BNS - A Segregated Network System for Bitcoin

CTO of Rings Network
0xea@ringsnetwork.io

Architect of Rings Network
0xca@ringsnetwork.io

March 7, 2023

Abstract

BNS is a proposed system that aims to enhance the Bitcoin network by introducing a structured peer-to-peer network based on Distributed Hash Tables (DHTs), providing DHT storage capabilities and DID services [1]. BNS is developed using Rings [**?**], a modern browser-native P2P network that implements the Chord algorithm, which can provide extendable DNS services and secure network traffic for web2 and web3 applications on blockchains. Rings Network can be run on servers and browsers using WebAssembly [2], with both nodes being functionally equivalent.

This paper delves into the technical aspects of BNS, including its architecture, protocols, and algorithms, and highlights potential use cases for the system. Additionally, this paper presents the implementation of DID proofs, DID Controller, and scalable DID Document using Rings Network and showcases how the system's convenience can be utilized for applications such as MPC and atomic swaps.

1 Introduction and Motivation

Bitcoin was created in 2009 by an anonymous individual or group known as Satoshi Nakamoto [3]. The Bitcoin network operates as a decentralized and unstructured peer-to-peer network, with nodes communicating through a gossip protocol [4]. In this protocol, nodes broadcast messages to their neighbors, and these messages propagate throughout the entire network. This distributed communication allows for a trustless and transparent ledger of transactions, without the need for a central authority [5].

Because the Bitcoin network is unstructured, it can be challenging to implement end-to-end message routing [4]. To address this limitation, a newer version of the blockchain technology was developed, it features a built-in structured Distributed Hash Table (DHT) network and a virtual machine (VM) for more powerful scripting capabilities [6]. These additions allow for improved message routing and the execution of more complex smart contracts [7], and it's Ethereum.

Despite its advantages over Bitcoin, it still faces challenges. One issue is the problem of state explosion, which can cause the network to become slower and more expensive to use as more data is added to the blockchain [6]. Additionally, it's network implementation is based on the Kademlia algorithm, which makes it difficult to implement broadcast over the network [8]. These challenges have prompted research into alternative network designs that can address these issues while still providing the benefits of

decentralized and secure ledger systems.

Bitcoin was originally designed as a decentralized and secure ledger system, and it should ideally be used only for that purpose. However, as the technology has evolved, there have been attempts to expand its functionality to include storage, oracles, smart contracts, and other network-layer features. To prevent unintentional transaction malleability in Bitcoin, a separate system called BNS (Bitcoin Network System) has been introduced. BNS is built on top of the Bitcoin network and includes a structured peer-to-peer network based on Distributed Hash Tables (DHTs), DID services, and DHT storage capabilities. BNS also implements the Rings Network, which provides extendable DNS services and secure network traffic for traditional web2 and web3 applications on blockchains. With BNS, these additional functionalities can be integrated securely and without interfering with the integrity of the Bitcoin ledger.

1.1 Browser-native P2P network

BNS aims to enhance the Bitcoin network by enabling interoperability with other systems. This is achieved by providing a structured peer-to-peer network based on Distributed Hash Tables (DHTs) that is browser-native and can run on both servers and browsers using WebAssembly. The Rings Network, which implements the Chord algorithm, is used to provide extendable DNS services and secure network traffic for web2 and web3 applications on blockchains. By providing a structured network layer that can communicate with other systems, BNS enables greater interoperability and opens up new possibilities for the use of Bitcoin as a secure and decentralized ledger system.

And by providing a Distributed Hash Table (DHT), BNS can provide a decentralized way for implementing Lightning Network hubs, such as lndhub. The DHT can be used to store routing information and channel updates, allowing for more distributed and fault-tolerant Lightning Network deployments. Additionally, the Rings Network provides secure network traffic and extendable DNS services, making it easier to connect Lightning Network hubs with other decentralized applications and services. With these capabilities, BNS can provide a more scalable and reliable solution for Lightning Network implementations, enabling a more decentralized and resilient ecosystem for micropayments on the Bitcoin network.

