Cracking the Code: Web 3.0 Software Development Challenges and Guidelines

Surya Bahadur Kathayat
Department of Computer Science, Norwegian University of Science and Technology, Trondheim, Norway

Keywords: Web 3.0, Sofware Development, Challenges and Guidelines.

Abstract: In the recent years, as the Web 3.0 continues to shape the digital landscape, developers faces many challenges in navigating this dynamic terrain. This paper, informed by extensive literature review and insights from multiple Web 3.0 applications development, examines the primary hurdles faced in Web 3.0 software development. These challenges span from scalability and interoperability to security and user experience. Furthermore, the paper presents essential architectural guidelines and best practices aimed at assisting developers in overcoming these obstacles and constructing robust and future-ready Web 3.0 applications. By addressing these observed challenges and adhering to the proposed guidelines, developers can effectively harness the full potential of Web 3.0 and facilitate its widespread acceptance.

1 INTRODUCTION

Web 2.0 and Web 3.0 represent two distinct phases in the evolution of the internet, each characterized by its own unique features, capabilities, and philosophies regarding user interaction and data management. Web 2.0, often referred to as the *participative social web*, signifies the transition from static web pages to dynamic or user-generated content and the growth of social networks. It's the era of the internet that enabled users to interact with sites and with each other, creating, sharing, and commenting on content online. This phase saw the rise of blogs, social media platforms, wikis, and services that emphasized user collaboration. The underlying technology of Web 2.0 mainly includes client-server based frameworks, making data centralized and in the hands of a few large companies, leading to concerns over privacy, data ownership, and the monetization of personal information.

Web 3.0, on the other hand, is the *decentralized web*, aiming to make internet more autonomous, intelligent, and open. The Web 3.0 emphasizes decentralization, aiming to give users more control over their data through decentralized technologies, which facilitate secure, transparent transactions and interactions without the need for centralized authorities or intermediaries. In essence, the transition from Web 2.0 to Web 3.0 marks a shift from a centralized web dominated by large corporations to a decentralized ecosystem that prioritizes user privacy, data ownership, and intelligent, personalized web experiences. While Web 2.0 made the internet more social and interactive, Web 3.0 seeks to make it more secure, intelligent, and user-centric.

As we navigate this new era, characterized by decentralized architectures, enhanced user experiences, and advanced data handling capabilities, developers are faced with a myriad of challenges and opportunities. In this paper, we delve into the intricacies of developing software for the Web 3.0 environment. Our exploration encompasses not only the technical hurdles but also the strategic considerations essential for success in this dynamic ecosystem. By identifying key challenges and offering practical guidelines, we aim to provide developers and innovators with the insights and tools necessary to navigate the complexities of Web 3.0 software development effectively. Through this endeavor, we seek to contribute to the advancement of Web 3.0 technology and empower innovators to unlock its full potential.

Here's what's coming up: In Section 2 we'll examine relevant literature. Section 3 outlines the research methodology and the experiments conducted. The subsequent sections present guiding principles, challenges and technical advice for Web 3.0 software development. A comprehensive discussion and conclusion is presented towards the end.

Kathayat, S.
Cracking the Code: Web 3.0 Software Development Challenges and Guidelines.
DOI: 10.5220/0012852800003753
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 19th International Conference on Software Technologies (ICSOFT 2024)*, pages 458-468
ISBN: 978-989-758-706-1; ISSN: 2184-2833
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

2 RELATED LITERATURE

2.1 DApps Development

Central to the Web 3.0 transformation are decentralized applications (DApps) (Cai et al., 2018). The DApps have some sort of decentralized storage, and user interface design! Developing robust and scalable Web 3.0 applications presents unique challenges and require innovative methodologies and best practices. Most of the literature are based on the Ethereum blockchain platform, which offers unprecedented levels of transparency, security, and trustlessness. Authors (Wöhrer and Zdun, 2021) explore practical approaches to building DApps using Ethereum, with a focus on design patterns, security, and scalability. Raval's *Decentralized Applications: Harnessing Bitcoin's Blockchain Technology* offers hands-on guidance for building DApps using blockchain technologies beyond Ethereum (Raval, 2016).

2.2 Smart Contracts

Smart contracts serve as the backbone of Web 3.0 applications, enabling automated execution of digital agreements. *Mastering Ethereum* by Antonopoulos and Wood offers comprehensive guidance on writing, deploying, and interacting with smart contracts (Antonopoulos and Wood, 2018). Additionally, *Solidity Programming Essentials* by (DeForest, 2023) provides practical insights into Solidity, the primary language for writing smart contracts. Ethereum Cookbook by (Manoj, 2018) offers a collection of recipes for common Ethereum development tasks, including deploying and interacting with smart contracts.

