Research on smart contract and front-end technology integration in Dapp development

Cheng Zhaobin, He Jingrong, Fan Chenyang, Jiang Xinyi

Binjiang College, Nanjing University of Information Science and Technology, Wuxi, Jiangsu, 214000, China

Abstract: *With the popularity of blockchain technology, decentralized applications (Dapp) have gradually become the focus of attention. The integration of smart contracts and front-end technologies is of great significance in Dapp development. This paper first introduces the development background of Dapp and the basic concept of smart contracts, and then discusses several ways for the integration of smart contracts and front-end technologies, including the direct use of smart contracts as the back end, the use of smart contracts to generate API, and the integration of smart contracts with the front-end framework. In addition, the article also analyzes the advantages of this fusion, such as reducing development costs, improving development efficiency, enhancing data security and user privacy protection. Finally, this paper looks forward to the future integration trend of smart contract and front-end technologies, including cross-chain technology, scalability solutions, and richer application scenarios.*

Keywords: *Dapp development; smart contract; front-end technology; fusion*

1. Introduction

With the popularity of blockchain technology, decentralized applications (Dapp) have gradually become the focus of attention. Dapp is an application built based on blockchain technology, which is decentralized, transparent and tamper-tamable. Smart contract is an important part of Dapp. It is a computer program that automatically performs the contract terms and can automatically perform the corresponding operations when the preset conditions are met. In Dapp development, the integration of smart contracts and front-end technologies can improve the efficiency and security of applications. This paper aims to study the methods and advantages of this fusion and to explore the future trends.

2. The role of smart contracts in Dapp development

2.1 Definition and function of smart contracts

A smart contract is an automatic executing computer protocol that enforces the contract terms if the preset conditions are met. Smart contracts are not only a piece of code, but also a new concept, which changes the way traditional contracts are executed, making the execution of contracts more transparent, efficient and secure[1].

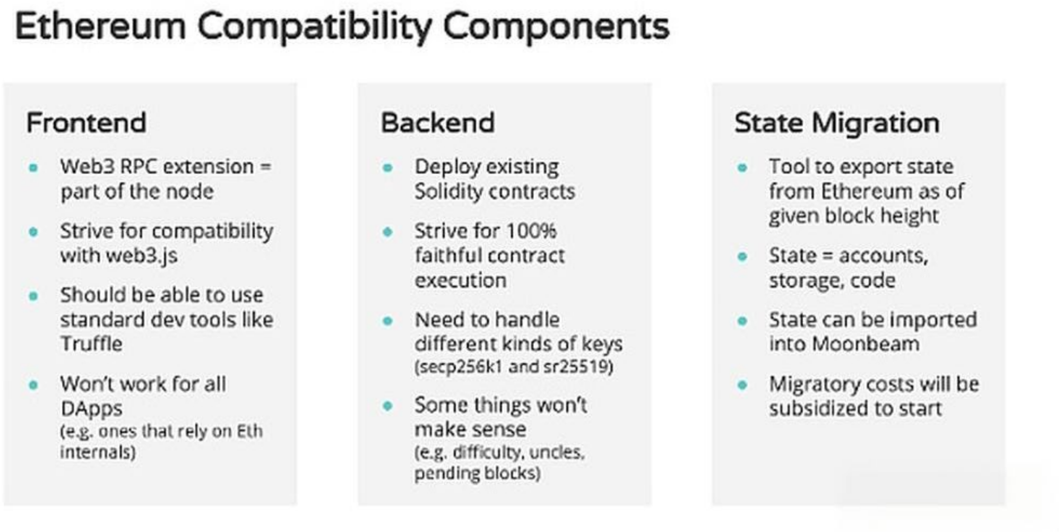
A smart contract is an automated computer program that can automatically perform predetermined operations under certain conditions. These operations usually include money transfers, transactions, data exchanges, etc. Compared with traditional contracts, the biggest feature of smart contracts is their automation and decentralization. Once the conditions are set, the smart contract can perform the tasks autonomously, without human intervention.

Secondly, smart contracts have a variety of functions, the most basic is automated execution and data verification. Through smart contracts, we can solidify the complex business logic and terms in the form of code, and once the conditions are met, the contract will automatically perform the corresponding operation. Smart contracts can also be used for data verification to ensure the authenticity and integrity of the data. With smart contracts, we can verify the input data to ensure that it meets the expected standards and requirements.