1.2 Distributed Name Services

In addition to its DID capabilities, BNS also provides a distributed name service that is based on the ownership of a secp256k1 private key. This naming service allows users to register and manage human-readable names that can be used to represent complex blockchain addresses or other cryptographic identifiers. By associating these names with specific private keys, BNS provides a way for users to easily manage their blockchain identities and simplify the process of interacting with decentralized applications. The distributed nature of the naming service ensures that there is no central point of control, providing enhanced security and decentralization.

The figure **??** depicts the prototype of the Rings DHT, which enables the resolution of domains into various services, including direct messages with users, hosted websites, and provided services, among others.

1.3 DID Methods & Proofs

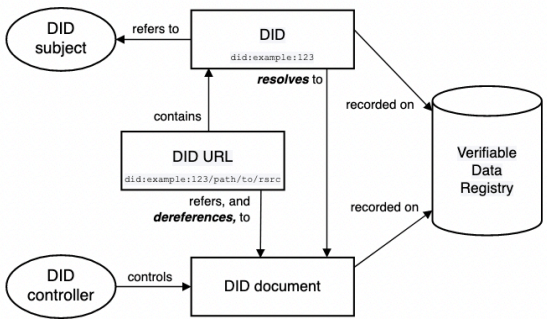

Figure 2: OvervieDID archiecture

BNS, with the assistance of the Rings Network, has the potential to be the most powerful DID (Decentralized Identifier) method in history. BNS meets the four requirements set by the W3C for DIDs: decentralization, persistence, cryptographic verifiability, and resolvability. The BNS DID schema is specified as follows: did:bns:<protocol>:<did>.btc, where the "protocol" component supports a variety of services.

DID proofs refer to how the relationship between the DID controller and the DID document is proved. For BNS, the DID document includes an additional part in the Rings DHT, where the DID controller can register and update the protocols and services provided by the DID through the Rings Network. BNS supports any kind of cryptographic proofs, including the implementation of secp256k1 and ed25519 curves. In contrast, only one out of 17 Ethereum-related DIDs has met the standards set by the W3C for DIDs.

Figure 1.3 shows an overview of the architecture for the BNS DID system, depicting the relationship between the DID controller, DID document, and DID subject. With its powerful capabilities and flexible DID proofs, BNS has the potential to revolutionize the way decentralized identifiers are implemented and used.

2 Related Work

Chord can provide the same service by converting each hostname to a key. Chord-based [9] DNS does not require special servers, while normal DNS relies on a special set of root servers. DNS requires manual management of routing information (NS records) that allow clients to navigate a hierarchy of nameservers; Chord automatically maintains the accuracy of analog routing information. DNS works correctly only if the hostname structure reflects administrative boundaries; Chord does not enforce a naming structure. While DNS is specialized for the task of finding named hosts or services, Chord can also be used to find data objects that are not tied to a specific machine.

Pastry is an overlay network and routing network for the implementation of a distributed hash table (DHT) similar to Chord. The key-value pairs are stored in a redundant peer-to-peer network of connected Internet hosts. The protocol is bootstrapped by supplying it with the IP address of a peer already in the network and from then on via the routing table which is dynamically built and repaired. It is claimed that because of its redundant and decentralized nature there is no single point of failure and any single node can leave the network at any time without warning and with little or no chance of data loss. The protocol is also capable of using a routing metric supplied by an outside program, such as ping or traceroute, to determine the best routes to store in its routing table.

Freenet is a peer-to-peer platform for censorship-resistant, anonymous communication. It uses a decentralized distributed data store to keep and deliver information and has a suite of free software for publishing and communicating on the Web without fear of censorship. Both Freenet and some of its associated tools were originally designed by Ian Clarke, who defined Freenet's goal as providing freedom of speech on the Internet with strong anonymity protection.

