**HKUST President's Cup Final Report**

# Vivian: Decentralized Global Naming and Storage System on Tangle Distributed Ledger


Written by:

TIAN Xiangan

Student ID: 20583620

Year of Study: 3

Supervised by:

Prof. TSOI, Yau Chat

Department of CSE

Department of Computer Science and Engineering

School of Engineering

Hong Kong University of Science and Technology

# Vivian: Decentralized Global Naming and Storage System on Tangle Distributed Ledger

TIAN Xiangan
xtianae@connect.ust.hk

Supervisor: Prof. TSOI, Yau Chat
desmond@cse.ust.hk

***Abstract*—With the booming of distributed ledger technology (DLT) such as blockchain, many previous IT architectures can have alternative decentralized approaches for more secure, transparent, and immutable data storage. In this paper, we present the design and implementation of Vivian, a new decentralized global naming and storage system based on IOTA Tangle distributed ledger for re-decentralizing the current Internet service and building decentralized applications. The system has no single point of failure and the nodes in the network do not need to trust each other. Unlike the traditional Domain name System (DNS), trust points like DNS root servers are removed and critical data bindings are secured by the distributed ledger. All the nodes in the system form a peer-to-peer (P2P) network for user quires' routing. The P2P network is established through peer discovery protocols such as mDNS, Kademlia DHT and peers exchange data and achieve eventual consistency via Gossip protocol. We also provide a decentralized storage system which can hold user data securely without the control of central trust parties or revealing information to storage providers. By using IOTA Tangle, a directed-acyclic-graph (DAG) distributed ledger, the system inherits its scalable, lightweight, and feeless characteristics and most IoT devices have enough computational power to sign and send transactions. This extends the usage of Vivian to Internet-of-Thing (IoT) services for decentralizing IoT networks and enhancing IoT data security and privacy.**

## I. Introduction

A distributed ledger (DL) is a database that tolerates nodes with malicious intentions in a distributed manner. And distributed ledger technology (DLT) enables the realization and operation of distributed ledgers, which allows almost all the nodes in the network, to agree on an almost immutable record of transactions with Byzantine failure tolerance (BFT) and eventual consistency via a predefined consensus mechanism [1]. Blockchain is one of the most well-known DLTs which was first implemented on Bitcoin. It proposed a simple but robust way for transaction data storage without relying on the trust of third parties [2]. Blockchain also ensures improved security and anonymity of Bitcoin transactions compared with traditional electronic transactions. Since the introduction of Bitcoin in 2009, DLT based cryptocurrencies have made a great impact on financial industries. After this people also discovered that the usefulness of DLTs is beyond exchange of currencies and significant adoption of DLTs was made in many other industries for other different services. Namecoin is the first altcoin[1] for being the first to create its own blockchain separate from Bitcoin's [3]. And its functionalities are not limited to financial transactions. The creation of Namecoin was inspired by the idea of BitDNS [4] and for establishing a decentralized domain name looking-up system.

The Internet today is a widespread information infrastructure and its history can date back to 1970s when ARPANET[2] was developed. For most of the current Internet applications, data is stored in a centralized manner and users do not own data by themselves. As shown in Fig 1, if users want to do actions like checking their emails or browsing the content of a website, first, they need to connect to the web servers via the Internet with web browsers, then the web servers retrieve the data from the database and then send it back to users. Usually, users' data is hidden behind service providers' application code. This kind of arrangement has been very successful as it is easy to implement. However, it is not ideal since:

- Users must use the requested web user interface if they want to access their data.
- The websites control the rules and access rights of the data.
- The websites may snoop your data and sell users' information to others.
- Illegal use of data by websites' employees for personal purposes.

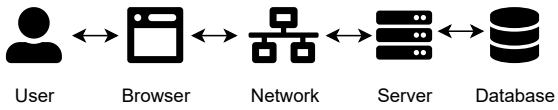



Fig. 1: User data arrangement of traditional (centralized) Internet application

Since the early Internet, hosts in the network were assigned names for more convenient use and memorization by humans. With the growth of the network, it became impossible to store all the hosts in a single table. And Domain Name System (DNS) invented by Paul Mockapetris of USC/ISI permitted a scalable distributed mechanism for resolving hierarchical host names into Internet addresses [5]. Now the coordination and management of DNS Root, IP Addressing, and other Internet protocols are in the charge of IANA[3] [6]. These DNS root servers are central nodes of trust and failure, and cyber-attack

[1]Altcoin: any cryptocurrencies that are not Bitcoin.

[2]ARPANET: Advanced Research Projects Agency Network. The first wide-area packet-switching network with distributed control originally established by the United States Department of Defense.

[3]IANA: Internet Assigned Numbers Authority. Website: https://www.iana.org

such as DDoS[4] towards them may leads to the whole system taken down. It is reported that a large scale of DDoS attack was launched towards 13 root servers on March 21st, 2002. Fortunately, the attack only lasted for one hour and didn't cause severe damage [7]. Besides, these central points may also be exploited and misleading users into connecting to malicious websites like Turkish fake site certs incident [8].

Internet-of-Things (IoT) refers to *"physical or virtual objects which connect to the Internet and has the ability to communicate with human users or other objects"* [9]. These devices such as smart webcam and wearable health monitors are widely used in our daily life. It is estimated that there will be approximately 30.9 billion active IoT device connections installed worldwide by 2021 [10]. Due to the heterogeneity and complexity of IoT devices, their security and privacy issues are becoming more and more severe [9]. With the increasing number of devices connected to the network, the load of centralized servers for handling the connection will become much higher. DLT supported IoT has been created for addressing the challenges like security, data integrity and reliability, and secured P2P sharing. It is a new decentralized and distributed solution to IoT services and enables the opportunity for developing new and creative applications and business models in vertical domains, e.g., from healthcare to supply chain, energy industry, and smart manufacturing [11].

**Motivation.** Many data management issues related to security, integrity, access control have been exposed from the centralized data model of the traditional Internet. When accessing web services, user data control is maintained by service vendors rather than users themselves. Domain Name System containing central nodes like DNS root servers are vulnerable to cyber attacks such as DDoS. We wish to re-decentralize the current Internet service via distributed ledger technology for better security and data integrity. Distributed ledger technologies like blockchain enhance the security and data integrity of IoT services. However, many current DLTs are based on Proof-of-Work (PoW) consensus mechanism [12], requesting strong computing power and huge energy consumption for solving hash computational puzzles. These mechanisms are not suitable for IoT devices that have limited computational power and energy consumption restriction. In addition, transaction fees paid to the miners in the network caused an extra cost for the service. DLTs like Bitcoin blockchain are also facing problems like low TPS[5], bad scalability, etc. They are not suitable for IoT service scenarios like sending a large number of micro-transactions in a short period of time. In summary, we want to use a feeless, lightweight, and scalable DLT that can support IoT use cases.

