

Towards partial fulfillment for Undergraduate Degree Level Programme
Bachelor of Technology in Computer Engineering

A Project Preliminary Evaluation Report on:

Decentralized Domain Name Broker Service

Prepared by :

Admission No.

Student Name

_U17CO094_____Hrishabh Sharma_____

_U17CO089_____Ujjwal Kumar_____

_U17CO003_____Amruta Mulay_____

_U17CO052_____Rishabh Kumar_____

Class : B.TECH. IV (Computer Engineering) 7th Semester

Year : 2020-2021

Guided by : Dr. Sankita J. Patel_____



DEPARTMENT OF COMPUTER ENGINEERING
SARDAR VALLABHBHAI NATIONAL INSTITUTE OF TECHNOLOGY,
SURAT - 395 007 (GUJARAT, INDIA)





Student Declaration

This is to certify that the work described in this project report has been actually carried out and implemented by our project team consisting of

Sr.	Admission No.	Student Name
1	U17CO094	Hrishabh Sharma
2	U17CO089	Ujjwal Kumar
3	U17CO003	Amruta Mulay
4	U17CO052	Rishabh Kumar

Neither the source code there in, nor the content of the project report have been copied or downloaded from any other source. We understand that our result grades would be revoked if later it is found to be so.

Signature of the Students:

Sr.	Student Name	Signature of the Student
1	Hrishabh Sharma	
2	Ujjwal Kumar	
3	Amruta Mulay	
4	Rishabh Kumar	

Certificate

This is to certify that the project report entitled Decentralized Domain Name
Broker Service *is prepared and presented by*

Sr.	Admission No.	Student Name
1	U17CO094	Hrishabh Sharma
2	U17CO089	Ujjwal Kumar
3	U17CO003	Amruta Mulay
4	U17CO052	Rishabh Kumar

Final Year of Computer Engineering and their work is satisfactory.

SIGNATURE :

GUIDE

JURY

HEAD OF DEPARTMENT

Abstract

In this growing age of the internet, people want to have a domain name that is relevant to their work. In many cases, it might not be possible to purchase a preoccupied domain name, and they might want to buy it from someone through the secondary market and brokers. The existing solutions for the secondary market of domain names are centralised in nature and demand huge commission-fee. We aim to provide a platform that is secure, decentralized, and cheaper than the currently available options. The project focuses on a growing secondary market of domains and entities involved with legacy DNS architecture but shares nothing with Blockchain-based DNS architecture. With that said, the idea is to implement a Blockchain-based service on top of the legacy system to allow Domain owners (potential sellers) and their Buyers to interact directly without any middle-man in between the two parties involved.

Keywords: *Domain name transfer - Blockchain - Decentralised model.*

Contents

Abstract	i
List of Figures	iv
List of Tables	v
List of Acronyms	vi
1 Introduction	1
1.1 Applications	1
1.2 Motivation	1
1.3 Objectives	2
1.4 Organization of project report	3
2 Literature Survey	4
2.1 Domain Names	4
2.2 DNS related terminologies	5
2.3 Domain Name Transfer	5
2.4 Domain ownership verification	6
2.5 Accessing Registry Data	6
2.6 Blockchain	7
2.6.1 Advantages of Blockchain	8
2.6.2 Types of Blockchain	8
2.6.3 Smart Contract	8
2.6.4 Consensus Protocols	8
2.6.5 Comparison of Public Blockchain Networks	9
2.7 Detailed Comparison between Ethereum and EOS	10
2.8 Smart contrac	11
2.8.1 Major pitfalls in Smart Contracts	11
2.9 Current System	12
2.9.1 Domain Brokers	12
2.9.2 Domain transfer from one Registrant to another	12
2.9.3 A typical (centralised) escrow service	13
2.9.4 Problems with Centralised Approach	14
3 Related Work	15
4 Proposed Work	16
4.1 Blockchain based approach	16
4.1.1 Advantages	16
4.1.2 Challenges	16
4.1.3 Proposed Implementation	16
4.1.4 Value proposition of Domain Names using Auction	17

4.1.5	Components	17
4.1.6	Detailed Procedure	19
4.1.7	Actors' involvement	21
5	Implementation	22
5.1	Selection of Tools	22
5.2	RDAP	22
5.3	Auction algorithm	25
5.4	Smart Contracts	25
5.4.1	Main Contract - DomainMarket.sol	25
5.4.2	Auction Contract - DomainAuction.sol	28
6	System Analysis	30
6.1	Security Analysis	30
6.1.1	Attacks on decentralised components	30
6.2	Limitations of Blockchain	31
6.3	Challenges	32
7	Conclusion	33
	References	36
	Acknowledgement	37

List of Figures

1	Rise of Domain names [12]	2
2	An illustration for tree-structure domain name governance	4
3	Illustration of Blockchain [9]	7
4	Broker's role in a centralised environment	13
5	Auction Activity	18
6	Blockchain-based approach to transfer domain names	20
7	Involvement of various actors in the system	21
8	Response of API End Point	24

List of Tables

1	Comparison of Public Blockchain Networks	10
---	--	----

List of Acronyms

SEO	Search Engine Optimization
DNS	Domain Name System
URL	Uniform Resource Locator
TLD	Top Level Domain
gTLD	Generic Top Level Domain
ccTLD	Country code Top Level Domain
CA	Certificate Authority
PKI	Public Key Infrastructure
TPS	Transactions Per Second
OTP	One Time Password
ICANN	The Internet Corporation for Assigned Names and Numbers
RDAP	Registration Data Access Protocol
GDPR	General Data Protection Regulation
HTTPS	Hypertext Transfer Protocol
PBFT	Practical Byzantine Fault Tolerance
RBFT	Redundant Byzantine Fault Tolerance
PoW	Proof of Work
PoS	Proof of Stake
DPoS	Delegated Proof of Stake
P2P	Peer to Peer
EVM	Ethereum Virtual Machine
AFNIC	Association Française Pour le Nommage Internet en Coopération
DDoS	Distributed Denial of Service
BGP	Border Gateway Protocol
DAO	Decentralized Autonomous Organization

1 Introduction

With the advent of Blockchain Technology, the world is shifting towards decentralisation. The internet community is thriving to shift towards decentralised architecture, where users can have true control on their data. Attempts are being made to remove the brokers and centralised entities present in various use-cases. One of such use-case, which haven't got much attention in years, but required to be hold up is the-Secondary market of Domain Names.

In the current market of Domain Names, the domain brokers provide the service of buying/selling domain names. Domain name brokers are centralized entities, who act like a middle-men in exchanging the ownership of domains and the value associated with it between the buyer and seller. These brokers charge a commission to provide the service to both the parties. At the end of the process, the buyer gets the domain name access and the agreed price is settled with the seller.

Our goal is to provide a service based on blockchain to make the process transparent and reduce the commission charges to minimal. In the proposed service there is no scope of scams [detailed here] which is a very big concern with the current scenario of the market. The service (if economically feasible) has the potential to create a competitive marketplace for domain selling and completely replace other domain escrow services provided by the domain name registrars themselves.

1.1 Applications

A meaningful and appropriate domain name can increase the traffic on the website by improving the SEO (Search Engine Optimization) ranking. People might want to sell their domain names which are no longer active, and thus it will help them also get some amount of monetary benefits. The model aims to connect potential buyers and sellers to transfer domain name ownership.

The application scope of the model is not limited to only companies or organizations, but it can be used by almost everyone on the internet.

1.2 Motivation

The exponential growth of the secondary market of domain that motivates us to work on this idea are :-

- With the exponential growth of internet users, domain ownership has grown exponentially too. Thus, creating a huge demand for domain names.

