A Survey on Content Retrieval on the Decentralised Web

NAVIN V. KEIZER, University College London, UK
ONUR ASCIGIL, Lancaster University, UK
MICHAŁ KRÓL, City, University of London, UK
DIRK KUTSCHER, The Hong Kong University of Science and Technology (Guangzhou), China
GEORGE PAVLOU, University College London, UK

The control, governance, and management of the Web have become increasingly centralised, leading to several issues which include a lack of security and privacy protection, as well as censorship. To combat this, a number of decentralised initiatives have emerged that offer decentralised counterparts of various components of the Web, starting initially with content storage systems, *i.e.* decentralised file systems. Over the years, decentralised file systems have gained increasing popularity: a major file system has recently reported serving tens of millions of content requests per day [55, 176].

Decentralised file systems have also been slowly complemented with two other important components, namely decentralised search engines and decentralised name registry infrastructures—these three components together form the foundations for a *Decentralised Web* (DWeb). In this survey paper, we analyse research trends and emerging technologies used for decentralising content retrieval on the DWeb. We focus our scope and discussion on both research literature, while also surveying existing industrial projects with sufficient technical details relevant for researchers in the area.

Our analysis highlights several open issues, which need to be overcome to realise a truly decentralised Web. For search engines, achieving good performance (*i.e.* comparable to the current, centralised search engines) without sacrificing decentralisation is a major challenge. A recent promising direction is to employ hybrid infrastructures that use centralised components for performance-demanding tasks, but with verifiability (*i.e.* accountability) mechanisms, to achieve good performance with some level of security. On the other hand, the state-of-the-art decentralised file systems achieve secure content retrieval without a trusted third party. Still, these systems face both usability, *e.g.* lack of human-readable and persistent names for (mutable) content, performance and privacy challenges. A promising direction is to combine these file systems with decentralised name registries for improved usability. Our analysis also highlights the need for further research to analyse the security of decentralised name registries and possibly enhance them with better governance and crypto-economic incentive mechanisms.

Additional Key Words and Phrases: Decentralised Web, DWeb, Web3, Blockchain, Peer-to-Peer

1 INTRODUCTION

Over the past decade, the World Wide Web has become a major part of people's lives. The Web supports the global economy, provides entertainment and is often the main source of information about the world [197]. Furthermore, the Web has a tremendous impact on shaping people's views, opinions, and choices [7].

In recent years, the infrastructure providing the core web services on the Internet has become increasingly consolidated with a handful of players controlling the majority of the market [15]. While these players provide outstanding services and quality-of-experience (QoE) for users, their centralised model of service delivery has introduced several drawbacks such as lack of transparency [36], lack of privacy-protection [50], a single point of failure [171], and censorship [69].

Authors' addresses: Navin V. Keizer, navin.keizer.15@ucl.ac.uk, University College London, UK; Onur Ascigil, o.ascigil@lancaster.ac.uk, Lancaster University, UK; Michał Król, michal.krol@city.ac.uk, City, University of London, UK; Dirk Kutscher, dku@hkust-gz.edu.cn, The Hong Kong University of Science and Technology (Guangzhou), China; George Pavlou, george.pavlou@ucl.ac.uk, University College London, UK.

Recent initiatives in research and industry aim to tackle these issues by creating an open and decentralised Web (DWeb), also known as *Web 3.0*, which aims to fix the problems that come with centralisation—in particular, they focus on openness, security by design, and decentralised governance and control. This is achieved by using transparent, open-source software and peer-to-peer (P2P) networking [126], allowing anyone to join and contribute to the system. Furthermore, tools like blockchains [185], proofs of work [97] and self-certification through content addressing [24] are used to establish trust between anonymous users and reliably reward system contributors.

While the objective of the DWeb is to achieve decentralisation—*i.e.* redistribution of ownership and control from centralised infrastructures to individual users—it is an open question whether this can be achieved in practice. Current Web centralization is driven by economic concentration, and it is unclear whether the same would happen to the DWeb. Furthermore, interacting with untrusted, anonymous peers requires additional security mechanisms that are difficult to design and can lower the overall performance of the system. Finally, the current centralized model emerged from ad-monetised services usually delivered with high quality-of-service (QoS) to users without monetary compensation. Although end-users do not directly pay for these centralised services, the service providers collect user-related data to display targeted advertisements, making the ecosystem economically viable [129]. To be successful, the DWeb would have to reward service providers and content creators while combating users' intrinsic reluctance to spend money.

1.1 Contributions

In this paper, we provide a survey on content retrieval on the DWeb. We explore whether the decentralisation objective is realised by investigating the *incentive structures*, as well as the *performance, security and privacy* aspects of the content retrieval process (Fig. 1): starting with decentralised search engines, decentralised name-registries, and finally decentralised file systems.

We identify these focus areas as key components for which decentralised alternatives need to be developed. For each of these, we first describe the status quo, *i.e.* how operations are performed in the current Web. We then compare them with state-of-the-art decentralised implementations and proposals from both academia and industry. We use insights gained throughout the process to define a number of *open issues*.

Many of the discussed platforms lack clear documentation and a vision of integration to realise a DWeb. Furthermore, terms used in documentation differ greatly across projects and the fast development pace in the field makes obtaining a clear view and deep understanding challenging. With this work, we hope to clear up some of the contradictions and confusion. By defining a clear framework, we help to provide a big picture to understand and define future research opportunities.

1.2 Scope

While the documentation of novel DWeb projects is often scarce, their underlying concepts are usually derived from an extensive body of research. In this work, we utilise this underlying literature for background but do not go in-depth into the specific implementations. Rather, we focus on recent initiatives over the period 2009-2023 that have produced working implementations, as well as research proposals. While we give an overview of how components are handled in the current Web, we do not mention specific centralised solutions, except when this is appropriate for comparisons. Furthermore, this paper highlights architectures, their properties, and their aims. However, it is too early to definitively conclude that they can live up to their claimed potential, and we have added this nuance in the open issues. This work mainly serves as a general analysis of the DWeb at large, a framework for analysing and implementing new initiatives, and the first comprehensive body of work looking at DWeb technologies and their role in content retrieval analogous

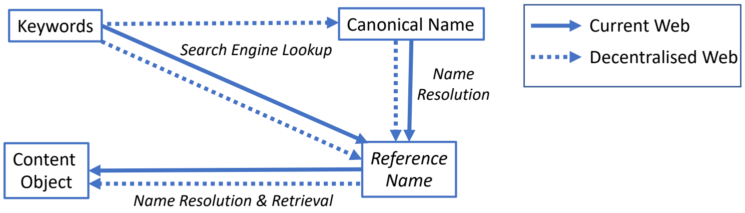

Fig. 1. Decentralised content retrieval process.

to the current Web. This survey is therefore relevant both to industry practitioners and researchers who aim to get a better understanding of the field at large.

1.3 Methodology

In order to survey a relevant body of work we started by querying research search engines (*e.g.* Google Scholar) for works which contain *{decentralised + Web}* in their title, keywords, or abstract. We also queried for *{distributed + Web}*, which generally returned works of the prior P2P era. We used these earlier works, as well as general *{Web + content retrieval}* works to identify key components and determine our framework. We also looked at related work specifically in our key components of *{search engine, name registry, file system}*, and surveyed works which combined components with keywords like *{Web3, blockchain}*.

Besides academic works, we surveyed industry works which included white papers, yellow papers, blog posts and more. We paid particular attention to works which were additionally cited in academic sources, in order to curate a high-quality body of work without marketing focus, obscure or incorrect jargon, and over-optimistic claims. To verify quality, we did manual inspection and selection. We highlight industry works because the area is still rapidly developing and many concepts have not made it into the formal research stage yet. We always inspect the underlying technology and design and check third-party sources to ensure objectivity.

The rest of this survey is structured as follows. In Section 2, we give an overview of Web content retrieval, provide a timeline of advancements, and present a systematisation framework used for structuring this survey. We subsequently provide general background on key concepts in Section 3. In Section 4 we discuss search engines, in Section 5 we describe name-registry, and in Section 6 we examine decentralised file systems. Finally, we review related work in Section 7, and summarise our key findings and conclude the paper in Section 8.

2 WEB CONTENT RETRIEVAL

In this section, we describe the process of retrieving Web content, discuss the need for decentralisation, and define our systematisation framework.

2.1 Retrieving Content on the Current Web

Content retrieval on the current Web involves a multi-step process. Often, a search-based workflow is used, where users submit a query to their favourite search engine with a description of a content object of interest in the form of a few keywords. The description may include a content creator or publisher name, a real-world description of the content, and more. In turn, a search engine returns results consisting of Web references; that is, the *Uniform Resource Locators (URLs)* such as *https://example.com/category_B/subcategory_C/Foo/*.

	Current Web	***Decentralised Web (DWeb)***
Trust	Centralised Root of Trust	Distributed Trust Model
Retrieval	Location-Centric	Content-Centric
Addressing	URL	Hash-based
Infrastructure	Centralised Entities (e.g. DNS)	Node Resource Sharing
Benefits	Performance, Accessibility, Scalability	Censorship Free, Availability
Drawbacks	Power Imbalance, Transparency, Privacy, Replication	Usability, Incentivisation, Interoperability

Table 1. Comparison of the current and decentralised Web.

With its "hostname/pathname" structure, a URL referencing a content object embeds both the hostname of the content's provider and the (server-specific) location of the object within the directory structure of the hosting provider's server(s). As a result, moving a content object to a different provider invalidates existing reference names to the content. Furthermore, replicating an object across different servers requires duplicating server-specific directory structures across different servers; this makes both replication and movement of content a difficult task in the current Web [186] and has led to increased centralisation.

Once a user obtains a valid URL of a content object, the next step in the content retrieval process is to perform a name resolution on the hostname component of the URL to obtain the storage location of the content provider. In the current Web, the Domain Name System (DNS) performs the name resolution service through a distributed database storing mappings from domain names (*i.e.* hostnames) to IP address (*i.e.* location) of hosts. Once a host location is resolved through the DNS, users can retrieve a content object.

In the current web, content producers increasingly rely on Content Distribution Networks (CDNs) for large-scale content distribution such as video streaming to a large number of geographically distributed users. These networks use proprietary technologies to serve content requests using a distributed infrastructure of content caches. Although CDNs achieve scalable content distribution using a distributed system of centrally-controlled caches, one can argue that the need for CDNs in the current web stems from the lack of a viable decentralised content delivery technology. Several DWeb projects [24, 170, 199] aim for replacing the CDNs with decentralised file systems, which we discuss in Section 6.

The current host-centric content retrieval ecosystem on the Web is exposed to serious flaws and vulnerabilities because of centralisation in control and ownership of the entities involved such as the search engines, the DNS, and the content storage (*e.g.* Cloud) providers. At present, there is a large power imbalance between these centralised entities and users, which allows these centralised parties to influence users by adding bias and censorship, tracking and selling personal data, influencing public opinion, and so on. The users are expected by default to trust these centralised entities unconditionally (*i.e. a centralised trust model*), while they operate without much transparency.

2.2 Retrieving Content on the DWeb

In *a distributed trust model*, the content retrieval can no longer depend on trusted third parties (*e.g.* a single root of trust as in a Public-Key Infrastructure [PKI] or DNS). Instead, the users must ideally be able to verify each step of the content retrieval process (Fig. 1). For example, the users must be able to verify the binding between the contents of a retrieved data object and its reference name; that is, to verify that the object is the correct one for the given reference name, without a centralised, third-party vouching for its provenance (*i.e.* the data object came from the appropriate source). This verification can be achieved by technical solutions such as self-certifying names and zero-knowledge proofs [70]. We further elaborate on the challenges and tools that can be used to establish distributed trust in Section 3.

The DWeb aims to evolve the current Web away from a host-centric paradigm and instead use *a content-centric paradigm* where reference names (*i.e.* content identifiers or *CIDs*, for short) directly identify content objects (also referred to as content addressing). This allows retrieval of content objects from anywhere in the network, rather than being restricted to retrieving them only from one of the content providers' locations. The location-independence of this paradigm is important because frequent replication and migration of content is the expected norm in the DWeb. Furthermore, decentralised services are realised by nodes in the network who share their resources for the network to outsource tasks like storage, computation, and bandwidth to them. Incentives and rewards play an important role in ensuring fair compensation for resource sharing and mitigating against malicious entities. Table 1 provides a brief comparison of key differences between the current and decentralised Web.

We envision a similar search-based workflow to take place in the DWeb, starting with decentralised search engines. Because CIDs are typically not human-readable for reasons of security[1] (see Section 3.2), *canonical names* for content have an important functionality to serve as names that humans can refer to. A decentralised *name-registry* service replaces the DNS and performs resolution of canonical names to CIDs. For the actual content retrieval, an extra resolution is needed to obtain location(s) from CIDs, and this is typically performed by decentralised content storage networks (*i.e.* decentralised file systems).

The resulting search-based content retrieval process in both the current Web and the DWeb are depicted in Fig. 1. Although the search-based workflow is popular, other workflows exist to access content on the current Web such as following hyperlinks from one page to another, as well as shared direct links to objects on Cloud-based shared drives. In this work, we focus on the search-based workflow, as we argue that it encompasses the other workflows – *i.e.* the other workflows start from later points in the same sequence of events, and therefore analysing only the search-based workflow is sufficient.

2.3 Timeline

The content of this paper mainly spans the time period between 2009 and 2023. However, there is a large body of foundational works and developments. To illustrate the relations and chronological advancements, Table 2 presents a timeline of developments, their benefits, and key works studied.

The period 1980-2000 is characterised by the emergence of fundamental structures like the DNS and the Web itself, bringing about global connectivity. Going into the 2000s, P2P networks emerged, as well as Web 2.0. In the period 2005-2010, social media's exponential growth transformed connectivity, communication, and business interactions while innovations like blockchain and mobile edge computing emerged.

Between 2010 and 2015, the rise of smart contract blockchains, decentralised name registries, and novel storage and search technologies increased the focus on decentralisation, security, and transparency. Most recently, developments in non-fungible tokens (NFT) and blockchain scalability have contributed to a vision of a DWeb or Web3, with shared services and resource sharing.

2.4 Systematisation Framework

We use the process of traditional Web retrieval, as described in Section 2.1, to define a framework which can be applied to study DWeb initiatives. As shown in Fig. 2, we divide Web retrieval into three main components: **search engine**,

[1] CIDs are typically self-certifying names to secure the binding between name and content object it refers to.

Time	Event	Benefits	Notable Works
1980-2000	Introduction of the DNS	Human-readable name resolution, scalability, standardisation	[14, 127, 174]
	Introduction of the Web	Global connectivity, information accessibility	[29, 94]
	Unstructured P2P	Decentralised information storage and retrieval with low overhead	[30, 101, 156]
2000-2005	Structured P2P	Efficient decentralised content retrieval, scalability	[121, 151, 168]
	Introduction of Web 2.0	Dynamic content, further engagement and collaborations	[43, 133, 139]
	Popularity of Search Engines	Information accessibility, content discovery, monetisation	[31, 32, 40]
2005-2010	Growth of Social Media	Connectivity, communication, global sharing, business	[120, 144]
	Introduction of Blockchain	Decentralised trust, cryptocurrency, security, transparency	[64, 134, 201]
	Mobile Edge Computing	Reduced latency, improved performance, scalability	[117, 118, 159]
2010-2015	Smart Contract Blockchains	Decentralised and trustless execution, incentivisation	[33, 131, 191]
	Decentralised Name Registry	Decentralisation, security, immutability, censorship resistance	[88, 202, 210]
	Novel Decentralised Storage	Availability, security, persistence	[24, 48, 183, 199]
	Novel Decentralised Search	Censorship resistance, transparency, decentralised governance	[90, 103, 146, 150]
2015-present	NFTs and Blockchain Scalability	Immutable ownership records, business models, improved performance	[73, 188, 189]
	Novel Resource Sharing and Web3	Collaboration, incentivisation, trust, transparency	[26, 132]

Table 2. Timeline of key Web advancements.

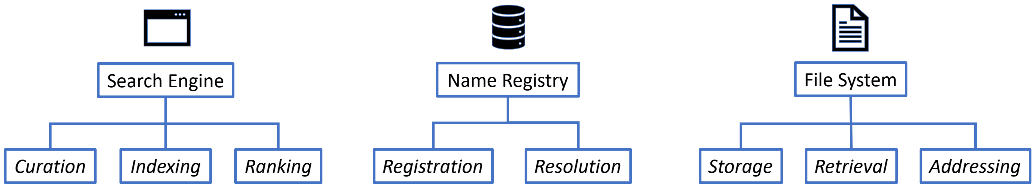

Fig. 2. Overview of key content retrieval components on the Web.

name-registry, and **file system**. For each of these areas, decentralised initiatives should be developed. This framework should allow them to position themselves amidst others in the space, and define how interoperability can be achieved.