**Contribution.** We introduce the design of Vivian, a decentralized global naming and storage system secured by IOTA Tangle distributed ledger [13]. It is a new decentralized Public Key Infrastructure (PKI) system that enables users to register human-readable and unique domain names with binding information. In the system, there is no central trust points and users control their own data. We describe the structure of the system: Tangle DL layer, peer network layer, and storage layer. Tangle distributed ledger layer records critical data bindings. Peer network layer handles user queries and help user find the binding value of the names quickly. Storage layer stores user data under users' control securely and the storage providers cannot tamper with the data. We describe the implementation of Tangle DL for critical data binding and naming operations. We also describe the P2P network implementation including peer discovery and peer communication. In addition, we discuss about the application of the system in real-world scenarios.

## II. Background

### A. Blockchain and Tangle (DAG)

Blockchain DLT became widely known in 2009 with the launch of Bitcoin network. Participants in the network can validate and verify the transactions independently, without relying on central trust parties. Additionally, Blockchain is usually maintained and managed by a distributed group of participants independently. These along with its cryptographic mechanisms ensures the data recorded on the ledger immutable [14]. The structure of a traditional blockchain can be simplified as a singly linked list[6], you can traverse from the latest block on the chain to the Genesis block[7] (as shown in Fig 2). Transactions are hashed in a Merkle Tree [15] for saving storage space and simplifying transaction validation.

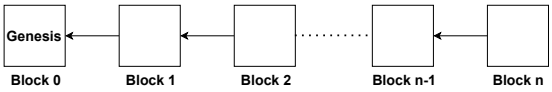



Fig. 2: Blockchain structure

In the original Bitcoin whitepaper [2], Satoshi Nakamoto listed the procedures of handling the transactions and record them on the blockchain ledger. All the nodes listen to the transactions broadcast to the network and each node collects new transactions for generating a new block. Then each node will do proof-of-work and broadcast the block to the network once it finds the solution for other nodes' validation. However, in the real case, almost all the transaction wrapping and PoW are finished by specific miners in the network. Miners also require transaction fees for the reward of doing these jobs. The increasing PoW complexity makes it nearly impossible to gain profit from mining Bitcoin with general PC hardware. Instead, dedicated miners have to use the equipments like ASIC[8]

[4]DDoS: Distributed Denial-of-service Attack. Usually, the attack attempts to disrupt the normal traffic of the victim's server by a large number of requests made by attacker devices.

[5]TPS: Transaction Per Second. The approximate average TPS of Bitcoin blockchain is around 5.

[6]Singly linked list: linked list which can only be traversed in one direction.

[7]Genesis block: the first block of a blockchain. It is a special case as it does not reference a previous block.

[8]ASIC: application-specific integrated circuit. It is the integrated circuit designed for a specific use case. Bitcoin ASIC chip can only handle computing tasks for Bitcoin mining, and cannot be used for any other tasks.

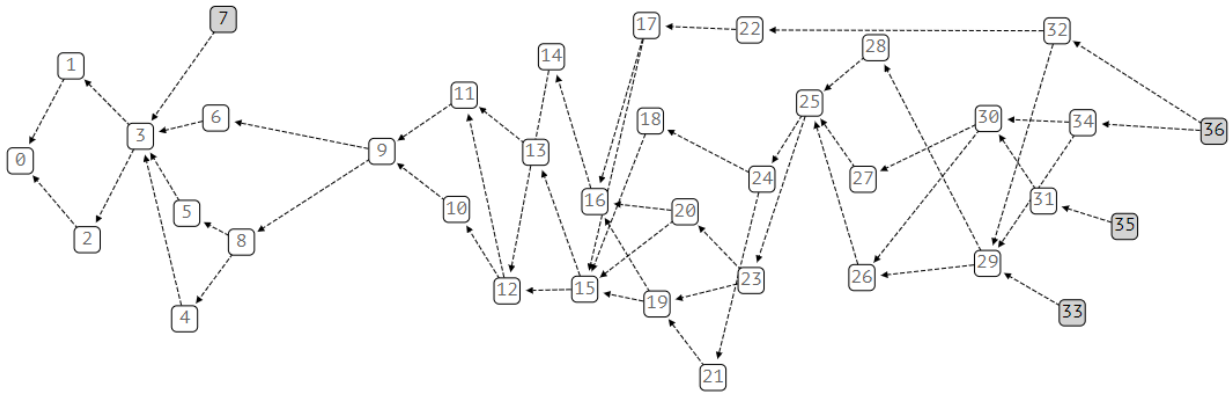



Fig. 3: Tangle directed-acyclic-graph structure. Shaded vertices represent tips (new transactions attaching to Tangle). The very first transaction, denoted as 0-th vertex, is the genesis transaction. It gave the total supply of IOTA tokens to one address.

specifically for the mining algorithms. It is a potential hazard that the blockchain network will be centralized and controlled by the parties that own the most of these "mining rigs". In addition, blockchains relying on PoW are consuming massive energy for their operations [16]. According to CBECI[9], only Bitcoin network consumes approximately 130.51 TWh electricity per year, which is far more than the annual electricity consumption of some countries like Ukraine and Argentina. Carbon dioxide emissions caused by these PoW blockchains will cause environmental issues like global warming.

Transaction throughput and transaction confirmation latency are the two most critical performance issues of blockchain technology [17]. Transactions can only be recorded on the blockchain in sequence due to its linear structure. Also, the limitation of the size of each block makes it struggling to handle the enormous volume of transactions nowadays. Blockchains including Bitcoin and Ethereum are facing problems like low TPS and bad scalability that result in transaction backlog and high transaction fees. To tackle these issues, people have come up with many alternative solutions such as side-chain, cross-chain, improved consensus, sharding, DAG, etc. IOTA Tangle is a DAG[10] DLT addressing solving the problems above for IoT services. It has the following advantages comparing with traditional blockchain technologies:

1) **High scalability.** Directed-acyclic-graph structured Tangle ledger enables its high scalability. Serguei Popov has analyzed the performances of the system under two different regimes: low load and high load in the article *The Tangle* [13] In high load regime when more new tips[11] are attached to Tangle, the typical time of a tip being approved is reduced. So the larger the scale of the network, the more efficient it will be.
2) **Feeless & Environmentally Friendly.** In Bitcoin network, we need to pay transaction fees to the miners for rewarding them wrapping our transactions to the block and conducting PoW computations. Transaction fees are considered part of the incentive for nodes to support the network. In IOTA Tangle, however, PoW consensus and miners are removed [18]. So it is more economical and environmentally friendly to use Tangle for sending transactions.
3) **Quantum Robust Cryptography.** IOTA Tangle uses post-quantum cryptography for securing data on the ledger [19]. For instance, IOTA uses Winternitz One-Time Signature (WOTS), which is promising to be resistant to quantum computers [20] as a signature scheme protocol.
4) **Lightweight.** IOTA node applications like GoHornet[12] are lightweight and can be easily installed and run on low-end devices such as Raspberry Pi 4.