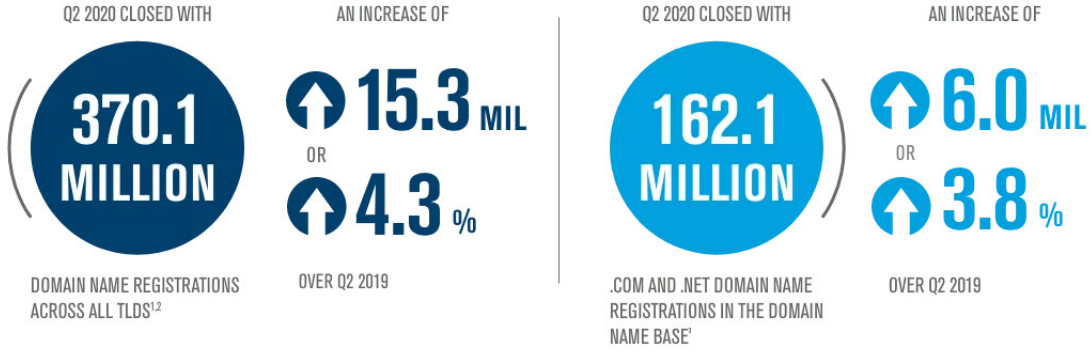


Figure 1: Rise of Domain names [12]

- Different categories of domainers (individuals, organizations, etc.) have realized the value of attractive and unique domain names, thus creating a competitive secondary marketplace for domain names.
- There lies a paucity of good and appealing domain names. It even becomes infrequent for these domain names to return to the public domain. Even if it turns out to be so, it is regarded as a casualty or carelessness from the registrant's side.
- With the growth of the secondary market, it has also marked the advent of mediators or negotiators who play a vital role as third parties building favorable situations for transactions. The third parties also stand a chance to scam the buyer/seller involved to make extra profit.

1.3 Objectives

The proposed work focuses on a growing secondary market of domains and entities involved with legacy Domain Name System (DNS) architecture and implements a Blockchain based service on top of the legacy system to allow domain owners (potential sellers) and their Buyers to interact directly without any middle-man in between the two parties involved.

The aim is to make the process reliable, transparent, and reduce the commission charges to minimal thereby offering hassle-free service to domain buyers/sellers. In the proposed service, there is no scope of scams which is a very big concern with the current scenario of the market. The proposed service is economically feasible and has the potential to create a competitive marketplace for domain selling. The proposed system is transparent, reliable, cost-effective, and has the prospective to completely replace other domain escrow services provided by the domain name registrars themselves.

1.4 Organization of project report

Chapter 1 gives us a brief introduction towards the topic highlighting the objectives, applications and the motivation that encouraged to choose this project. Chapter 2 contains the literature survey concerned with the concepts and terms used in the work. It also provides a theoretical background of all the topics covered to help us get an in-depth understanding towards the subject. Chapter 3 discusses related work done in this field. Chapter 4 contains a detailed report about the proposed implementation methods needed to carry out the project. Chapter 5 gives us an insight towards the different tools that have been used along with the technologies executed in the project. We get a proper understanding of the steps taken to practically employ the theory that we have come across in the previous chapters. Chapter 6 includes the various security attacks and the limitations of the project being implemented. It even highlights the challenges that are faced in the current model and future prospects of resolving those limitations. Chapter 7 marks as the conclusion of the entire report.

2 Literature Survey

This chapter covers the theoretical background needed to understand this project report. It includes a detailed explanation of all the terms of a blockchain-based system and it also highlights the terms related to the domain name and its transfer.

2.1 Domain Names

Domain names are essentially a combination of letters, digits and hyphen that are generally chosen so as to convey some meaning, acronyms or even brand names. A domain name forms the base of Uniform Resource Locator (URL) hence a domain name can also be seen as a component of URL. Further, there are different parts to a domain name but for the sake of convenience, we will only look at the top two levels of it.

A domain name consists of different parts which are called labels; and are concatenated by dots for example google.com. From the rightmost side, the first part is called top-level domain (TLD); in google.com the TLD (top-level domain) is 'com'. The second part/label in the domain name is called second-level domain; in google.com the second-level domain is 'google'. The parts of domain name from left to right represent the hierarchical structure of DNS; where each label in the left signifies further subdivision. A TLD has further two categorisation namely generic top-level domain (gTLD) and country code top-level domain (ccTLD). Initially there were total of 7 gTLDs at the time of DNS designing which now has crossed the mark of 1200 gTLDs as of 2018 and 300+ ccTLDs.

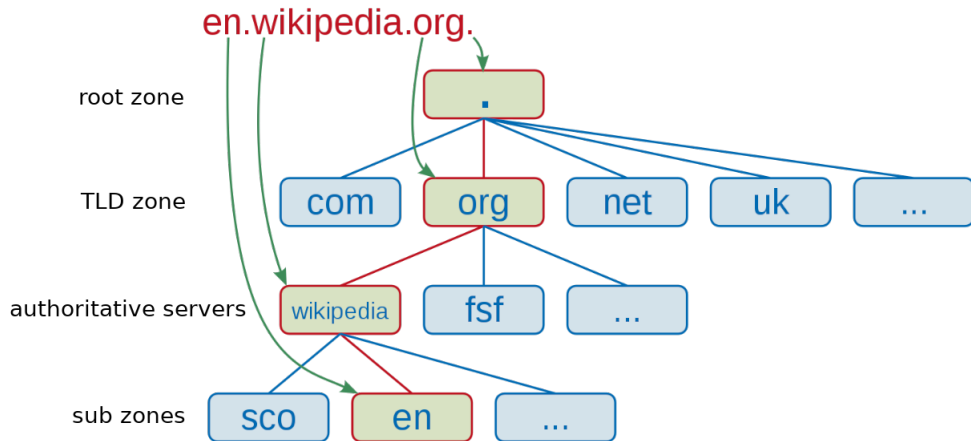


Figure 2: An illustration for tree-structure domain name governance

2.2 DNS related terminologies

Referred from: [3]

TLD: Acronym for top-level domain. Examples of TLD include com, org, uk, us, in.

gTLD: Acronym for generic top-level domain which is a category of TLD. Examples include com, org, net, gov.

ccTLD: Acronym for country code top-level domain which is the second categorisation of TLD. It is primarily a two-character territory codes of ISO-3166 country abbreviations.

DNS: It is an acronym for Domain Name System, which is a service on the Internet that is responsible for maintaining a mapping of two primary naming system on the Internet viz. Domain name and IP addresses. The system is also responsible for replying to queries for resolving domain names to their corresponding IP addresses.

CA: Acronym for Certificate Authorities; are trusted entities in the centralised and hierarchical PKI (Public Key Infrastructure) which are responsible for approving other certificates on the Internet.

Registry Operator: Registry Operators are entities that manage the registry (database) of a particular TLD (categorised as gTLD and ccTLD) such as VeriSign which operates the .com and .net (both are gTLDs) registries.

2.3 Domain Name Transfer

- **Registrar or Domain Name Registrar**

Registrar or Domain Name Registrar is a company that manages the reservation of Internet domain names.

- **Registrant**

A registrant is an entity owning a registered domain name.

- **Buyer**

Buyer in this document refers to an entity who is interested in buying a domain name that is currently owned by a different entity.

- **Seller**

Seller refers to an entity that currently owns a particular domain name and is interested in selling that domain name to some other entity.

- **Domain Broker Service**

Domain Broker Service refers to the service provided by various entities for domain ownership transfer between two interested parties. These are the entities that are

in business for their own profit and act as a mediator between buyers and sellers. An entity providing this service is called Domain Broker.

2.4 Domain ownership verification

When a registrant buys a domain name, it is mandatory to provide required contact details (that include name, postal address, contact number, email address). This information is stored by the registrar which is shared with the registry holder for the gTLD. Every gTLD database is managed by an organisation called registry holder which gets information from the registrars.

Verifying domain ownership through email address is acceptable because it's the primary mode of communication to the registrant. Also, we can easily verify an email address using One-Time-Password (OTP) based methods.

2.5 Accessing Registry Data

The registrant data (information shared at the time of registration) is stored with the registrar and shared with the registry. Till May 2018, one could find the contact information (name, email address, contact number, postal address) associated with a domain name, using WHOIS [14] protocol service. [15] This was possible because ICANN made it mandatory for the registrars and registry to implement this public service (WHOIS) for the sake of identifying and contacting the owner of the domain. But this service has now been modified to bring it inline with the General Data Protection Regulation (GDPR) [19] policies (enforced in May 2018). So, WHOIS at its present state does not provide the contact details of the registrants (postal address, email address, contact number) hence we cannot use WHOIS for our verification process.

But WHOIS has now been accompanied with another service, namely Registration Data Access Protocol (RDAP) [16] which provides many features over the previous WHOIS protocol. RDAP is a successor to WHOIS which provides access to information about Internet resources (domain names, IP addresses, and autonomous systems).