2.3 Integration and Interoperability

Like a traditional software development, web 3.0 development also intersects integration and interoperability as developers design, implement, and test systems to facilitate seamless communication and data exchange between diverse software components and external systems. There are some standardized data formats and protocols, and leveraged middleware to streamline integration processes. *Blockchain Basics* by (Drescher, 2017) offers insights into integrating blockchain solutions with existing web 2.0 systems. *Enterprise Blockchain Development* in (Lee and Ghosh, 2018) explores enterprise-grade blockchain development methodologies, covering topics such as consortium networks, permissioned blockchains, and interoperability standards. *Cryptoassets* by (Burniske and Tatar, 2017), navigates the complexities of cryptocurrencies, offers insights into their integration with traditional web services. *Understanding Ethereum* by (Scott, 2017) provides exploration of the platform's architecture, protocols, and design principles, shedding light on its interoperability with existing web infrastructures.

2.4 Security and Governance

Security is paramount in Web 3.0 development, particularly when dealing with smart contracts and decentralized applications. Sajid's *Ultimate Blockchain Security Handbook* offers insights into securing smart contracts and DApps on the Ethereum blockchain, including best practices for auditing and testing (Sajid, 2023). *Blockchain Cybersecurity, Trust and Privacy* by (Choo et al., 2020) delves into various DApp use cases and their implications for privacy, security, and governance. *The DAO of Capital* and *Blockchain and the Law* respectively explores the concept of decentralized autonomous organizations (DAOs) in Web 3.0 governance and delves into legal and regulatory challenges associated with decentralized applications (De Filippi and Wright, 2018).

2.5 Web 3.0 Game Developments

The game development within the context of Web 3.0 is a burgeoning area with a focus on exploring the convergence of blockchain technology, decentralized systems, and gaming (Sahin, 2023). The literature in this domain typically investigates the integration of blockchain and smart contracts to enable novel gameplay mechanics, true ownership of in-game assets through non-fungible tokens (NFTs), and decentralized economies within gaming ecosystems (Min et al., 2019). Additionally, research in game design and user experience examines how to optimize user onboarding processes in decentralized environments (Iyer and Dannen, 2018). The literature in this field is still emerging.

2.6 The DeFi Revolution

Development on decentralized finance (DeFi) is probably one of the most talked topic of Web 3.0. Friesendorf et al.'s *Decentralized Finance (DeFi): How Decentralized Applications (DApps) Disrupt Banking*, offer deep explorations into the intricate mechanics of DeFi protocols, including decentralized exchanges (DEXs), lending platforms, and governance mechanisms (Friesendorf, 2023). Technical documentation and whitepapers from leading DeFi projects, like Uniswap's whitepaper *Automated Market Making* (Uniswap, 2024) and Compound's *The Money Market Protocol* (Compound, 2024), provide foundational knowledge and implementation details crucial for developers and researchers. Furthermore, conferences like the Ethereum Foundation's Devcon and ConsenSys Summits serve as hubs for the dissemination of cutting-edge research, best practices, and innovative ideas in the realm of decentralized finance, fostering collaboration and knowledge exchange among developers, and industry stakeholders.

3 METHODOLOGY

Our research methodology draws inspiration from agile principles and design science research, focusing on tackling one problem at a time and iterating in small steps. This approach prioritizes flexibility, adaptability, and continuous improvement, enabling us to effectively address changing requirements and uncertainties. By integrating experimentation, hypothesis testing, and data-driven decision-making, Agile Design Science fosters curiosity, exploration, and learning. This allows for continuous prototyping, and shorter user feedback loops to refine solutions iteratively and drive innovation. Through this iterative process, Agile Design Science promotes creativity, resilience, and a culture of continuous learning, ultimately leading to more impactful outcomes.

3.1 Experiments

We've undertaken a series of experimental projects, each distinct in nature.

3.1.1 Academic Diploma Verification Platform

Our first experiment (Hustad and Fredrik, 2021) is to develop a simple decentralized app resembling a Web 3.0 wallet designed to revolutionize academic credentials storage and transparent verification. Users can import verified documents, such as academic diplomas, into a web wallet and share them with potential employers during recruitment. While the technology functions well, we've encountered several challenges. The primary issue is the performance of storing and retrieving large documents, along with associated gas fees for blockchain transactions. Additionally, storing documents transparently in a decentralized network somewhat limits data ownership aspects, despite the preservation of document integrity by Web 3.0 technology. The architecture of the application was quite simple - set of JavaScript based front-end components representing issuer and verifier, and a back-end with a set of smart contracts communicating to the blockchain.

3.1.2 Identity Wallet Platform

In this experiment (Bliudzius et al., 2022), we have envisioned a cutting-edge identity platform to transcend the boundaries of conventional digital identity based authentication methods. The idea behind was based on self-sovereign identity (SSI) where user identity verification (e.g Norwegian BankID) is done once, stored securely in the wallet in the form of verified credential (VC) and later can be used by the user for verification in the form of verifiable presentations (VP). This not only offers ownership of identity data back to the users, but also has large economical advantages along with increased privacy and security. The platform boasts both mobile and web-based interfaces, and seamlessly integrated web 3.0 into a sophisticated Web 2.0 back-end infrastructure. With this fusion of traditional web 2.0 functionalities with the immutable, trustless architecture of web 3.0 technology, we have drawn some Web 3.0. architectural guidelines and design principles ensuring seamless interoperability and enhanced security. The guidelines are comprehensively presented later section in this paper.