These characteristics are very crucial for IoT applications and favorable for implementation of Vivian.

### *B. Decentralized Naming System*

Before distributed ledger technology came into being, building a decentralized naming system was considered almost impossible [21]. To build a system like it, three properties have to be satisfied:

1) **Human-meaningful.** The names the system provides should be meaningful and easy to memorize, e.g. bitcoin.org. Hash values like `764569e58f53` are obviously not meaningful and hard for people to memorize.
2) **Secure.** The names should return the correct binding values. Damage caused by malicious entities should be as low as possible.
3) **Decentralized.** No central authority controls the system. Names can be chosen by users at the edge of the network rather than representative central parties.

[9]CBECI: Cambridge Bitcoin Electricity Consumption Index. Website: https://cbeci.org/

[10]a directed graph with no directed cycles. It is a directed graph, which means each edge has an orientation, from one vertex to another. However, no path in the graph forms a circle.

[11]Tip: every new (unconfirmed) transaction is known as a tip.

[12]GoHornet: IOTA full node software built in Go. Github source code: https://github.com/gohornet/hornet

The challenge is that, with traditional approaches, a naming system can satisfy some of the properties, but not all three of them at the same time [22]. This trilemma is called Zooko's triangle [21] (very similar to CAP theorem[13] in distributed system design). Besides, the system should also ensure the uniqueness of the names, which means two different users cannot create and use the same name [23]. Domain Name System in the current Internet is human-meaningful and secure, but not decentralized. Public keys are secure and decentralized, but not human-meaningful. Zooko's Triangle has been squared by distributed ledger technology [21] and Namecoin was the first implementation to build a decentralized naming system over blockchain DL.

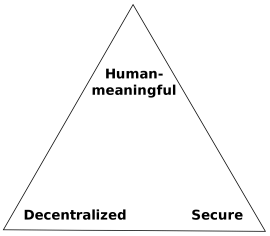


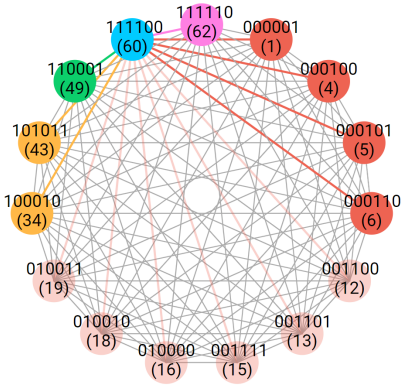

Fig. 4: Zooko's Triangle

Blockstack is a decentralized public key infrastructure (PKI) service built on Bitcoin for building a decentralized Internet [23]. The implementation of its naming system is based on the definition of a state machine and rules for state transitions on its virtualchains [24], [25]. In the design of Vivian's naming system, we referenced the ideas from Blockstack and Namecoin.

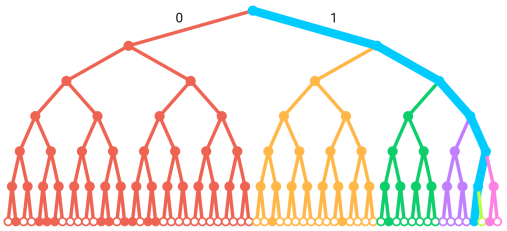
### C. P2P Network

In conclusion, peer-to-peer (P2P) networks are distributed networks in which participants provide services and contents that can be accessed by other peers directly without passing through intermediate entities by shared hardware resources [26]. In a P2P network, all the peers are equally privileged, equipotent nodes that forms the network [27]. Unlike the client-server model, a peer plays the roles of suppliers and consumers of the sources at the same time.

*1) Peer Discovery:* In a P2P network, resources like shared files are stored on different servers in a distributed way. In normal client-server model, if a client server wants to find another node or specific resources in the network, it usually sends requests to the central servers for targets' information. In the original implementation in BitTorrent [28], a well-known P2P file sharing protocol, tracker servers keep the global registry of all the downloaders and seeds of the corresponding files [29]. When the trackers are crashed or taken down, peers will not be able to connect with other peers and retrieve the files successfully. After that, a decentralized approach for finding other peers has been implemented: Distributed Hash Table (DHT). In Vivian, we use **Kademlia Distributed Hash Table (Kad-DHT)** [30] for peer discovery. In Kademlia, DHT is used for storing the resources location in the network. Every node and key has a unique ID (usually 160 bits). Each value is stored at the nodes whose ID are closest to the key ID. And the distance between every two nodes are calculated as exclusive or (XOR) of the two nodes' IDs. When searching through n nodes in the system, Kad-DHT only needs to contact $O\log(n)$ nodes.




(a) Nodes in the network




(b) Binary tree based on node ID

Fig. 5: Kademlia DHT. In this example, node ID is 6 bits, and routing table size is 64. Originating the look-up at Node 60 (111100), the k-buckets for it is [(1, 4, 5, 6), (34, 43), (49)] when k = 4 and alpha = 3.

Another peer discovery method used in Vivian is **Multicast DNS (mDNS)** [31]. In mDNS protocol, a device in the local area network broadcasts messages to all the other devices in the network to tell them its identity and address. The other devices which enable mDNS will respond the corresponding message after receiving the broadcast message.

*2) Peer Communication:* Nodes in Vivian use **Gossip Protocol** [32] for the dissemination of information such as naming key-value bindings. It is an eventual consistency algorithm based on the way of epidemic spreads. Gossip protocol has advantages like simple, highly scalable, faut-tolerant, and decentralized. Many well-known platforms such as Redis Cluster, Consul, and Bitcoin are using this protocol.

[13]CAP theorem: it is impossible for a distributed data store to provide, Consistency, Availability and Partition tolerance, all of the three guarantees simultaneously.