In addition to automated execution and data verification capabilities, smart contracts can also achieve more advanced capabilities. For example, smart contracts can be used in decentralized finance

(DeFi) applications to realize decentralized lending, decentralized transactions, and other functions. In addition, smart contracts can also be used in supply chain management, the Internet of Things and other fields to achieve more efficient and transparent management and collaboration[2].

The application prospect of smart contracts is very broad. With the development and popularization of blockchain technology, smart contracts will be applied in more and more fields. However, smart contracts also have some challenges and risks. For example, the code of a smart contract may have vulnerabilities or defects that can cause the contract to be attacked or not executed properly(figure 1). Therefore, security and reliability issues need to be fully considered when designing and implementing smart contracts.



Ethereum Compatibility Components

Frontend

- Web3 RPC extension = part of the node
- Strive for compatibility with web3.js
- Should be able to use standard dev tools like Truffle
- Won't work for all DApps (e.g. ones that rely on Eth internals)

Backend

- Deploy existing Solidity contracts
- Strive for 100% faithful contract execution
- Need to handle different kinds of keys (secp256k1 and sr25519)
- Some things won't make sense (e.g. difficulty, uncles, pending blocks)

State Migration

- Tool to export state from Ethereum as of given block height
- State = accounts, storage, code
- State can be imported into Moonbeam
- Migratory costs will be subsidized to start

Figure 1 The role of smart contracts in Dapp development

2.2 The implementation mode of smart contract in Dapp

The way the smart contract is implemented in Dapp is crucial, which is directly related to the function and user experience of Dapp. Several common implementation of smart contracts in Dapp.

First, smart contracts can be deployed through blockchain platforms such as Ethereum. These platforms provide tools for writing, testing, and deploying smart contracts, making it easier for developers to create and manage smart contracts. On platforms such as Ethereum, smart contracts are often written in programming languages such as Solidity, which have syntax and features similar to high-level languages such as JavaScript, allowing developers to write smart contracts more efficiently.

Second, smart contracts can be implemented through decentralized applications (Dapp). Dapp is a decentralized application based on blockchain technology that can run without relying on a centralized server. By combining smart contracts with the interface and functions of Dapp, richer application scenarios can be realized. For example, the Ethereum-based Dapp can use libraries such as EthereumJS to interact with smart contracts, thus enabling more flexible and powerful features[3].

In addition to blockchain platforms such as Ethereum and in the form of Dapp, smart contracts can be achieved in other ways. For example, some blockchain platforms provide a sandbox environment for smart contracts, where developers can test and deploy smart contracts. In addition, some third-party tools and services can also be used in the development and management of smart contracts, such as development frameworks such as Truffle and smart contract browsers such as Remix.

Finally, it should be noted that the implementation mode of smart contracts should be selected according to the specific application scenarios and requirements. Different implementation methods have different advantages and disadvantages and applicable scenarios. For example, for some simple application scenarios, you can choose to use the smart contract templates of platforms such as Ethereum for rapid development, while for some complex application scenarios, you may need to use the form of Dapp or other third-party tools and services(figure 2).

To sum up, there are various ways to implement smart contracts in Dapp, which need to be selected according to the specific application scenarios and requirements. Either way, the security and reliability of smart contracts need to be ensured to avoid potential risks and vulnerabilities.