The overall architecture of the platform has following main components.

- Mobile and web wallets written in react-native and uses Veramo APIs for handling verifiable data and SSI.
- Web 2.0 backend serving as orchestrator for Issuers, User wallets, and Verifiers. It also manages keys, resolves identities and account data.
- Web 3.0 backend with a set of smart contracts written in Rust. It communicates the underlying Polkadot Blockchain.
- Issuers web applications mocking the issuance of identities.
- Verifiers or consumers web applications that simulate services consuming identities.

3.1.3 Web 3.0 Gaming Applications

We have then experimented by developing multiple web 3.0 games. First, in BlackJack (Gimre, 2023) and Tic Toc (Groven, 2023), we focused on the concurrency aspect, where players play the game simultaneously. We also wanted to have most of the game logic decentralized i.e in the blockchain. Technically, it is possible but we have faced several obstacles. Transaction delay was the main factor.

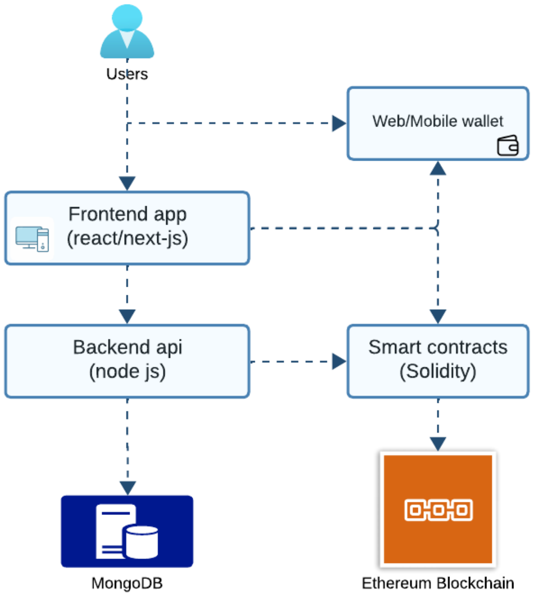

Figure 1: System architecture of the Web 3.0 games.

Then in TokenTrivia (Bendvold et al., 2024), we shifted our focus on having a optimal technical solution. We combined best of web 2.0 and web 3.0. It helped us to validate our architectural guidelines that underpin the development process, advocating for scalable, modular architectures that facilitate seamless integration of web 2.0 and web 3.0 ensuring optimal performance across diverse gaming environments.

The generic system architecture for web 3.0 enables games is shown in Figure 1. Note the learning from our earlier experiment on Identity Platform, we have decided to have *Node.js* based web 2.0 backend component. We discovered that this option offers better performance. Furthermore, there are numerous JavaScript-based frameworks and resources available that facilitated a swift start and allowed for a short development cycle.

4 THE GUIDING PRINCIPLES

After extensive experimentation and research, we've established best practices for Web 3.0 solutions, covering interoperability, scalability, security, and user experience. These guidelines provide developers with a roadmap for building resilient, future-ready applications in decentralized technologies.

4.1 Think Decentralization at the Core

One needs to think fundamentally differently when comes to web 3.0 software development compared to traditional web 2.0! In a decentralized system, there's no single point of control or failure; instead, responsibilities are shared among various nodes or participants ensuring resilience and autonomy. Accordingly, the software architecture should be such that distributes power, application logic, authority, and decision-making across a network rather than concentrating it in a single entity or server.

4.2 Continuous Innovation

A fundamental guiding principle for Web 3.0 software development is fostering a mindset and culture of continuous innovation. Given the youthful and evolving nature of the field, it's imperative to remain agile, embrace change, and adopt emerging technologies to stay ahead in Web 3.0 software development.

4.3 Scale with Agility

Scaling with agility in Web 3.0 development involves carefully selecting blockchain technology tailored to specific application needs, such as Ethereum for its robust ecosystem or Solana for high-speed transactions. Layer 2 solutions like state channels, sidechains, and rollups can enhance scalability by handling transactions off the main blockchain, thus improving throughput and reducing costs. Off-chain computation through oracles or services like The Graph helps optimize data handling and costs.

It's also critical to maintain agile development practices, focusing on iterative development, continuous integration, and automated testing to adapt quickly to changes. Optimizing smart contract code for efficiency, designing user-friendly interfaces, and staying updated with the latest technologies and regulations are also essential for creating scalable, efficient, and compliant Web 3.0 applications. This approach not only addresses technical scalability but also ensures the application remains accessible and practical for end-users, leveraging the decentralized nature of the web without compromising on performance.