In order to search for content on the DWeb, users will need to use a search engine, which can index DWeb content. The search engine also needs to decide which content to index through curation and in what order results are returned to users, which is decided by the ranking algorithm. Indexing and convenient retrieval of content are both dependent on human-readable names (*i.e.* canonical names), which are linked to CIDs using decentralised name-registries. Users need to be able to register name-to-value mappings to this service and resolve names to CIDs. Finally, content is stored on a decentralised file system, blockchain, or Web servers, and needs to be retrieved from these networks using its address or CID.

Our framework identifies these orthogonal components in order to clearly describe key pillars of a DWeb infrastructure. However, in practice a lot of components may be overlapping, and they may share underlying technologies. For example, each of our components uses blockchains to promote honest participation in resource sharing through incentives. Each component could even use the same blockchain network and underlying P2P network (*e.g.* Ethereum [201]). To keep clarity and structure in this work, we describe these overlapping components in the background (Section 3), and only refer to them in later analyses when relevant or distinct in implementation.

In Sections 4, 5, and 6 we will go through each of the key components in our framework and discuss the status quo of centralised systems; after that, we discuss and compare these to decentralised initiatives, and identify open issues.

Concept	Description	Variations	Challenges
P2P Networks	Distributed application architectures, allowing for resource sharing between peers	Structured, unstructured, hybrid	Churn, scalability, discovery, efficiency, security
Addressing Web Content	Method for addressing Web content. Content addressing uses hashing and verifiable bindings	Hash of content, hash of public key	Human-readability, security, decentralisation
Incentivising Participation	Incentives for sharing resources in decentralised protocols, often governed by smart contracts	Tokens, cryptocurrency	Fair exchange, sybil attacks, reputation

Table 3. Overview of background concepts.

3 BACKGROUND

In this section, we provide background on key concepts in content retrieval on the DWeb, consisting of peer-to-peer networks, addressing of Web content, and incentivisation of participation. Table 3 provides an overview of the concepts covered in this section.

3.1 Peer-to-Peer Networks

Peer-to-peer (P2P) networks are distributed application architectures that partition tasks or workloads between peers. Peers are equally privileged participants in the application, making P2P networks a sound basis for the DWeb. Peers make a portion of their resources, such as processing power, disk storage or network bandwidth, directly available to other network participants, without the need for central coordination by servers or stable hosts. Peers are both suppliers and consumers of resources, in contrast to the traditional client-server model.

P2P networks implement a virtual overlay network on top of the physical network topology, where the nodes in the overlay form a subset of the nodes in the physical network. Data is still exchanged directly over the underlying TCP/IP network, but at the application layer, peers can communicate with each other directly via the logical overlay links. Overlays are used for indexing, peer discovery and also to make the P2P systems independent from the physical network topology. The two main types of P2P networks are *(i)* unstructured and *(ii)* structured.

3.1.1 Unstructured P2P Networks. Unstructured P2P networks do not impose a particular structure on the overlay network by design but rather are formed by nodes that randomly form connections to each other [30, 101, 156]. Without a globally imposed structure, unstructured networks are easy to build and are highly robust to churn.

On the other hand, finding content is difficult in an unstructured network. In the earlier P2P networks such as Gnutella [156], the search queries were flooded through the overlay network to find as many peers as possible for the searched data. However, flooding is unscalable as its overhead on the network grows linearly with the number of search queries, which in turn grows with system size. The problem gets more severe for unpopular content which is present at only a few nodes. More recent P2P systems use slightly more scalable search mechanisms such as random walk, as we discuss in Section 4.2.1.

3.1.2 Structured P2P Networks. In structured P2P networks, the overlay is organized into a specific topology, and the protocol ensures that any node can efficiently search the network for content, even if the resource is extremely rare. The most common type of structured P2P networks implement a distributed hash table (DHT) [121, 152, 157, 168, 216] in which a variant of consistent hashing is used to assign responsibility for maintaining each content or resource to a particular peer. This enables peers to search for resources on the network using a hash table; that is, (key, value) pairs are stored in the DHT, and any participating node can efficiently retrieve the value associated with a given key within a bounded number of steps (usually $O(log(n))$, where n is the number of peers in the network).

Unfortunately, maintaining a structured overlay topology makes this type of network less robust in networks with a high rate of churn. Maintaining a structure also exposes the network to a vast range of attacks that can be more difficult to perform in an unstructured P2P network [177].

3.2 Addressing Web Content

As mentioned in Section 2.2, a distributed trust model requires a secure and verifiable content retrieval process. Therefore, the authenticity of the binding (mapping) between reference names to the retrieved content object must be verifiable by the users. This can be achieved by using content addressing, where more importance is given to the integrity of the file, rather than its origin.

Decentralised file systems typically use verifiable (*i.e.* self-certifying [122]) CIDs as reference names to achieve verifiability in the absence of trusted third parties. Self-certifying names for content objects are typically generated using one of the two mechanisms:

(1) *Hash of the content:* Generated by applying a well-known hash function to the contents of the object. The users can simply apply the same hash function on the retrieved content object to verify the binding between the name and the object.

(2) *Hash of a public key:* Generated by hashing a public key whose private counterpart is used to sign the content object. In this case, the content object includes a signature, which can be used to verify the name-to-content binding of an object. The signature is typically generated by the content publisher who owns the private key.

The properties of distributed trust (*i.e.* decentralisation), security (*e.g.* binding between names to object), and usability (*i.e.* a system with human-readable names) are non-trivial to achieve simultaneously, as also conjectured by *Zooko's trilemma* [198], which states that naming systems can only have two of the following three properties: *human-readability, security*, and *decentralisation.*

Among these three properties, some are contradictory. For instance, security is at odds with human-readability, because secure, self-certifying names are not human-readable due to the hash function applied. Similarly, the intrinsic binding between a human-readable name and its content (producer) is weak and verification of this binding through a centralised trusted party contradicts decentralisation. Another desirable property is persistence (*i.e.* names should not change when location or ownership changes). Ideally, a minor update to a Web file should not produce a completely different name. This, however, can be at odds with the security property, because self-certifying names lead to modifications in the names of mutable (*i.e.* dynamic) content upon updates to content (*i.e.* hash of the content) or ownership (hash of the public key).

As DWeb content uses hash-based addressing, they satisfy only the decentralisation and security properties, which are discussed further in Section 6. Decentralised name-registries have the potential to “square” Zooko's trilemma; that is, achieving usability while maintaining security and decentralisation. This is done by mapping human-readable, canonical names to CIDs in a decentralised manner using a blockchain, as discussed further in Section 5.

3.3 Incentivising Participation

One of the core foundations of DWeb is the distribution of trust. Rather than relying on a single root of trust, the responsibility of upkeep is delegated to a network of nodes, who share their resources for the protocol. As these nodes are required to spend resources for network upkeep, there need to be incentives, financially or otherwise, in order to keep them performing the work and keeping them honest. This can be seen as an overlapping component of all DWeb components, and therefore we now discuss resource sharing and particularly incentivisation as essential components and refer to these in our later sections.

Although early P2P networks survived on the basis of resource sharing based on altruism [181], they eventually failed to reach their full potential, partially due to the absence of incentives [113]. Recently, financial incentives powered by blockchains have been implemented and studied extensively.

Blockchains are secure, immutable shared ledgers that allow for value transfer in a network without a trusted third party. Blockchains are especially useful in the DWeb architectures due to their ability to incentivise users to participate and contribute to a network by paying them rewards, using tools such as smart contracts [33] and off-chain micropayment channels [73]. Blockchains allow trust to be exercised given that at least a certain percentage of the participants are honest (*e.g.* more than 50%); that is, they execute the blockchain consensus protocol correctly.

However, blockchains are only able to ensure a fair exchange of reward for work, if the resource contributors can produce verifiable proofs of resource consumption towards getting *useful* work (*e.g.* for up-keeping) done. For example, a node can prove that bandwidth [67], computation [64, 213], or storage [25] resources were actually provided, and a subset of the participants in the system can collectively verify these proofs as part of a consensus protocol [21], which can then trigger automatic rewarding of contributors for their valid proofs. Proving useful work done is not always plausible, for instance, for continuous services that take place for a period of time. When such proofs are unavailable, beneficiaries may issue periodic payments (*e.g.* using off-chain channels) to contributors (at the end of fixed or increasing time intervals) as long as the provided service is satisfactory. However, if the counter-party is malicious, it could lead to a loss of revenue for at least one interval, and the absence of penalties for malicious behaviour may encourage more nodes to behave undesirably.

A well-known way to counteract malicious actions when a fair exchange is unavailable is by using reputation systems. Centralised reputation systems have been explored thoroughly for online retail [87]. More recently, a number of works [22, 51, 92] specifically focus on decentralised reputation systems targeted to work with blockchains [23]. These works aim to incentivise honest collaboration between peers, as malicious behaviour results in a deduction of reputation. The deduction in reputation in turn leads to lower rewards in the future, either directly [97] or indirectly due to a loss of future revenue.

Another method which can be used to achieve fair exchange and thereby facilitate resource sharing in DWeb scenarios is using *Trusted Execution Environments* (TEE), which are secure computation enclaves to be used in various use-cases. Specifically in the case of computation outsourcing, using an enclave can maintain privacy and correctness, while greatly improving performance compared to smart contracts. A number of works use TEE's in combination with smart contracts to achieve distributed computation [6, 38, 47, 96, 214].

In a decentralised system, any participant can create and control an identity without the involvement of a trusted third party. This makes it possible for malicious nodes to simultaneously use multiple identities as part of a *Sybil attack*. By generating multiple *Sybil* identities (that pose as real users), malicious parties can trick a fair exchange mechanism into issuing undeserved rewards, for instance, by either bypassing a reputation system or by inflating the amount of actual resources consumed by the node. To prevent such attacks, proofs of resource consumption must be Sybil-resistant. In addition, researchers have proposed reputation systems that can identify Sybil nodes through mechanisms such as voting [128] and social network analysis [212]. These mechanisms identify outlier nodes as Sybils in the presence of an honest majority, *e.g.*, by taking the absence of a node's connections to other honest nodes in a social network as a sign of Sybil behaviour.

In the next sections, we go through the components of content retrieval on the DWeb.

4 SEARCH ENGINE

	Curating	**Indexing**	**Ranking**		**Incentive**	**Advertisement**	**Decentralised**		**Network**
			Function	**Location**			**Search**	**Content**	
Presearch [146]	Crawling	-	-	Gateway Server	Y	Y	Y	N	Ethereum
Yacy [206]	Voluntary Crawling	Distributed By Document	Combined	Local	N	N	Y	N	Hybrid P2P
Brave [164]	Crawling	Centralised	-	Centralised	-	Y	N	Y	-
Nebulas [137]	Crawling	Centralised	NebulasRank	Centralised	N	N	N	Y	-
The Graph [150]	Token Signaling	Subgraph At Indexer	-	-	Y	N	Y	Y	Ethereum

Table 4. Overview of decentralised search engine industry projects.

In this section, we first investigate how search engines currently work and identify a number of their characteristic components. After describing these currently centralised components, we introduce several decentralised search engines. We then analyse these based on how they incorporate the key components. Specifically, we discuss how they differ in terms of *curating*, *indexing*, *ranking*, and *incentives*.

4.1 Overview of Centralised Search Engines

Currently, when a user looks for content on the Web, they often start by submitting a query to a centralised search engine, consisting of one or more keywords. Proactively, the search engine has **curated** content to add to an index by crawling the Web. Keywords are then extracted from the content and added to an **inverted index**, which maps keywords to the Web pages where they can be found.

Upon receiving queries, the inverted index is used to compile a list of pages which might be relevant to the users. These results are then **ranked** using a ranking algorithm and returned to the users. The centralised search engines control what ranking mechanism (e.g. PageRank [31]) is used and are not always transparent about the specifics. Furthermore, ranking is generally personalised, which may lead to filter bubbles [142].

To incorporate a healthy business model, most centralised search engines monetise their services by adding advertisements through keyword auctions in search results, which allows the service to be free for users [40]. While the network infrastructure used might be distributed, the control, management, security, and policy are centralised, thus introducing a single point of failure which may also lead to cascade failures. As these network tasks are managed centrally they do not need to add **incentives** for participation. However, in a decentralised model, services likely need to leverage alternative business models and incentives for economic feasibility.

4.1.1 Key Challenges. There are still a number of key challenges surrounding *decentralised search engines*. Foremost among these challenges is the establishment of true decentralisation, where curation, indexing, ranking, and incentive mechanisms operate without reliance on trusted entities. Moreover, privacy and security concerns remain important topics of attention, in order to protect user data while maintaining search efficiency, robustness, and scalability. These challenges are further highlighted in Section 4.7.

Type	Addressing	Location	Name Registry
Blockchain Data	Block Hash	Blockchain	Blockchain Name-Registry
Decentralised Storage Data	Content Hash	Decentralised File System	Blockchain Name-Registry
Traditional Web Data	IP	Web Servers	DNS

Table 5. Classification of decentralised web content.

4.2 Implementations

We can generally classify decentralised search engines by their degree of decentralisation. The content which is being searched can also be classified similarly. We refer to centralised data as 'traditional' Web content which is hosted on Web servers. On the other hand, decentralised data encompasses content stored using decentralised file storage (see Section 6), as well as blockchains. Table 5 provides an overview of these content types. Using this, we can distinguish between three different decentralised search types: *centralised search on decentralised data, decentralised search on centralised data*, and *decentralised search on decentralised data.*

We use these classifications to analyse early-stage implementations, as well as several proposals in the research literature which generally have a narrow but detailed focus. Table 4 gives an overview of notable industry projects and summarises how they approach the various search components. Table 6, on the other hand, presents an overview of research proposals, focusing specifically on decentralised search mechanisms on decentralised storage networks. We have divided research from industry works because the former generally focus on one or a few aspects of search, rather than presenting complete systems, and therefore they have been analysed using different properties. As these projects are generally narrow in focus, we will now discuss their main properties, only referring to them occasionally in the rest of the analysis, as they do not present full and operational systems.

4.2.1 P2P Search Engines. The idea of decentralised search engines was first conceived by P2P search engines in order to improve the privacy, security, and performance of search on the Web and P2P storage networks. A number of initially distributed search engines relied on unstructured P2P networks [195], which offered high resilience to peer churn and good performance in retrieving popular items [148]. Some projects focused on improving the performance of unstructured search using techniques such as replication [37, 116, 173] and random walks [37, 116].

Another method of realising distributed search engines leveraged structured overlays, specifically DHTs [60, 125, 161, 208]. This allows for more reliable performance guarantees and better efficiency, especially when retrieving less popular items. A number of these focused on performance optimisations such as incorporating Bloom filters [114, 155] and caching [63, 155], as well as efficient routing using ant-like behaviour [179]. Some of these used popularity scores to determine the number of indexers per file [63] or ranking of results [114].

In order to optimise performance, a hybrid of structured and unstructured networks was used. For example, Yacy [206] structures all peers in a DHT, without implementing DHT routing. Another approach [112] locates rare items using a structured overlay, while popular items are located using flooding, leading to better performance and lower overhead.

These early search engines, however, often lacked additional security measures and incentives for useful work, which are needed due to the absence of a trusted third party [81]. This ultimately led to their loss in popularity. The rest of this section focuses on recent initiatives which are able to query novel decentralised file systems (see Section 6) or blockchains.

4.2.2 Centralised Search on Decentralised Data. There are a number of centralised search engines, which are able to query decentralised data. Recent works often focus on allowing users to fetch content using CIDs [164]. However, keyword search is also possible [83], where the central entity sniffs the structured [78] or unstructured network [19] to discover new content to add to the index.

Rather than creating search engines for decentralised file systems, some works have aimed to make centralised [137] and decentralised [175] search infrastructures for blockchain and smart contract data. While the projects above rely on centralisation, they are likely to play an important role towards the adoption of the DWeb.

4.2.3 Decentralised Search on Centralised Data. Another class of search engines are those that are decentralised but search the traditional Web. These offer much better privacy guarantees than centralised engines but are not suitable for the DWeb, as they currently do not support indexing content on blockchains or decentralised file systems.

