Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/48/20241132

Design, development, and deployment of decentralized applications

Zefeng Chen

School of Cyber Science and Engineering, Southeast University, Nanking, 211189, China

213203344@seu.edu.cn

Abstract. Beginning with a comprehensive definition of Decentralized Applications (DApps) and their developmental trajectory, this treatise delves into their inception around 2010. It is intriguing to note that by 2020, DApps had already found preliminary applications in diverse sectors, ranging from finance to archaeology. Yet, there remains vast untapped potential awaiting exploration and refinement within the realm of DApps. The discourse then navigates the intricate web of DApps' system architecture, illuminating the cardinal aspects of their design, evolution, and eventual deployment. Herein, the essence of systematic planning during the design phase is underscored, underpinning its pivotal role in shaping the efficacy of the application. Further shedding light on DApps' expansive utility, the paper underscores their transformative influence in areas such as authentication systems and real-time operational control. However, the journey of DApps is not without its challenges. The document elucidates the complexities associated with crafting robust smart contracts, mitigating scalability concerns, and nurturing user acceptance and integration. In light of these hurdles, a clarion call is made for persistent research and avant-garde innovation, propelling DApps to their true potential in the evolving digital landscape.

Keywords: Blockchain, Smart Contract, Decentralized Applications.

1. Introduction

Decentralized applications represent a novel class of software applications rooted in blockchain technology, with their origins tracing back to circa 2010. Their adoption across various sectors underscores their relevance in contemporary times. Here are some distinguishing advantages over their traditional counterparts:

Decentralization: Operating on a distributed network or blockchain, DApps eliminate the dependency on a centralized server or authority. This architecture bolsters their resilience against single points of failure, ensuring greater reliability and resistance to censorship.

Security: With the inherent attributes of blockchain offering robust security and immutability, data within DApps remains safeguarded on a distributed network. The integration of cryptographic measures further renders these applications resistant to malicious intrusions [1]. Transparency: One of blockchain's salient features is its transparency. It grants audibility to all transactions and operations, allowing users to trace and authenticate every transaction's history, thereby upholding fairness and transparency. Trust Minimization: By bypassing the need for a centralized authority, DApps obviate the

DOI: 10.54254/2755-2721/48/20241132

necessity for blind trust in a singular entity, mitigating potential abuse or malfeasance risks. Openness: The open-source nature of many DApps invites scrutiny, fostering innovation and stimulating community participation by allowing everyone to inspect their codebase.

Yet, it's essential to acknowledge that the realm of decentralized applications is nascent, with inherent challenges awaiting resolution — such as latency issues, throughput bottlenecks, among others. The majority of software developers, while seasoned, may lack a holistic understanding of DApps. They might grapple with aligning their software's design, development, and deployment to the unique nuances and constraints posed by blockchain technology and decentralized applications [2]. The cardinal aim of this study is to galvanize the evolution of the blockchain-centric application ecosystem. By delving deep into DApps' intricate design and development, this research endeavors to augment understanding and awareness of DApps within the blockchain milieu, thereby propagating blockchain technology's widespread assimilation. In doing so, it paves the way for conceiving and actualizing decentralized application solutions across diverse scenarios.

2. Relevant theories

2.1. Definition of Decentralized Applications (development history)

Decentralized applications are applications built on a decentralized network, combining smart contracts and a front-end user interface. DApps have back-end code that runs on a P2P decentralized network and have front-end code and a user interface that calls its back-end. In addition, its front-end can be hosted on decentralized storage [3].

According to the definition given in the literature, decentralized apps have the following four characteristics:

- Open source: Due to the trustworthiness of the blockchain, dApps need to make their code open source for third-party auditing.
- Internal cryptocurrency support: Internal cryptocurrency is the tool to run a particular dApp ecosystem. With tokens, dApps can quantify the number of system all credits and transactions between participants, including content providers and consumers [4].
- Decentralized Consensus: Consensus among decentralized nodes is the basis for transparency.
- No Central Points of Failure: A fully decentralized system should have no central points of failure, as all of the application's nodes have a central point of failure.

