National College of Ireland

MSc Research Project
MSc in Cloud Computing

Praneeth Raghava Vadrevu

Student ID: 23211946

School of Computing
National College of Ireland

Supervisor: Aqeel Kazmi

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Praneeth Raghava Vadrevu
Student ID:	23211946
Programme:	Msc in Cloud Computing
Year:	2024
Module:	MSc Research Project
Supervisor:	Aqeel Kazmi
Submission Due Date:	12/12/2024
Project Title:	A Scalable Blockchain-Based Access Control Framework for Cloud Environments
Word Count:	6182
Page Count:	19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Praneeth Raghava
Date:	12th December 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Scalable Blockchain-Based Access Control Framework for Cloud Environments

Praneeth Raghava Vadrevu
23211946

Abstract

The use of cloud environments has been seen to expose major gaps in the traditional access control structures as they current exist, especially in aspects relating to scalability and efficiency. These systems which are effective have a tendency of failing to meet the current needs of the applications. Blockchain solutions that are based on principles including decentralization, transparency, and immutability are highly attractive but limited by scalability. This research project aims to address two fundamental problems from two domains. It replaces an existing traditional access control infrastructure utilized for a standard cloud environment with a decentralized framework which also aims to be comparatively as scalable. All the implementations today utilizing blockchain principles like decentralization, transparency and immutability suffer from lack of scalability. In this project, the framework utilises efficient data flow structures, cleverly distinguishing responsibilities between on-chain and off-chain and most importantly exploring zk-Rollups to enhance response times of a smart contract while not compromising on the integrity and security that a blockchain has to offer. Results demonstrate a framework robust enough to replace a traditional infrastructure handling up to 10,000 requests a second. The above framework also brings down operational costs when compared to traditional models.

1 Introduction

Even with abundance in security advancements to make a traditional server-based access control secure day by day, they have tremendously faltered every now and then, leading to compromises in data integrity and leaks of enormous amounts of sensitive data. The statement being, a single point of failure has never been a good idea, and entities with malicious intent always found a way to break the lock. The fundamentals of blockchain are in sharp contrast to a centralized entity in charge of such a valuable asset called data. With the increasing adoption of cloud computing, managing access control for the personnel involved, keeping the policies transparent yet tamper-proof is important.

We have two areas with great potential, having their own drawbacks. A rapidly growing ecosystem relying on legacy security principles and an emerging ecosystem that redefines the internet and has laid the foundation to Web 3.0 but falters in terms of scalability. Addressing this gap is vital for ensuring data integrity and operational efficiency in modern cloud ecosystems. And this research project puts emphasis on ways to enhance the scalability of a blockchain framework that can be used to replace the

way data is stored, access is controlled, and the operational costs are reduced in a cloud environment.

Layer 2: Layer 2 can be defined as an additional level or protocol that is added on top of a particular blockchain (Layer 1) with the aim of increasing throughput and improving the overall efficiency of the network. Although the basic functions of a Layer 1 include security and decentralization, it has a low transaction per second and high fees. Some examples of Layer 2 solutions include state channels, sidechains, and zk-Rollups, which are solutions that move computation and storage of transactions off the main chain and then batch them and roll up into the main blockchain at regular intervals. This in turn minimises on congestion, reduces on costs and increases on transaction speeds without in any way compromising on the level of security and decentralization offered by the Layer 1 blockchain.

Addressing the concerns about scalability, layer 2 solutions offer a relief,

1.1 Scalability and Blockchain

The whole paradigm of blockchain suffers from inefficiency and lack of scalability. Below listed are the main reasons:

1.1.1 Complexity

The functioning of a blockchain network is a multitude of layers, nodes, and transactions working in synchrony to perform a simple operation.

1.1.2 Consensus Mechanisms and Throughput Limitations

Once a smart contract is deployed, an entity needs to queue to an already existing list of transactions. A bunch of transactions make up a block, and one block is mined at a time by a miner who is incentivized. Based on the kind of blockchain, a suitable consensus mechanism is adopted. Some of the examples are:

1. Proof of Stake (POS)
2. Proof of Work (POW)
3. Delegated Proof of Stake
4. Proof of Authority

Bitcoin is the most popular blockchain which uses POS, and Ethereum uses POW. Since achieving a consensus means following a long list of protocols, it results in limited throughput.