The Globe system has a wide-area location service to map object identifiers to the locations of moving objects. Globe arranges the Internet as a hierarchy of geographical, topological, or administrative domains, effectively constructing a static worldwide search tree, much like DNS. Information about an object is stored in a particular leaf domain, and pointer caches provide search shortcuts. The Globe system handles a high load on the logical root by partitioning objects among multiple physical root servers using hash-like techniques. Chord performs this hash function well enough that it can achieve scalability without also involving any hierarchy, though Chord does not exploit network locality as well as Globe.

PNRP(Peer Name Resolution Protocol) is a peer-to-peer protocol designed by Microsoft. PNRP enables dynamic name publication and resolution and requires IPv6. PNRP was first mentioned during a

presentation at a P2P conference in November 2001. Other hosts can then resolve the peer name, retrieve the corresponding addresses and other information, and establish peer-to-peer connections. Internally, PNRP uses an architecture similar to distributed hash table systems such as Chord or Pastry. The cache maintenance algorithm ensures that each node maintains adequate knowledge of the "cloud". It is designed to ensure that the time to resolve a request varies as the logarithm of the size of the cloud. PNRP now works on the released Windows systems but no evidence shows it could be used on other platforms.

DoX is a peer-to-peer DNS networking method for detecting and correcting inaccurate DNS records caused by cache poisoning attacks. DoX uses Chord to store and synchronize hostnames and keys. Only detecting or correcting is the solution after the problem occurs. We believe that using crypto algorithms to verify and transfer data among nodes is better and safer. Two-way hostname data synchronization between DNS network and blockchain datastores is necessary for long-term data storing and full decentralized implementation.

Onion Routing(the core principle of Tor [10]) is implemented by encryption in the application layer of the communication protocol stack. As Tor cannot encrypt the traffic between an exit node and the target server, any exit node is in a position to capture traffic passing through it that does not use end-to-end encryption such as Secure Sockets Layer (SSL) or Transport Layer Security (TLS). Shreds of evidence show that hackers can exploit usernames and passwords for email accounts by operating and monitoring Tor exit nodes.

3 System Design

This chapter will describe the BNS system and protocol in three parts. First, we will introduce how BNS's DHT can be used to decentralize Lightning Network hubs, such as lndhub. Second, we will introduce our distributed name system, which allows users to register and manage human-readable names that are associated with specific private keys. Finally, we will describe the lookup protocol used by BNS to enable efficient and reliable lookups of DIDs and other distributed resources. With these three components, BNS provides a powerful and flexible platform for building decentralized applications and services on top of the Bitcoin network, enabling a more secure and resilient ecosystem for blockchain-based innovation.

3.1 Distributed hub for lightning

BNS implements Lightning Network hubs using the storage features of the Rings Network. The Rings Network provides a structured peer-to-peer network based on Distributed Hash Tables (DHTs), which can be used to store routing information and channel updates for Lightning Network transactions. This approach provides a more decentralized and fault-tolerant solution for implementing Lightning Network hubs, enabling better scalability and reliability for micropayments on the Bitcoin network. Additionally, the Rings Network provides secure network traffic and extendable DNS services, making it easier to connect Lightning Network hubs with other decentralized applications and services. With these capabilities, BNS provides a more powerful and flexible platform for building Lightning Network applications, paving the way for a more decentralized and resilient ecosystem for Bitcoin micropayments.

To implement a decentralized Lightning Network hub using BNS and the Rings Network, it is necessary to first register the hub's public key and network address as a DID in the BNS system. This can be done using the BNS DID schema: did:bns:<protocol>:<did>.btc. The hub's routing information and channel updates can then be stored in the Rings Network's DHT, which provides a more resilient and fault-tolerant storage solution than local data storage.

When a Lightning Network transaction is initiated, the transaction can be routed through the Rings Network's DHT, using the registered DID as a key to look up the appropriate routing information and channel

updates. This decentralized approach to Lightning Network routing enables greater scalability, improved security, and enhanced reliability, making it easier to build and deploy Lightning Network hubs that are more decentralized and resilient.

