OJ Logo

Received XX Month, XXXX; revised XX Month, XXXX; accepted XX Month, XXXX; Date of publication XX Month, XXXX; date of current version XX Month, XXXX.

Digital Object Identifier 10.1109/OJIM.2022.1234567

Blockchain-based Decentralized Application: A Survey

Peilin Zheng †, MEMBER, IEEE, Zigui Jiang †, MEMBER, IEEE, Jiajing Wu #, MEMBER, IEEE, AND Zibin Zheng †, Fellow, IEEE

[1] †School of Software Engineering, Sun Yat-sen University, China
[2] #School of Computer Science and Engineering, Sun Yat-sen University, China

CORRESPONDING AUTHOR: Zigui Jiang (e-mail: jiangzg3@mail.sysu.edu.cn).

ABSTRACT

Blockchain-based decentralized applications (DApp) draw more attention with the increasing development and wide application of blockchain technologies. A wealth of funds are invested into the crowd-funding of various types of DApp. As reported in August 2022, there are more than 5,000 DApps with more than 1.67 million daily Unique Active Wallets (users). However, the definition, architectures, and classifications of the DApps are still not cleared up till now. This survey aims to provide a comprehensive overview of DApps for further research. First, the definitions and typical architectures of DApps are presented. Then we collect 3,118 popular DApps and categorize them into different types, and summarize their typical advantages and challenges. Finally, we provide an overview of the recent research problems of DApps from the perspectives of economics, security, and performance and then figure out promising research opportunities in the future.

INDEX TERMS Blockchain, Decentralized Application

I. Introduction

The idea of blockchain was first proposed as the underlying technology of Bitcoin [1]. A blockchain is usually maintained by peers in a P2P transaction network, where peers record transactions in a period of time and package them together into a block to join the blockchain. Blockchain technology is decentralized, tamper-resistant, and traceable [2]. On a blockchain, a smart contract [3] is an event-driven promise defined by the programming language. A smart contract on the blockchain can be invoked by sending a transaction to the blockchain peers, with the independent execution of every peer. Finally, the contract execution is finished, with the result returned to the blockchain. The protocol called consensus protocol keeps every peer having the same blockchain. Since such execution is independent of every peer, the result is controlled by all the participants and, therefore, can be trusted by everyone.

Decentralized applications were proposed much earlier than blockchain technology. Since blockchain-based decentralized applications (DApp) can enhance trustworthiness, decrease the cost of central trusted authority, and have wide applications (e.g., finance, IoT, data provenance, etc.), they have gained a lot of attention from both industry and academia in recent years [4]. In [5], decentralized applications are classified into two classes: fully anonymous decentralized applications and reputation-based decentralized applications. However, there is a substantial gray area between these two types. Therefore, the definition of blockchain-based decentralized applications is still undefined.

Although there are some surveys [4], [6], [7] about blockchain technologies, the definition, architectures, and categories of DApps are still unclear. Therefore a systematic overview of DApp is urgently needed for better understanding and further research work in different aspects.

This paper considers the blockchain-based decentralized application to be the application using blockchain as its underlying technology to ensure decentralized characteristics. In this paper, the architectures of blockchain-based decentralized applications are summarized into four types, which are Native Client as a DApp, Smart Contract as a DApp, Web & Contract as a DApp, and Fully-decentralized DApp, based on their different architectures.

In a narrow sense, the so-called DApps nowadays commonly refer to the third type, that is, the Web & Contract as a DApp. As reported [8], there are more than 5,000 DApps that belong to this type, of which the number of daily Unique Active Wallets was 1.67 million in August 2022. Hence this paper further investigates the Web & Contract as a DApp.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

An increasing amount of money has been devoted to the crowd-funding of such types of DApps as investments [9]. Therefore, various DApps need to be classified to improve the understanding of DApps. In this paper, 3,118 DApps are collected and categorized into different types, including the popular DeFi (Decentralized Finance), NFT (Non-Fungible Token), and GameFi (Game+DeFi) DApps in recent years. These categories are proposed with advantages over centralized solutions.

Moreover, this paper also discusses the recent research on DApps in the aspects of economics, security, and performance. As for economics, we summarize the economic problem of DApps into incentive policy, risk evaluation, and miner effect. As for DApp security, we divide the vulnerability into three layers: web, blockchain, and smart contract. As for performance, we compare tools with metrics. In these aspects, we present recent advances and research opportunities.

The main contributions of this paper are summarized as follows:

- We give a comprehensive survey on the definition and typical architectures of blockchain-based decentralized applications.
- We collect and categorize the blockchain-based decentralized applications. Meanwhile, we summarize their typical advantages over centralized solutions.
- We conduct an overview of the research problems on DApps from the aspects of economics, security, and performance to provide research opportunities for researchers who are interested in this field.

This survey provides a comprehensive overview of the research on blockchain-based decentralized applications. The rest of the survey is organized as follows. §II gives an introduction to the basic concepts. §III shows the definition and typical architectures of blockchain-based decentralized applications. §IV categorizes the decentralized applications and compares them with traditional centralized applications. §V propose the economic, security, performance problems, and corresponding solutions of DApps. §VI concludes the paper.

II. Basic Concepts

This section introduces the basic principles and concepts of blockchain, consensus protocol, and smart contracts.

A. Blockchain

In a narrow sense, blockchain is a kind of data structure. The concept of the blockchain was first proposed as the underlying storage for peer-to-peer payments in Bitcoin [10]. In a blockchain, every block contains transactions for a period of time. Then every block is joined to a chain-like data structure named blockchain. Each peer in the peer-to-peer network maintains a blockchain by itself. And the peer keeps it the same with each other via consensus protocols. Since each block has a hash value of itself and the hash value is contained in the next block, the content (e.g., timestamps, transactions) is tamper-resistant and traceable. It should be noted that the blockchain can be described as a comprehensive technology that includes the underlying data structure, consensus protocols [11], and upper applications [12] in a broad sense [6]. But in this paper, blockchain is considered as a kind of data structure. And the blockchain-based decentralized application is the application that uses this underlying data structure.

B. Consensus Protocol

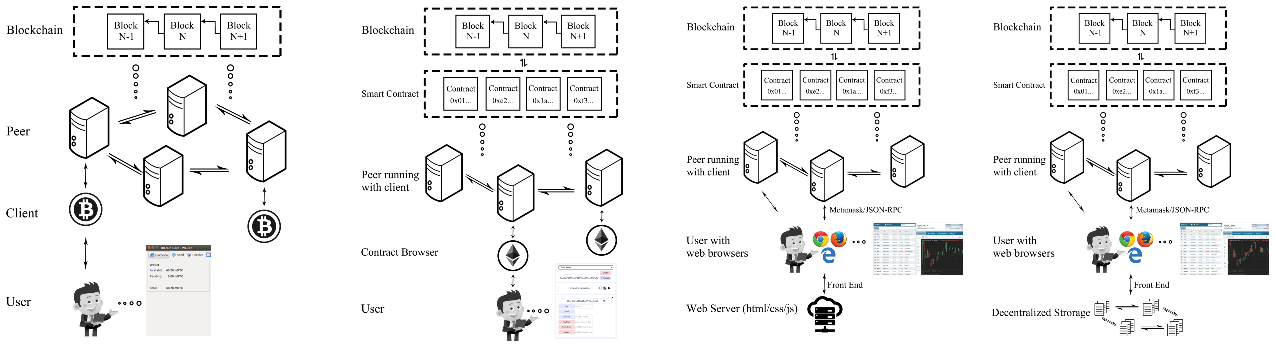
The consensus protocol [11] is a protocol that is implemented in every node of a blockchain system to keep them having the same ledger. Consensus algorithm has been developed in traditional distributed systems for years. But in blockchain systems, especially public blockchain, the peers have more motivation for dishonesty, so there are more problems, such as the double-spending problem. Thus the blockchain systems need different consensus protocols to balance the technical and economic motivations. Different DApps (or their underlying blockchains) use different consensus protocols. In this case, some of the problems with DApps in economics, security, and performance result from the consensus protocols, which will be shown in §V.