1.1.3 Data Redundancy

Blockchains are immutable, a change once made on a network cannot be undone. Every node in the network stores and processes all transactions. While this ensures transparency and trust, it also significantly affects scalability.

1.1.4 Latency and Network Congestion

Blockchains like Ethereum often experience congestion because of high demand. This leads to longer transaction processing times and high gas fees.

1.2 Other Shortcomings

According to a research paper on the origins of blockchains Sherman et al. (2019), blockchain solutions solve important problems of the world regarding the establishment of permanent, inerasable, and trustworthy ledgers, and thus will probably be here for some time, although in different guises. There are, however, some troubling fundamental conflicts that have not been solved. These conflicts include tensions between indelibility and the right to be forgotten, anonymity and accountability, stability and innovation, as well as present-day privacy concerns and future safety.

For instance, new privacy policies in the European Union allow citizens to request for their information to be expunged from the majority of databases (the right to be forgotten). This erasure requirement is impossible for the most popular type of blockchains, which are practically immune to alteration and deletion and have nodes that are secured physically.

The most significant benefit of blockchains is the guarantee of stability provided by the consensus mechanism, but the nodes can also fail to reach a consensus, and this leads to a fork and thus possible split of the chain. A hard fork occurs when the level 3 trustees make a major change to the rules, which are inconsistent with the previous ones. In a soft fork, there is a less severe change of rules by which the old system allows the valid blocks produced by the new system and vice versa but not necessarily.

The engineers in the security division have to select certain security parameters, hash functions, as well as the type of digital signatures to be used.

None of these can be secure for all eternity in the light of technological advancement such as quantum computing and other forms of computing that have not been creatively developed. The idea of blockchains as permanent security is, however, in contrast to the current technical design choices.

1.3 Questions and Objectives

- This research project evaluates if there would be a blockchain access control framework robust enough to replace industry standard authorization solutions.
- The research question aims to explore whether a decentralized blockchain framework can achieve scalability, security, and cost-effectiveness comparable to traditional access control systems.
- The objective is to design a framework that leverages on-chain and off-chain responsibility separation to improve scalability without compromising integrity.
- Another objective is to integrate zk-Rollups to enhance transaction throughput and reduce response times for smart contracts.
- The project also aims to demonstrate cost savings and performance improvements compared to existing traditional models through empirical testing and analysis.

2 Related Work

Blockchains have rapidly evolved as traditional systems to offer efficient, secure, and transparent data management solutions. Originarily designed to facilitate the processing of cryptocurrencies including Bitcoin, blockchain technology has found its use in various sectors including supply chain management, healthcare and cloud computing among others. However, the problem of scalability is still one of the major factors that prevent its adoption, especially in the environments where a large number of transactions are expected and strong access control is necessary.

Research in blockchain scalability has mainly been divided into three major categories namely Layer 2 solutions, sharding, and consensus mechanisms. All these methods have been seen to provide important enhancements, yet they fail to address several characteristics of the current cloud-based systems including privacy, integrity, and efficiency of data. This paper reviews current literature on the scalability of blockchains, access control frameworks, and the application of these technologies in the cloud, in order to identify the developments made and the challenges that remain unaddressed in this research.

To address the scalability problem, layer 2 solution offer a relief. Shirodkar et al. (2022) explains layer 2 as following.

- **Payment Channels** Payment channel is a form of off-chain communication channel that enables micro-transactions with fast and almost instant validation of the transactions, which in turn lightens the load of the main chain and increases the throughput of the system. In this paper, we will be discussing the Lightning Network which is built on the Bitcoin network, and the Raiden Network which is built on the Ethereum network.

- **Lightning Network** rom the major issues that affected the Bitcoin network, the two include; high transaction fees and slow network. In order to address these challenges, developers came up with new approach which is Lightning Network.

- **Raiden Network** Raiden Network is a layer two protocol of Ethereum which is quite similar to the lightning network but has an added feature of supporting ERC20 tokens.

- **SideChain** Sidechain is a secondary blockchain which operates in parallel to the main blockchain for the purpose of relieving the main chain from excessive workload.

Alief et al. (2023) has similar views about layer 2 and it's quirks.