Unlike WHOIS, RDAP provides:

- A machine-readable representation of data
- Secure access to data
 - Over Hypertext Transfer Protocol Secure (HTTPS)
- Differentiated access

- Limited access for anonymous users
- Full access for authenticated users
- Standardized query, response and error messages
- Internationalisation
- Extensibility
 - Easy to add output elements

Since RDAP provides differentiated access, one can query RDAP service either anonymously or with some authentication. The response to anonymous queries contain redacted information (email addresses are redacted; along with other information). So for our use case, we need to make authenticated queries to the RDAP service of respective registry or registrar to verify the domain ownership via email address verification.

To get an authenticated account (for RDAP access), registrars provide an application form (similar to [20]) for access request and upon successful application, the credentials are shared and authenticated queries will get full access to the required information.

Many RDAP clients have been implemented and some of them have already been deployed such as [17] and [18] which can be used to send anonymous queries.

2.6 Blockchain

Blockchain is a distributed database in which data is store in the form of blocks and each block (except the first block, known as genesis block) contains the cryptographic hash of the previous block, thus giving it a chain like structure. The linked structure makes the Blockchain an immutable data structure. The another important property of the Blockchain Network architecture is the 'Decentralisation'. However the degree of decentralisation varies from one blockchain platform to other, due to obvious trade-offs among characteristics of blockchain. [4]

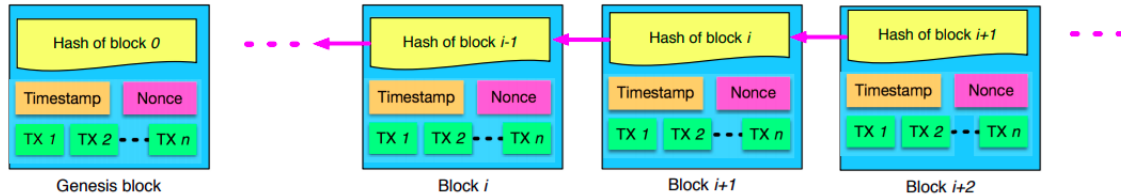


Figure 3: Illustration of Blockchain [9]

2.6.1 Advantages of Blockchain

- The main aspect of blockchain is that it is a decentralised system. This eliminates the central server. The decentralised system distributes the copy of the database to each of its participants on the network, thus making it almost infeasible to hack into the system. (except for the “attack 51%”)
- Every participant on the blockchain network holds the information of complete chain of blocks in encrypted form. These distributed copies are continuously and periodically synchronized.
- The beauty of blockchain is the transparency that it provides in the system which makes almost any operation to be easily traceable.
- Each time a new information is added into the database, it gets added in the form of new blocks. This new data is collectively agreed upon by each of the other registered nodes on the network, thus becoming a part of the chain permanently.
- Ubiquitous Nature - the technology is relevant not only in financial sector, but also in other fields.

2.6.2 Types of Blockchain

- **Public:** Public Blockchain Networks are open & anyone can join the network any-time and leave without any permission. In public blockchains, there is no message passing between the participants for reaching the consensus. So, a single person is elected in some way, and incentivized, to decide the next state, and other participants verify. Therefore, a monetary asset is mostly associated with such blockchains.
- **Private:** In private blockchains, users are able to join or leave if they are authorised to do so. In private networks consensus can be achieved through various classical algorithms, hence no monetary currency is attached to them.

2.6.3 Smart Contract

Smart contracts are basically pieces of code stored inside a blockchain which serves as a type of agreement. It gets executed when certain pre-written conditions are met. Smart contracts eliminates the risk of any fraudulent activity from the participants.

2.6.4 Consensus Protocols

In a Blockchain Network, consensus protocols are a way through which distrusted nodes or participants reach to a single agreement about the next block to be added in the

Blockchain.

In the case of permissioned and private blockchain networks, participating nodes have a valid identities issued by authorities like CAs and are known to each other. Such blockchain networks use classical consensus protocols like Paxos, Raft, PBFT, RBFT, etc to reach agreement. On the other hand, the participating nodes are anonymous to each other, in case of public blockchain networks. So, in order to reach an agreement, a leader election takes place. The leader election can be done in many ways, and that leads to various existing consensus protocols for such network. The elected leader will be the one to propose a decision for the next block and rest of the participating nodes validate the new proposed block. Some of the consensus protocols involving leader election, in the scope of our project are discussed below:

- **PoW**: Proof of Work (PoW) involves a leader election on the basis of solving a complex mathematical problem directly related to hashing. The selected problem is hard to solve but can be easily verified by the rest of the nodes of the network. A high computation power is required to solve the complex problem first, so the probability of getting elected as leader is directly dependent on the computation power. Bitcoin [7] and Ethereum [8] Network uses PoW to reach consensus. However, Ethereum is constantly making a move to shift towards PoS. [6]
- **PoS**: Proof of Stake (PoS) involves a leader election, in which the probability of being elected directly depends upon the amount of stake(can be the amount of cryptocurrency) the node/participant holds.
- **DPoS**: Delegated Proof of Stake (DPoS) [5] protocol is a variation of PoS mechanism. In the procedure involved, a fixed number of delegates are voted by the network participants. The delegates, also known as "Witnesses" decides the next block.

2.6.5 Comparison of Public Blockchain Networks

1. **Bitcoin** - It is a peer to peer(P2P) electronic cash system, backed on a widely spread Bitcoin Network. All the transactions are made in terms of its cryptocurrency transfer viz. Bitcoin. It ensures the classical problem of double spending using a peer to peer network and using a powerful consensus algorithm 'Proof of Work'. It uses special scripts called 'Bitcoin Scripts' to govern the transactional activities like sending bitcoins, claiming bitcoins, etc.
2. **Ethereum** - It is an open-source public blockchain network, which along with a cryptocurrency associated with it, also is able to carry other assets on its backbone blockchain. The state changes of Ethereum can be easily governed by the

Property	Bitcoin	Ethereum	EOS
Consensus	PoW	PoW	dPoS
Transaction fee	Yes	Yes	No
Smart Contract	No	Yes	Yes
TPS	7-8	15-20	50000
Decentralisation level	High	High	Low

Table 1: Comparison of Public Blockchain Networks

Smart Contracts which are executed on Ethereum Virtual Machine(EVM). The transaction on the Ethereum Transactions requires a fee, called Gas in order to be successfully executed. The higher the gas amount a person is willing to pay for a transaction, the more early his/her transaction is executed on the Ethereum Network.

3. **EOS** - It is a public blockchain network, which allows building decentralised applications on the top of it. It also uses the Smart Contracts to govern the transaction. But it differs from the existing public blockchain platforms, in a way that the transactions on EOS are completely free. It uses a hybrid version of Proof of Stake, called Delegated Proof of Stake(DPoS) to eliminate the transactional fee in the network. But this comes at the cost of a little amount of decentralisation.

2.7 Detailed Comparison between Ethereum and EOS

Ethereum Platform introduced the concept of smart contract and introduced the world to the decentralised applications, called dApps. Soon after the launch ethereum started getting more and more attention from the developer community. But the main concern of the Ethereum dApp users is the gas price required to change the state of smart contract. To overcome the concern of the users, EOS came into the existence, claiming itself to be the dApp suitable platform, in which instead of users, developers have to pay. So there is no transaction fee in EOS platform. They avoid the user transaction fee by claiming the cost through inflation.

Ethereum uses the mining based consensus protocol, called Proof of Work. Due to the competition and time require to solve the challenge, the transaction throughput of Ethereum is low. Thus, at present stage Ethereum suffers from the scalability factors. On the other hand, the decision of the next block is in the hand of 21 Block Producers (BPs), which are elected though staking. This less participation of nodes in the consensus process makes the entire process faster, giving EOS a higher transaction through put.

EOS always justify its existence by the fact that it is more decentralised, by expressing the probability of shift of governance into few pool mines in the case of proof of work

based blockchain platforms. But its own governance lying in the hands of 21 BPs, shows that EOS is preferring throughput over the decentralization level in the trade off involved.

2.8 Smart contract

As explained above, smart contracts are piece of code that contain the business logic for managing fund transfer and other blockchain related transactions. These codes, once deployed, cannot be updated to apply any security patches. Since, they are written in a programming language, it is quite obvious to introduce bugs while writing a smart contract and these bugs can lead to a loss of assets from their rightful owners.