C. Smart Contract

The smart contract is a promise defined by digital form [13]. A blockchain-based smart contract is an event-driven promise defined by the programming language. A smart contract can be invoked in the way of sending a transaction (including the address of the contract, the calling function, and the parameters) to the validating peers. After that, the smart contract will be executed independently by each peer [14]. Finally, different peers reach a consensus and save the result back to the blockchain. Under some scenarios, a smart contract on blockchain could be considered a decentralized application. Yet it is still controversial. The different definitions and architectures of blockchain-based decentralized applications will be described in §III.

D. Growth

In the early stages of DApp development, most DApps are constructed on Ethereum [15]. However, with the development of DApps, the performance of Ethereum cannot afford the rapid growth of users. Therefore, there are more and more platforms (blockchains) developed and used by users. Nowadays, the other popular smart contract platforms are: BinanceSmartChain [16], EOSIO [17], TRON [18], Fantom [19], Polygon [20], Solana [21], Avalanche [22], and so on. There is no simple and intuitive evidence to compare the whole DApp ecology of each platform. However, we will introduce the metric of Total Value Locked (TVL) in Decentralized Finance (DeFi) to compare the financial DApps on these blockchains.



(a) Native Client as a DApp (b) Smart Contract as a DApp (c) Web & Contract as a DApp (d) Fully-decentralized DApp

FIGURE 1. **Architectures of Decentralized Applications**

III. Definition and Architecture

The decentralized application is the application that does not be controlled by a centralized organization. The motivation for decentralized applications is that traditional centralized application/structure is very vulnerable to attacks and breed corruption. The concept of the decentralized application was proposed much earlier than blockchain. For example, BitTorrent [23] is a decentralized application, and so does much peer-to-peer software. For rigorous and convenient representation, the “DApp” or “decentralized application” mentioned below all refer to blockchain-based decentralized applications.

Different architectures of DApps are proposed in the following subsection.

A. Native Client as a DApp

Bitcoin can be considered as one of the blockchain-based decentralized applications in payment. Every user runs a client on a peer and then joins the peer-to-peer network. Because the ledger of payment is decentralized, people can use their own client (e.g., Bitcoin Wallet) to transfer Bitcoin to other people. Because the user only uses the client to interact with the network, the client is a DApp. In this survey, this architecture is called Native Client as a DApp.

Fig. 1(a) shows the architecture of Native Client as a DApp. It is used by most of the early Bitcoin-like cryptocurrencies, such as Litecoin [24], PPcoin [25], and so on. The shortcoming of this architecture is that the blockchain is customized for the application (e.g., payment).

B. Smart Contract as a DApp

In the “Native Client as a DApp”, modifying the blockchain for new applications is hard. And the developers of every new DApp need to develop a new blockchain and client, which reduces the efficiency. This can be solved by smart contracts. Developers can use smart contracts on the blockchain (e.g., Ethereum) to record any information they want. Thus the DApp developers can choose to write a smart contract as a DApp for the users. Taking Ethereum as an example, if the developers want to develop a DApp for transferring tokens, they can write a token contract within 100 lines of code on Ethereum. Then the users can use a smart contract browser (e.g., Remix, Mist, etc.) to load a contract on the Ethereum network and call the functions written by the developers. In some cases, the client is also the contract browser. As shown in Fig. 1(b), this architecture is called Smart Contract as a DApp.

However, the bottleneck of this architecture is that it requires the users to have some basic knowledge of programming. Meanwhile, the contract browsers of many platforms are not so easy for users because few of them have graphic user interfaces.

C. Web & Contract as a DApp

To improve the bottlenecks of Contract as a DApp and make it easier for users to use DApp, most DApp developers create a web front end for the smart contracts. As shown in Fig. 1(c), the front end is provided as web pages, including the graphic user interface written in code of html/css/js. And the web browsers run the JavaScript (or install Metamask [26]) to connect to the blockchain peers. There are also some light clients (e.g., imToken [27]) that set web browsers and wallets together so that it will be easier for users to use DApps. Note that the Web client in this architecture is different with the Native client before. The peers can be remote or local. Finally, the browsers can get the important information (e.g., balance, token) from the blockchain and then present it to the front end.

This Web & Contract as a DApp is widely used by most DApps. The main idea is to store the GUI (Graphic User Interface) on the website and the important information (e.g., balance) on the blockchain. This seems to be more familiar to users. However, it causes another centralized problem: Although a few DApps (e.g., Compound [28], Uniswap [29], ForkDelta [30], etc.) are open-source in both web and contract code, many DApp developers do not open their source code of front end.

D. Fully-decentralized DApp

Taylor Gerring proposes an architecture [31], which can obliterate the notion of separating content from presentation by removing the need to have servers at all. It consists

TABLE 1. Survey on four architectures used by DApp

Architecture	DApp
Native Client as a DApp	Bitcoin [10], Zcash [36], Monero [37]
Smart Contract as a DApp	DanKu [38], EurocupBet [39], The DAO [40]
Web & Contract as a DApp	MakderDAO [41], Uniswap [29], Curve [42], Compound [28], Aave [43], Kyber [44], CryptoPunk [45], OpenSea [46], Augur [47], CryptoKitties [48], ForkDelta [30], ENS [49], EosBet [50], AxieInfinity [51]
Fully-decentralized DApp	TornadoCash [34]

of three modules: Ethereum for decentralized logic, Swarm [32] for decentralized storage, and Whisper [33] for decentralized messaging. In this survey, this architecture is called Fully-decentralized DApp. If this concept could be totally implemented, then the developers and users will use the Fully-decentralized DApp. The major difference between Fully-decentralized DApp and Web & Contract as a DApp lies in the storage of the front-end. The storage of Fully-decentralized DApp does not depend on the centralized service, but on the decentralized file systems, as shown in Fig. 1(d). TornadoCash [34] is restricted by some countries for economic regulation, which will be described in Section V. There are few centralized services provide the storage for it. Hence, TornadoCash has to move its front-end files (html/css/js) into decentralized file systems (e.g., IPFS [35])

In summary, four architectures of DApp are listed in this survey. And Table 1 shows the DApps which are conducted by these architectures.

IV. Type of Applications

In this section, we will first collect and give an overview of 3,118 DApps from the StateOfTheDApps. Then we categorize the blockchain-based decentralized applications and summarize their typical advantages over centralized solutions.

A. Overview

There are many platforms of DApps, such as Ethereum, EOS, and so on. Different from traditional applications, there is no centralized app store like AppStore to distribute applications. However, there are still some guiding websites that record the information of the DApps. DApp markets have already grown. There are several DApp market websites, such as StateOfTheDApps [52], DAppReview [53], DApp.com [54], DAppRadar [55], and so on.

3,118 DApps from the StateOfTheDApps are collected and categorized in this survey. The statistics of DApp in different categories are shown in Fig. 2. Most of the DApps published on StateOfTheDApps are games. And the second is exchanges for cryptocurrency. The following are finance, community, gambling, media, property, governance, storage, energy, health, and insurance. It should be noted that the DApps of Exchange have higher DAU (Daily Active Users) since the exchange of cryptocurrency is really active and hot in the market. Some categories of DApps will be described in detail.

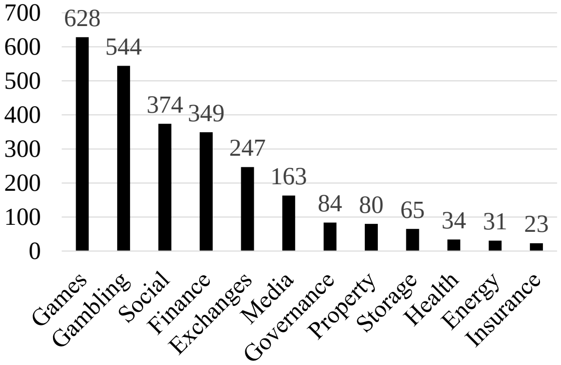


FIGURE 2. Statistics of DApp in Different Categories from StateOfTheDApps (Oct. 2022)

B. Finance (DeFi)

Traditional financial services depend on a trusted party to take some risk and get the benefit (e.g., financial investment, insurance, etc.). But in some way, DApps can remove the third trusted parties. So DApps have wide applications in finance. In this subsection, three typical fields of DApps in finance will be introduced. A general concept of this kind of DApps is Decentralized Finance (so-called DeFi).