As mentioned above, P2P search engines lacked incentives to add robustness and security to the system. Recent decentralised search engines often leverage a blockchain to add financial rewards, thereby making the network more secure and robust. For example, Presearch [146] rewards users for participating in upkeep functions such as crawling and indexing. Instead of centralised methods of issuing and distributing rewards, smart contracts may be used for decentralised incentive governance [191]. Smart contracts can be also used for reaching consensus on indexing and ranking, as is done by Raza et al. [153] to create a framework for privacy-preserving, decentralised search.

	Index Storage	**Ranking**	**Performance Optimisation**	**Security Features**	**Privacy Features**	**Governance**
SIVA [93]	IPFS DHT	-	Bloom Filter & Caching	-	-	-
Li et al. [103]	Kanban Cloud	-	Decoupled State and Computation	Verifiable Search, TEE, Decoupled Verification	Message Equalising, TEE	
Zichichi et al. [219]	Hypercube DHT	-	Routing using Hypercube	-	-	DAO
Zhu et al. [218]	B+ Tree / Hashmap	-	Index Storage Methods	Version Control	-	-
Wang and Wu [187]	IPFS DHT	Network Metrics	-	-	-	-

Table 6. Overview of research proposals for decentralised search mechanisms on decentralised storage networks.

4.2.4 Decentralised Search on Decentralised Data. We finally discuss decentralised search engines which operate on decentralised data, as these are the only suitable ones for a fully decentralised Web. However, at the time of writing and to the best of our knowledge there are no implemented projects which entirely achieve this. A number of projects [11, 78, 79, 91] focus on decentralised crawling and indexing of decentralised storage and blockchain data. Most notably, The Graph [150] is a decentralised indexing protocol for blockchain data, which itself is built on top of a blockchain.

Besides these industry projects, a number of research works have proposed a decentralised keyword-search mechanism for decentralised storage networks like IPFS [24] (see Section 6). As these projects are generally narrow in focus, we will now discuss their main properties, only referring to them occasionally in the rest of the analysis as they do not present full and operational systems.

Li et al. [103] proposed DeSearch, which is a search engine for decentralised services which decouples state from computation by using a centralised Cloud solution to store the index with high data availability, while maintenance of the index uses decentralised workers executing verifiable tasks (*e.g.* indexing, query processing). The verifiability property ensures that any third party (*e.g.* consumers of search results) can confirm that any task involved in the search process (carried out by an untrusted worker) is performed properly. This property is crucial in a decentralised setting where any worker can misbehave.

A number of works present systems which are fully decentralised (*i.e.* they also store the index over a P2P network). SIVA [93] builds a decentralised index for IPFS and stores it on the IPFS network using the native DHT. To increase performance, caching based on the Least Recently Used (LRU) [130] strategy and bloom filters are used. Wang and Wu [187] also propose to use the IPFS DHT to store the index and rank retrieved results from the index based on network metrics such as freshness, proximity, resource quantity, and bandwidth.

To increase performance, existing work has proposed storing the index in optimised structures rather than a general-purpose DHT. For example, Zhu et al. [218] propose decentralised keyword search on decentralised data networks using B+ Tree and hashmap data structures to store the index. Zichichi et al. [219] propose a hypercube DHT to store index items, structuring network topology using keywords. Furthermore, existing work proposes delegating governance of the index to a Decentralised Autonomous Organisation (DAO) [190], which allows peers to make governance decisions in a decentralised manner, *e.g.* propose and vote for changes, as well as implement tokens.

Another interesting idea is proposed by Fujita [62], who argues for implementing similarity search on IPFS based on locality-sensitive hashing (LSH), as an alternative to the prevalent keyword-search mechanisms. In their system, content hashes are stored on a DHT, although further implementation details and feasibility analysis are an interesting avenue for future work. Furthermore, it remains unclear if this scheme is sufficient for users who expect to submit queries consisting of keywords and retrieve a range of relevant information, rather than submitting content and retrieving similar content. Ditto [90] is another initiative which uses LSH to provide search functionality, and stores identifiers on a DHT, irrespective of the underlying content network or addressing scheme.

As we will discuss in Section 4.7, while these research systems seem promising, they are mostly early-stage works and therefore suffer from a number of limitations and require further work. A particularly interesting question is whether they truly achieve decentralisation. In the remainder of this section, we examine implemented projects and highlight how some of these projects uniquely implement the components of a search engine.

4.3 Curating

The curation process defines which content is added to the index. A number of projects take a similar approach to centralised search engines, which rely on crawling. Yacy is an example of a decentralised crawler, which allows users to crawl locally, either manually or proactively. Optimisations for decentralised crawlers have also been proposed such as leveraging the geographic proximity of resources [163]. Most other projects [137, 161, 164] remain reliant on centralised crawlers.

In order to crawl decentralised storage networks, however, different approaches are needed. To gain insights on peers and content in structured networks one may sniff the DHT traffic to discover new peers and CIDs, which can be fetched to gain insights [78, 83]. A similar approach may be used for unstructured networks, for example, in the case of the IPFS Bitswap [2] protocol traffic (Section 6.4), which is used to query peers for CIDs, may be monitored [19].

Another approach besides crawling is curation based on network consensus, as is used in The Graph [150]. Nodes in the network act as curators and use tokens to signal to indexers what content is valuable. While this might be a viable approach for on-chain data, it remains to be seen if this approach would work for other content types. This can be compared to research works which use popularity scores or managers [63] to signal which items should be indexed, although the latter lack monetary incentives and are therefore more prone to performance problems.

4.4 Indexing

The indexing process in decentralised search engines consists of two main steps. First, metadata is collected from content to create index entries that map extracted keywords to content identifiers. The second step determines how and where the index is stored, which is generally based on partitioning *by document* or *by keyword*.

Partitioning by document means that the content objects to be indexed are divided among peers who each maintain a reverse word index for a subset of the content objects, as is often the case in unstructured networks. This is inefficient when locating rare items, as nodes are required to flood the network in order to locate and retrieve the query results. Storing replicas of popular items can increase the performance in these networks [63], and in general many distributed search engines offer a degree of replication, which also adds resilience against Denial-of-Service (DoS) attacks.

Most structured and hybrid engines are based on partitioning by keyword, where each node maintains an index for the words that appear across different content, generally by mapping to the closest peer in a DHT [121, 166].

Another distinct approach is used in The Graph, where the indexers simultaneously perform the tasks of producing and storing an index in the form of subgraphs of blockchain data. Users can then directly contact these indexer nodes to access the indexed data, and in return issue off-chain conditional micropayments. Other recent engines manage the index centrally [137, 146, 164].

In DeSearch [103], decentralised workers perform indexing of content in a verifiable manner through a "witness" process which runs in the Trustable Execution Environment (TEE) within each worker. The witness process provides logs of inputs and outputs of tasks carried out by workers for third parties to verify the causality between the inputs and outputs. The witness logs are also stored in a verifiable data structure, albeit, in a centralised public cloud. Other research works [93, 187] have proposed to store the index directly on the storage network on which they operate, as well as optimised structured overlay networks [218, 219].

4.5 Ranking

When a user submits a search query, the relevant entries are fetched from the index, after which the results need to be ranked based on various metrics to be ordered and returned to the user. There are various ranking algorithms, which may be applied to decentralised search engines. The most well-known is PageRank [31], which scores the importance of Web pages based on the references pointing to and from the pages.

PageRank can be modified to determine the value of an entity on the blockchain, as done in NebulasRank [137]. In this work, transaction graphs are used to infer an entity's liquidity, propagation, and interoperability to determine its value. Nodes, smart contracts, as well as an entity's contribution to the network over a time period can be ranked, in a similar fashion to LeaderRank [105].

In centralised search engines, the ranking process generally runs globally. In a decentralised search, clients may locally select and implement their own ranking policies [169], or combine pre- and post-rankings, where results are initially ranked based on a number of standard metrics, after which they can be ranked again by the user based on local configurations [206]. While most research proposals overlook ranking of results, it has been proposed [187] to use network metrics such as freshness, proximity, resource quantity, and bandwidth.

Distributed ledgers can also be utilised to reach consensus on ranking, for example using random groups of TOR (The Onion Router) [53] block nodes and the Practical Byzantine Fault Tolerance (PBFT) algorithm [153].

4.6 Incentives

Centralised search engines can offer free services by monetising advertisements and user data. Most early distributed engines rely on an altruistic model where users are assumed to participate in the system honestly without the need for rewards. Recent systems have incorporated incentives using the blockchain. For instance, the revenue collected from advertisements could be used as rewards for up-keeping of the system [99]. We now discuss the monetary inflow and outflow of the system separately to illustrate this decentralised network economics.

4.6.1 Inflow. There are generally three sources of inflow of money in the decentralised search mechanisms. The first is users paying for a service. For example, this is the case for users querying the indexed data in both The Graph [150] and DeSearch [103]. This assumes that users are willing to pay for decentralised services instead of using free centralised options, which may not hold in practice.

The second source of inflow comes from advertisements. Generally, advertisers submit bids to show their advertisements with higher priority for particular keywords on search engines. Centralised engines generally use auctions to determine which advertisements are shown with higher priority [98, 149], although decentralised advertisement markets have been proposed as alternatives. An interesting example is keyword staking in Presearch, where the advertiser who stakes the most tokens on-chain for a particular keyword will be shown. In this case, the inflow is expected to come from per-click fees. However, currently, this approach retains centralisation as it relies on dedicated ad servers.

The advertisements shown to users are generally personalised, which is based on data collected from previous search behaviour. In this scenario, the user loses control over their privacy and is required to trust the central entity. To alleviate these concerns, Google had introduced but then later scrapped a proposal named FLOC[2], which was to use federated learning [59, 104] to group users in clusters, without data leaving the user's device. Although this is argued to be decentralised and privacy first, it might have led to an advertisement monopoly, as other third-party cookies would have been removed. Several other research works have investigated decentralised and privacy-preserving methods of personalised advertisements [18, 74], for example using blockchains [109, 147, 178].

Finally, in the search protocols built on top of blockchains, there is a third source of inflow. These are newly minted tokens, which are periodically released to reward for network upkeep [52]. There are also transaction fees that clients pay to use the underlying blockchain network, which are proportional to the added load placed on the miners. These fees are often collected directly by miners.

4.6.2 Outflow. The monetary inflow into the search protocols needs to be redistributed and flow out towards involved parties. In centralised search engines, the revenue generated by advertisements is collected by the centralised operator. In contrast, decentralised systems may delegate the ad revenue back to the users who watch the ads [164], or to nodes who assist in network upkeep [99, 146].

For example, in the Graph, *Indexers* earn tokens by serving client queries to their indexed subgraphs. *Delegators* can decide to stake tokens for a specific indexer, for which they will receive a percentage of their profits. *Curators* are incentivised to signal subgraphs honestly, as they can earn a percentage of the query fees.

Similar to other platforms, slashing of tokens [34] may occur when malicious behaviour is detected. This leads to a penalty deduction of a node's staked deposit on-chain.

On the other hand, DeSearch [103] rewards both workers for carrying out search-related tasks (*e.g.* indexing) and publishers of content using tokens. The reward tokens flow from the *consumers* of search results all the way to the

[2] https://www.wired.co.uk/article/google-cookies-floc

publishers of content (that appear in the search results) as in the following chain: consumers → rankers → indexer → crawlers → publishers. This chain follows the functional dependency between the tasks involved in the search process and rewards publishers of content based on their popularity, as similarly done in decentralised social media platforms [3].

4.7 Open Issues

4.7.1 Reliance on centralised infrastructures. As discussed, there are only a few projects which aim to provide a fully decentralised search on decentralised data, and many still rely on centralised back-end or gateway servers. For example, DeSearch [106] uses a hybrid infrastructure consisting of both centralised and decentralised components, but with built-in accountability (verifiability), achieving some of the desirable properties of decentralisation with good overall performance. On the other hand, while being more decentralised, storage of the index directly on the storage network like IPFS introduces new challenges. Because the index should be a mutable object that is frequently updated, storing it on an immutable storage solution is difficult. We can use the naming layer to alleviate the problem of mutable data, for example using name-registries. However, there still remain a number of issues such as the management of private keys. In Section 5 this is discussed further.

We conclude that building a truly decentralised search engine is non-trivial, and therefore a feasibility analysis is required. Specifically the question: "*are industry or research projects actually able to provide true decentralisation?* " needs to be answered. Particularly, the process of curating content to be indexed, maintaining and partitioning the distributed index, and ranking in a decentralised fashion need to be explored further. The difficulty here also applies to designing a system which encompasses all of these simultaneously. Alternative search workflows such as those based on similarity search [62, 90] seem promising in achieving higher degrees of decentralisation, but these and other workflows should be investigated further. On top of this, while privacy improvements are desirable, they should not come with significant performance degradation, and thus this trade-off should be analysed.

4.7.2 Complete systems. The area of decentralised search engines has relatively been investigated less compared to other DWeb infrastructures, and this is reflected in the fact that most systems are not complete in coverage of all search steps users expect. For example, the industry projects covered generally have a specific niche in terms of DWeb network, data type, or application. They also are not as sophisticated in implementation as some research works, which have a much more narrow focus.

While most research works have proposed some performance optimisations, few have looked past the structuring and storing of the index, and routing of queries. For example, how results are ranked after fetching them from the index has been barely explored in these works. Furthermore, how governance using incentives can be used to make the system more secure, robust, efficient, and usable has been largely overlooked.

4.7.3 Analysis of claims. It is argued in most works, both in industry and research, that a decentralised search will lead to better privacy and security, but this has not been shown in practice, as new attacks may arise in a new infrastructure. Therefore, we believe security analyses to be vital. Security is partially dependent on the crypto-economic incentives and mechanism design, which has not been considered in detail in most works, specifically in industry. Similarly, there is the issue of trust, as not all operations can be mediated through the blockchain. Here, reputation systems could play an important role.

	Scope	Ownership	Off-Chain Storage	Registry Fee	Resolution	Allow Subdomains	Network
Namecoin [88]	TLD	Permanent	N	Flat Fee	Local	N	Bitcoin
BNS [165]	Root zone	TLD Dependent	Y	TLD Dependent	Local	Y	Bitcoin
Handshake [76]	Root zone	Permanent	N	Auction	Local	Y	Handshake
ENS [162]	TLDs	Lease	N	Length Based	Local	Y	Ethereum
NXT [42]	TLD	Permanent	N	Flat Fee	Local / Server	N	NXT
Emercoin [58]	TLDs	Lease	N	Length Based	Local / Server	N	Emercoin
CNS [56]	TLD	Permanent	N	Premium / Regular	Local	Y	Ethereum

Table 7. Overview of decentralised name-registry projects.

5 NAME REGISTRY

In this section, we first give an overview of the name-registry currently used on the Web: the DNS. While the DNS is physically distributed, it is controlled and managed by a centralised entity. Then, we describe two important aspects of name-registry systems, namely *registration* and *resolution*. Finally, we present a number of decentralised name-registries and DNS alternatives and analyse how they differ in these aspects.

5.1 Overview of the DNS

The DNS is the default name-registry system used in the current Web, and one of its uses is to maintain name records, which map domain names (*e.g.* hostnames in URLs) to locations (*i.e.* IP addresses). The DNS servers use these records to respond to user queries.

The domain namespace is hierarchical: at the root of the hierarchy are the top-level domains (TLDs) such as *.edu* and *.com*. These TLDs extend to subdomains such as *acme.edu* which in turn can extend arbitrarily to sub-domains such as *mail.acme.edu*. The DNS namespace consists of portions called *zones*, each managed by a specific organisation or administration. The DNS records for each zone are permanently stored on an *authoritative* DNS server (under the control of the zone's administration) that has the authority to respond to DNS queries for its zone(s) [127].

An authoritative DNS server for a zone (*e.g. acme.edu*) can delegate its authority over the subdomains (*e.g. mail.acme.edu*) to other servers. The result is a hierarchy of distributed DNS servers across the globe, each responsible for a portion of the hierarchical domain namespace. The hierarchy of servers starts from the *root name servers* that hold "pointer" (*i.e.* NS) records, mapping each TLD zone to its corresponding authoritative DNS servers. Similarly, each authoritative server for a zone maintains a list of authoritative servers of its delegated subdomains.

The **resolution** of a hostname starts with a user contacting its local DNS server. If the local server has not previously cached the result, it returns either a root name server or an authoritative name server for one of the zones that are part of the queried domain name. If a server is not able to resolve the name, it returns the authoritative name server for the next subdomain using its NS record.

The root zones (*i.e.* TLD names) are centrally controlled by the Internet Corporation for Assigned Names and Numbers (ICANN), which delegates the administrative responsibility of each zone to a single manager such as an organisation or government, who in turn runs authoritative servers for the zone and can allocate (*e.g.* sell) subdomains (and delegate the control over that zone) to others. Domain names under TLDs are **registered** with a registrar or reseller, who is accredited by ICANN and certified by the registries.