2.8.1 Major pitfalls in Smart Contracts

- **Reentrancy** - It is the type of smart contract vulnerability which occurs due to unintended recursive call within the smart contract by the attacker. [23] Suppose there is a crowdsourcing smart contract 'A', having a simple deposit and withdraw functions. If there is presence of reentrancy in A, then a fallback function from another attacking smart contract 'B' can make recursive calls to the withdraw function of 'A', and withdraw all the crowdsourced funds.
Such attacks can be prevented by ensuring that all the changes in state occurs before calling an external contract. Use of functional modifiers to lock the state of smart contract for preventing the recursive call, when the smart contract is already being executed, is also one of the suggested good practice to avoid 'Reentrancy'
- **Overflow and Underflow** - The variable data-types in the languages concerned with smart contracts have definite maximum size to store the values. The maximum and minimum value are generally cyclic in nature. If the value of certain variable is increased more than the maximum, then it will end up resulting a number having lesser value. This is called overflow. Contrary, if a value when decreased ends up resulting a greater value, then this is called underflow. Such type of attacks are common while writing smart contracts. Due to the underflow and overflow, the attacker can exploit the smart contract by giving invalid inputs. Such attacks can easily be handled by putting checks on the inputs accepted.
- **Short Address attack** - This attack is more on the user-interface level rather than the smart contract level. It occurs when the user input invalid address and subsequently the Smart Contract Engine executes it after padding.
- **Delegate Call** - It is a function with a slight difference with the normal CALL method. The DELEGATE CALL are always executed in the context of the caller environment. The primary use of this type of call is to make upgradable smart contracts. But these advantages can also bring serious vulnerability. If the functional

signature of the delegate call and the the caller's contract function doesn't match, then the execution jumps to fallback, and can cause many attacks possible.

Like above mentioned vulnerabilities, there are several more attacks such as Timestamp Manipulation, Default Visibility, Exception Disorder, Type cast inconsistencies, Stack Size Limit, etc. [21] [22]

2.9 Current System

The current system to transfer domain names consists of a broker which acts as a middle man between the participants. Domain name brokers are centralised entities. [1]

2.9.1 Domain Brokers

There are different entities that provide broker service such as Hosting providers, a registrar may also provide a service, third-party services to serve as domain broker. But the mandatory entity that is involved is a Domain Name Registrar where the final ownership records are modified to complete the ownership transfer process.

Let's take the case of GoDaddy (an organisation that provides Hosting Service, Domain Broker Service, and is a Domain Name Registrar itself). Here you can buy a new domain. But let's say we search for "paytm.com" (which is already owned by an entity), GoDaddy's interface provides us with an option to take their Domain Broker Service. The "Broker Service Fee" as mentioned there is around > 4000 INR. Now, this fee is for hiring a personal Agent ("Domain Buy Agent" as GoDaddy calls it). Once an entity buys their service, then the negotiation with the current Registrant of "paytm.com" (which may or may not be interested in selling) proceeds. If the negotiation is finalised, then the Buyer will have to pay the final settled price for the domain plus any commission charged by the Service Provider. The GoDaddy's commission for the same is 20% of the settled price. So in total, the Buyer pays $(4000 + 1.2 * (\text{Price of Domain}))$ INR to GoDaddy and the original owner of "paytm.com" gets (price of domain) INR.

2.9.2 Domain transfer from one Registrant to another

To transfer a domain name to another registrant, the owner can initiate a change of registrant by contacting the current registrar. The registrar will then ask for the owner's confirmation via a secure mechanism (which typically will take the form of an email to the registered name [3] [2]holder).

The owner must provide their confirmation within the number of days set by the registrar (not to exceed 60 days) or the transfer will not proceed. Once the registrar receives confirmation from the owner, they will process the transfer and notify the owner and the new registrant once the transfer is completed.

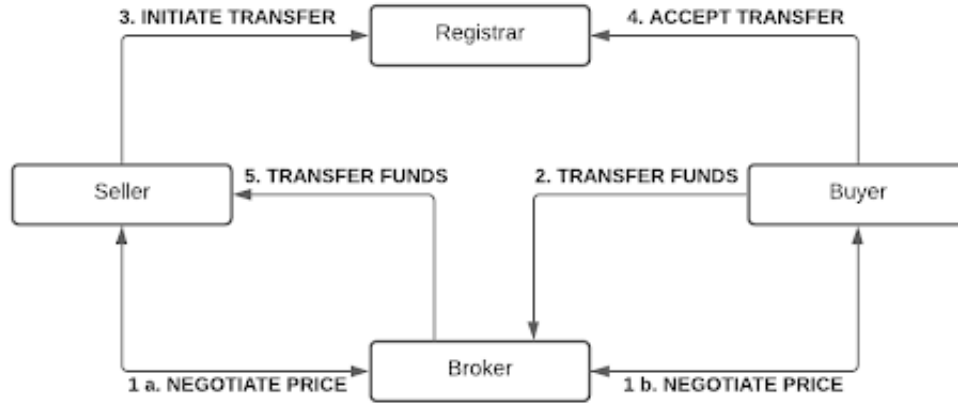


Figure 4: Broker's role in a centralised environment

Going through the domain transfer process, it is evident that there is no extra charge involved in ownership transfer between registrants. But the current Domain Broker Services charge a lot.

2.9.3 A typical (centralised) escrow service

1. The seller (Current Owner - O1) puts a listing on the broker's list through the interface.
2. The buyer (Final Owner - O3) may visit the interface and see for different listings, showing interests according to the requirement.
3. The buyer and seller agree to terms of the service by the escrow.
4. The settlement process -
 - The buyer (O3) pays the decided amount + Escrow's commission (status may be viewed by the seller).
 - The seller (O1) transfers the ownership of the domain to the escrow service (O2) or the buyer (O1) (depending on the terms of service).
 - When both of the above steps are done, the settlement process continues further.
 - The broker starts the transfer process simultaneously. Sends the decided amount to O1 and the domain ownership to O3.

2.9.4 Problems with Centralised Approach

The current approach on which the secondary market is working has flaws related to trust between the parties involved. The following problems can easily arise in the current architecture.

- The buyer transfers the funds, to broker, seller transfers the ownership of domain name but broker scams and deny the funds transfer to seller.
- The buyer transfers the funds to broker, but seller doesn't complete the ownership transfer of domain name and broker scams with the money received from the buyer.
- The existence of Brokers introduces the scope of various scams such as phishing attacks, click-baits, masquerade attacks, etc.
- Due to non-transparency in the system there is a big scope of overpricing from the broker's end to make more profit. As a result, the buyer may have to pay a greater amount.

3 Related Work

Currently in the secondary market of Domain names we can see some major players who have been in the market for few years now. They have developed strong trust and stability in the market.

Sedo.com [24] is a popular web platform for domain auction, a marketplace for buying and selling, with option to park domains. Sedo also provides a domain brokerage service. But it follows centralised model of operation and the organisation is the central entity that facilitates these services.

Another popular platform for domain selling is Escrow.com [25] which does not specialise in secondary market of domain but it is a general platform to facilitate and provide escrow service for buying and selling goods and service online. This is also an established platform since its inception over a decade ago. Escrow.com is also based on centralised model.

Apart from these consolidated platforms that involve with domain name secondary market, some other popular players are the major organisations around the domain name such as GoDaddy which provides both Broker Service [26] and Auction platform [27].

So, although these common platforms do exist but their mode of operation is centralised and is not very much trusted when we have newer technologies that can be used to design an open and much secure mechanism to carry out these transactions in the domain market. Till now, there are no decentralised platform for the secondary market of domain name.

4 Proposed Work

4.1 Blockchain based approach

A Blockchain-based service will replace the current Broker Services offered by various entities. The service will be replacing some of the costly, fraudulent, semi-transparent practices involved in the process.

4.1.1 Advantages

- The middle man plays a vital role in the legacy DNS but there are high possibilities for any scam to take place. In the new blockchain based approach, the elimination of the middle man assures no scope of fraud.
- The blockchain based model is a decentralised system. This approach ensures that the brokers do not have a monopoly over the system and extract an unreasonable amount of money from both the ends.
- Implementation of the smart contracts in the system encourages trustful ownership transfer between the buyers and sellers.
- Once a transaction concerning a particular buyer and seller has been settled, the ownership details of the same remains in the system, unchanged and permanent.

4.1.2 Challenges

- To decide the favorable architecture of the Blockchain from the available Public and Private platforms. All the available Blockchain Platforms available in the market have some trade-off between the characteristics.
- Secure procedure for verification of ownership transfer using Blockchain.