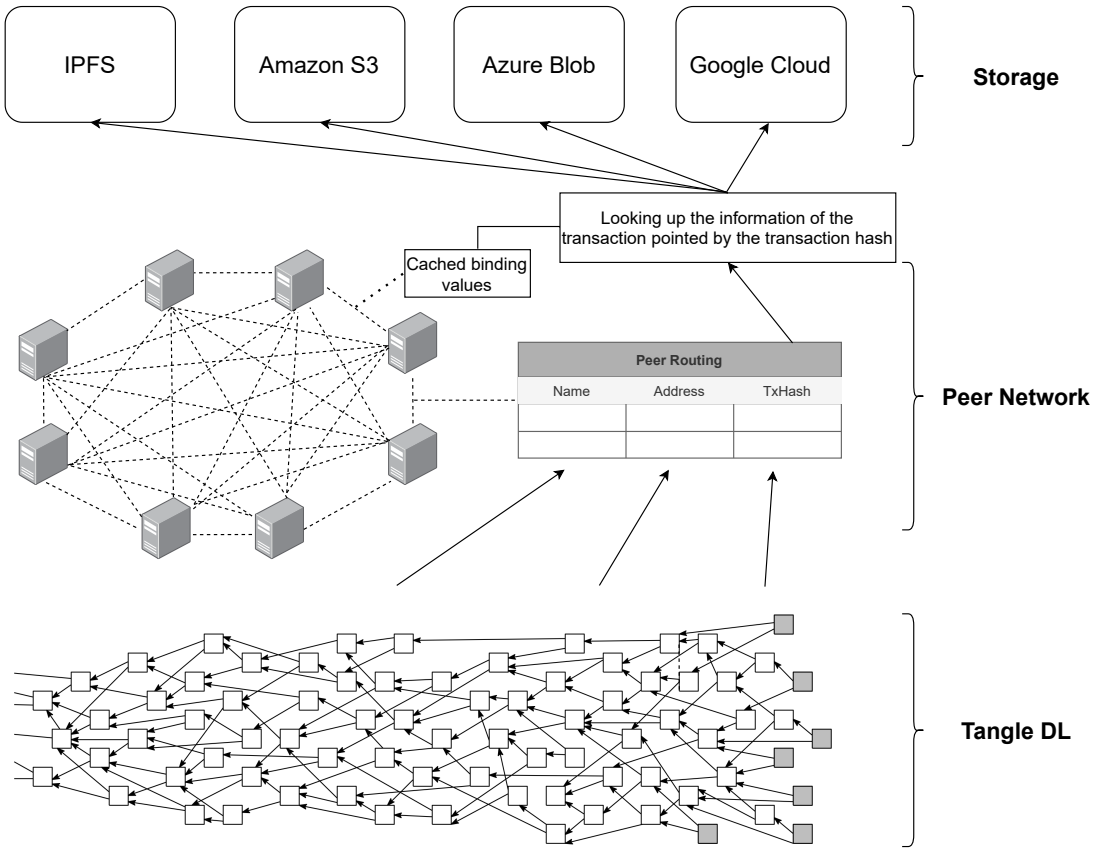



Fig. 6: Vivian Architecture

## III. System Design

### A. Overview

In designing Vivian, we want the system to achieve the following properties:

- **Decentralized Naming and Look-up.** End-users can register for and bind values to human-meaningful names and look up the values of names without relying on the trust of central authorities.
- **Decentralized and Secure Storage.** End-users can store their data in a decentralized manner. Besides, users can control the access rights of their data.
- **IoT Device Supported.** The whole process like registering a name, looking up a name, and file handling should not be energy-hungry or hardware resource hungry. They can be accomplished by devices with limited hardware resources such as Raspberry Pi.

The overall architecture of Vivian can be divided into three layers: **Tangle DL layer, Peer Network layer, and Storage layer**. Tangle distributed ledger is used as storage intermediate of critical data binding and for consensus of the order of the system operations. Nodes in the peer network layer are equally privileged, equipotent peers which perform the live listening to transactions on Tangle, parsing the operations, storing naming key-value pairs, and routing for users to find the information of the names. Storage layer is an alternative layer for users to store the name pointing data.

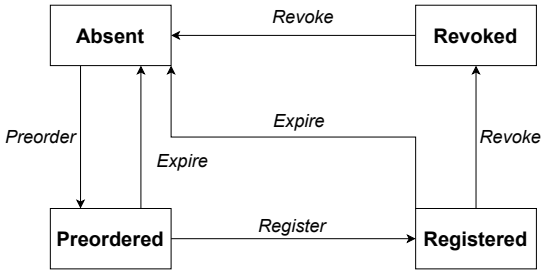



Fig. 7: Name state transitions

### B. Naming System

The design of the naming system is inspired by the implementation of Namecoin [3] and BNS of Blockstack [23], [25]. Names are owned by the cryptographic public addresses and the corresponding seed which generates the address of IOTA. In IOTA, addresses are generated by the user's private seed with Kerl Hash Function [33]. As shown in Fig 7, operations in the naming system perform the state transitions of names. Each name has four states: **absent, preordered, registered, and revoked**. And **preorder, register, revoke, and expire** operations transit the state of the name into one another. In

addition, as a registered name, **renew, update, and transfer** operations can be performed for changing its binding value.

To claim a name, a user needs to first preorder the target name and then register it. Preorder operation is for users to indicate interest in the name. The operation sends a transaction with the message containing the hash commitment of desired name rather than plain text. The reason for sending hash commitment is to prevent front-running [3], e.g., if a name is revealed to the public, an attacker can race the user for claiming it [25]. The preorder operation also acts as a signal for the nodes in the peer network to start the preprocessing of a name registration. After preorder, the user can send the register operation for providing the name he/she wants to claim, the transaction hash of the previous preorder transition, the keys for decrypting the hash commitment, and the value for name binding. The first user who finishes both preorder and register operation successfully will grant ownership of the name. And any preorder operations performed by the other users will be invalid. In Namecoin's implementation, `NAME_NEW` and `NAME_FIRSTUPDATE` operations are equivalent to the pre-order and register operations in Vivian. For the effectiveness of `NAME_NEW` and front-running prevention. The gap between `NAME_NEW` and `NAME_FIRSTUPDATE` transactions should be no less than 11 blocks [3]. In Vivian, we count the gap by timestamps between two transactions. The details will be described in Section IV.

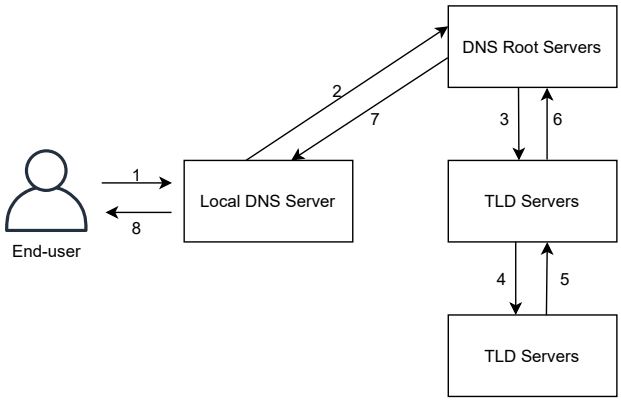
After a name becomes registered, the owner can send an update transaction for updating the name-value binding. He/She can also transfer the name to others by sending the transfer transaction to change the address that signs subsequent transactions. Revoke operation is used for disabling any further operations on the name. After that, the name-binding value cannot be changed until expiration.

A name in Vivian is not permanent and has a certain time for expiration. In Namecoin, the original time period for a name to expire is 36,000 blocks (around 250 days). It means if a name hasn't been mentioned in `NAME_UPDATE` or `NAME_FIRSTUPDATE` in 36,000 blocks, it becomes absent again [3]. In Vivian, the expiration time is counted by the timestamp of the register transaction. Besides, users can send the transaction for renewing their names before expiration.