Centralisation in DNS refers to ICANN's control and management of TLD zones and the root name servers. In addition to the top-level zones, governments have full power over the DNS servers residing within their territory. This may lead to censorship (*e.g.* blocking of wikileaks.org by several countries). Furthermore, there are other known security issues with the current infrastructure such as DoS attacks [141], DNS hijacking [158], DNS spoofing [167], and DNS cache poisoning attacks [95]. Existing security extensions, such as DNSSEC [14], have slow adoption [107] due to large overheads impacting performance and also due to intrinsic reluctance to change already deployed protocols.

5.1.1 Key Challenges. A number of critical challenges remain for *decentralised name registries*, for example in the areas of management of namespaces. Questions regarding ownership, pricing, and conflict resolution remain, for instances where multiple entities vie for the same domain. Moreover, their deployment and adoption require practical support and ease of integration. These challenges are further highlighted in Section 5.5.

5.2 Implementations

We now discuss a number of decentralised name-registry systems from industry and research. Within the context of the DWeb, these provide registration and resolution from human-readable names to CIDs. In doing so, they have the potential to overcome Zooko's trilemma, as the content names remain secure (due to hashing), human-readable (due to the name-registry), and also decentralised (as the registry happens on a decentralised network or blockchain).

5.2.1 P2P DNS Alternatives. Decentralisation of the DNS was initially proposed by research in P2P systems, with various goals in mind. For example, Overlook [174] aimed to improve the scalability and performance of the DNS by using dynamic replication and a DHT for servers.

Several works also aimed to improve the security of the DNS against various attacks by structuring DNS nodes in a P2P network, thereby distributing the top-level namespace. This was argued to protect against attacks such as DoS and both malicious root and TLD servers [4, 75]. However, the added security could present a trade-off, with a loss of performance [45]. These P2P initiatives suffered from the limitations of P2P networks, such as the lack of incentives.

5.2.2 Hybrid Name-Registry. Several hybrid approaches have aimed to provide name-registry improvements over the current DNS by leveraging a combination of centralised and decentralised infrastructures. For instance, DNSLink[3] allows IPFS [24] CIDs to be mapped using DNS txt records to DNS names. This does not overcome Zooko's trilemma (see Section 3.2), as it remains reliant on the centralised DNS.

Some works use consortium blockchains to create a decentralised DNS. We consider these to be hybrids as well because these networks are not entirely open and decentralised. These consortium blockchains generally publish domain name operations on-chain, but store actual domain name data off-chain. Besides singular blockchain implementations [193], a hierarchical structure of multiple chains may also be used [57, 111]. The rest of this section focuses on solutions implementing open blockchain and smart contract based name-registry systems.

5.2.3 Blockchain-based Name-Registry. A number of industry and research projects have proposed using blockchains for name-registry, mapping human-readable names to CIDs in a decentralised manner and claim that they overcome Zooko's trilemma. We first describe projects which use first-order registration—*i.e.* those that modify the blockchain state directly using transactions, rather than through intermediary smart contracts. While smart contract systems also modify the blockchain state, they operate at a different level of abstraction and allow for more flexibility and complex

[3] https://dnslink.io

logic. We make the distinction between the two in order to bring structure and grouping to a large number of works, but also because smart contract systems can be seen as second-generation blockchain systems.

A generic name-value registration system is implemented by Namecoin [136], offering a naming system with decentralised governance. Similarly, NXT [42] and Emercoin [58] implement generic name-value storage services on their native blockchains. Another blockchain-based naming protocol is Handshake [76], which aims to replace the root zone file and root servers. Rather than targeting to replace the entire DNS infrastructure, the control of the TLDs is decentralised, allowing an infinite amount of names to be created. Therefore, compared to other solutions which allow naming operations within the scope of one or a few TLDs (e.g. *.bit* for Namecoin), Handshake is more flexible and customisable. On top of these naming protocols, other systems can be built to create secondary marketplaces for reselling names and easy participation in name auctions [135], as well as to add security and accessibility[4].

Besides these industry initiatives, several research works [28, 71, 196, 210] have focused on the security vulnerabilities of the DNS and propose using blockchain solutions to enhance the security of the current infrastructure. Security issues are partially due to the absence of a method to certify the integrity of information of queried name records. A number of works have improved this by storing verifiable record hashes on the blockchain [110, 211]. Blockchain-based registry systems may also be extended to Public-Key Infrastructure (PKI) encryption schemes, which generally suffer from similar issues due to reliance on centralised certificate authorities [8, 89].

5.2.4 Smart Contract Name-Registry. Decentralised name-registry systems can also be implemented using smart contracts on top of existing blockchains. The advantage of using smart contracts is that many services can be offered on the same blockchain. Blockchains that solely implement naming operations can be less secure as the network is often smaller, and may have limited functionality. On the other hand, as there is less overall traffic, better performance can be expected. The advantages of both are expected to converge with sharding [188] and layer-2 solutions [73].

A number of recent projects use the Ethereum blockchain as the underlying infrastructure [56, 160], and generally use a set of smart contracts for registration and resolution. Most developed among these is the Ethereum Name Service (ENS) [162], which is a general name-registry for Web 3.0 content including cryptocurrency addresses. However, around 98% of currently registered names on ENS seem to identify Ethereum addresses [202]. Stacks [9, 165] also created the Blockchain Naming System (BNS) on top of their native blockchain using a smart contract, after initially using the Namecoin blockchain [10].

The industry projects discussed above still have many security vulnerabilities [143], particularly in the areas of malware, name-registration mechanisms and markets, phishing, and immutability. Specifically, looking at name registration, *domain squatting* [215] attacks present a big threat. In this attack, malicious users register as many names as possible at low costs, with the sole purpose of selling them in the future for profit rather than using them, or using them for fraudulent activity based on misdirection or impersonating another source. To illustrate some of these issues, studies have identified that in Namecoin, squatting is a significant problem [88], a single entity controlled over 51 % of the network [10], and that there are possible domain extortion and phishing schemes [143].

Another aspect often overlooked in the design of decentralised name-registries are **incentives**. Similar to the other components in a DWeb infrastructure, nodes will need to collaboratively perform work to keep the system working, for which they expect rewards. In the case of blockchain and smart-contract based solutions, some of the incentivisation for networking functions is taken care of by the underlying blockchain and consensus protocol. However, to mitigate some of the attacks mentioned, malicious behaviour should be protected against by aligning incentives with honest

[4] https://github.com/okTurtles/dnschain

behaviour, specifically tailored for the name-registry use-case. This has been partially achieved by the registration mechanism, as we describe in Section 5.3.

In the remainder of this section, we highlight unique aspects of blockchain and smart contract name-registries, specifically in the areas of registration and resolution. Table 7 gives an overview of key aspects for select projects described in the previous sections.

5.3 Registration

Ownership, pricing, and control of names are handled differently among projects. Ownership of a namespace can be permanent [42, 56, 76, 136], in which case the owner has control over the subdomains indefinitely, although there may be periodic renewals required to ensure liveness at no cost. Conversely, ownership may also be temporary and require periodic renewal fees to extend the lease period [58, 162], which may deter squatting attacks. Ownership permanence may also be set differently among namespaces within the same system [165].

Pricing of domains and namespaces also varies across systems (and within the same system [165]). Initially, low flat fees were the norm for acquiring domains [42, 136]. However, it was shown that in the case of Namecoin, this pricing model made the system susceptible to squatting. To counter this, a number of projects started charging differently based on the perceived value of a name [56], for example, based on their length [58, 162]. Another method leverages Vickrey sealed-bid auctions [182] on-chain to allocate names [76].

All systems allow for reselling of domain names on a secondary market, as this is seen to be a security feature against squatting. Some extend this further by allowing the sale of subdomains of a name [56, 76, 162, 165].

5.4 Resolution

The hybrid projects mentioned either rely on servers[5], the current infrastructure, or a permissioned chain to resolve names. On the other hand, for blockchain and smart contract based solutions, the main difference in resolution with the DNS is that they directly use the blockchain to resolve names. This can be done locally by running a full node on the network, using a simplified payment verification (SPV) node [76], relying on browser extensions, or using servers [42, 58, 136].

When querying the blockchain, the entire naming records could be traversed to find a relevant entry. A faster method uses separate resolver (which maintains an “authoritative” record set by the owner) and registry (where the search starts) smart contracts [56, 162].

5.5 Open Issues

5.5.1 Security. Decentralised name-registries and DNS alternatives are recent developments, especially those built on top of blockchains. While the initial implementations and results seem promising, more research is needed into how they hold up in practice, especially in terms of security.

Recent works [88, 143, 202] have exposed some serious security threats and design flaws in early systems. They focus on specific vulnerabilities such as domain squatting and phishing, but a wider attack vector needs to be analysed and evaluated before we can claim that they offer better or similar security guarantees as the DNS, and that they are actually able to “square” Zooko’s trilemma. Furthermore, they rely on trust and performance assumptions of the underlying

[5] https://www.opennic.org

blockchain network, which has been shown to be too slow [111] in certain instances. Some projects rely on centralised servers for name resolution to increase performance, but this adds a layer of centralisation [42, 58, 136].

5.5.2 Namespace management. Another aspect which has often been overlooked is how these systems handle instances where public keys to alter names are lost, compromised, or even just upgraded. It may also be desirable to use a threshold of public keys, instead of just one, to verify the identity of owners or publishers for security reasons. Although P2P literature has attempted to tackle these issues, for example using social and personal naming systems [61], the blockchain-based systems have, as far as we know, not identified or addressed these issues.

The prevalence of various financially-motivated attacks (such as domain squatting) is a sign that there is room for improvement in the decentralised management and governance of namespaces. For example, popular names, especially those with commercial values (*e.g.* registered trademarks), require careful management, as they are obvious targets for such attacks [202]. While decentralised name registries that are governed by smart contracts have developed mechanisms (*e.g.* auctions) to manage namespace ownership, more research is needed for building algorithmic mechanisms for robust namespace governance (possibly together with crypto-economic incentive mechanisms) to deter financially-motivated attacks on the namespace.

5.5.3 Deployment and support. In terms of ease and practicality of deployment for decentralised name registries, recently several browsers have introduced extensions (plug-ins) for ENS support. However, despite the browser support, a recent study [202] has reported only a few thousand URLs being stored in ENS, while the vast majority (*i.e.* 98%) of the names identify blockchain addresses (*e.g.* addresses of popular cryptocurrency addresses such as crypto exchanges). On the positive side, unlike NameCoin which was deemed dysfunctional by a recent measurement study [88], the number of names registered on the ENS system (including the number of URLs) has been reported to be steadily rising [202].

6 DECENTRALISED FILE SYSTEM

Architecture	Hash	Decentralised	Self-Certifying	Human Readable	Hierarchical
IPFS [24]	Multihash	Y	Y	N	N
Swarm [170]	bzzhash	Y	Y	N	N
BitTorrent [145]	-	N	N	Y	N
Skynet [183]	Skylink Hash*	Y	Y	N	N
Storj [199]	-	N	N	Y	Y

Table 8. Comparison of addressing of decentralised Web content. * *unclear which hashing algorithm is used.*

In this section, we first describe how content is currently stored on the Web, and we discuss how *storage*, *retrieval*, *addressing*, and *incentivisation* are handled. We then describe a number of decentralised file system implementations, and analyse how they handle these aspects.

6.1 Overview of Web Storage

In terms of content **storage**, the current Web ecosystem is dominated by silos of providers residing in centrally-controlled, public Cloud infrastructures. While these public Clouds provide users with on-demand access to a large pool of shared resources, they operate with little or no transparency. As a result, concerns over the security of confidential or sensitive data can favour the deployment of private Cloud infrastructures which require large upfront costs.

More importantly, the centralisation in the infrastructures of these silos means that they reside in only a few locations on the Internet. As a consequence, even simple network failures can lead to the unavailability of these silos, as experienced by users during recent outages at Amazon Web Services (which resulted in the loss of access to a significant portion of the Web) and Facebook [46, 172]. While replication of content across silo boundaries would lead to better performance and availability for users, the lack of incentives prevents such cooperative action among the silos.

Content **retrieval** from centralised Cloud infrastructures deployed at remote datacenters can experience large communication latency. To reduce this latency, the emerging *edge computing* [159] paradigm promises to deploy small-scale datacenters at locations close to users. However, such small-scale edge infrastructures are mostly appropriate for small-scale, low-latency (or high-bandwidth) applications and can not cope with the workload of the Web. Instead, a truly decentralised Web can be realised by pooling the vast amount of global user resources and **incentivising** their proper usage to achieve scalability and sufficient performance.

Other important actors in content retrieval in the current Web are Content Distribution Networks (CDNs), which provide large-scale retrieval of quality-of-service (QoS) sensitive content through on-demand replication of content at distributed caches around the world. While on-demand replication of content with simple reactive caching policies (such as LRU) are effective in providing sufficient content retrieval performance, the location-based nature of Web references (*i.e.* **addressing**) makes replication and moving of content difficult, as such actions invalidate existing references to the content. To deal with this problem, CDNs use proprietary name resolution mechanisms that immediately update the invalid Web references to content upon movement or replication. Despite being a distributed infrastructure, CDNs are centrally-governed systems and charge content producers for distributing their content. This makes content delivery expensive, especially for small content producers. Finally, to serve content using HTTPS, CDNs need to hold the content publisher's private keys further increasing centralisation and lowering the security of the entire Web [82].

6.1.1 Key Challenges. *Decentralised file systems* encounter several crucial challenges. First of all, the balance between desired properties like security, decentralization, and human readability needs to be maintained, as described by Zooko's trilemma. Another critical concern revolves around mutable content, where an update to the content should not alter its identifier. Finally, there remain challenges around privacy and legal aspects. These challenges are further highlighted in Section 6.7.

6.2 Implementations

The ideas behind storage networks were first developed for P2P networks and produced unstructured networks like Gnutella [156]. While these were able to perform well in fetching popular items, they were not as successful in quickly retrieving less popular content.

A number of projects started leveraging structured networks, particularly the DHTs, to achieve more reliable performance guarantees. Most prominently among these was BitTorrent [145]. Over time it became clear that many of these networks lacked robustness in terms of availability, security, and stability, partially due to the lack of incentives. Furthermore, BitTorrent's main use became the distribution of unlicensed products [119], leading to copyright and legal issues.

Recently, novel storage networks have emerged and gained popularity [48], most notably IPFS [24], Sia [183], and Swarm [170]. These can be built on structured, unstructured, or hybrid networks and use content addressing. While the principles of these projects are closely related to Information-Centric Networking (ICN) [5, 205]—the implementation of a content-centric paradigm directly in the network layer that replaces IP—these novel projects work in the application layer.

Content addressing (see Section 3.2) is a natural fit for decentralised file systems targeting a public DWeb, as content is distributed over the network with a level of replication, and therefore any node (or a set of nodes) may be able to serve a requested file. It would be counter-intuitive to restrict file retrieval to only a single location as is done in the current Web. For storage of private data, however, similar to personal Cloud storage, content addressing is not always necessary. Such is the case with Storj [199], which also introduces optimisations targeted towards decentralised Cloud storage and uses satellite nodes which manage parts of the network.

DStore [204] takes another approach to create a distributed outsourced data storage and retrieval scheme. It uses smart contracts to audit the integrity of the outsourced data, achieving security and efficiency. Liang et al. [108] designed a storage and repair scheme for fault-tolerant data coding, realising a regeneration code with high precision and repairability, focusing on blockchain-based networks.

Another distinct project that proposes decentralising storage is led by Tim Berners-Lee and is called Social Linked Data (SoLiD) [35]. SoLiD is designed to decouple user's personal data from the applications that use them and allows users to set access control policies to maintain the privacy of their data stored in decentralised storage units. However, the users must trust the decentralised storage units with properly authenticating applications and following their access control policies. More importantly, the current SoLiD protocols rely on centralised infrastructures such as the PKIs and DNS.

Finally, we mention blockchains as an alternative method of storing data in a decentralised manner. While storing on the blockchain is secure, it is extremely expensive, as the data is replicated over all peers and thus distributed with extreme redundancy. In the rest of this section, we focus on recent decentralised file systems on the application layer with live implementations and analyse their key aspects.