4.4 Community-Centric Approach

State of the art web 3.0 software development is fragmented based on different blockchain platforms. Such platforms have their own dedicated communities engaging users, developments and innovators.

Many of the communities even engage users in governance, feedback loops, and decision-making processes to ensure your software evolves with the needs of the community. Therefore emphasize collaboration, transparency, and inclusivity. Involve or get involved in decentralized governance mechanisms like DAOs, open contribution to project development, token-based incentives, education, and a commitment to transparency and accountability. This approach empowers users, developers, and stakeholders to actively participate, shape projects, and collectively advance further.

4.5 Interoperability Bridges

Web 3.0 is still very diverse! It is useful to use protocols or mechanisms that facilitate communication and data exchange between different decentralized systems such as blockchain networks. This enables ability of diverse systems and platforms to work together seamlessly and overcome the fragmentation and siloed nature of blockchain ecosystems. This also means fostering ecosystem growth and collaboration enabling organizations to leverage existing infrastructures and tap into diverse networks.

However, while these bridges offer significant advantages, they also introduce security risks, as the bridge itself can become a target for attacks. Developers must therefore also focus on building secure bridging protocols to safeguard cross-chain interactions.

4.6 Fortify with Security

Boost the security of your applications by incorporating robust measures to defend against potential threats. Utilize encryption, conduct frequent audits, and implement trustworthy authentication methods to fortify your systems. By adopting these comprehensive security practices, you can safeguard your data and infrastructure from unauthorized access and malicious activities. Additionally, fostering a culture of security awareness within the development process ensures that all stakeholders remain vigilant and proactive in addressing evolving security challenges.

4.7 Empower User Sovereignty

Provide individual users with the empowerment to oversee their digital existence by granting them autonomy over their data, identities, and online interactions. This empowerment entails not only ownership and management of personal data but also the ability to regulate digital identities, safeguard privacy, embrace decentralized platforms, and prioritize design that revolves around the needs and preferences of users. By considering these principles, users can feel more empowered and in command of their online experiences, fostering a digital environment that is both inclusive and respectful of individual rights and preferences.

4.8 Tokenomics for Sustainability

Tokonomics and reward is the beauty of web 3.0. Especially rewarding those who offer distributed resources such as computing power is good in terms of sustainability. Therefore if your application has tokonomics, design ecosystems with robust and sustainable token economics. Incentivize participation and contribution while maintaining long-term viability and value creation.

4.9 Craft Seamless Experiences

Engaging users is a fundamental challenge in any software systems! This is even more essential for young and emerging web 3.0 systems. Therefore craft experiences that captivate and delight end users. Prioritize intuitive interfaces, friction less onboarding for example from web 2.0 to web 3.0, and user-centric design to ensure adoption and retention.

4.10 Immutable Accountability

In Web 3.0, *immutable accountability* refers to using blockchain technology to record transactions in a way that cannot be altered, ensuring all actions are permanently and transparently logged. This approach is crucial in applications like smart contracts and decentralized apps (DApps), where it guarantees that once data or agreements are entered into the blockchain, they remain unchangeable and verifiable by all participants. This transparency builds trust, reduces fraud, and ensures that all network activities can be audited at any time, fostering a reliable and secure digital environment. On the other hand, it may have consequences against privacy. So use it wisely.

5 THE ROADBLOCKS

The following challenges emerged during our endeavors.

5.1 Complexity in Development

One of the most formidable challenges we faced in Web 3.0 software development is the steep learning curve associated with blockchain technology and decentralized systems. Developers must grapple with new programming languages like Solidity for Ethereum, Rust for Polkadot, etc as well as understand the intricacies of smart contracts and consensus algorithms.

5.2 Scalability Challenges

When it comes to multiple users and increased number of transactions, the underlying blockchain platforms face scalability issues, struggling to handle increased transaction volumes without compromising on speed or inflating costs.

5.3 Security Vulnerabilities

The immutable nature of blockchain is a double-edged sword. While it provides security against data tampering, it also means that any vulnerability in smart contracts or the decentralized application itself is permanent and can be exploited repeatedly. That requires careful design and development of smart contracts.

5.4 User Experience Hurdles

Creating user-friendly experiences in Web 3.0 is quite challenging and they stem from the complexity of decentralized technologies. Challenges include managing wallets and cryptographic keys, high transaction fees and slow speeds, complex onboarding processes, unfamiliar interfaces, lack of support systems, and data privacy concerns. Simplifying wallet management, streamlining onboarding processes, improving interface design, establishing robust support systems, clarifying regulations, and enhancing data privacy measures are key steps to address these hurdles and make Web 3.0 more accessible and user-friendly.

5.5 Regulatory Ambiguity

Regulatory ambiguity poses significant challenges for Web 3.0 development. The decentralized world is still in a legal gray area, with varying regulations across jurisdictions and different geographic locations. Unclear legal status, compliance burdens, regulatory arbitrage, enforcement risks, and impacts on innovation hinder progress. Collaboration between stakeholders is very crucial to develop clear and balanced regulatory frameworks that support innovation.