The development history of Decentralized Applications could be traced back to early 2010s. In that time, developers put their main efforts on programming system, languages like solidity, applications that were used for tasks such as processing simple transactions and decentralized control abstractions [5]. These efforts aimed to solve problems and engineer systems using a decentralized control mindset.

After this, decentralized applications are gradually being used for actions and content outside of blockchain transactions. As early as 2022, decentralized applications have been used in archaeology, where it is used for decentralized management of evidence [6]. The application uses blockchain technology, specifically ethereum blockchain technology, to store and protect evidence and data related to archaeology and make it transparent and difficult to modify through the properties of blockchain technology.

And in the early 2020s, another important direction of decentralized applications, Decentralized Finance (DeFi) applications are gaining popularity, aiming to provide financial services in a decentralized and transparent way [7].

In blockchain, there are also some decentralized applications in the form of games that are gradually gaining popularity in the early 2020s, and they realize the idea that gamers dream of - virtual items and avatars that belong to them are unique [8]. However, due to the characteristics of decentralized applications, all of these games have been designed with some limitations, such as higher latency of the blockchain and limited throughput of blocks.

Although the idea of decentralization is currently very common in the case of decentralization being applied to a number of fields. However, the application to software development is still relatively new

DOI: 10.54254/2755-2721/48/20241132

[9]. At present, because of the characteristics of blockchain technology itself. Most of the latest decentralized applications still need external support to provide the full functionality of the application, such as smart contracts can not automatically trigger events, maintainability and identity management is more difficult than Web applications. These issues still need to be resolved in order to drive the further development and promotion of blockchain applications.

2.2. Decentraized Applications system

Decentralized Applications are built on blockchain technology, providing a flexible and better-incentivized structure compared to traditional software models. The system structure of a common decentralized application usually has the following components:

Blockchain &its network: The underlying technology of DApps, a blockchain is a distributed and cryptographically stored ledger that records all transactions and data in a transparent and immutable manner. It ensures the integrity and security of the DApp's data. The network between Blockchains is a Peer-to-Peer network which provides communication and data sharing among participants. This network is responsible for maintaining the decentralized nature of the application and ensuring its resilience [10].

Smart Contract: mart contracts are self-executing contracts with the terms of the agreement directly written into code. They are stored and executed on the blockchain, enabling trustless and automated interactions between parties. Smart contracts play a central role in decentralized applications, and they are responsible for executing the business logic of DApps.DApps typically utilize smart contracts to handle a variety of tasks, such as asset transfers, voting, and other operations that require consensus among participants. By executing smart contracts on the blockchain, DApps can ensure data integrity and security while eliminating dependence on centralized servers or trusted third parties.

Event and Notification System: Event and Notification System occupies an important position in decentralized applications. The Event and Notification System is responsible for the smart contract logic and event handling in the core smart contract of a decentralized application. They can issue events to notify other components or users of specific events or state changes. The event handling mechanism is used to process these events and trigger appropriate response actions, determining the main logic of the decentralized application. Its design also determines the security and privacy of the decentralized application, as well as his future scalability and performance.

Cryptocurrency: Cryptocurrency, like Bitcoin and Ethereum, is also a very important ingredient. It is usually seen as the currency circulating among users. Users own these currency to use them as their balance and pay for their use of blockchain services.

3. Design, Development, and Deployment of Decentralized Applications

3.1. Design and Plan of DApps

Designing and Planning Decentralized Applications Differ from Traditional Applications In the design and planning of decentralized applications, it is necessary to first consider the differences and similarities of the blockchain platforms on which they are running as well as the smart contract architecture used by decentralized applications in terms of functional implementation as compared to the architecture of traditional applications. Therefore, this thesis proposes the following process in order to facilitate developers in designing their own applications:

Clarify the problem and use cases: Clarify the problem to be solved by the decentralized application to be developed and the scenarios to which it applies. In particular, pay attention to whether the solution to the problem is in line with the decentralization idea, and whether the pain point of the problem can be better solved by using a decentralized application. At the same time, read the relevant development documents and literature to clarify whether the problem and scenarios can be realized under the framework of smart contracts, and if external means are needed (such as external scripts, front-end user pages, decentralized storage, etc.), it is also necessary to identify whether the chosen environment supports such external means.