4.1.3 Proposed Implementation

In the concerned use case, we want to bring trust between the buyers and sellers for the transactions in the form of digital currency, governed by smart contracts without any need of centralised fiat currency.

Keeping in mind all these points, choosing a public blockchain platform will be more feasible for the selected use-case.

The implementation will include a web-interface where sellers can list the domain for selling. While making an initial request for listing a domain for sale, the seller has to set a base price for auction on the platform. Interested buyers can put on their bids on the

domains which they want to buy. The auction process will be completely governed by a separate smart contract. During bidding the visitors will have to transfer the bid amount at the contract address.

The amount paid by the auction winner will be transferred to the seller's account after ownership transfer of domain and rest of the pooled money will be transferred back to the bidders addresses. All the data related to listing of the domains will be governed by another smart contract.

4.1.4 Value proposition of Domain Names using Auction

AFNIC has suggested in one of their Issue Paper [1] that, the value of a domain name is determined mostly by factors (like search engine rankings, meaning of name, public perception, keyword competition, traffic analysis etc.) which cannot be formulated easily (at least has not been formulated till date). Moreover, an individual/organisation may prioritise these factors differently (based on their opinion) making the valuation even more difficult.

Hence in our opinion a bidding platform is the most suitable way to determine the value associated with a domain name. For the scenario at hand, we have decided on English auction type to keep the bidding process intuitive to the bidders. The bidding will be public (i.e. all the bidding made will be publicly visible as the bidding continues). Also, a person can put multiple-bids for the same listed domain. [13]

4.1.5 Components

- Client decentralised application,also known as 'dApp'
- A Public Blockchain Platform with turing complete smart contract support.
- A centralised service to interact with and perform off-chain tasks required in the use-case involved like, email verification, domain name ownership verification, and domain name transfer verification.
- Smart Contracts governing the business logic for the auction process and automatic settlement of funds.

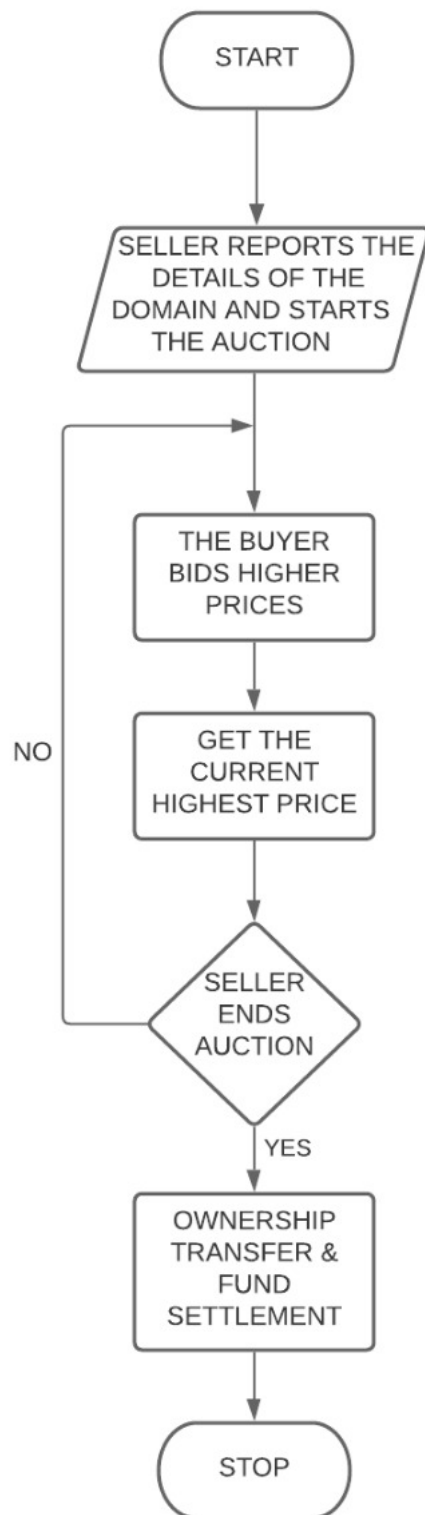


Figure 5: Auction Activity

4.1.6 Detailed Procedure

- The whole process starts with a seller listing the domain name owned by him/her on the platform. While listing, the seller is supposed to provide some details through a web interface - contact email, domain name, base price.
- A domain submitted for listing is checked for seller's ownership. If the current ownership is found to be with the seller, the domain is successfully listed for bidding otherwise an error is displayed to the seller (through the interface).
- For the ownership verification process, a request is sent to the centralised servers that handle the verification off-chain using information fetched from registry's RDAP service. Successful verification response is conveyed to the Smart Contract securely.
- Once a domain name is listed for bidding, the seller has the choice of starting the bidding and if a bidding is in progress, he/she can end the bidding too.
- A potential buyer can bid in an ongoing bidding for a domain listing (again through the interface). The information related to a domain listing (bid value and bidders) is also displayed on the webpage. To participate in the bidding, the bidder will have to pay the amount that he/she bids with.
- Upon the end of bidding process, the bidder with highest bid value is the potential buyer. The seller is supposed to (and is informed accordingly by our service) transfer the ownership of domain name to the highest bidder in a given time duration (downtime period).
- The service's back-end structure will check for the ownership transfer progress to the rightful owner. If the transfer is successful in the given duration, the fund settlement with seller takes place and the pooled funds of other bidders is refunded.
- However, if the ownership transfer doesn't happen in the given span of time, the bidding is revoked and all the pooled funds are refunded to the respective bidders.