Crowd-Funding. Traditional capital markets make it difficult for people to raise money or make investments. The settlement time can be longer than one month due to the financial review. But nowadays, many developers raise the crowd-funding on the smart contract. Then they can get a large number of cryptocurrencies in a very short period of time. It is called Initial Coin Offering (ICO) [9]. Since the ICO DApps record all the financial contributions, it can reward people's financial contributions to a project with actual shares of the project. Tapscott et al. [56] propose a review of ICO. The scale of crowd-funding on DApps is growing fast. However, many frauds appear in the meanwhile.

Token Exchange. In Ethereum and other platforms, after sending cryptocurrencies to a Crowd-Funding contract, the users would be rewarded with tokens, which are proof of their investment. ERC20 protocol on Ethereum enables the token holders to send the tokens to others. Thus many DApps for decentralized exchange (DEX) of tokens show up. IDEX [57] is a decentralized exchange for trading Ethereum tokens, combining the speed of centralization with the security of blockchain settlement. And ForkDelta [30] is also an exchange similar to IDEX. KyberNetwork [44] is an exchange service that enables instant conversion of tokens with guaranteed liquidity. 0x [58] is proposed as a permissionless protocol to trade ERC20 tokens on Ethereum. With the development of DEX, a new type of protocol called Automated Market Maker (AMM) is introduced to DApps. It allows digital assets to be traded without permission and automatically by using liquidity pools instead of a traditional

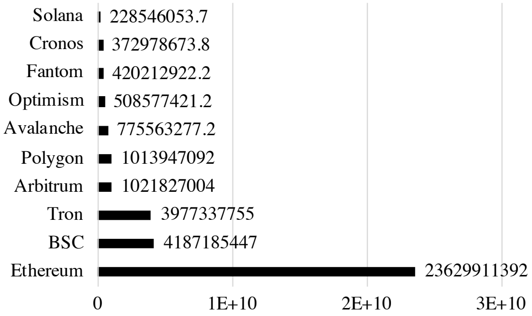


FIGURE 3. **Total Value Locked (USD) in DeFi DApps in Different Blockchains (Top10) from DefiLlama (Jan. 2023)**

market of buyers and sellers [59]. The typical DApps using AMM are Uniswap [29] and Curve [42].

Token Lending. Lending, or the so-called "loan", is one of the most used cases in the traditional financial market. In the blockchain, there are also such needs for borrowing and lending the tokens and paying with the interests. Hence there are some DApps that focus on supporting token lending. Compound [28] is a DApp for supplying or borrowing assets. Accounts on the blockchain supply capital to receive or borrow assets from the protocol. Its smart contracts track these balances and algorithmically set interest rates for borrowers. Aave [43] is similar to Compound but provides more patterns of lending. MakerDAO [41] is designed to lend stable tokens (bound to US dollars) for users and becomes one of the most popular DeFi DApps [60].

Insurance. Mainelli et al. [61] explore the potential for blockchain technology to transform personal insurance. The decentralized applications in the insurance industry can improve efficiency, save costs, and reduce the processing overheads in claims handling [62]. And the lower premiums are paid by the consumers.

As mentioned in Section II, the metric of Total Value Locked (TVL) can be used to evaluate the popularity of the DeFi DApps on different blockchains. We collect the TVL data from DefiLlama [60] in US Dollars and then show the statistics in Fig. 3. As shown in this Fig. 3, Ethereum is now the most popular blockchains for DeFi DApps. And it takes up more than 50% of TVL in all Top10 blockchains. The other blockchains (BinanceSmartChain [16], TRON [18], Fantom [19], Polygon [20], Solana [21], Avalanche [22], etc.) also attracts some DeFi users to lock their cryptocurrencies in their DeFi DApps.

The financial DApps can truly improve efficiency, reduce time costs and make automatic execution. And the challenges are listed as follows: **(1) Tax evasion:** Traditional financial transactions is easy to audit. But the financial DApps are difficult to audit since the users could be anonymous. One user can be divided into many accounts to reduce the tax. Thus the tax could be evaded by the DApps users. **(2) Difficult operation:** Take the insurance DApps, for example, it is not easy to report an accident on the blockchain. It needs a lot of complex and difficult operations to confirm the accident.

C. Game (GameFi)

The game built on blockchain is one of the hottest fields of DApps. As shown in Fig. 2, the game is also the most type of DApps. Moreover, in recent months, GameFi has been a hot topic in DApps. It combines Game and DeFi, and users can sell or buy the game things through DeFi protocols.

Axie Infinity [51] is the most popular GameFi DApps currently. Players can get profits through playing the game and selling the tokens on the decentralized markets. However, it also costs money to start the game. Note that the GUI of Axie is not all based on the Web. Partial GUIs of it are based on a single client on personal computers.

CryptoKitties [48] is a famous game built on the Ethereum blockchain. CryptoKitties are digital, collectible cats built on the Ethereum blockchain. They can be bought and sold using Ether and bred to create new cats with exciting traits and varying levels of cuteness. The key mechanics are tied to actions associated with cryptocurrencies and smart contracts. In summary, Cryptokitties are proof that you can create something on Ethereum, and users can buy, sell, and trade CryptoKitties. The reason for the use of blockchain is that it ensures that each cat is truly unique and persistent.

The main idea of the games is to use blockchain as a data structure to store gameplay and executable elements of the game program. But it also causes some problems: **(1) Throughput:** The throughput of a public blockchain is limited now. And it is reported that Crypto-Kitties have disrupted the Ethereum Network to be too crowded in a few days. [63] **(2) Non-open-source (centralized control):** Some DApp codes are fully controlled and updated only by the developers. **(3) Transaction-Ordering Dependence:** This is a kind of vulnerability that could affect the users to gain profit in the game.

D. Data Storage and Provenance (NFT)

The public blockchain provides permanent storage for the data stored on it, and it is also useful for provenance.

CryptoPunks [45] is a DApp that provides 10,000 uniquely generated characters stored on Ethereum. In this DApp, the characters can be purchased from someone via its marketplace, that's also embedded in the blockchain. The underlying protocol of such buying and selling is the Non-Fungible Token (NFT). In this way, the traces and provenances of the tokens (in the DApp) can be temper-resistant on the blockchain since all the key data is stored on the blockchain. The website of CryptoPunks shows that the current lowest price of a character is more than 300,000 US dollars.

Moreover, since CryptoPunks (and NFTs) are popular with DApp users, many DApps imitating it has been produced. OpenSea [46] provides another independent marketplace for NFTs, rare digital items and crypto collectibles. Users can buy, sell, auction, and discover the NFTs from other DApps, such as the mentioned CryptoKitties, CryptoPunks, and so on.

Note that not all the DApps in this subsection are NFTs. NFT DApps is only a subset. There are also other DApps that leverage blockchain for data storage and provenance. EtherShare [64] is a DApp for users to share information with permanent storage and open access. EthereumNameService [49] takes a decentralized domain name as an NFT, and then resolve it to a specific address, in order to ease the usage of the address. Liang et al. [65] propose a ProvChain for cloud data provenance, including three phases: provenance data collection, data storage, and data validation.

However, some challenges are listed as follows: **(1) Waste of storage:** The DApps usually store the data on each blockchain peer. It is necessary, but it also causes a waste of storage, especially some kind of big data. **(2) Identity authentication:** A DApp user is represented as "address" on the blockchain. Thus it is difficult to link the address to the real-world identity in the decentralized situation. **(3) Piracy problem:** Some DApps only provide the solution to store the data so that the data is easy to be copied. The piracy problem is urgent for the DApps of data storage.

