Last accessed on: June-04-2020

https://www.stalawfirm.com/en/blogs/view/enforceability-of-smart-contracts-in-india.html. (n.d.) Last accessed on:September-04-2020