3.2 Name Services of BNS

Our system handles lookups at the granularity of resource record sets (RRSets), as in conventional DNS. An RRSet is a list of all the records matching a given domain name and resource type. DNSSEC uses public-key cryptography to sign re-source record sets. When we retrieve an RRSet from an arbitrary server, we need to verify the signature (included as a signature (SIG) record). To find the public key that should have signed the RRSet, we need to execute another DNS lookup, this time for a public key (KEY) RRSet. This RRSet is in turn, signed with the public key for the enclosing domain.

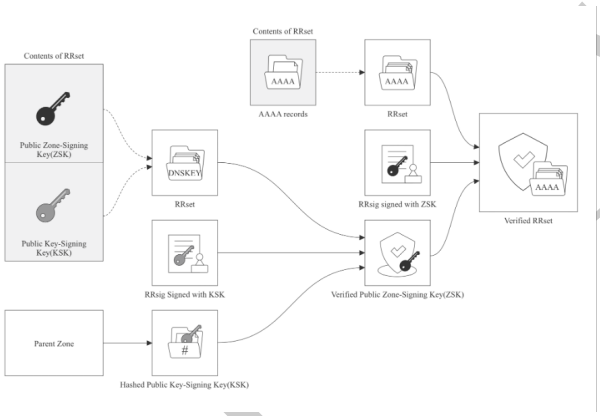

Figure 4: dDNS

dDNS stores and retrieves resource record sets using DHash, a Chord-based distributed hash table. DHash has two properties useful for this discussion: load balance and robustness.

DHash uses consistent hashing to allocate keys to nodes evenly. Further, as each block is retrieved, it is cached along the lookup path. If a particular record is looked up n times in succession starting at random locations in a Chord ring of m nodes, then with high probability, each server transfers a given record only log m times total before every server has the record cached. DHash is also robust: as servers come and go, DHash automatically moves data so that it is always stored on a fixed number of replicas (typically six). Because the replicas that store a block are chosen in a pseudo-random fashion, a very large number of servers must fail simultaneously before data loss occurs.

To create or update a DDNS RRSet, the owner prepares the RRSet, signs it, and inserts it into DHash. The key for the RRSet is the SHA1 hash of the domain name and the RRSet query type.

Naively verifying a DNS RRSet for a name with n path elements requires n KEY lookups. We address this problem by allowing the owner to present additional relevant KEYs in the RRSet. To avoid inflating the responses, we can omit KEY RRSets for popular names.

To ease the transition from conventional DNS to our system, a simple loopback server listening on 127.1 could accept conventional DNS queries, perform the appropriate Chord lookup, and then send a conventional response. Then systems could simply be configured to point at 127.1 as their name server.

3.3 BNS lookup protocol

BNSLS (BNS lookup service) is based on Rings Network, which queries, searches, and resolves services through Rings DHT. Since Rings Network's network layer is based on WebRTC [11] and WASM, BNS provides a browser-based, DID-to-DID access channel. Figure **??** shows how the node is connected directly via RTCPeerConnection. An RTCPeerConnectioninstance allows an application to establish peer-to-peer communications with another RTCPeerConnection instance in another browser, or to another endpoint implementing the required protocols. Communications are coordinated by the exchange of control messages (called a signaling protocol) over a signaling

channel which is provided by unspecified means.

Based on the Rings DHT, the BNS allows the DIDs to better lookup each other, which may include the person behind the DID, the machine behind the DID, the service, or a subnet consisting of a Rings network - which will usually also have a DID.

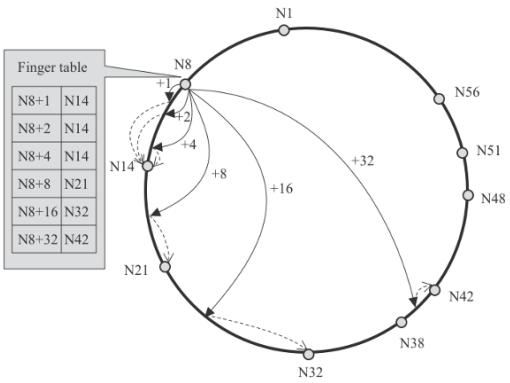

Figure 5: Rings DHT

Rings DHT is a variant of Chord DHT that stores the BNS domain on a ring structure and allows the BNS controller to register and discover other services. The 5 figure shows how a rings node stores nearby DIDs.