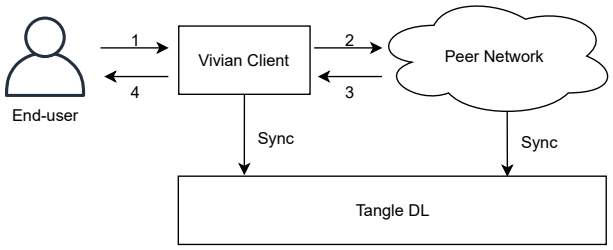
By using Tangle DL, most IoT devices have enough computational power to sign and send the transactions for naming operations .

### C. Peer Network

Nodes in the peer network form a routing layer for users to find the binding value of names. Every node will consistently listen to all the transactions on IOTA Tangle and filter the transactions that relates to Vivian naming operations. Then it parse the transaction and update the name key-value bindings in its local database. In the local database, basically, a node only needs to record the registered names and the address and hash of the latest transactions that modify a name binding value. A node can also cache the binding values of the names. When a node discovers a change of a name, e.g., a new name has been registered or a name's binding value has changed, it will record the change to the change list and periodically fan-out the change list to other k random neighbors via Gossip protocol [32]. If a node receives the change list from another node, it will compare the changes with items in the local database. If a difference is found, it will then choose the records with the latest transaction (through timestamp comparison) and validate it by checking the transaction on Tangle DL. If the change is validated, the node then updates the change to its local database and appends the change to its change list. If a user wants to get the binding value of a name, he/she can connect to one or more nodes in the network and query for name information. The node should return the name, the recorded transaction hash and cached binding value (for fast access if available). The end-user does not need to trust the node as the data integrity can be validated on Tangle DL via transaction hash. The whole peer network is established via Kademlia Distributed Hash Table (Kad-DHT) [30] and new nodes can join the network. Once a new node joins the network, it can construct all the naming information by iterate all the transactions on Tangle DL.




(a) DNS




(b) Vivian

Fig. 8: Comparison of DNS and Vivian queries

Through peer network routing layer, users can query the value of a name without relying on the trust of third parties.

[13]For more detail: https://docs.iota.org/docs/iot/0.1/introduction/overview

### D. Storage System

There is no restriction on what type of data a name should map to. The storage layer provides an alternative way for binding large files to the names. Due to data size limitation of Tangle transaction and complexity for checking the message content of the transaction, actual data are stored on the storage layer instead of Tangle DL. Users can choose to use InterPlanetary File System (IPFS) [34], a P2P decentralized file system, to save their files or traditional storage backends such as Amazon S3, Azure Blob, Google Cloud, etc. The design goals are to let the users own the access right of their data, users do not need to trust the storage layer and can verify the data integrity, and the other third parties like the storage providers cannot tamper with the data. Possible implementations will be discussed in Section IV.

## IV. Implementation

In this section, we introduce some existing implementations of Vivian and some other potential implementations for the planned features.

### A. Software & Dependencies

The whole project is currently developed in Golang v1.16. And the platform is divided into two parts: the node software, *vivian-node*, and the client software, *vivian-client*. In the future, there will be another version of the node software, which runs as a plugin of GoHornet, IOTA's node software built in Golang. The advantage of this version is for direct access to a network instead of connecting to and trust other parties' nodes.

*1) Node Software:* It sets up a node in the peer network layer. These nodes form and help to maintain a decentralized P2P network which let users to find naming information. Each node keeps track of the naming key-value bindings by listening to the live transactions on Tangle DL, filtering naming transactions, parsing transactions, and update related key-value information on local database. For listening live transactions, the current implementation subscribes "tx" (incoming new transactions) and "sn" (confirmed new transactions) topics of a IOTA node's ZMQ[14] port. In the plugin version, Vivian node is also an IOTA node, so that live transactions can be retrieved from local database directly. Besides, the node software is also responsible for handling user requests such as checking whether his/her naming operations are successful and , getting the name binding values, checking the Availability of a name, etc.

*2) Client Software:* It provides a user interface (currently CLI) for sending name operation transactions and requests to peer network nodes. When initializing the client software, it creates an IOTA seed and encrypt it with password or load a seed from a local encrypted file. Then users can use the seed to generate public addresses and send transactions for name registration, update, revoke, etc. The client software can also connect to the nodes in peer network layer, for finding the name binding value, and validating name operations.

*3) Dependencies:* In operating Vivian, we need to do a lot of work related to key-value pairs, and information is stored in *BadgerDB*[15]. We use libraries in *iota.go*[16] for handling transactions on Tangle. We use functions in *go-libp2p*[17] for peer discovery and peer Communication in the P2P network. Communication messages are serialized by *Protocol Buffers*[18] for better performance.

### B. Naming Operations

In this subsection, we introduce how naming operation information is recorded via transactions on Tangle. And taking name registration as an example, for having an overview of how the naming operations are executed.

*1) Name Operation Transactions:* There are two types of transactions in IOTA Tangle: zero-value transaction and value transaction. And currently we use zero-transaction format for name operation transactions. Zero-value transactions are barely transactions with value 0 in the `value` field. Data of transactions are encoded in trytes[19]. Each transaction consists of 2,673 tryte-encoded characters. After decoding the trytes, we can obtain the a list of data fields of a transaction. The fields related to naming operations are:

- `hash`: 81-tryte string. It is the transaction hash derived from the values of every transaction field and contains part of the proof-of-work. We can use transaction hash to track and validate the information of a transaction.
- `signatureMessageFragment`: 2,187-tryte string. It is a signature or a message, both of which may be fragmented over many transactions in a bundle. This field is used for saving the naming operation information, and name binding values. 2,187 trytes' size is enough for most of the use cases.
- `attachmentTag`: 27-tryte integer. It is user-defined tag. This field is used for accelerating the filtering of the live transactions. For fast selection of related transactions, users are required to set the corresponding two-character logogram for each naming operation transaction. And nodes in peer network layer can use tryte formats of the logograms for comparison and identifying name operation transactions without parsing `signatureMessageFragment` (see TABLE I).
- `address`: 81-tryte string. It contains either the sender's or recipient's address. The address is used for justify the ownership of a name.

[13]Golang website: https://golang.org/

[14]ZMQ: ZeroMQ, an asynchronous messaging library for distributed or concurrent applications by providing a message queue.

[15]BadgerDB: an embeddable, persistent and fast key-value (KV) database written in pure Go. Github repository: https://github.com/dgraph-io/badger

[16]iota.go: official Go client library for interacting with the Tangle. Github repository: https://github.com/iotaledger/iota.go

[17]go-libp2p: implementation of the libp2p networking stack in Golang. Github repository: https://github.com/libp2p/go-libp2p

[18]Protocol Buffers (Protobuf): a method of serializing structured data. Its performance is even better than JSON.

[19]Tryte: a concept in ternary numeral system similar to byte in binary numeral system. See details on: https://docs.iota.org/docs/getting-started/0.1/introduction/ternary

- `attachmentTimestamp`: 9-tryte integer. It is Unix epoch (milliseconds since Jan 1, 1970 after proof of work was done). The timestamp is used for counting the time of the operations, e.g. deciding if the current name is expired.