5.6 Privacy and Transparency

Privacy and transparency in Web 3.0 present a delicate balance. While decentralized technologies offer increased privacy by decentralizing control over personal data, they also introduce challenges. Immutable public ledgers, like blockchains, may compromise user confidentiality, while fully private transactions can hinder auditability. Similarly, while blockchain enhances transparency by providing an immutable record, it raises concerns about data visibility. Addressing these challenges requires innovative solutions, collaboration among stakeholders to develop and promote standards that promote both privacy and transparency.

6 BEST PRACTICES: THE ROUTE TO SUCCESS

Choosing the right technologies is crucial for developing robust, scalable, and future-ready web 3.0 applications. The recommendations provided in this section are based on our experience and expertise in the current technological landscape, aiming to empower individuals to make informed decisions.

6.1 Blockchain Platforms

Various Blockchain platforms support app development, but they often lack compatibility, requiring apps to be rewritten for different platforms. Our experiments have been conducted on Ethereum and Polkadot. Some popular options for app development include Ethereum, Polkadot, Solana, Cardano, etc. Select the one that is suitable for your requirements.

6.1.1 Ethereum

It remain the most popular choice for developing DApps due to its widespread adoption, smart contract functionality, and developer-friendly ecosystem. It provides Ethereum virtual machine for running DApps. However, Ethereum also faces challenges, such as scalability issues and high gas fees.

6.1.2 Polkadot

It's interoperability and scalability features make it an excellent choice for building cross-chain DApps. Its multi-chain framework enables parallel processing and seamless communication between diverse blockchains.

6.1.3 Solana

It is known for its high throughput and low latency. With its innovative consensus mechanism, Solana can process thousands of transactions per second, making it ideal for demanding applications like DeFi and NFTs.

6.2 Smart Contract Languages

Solidity (Zheng et al., 2020) and Rust (Sharma, 2024) are two of the most widely used smart contract languages. Solidity is for Ethereum-based DApps, offering a JavaScript like syntax and extensive tooling support. Rust is the preferred language for building smart contracts on Substrate-based blockchains like Polkadot, offering strong type safety and performance benefits.

6.3 Web Development Frameworks

Web3.js and ethers.js are two popular libraries that allow developers to interact with Ethereum nodes and smart contracts, making it essential for building Ethereum-based DApps. The ethers.js is newer, modular, and lightweight.

Moralis is a backend infrastructure provider that simplifies Ethereum development by offering tools and services for managing user authentication, data storage, and real-time updates. Truffle Suite and Hardhat are the popular development environments providing a comprehensive development environment for Ethereum developers, including tools for smart contract compilation, debugging, testing, and deployment.

Popular web 2.0 frontend frameworks, such as React.js, Vue.js or Angular can be seamlessly used for web 3.0 app developements. All of these offer a component-based architecture and a large ecosystem of libraries and tools for building interactive and responsive DApps, making it suitable for complex Web 3.0 projects.

6.4 Distributed Storage

Data and information in web 3.0 applications are distributed along time and location axis. The information can be in the form of files, log messages, structured data in databases or transactions.

6.4.1 Interplanetary File System

Is a distributed file storage protocol that enables decentralized and censorship-resistant data storage. It is commonly used in Web 3.0 applications for storing content such as images, videos, and documents.

6.4.2 Databases

Store various types of structured data, such as user profiles, transaction records, and application state. They support CRUD operations to manipulate data and enforce data consistency and integrity through transactions and constraints. Use of databases in Web 3.0 is similar as in Web 2.0, and can be implemented using SQL, or NoSQL databases depending on the requirements.

6.4.3 Logs

In Web 3.0 applications record events such as user interactions, system errors, smart contract executions, etc. They capture metadata like timestamps, user IDs, and transaction IDs to provide context for each event. Logs are essential for diagnosing and troubleshooting issues, auditing user activities, and analyzing system performance. In addition, Logs in blockchain systems serve to record and provide transparency for various events and transactions within the network.

6.4.4 Ledgers

In Web 3.0 applications maintain a distributed ledger of transactions and smart contract state changes across multiple nodes in a network. They use consensus mechanisms like Proof of Work or Proof of Stake to validate and append new transactions to the ledger in a secure and decentralized manner. Ledgers are used for recording financial transactions, asset transfers, digital identity management, and other decentralized interactions. They ensure data integrity, transparency, and censorship resistance by providing a shared source of truth that is verifiable by all participants in the network.