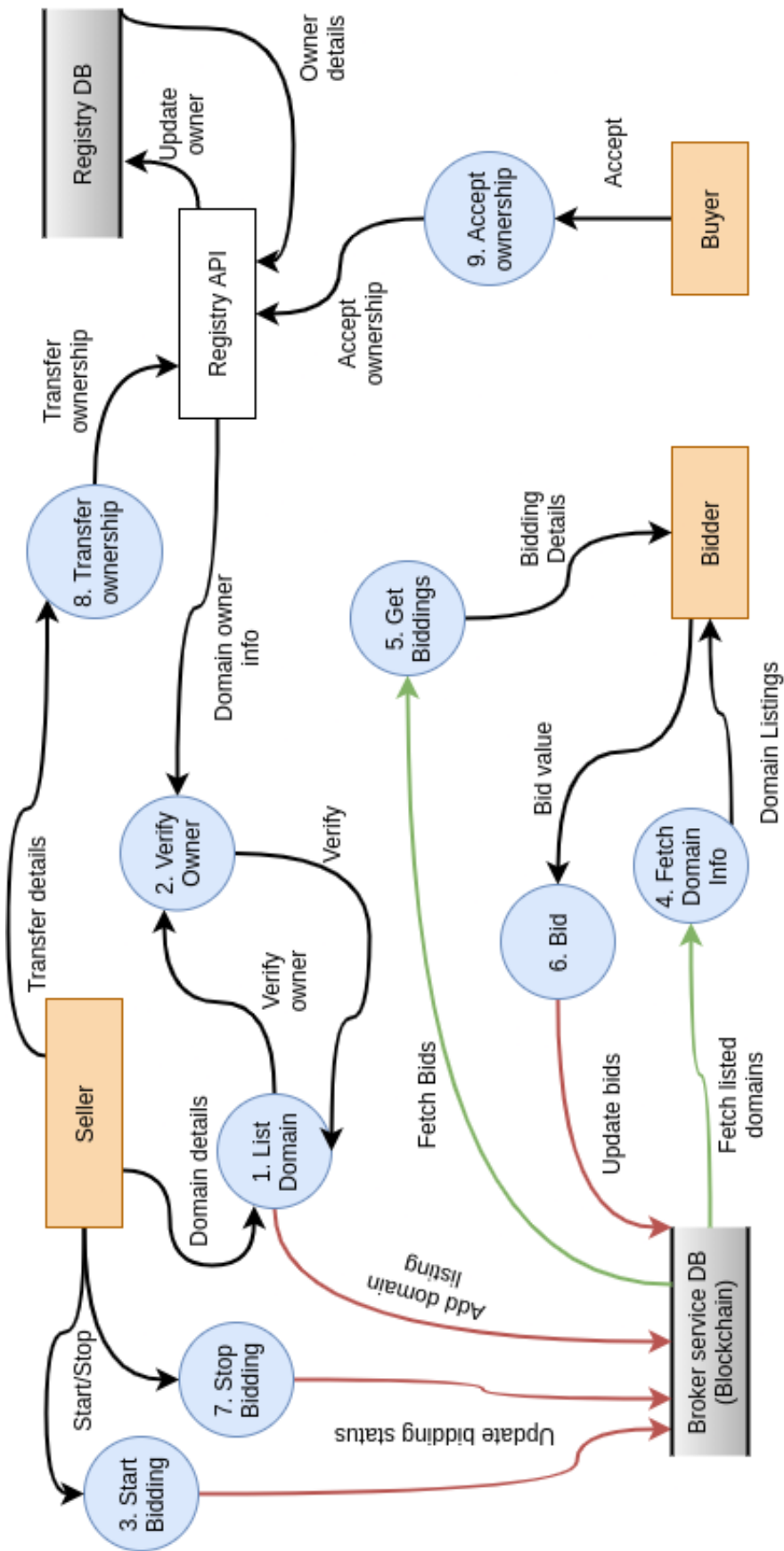


Figure 6: Blockchain-based approach to transfer domain names

4.1.7 Actors' involvement

- Seller
- Bidder
- Buyer/Highest bidder
- Visitor

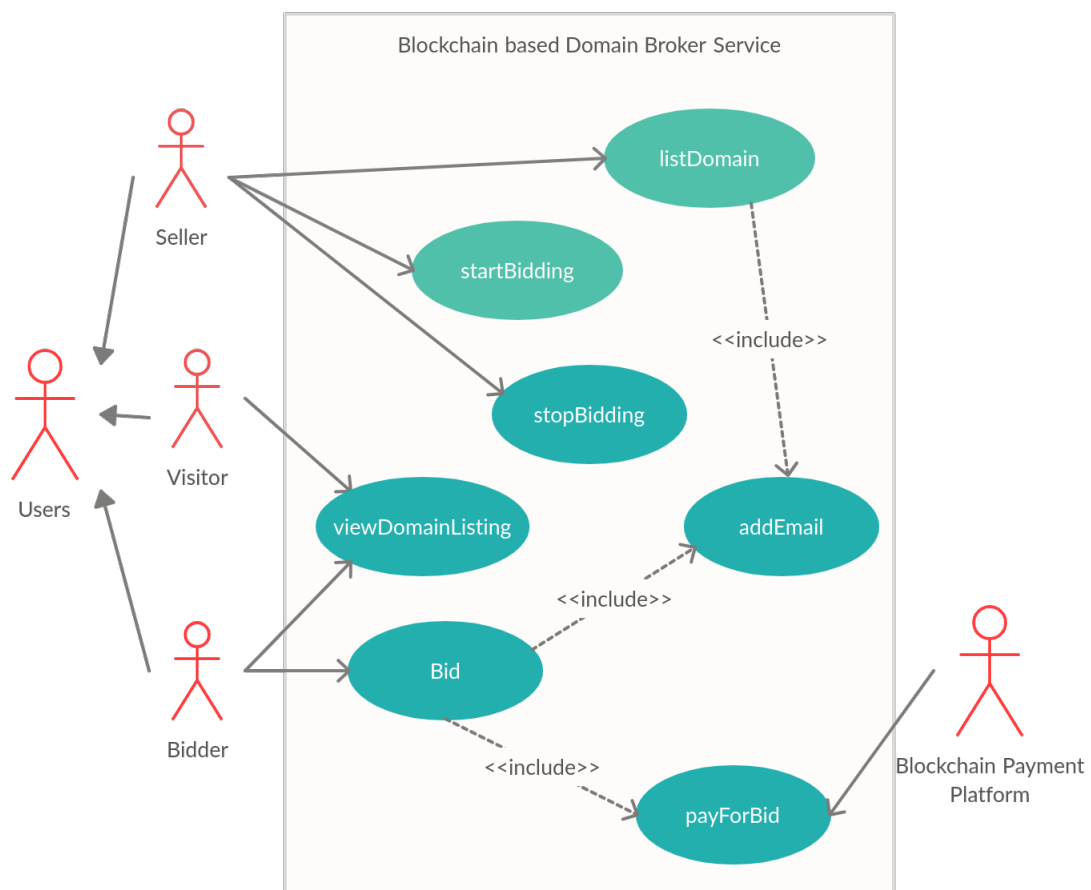


Figure 7: Involvement of various actors in the system

5 Implementation

5.1 Selection of Tools

Choosing the right blockchain platform is necessary as it will also determine what other tools, frameworks, Software Development Kit, and Integrated Development Environment (IDE) will be used in the process. From our detailed comparison in Section 3.2, we get an idea of trade-offs made by the platform. According to the requirement of the service, we believe that Ethereum would be a good choice for the Blockchain platform as it is a public blockchain network, provides decentralized application development, has matured itself in terms of both popularity and improvements to its software and protocols. Apart from that the development tools, Application Programming Interface (API), IDEs, programming language, and compiler are all well-managed by the community and very popular. Also, the community support of Ethereum has been growing ever since. For building a decentralized application, known as dApp on Ethereum, the following tools, libraries, and frameworks are enough: Remix IDE, Ganache, Truffle JS, Web3 JS and Metamask. Remix IDE is an IDE to write smart contracts for ethereum. It supports famous languages like Solidity and Vyper. It also allows the efficient debugging of smart contracts. Ganache is a local implementation of ethereum which you can run on your own machine. It allows the local system testing of the dApp. Truffle JS is a javascript library to facilitate the development of dApp on your system. It provides a local solidity compiler to compile the smart contracts. It also allows the deployment of the smart contract to the local ganache network or ethereum test-nets. Web3 JS is also a javascript library that helps in creating a connection between the ethereum blockchain and the web-application. Metamask is a wallet, which helps in signing the transaction and connecting the dApp with the local ethereum ganache network, test-networks, or main-networks.

Lastly, we talked about smart contract vulnerabilities in Section 2.8 and also mentioned some scenarios where small bugs in contracts led to a huge economic loss to the people involved. It also raises questions about trust issues on Blockchain platforms. To avoid bugs in the smart contract, there are some popular tools (called fuzzers) available that automate the fuzzing on smart contracts to find potential bugs. Some of the tools that provide support for Ethereum and Solidity language are Echidna and SmartCheck.

5.2 RDAP

The Registration Data Access Protocol(RDAP) allows the users to access the current registration data. Its main focus lies in standardizing the data access as well as the query response formats. Due to the various advantages that have been offered by this protocol, it has eventually replaced the common WHOIS protocol used before. RDAP

is an HTTP-based REST-style protocol that normalizes responses specified in JSON. It assists the JSON responses over the HTTP/HTTPS using URLs to differentiate the various resources present.

Taking into account the standardized form of output data that it provides while being able to access the domain registration, the project focuses on building a model that is able to mimic these features of RDAP. We model a REST API that is able to give and take requests from a dummy database through the 'GET' method. The user provides the details of the email address and the domain name associated with it. The python script performs a check over whether the email address has the corresponding domain name in the database. If it does, then the user is said to be an authorized one and can enter into the blockchain network to perform the desired activity. The main reason on why we are imitating the RDAP service and not implementing one is due to the fact that we would require to do the registration of the registries. We save time by not going into the intricacies of registering every registry and therefore, exploit the essence of RDAP through this imitation.

We have an end point in our API which provides all the data associated with a domain name.

```
[GET] /data?domain=DomainName
```

This endpoint gives the details associated with the domain name such as Email, Expiry Date, Status, Username registered with the domain, which can then be used to verify the ownership details.



Figure 8: Response of API End Point

5.3 Auction algorithm

```
1 bids_list ← empty;
2 domain ← "example.com";
3 seller ← < SellerAddress >;
4 base_price ← < BasePrice >;
5 highest_bid ← < HighestBid >;
6 winner ← (null, 0);
7 Function makeBid(bidder, price):
8   if bid is active && price ≥ base && price > highest_bid then
9     bids_list ← bids_list + (bidder, price);
10    highest_bid ← price;
11 Function onAuctionEnd():
12   max ← (null, 0);
13   foreach bid in bids_list do
14     if max.price < bid.price then
15       max ← (bid.bidder, bid.price);
16   winner ← (max.bidder, max.price);
```

Algorithm 1: Auction algorithm

5.4 Smart Contracts

The platform will rely on two smart contracts - Domain Market and Domain Auction. The Domain Market smart contract will keep track of all the available domains available for sale on the platform and provide basic transactional functions like submitting new domain for sale, checking the status of auction, etc.