E. Privacy Protection

DApps can be considered native anonymous because blockchain technology is natively anonymous. So in some way, DApp can protect the privacy of users. Zyskind et al. [66] propose a DApp as a decentralized personal data management system, ensuring users take full control of their private data. And Linn. et al. [67] also describe a blockchain-based access-control manager in the health IT ecosystem, named Health Care Blockchain. Zyskind et al. [68] propose Engima, a decentralized computation platform to enable users to share their data with cryptographic guarantees regarding their privacy.

However, in some public blockchain systems, all the transactions are visible and exposed all over the world. Zerocash, proposed by Ben-Sasson et al. [69], is conducted with strong privacy guarantees from Bitcoin, with the advances in zero-knowledge Succinct Non-interactive Arguments of Knowledge. In another cryptocurrency called Monero, a confidential ring method is proposed by Noether et al. [70] to hide transaction amounts, which enhances the privacy of Monero. Another obfuscation improvement of Monero is proposed by Mackenzie et al. [71] to provide long-term resistance of the cryptocurrency against blockchain analysis.

As for this kind of DApps, there are also some disadvantages and challenges: **(1) Violation of law:** For example, Monero enables people to transfer money against the censorship of the government. In some cases, it is so-called "freedom" and "privacy". However, this will also help the criminals to receive money. It is hard to make a balance between law and privacy in DApps. **(2) Computing resources consumption:** The algorithms to generate the privacy-protected transaction always consume a lot of computing resources. For example, Zerocash takes the user a few minutes to generate a transaction. Thus a faster algorithm is needed.

F. Sharing

One key advantage of DApp is to enable peer-to-peer sharing without a trusted third party. Users can use DApps to share the things they want for free or for a fee. Xu et al. [72] propose Prc, a blockchain-based sharing economy platform to maintain desirable features that public blockchain offers to share economy applications without sacrificing user's privacy. Bogner et al. [73] demonstrate a DApp for sharing everyday objects based on the smart contract on Ethereum. Kang et al. [74] design a localized P2P electricity trading system with consortium blockchain to illustrate detailed operations of localized P2P electricity trading. Luu et al. [75] implement and deploy SMARTPOOL, a DApp for the decentralized mining pool, enabling the Ethereum miners to contribute their hash rate and share the rewards.

In the field of cloud computing, DApps can be used to share the computing resources of users. IExec [76] relies on Ethereum smart contracts and allows the building of a virtual cloud infrastructure that provides high-performance computing services on demand. Similar to IExec, Golem [77], and SONM [78] are also the DApps to share computing resources. The differences are: Golem assembles a network to attract regular 3D rendering users first, and SONM aims at fog and edge computing. In electric vehicles cloud and edge computing, Liu et al. [79] propose blockchain-inspired data coins and energy coins, in which data contribution frequency and energy contribution amount are applied to achieve the proof of work.

There are some problems with the sharing DApps. **(1) Insufficient supervision:** Decentralized sharing means that anyone can share in the P2P network. However, once controversy about sharing shows up, supervision is missing. A way of supervision or arbitration of sharing is needed. **(2) Low throughput:** Similar to the IoT DApps, the sharing DApps need high throughput to ensure the user experience. Thus this kind of DApps also suffers from the low throughput of blockchain.

G. Gambling and Prediction Market

Although there are some differences between gambling and prediction market [91], this survey puts these two types together since the action of the DApp users is almost the same: Bet on a prediction with some money, then get the rewards if it is true. Traditional gambling and prediction market cost users some fees for trusted third parties (e.g., casinos), and it is easy to be unfair to users. Nowadays, there are lots of DApps for gambling or prediction markets. For example, Etheroll [87] is a DApp for placing bets on our provably with no deposits or sign-ups. Each dice roll is provably random and cryptographically secure. Miller et al. [88] present a zero-collateral lottery protocol in Bitcoin and Ethereum. Cryptocup [89] is a DApp as a World Cup

TABLE 2. Comparison of Different Types of DApps

Types of DApps	Advantages	Challenges
Finance (DeFi) [9], [44], [57], [61], [62], [80]	(1)Improving efficiency (2)Reducing the time costs (3)Automatic execution	(1)Tax evasion (2)Difficult operations
Game (GameFi) [48], [81], [82]	(1)Permanent game assets (2)Improving asset mobility (3)Ensuring role uniqueness	(1)Vulnerabilities of randomness (2)Non open-source (2)Transaction-Ordering Dependence
Data Storage and Provence (NFT) [64], [65], [83], [84]	(1)Permanent storage (2)Preserving privacy (3)Improving reliability	(1)Waste of storage (2)Identity authentication (3)Piracy problem
Privacy Protection [66], [67], [69]–[71], [85], [86]	(1)Preserving privacy (2)Improving user ownership of data	(1)Violation of law (2)Computing resources consumption
Sharing [72], [73], [75]–[79]	(1)Improving efficiency (2)Removing trusted third party (3)Promoting resource sharing	(1)Insufficient supervision (2)Low throughput
Gambling and Prediction Market [47], [87]–[90]	(1)Removing trusted third party (2)Less fees (3)Automatic execution	(1)Centralized oracle (2)Not absolute truth (3)Higher delay

prediction game with ERC 721 tokens. Users will predict the World Cup matches to gain potential rewards. As for the prediction market, Peterson et al. [47] propose a decentralized oracle and prediction market platform called Augur.

A key problem of the gambling and prediction market is how to input the real-world result (e.g., champion of the World Cup) into the smart contracts. Adler et al. [90] propose ASTRAEA, a decentralized oracle based on a voting game, to solve the problem. Oraclize [92] and Reality Keys [93] are also the oracle solutions.

There are also some challenges of this kind of DApps: **(1) Centralized oracle:** Although Oraclize and ASTRAEA try to help input the real-world data to the smart contract, they are still not decentralized. A new oracle solution is an opportunity. **(2) Not absolute truth:** In the prediction market, the prediction result could be affected by the users. For example. The champion of the World Cup is input by the users. If most users choose to lie, the result could be fake. **(3) Higher delay:** Traditional gambling and prediction markets can ensure very low delay. But DApps require a higher delay on committing the block or voting for the result.

In summary, DApps show great advantages in many fields of applications. Table 2 shows the summarized advantages and challenges of different types of DApps.

V. Problems of DApps

In this section, we will discuss the problems of DApps. We will summarize the problems of DApps into three fields: economic, security, and performance.

A. Economic Policy and Risk

In this survey, the economic problems of DApps fall into three folds: Incentive Policy, Risk Evaluation, and Miner Effects, as shown in Table 3.

TABLE 3. Economic Problems in DApps

Economic Problem	Related Studies
Incentive Policy	Design [29], [46]
	Measurement [94]
Risk Evaluation	Scam [95], [96]
	Measurement [94]
	Management [97]
Miner Effect	Definition [98]
	Quantifying [99], [100]
	Explorer [101]
Economic Regulation	Censorship [102]

Incentive Policy. In the above-mentioned DApps, including the DeFi, GameFi, and NFTs, there is a problem: How to attract users to use the DApp? The answer results in incentive policy economics. In other words, in most DApps, the common way to motivate users is to make users make money, which is an economic problem. For example, Uniswap [29] rewards the users with fees and governance tokens as incentives. OpenSea [46] returns the customized fees to the creators of NFTs as incentives. Qin et al. [94] propose an empirical study on the measurement of the incentive of the lending DApps by processing the on-chain blockchain data. Research on incentive policies can be an opportunity.

Risk Evaluation. As for a user, when using a DApp, the risk comes from many areas. Researchers have found that the DApp itself can be a scam [95], [96]. Moreover, the risk also comes from market volatility. Qin et al. [94] measure various risks that liquidation participants are exposed to and quantify the instabilities of existing lending DApps. With more DApps being developed, measuring and managing the risk [97] for users can be helpful and challenging.