6.5 Oracles

Oracles play a crucial role in decentralized applications, and should be used whenever possible. They act as bridges between blockchain smart contracts and real-world data sources. They fetch and verify external information ensuring data authenticity and integrity. Common usages of oracles include:

- to provide real-time price feeds in DeFi protocols like decentralized exchanges or lending platforms
- to track real-time data in supply chains
- to provide weather data to smart contracts to hedge against weather-related risks, etc

6.6 The WebSockets

WebSockets play a pivotal role in Web 3.0 app development by establishing persistent, bidirectional communication channels between clients and decentralized nodes. Unlike traditional HTTP requests, which are stateless and require repeated polling for updates, WebSockets maintain an open connection, enabling seamless real-time updates and notifications. This allows DApps to receive instant updates on blockchain transactions, smart contract events, and other network activities, ensuring users stay informed about the latest developments in the ecosystem. Moreover, WebSockets provide efficient data transfer, minimizing network overhead and latency compared to polling. Maintaining persistent connections, they enhance the responsiveness and interactivity of Web 3.0 apps. Scalability is another key advantage of WebSockets, as they support distributed architectures and horizontal scaling across multiple servers or nodes. This enables Web 3.0 applications to handle large numbers of concurrent connections and maintain responsiveness even under heavy loads. Note however that, WebSockets should be secured using Transport Layer Security encryption, ensuring that data transmitted is encrypted and protected from eavesdropping. This is crucial for maintaining the confidentiality and integrity of user data.

6.7 Usage of Wallets

Wallets are key part of web 3.0 due to their decentralized in nature. Unlike in web 2.0, the wallets in web 3.0 allows users to have full control over their assets without the need for intermediaries. Here are guidelines for using wallets in Web 3.0 development.

6.7.1 Wallet Providers

Select established and reputable wallet providers, such as Metamask, with a track record of security and reliability. Prioritize wallets that are open-source and have undergone third-party security audits.

6.7.2 Privacy and Data Protection

Respect user privacy by minimizing data collection and ensuring the protection of sensitive information such as transaction history and account balances. Inform users about privacy features and options available within the wallet.

6.7.3 Regular Awareness

Educate users about security practices such as safeguarding private keys, using hardware wallets where possible, and enabling additional security features like multi-factor authentication. Encourage users to verify transaction details carefully before confirming transactions.

6.8 App Deployment Strategies

Smart contract and decentralized application deployment strategies require meticulous planning, focusing on effectiveness, security, reliability, and scalability. Here are some recommendations:

6.8.1 Network Selection

Select the suitable blockchain network considering factors such as scalability, security, and decentralization, with options like Ethereum. Solana and Polkadot, each presenting unique trade-offs.

6.8.2 Testing

Conduct comprehensive testing before deployment, including unit tests, integration tests, and security tests. Choose an appropriate deployment environment, starting with Testnets before transitioning to public or permissioned networks, each offering varying levels of access control, privacy, and scalability.

6.8.3 Upgradability and Governance

Consider requirements for contract upgradability and governance mechanisms to manage future changes and improvements to deployed contracts. Implement upgradable contract patterns or use governance protocols like DAOs to enable community-driven decision-making and consensus on contract upgrades.

6.8.4 Gas Costs and Transaction Fees

Deployments and transactions incur costs! Therefore, optimize contract code and transaction parameters to reduce gas consumption and improve cost-effectiveness.

6.8.5 Security Auditing

Implement security best practices (Naha, 2023) and conduct security audits to mitigate risks associated with smart contract vulnerabilities, such as reentrancy, overflow/underflow, and access control issues.

6.8.6 Tools and Infrastructure

Use tools and infrastructure to streamline the deployment process and automate repetitive tasks. Tools like Truffle Suite, Hardhat, and Remix provide features for contract compilation, testing, and deployment, making the deployment process more efficient and reliable.

7 DISCUSSION

7.1 Unprecedented Opportunities

Web 3.0 presents a fertile ground for innovative applications, spanning from DeFi as an alternative to traditional banking to NFTs revolutionizing art ownership and copyright. Users now have true ownership of digital assets, data, and online identities, promising enhanced privacy and control. With a focus on interoperability, Web 3.0 aims for a more connected internet where data and assets can move seamlessly across platforms. Open-source Web 3.0 projects foster community-driven development, enabling rapid iteration, innovation, and shared ownership. Decentralized identity solutions and the rise of DAOs are reshaping governance structures, offering new monetization avenues, community rewards, and enhanced network security. Possibilities are endless.

7.2 Unpredictable in Nature

Web 3.0 is still unpredictable in nature. The unpredictable nature stems from various factors challenging traditional internet governance, data management, and economic models. Regulatory ambiguity and security vulnerabilities persist despite cryptographic protocols, while interoperability issues hinder seamless integration among diverse platforms. The dynamic technological landscape and market dynamics further compound uncertainty, necessitating constant adaptation and collaboration within the ecosystem. This unpredictability affects software development practices, requiring developers to embrace agility, risk management, and modular design. Collaboration with decentralized communities, prioritizing user experience, and privacy considerations are crucial for navigating and capitalizing on Web 3.0's unpredictability.