6.3 Storage

The decentralised file systems are implemented over P2P storage networks, where a DHT structure may be used to find peers who are the providers of specific content. Each content is initially stored only by the publisher who then serves the file, given that the publisher can (and is willing to) actively function as a provider of that content. Additionally, any peer downloading content can then cache that content and become a provider [24]. Furthermore, some protocols allow for nodes to formally publish deals governed by a blockchain, where one node pledges to store a particular content item [26, 183]. Secondary off-chain markets have also emerged where providers offer to *pin* specific files (*i.e.* permanently make the file available). Some systems also introduce coding techniques (*e.g.* erasure coding) to improve the retrievability of content (*e.g.* only a certain percentage of coded segments of content is sufficient to restore the content). Combined with incentivised pinning of files at multiple locations, coding can further improve the permanence of content stored in these systems.

The content stored on the decentralised file systems is generally public data, and anyone in the network with the CID can fetch the file. This approach causes privacy concerns for the users that are overlooked in some systems. For example, the content searched by a peer can be easily monitored, especially by others that are directly connected to the peer in the P2P network[19]. Some systems such as OneSwarm [84] distinguish between trusted (*e.g.* friends and family) and untrusted peers and introduce address obscuring techniques to increase the privacy protection of their participants.

Protecting the privacy of storage nodes is important to avoid censorship of content [123]—the operators of servers within certain regions in the world may be legally required to report any prohibited content that is stored on their machines or that is requested by users in queries. Therefore, the server operators would rather be able to deny knowledge of content that they store or the content of the queries that they receive; that is, the ability of plausible deniability.

Servers can deny knowledge of the content they store if the clients store content in encrypted form and separately from the keys used to encrypt the content [180]. In both MaidSafe [100] and Storj [199], the data are stored in the network in an encrypted form. In both systems, content is divided into a sequence of chunks and the individual chunks are stored on the DHT. In MaidSafe [100], each chunk of content is encrypted with the hash of the previous chunk in the sequence, and each encrypted chunk is then XORed with the concatenated hashes of the original chunks for further obfuscation. Together with the encrypted chunks, a publisher must also publish a manifest file (*i.e.* containing meta-data) that maps the hash of obfuscated chunks to the hash of the real chunks.

In some DHT implementations, the network allows nodes to search for peers who store (*i.e.* the providers of content) a given CID [24]. In such DHT implementations, a client searching for a content object by its CID can retrieve "provider records", *i.e.* the IP addresses of peers that store that content, by querying the DHT. A possible privacy extension is to store encrypted provider records and allow content publishers control over access to the key, as similarly done in SoLiD [35]. Another recent idea is to encrypt provider records of a CID using a key derived from the CID itself [124]. This approach allows providers to be revealed to only those who know the CID of that content—in this case, the CID can be constructed as a combination of the hash of the content and secret information known only to the publisher of the content. However, storage privacy is a challenging problem since the systems often prioritise ease of discovery over privacy.

6.4 Retrieval

Retrieving content from decentralised file systems can happen through the network they are implemented on, being either unstructured, structured, or hybrid. In the unstructured case of Sia [183], nodes gather hints of the possible location through for example the blockchain deals, after which a select number of nodes are queried, rather than using a flooding-based approach. The other projects use modified versions of the Kademlia [121] DHT either just for locating peers [26, 145], or both peer and content discovery [24, 170, 199].

The hybrid approach in IPFS aims to optimise the performance of content retrieval through both unstructured connections with a set of peers and the structured DHT (*i.e.* Kademlia) network. As part of the unstructured network, each node maintains connections with a small set of peers that are discovered either through DHT communications or incoming content requests. These connections are used as part of the Bitswap [2] protocol to request for content. In the Bitswap protocol, nodes exchange lists of wanted content (using their CIDs) with their directly connected peers. Upon receiving a Bitswap *want* request, one or more peers may respond with an acknowledgement of having the content stored (*i.e.* cached) locally. Upon receiving one or more acknowledgements, the node then attempts to retrieve the content from all of the acknowledging peers in parallel (*e.g.* request individual chunks of the content from different peers), similar to downloading content using BitTorrent [145].

A node that wants to download a content object first asks its Bitswap peers for that content's CID. If none of the direct peers have the requested content locally cached, then the node queries the DHT for a list of peers who can provide the content. In general, clients may be able to retrieve content using their direct connections (especially, the popular content that is previously retrieved and cached by many peers) without using the global DHT. Because retrieval of content through a global DHT can be slow (*i.e.* requires contacting O(log n) peers), Bitswap can reduce the content retrieval latency. The Bitswap protocol also helps with reducing the burden on the DHT network, as the content requests tend to follow a power-law distribution, *i.e.* the majority of requests demand the most popular content, in most content networks [68]. However, attempting to retrieve unpopular content from BitSwap peers may end up delaying the retrieval, as it delays switching over to the DHT to query for content. Therefore, a hybrid system may require optimisations to improve the content retrieval latency by perhaps using both networks at the same time at the cost of additional overhead in the system.

In addition to the performance of content retrieval, privacy is another important consideration. Ideally, a system should not reveal which particular content is searched by a given client, providing a form of "reader" privacy. A recent work has proposed lightweight extensions to the IPFS DHT to enable clients to search for content without revealing the exact content that they are looking for [124]. This extension proposes decoupling the discovery of providers of a CID from querying the content providers for content: a client searches for providers of a CID by querying the DHT, but using a prefix of the CID (as the search key), which still allows the DHT to find a region with potential providers, while hiding the original CID from the DHT nodes that route the query. Once potential providers of the CID are found, the client queries these peers using the hash of the CID, instead of the CID itself. This way, the providers are unable to find out which CID a reader is looking for, although they could find the requested content in their cache, given that they store it.

6.5 Addressing

As discussed in Section 3.2, addressing content on the DWeb is not straightforward, because many of the desirable properties cannot be achieved simultaneously, as described by Zooko. Most recent projects targeting public data, such as those for the Web use content-addressed, self-certifying hashes to refer to content [170, 183]. This can be extended to support multiple hash functions by using prefixes, as is done by multihash [24]. Human readability can be achieved using trackers [145], at a loss of security or decentralisation.

A desirable property of naming is that even mutable content objects have persistent names that users can always use to refer to them. This means that the CIDs of content objects should not change when their attributes (*e.g.* location, file contents, or ownership) change. Hash-based names do not provide persistence, because the contents of a file determine its name. This could however be achieved with public-key-based names (refer to the types of self-certifying names in Section 3.2) such as IPNS [6], which allow identifiers to be linked to public keys. This way, a user can update a file by signing the updated file with their private key, while keeping the name of the file the same.

6.6 Incentives

Currently, centralised storage options are cheap or even free, because the centralised parties are able to monetise their services. Decentralised services mitigate against security vulnerabilities and add transparency, but still require workers to be able to cover their cost of work, which is why incentives play an important role.

Early P2P storage networks generally leveraged non-financial incentives, such as BitTorrent's tit-for-tat [41], which rewards for resources put towards the network by faster downloads in return. Another example is Samsara [44] which focuses on tit-for-tat behaviour for contributing storage resources, *i.e.* symmetric storage relationships between peers. In Samsara, a peer *S* stores a chunk of data for a peer *R* in exchange for *R* storing an equally-sized *storage claim* by *S*. *S* can periodically verify the existence of the claim through a challenge-response protocol which prevents *R* from removing or compressing the claim, and eventually *S* can request *R* to store a data chunk from *R*, in which case *R* stores *S*'s data replacing the claim. However, malicious peers can refuse to store data later when requested as the

[6] https://docs.ipfs.io/concepts/ipns/

claim mechanism can not enforce peers storing claims to replace them with data. Also, the verification of claims adds significant overheads on the peers.

A number of projects have also started incorporating blockchain-based rewards in their networks. Filecoin [26] creates an incentive layer on IPFS where nodes create on-chain storage deals. Storage nodes regularly submit proof that they have been storing unique copies of the data, for which they receive off-chain micropayments. Similarly, BitTorrent issued a token to add robustness in their platform, while Skynet, a decentralised CDN, leverages the Sia blockchain. Swarm and Storj issued blockchain tokens as well. Arweave [200] takes another approach towards realising decentralised storage and uses a blockchain-like linked structure with mining rewards based on pseudo-random previous blocks linked to the latest state. Therefore, users pay a one-time mining fee for storage, assuming that miners are honest in keeping and providing their data, which may not hold in practice and lead to poor scalability and performance.

6.7 Open Issues

6.7.1 Achieving desired properties. One of the main issues in decentralised file systems remains the contradiction of desired properties of names (*i.e.* Zooko's trilemma). For example, secure, self-certified naming is at odds with human-readability. While solutions, such as name-registries, have been proposed to handle these contradictions, they need further analysis and evaluation. Also, storage of mutable content is another challenge in decentralised file systems. Even when the hash of a public key (see Section 3.2) is used for persistent naming of dynamic content, the file system must guarantee that a retrieval operation on a name would return an up-to-date content and not an outdated one that is cached by the nodes in the network. One possible workaround is to add version numbers to names (*e.g.* as a suffix in systems that support hierarchical naming), but this also comes with problems such as retrievers not necessarily knowing the current version of the content. We believe that the current decentralised file systems have still room for improvement in supporting mutable content. Another area for improvement is the centralisation of file systems. Recent measurement studies [20] expose significant reliance on centralised infrastructures in the IPFS network. This needs to be explored on other networks and addressed in order to achieve a truly decentralised Web.

6.7.2 Privacy and performance. Privacy of both the content retrievers (*i.e.* readers) and content publishers/providers (*i.e.* writers) in decentralised file systems is an active area of research, as we discussed in Sections 6.3 and 6.4.

Recursive routing (*i.e.* querying) approaches can slightly improve the privacy of readers (and the writers) by routing the readers' content requests and the corresponding data sent by the writers through intermediate nodes [100]. As opposed to IPFS and many other systems that use the iterative routing approach of vanilla Kademlia [121], MaidSafe [100] relies on a recursive routing approach, which can also reduce the latency of content retrieval, as fewer round-trip times are required to locate storage nodes for a given content with recursive querying compared to the iterative approach.

In terms of reader privacy, there are a few other promising approaches (see Section 6.4) that aim to hide the content that clients are searching for from other participants in the decentralised file system. Although these approaches are somewhat effective, the rather lightweight approaches are still prone to leakage of the content being searched to other nodes. For example, in the approach by Michel et al. [124], the nodes use a prefix of the CID as a search key to find the peers that provide the content in the DHT. Even though the use of the prefix hides the original CID, servers can still infer the CID by matching the key with popular CIDs that are close to the search key. Backes et al. [17] use threshold cryptography along with quorums to enable routing queries with privacy. Their approach, however, requires significant overhead to content retrieval.

In terms of performance, decentralised file systems can suffer from slow response times as reported recently by measurement studies [176], and it is an active area of research to improve the performance of content retrieval in these systems. The use of hybrid P2P networks is an effective approach, especially for retrieving popular content with low latency, as discussed in Section 6.4. However, striking a balance between performance (*e.g.* ease of content retrieval) and privacy is a challenging problem that requires more attention. Overall, we find that more research is needed to improve writer (*i.e.* content provider) privacy, especially in those systems that rely on the list of the providers of each content to be public [24].

6.7.3 Legality and moderation. We finally highlight the unclear legal implications across all components described in this work, which will require cross-disciplinary work. For example, the legal implications of adding illegal content to the index and being returned by a search engine, registering a domain name for a company by someone else, or storing illegal files on the decentralised file system are not clear.

7 RELATED WORK

To the best of our knowledge, our work is the first to provide a holistic view of the technologies that are useful for decentralised content retrieval. Although the main focus of this survey is on the recent works, *i.e.* Blockchain-era technologies related to Web3.0, we also introduce some of the notable P2P-era research that introduce the key concepts used by the next-generation decentralised content retrieval systems.

In one of the earliest works on Web information retrieval, Kobayashi et al. [94] survey the content retrieval technologies in the early Web, *i.e.* Web1.0, when it was only few years old. In this work, the authors discuss the search engines and the users' experience with the early search technologies of that time.

Other surveys have focused on only a subset of the technologies involved in decentralised content retrieval in blockchain-era systems. For example, a recent survey by Daniel et al. [48] discus decentralised storage systems in Web3.0, but without attention to the technologies that enable search and retrieval of content in those storage systems. Li et al. [106] take a different focus and survey how future data-driven networks can be realised using blockchains as the underlying technology to enable decentralisation, security, privacy, and resource sharing. However, their focus is mainly on the blockchain-based solutions and do not take into account the rest of the Web3.0 stack. Similarly, Benisi et al. [27] describe how blockchains are used to create decentralised storage networks, where nodes can rent out their untrusted storage hardware using smart contracts.

In terms of blockchain, Zheng et al. [217] present a comprehensive overview of blockchain technologies, which includes technical components such as consensus, as well as potential applications. While certain aspects such as security and privacy enhancements, and reputation systems have been mentioned, a global Web3.0 use-case has not been mentioned. Neudecker and Hartenstein [138] describe the network layer aspects in terms of attacks, and design implementations and considerations for permissionless blockchain networks. While this work focuses on blockchains, often other Web3.0 components like file systems share network layer design and concepts. For example, Filecoin and Ethereum 2.0 have both used GossipSub based messaging protocols in their network layers [184].

Earlier work also surveys the P2P-era content distribution research before Web3.0, which we also briefly touch upon in this paper. For a general overview of P2P networks, Keong et al. [115] study and compare network overlay architectures. More related to our work, Androutsellis-Theotokis and Spinellis [12] present an early survey and framework for analysing P2P content distribution technologies. Similarly, Hasan et al. [77] focus on storage techniques within distributed file systems.

Xylomenos et al. [205] present a comprehensive survey of information-centric networking (ICN), which aims to implement a content-centric network layer replacing IP. Although the content-centric paradigm (*i.e.* fetch content by name, not location) is also central to many decentralised file systems like IPFS, these are application-level systems designed to run as overlays on top of an IP network layer.

A number of works have also surveyed popular techniques which distribute Cloud solutions, but do not necessarily decentralise their ownership and governance. Zolfaghari et al. [220] discuss the state-of-the-art solutions and future directions for CDNs. They also describe how CDNs converge with emerging paradigms like Cloud and edge computing. Ghaznavi et al. [65] focus on CDN security challenges and possible solutions to these. Mach et el. [117] describe the emerging concept of mobile edge computing, and present use-cases, integration and standardisation efforts, and technical solutions. Mao et al. [118] also survey mobile edge computing, but focus on the communication perspective. As mentioned before, while these solutions tackle some issues associated with centralised Cloud and Web, they remain centralised in their control and governance.

A number of surveys focus on hybrid solutions which combine distributed storage and computation techniques with decentralised solutions and governance such as P2P networks and blockchains. Related to content retrieval, Anjum et al. [13] survey techniques that complement the centralised content delivery with P2P content retrievals in CDNs. However, such techniques use a centralised architecture with trusted CDN servers resolving requests to appropriate peers. Jia et al. [86] also present a survey on collaboration for content delivery, focusing on collaboration techniques in network infrastructures including P2P-CDN, collaborative caching, SDN, ICN and more. Finally, Yang et al. [209] survey attempt to integrate blockchains with edge computing solutions in the areas of network, computation, and storage. If these techniques can be integrated with security and privacy first, they could be used as a building block for Web3.0, for example using computation and storage platforms to crawl and create indexes, maintain blockchains, and enhance storage networks.

8 SUMMARY AND CONCLUSION

In this survey, we present a thorough overview and analysis of the content retrieval process on the decentralised Web, also known as Web3.0. After describing how content retrieval is handled on the current Web, we identify essential components of the retrieval process, consisting of search engines, name-registries, and file systems. In each of these areas, we have provided an overview of the state-of-the-art projects and proposals, and provide a comparative analysis with the current centralised model.

Our analysis highlights a number of open issues, which need to be addressed for a decentralised Web to be realised. In the area of search engines, we found that most existing projects are not able to truly meet the demands of DWeb, and often violate the decentralisation property at one or more levels. Furthermore, there is more work needed to verify the claims made in terms of security, privacy, and performance guarantees. While name-registries have more mature implementations on blockchains, they also require feasibility analyses and recent works have pointed out a number of security vulnerabilities which need to be addressed. Furthermore, it is not clear whether true decentralisation is achieved. Out of the three key components, the file systems are the most mature and have working implementations and applications. Similar challenges and open questions remain in this field regarding decentralisation and key ownership, as well as legal and privacy concerns. A number of contradictions also arise from the self-certifying property of names, which ensures security but comes at a loss of usability.

REFERENCES

[1] 2001. Gnutella. www.gnutellanews.com.

[2] 2021. Accelerating Content Routing with Bitswap: A multi-path file transfer protocol in IPFS and Filecoin. (2021).

[3] 2022. SteemIt. https://steemit.com/.