Proof-of-work in the previous description is different from the PoW consensus mechanism in Bitcoin blockchain. Proof-of-work in IOTA is the method for preventing spamming in the network, similar to Hashcash[20] in email system.

| Operation | Two-Char Value | Tryte Value |
|---|---|---|
| Preorder | PO | ZBYB |
| Register | RG | ACQB |
| Renew | RN | ACXB |
| Update | UD | DCNB |
| Transfer | TF | CCPB |
| Revoke | RV | ACEC |

TABLE I: Tags for fast name operation identification

*2) Name Registration:* For sending each naming operation transaction, the client software needs to generate a new address by the private seed. To register a name, the user should send name preordering first. In the current implementation, we use Pedersen Commitment [35] for committing the name. Client software creates and sends a zero-value transaction with tag "PO" and the message: "`PREORDER : ⟨Hash Commit Value⟩`". After the transaction is successfully sent, the client software will record the name, transaction hash, G, H, R value of Pedersen commitment to BadgerDB. IOTA nodes will not receive transactions with timestamps more than 10 minutes ago, so in order to prevent front-running, we set the time gap of preorder operation and register operation to be 15 minutes. After 15 minutes, the user can then send the transaction for name registration. Client software will send a zero-value transaction with tag "RG", and message containing the name, G, H, R of pedersen commitment, preorder transaction hash and binding value of the name. Nodes in the peer network layer keeps listening the live transactions on Tangle. If it detects a transaction with tag value in tryte "ACQB" with appending 9s, it will parse this transaction and check the format of the message. Then it use the information provided in the message field to validate name registration. Once the validation is confirmed, it records the name information on its local database and append this change to the change list. The change will be broadcast to other nodes in the network via Gossip protocol later.

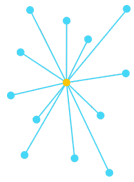
### C. Network

libp2p network stack provides very powerful functionalities for establishing a distributed network.

[20]Hashcash: a proof-of-work system used to limit email spam and denial-of-service attacks.

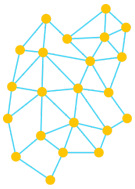
*1) Peer Discovery:* Following procedures are implemented in peer discovery:

1) **Bootstrap.** Nodes will often connect to a bootstrap node through its well-know IP address, to join the network for the first time.
2) **mDNS.** Nodes can discover peers under the same local network through multicast DNS.
3) **DHT Random Walk.** This process is for finding more peers in the network. First we make a random node ID in the network, and random walk is a query for finding the peer with that ID. We ask the peers we know whether they know the node with target ID or not. If they do not know, they will recommend the nodes with closer IDs (logic distance calculated by XOR). We continue this process until no closer node can be found.



(a) Bootstrap

(b) mDNS



(c) DHT Random Walk

Fig. 9: Network structure after each peer discovery procedure

*2) Peer Communication:* Vivian defines different libp2p protocols for different types of communications. Messages are serialized via Protocol Buffers for better transmission performance. For naming information updating among peers. We will going to implement Episub (similar to Gossip protocol) in libp2p pubsub protocols. Considering the message redundancy issue, we plan to store the recently changed naming key-value pairs in cache for fast loop-up when handling gossip message from neighbors.

### D. Storage

In this subsection we describe two potential implementations of file storage layer: via IPFS and via normal storage backends.

*1) IPFS:* IPFS is a decentralized peer-to-peer file system. When a file is uploaded to the system, it is split into many small blocks linked by a Merkle DAG. The file and all of its blocks are given a unique hash fingerprint. The blocks of a file will be stored on different peers in the network in a distributed way. Users can use the fingerprint of the file for content addressed searching for the file content. The whole system is decentralized and files stored on the system are immutable. However, the fingerprint of a file is a cryptographic

hash which is not human-meaningful. Besides, traffic between nodes and content of the blocks are not encrypted and are visible to public. In Vivian, if a user want to store his/her data via IPFS, he/she can first encrypt the file using GPG[21] and then upload to IPFS, then bind the fingerprint of the file to the name he/she owns.

*2) Normal Storage Backends:* Users can storage their data on normal storage Backends such as Amazon S3, Azure Blob and Google Cloud. The intuitive idea is to split the file into small chunks encrypted and/or signed by user. Then record the URI of these chunks on Tangle, and bind it to the name the user owns. The message field of a Tangle transaction is 2,187 trytes in size and can sufficiently records a large number of URLs. If the a single transaction is not enough to record all the URL, hierarchy tables can be implemented (similar to the idea of Bigtable by Google [36]). In this way, the storage providers are only able to see the encrypted file chunks, and cannot parse the actual content of the files.

## V. Evaluation and Discussion

One of the design goals of Vivian is to let IoT devices be able to use client software for naming operations and file storage. For naming operations, client software needs to handle transactions on Tangle and communicate with nodes in peer network layer. We evaluate the compatibility of the naming operations on IoT devices by testing the performance of sending and reading IOTA transactions on devices with limited hardware resources. The devices used for performance evaluations are:

1) Raspberry Pi Zero W:
   - Hardware: 802.11 b/g/n wireless LAN, 1GHz single-core CPU (ARMv6 architecture), 512MB RAM, with 16GB SD card.
   - Software: Raspberry Pi OS Lite.
2) Raspberry Pi 4 B:
   - Hardware: 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, 1.5GHz quad core Cortex-A72 (ARM v8) 64-bit CPU, 4GB RAM, with 16GB SD card.
   - Software: Ubuntu Server 20.04.2 LTS.

We prepared two sets of program for the test: sending transactions and reading transactions. They are all written in Golang (v 1.16) based on `iota.go`, the official Golang library for IOTA. Sending transaction program will generate a new seed and continuously send zero-value transaction for 100 times, there is 10 seconds' delay between each sending operation. There are three steps for each sending operation: getting new address, preparing transfer for creating a bundle in trytes, and sending trytes which handles tip selection, remote proof of work, and sending the bundle to the node. Please note that proof-of-work procedure here is different from PoW consensus mechanism in Bitcoin, it is a method for preventing spam in the network, and the workload is much less than PoW in Bitcoin. Reading transaction program will continuously read transactions according to different tail transaction hash for 100 times, there is 10 seconds' delay between each reading operation. There are two steps for each reading operation: getting the bundle to get all transactions in the tail transaction's bundle, and extracting JSON to decode the JSON messages in the signatureMessageFragment fields of the transactions. The programs measure the cost of each procedure by recording the execution time.