3.4 DNS Provider

BNS supports a wider range of applications and scenarios by providing web extensions and JSON-RPC. As the Rings Network is a browser-native network, it enables many applications based on WebAssembly (WASM) to run directly in the user's browser.

BNS Providers run a full node of the Rings Network on the backend and inject themselves into window.rings.bns, allowing users to directly access BNS's DNS resolver through the Rings Provider, thereby enabling access to the desired decentralized applications.

3.5 DID Protocol

As a DID Method, BNS provides a scalable and secure method for managing DID Documents by leveraging both distributed ledger technology and a decentralized network. This allows for a more reliable and secure solution compared to traditional centralized approaches. The BNS DID syntax conforms to the specifications defined by the World Wide Web Consortium (w3c) and extends to support various protocols and DID URLs. The integration of Rings Network into BNS further enhances its capabilities by abstracting the DID into a mathematically finite ring. This enables BNS to easily perform complex cryptographic operations, such as constructing secret sharing schemes based on Shamir's Secret Sharing (SSSS) technique. Additionally, this abstraction also allows for greater privacy and security as well as a more decentralized approach to DID document management.

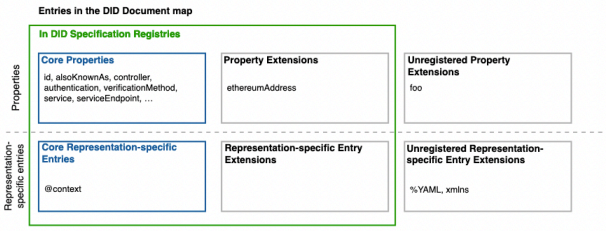

Figure 6: Did Document

figure 7 shows the basic structure of a DID Document.

BNS offers a flexible extension to DID Documents, using the BNS DID Controller, users can implement decentralized web and service registration, service discovery, and other functions through the Rings Network. Additionally, interactions such as atomic swaps and multi-signatures can be achieved through DID discovery between DID Documents.

4 Applications

BNS as a DID Method offers very big possibilities for dweb and web3. We have also implemented several simple demos for it, including BNSChat, BNSWeb and atomic swap.

BNSChat is a secure and decentralized chat application that employs end-to-end encryption to facilitate direct browser-to-browser communication. Utilizing a Decentralized Identifier (DID) as a decentralized identity solution, users can easily establish peer-to-peer encrypted communication channels without the need for intermediaries. In scenarios where access to third-party nodes is not feasible, communication can still occur through the direct exchange of Session Description Protocols (SDPs).

Figure 7: BNSChat

Moreover, BNSChat utilizes the Rings Network to facilitate highly efficient interactions with other BNS holders, incorporating all the functionalities of an instant messaging platform and supporting atomic swaps, which will be further discussed.

BNS Mailbox: BNS implements anonymous email functionality through the Rings Network. When a user needs to send an email to a DID, they simply send the email to the anonymous mapping of that DID in the Rings DHT network. This process does not reveal any real information. The message is delivered to the corresponding Mailbox via a Relay, and the recipient retrieves the message by proving ownership and decrypting it from the Mailbox.

Retrieving messages through a DID is a safer method than direct messaging. BNS Mailbox helps DIDs establish highly anonymous and secure new channels for messages.

BNS Web is a decentralized web implementation modeled after the Tor network that enables users to access web information anonymously and bypass censorship. While BNS Web represents a small component of the BNS Service Register/Lookup Protocol, the protocol also offers Oracle API services based on BNS and the Rings Network, including APIs similar to those offered by Ethereum [12].