Miner Effects. Blockchain miners have large effects on DApps. Miner extractable value (MEV) is a measure of the profit a miner (or validator, sequencer, etc.) can make through their ability to arbitrarily include, exclude, or re-

TABLE 4. Vulnerabilities and Attacks in DApps

Layer	Vulnerability	DApps
Web	Front-end Tampering	BadgerDAO [104]
	Inconsistent Synchronization	Augur [105]
	DNS Server Hijacking	Myetherwallet [106]
	Centralized Down	Infura [107], [108]
Blockchain	51% Vulnerability	Bitcoin Gold [109]
	Balance Attack	R3 [110]
	Double Spending	Bitcoin [111]
	Transaction-Ordering Dependence	Rock-ps [112]
	Timestamp Dependence	TheRun [113]
Contract	Mishandled Exception	KoET [114]
	Reentrancy Vulnerability	TheDAO [40]
	Immutable Bugs	Rubixi [115]
	Blockhash	EtherPot [116]

order transactions within the blocks they produce [98]. And lots of studies are proposed for MEV. FlashBots [99] provides a study on the front-running transactions on the DEX DApps. They also provide a tool for the Ethereum miners in the latter version, which has been applied to many miners to gain extra profits. Qin et al. [100] quantify the MEV in another perspective called blockchain extractable value. The MEV explorer website [101] shows that there are more than 24 million US dollars were extracted by miners in Nov. 2021. Hence, investigating the miner effects of DApps can be a research opportunity.

Economic Regulation. The decentralization of the DApps makes it hard for economic regulation. For example, TornadoCash [34] is a project designed for mixing cryptocurrencies in a decentralized way. This project enhances the privacy for the users but also the illegal assets. Therefore, some countries including the United States have restricted this project. For example, U.S. persons are prohibited from engaging in transactions involving TornadoCash, including through the virtual currency wallet addresses that the government has identified [103]. As mentioned before, although TornadoCash has been migrated to the decentralized file systems, its transactions are reported to be rejected [102]. More solutions for the balance between privacy and regulation could be the research opportunities.

B. Security Risk

Since most DApps are conducted with cryptocurrencies, security is very important. Once the DApps were attacked, billions of cryptocurrencies could be stolen, with no way to get the money back because of the features of blockchain. In this section, typical vulnerabilities and attacks will be presented, with the security solutions.

Vulnerabilities and Attacks. §III shows different architectures of DApps. Web & Contract as a DApp is the most widely used architecture until now. This architecture can be abstracted into three layers: web, smart contract, and blockchain. Then these three layers can be attacked by different vulnerabilities, as shown in Table 4. Table 4 shows the vulnerabilities and attacks in real-world cases. The centralization in DApps resulted from the centralization of the Web GUI. As shown in Table 4, the Web layer is one of the most important vulnerable layers. In Dec. 2021, the front-end of BadgerDAO was tempered by hackers [104]. In this attack, various tokens worth about 120 million US dollars are stolen. Augur [47] and other DApps are reported with inconsistent synchronization bug [105]. MyEtherWallet is a famous DApp widely used as a wallet for transferring tokens. It is reported [106] to be attacked, and over $152,000 is stolen by the hackers via DNS hijacking. In November 2020, Infura.io was reported to be down for hours [107]. At that time, several DApp browsers were reported for exceptions of users' balances and DApp operations, which might cause wrong operations of users [108]. As for the DApp vulnerabilities resulting from blockchain and smart contracts, details can be found in previous surveys [117].

Recent Advances. As there are many vulnerabilities and attacks in DApps, the tools and solutions for DApps are urgently needed. And most tools are based on solving the vulnerabilities of blockchain and smart contracts. Formal verification works as one of the solutions. OYENTE [114] is built as a symbolic execution tool to find potential security bugs. The tool can check the bytecode of the contracts and then help the developers to avoid vulnerabilities. Bhargavan1 et al. [118] propose a framework for runtime safety and the functional correctness of smart contracts, translating the contracts to a functional programming language named F*. KEVM [119] is proposed as a complete executable semantics of the running environment of smart contracts. Another semantic framework is presented in asemantic as a complete small-step semantics of bytecode of smart contracts. DappGuard [120] is developed as a tool to classify known attacks from transaction data, protect the DApps from attacks and determine malicious actors to learn new attacks. DArcher [105] is a tool to detect on-chain-off-Chain synchronization bugs for DApps. Pettersson et al. [121] implement a proof-of-concept compiler for smart contracts to reduce the risk of errors and the need for testing. CertiK [122] is a formal verification framework to help mathematically prove whether a DApp is hacker-resistant. Another way to maintain security is to generate smart contracts automatically. FSolidM [123], [124] is a framework rooted in rigorous semantics for designing contracts as finite state machines, with a tool to create the contracts on a graphical interface. Frantz et al. [125] propose a modeling approach to support the automatic translation from human-readable contract representations to executable smart contracts. Wohrer et al. [126] also find design patterns for smart contracts are found in detail and provide the code for better illustration. Modifying the mechanism of blockchain is also the solution. Karame et al. [111] propose a modification to the existing Bitcoin implementation to ensure the detection of double spending attacks. Chen et al. [127] propose an adaptive gas cost mechanism to defend against known and unknown DoS attacks with flexible parameter

TABLE 5. Comparison of Metrics Used in DApp Performance Research

Layer	Studies	Throughput	Latency	Hardware Consumption	Contract Execution	Fault Tolerance	Read Availability	Consensus Cost
Blockchain	Zheng et.al [130]	√	√	√	√		√	√
	Weber et al. [131]		√					
	Kalodner et al. [132]	√						
	Dinh et al. [133]	√	√	√		√		
	Gupta et al. [134]	√	√	√		√		√
	Gervais et al. [135]	√						
	Li et al. [136]						√	
Decentralized Storage	Abdullah et.al [137]		√	√			√	
	Ismail et.al [138]		√					
	Shen et.al [139]	√	√					
	Trautwein et.al [140]		√					

settings in Ethereum. Marino1 et al. [128] set the standards to alter and undo the smart contracts so that users can avoid losing money in the unsafe contracts. And the developers can also try to use safe smart contract programming languages, such as Pact and Liquidity, in which fewer vulnerabilities are found [129].

Research Opportunities. The security tools and solutions are great opportunities in both academia and industry. The research opportunities are summarized as follows: **(1)Reliable Web for DApps** Developing a reliable web layer of DApps is needed. There are two optional ways. One is to develop decentralized file systems. The other is to develop the tools that defend the centralized web page from attacks, such as Darcher [105]. **(2)Formal Verification:** Although there is already some research on the formal verification of smart contracts, the code of smart contracts is developing fast. Thus the formal verification of the contract code is still a good topic for research. **(3)Standard Templates:** For new DApp developers, it is difficult to write code that ensures no bugs or vulnerabilities. A possible solution is to provide standard DApp templates for the developers. This could be similar to the research of FSolidM [123], [124]. But this field is still quite blank. **(4)More Vulnerabilities and Tools:** DApps are still at a very early time. More platforms and types of DApps are in the working process. Thus more and more vulnerabilities and corresponding tools should be found to avoid economic loss. **(5)Similarity Detection:** For the new DApp developer, it is not easy for them to use formal verification tools and other vulnerability detection tools. However, it might be a good idea to conduct a similarity detection of the DApps. In this way, the developers can find out whether there are some similar vulnerabilities in their DApps.

C. Low Performance

This subsection discusses the challenges of DApps in performance and also presents the recent advances with a comparison.

Challenges. DApps have not yet been used as widely as PC and mobile apps because DApps do not meet daily use as easily as mobile apps. One of the most urgent problems is performance. There are so many DApps that suffer from the low throughput of blockchain systems. It is reported that Ethereum has been disrupted by DApps. Resulted from this, many DApps can not work well since the transactions can not totally be confirmed. And there are thousands of peers in a blockchain system, so it is necessary to know what is going on in the system. In this way, the people who run the peers can do some analysis or fix errors if the blockchain system becomes abnormal. But the peers belong to different parties. Thus here comes the challenges of how to monitor the whole status, including the blockchain transactions and overall performance. For example, EOS [17] is declared to achieve an extremely high throughput of hundreds of thousands. But Bitmex Research shows that the real-world throughput on EOS is not much better than the one on Ethereum [141]. In traditional distributed systems, there are some black-box studies such as Project5 [142], WAP5 [143] and the Sherlock system [144]. .Therefore the universal or standard benchmarks of different blockchain systems are needed. Finally, some studies have shown that the availability of DApps can not meet the requirement of daily applications. Then how optimizing the performance of DApps becomes a challenge.