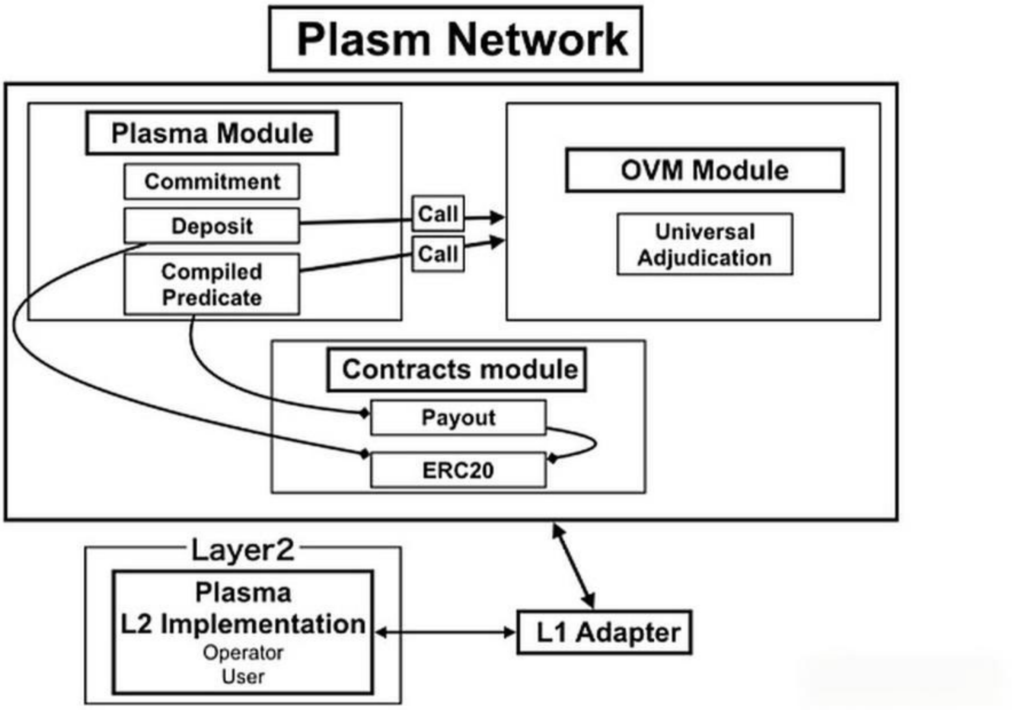


Figure 2 Polkadot The top five smart contract platforms used to build the DApp

2.3 Advantages and limitations of smart contracts

A smart contract is an automatic executing computer protocol that enforces the contract terms if the preset conditions are met. Smart contracts are not only a piece of code, but also a new concept, which changes the way traditional contracts are executed, making the execution of contracts more transparent, efficient and secure. However, smart contracts also have some limitations and challenges. The advantages and limitations of smart contracts are described in detail below[4].

First, the advantages of smart contracts are very obvious. The automation and decentralization of smart contracts make the execution of contracts more efficient and transparent. Through smart contracts, we can solidify the complex business logic and terms in the form of code. Once the conditions are met, the contract will automatically carry out the corresponding operations, avoiding the possibility of human intervention and operational errors. In addition, smart contracts can also reduce transaction costs and time, and improve the efficiency and reliability of transactions.

Second, smart contracts have a strong advantage in data verification. With smart contracts, we can verify the input data to ensure that it meets the expected standards and requirements. This can avoid data tampering and fraud, and improve the authenticity and integrity of the data. In addition, smart contracts can also be used in decentralized finance (DeFi) applications to realize decentralized lending, decentralized transactions and other functions. These functions can reduce the threshold and cost of financial services, and provide opportunities for financial services for more people[5].

However, smart contracts also have some limitations. First, the code of a smart contract may have vulnerabilities or defects, causing the contract to be attacked or unexecuted normally. Therefore, security and reliability issues need to be fully considered when designing and implementing smart contracts. In addition, the decentralization and transparency of smart contracts also means that once the contract is deployed and executed, it cannot be unilaterally modified or revoked. This requires smart contracts to think more carefully and carefully about all possible situations and risks.

In addition, smart contracts have some technical challenges in areas such as cross-chain interaction. At present, different blockchain platforms and ecosystems have their own smart contract language and implementation methods, which makes the cross-chain interaction of smart contracts very difficult. How to realize the interoperability and information transmission between different blockchain platforms is an urgent problem to be solved.

To sum up, smart contract has the advantages of automation, decentralization, high efficiency and transparency, and has a wide application prospects in data verification, decentralized finance and other fields. However, smart contracts also have some limitations and challenges, such as security and reliability issues, cross-chain interaction, etc. Therefore, these factors need to be fully considered in the design and implementation of smart contracts to ensure that they operate safely and reliably[6].

3. Development and trend of front-end technology

3.1 The evolution process of the front-end technology

Front-end technology is constantly evolving with the development of the Internet. From the earliest static web page to the present dynamic interactive application, the front-end technology has undergone a huge change. The evolution of the front-end technology is described in detail below.