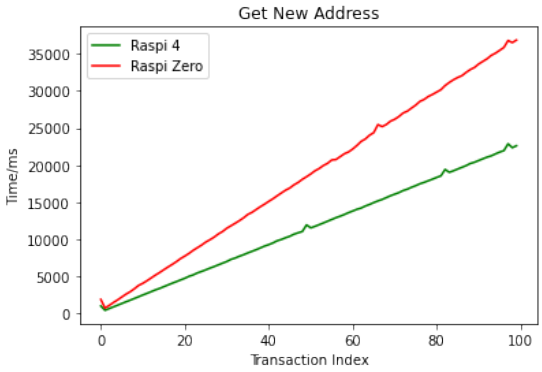



Fig. 10: Cost of getting new address procedure

The first step of sending a transaction is getting a new unspent address generated by user's private seed. From the result of cost of getting new address, we noticed that the time taken for getting a new unspent address is increasing significantly. We analyzed the possible reason for this result. The test program uses API `getNewAddress()` to generate new addresses, it will first generate new addresses with private seed locally and request IOTA nodes for checking if the address is a spent from a given index. If the address with given index if spent, the index in incremented until the nodes find an unspent address. As number of unspent addresses is increasing, time taken for finding an unspent one becomes longer. We will consider keeping track of spent addresses on local database for better management to reduce new address generation time in real implementation.

From the result of getting the bundle of transactions, we found that the time taken for the first request is much longer than the others. We think it is possibly due to IOTA node's orientation process from a new device. Fig 13 shows the time cost of the remaining 99 requests.

In summary, with Raspberry Pi 4, the mean time of sending a transaction (excluding getting new addresses) is 1037ms, and mean time for reading a transaction is 217ms. With Raspberry Pi Zero, the mean time of sending a transaction (excluding getting new addresses) is 1187ms, and mean time for reading a transaction is 255ms. We think this performance is good enough for real use cases in IoT services.

Due to the complexity of peer network communication and file storage, detailed evaluations are left for future work.

In this work, we proposed a decentralized approach for the current Internet services and Domain Name System to enhance better security and user data security. By removing central trust parties, the system has no single-point failure and users can

[21]GPG: GNU Privacy Guard. Open source alternative of Pretty Good Privacy (PGP)

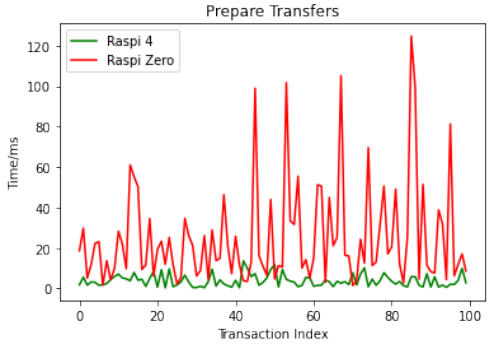



Fig. 11: Cost of preparing transfers

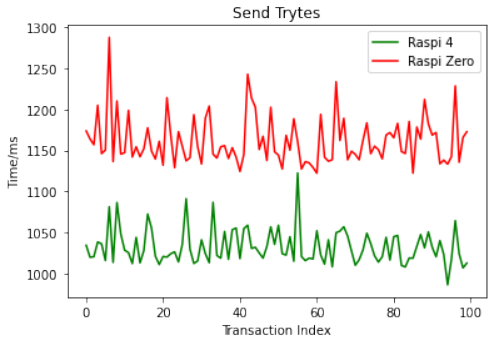



Fig. 12: Cost of sending trytes

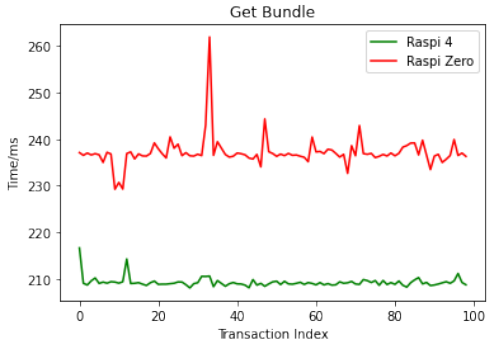



Fig. 13: Cost of getting bundle

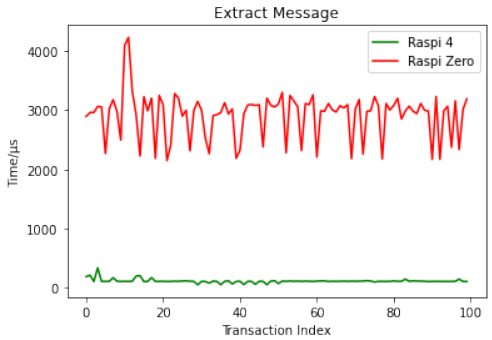



Fig. 14: Cost of extracting JSON

register human-meaningful names without relying on the trust of third parties. The system also provides an alternative storage layer for a decentralized and secure storage of user data. Decentralized naming system has been proposed for a long time and there are already existing implementations like Namecoin [3], Blockstack BNS [23] and Ethereum ENS[22]. However, these systems rely on PoW blockchains that require miners and huge energy consumption for Proof-of-Work consensus. Transaction fees need to be paid for miners' work. Also, the blockchains they are relying on are also facing scalability, low TPS and long transaction validation time issues. Instead, in our system, we decide to adopt another type of distributed ledger, DAG, for solving these issues. In addition, there is no miners in the network and confirming a transaction does not require much computational work. This makes the system more economical and environmentally friendly. Its high scalability, feeless, and lightweight characteristics also enables the ability of extending the system features on IoT devices.

Besides the design and implementation of the system included in this paper, there are more topics to be cover in the future: incentive to run a node in the peer network layer, pricing function for a name, how to ease node eclipse attack [37], etc.

## VI. Applications

Vivian is designed to be a infrastructure for reconstructing the current Internet services in a decentralized manner. It is built on Tangle distributed ledger for a better expandability of features in Internet-of-Things Services. To better understand how Vivian can be implemented in real-world scenarios, we list some potential use cases below:

- **Decentralized Website.** Users can build and maintain their website through Vivian decentralized naming system and storage layer. Compared with traditional websites, decentralized websites are censorship-resistance (only the owners can censor or modify the the content of their websites), more robust (it is much more difficult to take them down or DDoS), and private (the owner of the decentralized websites can be anonymous).
- **Identity Attachment.** Users can attach identity information such as GPG public keys, email address, cryptocurrency addresses that are not human-meaningful or easy to memorize to name they like.
- **Enhance IoT Scalability and Privacy.** Inferior scalability, server failure, large-scale data management, and data privacy are four of the weaknesses of the current IoT network implementation [38]. Traditionally, all the data is transmitted from a device or an object to central cloud servers where it is stored and analyzed. If the centralized server fails, the whole network is at risk of taken down. As more and more devices joining IoT network, scalability issues are getting worse. Also these devices are producing massive amounts of data including sensitive information, and large-scale data management and data privacy issues are becoming more severe. Vivian and

[22]ENS: Ethereum Naming System. Documents: https://docs.ens.domains/

IOTA can help to decentralize the current IoT network, and data can be stored on Tangle DL or storage layer provided by Vivian. This improves the scalability and data privacy of IoT services.