Recent Advances. The recent advances of blockchain-based DApps are summarized into the layer of blockchain and decentralized storage. **(1)** As for the blockchain layer, Zheng et al. [130] propose a scalable framework for detailed and real-time monitoring of blockchain systems, which has much lower overhead and more details about the blockchain systems compared with previous approaches. One of the main ideas is to divide the metrics into overall metrics for users and detailed metrics for developers. Weber et al. [131] a method to identify the availability limitations of Bitcoin and Ethereum, showing that the read availability is high while the write availability is low. Kalodner et al. [132] propose an open-source software platform for blockchain systems, which parses the data from the p2p nodes and raw blockchain data for users to monitor and analyze the system. Zheng et al. [130] collect and classify 1,000 open-source smart contracts and do the performance evaluation on four well-known blockchain systems. Dinh et al. [133] describe frameworks for analyzing private blockchains in varying workloads. Gupta et al. [134] also propose a method for analyzing performance. Gervais et al. [135] present a novel quantitative framework for the security and performance of PoW blockchains. In some cases, DApps show low performance thanks to the limited query support of blockchain systems. Thus Li et al. [136] propose EtherQL as a query layer for Ethereum. It also provides two levels of interfaces for data retrieving or serving as a RESTful data provider. **(2)** As for the layer of decentralized storage, Abdullah et.al [137] record and analyze the performance metrics of IPFS [145] and FTP. Ismail et.al [138] evaluate the costs and latency of the existing decentralized file systems. Shen et al. [139] use Amazon EC2 servers to evaluate the performance of data I/O operations from the perspective of IPFS client. Trautwein et.al [140] evaluate the performance of IPFS and uncover the characteristics of the IPFS peers.

Research Opportunities. Table 5 shows the comparison of metrics used in the performance of DApps. Thus throughput and latency are the two metrics on which many researchers focus. The main reason is that there is a big gap between the throughput and latency of DApps now and the requirement in real-world applications. And some other metrics, such as hardware consumption and fault tolerance, should also be evaluated in different blockchain platforms of DApps. There are many peers running the blockchain clients nowadays. Thus, these metrics are also the key metrics for the evaluation of DApp platforms. The rest metrics in the table, such as contract execution and consensus cost, seem to receive less attention. However, these metrics also reflect the bottlenecks of blockchain systems. If the throughput is high enough and the hardware consumption is low enough, these metrics will become the key to the next optimization of the DApps. Some research opportunities are listed as follows: **(1)Performance of Contract Execution:** Smart contract is one of the most important parts of DApp. There are already lots of research on the performance of the underlying blockchain system. However, the research on the performance of smart contracts is blank. **(2)Standard Benchmark:** Although there are some papers that focus on the benchmark of blockchain systems. However, as for DApp, there is no benchmark. A standard set of workflows or operations for the benchmark of DApp is necessary. **(3)Automated Testing:** There are many automated testing tools for traditional computer applications or mobile applications. That will help developers to know the reliability of the application. The testing tools for DApp are missing. Thus it is also an opportunity.

VI. Conclusion

This survey provides a comprehensive overview of the research on blockchain-based decentralized applications. The definition and typical architectures of DApps are summarized with their strengths and weaknesses. Moreover, we collect and categorize the DApps into different types with the details, of which the advantages over centralized solutions are presented. As for recent research aspects from economics, security, and performance in DApp, this paper also provides an overview and the research opportunities in these aspects.

REFERENCES

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

[2] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. An overview of blockchain technology: Architecture, consensus, and future trends. In *IEEE ICBD*, pages 557–564, 2017.

[3] Nick Szabo. The idea of smart contracts. 1997.

[4] Z. Zheng, S. Xie, H. Dai, et al. An overview of blockchain technology: Architecture, consensus, and future trends. *IEEE International Congress on BigData*, pages 557–564, 2017.

[5] Vitalik Buterin and DACs DAOs. *DAs and More: An Incomplete Terminology Guide*. ETHEREUM BLOG, https://blog.ethereum.org/2014/05/06/daos-dacs-das-and-more-an-incomplete-terminology-guide/, 2014.

[6] Z. Zheng, S. Xie, H. N. Dai, et al. *Blockchain challenges and opportunities: A survey*. International Journal of Web and Grid Services, 2016.

[7] X. Li, P. Jiang, T. Chen, et al. *A survey on the security of blockchain systems*. Future Generation Computer Systems, 2017.

[8] DAppRadar. *2022 August DApp Industry Report*. https://dappradar.com/blog/dappradar-blockchain-industry-report-august-2022, 2022.

[9] Wikipedia. *Initial Coin Offering*. https://en.wikipedia.org/wiki/Initial_coin_offering.

[10] A Nakamoto S. Bitcoin:. peer-to-peer electronic cash system[J], 2008.

[11] D. Mingxiao, M. Xiaofeng, Z. Zhe, et al. A review on consensus algorithm of blockchain. *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2567–2572, 2017.

[12] G. Foroglou and A. L. Tsilidou. Further applications of the blockchain. In *12th Student Conference on Managerial Science and Technology*, 2015.

[13] Szabo N. Smart contracts. building blocks for digital markets[j]. *EXTROPY: The Journal of Transhumanist Thought*, 16, 1996.

[14] C. Sillaber and B. Waltl. Life cycle of smart contracts in blockchain ecosystems. 41(8):497–500, 2017.

[15] Vitalik Buterin. *Ethereum white paper*. https://ethereum.org/whitepaper/, 2013.

[16] BinanceSmartChain. *Binance Smart Chain*. https://github.com/binance-chain/bsc.

[17] EOSIO. *Technical White Paper v2*. https://github.com/EOSIO/Documentation/blob/master/Technical.

[18] TRON. *TRON Website*. https://tron.network/.

[19] Fantom. *Fantom Website*. https://fantom.foundation/.

[20] Polygon. *Polygon Website*. https://polygon.technology/.

[21] Solana. *Solana Website*. https://Solana.com/.

[22] AVAX. *Avalanche Website*. https://www.avax.network/.

[23] B. Cohen. *The BitTorrent protocol specification*. 2008.

[24] Lee C. *Litecoin White Paper*, 2011.

[25] S. King and Nadal S. Ppcoin. Peer-to-peer crypto-currency with proof-of-stake. 19, 2012.

[26] MetaMask. *MetaMask Browser Extension*. https://github.com/MetaMask/metamask-extension.

[27] ImToken. *ImToken Medium*. https://medium.com/imtoken/imtoken-1-2-0-fully-supports-the-offline-signing-of-transactions-d385899d3350.

[28] Robert Leshner and Geoffrey Hayes. Compound: The money market protocol. *White Paper*, 2019.

[29] Guillermo Angeris, Hsien-Tang Kao, Rei Chiang, Charlie Noyes, and Tarun Chitra. An analysis of uniswap markets. *arXiv preprint arXiv:1911.03380*, 2019.

[30] ForkDelta. *ForkDelta*. https://forkdelta.github.io/about/.

[31] Taylor Gerring. Building the decentralized web3. 2014.

[32] Swarm Team. *ETH Swarm*. https://www.ethswarm.org/.

[33] Whisper Team. *Whisper*. https://github.com/ethereum/whisper.

[34] Alexey Pertsev, Roman Semenov, and Roman Storm. Tornado cash privacy solution version 1.4. 2019.

[35] Juan Benet. Ipfs-content addressed, versioned, p2p file system. *arXiv preprint arXiv:1407.3561*, 2014.

[36] Greenberg A. Zcash. *an untraceable bitcoin alternative*. 2016.

[37] Courtois N. T. *Stealth Address and Ring Signatures*, 2016. Monero.

[38] A. B. Kurtulmus and K. Daniel. *Trustless Machine Learning Contracts; Evaluating and Exchanging Machine Learning Models on the Ethereum Blockchain[J]. arXiv*, 2018. preprint.

[39] EurocupBet. *Eurocup Bet*. https://forum.ethereum.org/discussion /7425/dapp-lets-bet-on-the-euro-cup-2016-winner.