[4] Marwan Abu-Amara, Farag Azzedin, Fahd A. Abdulhameed, Ashraf Mahmoud, and Mohammed H. Sqalli. 2011. Dynamic peer-to-peer (P2P) solution to counter malicious higher Domain Name System (DNS) nameservers. In *2011 24th Canadian Conference on Electrical and Computer Engineering(CCECE)*. 001014–001018. https://doi.org/10.1109/CCECE.2011.6030613

[5] Bengt Ahlgren, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, and Borje Ohlman. 2012. A survey of information-centric networking. *IEEE Communications Magazine* 50, 7 (2012), 26–36. https://doi.org/10.1109/MCOM.2012.6231276

[6] Mustafa Al-Bassam, Alberto Sonnino, Michal Król, and Ioannis Psaras. 2018. Airtnt: Fair Exchange Payment for Outsourced Secure Enclave Computations. *CoRR* abs/1805.06411 (2018). arXiv:1805.06411 http://arxiv.org/abs/1805.06411

[7] John H Aldrich, Rachel K Gibson, Marta Cantijoch, and Tobias Konitzer. 2016. Getting out the vote in the social media era: Are digital tools changing the extent, nature and impact of party contacting in elections? *Party Politics* 22, 2 (2016), 165–178.

[8] Faizan Safdar Ali and Alptekin Küpçü. 2020. Improving PKI, BGP, and DNS Using Blockchain: A Systematic Review. *CoRR* abs/2001.00747 (2020). arXiv:2001.00747 http://arxiv.org/abs/2001.00747

[9] Muneeb Ali. 2020. Stacks 2.0: Apps and Smart Contracts for Bitcoin.

[10] Muneeb Ali, Jude Nelson, Ryan Shea, and Michael J Freedman. 2016. Blockstack: A global naming and storage system secured by blockchains. In *2016 {USENIX} Annual Technical Conference ({USENIX} {ATC} 16)*. 181–194.

[11] Almonit. 2021. Almonit. https://almonit.eth.link/.

[12] Stephanos Androutsellis-Theotokis and Diomidis Spinellis. 2004. A survey of peer-to-peer content distribution technologies. *ACM computing surveys (CSUR)* 36, 4 (2004), 335–371.

[13] Nasreen Anjum, Dmytro Karamshuk, Mohammad Shikh-Bahaei, and Nishanth Sastry. 2017. Survey on peer-assisted content delivery networks. *Computer Networks* 116 (2017), 79–95.

[14] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. 2005. *DNS Security Introduction and Requirements*. RFC 4033. RFC Editor. http://www.rfc-editor.org/rfc/rfc4033.txt http://www.rfc-editor.org/rfc/rfc4033.txt.

[15] Jari Arkko, Brian Trammell, Mark Nottingham, Christian Huitema, Martin Thomson, Jeff Tantsura, and Niels ten Oever. 2020. *Considerations on Internet Consolidation and the Internet Architecture*. Technical Report. IETF.

[16] Onur Ascigil, Sergi Reñé, Michał Król, George Pavlou, Lixia Zhang, Toru Hasegawa, Yuki Koizumi, and Kentaro Kita. 2019. Towards Peer-to-Peer Content Retrieval Markets: Enhancing IPFS with ICN. In *Proceedings of the 6th ACM Conference on Information-Centric Networking* (Macao, China) *(ICN '19)*. Association for Computing Machinery, New York, NY, USA, 78–88. https://doi.org/10.1145/3357150.3357403

[17] Michael Backes, Ian Goldberg, Aniket Kate, and Tomas Toft. 2012. Adding query privacy to robust DHTs. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*. 30–31.

[18] Michael Backes, Aniket Kate, Matteo Maffei, and Kim Pecina. 2012. ObliviAd: Provably Secure and Practical Online Behavioral Advertising. In *2012 IEEE Symposium on Security and Privacy*. 257–271. https://doi.org/10.1109/SP.2012.25

[19] Leonhard Balduf, Sebastian Henningsen, Martin Florian, Sebastian Rust, and Björn Scheuermann. 2021. Monitoring Data Requests in Decentralized Data Storage Systems: A Case Study of IPFS. arXiv:2104.09202 [cs.NI]

[20] Leonhard Balduf, Maciej Korczyński, Onur Ascigil, Navin V. Keizer, George Pavlou, Björn Scheuermann, and Michał Król. 2023. The Cloud Strikes Back: Investigating the Decentralization of IPFS. In *Proceedings of the 2023 ACM on Internet Measurement Conference* (<conf-loc>, <city>Montreal QC</city>, <country>Canada</country>, </conf-loc>) *(IMC '23)*. Association for Computing Machinery, New York, NY, USA, 391–405. https://doi.org/10.1145/3618257.3624797

[21] Shehar Bano, Alberto Sonnino, Mustafa Al-Bassam, Sarah Azouvi, Patrick McCorry, Sarah Meiklejohn, and George Danezis. 2017. Consensus in the Age of Blockchains. *CoRR* abs/1711.03936 (2017). arXiv:1711.03936 http://arxiv.org/abs/1711.03936

[22] Ammar Battah, Youssef Iraqi, and Ernesto Damiani. 2021. Blockchain-Based Reputation Systems: Implementation Challenges and Mitigation. *Electronics* 10 (01 2021). https://doi.org/10.3390/electronics10030289

[23] Emanuele Bellini, Youssef Iraqi, and Ernesto Damiani. 2020. Blockchain-Based Distributed Trust and Reputation Management Systems: A Survey. *IEEE Access* 8 (2020), 21127–21151. https://doi.org/10.1109/ACCESS.2020.2969820

[24] Juan Benet. 2014. Ipfs-content addressed, versioned, p2p file system. (2014). https://arxiv.org/abs/1407.3561

[25] Juan Benet, David Dalrymple, and Nicola Greco. 2017. *Proof of Replication Technical Report (WIP)*. Technical Report. Protocol Labs.

[26] J Benet and N Greco. 2018. Filecoin: A Decentralized Storage Network. *Protocol Labs* (2018).

[27] Nazanin Zahed Benisi, Mehdi Aminian, and Bahman Javadi. 2020. Blockchain-based decentralized storage networks: A survey. *J. Netw. Comput. Appl.* 162 (2020), 102656.

[28] Brendan Benshoof, Andrew Rosen, Anu G. Bourgeois, and Robert W. Harrison. 2016. Distributed Decentralized Domain Name Service. In *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. 1279–1287. https://doi.org/10.1109/IPDPSW.2016.109

[29] Tim Berners-Lee, Robert Cailliau, Ari Luotonen, Henrik Frystyk Nielsen, and Arthur Secret. 1994. The world-wide web. *Commun. ACM* 37, 8 (1994), 76–82.

[30] Ken Birman. 2007. The Promise, and Limitations, of Gossip Protocols. *SIGOPS Oper. Syst. Rev.* 41, 5 (oct 2007), 8–13. https://doi.org/10.1145/1317379.1317382

[31] Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems* 30, 1 (1998), 107–117. https://doi.org/10.1016/S0169-7552(98)00110-X Proceedings of the Seventh International World Wide Web Conference.

[32] Andrei Broder. 2002. A Taxonomy of Web Search. *SIGIR Forum* 36, 2 (Sept. 2002), 3–10. https://doi.org/10.1145/792550.792552

[33] Vitalik Buterin. 2015. A Next-Generation Smart Contract and Decentralized Application Platform.

[34] Vitalik Buterin and Virgil Griffith. 2017. Casper the Friendly Finality Gadget. (2017). http://arxiv.org/abs/1710.09437

[35] Sarven Capadisli, Tim Berners-Lee, Ruben Verborgh, Kjetil Kjernsmo, Justin Bingham, Dmitri Zagidulin, and Aaron Coburn. 2021. Solid Protocol Draft Version 0.9.0. https://solidproject.org/TR/protocol.

[36] Carlos Castillo. 2019. Fairness and Transparency in Ranking. *SIGIR Forum* 52, 2 (jan 2019), 64–71. https://doi.org/10.1145/3308774.3308783

[37] Yatin Chawathe, Sylvia Ratnasamy, Lee Breslau, Nick Lanham, and Scott Shenker. 2003. Making Gnutella-like P2P Systems Scalable. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* (Karlsruhe, Germany) *(SIGCOMM '03)*. Association for Computing Machinery, New York, NY, USA, 407–418. https://doi.org/10.1145/863955.864000

[38] Raymond Cheng, Fan Zhang, Jernej Kos, Warren He, Nicholas Hynes, Noah M. Johnson, Ari Juels, Andrew Miller, and Dawn Song. 2018. Ekiden: A Platform for Confidentiality-Preserving, Trustworthy, and Performant Smart Contract Execution. *CoRR* abs/1804.05141 (2018). arXiv:1804.05141 http://arxiv.org/abs/1804.05141

[39] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. 2001. *Freenet: A Distributed Anonymous Information Storage and Retrieval System.* Springer Berlin Heidelberg, Berlin, Heidelberg, 46–66. https://doi.org/10.1007/3-540-44702-4_4

[40] Eric K. Clemons. 2009. Business Models for Monetizing Internet Applications and Web Sites: Experience, Theory, and Predictions. *Journal of Management Information Systems* 26, 2 (2009), 15–41. https://doi.org/10.2753/MIS0742-1222260202 arXiv:https://doi.org/10.2753/MIS0742-1222260202

[41] Bram Cohen. 2003. Incentives build robustness in BitTorrent.

[42] Nxt Community. 2016. Nxt Whitepaper. https://nxtdocs.jelurida.com/Nxt_Whitepaper.

[43] Efthymios Constantinides and Stefan J Fountain. 2008. Web 2.0: Conceptual foundations and marketing issues. *Journal of direct, data and digital marketing practice* 9 (2008), 231–244.

[44] Landon P Cox and Brian D Noble. 2003. Samsara: Honor among thieves in peer-to-peer storage. *ACM SIGOPS Operating Systems Review* 37, 5 (2003), 120–132.

[45] Russ Cox, Athicha Muthitacharoen, and Robert Morris. 2002. Serving DNS Using a Peer-to-Peer Lookup Service. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems (IPTPS '01)*. Springer-Verlag, Berlin, Heidelberg, 155–165.

[46] Anthony Cuthbertson. 2021. FaceBook down: Users report issues with messenger and instagram. *The Independent* (2021). https://www.independent.co.uk/life-style/gadgets-and-tech/facebook-down-messenger-instagram-not-working-b1950938.html

[47] Hung Dang, Dat Le Tien, and Ee-Chien Chang. 2018. Fair Marketplace for Secure Outsourced Computations. *CoRR* abs/1808.09682 (2018). arXiv:1808.09682 http://arxiv.org/abs/1808.09682

[48] Erik Daniel and Florian Tschorsch. 2022. Ipfs and friends: A qualitative comparison of next generation peer-to-peer data networks.

[49] Christian Dannewitz, Jovan Golic, Borje Ohlman, and Bengt Ahlgren. 2010. Secure naming for a network of information. In *2010 INFOCOM IEEE conference on computer communications workshops*. IEEE, 1–6.

[50] Bernhard Debatin, Jennette P Lovejoy, Ann-Kathrin Horn, and Brittany N Hughes. 2009. Facebook and online privacy: Attitudes, behaviors, and unintended consequences. *Journal of computer-mediated communication* 15, 1 (2009), 83–108.

[51] Richard Dennis and Gareth Huw Owenson. 2016. Rep on the roll: a peer to peer reputation system based on a rolling blockchain. (2016).

[52] Dominic Deuber, Nico Döttling, Bernardo Magri, Giulio Malavolta, and Sri Aravinda Krishnan Thyagarajan. 2018. Minting Mechanisms for Blockchain – or – Moving from Cryptoassets to Cryptocurrencies. Cryptology ePrint Archive, Report 2018/1110. https://ia.cr/2018/1110.

[53] Roger Dingledine, Nick Mathewson, and Paul Syverson. 2004. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13* (San Diego, CA) *(SSYM'04)*. USENIX Association, USA, 21.

[54] Roger Dingledine, Nick Mathewson, and Paul Syverson. 2004. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13* (San Diego, CA) *(SSYM'04)*. USENIX Association, USA, 21.

[55] Trinh Viet Doan, Yiannis Psaras, Jörg Ott, and Vaibhav Bajpai. 2022. Towards Decentralised Cloud Storage with IPFS: Opportunities, Challenges, and Future Considerations. *arXiv preprint arXiv:2202.06315* (2022).

[56] Unstoppable Domains. 2020. Architecture overview. https://docs.unstoppabledomains.com/domain-registry-essentials/architecture-overview.

[57] Xinan Duan, Zhiwei Yan, Guanggang Geng, and Baoping Yan. 2018. DNSLedger: Decentralized and distributed name resolution for ubiquitous IoT. In *2018 IEEE International Conference on Consumer Electronics (ICCE)*. 1–3. https://doi.org/10.1109/ICCE.2018.8326118

[58] Emercoin. 2021. EmerDNS. https://emercoin.com/en/emerdns.

[59] Peter Kairouz et al. 2019. Advances and Open Problems in Federated Learning. (2019).

[60] Faroo. 2007. Faroo. https://web.archive.org/web/20150914205049/http://www.faroo.com/hp/p2p/p2p.html.

[61] Bryan Ford, Jacob Strauss, Chris Lesniewski-Laas, Sean Rhea, Frans Kaashoek, and Robert Morris. 2006. Persistent Personal Names for Globally Connected Mobile Devices. In *3rd USENIX Workshop on Real, Large Distributed Systems (WORLDS 06)*. USENIX Association, Seattle, WA. https://www.usenix.org/conference/worlds-06/persistent-personal-names-globally-connected-mobile-devices

[62] Satoshi Fujita. 2021. Similarity Search in InterPlanetary File System with the Aid of Locality Sensitive Hash. *IEICE TRANSACTIONS on Information and Systems* 104, 10 (2021), 1616–1623.

[63] Blaise Gassend, Thomer M. Gil, and Bin Song. 2001. DINX: A Decentralized Search Engine. (2001).

[64] Arthur Gervais, Ghassan O. Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. 2016. On the Security and Performance of Proof of Work Blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (Vienna, Austria) *(CCS '16)*. Association for Computing Machinery, New York, NY, USA, 3–16. https://doi.org/10.1145/2976749.2978341

[65] Milad Ghaznavi, Elaheh Jalalpour, Mohammad A. Salahuddin, Raouf Boutaba, Daniel Migault, and Stere Preda. 2021. Content Delivery Network Security: A Survey. *IEEE Communications Surveys Tutorials* 23, 4 (2021), 2166–2190. https://doi.org/10.1109/COMST.2021.3093492

[66] Ali Ghodsi, Teemu Koponen, Jarno Rajahalme, Pasi Sarolahti, and Scott Shenker. 2011. Naming in Content-Oriented Architectures. In *Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking* (Toronto, Ontario, Canada) *(ICN '11)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/2018584.2018586

[67] Mainak Ghosh, Miles Richardson, Brian Ford, and Rob Jansen. 2014. A TorPath to TorCoin: Proof-of-Bandwidth Altcoins for Compensating Relays. *7th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs)*.

[68] Phillipa Gill, Martin Arlitt, Zongpeng Li, and Anirban Mahanti. 2007. Youtube traffic characterization: a view from the edge. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. 15–28.

[69] April Glaser. 2018. How apple and amazon are aiding chinese censors. (2018).

[70] Oded Goldreich and Yair Oren. 1994. Definitions And Properties Of Zero-Knowledge Proof Systems. *Journal of Cryptology* 7 (1994), 1–32.

[71] Scarlett Gourley and Hitesh Tewari. 2018. Blockchain Backed DNSSEC.

[72] Christian Grothoff. 2017. The GNUnet System. (2017).

[73] Lewis Gudgeon, Pedro Moreno-Sanchez, Stefanie Roos, Patrick McCorry, and Arthur Gervais. 2020. SoK: Layer-Two Blockchain Protocols. In *Financial Cryptography and Data Security*, Joseph Bonneau and Nadia Heninger (Eds.). Springer International Publishing, Cham, 201–226.

[74] Saikat Guha, Bin Cheng, and Paul Francis. 2011. Privad: Practical Privacy in Online Advertising. In *8th USENIX Symposium on Networked Systems Design and Implementation (NSDI 11)*. USENIX Association, Boston, MA. https://www.usenix.org/conference/nsdi11/privad-practical-privacy-online-advertising

[75] Mark Handley and Adam Greenhalgh. 2005. The Case for Pushing DNS. (01 2005).

[76] Handshake. 2018. Handshake Whitepaper. https://handshake.org/files/handshake.txt.