DOI: 10.54254/2755-2721/48/20241132

Design Architecture: Design the overall architecture of the DApp, including the front-end, back-end, and blockchain components. Consider using a BaaS (Blockchain as a Service) architecture to connect the blockchain platform to other systems such as data storage and front-end applications. Also at this step, a basic design of the security and user privacy of this architecture should be done for subsequent testing and security performance. And the subsequent upgrading and expansion should be taken into account when designing the architecture, so as to avoid the difficulty of updating and replacing the subsequent applications:

Realize the development of smart contracts: Through the above design and the study of the relevant development documents, complete the realization of your application in the form of smart contracts.

Integration with other systems: If you need external help, you can connect the DApp with other systems such as data storage and front-end applications. For example, IPFS (Interplanetary File System) can be used as a decentralized storage solution for the DApp.

Ensure the security and privacy of the decentralized application: After the development of the basic framework and functionality, the application itself should be developed and tested for security and privacy.

Test and Deploy: At this point, the DApp should be almost complete, but test it thoroughly before deploying the DApp and smart contracts on the blockchain network. This will help you find and fix any issues or bugs.

Subsequent Updates and Optimizations: After the application is launched, optimize and enhance the user experience based on the feedback from the community users, and update the application with the latest blockchain technology as needed. Consider the scalability and performance of the DApp as the blockchain network may have limitations in terms of transaction throughput and latency. Optimize your smart contracts and architecture to handle increased demand and minimize transaction costs.

3.2. Development of Smart Contract

Developing a smart contract is the most central step in developing a decentralized application, and what sets it apart from the development of traditional applications is that the triggering of operations and functionality in a smart contract relies heavily on an event and notification system, which is critical to ensuring efficient information transfer and process automation. For example, ethereum smart contracts are favored by developers for their "event-triggered" mechanism, which allows for the construction of educational smart trading systems and smart learning platforms. Therefore, it is especially important to plan the most important few state variables at the beginning of designing the structure, which can effectively help us reduce the redundancy of the subsequent structure and improve security.

One paper describes an approach that combines Unified Modeling Language (UML) class diagrams and state-machine diagrams to model the structure and behavioral logic of smart contracts in multiple abstraction layers. This approach uses model-to-model transformations to generate Solidity Platform Specific Models (PSMs) from specified Blockchain Platform Independent Models (PIMs) and specified class state behaviors. The Solidity PSM is then used for Solidity smart contract code generation via model-to-text conversion.

Application security and privacy should also be considered early in development. Several researchers have studied security issues and provided contract security patterns for DApps, such as Termination pattern, Emergency Stop pattern, Speed bump pattern, Time constraint pattern and Check-Effects-Interaction pattern.

At the same time, when declaring the variables, we should also plan the memory size required by each variable as far in advance as possible, and simplify the code in the function as much as possible, which can effectively reduce the cost of running smart contracts. Some researchers have also studied this issue and proposed relevant patterns for reference. That is, Contract efficiency pattern: including Limit storage pattern, Minimize storage data pattern, Fewer functions and Limit modifiers pattern, Short constant string pattern, Tight variable packing pattern, Avoid redundant operations pattern and Low contract footprint pattern.

DOI: 10.54254/2755-2721/48/20241132

3.3. Deployment of DApps

Deploying DApps mainly lies in picking one or more blockchain networks to deploy on. This mainly depends on various factors such as the requirements of the application, scalability of the network, security and consensus mechanism. Developers can choose different blockchain networks such as Ether, Hyperledger Fabric, Corda, etc. for their different requirements. Ether is a popular choice for DApp development due to its smart contract feature and event triggering mechanism. Hyperledger Fabric, on the other hand, is the preferred choice for enterprise applications due to its permissioned network and modular architecture.

While deploying, developers need to consider the security and performance of the network as well as the absence of any logical vulnerabilities in the decentralized application itself. Blockchain networks are vulnerable to various attacks such as Denial of Service (DoS) attacks, which can affect network performance. And if it is due to exploitable logical vulnerabilities, it can result in substantial financial losses for users as well as developers.