[40] Wikipedia. *The Dao*. https://en.wikipedia.org/wiki /The_DAO_(organization).

[41] GIANLUCA BOGONI. Cryptocurrencies stabilization systems: a focus on the makerdao case. 2019.

[42] CurveFinance. *Curve Website*. https://curve.fi/.

[43] AAVE. *AAVE Website*. https://aave.com.

[44] A Luu Y. V. L. Kybernetwork:. *trustless decentralized exchange and payment service*.

[45] CryptoPunks. *CryptoPunks Website*. https://www.larvalabs.com/cryptopunks.

[46] OpenSea. *OpenSea Website*. https://opensea.io/.

[47] J. Peterson, J. Krug, M. Zoltu, et al. *Augur: a Decentralized Oracle and Prediction Market Platform*. 2018.

[48] CryptoKitties. *Collectible and Breedable Cats Empowered by Blockchain Technology*. http://cryptokitties.co/.

[49] Nick Johnson. Ens documentation release 0. 1.

[50] EosBet. *Whitepaper*. https://github.com/EOSBetCasino.

[51] AxieInfinity. *AxieInfinity Website*. https://axieinfinity.com/.

[52] StateOfTheDApps. *StateOfTheDApps Website*. https://stateofthedapps.com.

[53] DAppReview. *DAppReview Website*. https://dapp.review.

[54] DApp.com. *DApp.com Website*. https://stateofthedapps.com.

[55] DAppRadar. *DAppRadar Website*. https://dappradar.com.

[56] A. Tapscott and D. Tapscott. How blockchain is changing finance. *Harvard Business Review*, 1, 2017.

[57] AuroraLab. *A Real-Time and High-Throughput Ethereum Smart Contract Exchange*. https://idex.market/static/IDEX-Whitepaper-V0.7.5.pdf.

[58] W. Warren and Bandeali A. 0x. *An open protocol for decentralized exchange on the Ethereum blockchain*. 2017.

[59] Cryptopedia. *What Are Automated Market Makers?* https://www.gemini.com/cryptopedia/amm-what-are-automated-market-makers#section-liquidity-pools-and-liquidity-providers.

[60] Defillama. *MakerDAO on Defillama*. https://defillama.com/protocol/makerdao.

[61] M. Mainelli and B. Manson. *How blockchain technology might transform wholesale insurance*. https://www.pwc.lu/en/fintech/docs/pwc-how-blockchain-technology-might-transform-insurance.pdf.

[62] B. Cant, A. Khadikar, A. Ruiter, et al. *Smart Contracts in Financial Services: Getting from Hype to Reality*. Capgemini Consulting, 2016.

[63] Open Trading Networ. *How Crypto-Kitties Disrupted the Ethereum Network*. https://hackernoon.com/how-crypto-kitties-disrupted-the-ethereum-network-
845c22aa1e6e.

[64] EtherShare. *Share Information with Permanent Storage and Open Access*. http://etherShare.org.

[65] X. Liang, S. Shetty, D. Tosh, et al. Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pages 468–477, 2017.

[66] G. Zyskind and Nathan O. Using blockchain to protect personal data. *2015 IEEE Security and Privacy Workshops (SPW)*, pages 180–184, 2015.

[67] L. A. Linn and M. B. Koo. *Blockchain for health data and its potential use in health it and health care related research*. ONC/NIST, Gaithersburg, Maryland, United States, onc/nist use of blockchain for healthcare and research workshop edition, 2016.

[68] G. Zyskind, O. Nathan, and Pentland A. Enigma. *Decentralized computation platform with guaranteed privacy*, 2015. preprint.

[69] E. B. Sasson, A. Chiesa, C. Garman, et al. Zerocash: Decentralized anonymous payments from bitcoin. *2014 IEEE Symposium on Security and Privacy (SP)*, pages 459–474, 2014.

[70] S. Noether and Mackenzie A. Ring confidential transactions. 1:1–18, 2016.

[71] A. Mackenzie, S. Noether, and M. C. Team. *Improving Obfuscation in the CryptoNote Protocol*. 2015.

[72] L. Xu, N. Shah, L. Chen, et al. Enabling the sharing economy: Privacy respecting contract based on public blockchain. *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, pages 15–21, 2017.

[73] A. Bogner, M. Chanson, and A. A Meeuw. decentralised sharing app running a smart contract on the ethereum blockchain. *Proceedings of the 6th International Conference on the Internet of Things*, pages 177–178, 2016.

[74] J. Kang, R. Yu, X. Huang, et al. Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains. *IEEE Transactions on Industrial Informatics*, 13(6):3154–3164, 2017.

[75] L. Luu, Y. Velner, J. Teutsch, et al. Smart pool: Practical decentralized pooled mining. *IACR Cryptology ePrint Archive*, 2017:19, 2017.

[76] iExec. *iExec White Paper*. https://iex.ec/whitepaper/iExec-WPv3.0-English.pdf.

[77] Golem. *The Golem Project Crowdfunding Whitepaper*. https://golem.network/crowdfunding/Golemwhitepaper.pdf.

[78] SONM. *About SONM*. https://docs.sonm.com/home.

[79] H. Liu, Y. Zhang, and T. Yang. Blockchain-enabled security in electric vehicles cloud and edge computing. *IEEE Network*, 32(3):78–83, 2018.

[80] De Filippi P. *Blockchain-based Crowdfunding*. 2015.

[81] PetChain. *PetChain*. https://pet-chain.duxiaoman.com/.

[82] HyperDragons. *HyperDragons-WhitePaper*. https://storage.googleapis.com/hyperdragons-imgs/file/HyperDragons-WhitePaper(ch).pdf.

[83] A. Ramachandran and Kantarcioglu D. *Using Blockchain and smart contracts for secure data provenance management*, 2017. preprint.

[84] Etherscan. *SaveData*. https://etherscan.io/address /0xf34cd2fd11233df8f90646ab658b03bfea98aa92#code.

[85] D. Ron and A. Shamir. *Quantitative analysis of the full bitcoin transaction graph*. Springer, Berlin, Heidelberg 6-24, international conference on financial cryptography and data security edition, 2013.

[86] A. Kosba, A. Miller, E. Shi, et al. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. *2016 IEEE Symposium on Security and Privacy (SP)*, pages 839–858, 2016.

[87] EthRoll. *EthRoll Website*. https://etheroll.com/.

[88] A. Miller and I. Bentov. Zero-collateral lotteries in bitcoin and ethereum. *2017 IEEE European Symposium on Security and Privacy Workshops*, pages 4–13, 2017.

[89] CryptocupWhitepaper. *The first World Cup betting powered by Blockchan*. https://www.cryptocup.io/assets/pdf/Cryptocup/whitepaper.pdf.

[90] J. Adler, R. Berryhill, A. Veneris, et al. Astraea: A decentralized blockchain oracle.

[91] J. Wolfers and Zitzewitz E. Prediction markets. *Journal of economic perspectives*, 18(2):107–126, 2004.

[92] Oraclize. *Oraclize Website*. http://www.oraclize.it.

[93] Reality Keys. *Facts about the future cryptographic proof when they come true*. https://www.realitykeys.com/.

[94] Kaihua Qin, Liyi Zhou, Pablo Gamito, Philipp Jovanovic, and Arthur Gervais. An empirical study of defi liquidations: Incentives, risks, and instabilities. In *Proceedings of the 21st ACM Internet Measurement Conference*, pages 336–350, 2021.

[95] W. Chen, Z. Zheng, J. Cui, et al. Detecting ponzi schemes on ethereum: Towards healthier blockchain technology. *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 1409–1418, 2018.

[96] M. Bartoletti, S. Carta, T. Cimoli, et al. *Dissecting Ponzi schemes on Ethereum: identification, analysis, and impact*, 2017. preprint.

[97] Johannes Rude Jensen and Omri Ross. Managing risk in defi. In *JR Jensen and O. Ross,"Managing Risk in DeFi" in CEUR Workshop Proceedings, Aachen*, 2020.