## VII. Conclusion

Current Internet services based on centralized data models have potential risk of single point of failure. And data storage relies on the trust of third parties such as service providers. This leads to user data integrity and privacy issues. In this paper, we introduce Vivian, a new decentralized global naming and storage system, which is a possible solution to the problems above. Vivian squares Zooko's Triangle trilemma and provides a decentralized naming system, that allows users to register human-meaningful names with binding information. Its storage layer also helps users to save their files in a decentralized and secure way. Unlike other blockchain based decentralized naming system, Vivian uses IOTA Tangle DL for critical data binding. Tangle is a lightweight and highly scalable DAG distributed ledger, which allows devices with poor computing power to write and send transactions on it. It extends Vivian's usage in IoT networks. In addition, the system is more environmentally friendly compared with other PoW blockchain applications as it does not require miners to do Proof-of-Work computations. We hope the design of Vivian can help inspire more innovations of DLT applications to make more impacts in different industries.

## References


[1] A. Sunyaev, *Distributed Ledger Technology*, pp. 265–299. Cham: Springer International Publishing, 2020.

[2] S. Nakamoto and A. Bitcoin, "A peer-to-peer electronic cash system," *Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf*, vol. 4, 2008.

[3] H. A. Kalodner, M. Carlsten, P. Ellenbogen, J. Bonneau, and A. Narayanan, "An empirical study of namecoin and lessons for decentralized namespace design.," in *WEIS*, Citeseer, 2015.

[4] F2b, "BitDNS and Generalizing Bitcoin." https://bitcointalk.org/index.php?topic=1790.0/, 2010. [Online; accessed 25-Feb-2021].

[5] B. M. Leiner, V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts, and S. Wolff, "A brief history of the internet," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 5, pp. 22–31, 2009.

[6] J. Postel, "Domain name system structure and delegation," *RFC*, vol. 1591, 1994.

[7] D. McGuire and B. Krebs, "Attack on internet called largest ever," *The Washington Post*, vol. 22, 2002.

[8] S. Rosenblatt, "Fake turkish site certs create threat of bogus google sites." https://www.cnet.com/news/fake-turkish-site-certs-create-threat-of-bogus-google-sites/, Jan 2013. [Online; accessed 27-Feb-2021].

[9] Z. Zhang, M. C. Y. Cho, C. Wang, C. Hsu, C. Chen, and S. Shieh, "Iot security: Ongoing challenges and research opportunities," in *2014 IEEE 7th International Conference on Service-Oriented Computing and Applications*, pp. 230–234, 2014.

[10] S. R. Department, "Internet of things - active connections worldwide 2015-2025." https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/, Jan 2021. [Online; accessed 28-Feb-2021].

[12] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, (New York, NY, USA), p. 3–16, Association for Computing Machinery, 2016.

[11] B. Farahani, F. Firouzi, and M. Lücking, "The convergence of iot and distributed ledger technologies (dlt): Opportunities, challenges, and solutions," *Journal of Network and Computer Applications*, p. 102936, 2020.

[13] S. Popov, "The tangle," 2015.

[14] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain technology overview," *arXiv: Cryptography and Security*, 2018.

[15] R. C. Merkle, "Protocols for public key cryptosystems," in *1980 IEEE Symposium on Security and Privacy*, pp. 122–122, IEEE, 1980.

[16] J. Sedlmeir, H. U. Buhl, G. Fridgen, and R. Keller, "The energy consumption of blockchain technology: beyond myth," *Business & Information Systems Engineering*, vol. 62, no. 6, pp. 599–608, 2020.

[17] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to scalability of blockchain: A survey," *IEEE Access*, vol. 8, pp. 16440–16455, 2020.

[18] S. Popov and Q. Lu, "Iota: feeless and free," *IEEE Blockchain Technical Briefs*, 2019.

[19] L. Tennant, "Improving the anonymity of the iota cryptocurrency," 2017.

[20] J. Buchmann and J. Ding, *Post-Quantum Cryptography: Second International Workshop, PQCrypto 2008 Cincinnati, OH, USA October 17-19, 2008 Proceedings*, vol. 5299. Springer Science & Business Media, 2008.

[21] A. Swartz, "Squaring the triangle: Secure, decentralized, human-readable names." http://www.aaronsw.com/weblog/squarezooko, Jan 2011. [Online; accessed 1-Mar-2021].

[22] Z. Wilcox-O'Hearn, "Names: Distributed, secure, human-readable: Choose two," 2001.

[23] M. Ali, R. Shea, J. Nelson, and M. J. Freedman, "Blockstack technical whitepaper," *Blockstack PBC, October*, vol. 12, 2017.

[24] J. Nelson, M. Ali, R. Shea, and M. J. Freedman, "Extending existing blockchains with virtualchain," in *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, 2016.

[25] M. Ali, J. Nelson, R. Shea, and M. J. Freedman, "Blockstack: A global naming and storage system secured by blockchains," in *2016 {USENIX} annual technical conference ({USENIX}{ATC} 16)*, pp. 181–194, 2016.

[26] R. Schollmeier, "A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications," in *Proceedings First International Conference on Peer-to-Peer Computing*, pp. 101–102, 2001.

[27] R. Nemat, "Taking a look at different types of e-commerce," *World Applied Programming*, vol. 1, no. 2, pp. 100–104, 2011.

[28] B. Cohen, "The bittorrent protocol specification." https://www.bittorrent.org/beps/bep_0003.html, Jan 2008. [Online; accessed 2-Mar-2021].

[29] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips, "The bittorrent p2p file-sharing system: Measurements and analysis," in *International Workshop on Peer-to-Peer Systems*, pp. 205–216, Springer, 2005.

[30] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," in *International Workshop on Peer-to-Peer Systems*, pp. 53–65, Springer, 2002.

[31] S. Cheshire and M. Krochmal, "Multicast dns," tech. rep., RFC 6762, February, 2013.

[32] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry, "Epidemic algorithms for replicated database maintenance," in *Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing*, PODC '87, (New York, NY, USA), p. 1–12, Association for Computing Machinery, 1987.

[33] B. Baek, "Iota: A cryptographic perspective," 2019.

[34] J. Benet, "Ipfs-content addressed, versioned, p2p file system," *arXiv preprint arXiv:1407.3561*, 2014.

[35] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Annual international cryptology conference*, pp. 129–140, Springer, 1991.

[36] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, "Bigtable: A distributed storage system for structured data," *ACM Transactions on Computer Systems (TOCS)*, vol. 26, no. 2, pp. 1–26, 2008.

[37] A. Singh *et al.*, "Eclipse attacks on overlay networks: Threats and defenses," in *In IEEE INFOCOM*, Citeseer, 2006.

[38] B. Farahani, F. Firouzi, and M. Luecking, "The convergence of iot and distributed ledger technologies (dlt): Opportunities, challenges, and solutions," *Journal of Network and Computer Applications*, vol. 177, p. 102936, 2021.