Web service providers, akin to the role of hidden services in the Onion Network, can use the Rings Network's service registration feature to register their DID-associated services to the DHT network, ensuring their services remain online permanently. This register/lookup process simply involves the DID controller modifying the DID document.

For instance, in BNS Web, a node might provide IPFS [13] services, and other nodes can discover these services through service discovery and directly access IPFS data through their browser. This is more convenient compared to traditional IPFS access methods that necessitate running an IPFS node on a local computer or relying on a centralized server.

The Rings Network's chunking and obfuscation of traffic effectively thwarts censorship and surveillance from centralized organizations on DIDs, significantly boosting privacy.

BNS Secret Sharing

Based on the Rings Network, BNS can implement Secret Sharing using ElGamal [14] and Rings DID.

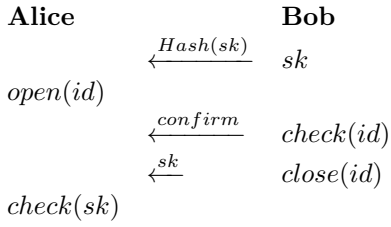

Figure 8: swap

This is commonly referred to as the SSSS algorithm. An example is where a secret is divided into 3n parts, and it can only be revealed when more than 2n + 1 participants are willing to participate in the interaction.

The requirement for participants to engage in multiple rounds of interaction increases the difficulty of implementing secret sharing based on the SSSS algorithm. The lack of decentralized channels further adds to the feasibility of the SSSS algorithm on BNS and the Rings network.

BNS Atomic Swap Atomic swap is a pioneering cross-chain algorithm that has faced challenges in widespread adoption due to the need for frequent interactions and the absence of decentralized channels. For instance, classical atomic swaps require price negotiation, the exchange of swap IDs, and other similar steps. BNS effectively addresses these challenges through its robust decentralized peer-to-peer network.

Classical atomic swaps can be broken down into the following steps, as described in [15]:

- Both Alice and Bob agree on the terms of the swap, including the cryptocurrencies to be exchanged and the exchange rate.
- Alice generates a hash of the secret key and sends the hash to the other party, along with the terms of the swap.
- Bob confirms that it agrees to the terms of the swap, and sends its own hash of the secret key to the first party.
- Both Alice and Bob use the hashes to create a smart contract on their respective blockchains, locking in the swap.
- When both Alice and Bob have confirmed that the other party's funds have been locked in the smart contract, they can reveal the secret key.
- The atomic swap executed, automatically exchanging the funds.

We briefly describe the close, expire algorithm in atomic swaps.

Algorithm 1 Close the Atomic Swap

function CLOSE($swapID$, $secretKey$)
    Input: $swapID$ - the identifier of the atomic swap, $secretKey$ - the secret key associated with the swap
    Requirement: The swap with identifier $swapID$ must be in 'Open' state and the caller must have the secret key associated with the swap
    Output: Closes the atomic swap and transfers the funds
    $swap \leftarrow swaps[swapID]$
    $swaps[swapID].secretKey \leftarrow secretKey$
    $swapStates[swapID] \leftarrow$ States.CLOSED
    $swap.withdrawTrader.transfer(swap.value)$
    Trigger: Close event with parameters ($swapID$, $secretKey$)

Algorithm 2 Expire

function EXPIRE($swapID$)
    $swap \leftarrow swaps[swapID]$
    $swapStates[swapID] \leftarrow States.EXPIRED$
    $swap.ethTrader.transfer(swap.value)$
    emit Expire($swapID$)

Algorithm 3 check(bytes32 swapID)

function CHECK(swapID)
    swap ← swaps[swapID]
    return (swap.timelock, swap.value)
    return (swap.withdrawTrader, swap.secretLock)

Algorithm 4 checkSecretKey()

```
1: function checkSecretKey(bytes32 swapID)
2:     public view onlyClosedSwaps(swapID)
3:     returns (bytes memory secretKey)
4:     Swap memory swap = swaps[swapID]
5:     return swap.secretKey
```