[98] CoinMarketCap. *Miner Extractable Value (MEV)*. https://coinmarketcap.com/alexandria/glossary/miner-extractable-value-mev.

[99] Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and

consensus instability. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 910–927, 2020.

[100] Kaihua Qin, Liyi Zhou, and Arthur Gervais. Quantifying blockchain extractable value: How dark is the forest? In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 198–214. IEEE, 2022.

[101] FlashBots. *MEV Explorer*. https://explore.flashbots.net/.

[102] Labrys. *MEV Watch*. https://www.mevwatch.info/.

[103] U.S. DEPARTMENT OF THE TREASURY. *Frequently Asked Questions on Tornado Cash*. https://home.treasury.gov/policy-issues/financial-sanctions/faqs/added/2022-09-13.

[104] Richard Lawler. *Someone stole $120 million in crypto by hacking a DeFi website*. https://www.theverge.com/2021/12/2/22814849/badgerdao-defi-120-million-hack-bitcoin-ethereum.

[105] Wuqi Zhang, Lili Wei, Shuqing Li, Yepang Liu, and Shing-Chi Cheung. Darcher: Detecting on-chain-off-chain synchronization bugs in decentralized applications. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 553–565, 2021.

[106] David Floyd. $150k stolen from myetherwallet users in dns server hijacking.

[107] Yogita Khatri. Ethereum infrastructure provider infura is down, crypto exchanges begin to disable eth withdrawals, 2020.

[108] Sctt Chipolina. Crucial ethereum service infura suffers major outage, 2020.

[109] Jeff John ROBERTS. *Bitcoin Gold Suffers Rare*, 51.

[110] C. Natoli and V. Gramoli. *The balance attack against proof-of-work blockchains: The R3 testbed as an example*, 2016. preprint.

[111] G. O. Karame, E. Androulaki, and S. Capkun. Double-spending fast payments in bitcoin. *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 906–917, 2012.

[112] Etherscan. *RockPaperScissors*. https://etherscan.io/address /0x1d77340D3819007BbfD7fdD37C22BD3b5c311350#code.

[113] Etherscan. *TheRun*. https://etherscan.io/address /0xcac337492149bdb66b088bf5914bedfbf78ccc18#code.

[114] L. Luu, D. H. Chu, H. Olickel, et al. Making smart contracts smarter. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 254–269, 2016.

[115] Etherscan. *Rubixi*. https://etherscan.io/address /0xf34cd2fd11233df8f90646ab658b03bfea98aa92#code.

[116] EtherPot. *EtherPot*. http://etherpot.github.io/.

[117] Tharaka Hewa, Mika Ylianttila, and Madhusanka Liyanage. Survey on blockchain based smart contracts: Applications, opportunities and challenges. *Journal of Network and Computer Applications*, 177:102857, 2021.

[118] K. Bhargavan, A. Delignat-Lavaud, C. Fournet, et al. Formal verification of smart contracts: Short paper. *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security*, pages 91–96, 2016.

[119] E. Hildenbrandt, M. Saxena, X. Zhu, et al. *A Complete Semantics of the Ethereum Virtual Machine*. KEVM, 2017.

[120] T. Cook, A. Latham, and Lee J. H. *DappGuard: Active Monitoring and Defense for Solidity Smart Contracts*.

[121] J. Pettersson and R. Edstrom. Safer smart contracts through type-driven development. Master's thesis, 2016.

[122] CertiK. *Towards Building Fully Trustworthy Smart Contracts and Blockchain Ecosystem*. https://certik.org/whitepaper.html.

[123] A. Mavridou and A. Laszka. *Designing Secure Ethereum Smart Contracts: A Finite State Machine Based Approach[J]. arXiv*, 2017. preprint.

[124] A. Mavridou and Laszka A. Tool demonstration: FSolidM. *for designing secure Ethereum smart contracts*. Springer, Cham 270-277, international conference on principles of security and trust edition, 2018.

[125] C. K. Frantz and M. Nowostawski. From institutions to code: Towards automated generation of smart contracts. *IEEE International Workshops on Foundations and Applications of Self* Systems*, pages 210–215, 2016.

[126] M. Wohrer and U. Zdun. *Design Patterns for Smart Contracts in the Ethereum Ecosystem*. 2018.

[127] T. Chen, X. Li, Y. Wang, et al. *An Adaptive Gas Cost Mechanism for Ethereum to Defend Against Under-Priced DoS Attacks*. Springer, Cham 3-24, international conference on information security practice and experience edition, 2017.

[128] B. Marino and A. Juels. *Setting standards for altering and undoing smart contracts*. Springer, Cham 151-166, international symposium on rules and rule markup languages for the semantic web edition, 2016.

[129] R. M. Parizi and A. Dehghantanha. *Smart Contract Programming Languages on Blockchains: An Empirical Evaluation of Usability and Security*. Springer, Cham 75-91, international conference on blockchain edition, 2018.

[130] P. Zheng, Z. Zheng, X. Luo, et al. A detailed and real-time performance monitoring framework for blockchain systems. *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice*, pages 134–143, 2018.

[131] I. Weber, V. Gramoli, A. Ponomarev, et al. On availability for blockchain-based systems. *IEEE 36th Symposium on Reliable Distributed Systems*, pages 64–73, 2017.

[132] H. Kalodner, S. Goldfeder, A. Chator, et al. *BlockSci: Design and applications of a blockchain analysis platform*, 2017. preprint.

[133] T. T. A. Dinh, J. Wang, G. Chen, et al. Blockbench: A framework for analyzing private blockchains. *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1085–1100, 2017.

[134] Anuj Das Gupta. Andrew Dickson. Analyzing performance in blockchain-based systems.

[135] A. Gervais, G. O. Karame, K. W"ust, et al. On the security and performance of proof of work blockchains. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 3–16, 2016.

[136] Y. Li, K. Zheng, Y. Yan, et al. *EtherQL: A Query Layer for Blockchain System*. Springer, Cham 556-567, international conference on database systems for advanced applications edition, 2017.

[137] Omar Abdullah Lajam and Tarek Ahmed Helmy. Performance evaluation of ipfs in private networks. In *2021 4th International Conference on Data Storage and Data Engineering*, pages 77–84, 2021.

[138] Aisyah Ismail, Mark Toohey, Young Choon Lee, Zhongli Dong, and Albert Y Zomaya. Cost and performance analysis on decentralized file systems for blockchain-based applications: State-of-the-art report. In *2022 IEEE International Conference on Blockchain (Blockchain)*, pages 230–237. IEEE, 2022.

[139] Jiajie Shen, Yi Li, Yangfan Zhou, and Xin Wang. Understanding i/o performance of ipfs storage: a client's perspective. In *2019 IEEE/ACM 27th International Symposium on Quality of Service (IWQoS)*, pages 1–10. IEEE, 2019.

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

[141] Bitcoin.com. *Report: Censorship-Prone Eos Needs to Re-Architect Its Infrastructure*. https://news.bitcoin.com/report-censorship-prone-eos-needs-to-re-architect-its -infrastructure/.

[142] Marcos K Aguilera, Jeffrey C Mogul, Janet L Wiener, Patrick Reynolds, and Athicha Muthitacharoen. Performance debugging for distributed systems of black boxes. *ACM SIGOPS Operating Systems Review*, 37(5):74–89, 2003.

[143] Patrick Reynolds, Janet L Wiener, Jeffrey C Mogul, Marcos K Aguilera, and Amin Vahdat. Wap5: black-box performance debugging for wide-area systems. In *Proceedings of the 15th International Conference on World Wide Web*, pages 347–356. ACM, 2006.

[144] Paramvir Bahl, Ranveer Chandra, Albert Greenberg, Srikanth Kandula, David A Maltz, and Ming Zhang. Towards highly reliable enterprise network services via inference of multi-level dependencies. In *ACM SIGCOMM Computer Communication Review*, volume 37, pages 13–24. ACM, 2007.

[145] Erik Daniel and Florian Tschorsch. Ipfs and friends: A qualitative comparison of next generation peer-to-peer data networks. *IEEE Communications Surveys & Tutorials*, 24(1):31–52, 2022.