4. Application Areas of DApps

4.1. Identity Authentication Systems

One of the more common areas where DApps are used is Identity Authentication Systems to deal with the challenge of authentication in blockchain networks. Here are some of the main applications:

Self-Self-Identity (SSI): dApps can implement the SSI principle through the use of Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs), allowing users to authenticate without relying on third-party services.

Enhanced Privacy and Security: dApps can leverage blockchain technology to provide enhanced privacy and security for authentication. Blockchain ensures consistent data storage and decentralized trust management, making it difficult for malicious actors to tamper with or falsify identity information.

Cross-domain authentication: In the context of cloud computing and IoT, interactions between customers in different application domains require secure and efficient cross-domain authentication. Blockchain-based dApps can solve the challenges faced by traditional PKI methods and provide a decentralized and efficient solution for cross-domain authentication.

MQTT Protocol Security: The security concerns of the MQTT protocol commonly used in IoT systems can be addressed by implementing blockchain-based identity and authentication schemes. These schemes leverage the decentralized and tamper-proof nature of blockchain to ensure the integrity and security of MQTT messaging in IoT environments.

4.2. Real time control of DApps

Decentralized applications are limited by the blockchain network, applications deployed in the blockchain network need to write operations in the blockchain, and the blockchain itself only has its own timestamp, so it is difficult to achieve complete de facto manipulation by controlling only in the smart contract itself. Currently how to implement control DApps is a popular topic that is often studied, and there have been many more mature real-time control schemes proposed, such as:

Oracles: Oracles are bridges that introduce external data into the blockchain and can be used to implement real-time control. By interacting with the oracles, smart contracts can access external data and make real-time decisions and operations based on that data. The prophecy machine can be provided by a trusted entity or by a trusted network of multiple entities. The prognosticator can provide a variety of data such as prices, weather, stock quotes, etc.

Layer-2 Solutions: Layer-2 solutions are protocols and networks built on top of the blockchain to provide higher throughput and lower latency. By moving some of the functionality of real-time control to Layer-2 solutions, higher performance can be achieved without sacrificing security and decentralization, Layer-2 solutions include state channels, payment channels, sidechains, and more.

Sidechain and off-chain computing: Sidechain and off-chain computing is the separation of some computation tasks from the main blockchain to improve performance and real-time. By performing real-

DOI: 10.54254/2755-2721/48/20241132

time control operations in the sidechain or off-chain, tasks can be responded to and executed faster without the limitations of the main blockchain. Sidechain and off-chain computing can be used to realize various application scenarios such as finance, IoT, gaming, etc.

Status Channels and Payment Channels: status channels and payment channels are a way to perform transactions and status updates outside the blockchain. By performing real-time control operations in a state channel or payment channel, higher throughput and lower latency can be achieved while still maintaining the security and decentralized nature of the blockchain. State channels and payment channels can be used to implement various application scenarios such as gaming, trading, voting, etc.

However, these solutions have their advantages and disadvantages, and need to be selected and implemented according to specific application scenarios and requirements. Further research is also needed to discover better solutions.

4.3. The environment of deploying DApps

Decentralized applications are software programs that run on a blockchain or peer-to-peer (P2P) computer network rather than on a single computer... They are free from the control and interference of a single organization, and have the advantages of protecting user privacy, freedom from censorship, and flexibility in development.... The environment in which dApps are deployed involves the following key components:

Blockchain Protocol: This is the network protocol on which the dApp is based. Ether is a popular platform for developing and deploying dApps because it provides accessible and transparent smart contracts.

Smart Contracts and the Blockchain Network: Smart Contracts are self-executing contracts where the terms of the agreement are written directly into the code. They are deployed on the blockchain and cannot be changed once deployed. Smart contracts are used for the logic and functionality of the dApp. The blockchain network, on the other hand, is the network that the app is deployed on, and the nature of the network will influence the development of the DApps, the protocols, and so on.

Front-end User Interface: The dApp combines a smart contract with a front-end user interface that allows users to interact with the dApp. The front-end code and user interface can be written in any language, similar to a traditional application.

Decentralized Storage: dApp data storage is typically decentralized, using technologies such as IPFS (InterPlanetary File System) Decentralized storage ensures data integrity and availability because it is not dependent on a single server or organization.