[77] R. Hasan, Z. Anwar, W. Yurcik, L. Brumbaugh, and R. Campbell. 2005. A survey of peer-to-peer storage techniques for distributed file systems. In *International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume II*, Vol. 2. 205–213 Vol. 2. https://doi.org/10.1109/ITCC.2005.42

[78] Sebastian Henningsen, Martin Florian, Sebastian Rust, and Björn Scheuermann. 2020. Mapping the Interplanetary Filesystem. In *2020 IFIP Networking Conference (Networking)*. 289–297.

[79] Sebastian Henningsen, Sebastian Rust, Martin Florian, and Björn Scheuermann. 2020. Crawling the IPFS Network. In *2020 IFIP Networking Conference (Networking)*. 679–680.

[80] Michael Herrmann, Kai-Chun Ning, Claudia Díaz, and Bart Preneel. 2014. Description of the YaCy Distributed Web Search Engine.

[81] M. Herrmann, R. Zhang, K. Ning, C. Diaz, and B. Preneel. 2014. Censorship-resistant and privacy-preserving distributed web search. In *14-th IEEE International Conference on Peer-to-Peer Computing*. 1–10.

[82] Stephen Herwig, Christina Garman, and Dave Levin. 2020. Achieving keyless cdns with conclaves. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*. 735–751.

[83] ipfs search. 2021. ipfs-search documentation. https://ipfs-search.readthedocs.io/en/latest/index.html.

[84] Tomas Isdal, Michael Piatek, Arvind Krishnamurthy, and Thomas Anderson. 2010. Privacy-preserving p2p data sharing with oneswarm. *ACM SIGCOMM Computer Communication Review* 40, 4 (2010), 111–122.

[85] Wael Issa, Nour Moustafa, Benjamin Turnbull, Nasrin Sohrabi, and Zahir Tari. 2023. Blockchain-based federated learning for securing internet of things: A comprehensive survey. *Comput. Surveys* 55, 9 (2023), 1–43.

[86] Qingmin Jia, Renchao Xie, Tao Huang, Jiang Liu, and Yunjie Liu. 2017. The Collaboration for Content Delivery and Network Infrastructures: A Survey. *IEEE Access* 5 (2017), 18088–18106.

[87] Audun Jøsang, Roslan Ismail, and Colin Boyd. 2007. A Survey of Trust and Reputation Systems for Online Service Provision. (2007).

[88] Harry A. Kalodner, Miles Carlsten, Paul Ellenbogen, Joseph Bonneau, and Arvind Narayanan. 2015. An Empirical Study of Namecoin and Lessons for Decentralized Namespace Design. In *14th Annual Workshop on the Economics of Information Security, WEIS 2015, Delft, The Netherlands, 22-23 June, 2015*. http://www.econinfosec.org/archive/weis2015/papers/WEIS_2015_kalodner.pdf

[89] Enis Karaarslan and Eylul Adiguzel. 2018. Blockchain Based DNS and PKI Solutions. *IEEE Communications Standards Magazine* 2, 3 (2018), 52–57. https://doi.org/10.1109/MCOMSTD.2018.1800023

[90] Navin V Keizer, Onur Ascigil, Michał Król, and George Pavlou. 2023. Ditto: Towards Decentralised Similarity Search for Web3 Services. In *2023 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS)*. IEEE, 66–75.

[91] Navin V. Keizer and Puneet Bindlish. 2021. Deece Search: Decentralised Search for IPFS. https://github.com/navinkeizer/Deece.

[92] Navin V. Keizer, Fan Yang, Ioannis Psaras, and George Pavlou. 2021. The Case for AI Based Web3 Reputation Systems. In *2021 IFIP Networking Conference (IFIP Networking)*. 1–2. https://doi.org/10.23919/IFIPNetworking52078.2021.9472783

[93] Nawras Khudhur and Satoshi Fujita. 2019. Siva-The IPFS search engine. In *2019 Seventh International Symposium on Computing and Networking (CANDAR)*. IEEE, 150–156.

[94] Mei Kobayashi and Koichi Takeda. 2000. Information retrieval on the web. *ACM Computing Surveys (CSUR)* 32, 2 (2000), 144–173.

[95] Maciej Korczyński, Michał Król, and Michel van Eeten. 2016. Zone poisoning: The how and where of non-secure DNS dynamic updates. In *Proceedings of the 2016 Internet Measurement Conference*. 271–278.

[96] Michal Król and Ioannis Psaras. 2018. SPOC: Secure Payments for Outsourced Computations. *CoRR* abs/1807.06462 (2018). arXiv:1807.06462 http://arxiv.org/abs/1807.06462

[97] Michał Król, Alberto Sonnino, Mustafa Al-Bassam, Argyrios Tasiopoulos, and Ioannis Psaras. 2019. Proof-of-prestige: A useful work reward system for unverifiable tasks. In *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE, 293–301.

[98] Sébastien Lahaie, David Pennock, Amin Saberi, and Rakesh Vohra. 2007. Sponsored search auctions. *Algorithmic Game Theory* (01 2007). https://doi.org/10.1017/CBO9780511800481.030

[99] Ziliang Lai, Chris Liu, Eric Lo, Ben Kao, and Siu-Ming Yiu. 2018. Decentralized Search on Decentralized Web. *CoRR* abs/1809.00939 (2018). arXiv:1809.00939 http://arxiv.org/abs/1809.00939

[100] Nick Lambert and Benjamin Bollen. 2014. The SAFE Network: a New, Decentralised Internet. (2014).

[101] N. Leibowitz, M. Ripeanu, and A. Wierzbicki. 2003. Deconstructing the Kazaa network. In *Proceedings the Third IEEE Workshop on Internet Applications. WIAPP 2003*. 112–120. https://doi.org/10.1109/WIAPP.2003.1210295

[102] Jiyang Li, Boon Loo, Joseph Hellerstein, M Kaashoek, David Karger, and Robert Morris. 2003. On the Feasibility of Peer-to-Peer Web Indexing and Search. (10 2003).

[103] Mingyu Li, Jinhao Zhu, Tianxu Zhang, Cheng Tan, Yubin Xia, Sebastian Angel, and Haibo Chen. 2021. Bringing Decentralized Search to Decentralized Services. In *15th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 21)*. 331–347.

[104] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. 2021. A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection. arXiv:1907.09693 [cs.LG]

[105] Qian Li, Tao Zhou, Linyuan Lv, and Duanbing Chen. 2013. Identifying Influential Spreaders by Weighted LeaderRank. arXiv:1306.5042 [physics.soc-ph]

[106] Xi Li, Zehua Wang, Victor C. M. Leung, Hong Ji, Yiming Liu, and Heli Zhang. 2021. Blockchain-Empowered Data-Driven Networks: A Survey and Outlook. *ACM Comput. Surv.* 54, 3, Article 58 (apr 2021), 38 pages. https://doi.org/10.1145/3446373

[107] Wilson Lian, Eric Rescorla, Hovav Shacham, and Stefan Savage. 2013. Measuring the Practical Impact of DNSSEC Deployment. In *22nd USENIX Security Symposium (USENIX Security 13)*. USENIX Association, Washington, D.C., 573–588. https://www.usenix.org/conference/usenixsecurity13/technical-sessions/paper/lian

[108] Wei Liang, Yongkai Fan, Kuan-Ching Li, Dafang Zhang, and Jean-Luc Gaudiot. 2020. Secure Data Storage and Recovery in Industrial Blockchain Network Environments. *IEEE Transactions on Industrial Informatics* 16, 10 (2020), 6543–6552. https://doi.org/10.1109/TII.2020.2966069

[109] Dongxiao Liu, Cheng Huang, Jianbing Ni, Xiaodong Lin, and Xuemin Shen. 2021. Blockchain-Based Smart Advertising Network With Privacy-Preserving Accountability. *IEEE Transactions on Network Science and Engineering* 8, 3 (2021), 2118–2130. https://doi.org/10.1109/TNSE.2020.3027796

[110] Jingqiang Liu, Bin Li, Lizhang Chen, Meng Hou, Feiran Xiang, and Peijun Wang. 2018. A Data Storage Method Based on Blockchain for Decentralization DNS. In *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*. 189–196. https://doi.org/10.1109/DSC.2018.00035

[111] Wenfeng Liu, Yu Zhang, Lu Liu, Shuyan Liu, Hongli Zhang, and Binxing Fang. 2020. A secure domain name resolution and management architecture based on blockchain. In *2020 IEEE Symposium on Computers and Communications (ISCC)*. 1–7. https://doi.org/10.1109/ISCC50000.2020.9219632

[112] Boon Thau Loo, Ryan Huebsch, Ion Stoica, and Joseph M. Hellerstein. 2004. The Case for a Hybrid P2p Search Infrastructure. In *Proceedings of the Third International Conference on Peer-to-Peer Systems* (La Jolla, CA) *(IPTPS'04)*. Springer-Verlag, Berlin, Heidelberg, 141–150. https://doi.org/10.1007/978-3-540-30183-7_14

[113] Pedro García López, Alberto Montresor, and Anwitaman Datta. 2019. Please, do not decentralize the Internet with (permissionless) blockchains! *CoRR* abs/1904.13093 (2019). arXiv:1904.13093 http://arxiv.org/abs/1904.13093

[114] Tim Lu, Shan Sinha, and Ajay Sudan. 2003. Panache: A Scalable Distributed Index for Keyword Search. (01 2003).

[115] Eng Keong Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. 2005. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys Tutorials* 7, 2 (2005), 72–93. https://doi.org/10.1109/COMST.2005.1610546

[116] Qin Lv, Pei Cao, Edith Cohen, Kai Li, and Scott Shenker. 2002. Search and Replication in Unstructured Peer-to-Peer Networks. In *Proceedings of the 16th International Conference on Supercomputing* (New York, New York, USA) *(ICS '02)*. Association for Computing Machinery, New York, NY, USA, 84–95. https://doi.org/10.1145/514191.514206

[117] Pavel Mach and Zdenek Becvar. 2017. Mobile Edge Computing: A Survey on Architecture and Computation Offloading. *IEEE Communications Surveys Tutorials* 19, 3 (2017), 1628–1656. https://doi.org/10.1109/COMST.2017.2682318

[118] Yuyi Mao, Changsheng You, Jun Zhang, Kaibin Huang, and Khaled B. Letaief. 2017. A Survey on Mobile Edge Computing: The Communication Perspective. *IEEE Communications Surveys Tutorials* 19, 4 (2017), 2322–2358. https://doi.org/10.1109/COMST.2017.2745201

[119] Alexandre M. Mateus and Jon M. Peha. 2011. Quantifying Global Transfers of Copyrighted Content Using BitTorrent (September 24, 2011). In *TPRC 2011 - The 39th Research Conference on Communication, Information and Internet Policy*.

[120] Antony Mayfield. 2008. What is social media. (2008).

[121] Petar Maymounkov and David Mazières. 2002. Kademlia: A Peer-to-Peer Information System Based on the XOR Metric. In *Peer-to-Peer Systems*, Peter Druschel, Frans Kaashoek, and Antony Rowstron (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 53–65.

[122] David Mazieres and Frans Kaashoek. 2000. Self-certifying File System.

[123] Miti Mazmudar, Stan Gurtler, and Ian Goldberg. 2021. Do you feel a chill? Using PIR against chilling effects for censorship-resistant publishing. In *Proceedings of the 20th Workshop on Workshop on Privacy in the Electronic Society*. 53–57.

[124] Guillaume Michel. 2022. Double-Hashing as a way to increase reader privacy.

[125] Sebastian Michel, Peter Triantafillou, and Gerhard Weikum. 2005. MINERVA ∞: A Scalable Efficient Peer-to-Peer Search Engine. In *Proceedings of the ACM/IFIP/USENIX 6th International Conference on Middleware* (Grenoble, France) *(Middleware'05)*. Springer-Verlag, Berlin, Heidelberg, 60–81. https://doi.org/10.1007/11587552_4

[126] Dejan S. Milojicic, Vana Kalogeraki, Rajan Lukose, and Kiran Nagaraja. 2002. *Peer-to-Peer Computing*. Technical Report.

[127] Paul V Mockapetris. 1987. RFC 1035: Domain names-implementation and specification.

[128] Arash Molavi Kakhki, Chloe Kliman-Silver, and Alan Mislove. 2013. Iolaus: Securing online content rating systems. In *Proceedings of the 22nd international conference on World Wide Web*. 919–930.

[129] Radha Mookerjee, Subodha Kumar, and Vijay S Mookerjee. 2017. Optimizing performance-based internet advertisement campaigns. *Operations Research* 65, 1 (2017), 38–54.

[130] Vijay S. Mookerjee and Yong Tan. 2002. Analysis of a Least Recently Used Cache Management Policy for Web Browsers. *Operations Research* 50, 2 (2002), 345–357. http://www.jstor.org/stable/3088501

[131] Malte Möser, Ittay Eyal, and Emin Gün Sirer. 2016. Bitcoin Covenants. In *Financial Cryptography and Data Security*, Jeremy Clark, Sarah Meiklejohn, Peter Y.A. Ryan, Dan Wallach, Michael Brenner, and Kurt Rohloff (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 126–141.

[132] Alex Murray, Dennie Kim, and Jordan Combs. 2023. The promise of a decentralized internet: What is Web3 and how can firms prepare? *Business Horizons* 66, 2 (2023), 191–202.

[133] San Murugesan. 2007. Understanding Web 2.0. *IT professional* 9, 4 (2007), 34–41.

[134] Satoshi Nakamoto. 2009. Bitcoin: A peer-to-peer electronic cash system. http://www.bitcoin.org/bitcoin.pdf

[135] Namebase. 2021. Namebase. https://learn.namebase.io.

[136] NameCoin. 2011. NameCoin. https://www.namecoin.org/.

[137] Nebulas. 2018. Nebulas Technical White Paper. https://nebulas.io/docs/NebulasTechnicalWhitepaper.pdf.

[138] Till Neudecker and Hannes Hartenstein. 2019. Network Layer Aspects of Permissionless Blockchains. *IEEE Communications Surveys Tutorials* 21, 1 (2019), 838–857. https://doi.org/10.1109/COMST.2018.2852480

[139] Tim O'reilly. 2009. *What is web 2.0*. " O'Reilly Media, Inc.".

[140] Athanasios Papagelis and Christos Zaroliagis. 2012. A Collaborative Decentralized Approach to Web Search. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 42, 5 (2012), 1271–1290. https://doi.org/10.1109/TSMCA.2012.2187887

[141] Vasileios Pappas, Dan Massey, and Lixia Zhang. 2007. Enhancing DNS Resilience against Denial of Service Attacks. In *37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07)*. 450–459. https://doi.org/10.1109/DSN.2007.42

[142] Eli Pariser. 2011. *The Filter Bubble: What the Internet Is Hiding from You*. Penguin Group , The.

[143] Constantinos Patsakis, Fran Casino, Nikolaos Lykousas, and Vasilios Katos. 2020. Unravelling Ariadne's Thread: Exploring the Threats of Decentralised DNS. *IEEE Access* 8 (2020), 118559–118571. https://doi.org/10.1109/ACCESS.2020.3004727

[144] Andrew Perrin. 2015. Social media usage. *Pew research center* 125 (2015), 52–68.

[145] Johan Pouwelse, Paweł Garbacki, Dick Epema, and Henk Sips. 2005. The Bittorrent P2p File-Sharing System: Measurements and Analysis. In *Proceedings of the 4th International Conference on Peer-to-Peer Systems* (Ithaca, NY) *(IPTPS'05)*. Springer-Verlag, Berlin, Heidelberg, 205–216. https://doi.org/10.1007/11558989_19

[146] Presearch. 2017. Presearch Whitepaper. https://www.presearch.io/uploads/WhitePaper.pdf.

[147] Matti Pärssinen, Mikko Kotila, Rubén Cuevas Rumin, Amit Phansalkar, and Jukka Manner. 2018. Is Blockchain Ready to Revolutionize Online Advertising? *IEEE Access* 6 (2018), 54884–54899. https://doi.org/10.1109/ACCESS.2018.2872694

[148] Yi Qiao and Fabián E. Bustamante. 2006. Structured and Unstructured Overlays under the Microscope: A Measurement-based View of Two P2P Systems That People Use. In *2006 USENIX Annual Technical Conference (USENIX ATC 06)*. USENIX Association, Boston, MA. https://www.usenix.org/conference/2006-usenix-annual-technical-conference/structured-and-unstructured-overlays-under

[149] Tao Qin, Wei Chen, and Tie-Yan Liu. 2015. Sponsored Search Auctions: Recent Advances and Future Directions. *ACM Trans. Intell. Syst. Technol.* 5, 4, Article 60 (jan 2015), 34 pages. https://doi.org/10.1145/2668108

[150] Brandon Ramirez. 2020. The Graph Network In Depth. https://thegraph.com/blog/the-graph-network-in-depth-part-1.