7.3 Maturity Is Happening

Implementing Web 3.0 libraries has proved more complex than expected, primarily due to two factors. Firstly, a lack of up-to-date documentation online posed a significant challenge. Secondly, existing documentation often lags behind due to ongoing library development. For example, the 'Ethers.js' library saw a substantial update to version 6.9.2, rendering many examples obsolete. Similarly, while web3.js documentation for 2024 covers basic transactions, guidance on raw transactions remains incomplete. Developers often face challenges in finding the correct methods due to scarce online resources. These instances underscore the Blockchain environment's ongoing intensive development phase. This dynamic environment is characterized by rapid technological advancement and frequent updates, necessitating continuous monitoring and adaptation by developers. Drawing from the experiences, it is advisable to opt for versions of libraries with robust online documentation. This decision can significantly streamline the development process and contribute to enhanced productivity.

7.4 Maintenance of Smart Contracts

Smart contracts are known for their immutability once deployed on the blockchain, ensuring that their rules remain unchanged, thus fostering trust and security. However, this immutability poses challenges for maintaining and updating contracts. Developers must deploy entirely new contracts to add features or address bugs, involving intricate processes like transferring data and user deposits. Communication with users is crucial for informing them about updates and facilitating data migration to the new contract, minimizing disruptions. To address these challenges, developers have explored practices like 'upgradable' contracts or proxy contracts, offering flexibility for updates without deploying entirely new contracts. Despite their challenges, these approaches aim to balance stability with the need for adaptation and enhancement over time.

7.5 Reliability

Reliability posed the most significant challenge in our works with TokenTrivia within the realm of Web 3.0, where the technology's attributes both bolstered and strained the system's stability and trustworthiness. Issues such as network stability and scalability persist in Web 3.0 and blockchain, with network congestion leading to delays and increased transaction costs, sometimes resulting in temporary outages. Another concern is the unpredictability of transactions, as variations in network traffic and gas fees can cause significant delays or transaction failures, potentially resulting in the loss of funds. Despite advancements in security, the complexity of blockchain technology introduces new challenges, including the risk of security breaches with severe consequences. While smart contracts ensure accurate execution of agreements and processes, providing a high level of trust and predictability in transactions, they also present challenges if designed with security vulnerabilities.

7.6 Web 3.0 Along with Web 2.0

Standalone Web 3.0 may face adoption challenges due to the need for users to transition from familiar Web 2.0 platforms and habits to new decentralized alternatives. Integrating Web 3.0 features and technologies into existing Web 2.0 platforms offers a gradual transition for users while harnessing the benefits of decentralization. This hybrid approach enables the incorporation of blockchain-based features such as tokenization, decentralized identity, and smart contracts into familiar Web 2.0 applications and services. By seamlessly integrating Web 3.0 functionalities with Web 2.0 infrastructure, one can facilitate user adoption and drive mainstream acceptance of decentralized technologies. However, it may also pose challenges in terms of technical complexity, interoperability, and maintaining user trust and privacy.

7.7 Can Web 3.0 Become an Everyday Technology!

Despite considerable hype surrounding its potential to revolutionize industries through decentralization and trustless systems, Web 3.0 technologies are still evolving and face challenges such as scalability and usability. Critics argue that current implementations fall short of their theoretical promise due to issues like transaction speeds and regulatory uncertainties, which ongoing development efforts may address over time. Moreover, Web 3.0 encounters significant hurdles in achieving greater accessibility and user-friendliness for the average internet user, with reliance on digital wallets. But, Web 3.0 has the potential to become an everyday technology. With its focus on user control, transparency, security, lower costs, and innovation, Web 3.0 technologies offer significant advantages over traditional web platforms. As they become more user-friendly and accessible, they are likely to seamlessly integrate into our daily lives, revolutionizing how we interact online, conduct transactions, and access services. And it can be said that Web 3.0 developers play a crucial role in making this as everyday technology!

8 SUMMARY

Web 3.0 software development presents unique challenges that must be navigated with precision and forward-thinking strategies. This paper shares experiences of developing number of web 3.0 applications, and presents key challenges encountered. Additionally, experience based architectural and technical best practices are presented.

Scalability continues to be a major challenge, especially for platforms like Ethereum, which struggle with increased transaction fees and delays during periods of high demand. This is compounded by user experience issues; the complexity of managing wallets, interacting with decentralized applications, and understanding blockchain operations can deter mainstream adoption.

Security is also paramount, as the decentralized nature of these applications exposes them to specific vulnerabilities in smart contracts which, if exploited, cannot easily be reversed. In addition, it typically leads to a negative impact on user sentiment - in the form of losing trust and becoming more hesitant to adopt these systems. It is therefore crucial to prioritize and apply strong security measures, like thorough audits and following established best practices in smart contract development, to reduce risks.

Interoperability between various blockchain systems is still limited, hindering user experience and functionality across different ecosystems.

Regulatory uncertainty further complicates development, due to changing legal environments, the deployment of DApps is impacted in terms of both location and process.