5 Conclusion

Rings network aims to build a new generation of Internet infrastructure, integrating web2.0 and web3.0 (especially blockchain technology) to provide a new generation of decentralized Internet solutions. Humanity created the first generation of the Internet out of the need for easy information sharing. The broader need gave rise to infrastructures such as large server rooms, submarine fiber optic cables, and satellite communications. The human nature of creation and communication allowed web 2.0 to develop, but the basic functionality no longer met the demand. People need new financial systems and privacy systems. We desire independence, privacy, and ownership, we want knowledge to be preserved and passed on, and blockchain technology gives us hope.

Web3.0 shows us a new chapter of the Internet, where people expect a fairer, more democratic, and more accessible Internet in the face of political, geopolitical, nationalistic, and economic sanctions that block communication. rings network hopes to build a new generation of decentralized peer-to-peer networks based on cryptographic systems to make all this possible. We are just moving a small step forward on the shoulders of giants.

We do not oppose web2, blindly oppose centralization, nor do we oppose government regulation to protect the interests of the people. We respect all anarchists and those who oppose anarchism. However, we insist that technology is innocent. Those who create advanced technology are innocent. Only those who abuse technology to infringe on the rights and interests of others are guilty. Technology itself should not bear any crime.

The original intentions of the blockchain and the new generation of decentralized networks starting from Bitcoin [**?**] are all good intentions, but we firmly believe that the principle of "Don't trust, verify it." can give us maximum security. We can't trust anyone (including regulators, of course) to be kind, and to do the most optimistic things with the most pessimistic and malicious speculations will give us a brighter future.

The technology that can protect the privacy of the bad guys can protect the privacy of the good guys. Technology that can easily invade the privacy of bad guys can easily invade anyone's privacy. Cyber privacy is already a basic human right now, and we agree to bring the bad guys to justice, but not at the expense of everyone's rights.

We hope to provide web3 users with a sense of cybersecurity.

References

[1] W3C. Decentralized identifiers (dids) v1.0. `https://w3c.github.io/did-core/`, 2021.

[2] WebAssembly Community Group. WebAssembly, 2017.

[3] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Bitcoin.org*, 2008.

[4] Shehar Bano, Alberto Sonnino, Mustafa Al-Bassam, Sarah Azouvi, Patrick McCorry, Sarah Meiklejohn, and George Danezis. Consensus in the age of blockchains. *arXiv preprint arXiv:1711.03936*, 2017.

[5] Andreas Antonopoulos. *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*. O'Reilly Media, Inc., 2014.

[6] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. In *Proceedings of the 2014 ACM conference on Computer supported cooperative work and social computing*, pages 455–462. ACM, 2014.

[7] Vitalik Buterin. A next-generation smart contract and decentralized application platform. *Ethereum*, 1(1):1–2, 2014.

[8] Arthur Gervais, Ghassan O Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. Security and scalability challenges in decentralized online social networks. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 435–450. IEEE, 2016.

[9] D. Karger I. Stoica, R. Morris. Chord: A scalable peer-to-peer lookup service for internet applications. `https://dl.acm.org/doi/10.1145/383059.383071`, 2001.

[10] Nick Mathewson Roger Dingledine and Paul Syverson. Tor: The second-generation onion router. *ACM Transactions on Computer Systems*, 22(3):303–327, 2004.

[11] Internet Engineering Task Force. WebRTC: Real-Time Communications between Browsers, 2021. Work in Progress.

[12] Gavin Wood. A next-generation smart contract and decentralized application platform, 2014.

[13] Juan Benet. Interplanetary file system: A p2p file system for the next web. *Proceedings of the 24th International Conference on World Wide Web*, pages 1149–1160, 2015.

[14] T. ElGamal. A public-key cryptosystem and a signature scheme based on discrete logarithms. `https://ieeexplore.org/abstract/4726576`, 1985.

[15] Confio. eth-atomic-swap. `https://github.com/confio/eth-atomic-swap`, 2021.