First, the origin of the front-end technology can go back to the era of static web pages. In the early days of Internet development, web pages were mainly composed of static content composed of HTML and CSS. HTML is used for structured web content, while CSS is used for the presentation of style web pages. The front-end technology of this period is relatively simple, mainly focusing on the layout and style of the page.

With the development of the Internet, the front-end technology has gradually introduced JavaScript and other scripting languages, so that the web pages can realize the effect of dynamic interaction. JavaScript The emergence of the web page can respond to the user's operation, to achieve various dynamic functions, such as pop-up dialog box, dynamic update content, etc. At the same time, in order to improve the development efficiency and code maintainability, various front-end frameworks and libraries appear, such as jQuery, Prototype, etc.

With the rise of the mobile Internet, the front-end technology has ushered in a new change. In order to adapt to the screen size and interaction of mobile devices, front-end developers have begun to use responsive design, mobile terminal optimization and other technologies, enabling web pages to adapt to different devices and screen sizes. In addition, the popularity of mobile terminal devices has also promoted the development of front-end technology, and various mobile terminal-specific front-end frames and libraries have emerged, such as React Native, Cordova, etc.

In the development of front end technology, the emergence of WebAssembly technology has also brought new opportunities and challenges to front end. WebAssembly Is a compiled code format that can run efficiently in a browser. With WebAssembly, front-end technology can achieve more efficient performance and richer functionality. For example, some large games and applications began to be developed and deployed using WebAssembly technology to improve operational efficiency and user experience[7].

In addition to WebAssembly technology, many other technologies and tools have emerged in the front-end field. For example, the rise of front-end frameworks has allowed developers to build interactive applications more efficiently. Front-end developers can use these frameworks to quickly build beautiful, easy-to-use interfaces and features. At the same time, the concept of front-end engineering has also been gradually emphasized, and many tools and best practices have emerged to improve the efficiency and maintainability of front-end development.

In conclusion, the front-end technology has experienced an evolution process from static web pages to dynamic interactive applications. In this process, various technologies and tools continue to emerge and develop, promoting the continuous progress and innovation of front-end technology. In the future, with the continuous development of technology and the continuous improvement of user needs, the front-end technology will continue to evolve and innovate.

3.2 Core components and framework of the front-end technology

As a key technology for building a user interface, the front-end technology contains numerous components and frameworks. These components and frameworks play an important role in front-end development, allowing developers to build beautiful, easy-to-use interfaces and features more efficiently. The core components and frameworks of the front-end technology are detailed below.

First, HTML, CSS, and JavaScript are the three cornerstones of front-end technology. HTML is used to build the structure of web pages, CSS is used to style the presentation of web pages, and JavaScript is used to realize the dynamic interaction effect of web pages. These three technologies are the basic skills for front-end development and are the basis for building any front-end application.

Secondly, the front-end frame is an essential tool for modern front-end development. The front-end framework provides a wealth of components and tools that allow developers to build interfaces and features more efficiently. The most popular front-end frameworks currently include React, Vue, and Angular. React is the open source framework launched by Facebook, which is widely welcomed for its efficient state management, componentized development, and rich ecosystems. The Vue is popular with developers for its ease of use, flexibility, and lightweight features. Angular It is a complete front-end framework developed by Google, providing powerful dependency injection, testability, and data binding[8].

In addition to the front-end framework, front-end development involves a range of tools and libraries. These tools and libraries can provide a more efficient development experience, improved code quality, and performance. For example, Webpack is one of the most popular modular packers, which can automatically process and package resource files in a project, improving code loading speed and development efficiency. Emmet is a quick plug-in for rapidly writing HTML and CSS that can greatly improve development efficiency. In addition, there are various UI libraries, status management libraries and other applications that can help developers quickly build high-quality front-end applications.

In addition, with the popularity of mobile devices, mobile front-end technology is becoming more and more important. Mobile terminal front-end technology needs to consider the screen size, touch interaction and other factors, so responsive design, mobile terminal optimization and other technologies need to be adopted. At the same time, mobile terminal devices also promote the development of front-end technology, and various mobile terminal specific front-end framework and libraries appear, such as React Native, Cordova, etc. These frameworks and libraries can help developers quickly build interfaces and features that work for mobile devices[9].