[151] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. 2001. A Scalable Content-Addressable Network. In *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* (San Diego, California, USA) *(SIGCOMM '01)*. Association for Computing Machinery, New York, NY, USA, 161–172. https://doi.org/10.1145/383059.383072

[152] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. 2001. A Scalable Content-Addressable Network. *SIGCOMM Comput. Commun. Rev.* 31, 4 (aug 2001), 161–172. https://doi.org/10.1145/964723.383072

[153] A. Raza, K. Han, and S. O. Hwang. 2020. A Framework for Privacy Preserving, Distributed Search Engine Using Topology of DLT and Onion Routing. *IEEE Access* 8 (2020), 43001–43012.

[154] Nebulas Research. 2019. Yellow Paper: Nebulas Rank. https://github.com/nebulasio/nr-report/blob/master/en/main.pdf.

[155] Patrick Reynolds and Amin Vahdat. 2003. Efficient Peer-to-Peer Keyword Searching. In *Proceedings of the ACM/IFIP/USENIX 2003 International Conference on Middleware* (Rio de Janeiro, Brazil) *(Middleware '03)*. Springer-Verlag, Berlin, Heidelberg, 21–40.

[156] M. Ripeanu. 2001. Peer-to-peer architecture case study: Gnutella network. In *Proceedings First International Conference on Peer-to-Peer Computing*. 99–100. https://doi.org/10.1109/P2P.2001.990433

[157] Antony Rowstron and Peter Druschel. 2001. Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In *Middleware 2001*, Rachid Guerraoui (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 329–350.

[158] Pratik Satam, H Alipour, Youssif Al-Nashif, and Salim Hariri. 2015. Anomaly behavior analysis of DNS protocol. *J. Internet Serv. Inf. Secur* 5 (01 2015).

[159] Mahadev Satyanarayanan. 2017. The emergence of edge computing. *Computer* 50, 1 (2017), 30–39.

[160] Philip Saunders. 2016. Nebulis.

[161] Seeks. 2014. Seeks FAQ. https://github.com/beniz/seeks/wiki/FAQ.

[162] Ethereum Name Service. 2021. ENS Documentation. https://docs.ens.domains.

[163] A. Singh, M. Srivatsa, L. Liu, and T. Miller. 2003. Apoidea: A Decentralized Peer-to-Peer Architecture for Crawling the World Wide Web. In *Distributed Multimedia Information Retrieval*.

[164] Brave Software. 2021. Basic Attention Token (BAT) Blockchain Based Digital Advertising. https://basicattentiontoken.org/static-assets/documents/BasicAttentionTokenWhitePaper-4.pdf.

[165] Stacks. 2021. Blockchain Naming System. https://docs.stacks.co/build-apps/references/bns.

[166] Moritz Steiner, Taoufik En-Najjary, and Ernst W. Biersack. 2007. A Global View of Kad. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement* (San Diego, California, USA) *(IMC '07)*. Association for Computing Machinery, New York, NY, USA, 117–122. https://doi.org/10.1145/1298306.1298323

[167] U. Steinhoff, A. Wiesmaier, and R. Araújo. 2006. The State of the Art in DNS Spoofing.

[168] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. 2001. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. *SIGCOMM Comput. Commun. Rev.* 31, 4 (Aug. 2001), 149–160. https://doi.org/10.1145/964723.383071

[169] Torsten Suel, Chandan Mathur, Jo-wen Wu, Jiangong Zhang, Alex Delis, Mehdi Kharrazi, Xiaohui Long, and Kulesh Shanmugasundaram. 2003. ODISSEA: A Peer-to-Peer Architecture for Scalable Web Search and Information Retrieval. (06 2003).

[170] Swarm. 2021. Swarm: Storage and Communication Infrastructure for a Self-sovereign Digital Society. https://www.ethswarm.org/swarm-whitepaper.pdf.

[171] Jake Swearingen. 2018. When Amazon web services goes down, so does a lot of the web. *New York Magazine* (2018).

[172] Jake Swearingen. 2018. When Amazon Web Services Goes Down, So Does a Lot of the Web. (2018). http://nymag.com/selectall/2018/03/when-amazon-webservices-goes-down-so-does-a-lot-of-the-web.html

[173] Wesley W. Terpstra, Jussi Kangasharju, Christof Leng, and Alejandro P. Buchmann. 2007. Bubblestorm: Resilient, Probabilistic, and Exhaustive Peer-to-Peer Search. *SIGCOMM Comput. Commun. Rev.* 37, 4 (Aug. 2007), 49–60. https://doi.org/10.1145/1282427.1282387

[174] M. Theimer and M.B. Jones. 2002. Overlook: scalable name service on an overlay network. In *Proceedings 22nd International Conference on Distributed Computing Systems*. 52–61. https://doi.org/10.1109/ICDCS.2002.1022242

[175] Hien Tran, Tarek Menouer, Patrice Darmon, Abdoulaye Doucoure, and François Binder. 2019. Smart Contracts Search Engine in Blockchain. In *Proceedings of the 3rd International Conference on Future Networks and Distributed Systems* (Paris, France) *(ICFNDS '19)*. Association for Computing Machinery, New York, NY, USA, Article 24, 5 pages. https://doi.org/10.1145/3341325.3342015

[176] Dennis Trautwein, Aravindh Raman, Gareth Tyson, Ignacio Castro, Will Scott, Moritz Schubotz, Bela Gipp, and Yiannis Psaras. 2022. Design and evaluation of IPFS: a storage layer for the decentralized web. In *Proceedings of the ACM SIGCOMM 2022 Conference*. 739–752.

[177] Zied Trifa and Maher Ali Khemakhem. 2012. Taxonomy of Structured P2P Overlay Networks Security Attacks. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering* 6 (2012), 470–476.

[178] Imdad Ullah, Salil S. Kanhere, and Roksana Boreli. 2020. Privacy-preserving targeted mobile advertising: A Blockchain-based framework for mobile ads. *CoRR* abs/2008.10479 (2020). arXiv:2008.10479 https://arxiv.org/abs/2008.10479

[179] H. Unger and M. Wulff. 2003. Towards a decentralized search engine for P2P-network communities. In *Eleventh Euromicro Conference on Parallel, Distributed and Network-Based Processing, 2003. Proceedings*. 492–499. https://doi.org/10.1109/EMPDP.2003.1183630

[180] Eugene Y Vasserman, Victor Heorhiadi, Nicholas Hopper, and Yongdae Kim. 2012. One-Way Indexing for Plausible Deniability in Censorship Resistant Storage.. In *FOCI*.

[181] Dimitrios K. Vassilakis and Vasilis Vassalos. 2007. Modelling Real P2P Networks: The Effect of Altruism. In *Seventh IEEE International Conference on Peer-to-Peer Computing (P2P 2007)*. 19–26. https://doi.org/10.1109/P2P.2007.30

[182] William Vickrey. 1961. Counterspeculation, Auctions, and Competitive Sealed Tenders. *Journal of Finance* 16, 1 (1961), 8–37. https://EconPapers.repec.org/RePEc:bla:jfinan:v:16:y:1961:i:1:p:8-37

[183] David Vorick and Luke Champine. 2014. Sia: Simple decentralized storage. *Nebulous Inc* (2014).

[184] Dimitris Vyzovitis, Yusef Napora, Dirk McCormick, David Dias, and Yiannis Psaras. 2020. GossipSub: Attack-Resilient Message Propagation in the Filecoin and ETH2.0 Networks. *CoRR* abs/2007.02754 (2020). arXiv:2007.02754 https://arxiv.org/abs/2007.02754

[185] Jim Waldo. 2019. A Hitchhiker's Guide to the Blockchain Universe. *Commun. ACM* 62, 3 (Feb. 2019), 38–42. https://doi.org/10.1145/3303868

[186] Michael Walfish, Hari Balakrishnan, and Scott Shenker. 2004. Untangling the Web from DNS.. In *NSDI*, Vol. 4. 17–17.

[187] Feng Wang and Yanjun Wu. 2020. Keyword Search Technology in Content Addressable Storage System. In *2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE, 728–735.

[188] Gang Wang, Zhijie Jerry Shi, Mark Nixon, and Song Han. 2019. SoK: Sharding on Blockchain. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies* (Zurich, Switzerland) *(AFT '19)*. Association for Computing Machinery, New York, NY, USA, 41–61. https://doi.org/10.1145/3318041.3355457

[189] Qin Wang, Rujia Li, Qi Wang, and Shiping Chen. 2021. Non-fungible token (NFT): Overview, evaluation, opportunities and challenges. *arXiv preprint arXiv:2105.07447* (2021).

[190] Shuai Wang, Wenwen Ding, Juanjuan Li, Yong Yuan, Liwei Ouyang, and Fei-Yue Wang. 2019. Decentralized Autonomous Organizations: Concept, Model, and Applications. *IEEE Transactions on Computational Social Systems* 6, 5 (2019), 870–878. https://doi.org/10.1109/TCSS.2019.2938190

[191] Shuai Wang, Yong Yuan, Xiao Wang, Juanjuan Li, Rui Qin, and Fei-Yue Wang. 2018. An Overview of Smart Contract: Architecture, Applications, and Future Trends. In *2018 IEEE Intelligent Vehicles Symposium (IV)*. 108–113. https://doi.org/10.1109/IVS.2018.8500488

[192] Taotao Wang, Chonghe Zhao, Qing Yang, and Shengli Zhang. 2020. Ethna: Analyzing the Underlying Peer-to-Peer Network of the Ethereum Blockchain. *CoRR* abs/2010.01373 (2020). arXiv:2010.01373 https://arxiv.org/abs/2010.01373

[193] Xiangui Wang, Kedan Li, Hui Li, Yinghui Li, and Zhiwei Liang. 2017. ConsortiumDNS: A Distributed Domain Name Service Based on Consortium Chain. In *2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. 617–620. https://doi.org/10.1109/HPCC-SmartCity-DSS.2017.83

[194] Yuan Wang, Leonidas Galanis, and David J. DeWitt. 2005. GALANX: An Efficient Peer-to-Peer Search Engine System.

[195] Steve Waterhouse. 2001. JXTA Search: Distributed Search for Distributed Networks. (2001).

[196] HU Wei-hong, AO Meng, SHI Lin, XIE Jia-gui, and LIU Yang. 2017. Review of blockchain-based DNS alternatives. 3, 3, Article 71 (2017), 6 pages. https://doi.org/10.11959/j.issn.2096-109x.2017.00157

[197] B. Wellman and C. Haythornthwaite. 2008. *The Internet in Everyday Life*. Wiley. https://books.google.co.uk/books?id=v-UR_2QRFpwC

[198] Zooko Wilcox-O'Hearn. 2001. Names: Decentralized, Secure, Human-meaningful: Choose Two. https://web.archive.org/web/20011020191610/http://zooko.com/distnames.html.

[199] Shawn Wilkinson, Tome Boshevski, Josh Brandoff, and Vitalik Buterin. 2014. Storj a peer-to-peer cloud storage network. (2014).

[200] Sam Williams, Viktor Diordiiev, Lev Berman, India Raybould, and Ivan Uemlianin. 2019. Arweave: A Protocol for Economically Sustainable Information Permanence. (2019).

[201] Gavin Wood. 2014. Ethereum: A secure decentralised generalised transaction ledger.

[202] Pengcheng Xia, Haoyu Wang, Zhou Yu, Xinyu Liu, Xiapu Luo, Guoai Xu, and Gareth Tyson. 2022. Challenges in Decentralized Name Management: The Case of ENS. In *Proceedings of the 22nd ACM Internet Measurement Conference* (Nice, France) *(IMC '22)*. Association for Computing Machinery, New York, NY, USA, 65–82. https://doi.org/10.1145/3517745.3561469

[203] Jie Xu, Cong Wang, and Xiaohua Jia. 2023. A survey of blockchain consensus protocols. *Comput. Surveys* (2023).

[204] Jingting Xue, Chunxiang Xu, and Lanhua Bai. 2019. DStore: A distributed system for outsourced data storage and retrieval. *Future Generation Computer Systems* 99 (2019), 106–114. https://doi.org/10.1016/j.future.2019.04.022

[205] George Xylomenos, Christopher N. Ververidis, Vasilios A. Siris, Nikos Fotiou, Christos Tsilopoulos, Xenofon Vasilakos, Konstantinos V. Katsaros, and George C. Polyzos. 2014. A Survey of Information-Centric Networking Research. *IEEE Communications Surveys Tutorials* 16, 2 (2014), 1024–1049. https://doi.org/10.1109/SURV.2013.070813.00063

[206] YaCy. 2004. YaCy Decentralized Web Search. https://yacy.net.

[207] B. Yang and H. Garcia-Molina. 2002. Improving search in peer-to-peer networks. In *Proceedings 22nd International Conference on Distributed Computing Systems*. 5–14. https://doi.org/10.1109/ICDCS.2002.1022237

[208] Kai-hsiang Yang and Jan-ming Ho. 2006. Proof: A DHT-Based Peer-to-Peer Search Engine. In *2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)*. 702–708. https://doi.org/10.1109/WI.2006.137

[209] Ruizhe Yang, F. Richard Yu, Pengbo Si, Zhaoxin Yang, and Yanhua Zhang. 2019. Integrated Blockchain and Edge Computing Systems: A Survey, Some Research Issues and Challenges. *IEEE Communications Surveys Tutorials* 21, 2 (2019), 1508–1532. https://doi.org/10.1109/COMST.2019.2894727

[210] Shi Yin, Yu Teng, Ning Hu, and Xu Dong Jia. 2020. Decentralization of DNS: Old Problems and New Challenges. In *Proceedings of the 2020 International Conference on Cyberspace Innovation of Advanced Technologies* (Guangzhou, China) *(CIAT 2020)*. Association for Computing Machinery, New York, NY, USA, 335–341. https://doi.org/10.1145/3444370.3444594

[211] Wondeuk Yoon, Indal Choi, and Daeyoung Kim. 2019. BlockONS: Blockchain based Object Name Service. In *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. 219–226. https://doi.org/10.1109/BLOC.2019.8751464

[212] Haifeng Yu, Chenwei Shi, Michael Kaminsky, Phillip B Gibbons, and Feng Xiao. 2009. Dsybil: Optimal sybil-resistance for recommendation systems. In *2009 30th IEEE Symposium on Security and Privacy*. IEEE, 283–298.

[213] Xixun Yu and Athanasios Vasilakos. 2017. A Survey of Verifiable Computation. *Mobile Networks and Applications* 22 (06 2017), 1–16. https://doi.org/10.1007/s11036-017-0872-3

[214] Rui Yuan, Yubin Xia, Haibo Chen, Binyu Zang, and Jan Xie. 2018. ShadowEth: Private Smart Contract on Public Blockchain. *J. Comput. Sci. Technol.* 33, 3 (2018), 542–556. https://doi.org/10.1007/s11390-018-1839-y

[215] Yuwei Zeng, Zang Tianning, Yongzheng Zhang, Xunxun Chen, and Yipeng Wang. 2019. A Comprehensive Measurement Study of Domain-Squatting Abuse. 1–6. https://doi.org/10.1109/ICC.2019.8761388

[216] Ben Y. Zhao, John D. Kubiatowicz, and Anthony D. Joseph. 2001. *Tapestry: An Infrastructure for Fault-Tolerant Wide-Area Location And.* Technical Report. USA.

[217] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen, and Huaimin Wang. 2018. Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services* 14 (10 2018), 352. https://doi.org/10.1504/IJWGS.2018.095647

[218] Liyan Zhu, Chuqiao Xiao, and Xueqing Gong. 2020. Keyword search in decentralized storage systems. *Electronics* 9, 12 (2020), 2041.

[219] Mirko Zichichi, Luca Serena, Stefano Ferretti, and Gabriele D'Angelo. 2021. Governing Decentralized Complex Queries Through a DAO. In *Proceedings of the Conference on Information Technology for Social Good.* 121–126.

[220] Behrouz Zolfaghari, Gautam Srivastava, Swapnoneel Roy, Hamid R. Nemati, Fatemeh Afghah, Takeshi Koshiba, Abolfazl Razi, Khodakhast Bibak, Pinaki Mitra, and Brijesh Kumar Rai. 2020. Content Delivery Networks: State of the Art, Trends, and Future Roadmap. *ACM Comput. Surv.* 53, 2, Article 34 (apr 2020), 34 pages. https://doi.org/10.1145/3380613