While the challenges are substantial, they are not impossible to overcome. To effectively address these challenges, developers should prioritize implementing scalable solutions such as layer 2 scaling, or sharding technologies to improve transaction speed and reduce costs. Improving the user interface and simplifying interactions within DApps will help demystify Web 3.0 for a broader audience, fostering wider adoption. Additionally, designing applications with interoperability in mind and employing protocols that facilitate cross-chain functionality will enhance utility and expand user reach. Engaging with community-driven development models or decentralized autonomous organizations can also play a pivotal role. These structures empower users by involving them in the decision-making processes, aligning with the decentralized ethos of Web 3.0 and potentially driving more robust and user-responsive development. Keeping up-to-date with regulatory developments will ensure compliance and adaptability in a rapidly changing digital environment.

In conclusion, understanding challenges and applying suitable strategies are essential for developers aiming to unlock the full potential of Web 3.0.

ACKNOWLEDGEMENTS

Gratitude is extended to the individuals listed below for their valuable contributions to validating the ideas through implementations, design, and analysis: Rokas Bliudzius, Martin Slind Hagen, Diderik Kramer, Jesper Hustad, Julsen Fredrik, John Fredrik Bendvold, Magnus Lutro Allison, Pedro Pablo Cardona, Leo Gimre, and Håkon Groven.

REFERENCES

Antonopoulos, A. and Wood, G. (2018). *Mastering Ethereum: Building Smart Contracts and DApps*. O'Reilly.

Bendvold, J., Allison, M., and Cardona, P. (2024). Tokentrivia: A multiplayer web3-based triviaspill game.

Bliudzius, R., Hagen, M., and Kramer, D. (2022). Decentralized identity - a mobile wallet and verification platform.

Burniske, C. and Tatar, J. (2017). *Cryptoassets: The Innovative Investor's Guide to Bitcoin and Beyond*. McGraw Hill LLC.

Cai, W., Wang, Z., Ernst, J., Hong, Z., and Feng, C. (2018). Decentralized applications: The blockchain-empowered software system. *IEEE Access*, 6.

Choo, K., Dehghantanha, A., and Parizi, R. (2020). *Blockchain Cybersecurity, Trust and Privacy*. Advances in Information Security. Springer International Publishing.

Compound (2024). The money market protocol. https://compound.finance/documents/Compound.Whitepaper.pdf. [Online; accessed 25-April-2024].

De Filippi, P. and Wright, A. (2018). *Blockchain and the Law: The Rule of Code*. Harvard University Press.

DeForest, E. R. (2023). *Solidity Programming Essentials*. Amazon Digital Services.

Drescher, D. (2017). *Blockchain Basics*. PApress.

Friesendorf, C. (2023). *Decentralized Finance (DeFi): How Decentralized Applications (dApps) Disrupt Banking*. Business Guides on the Go. Springer Nature Switzerland.

Gimre, L. (2023). Multiplayer game development using web 3.0 technologies.

Groven, H. (2023). Web 3.0: developing secure and robust multiplayer gaming platforms to ensure secure onboarding of new users.

Hustad, J. and Fredrik, J. (2021). Evaluation and implementation of digital identity ledger for blockchain systems.

Iyer, K. and Dannen, C. (2018). *Building Games with Ethereum Smart Contracts: Intermediate Projects for Solidity Developers*. Apress.

Lee, E. and Ghosh, S. (2018). *Enterprise Blockchain Development*. Consultants Network.

Manoj, P. R. (2018). *Ethereum Cookbook*. Packt Publishing.

Min, T., Wang, H., Guo, Y., and Cai, W. (2019). Blockchain games: A survey. In *2019 IEEE Conference on Games (CoG)*.

Naha, S. (2023). *Web3 Security: Use of Block Chain Technology to Build Secure Web Applications*. BookRix.

Raval, S. (2016). *Decentralized Applications: Harnessing Bitcoin's Blockchain Technology*. O'Reilly Media.

Sahin, F. (2023). Play to earn web 3.0: The future of gaming and marketing.

Sajid, T. (2023). *Ultimate Blockchain Security Handbook: Advanced Cybersecurity Techniques and Strategies for Risk Management, Threat Modeling, Pen Testing, and Smart Contract Defense for Blockchain*. Orange Education Pvt Limited.

Scott, M. (2017). *Ethereum: The Complete Guide to Understanding Ethereum*. CreateSpace Independent Publishing Platform.

Sharma, A. (2024). *Rust for Blockchain Application Development: Learn to build decentralized applications on popular blockchain technologies using Rust*. Packt Publishing.

Uniswap (2024). Uniswap. https://uniswap.org/whitepaper-v3.pdf. [Online; accessed 19-APril-2024].

Wöhrer, M. and Zdun, U. (2021). Architectural design decisions for blockchain-based applications. In *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*.

Zheng, G., Gao, L., Huang, L., and Guan, J. (2020). *Ethereum Smart Contract Development in Solidity*. Springer Nature Singapore.