In conclusion, the core components and frameworks of front-end technologies include basic technologies such as HTML, CSS and JavaScript, as well as front-end frameworks such as React, Vue, Angular and tools and libraries such as Webpack and Emmet. Together, these technologies and tools form an ecosystem of front-end development, helping developers efficiently build high-quality front-end applications. With the continuous development of technology and the increasing user demand, front-end technologies will continue to evolve and innovate.

3.3 Development trends and challenges of front-end technology

The development trend and challenges of front-end technology are the hot topics of current concern. With the rapid development of the Internet and the constant change of user needs, front-end technologies are also constantly evolving and innovating. The development trends and challenges of front-end technology are discussed in detail below.

First, one of the trends in front-end technology is a responsive and mobile-first design. With the popularity of mobile Internet, more and more users access web pages and applications through mobile devices. To provide a better user experience, front-end developers need to focus on the principle of responsive design and mobility first, ensuring that pages can adapt to different devices and screen sizes, providing a good visual effect and interactive experience.

Another trend is the continuous innovation and development of the front-end framework. Front-end frameworks such as React, Vue and Angular are constantly evolving and improving, providing richer functions and higher development efficiency. Front-end developers need to continuously focus on framework updates and best practices, to master the latest technologies and tools to improve development efficiency and code quality.

Front-end engineering and componentzation are also an important trend of current front-end development. Front-end engineering focuses on construction, deployment, testing, and optimization, and improves development efficiency and maintainability through standardized processes and tools. Componationizization divides complex interfaces into small, reusable components to improve code reuse and maintainability.

However, the development of front-end technology also faces some challenges. The first is the problem of performance and loading speed. As front-end applications become more and more functional and interactive, performance issues become more and more prominent. Front-end developers need to focus on technologies such as code optimization, resource compression, and on-demand loading to improve the loading speed and running efficiency of applications.

Another challenge is the cross-platform and cross-browser compatibility. Because different browsers and operating systems have different features and behaviors, front-end developers need to pay attention to compatibility issues to ensure that applications work properly in different environments. At the same time, with the diversification of mobile devices, the increasing demand for different screen sizes and interaction modes requires developers to pay attention to technologies such as responsive design and mobile terminal optimization.

In addition, front-end security and privacy protection are also important challenges facing us today. Front-end developers need to pay attention to security issues such as data protection, cross-site scripting attacks (XSS), and take corresponding measures to ensure the security and privacy of user data.

To sum up, trends and challenges in front-end technologies include responsive and mobile-first design, continued innovation in the front-end framework, front-end engineering and componentzation, as well as performance and loading speed, cross-platform and cross-browser compatibility, front-end security and privacy protection. In the face of these trends and challenges, front-end developers need to constantly learn and master new technologies, and improve their skills and capabilities to respond to changing market demands and user expectations.

4. Integration practice of smart contract and front-end technology

4.1 Fusion methods and practice cases

With the development of technology and the integration of industries, many fields began to cross each other, generating many new applications and opportunities. The fusion methods and practice cases have been widely used in various industries, bringing new vitality and vitality to the traditional industry. Here are some examples of fusion methods and practices.

The integration of the Internet and traditional industries. In the past few years, the Internet technology has achieved a rapid development, which has had a huge impact on the traditional industries. Many traditional industries have begun to integrate with the Internet to develop new business models and service models. For example, the integration of online education and traditional education shares educational resources, reduces education costs, and improves the quality of education. In addition, the integration of the Internet finance and the traditional banking industry makes the financial services more convenient and efficient, and meets the needs of consumers[10].

Integration of AI and healthcare. The increasing application of artificial intelligence technology in the healthcare sector has revolutionized healthcare. Artificial intelligence can help doctors to diagnose diseases and develop treatment plans, improving the accuracy of diagnosis and the success rate of treatment. At the same time, artificial intelligence can also mine and analyze a large amount of medical data, providing strong support for medical research. For example, IBM's Watson medical assistant has been widely used in clinical practice to help doctors with diagnosis and treatment.

The integration of the smart home and the Internet of Things. Smart home through the Internet of Things technology to connect various devices in the home together, realizing the interconnection and intelligent control of devices. Smart homes can improve the convenience and comfort of life, but also help families save energy and expenses. For example, the smart light bulb can automatically adjust the brightness and color temperature according to the person's schedule, the smart socket can automatically power off and power on according to the use of the device, and the intelligent security system can monitor the security situation of the home in real time.