An instance of Domain Auction, created will always be associated with each domain available for sell and it will govern the basic auction activities like bidding, ending auction, receiving funds from highest bidder, etc. The wrapper functions to call the functions of Domain Auction are also defined in Domain Market for easy access.

We used Solidity v0.5.11 for writing the Smart Contracts and debugging is done through Remix IDE.

5.4.1 Main Contract - DomainMarket.sol

```
1 // SPDX-License-Identifier: GPL-3.0
2
3 pragma solidity ^0.5.11;
4
5 import "../domainAuction.sol";
```

```

6
7 contract DomainMarket{
8
9 struct Domain {
10     uint id;
11     address payable owner;
12     string domainName;
13     string contactEmail;
14     uint basePrice;
15     bool ownerVerified;
16     bool succesfullySold;
17     DomainAuction auction;
18 }
19
20 Domain[] public domainForSell;
21 uint public id;
22
23 constructor() public {
24     id=0;
25 }
26
27 function sellDomain(string calldata _domainName,string calldata
    _contactEmail,uint _basePrice) external {
28     DomainAuction _auction = new DomainAuction();
29     Domain memory _domain = Domain(id,msg.sender,_domainName,
        _contactEmail,_basePrice,false,false,_auction);
30     id++;
31     domainForSell.push(_domain);
32 }
33
34 function getOwnershipVerifiedStatus(uint _id) external view returns(
    bool){
35     return domainForSell[_id].ownerVerified;
36 }
37
38 function setOwnerVerified(uint _id) external {
39     //will be trigegred by off-chain service only //to-do:add require
        condition
40     domainForSell[_id].ownerVerified=true;
41 }
42
43 function getSucsesfullySoldStatus(uint _id) external view returns(bool)
    {
44     return domainForSell[_id].succesfullySold;
45 }
46
47 function setSucsesfullySold(uint _id) external {

```

```

48     require(msg.sender == domainForSell[_id].owner);
49     domainForSell[_id].successfullySold=true;
50     domainForSell[_id].auction.endAuction();
51 }
52
53 function getAuctionContractAddress(uint _id) view public returns(
    address){
54     return address(domainForSell[_id].auction);
55 }
56
57 function stopAuction(uint _id) external {
58     require(msg.sender == domainForSell[_id].owner);
59     domainForSell[_id].auction.endAuction();
60 }
61
62 function showHighestBid(uint _id) external view returns(uint){
63     return domainForSell[_id].auction.getHighestBid();
64 }
65
66 function showHighestBidder(uint _id) external view returns(address){
67     return domainForSell[_id].auction.getHighestBidder();
68 }
69
70 function transferFundsToSeller(uint _id) external {
71     domainForSell[_id].auction.transferToSeller(domainForSell[_id].
        owner);
72 }
73
74 function makeBid(uint _id,uint _amount) public {
75     Domain memory _domain = domainForSell[_id];
76     require(_amount>_domain.basePrice);
77     _domain.auction.bid(_amount);
78 }
79
80 }

```

Listing 1: Main Contract - DomainMarket.sol

5.4.2 Auction Contract - DomainAuction.sol

```
1
2 pragma solidity ^0.5.11;
3
4
5 contract DomainAuction {
6
7     address public owner;
8
9     address public highestBidder;
10    uint public highestBid;
11
12    uint public deadline;
13    uint public auctionStartTime;
14
15    bool public auctionFinished;
16    bool public moneyTransferred;
17
18    event AmountSentToContract(uint _value,address _from);
19    event AmountSentToSeller(uint _value,address _to);
20
21    constructor() public {
22        auctionFinished = false;
23        moneyTransferred= false;
24        owner = msg.sender;
25        deadline=120; // the auction discard time in seconds
26        auctionStartTime=now;
27    }
28
29    // for direct deposit
30    function() external payable {
31        require(msg.sender==highestBidder);
32
33        // require statements commented as fallback function has low gas
34        // limit-2300, will work fine when implemented as a normal
35        // function in later phase
36        //require(now<auctionStartTime+deadline);
37        //require(msg.value==highestBid);
38        //require(moneyTransferred==false);
39        moneyTransferred=true;
40        emit AmountSentToContract(msg.value,msg.sender);
41    }
42
43    function getHighestBid() public view returns(uint){
44        return highestBid;
45    }
46}
```

```

44
45     function getHighestBidder() public view returns(address){
46         return highestBidder;
47     }
48
49
50     function bid(uint _amount) public {
51         require(_amount>highestBid);
52         highestBidder=msg.sender;
53         highestBid=_amount;
54     }
55
56     function endAuction() public {
57         auctionFinished = true;
58     }
59
60     function getContractBalance() view public returns(uint){
61         return address(this).balance;
62     }
63
64     // seller will trigger this in final version
65     function transferToSeller(address payable _seller) public {
66         require(auctionFinished==true && moneyTransferred==true);
67         _seller.transfer(address(this).balance);
68         emit AmountSentToSeller(address(this).balance, _seller);
69     }
70
71 }

```

Listing 2: Auction Contract - DomainAuction.sol

6 System Analysis

6.1 Security Analysis

Since Blockchain can resist traditional cyber attacks very well, the problems mentioned in Section 2.9.4 are easily handled because the core functions of the proposed work processes will be governed by the smart contracts containing the business logic for checking the ownership transfer and funds transfer.

Although, the secondary market established on the top of the Blockchain Technology provides advantages discussed in Section 4.1.1, the architecture is not full-proof from various attacks as cyber-criminals are coming up with new approaches specifically for Blockchain technology. In this section we try to explain possible attacks on the system.

The attacks on the proposed architecture can be categorised based on the components involved in the system. At the highest level, the system can face attacks that can be categorised as:

- Attacks on the centralised components/services (that aid the core decentralised components); and
- Attacks on the decentralised components

6.1.1 Attacks on decentralised components

- Attacks on Blockchain Network
DDoS, Routing attack (Border Gateway Protocol (BGP) hijacking), Timejacking, Ellipse attack
- Attacks on Consensus Protocols
51% attack, Long-Range Attack, Sybil Attack
- Attacks on Cryptocurrency wallet
Phishing, Dictionary attack
- Attacks on Smart Contracts [10] (detailed in Section 2.8.1)
Short address attack, Integer Overflow/Underflow, Reentrancy

The attacks on first three components are because of existing fault in the system itself and hence developing solutions for those are out of the scope of this project. Although some of those attacks can be prevented by making a choice of Blockchain network that is resilient to as many network based attacks as possible. Also, we can make selection of a Blockchain network that utilises a consensus protocol immune to most of the attacks.

Still, a special attention is to be given, while writing the pooling contracts for auction which can lead to various Smart Contract Attacks [10] due to Reentrancy, Underflow and Overflow, etc. Once deployed on the network Smart Contracts cannot be updated for security patches. This is because smart contract deployment occurs through a block transaction on the Blockchain network and due to Blockchain's immutability feature nothing can be modified on the network. The famous DAO Attack [11] is a great example of the security attack on a smart contract costing millions of dollars simply as a consequence of bugs in the smart contract code.

Therefore, the analysis to find smart contract vulnerabilities, needs to be done through various Smart Contract Analysis and Fuzzing tools. Fuzzing is a software testing technique which consists in finding implementation bugs with the help of malformed/semi-malformed data injection. Fuzzers are software tools that are used to automate the fuzzing process. Some of the popular fuzzing tools are Echidna and SmartCheck.

6.2 Limitations of Blockchain

Scalability issue - This is a big limitation of the Blockchain network that it is not at all scalable either horizontally or vertically. This comes as a result of consensus mechanisms that are independent of network capacity as a whole but depend on the computational capacity of individual nodes that have to solve the mathematically complex problem in PoW. A service such as a broker service may demand a relatively scalable architecture.

Solution: At present, there are two proposed solutions that try to introduce scalability to the Blockchain network. The first one is Lightning Network and the second is Plasma, both try to increase the TPS of any Blockchain network by creating secure channels between two parties/nodes on the network and thus allowing them to exchange information at a speed not limited by the original Blockchain network.