Open source and consensus: dApps are usually open source and required changes are determined by consensus of the majority of users. This requires that the code base be available for evaluation by all users and allows for community-driven development.

5. Challenges

Designing, developing, and deploying decentralized applications can be challenging due to their reliance on blockchain technology and smart contracts. Some of the main challenges include:

Writing secure smart contracts: Writing secure and safe smart contracts can be extremely difficult due to various business logics, as well as platform vulnerabilities and limitations.

Scalability issues: DApps often face scalability challenges due to the limitations of blockchain technology. As the number of users and transactions increases, the network may become slower and less efficient, affecting the user experience.

Interoperability: Interoperability between different blockchain networks and smart contract languages can be a challenge, as it requires coordination among multiple stakeholders and platforms.

Adoption and user education: DApps are still relatively new, and many users may not be familiar with the concept or the benefits they offer. Educating users and encouraging adoption can be a challenge for DApp developers and stakeholders.etc.

DOI: 10.54254/2755-2721/48/20241132

6. Conclusion

Research indicates that while the concept of decentralization has permeated various sectors, it grapples with substantial challenges, especially within the software development realm. Decentralized applications confront distinct hurdles in their design, development, and deployment stages, including high latency and significant deviations from conventional applications. Yet, these applications boast commendable security, resistance to censorship, and hold promise for diverse applicability across sectors. Despite the current nascent adoption and familiarity with DApps across industries, it's anticipated that, moving forward, they will leverage their inherent strengths to carve out an indispensable niche in numerous domains.

References

[1] Chen, Z., Fiandrino, C., & Kantarci, B. (2021). On blockchain integration into mobile crowdsensing via smart embedded devices: A comprehensive survey. Journal of Systems Architecture, 115, 102011.

[2] Wazan, A. S., & Cuppens, F. (2023). Cybersecurity in networking: adaptations, investigation, attacks, and countermeasures. Annals of Telecommunications, 78 (3-4), 133-134.

[3] Giannaros, A., Karras, A., Theodorakopoulos, L., Karras, C., Kranias, P., Schizas, N., ... & Tsolis, D. (2023). Autonomous Vehicles: Sophisticated Attacks, Safety Issues, Challenges, Open Topics, Blockchain, and Future Directions. Journal of Cybersecurity and Privacy, 3 (3), 493-543.

[4] Steiner-Otoo, D., & Jahankhani, H. (2022). An Investigation into How Smartphones Can Be Secured Against MiTM Attacks: Financial Sector. In Blockchain and Other Emerging Technologies for Digital Business Strategies (pp. 171-215). Cham: Springer International Publishing.

[5] Unal, D., Hammoudeh, M., Khan, M. A., Abuarqoub, A., Epiphaniou, G., & Hamila, R. (2021). Integration of federated machine learning and blockchain for the provision of secure big data analytics for Internet of Things. Computers & Security, 109, 102393.

[6] Baranski, S., & Konorski, J. (2020, November). Mitigation of fake data content poisoning attacks in ndn via blockchain. In 2020 30th International telecommunication networks and applications conference (ITNAC) (pp. 1-6). IEEE.

[7] Issa, W., Moustafa, N., Turnbull, B., Sohrabi, N., & Tari, Z. (2023). Blockchain-based federated learning for securing internet of things: A comprehensive survey. ACM Computing Surveys, 55(9), 1-43.

[8] Chaganti, R., Bhushan, B., & Ravi, V. (2022). The role of Blockchain in DDoS attacks mitigation: techniques, open challenges and future directions. arXiv preprint arXiv:2202.03617.

[9] Aliyu, I., Van Engelenburg, S., Mu'Azu, M. B., Kim, J., & Lim, C. G. (2022). Statistical Detection of Adversarial Examples in Blockchain-Based Federated Forest In-Vehicle Network Intrusion Detection Systems. IEEE Access, 10, 109366-109384.

[10] Miedema, F., Lubbertsen, K., Schrama, V., & van Wegberg, R. (2023). Mixed Signals: Analyzing {Ground-Truth} Data on the Users and Economics of a Bitcoin Mixing Service. In 32nd USENIX Security Symposium (USENIX Security 23) (pp. 751-768).