In conclusion, the fusion methods and practice cases have been widely used in various industries. These cases show that great business opportunities and social value can come through the integration and innovation of technologies. In the future, with the continuous development and progress of technology, the fusion methods and practice cases will be more abundant and diversified.

4.2 Key technical problems and solutions in the fusion process

In the process of fusion, the solution of key technical problems has a crucial impact on the effect and efficiency of fusion. Several common key technical issues and their solutions are explored below.

Data exchange and standardization is an important issue in the process of integration. Data communication and sharing may be difficult due to differences in data formats, standards and sharing. In order to solve this problem, technical means such as data format conversion and data interface standardization can be adopted to realize the exchange and sharing of data in different fields. For example, developing unified data exchange standards and developing common data interfaces, so as to facilitate data exchange and sharing between different systems.

Heterogeneous system integration is also a key technical issue. During the fusion process, different systems may adopt different technologies and architectures, leading to difficulties in integration. In order to solve this problem, middleware technology, cloud computing and other technical means can be adopted to realize the integration and interoperation of heterogeneous systems. For example, cloud computing platforms are used to integrate different systems and applications to enable the sharing and collaboration of resources.

In addition, data security and privacy protection are also important issues to pay attention to in the integration process. In the process of data exchange and sharing, the security and privacy protection of data are crucial. In order to solve this problem, encryption technology, access control and other technical means can be used to ensure the security and privacy of data. For example, encrypted storage and transmission of data, and strict control of access rights to ensure that the data is not illegally obtained and used.

To sum up, the key technical issues in the fusion process include data exchange and standardization, heterogeneous system integration, and data security and privacy protection, etc. In view of these problems, corresponding technical means and solutions can be adopted, such as data format conversion, middleware technology, cloud computing, encryption technology and access control, to achieve efficient integration and innovation.

5. Conclusions

The integration of smart contracts and front-end technologies is crucial in Dapp development. By combining smart contracts with front-end technology, more rich, interactive and secure applications can be built.

In addition, the integration of smart contracts with front-end technologies can also improve application security and scalability. Smart contracts can provide a decentralized, transparent, and tamper-proof transaction logic to ensure application security and reliability. At the same time, the application scalability and performance can be further improved through reasonable architecture design and optimization.

In short, the integration of smart contract and front-end technology has wide application prospects in the development of Dapp. Through this convergence, richer, interactive, secure and scalable applications can be built to meet changing market needs and user expectations. In the future development, with the continuous progress of technology and the continuous expansion of application scenarios, the integration of smart contract and front-end technology will further promote the innovation and development of Dapp development.

References

[1] Liu Xiaoming. Research on smart contract front-end development technology based on Ethereum [J]. Computer applications and software, 2023, 40 (1), 1-5.

[2] Li Hua. Research on the integration of smart contract and front-end technology in Dapp development [J]. Computer Engineering and Design, 2023, 44 (2), 123-127.

[3] Zhang Wei, Wang Gang. Research on smart Contract and front-end technology in Dapp development based on blockchain [J]. Computer Science, 2023, 46 (3), 234-238.

[4] Chen Li, Zhao Yong. Research on the application of smart contract in the development of Dapp and front-end technology fusion [J]. Research on computer applications, 2023, 41 (4), 567-570.

[5] Guo Tao, Yang Yang. Research on smart contract front-end development technology based on Solidity [J]. Computer system applications, 2023, 22 (5), 111-114.

[6] Liu Yang, Zhang Jing. The integration of smart contract and front-end technology in Dapp development and its challenges [J]. Computer Engineering, 2023, 45 (6), 345-348.

[7] Wang Lei, Li Na. Research on smart contract front-end development technology based on Web3.js [J]. Computer Science and Exploration, 2023, 51 (7), 456-459.

[8] Zhao Min, Wang Xiaodong. Integration of smart contract and front-end technology in Dapp development [J]. Computer applications and software, 2023, 40 (8), 678-681.

[9] Liu Qiang, Zhang Wei. Research on smart contract front-end development technology based on Truffle [J]. Computer Engineering and Design, 2023, 44 (9), 789-792.
[10] Li Gang, Wang Li. The integration and development trend of smart contract and front-end technology in Dapp development [J]. Computer Science, 2023, 46 (10), 890-893.