Inefficiency - A blockchain network is considered to be highly redundant in terms of storage and computation. The consensus algorithms and mining operations (that make up for the core part of Blockchain) are highly redundant in the sense that every node performs these operations individually on their own copy of the database (distributed ledger). Since there is so much computational redundancy from the computational point of view, it can be said that it is an inefficient system. This is the case of the PoW consensus mechanism.

Solution: (A) New consensus mechanisms are a solution to the current (though secure but) highly inefficient approach of PoW. PoS, DPoS can be a solution to this problem. (B) Lightning Network as mentioned earlier can also solve this issue.

Huge database - Blockchain derives part of its strength from a P2P network with distributed databases. Since a blockchain database is an append-only ledger that is

public, its size can grow exponentially. As of Q3 2020, the Bitcoin network has a total of 302 GB [28]. On similar grounds, Ethereum’s data amounts to 551 GB as of Q3 2020 [29]. Even though for verification purposes, the entire database is not required (Simple Payment Verification allows verification using only the required block), but for the blockchain network to be more distributed a significant number of nodes should download the entire database. Hence only systems with high resources can participate in the network thus reducing level decentralization.

Solution: (A) One idea is to deploy data servers. (B) The concept of the Merkle tree can reduce the size of the database by approximately 200 times. There can be several approaches to this problem and it is easy to tackle. One of the possible solutions is Sharding [30].

6.3 Challenges

The proposed solution discussed here tries to make a decent tradeoff between scalability, decentralization, consensus, and cost factors which might not be as suitable after a few years given that the current Blockchain technology is still evolving at a fast pace. Along with all the advantages and impact on the fee reduction in the secondary market of domain names through the blockchain-based approach, one serious challenge is scalability. As discussed the public blockchain platform will be more feasible for the use-case, but the tradeoffs in public blockchain usually lower down the performance. Overcoming the right solution to meet the high-performance benchmarks for the use-case will be a challenge. Layer-2 Blockchain solutions are possible approaches for the same. Another challenge is our system deals with two databases: On-chain database (Blockchain) and Off-chain database (Registry database) and connecting these together can pose a lot of security loopholes that can impact the overall system. This will require further testing and thorough analysis of proposed solutions. Currently, Oracles are used whenever smart contracts need to deal with off-chain data; but that may not be a feasible solution considering the scalability and cost factors.

7 Conclusion

In this report, we explored the replacement of current central brokers in secondary market of domain names. We looked at various issues that the current centralized model of operation has and how the inherent features of Blockchain can be utilized to design architecture without such issues. We have also explored how our blockchain-based implementation can be thought of as a replacement for the current market. Along with the implementation we discussed our way through what are the best possible choices of architecture for our implementation, various trade-offs made and the enhancements (to the current model) that the new blockchain-based service brings to the table. In conclusion, such a blockchain-based service is feasible to implement using the current technologies that will provide a trustless platform for the secondary market of domain name buying and selling.

References

- [1] AFNIC, The secondary market for domain names, 2010, [online] Available: <https://www.afnic.fr/medias/documents/afnic-issue-paper-secondary-market-2010-04.pdf>.
- [2] Escrow.com, "How to transfer domain names", [online] Available: <https://www.escrow.com/domains/how-to>.
- [3] ICANN, Resources, [online] Available: www.icann.org/resources.
- [4] Niclas Kannengießer, Sebastian Lins, Tobias Dehling, Ali Sunyaev, "*Trade-offs between Distributed Ledger Technology Characteristics*", ACM Computing Surveys, Vol. 53, No. 2, Article 42, May 2020
- [5] Daniel Larimer, "*Delegated Proof-of-Stake (DPOS)*", 2014.
- [6] Olivier Moindrot, Charles Bournhonesque, "*Proof of Stake Made Simple with Casper*", 2017.
- [7] Satoshi Nakamoto, "*Bitcoin: A peer-to-peer electronic cash system*", 2008, [online] Available: <https://bitcoin.org/bitcoin.pdf>.
- [8] Dr. Gavin Wood, "*ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER, EIP-150 REVISION*", 2014, [online] Available: <https://gavwood.com/paper.pdf>
- [9] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen, and Huaimin Wang. "*Blockchain challenges and opportunities: A survey*". International Journal of Web and Grid Services, 14:352, 10 2018.
- [10] S. Sayeed, H. Marco-Gisbert and T. Caira, "*Smart Contract: Attacks and Protections*," in IEEE Access, vol. 8, pp. 24416-24427, 2020, doi: 10.1109/ACCESS.2020.2970495.
- [11] X. Zhao, Z. Chen, X. Chen, Y. Wang and C. Tang, "*The DAO attack paradoxes in propositional logic*," 2017 4th International Conference on Systems and Informatics (ICSAI), Hangzhou, 2017, pp. 1743-1746, doi: 10.1109/ICSAI.2017.8248566.
- [12] Verisign.Com, "The Verisign Domain Name Industry Brief ,Q2 2020" [online] Available : https://www.verisign.com/en_IN/domain-names/dnib/index.xhtml
- [13] Wikipedia.Org, "Online auction - Wikipedia" [online] Available : https://en.wikipedia.org/wiki/Online_auction

- [14] ICANN, WHOIS, [online] Available: <https://whois.icann.org/en/about-whois>
- [15] ICANN, Temporary Specification for gTLD Registration Data, [online] Available: <https://www.icann.org/en/system/files/files/gtld-registration-data-temp-spec-17may18-en.pdf>
- [16] ICANN, Registration Data Access Protocol (RDAP), [online] Available: <https://www.icann.org/rdap>
- [17] A Tool, RDAP Client, [online] Available: <https://client.rdap.org/>
- [18] A Tool, RDAP client on ICANN, [online] Available: <https://lookup.icann.org/>
- [19] GDPR, What is GDPR, [online] Available: <https://gdpr.eu/what-is-gdpr/>
- [20] RDAP Access, RDAP Access request form from GoDaddy for .biz domain, [online] Available: <https://rddsrequest.nic.biz/>
- [21] Atzei, Nicola and Bartoletti, Massimo and Cimoli, Tiziana, A survey of attacks on Ethereum smart contracts, [online] Available : <https://img.chainnews.com/paper/f8084c122c0dfefd33e6bf03246597e8.pdf>
- [22] S. Sayeed, H. Marco-Gisbert and T. Caira, "Smart Contract: Attacks and Protections," in IEEE Access, vol. 8, pp. 24416-24427, 2020, doi: 10.1109/ACCESS.2020.2970495.
- [23] Chinen, Yuichiro & Yanai, Naoto & Cruz, Jason Paul & Okamura, Shingo. (2020). Hunting for Re-Entrancy Attacks in Ethereum Smart Contracts via Static Analysis.
- [24] Sedo, "Sedo company details, the best place for domains is Sedo.com", [online] Available: <https://sedo.com/us/about-us/>
- [25] Escrow, "About Escrow.com, The Online Escrow Service - Escrow.com", [online] Available: <https://www.escrow.com/why-escrowcom/about-us>
- [26] GoDaddy Broker Service, "Domain Broker — Your Domain Buy Service - GoDaddy", [online] Available: <https://godaddy.com/domains/domain-broker>
- [27] GoDaddy Auction, "Domain Auction — Buy & Sell Distinctive Domains - GoDaddy", [online] Available: <https://auctions.godaddy.com/>
- [28] Blockchain.com, "Blockchain Charts", [online] Available: <https://www.Blockchain.com/charts/blocks-size>
- [29] Etherscan.io, "Ethereum Full Node Sync (Default) Chart — Etherscan", [online] Available: <https://etherscan.io/chartsync/chaindefault>

- [30] S. S. M. Chow, Z. Lai, C. Liu, E. Lo and Y. Zhao, “Sharding Blockchain,” 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCoM) and IEEE Smart Data (SmartData), Halifax, NS, Canada, 2018, pp. 1665-1665, doi: 10.1109/Cybermatics.2018.2018.00277.

Acknowledgement

We would like to express our deep sense of gratitude and indebtedness to our project guide, Dr. Sankita J. Patel, Associate Professor, Computer Engineering Department, SVNIT Surat for her valuable guidance, useful feedback, and co-operation with kind and encouraging attitude at all stages of experimental work for the successful completion of this work. We would also like to thank our Head of Department - Dr. Mukesh A. Zaveri, Computer Engineering Department for all the support. We extend our sincere gratitude to SVNIT Surat and its staff for providing us with this opportunity which helped us in gaining sufficient knowledge to make our work successful.