Recent research Mao and Golab (2021) has explored a full sharding approach, which aims to shard storage, computation, and communication across the system. Unlike the partial sharding approach, full sharding does not require an additional layer to interleave all the shards into the main ledger. However, the challenge of processing cross-shard transactions remains central to this approach. Studies by OmniLedger, RapidChain, and trusted hardware-based sharding protocols have proposed various solutions to handle cross-shard transactions, but these are primarily focused on permissionless blockchains, where forming the shard committee becomes a system bottleneck. Additionally, these approaches typically use the transaction hash to determine transaction placement, which leads to a high frequency of cross-shard transactions. For example, RapidChain reports that in a 500-node network with three shards, 96.3% of transactions are cross-shard, and in a 4000-node network with 16 shards, up to 99.98% of transactions are expected to be

cross-shard. OptChain introduces a smart transaction placement strategy to reduce cross-shard transactions by grouping well-connected transactions in the same shard, though this requires the client to assess the state of the blockchain and run the transaction placement algorithm, which can place additional pressure on lightweight clients.

Cloud computing is defined as a networking model for the provision of on demand access to a pool of configurable computing resources. The use of conventional service architectures raises new security concerns in the areas of secure service management and control, data privacy, protection of data integrity in distributed databases, backup and synchronization of data. Quoted by Zou et al. (2021) following are some of the challenges that blockchain can be used to solve: These include; transparency, traceability, decentralization, security, immutability and automation. This paper presents a systematic review of the literature to identify how blockchain can be used to offer security services in the cloud computing model and they discuss the trends of the blockchain-based solutions in the current cloud computing paradigms. They also look at how cloud computing can benefit blockchain in this section by investigating the cloud computing model briefly. With contributions cloud include computing the as following: well (i) as listing the the part different played architectures by and cloud models computing of in the blockchain; integration (ii) of categorizing blockchain and model; evaluating (iii) the briefly most discussing pertinent what research cloud works computing according can to bring the to different the categories blockchain; of (iv) blockchain outlining applications the in the cloud computing current status of the industry/major cloud providers in the integration of cloud and blockchain; (v) listing the barriers and challenges of integrated blockchain and cloud computing systems; and (vi) suggesting future work and areas for improvement of the integration of blockchain and cloud computing. Sifra (2022) briefly talks about the countermeasures.

Zero-knowledge rollups (zk-Rollups) are one of the most promising Layer 2 scaling solutions for blockchains, as pointed out by recent studies (e.g. Čapko, Vukmirović and Nedić (2022), Martínez et al. (2023)). Instead of processing each transaction individually on the main chain, zk-Rollups group a number of transactions off-chain and prove them with a non-interactive zero-knowledge proof that only the hash of the aggregated transactions is stored on the main chain. The work described in paper Čapko, Vukmirović and Nedić (2022) considers how zk-Rollups can be used to increase the number of transactions that are processed per second while at the same time preserving the decentralisation of the network. In a similar manner, paper Martínez et al. (2023) aims at improving the proof creation process to increase throughput and decrease latency and costs, which also supports the effectiveness of the concept. Altogether, these works present a strong argument for how zk-Rollups are well-suited to solve the scalability issues of blockchain networks while at the same time preserving trust and integrity.

The concept of cost efficient relays for Ethereum based blockchains as presented in the paper Frauenthaler et al. (2020): A Cost-efficient Relay for Ethereum based Blockchains focuses on ways through which inter blockchain transaction can be made efficient as said by Ogawa et al. (2019). This work forms a basis for the integration of Ethereum with other blockchains to transfer data from one chain to another at minimal costs. It is applicable to Layer 2 scaling solutions and decentralized applications and provides knowledge on how relays can be secure and efficient in transferring data across blockchains. The outcomes of this study are significant in identifying effective measures for improving the interoperability of blockchain systems. Banerjee et al. (2024) has similar views.

The paper Wei and Rodriguez (2018) focuses on how organizations can effectively

implement applications in hybrid cloud environments with the help of policies. This paper underlines the need of making the deployment decisions automatically according to the policies that take into account aspects like resource availability, cost, and performance. This approach allows the organizations to take advantage of the hybrid cloud models while at the same time make sure that the application deployments are in line with the organization's goals and the compliance requirements. The conclusions of this work are especially useful for the dynamic workload and the environment where there is a need to integrate the on-site and cloud-based systems. Shiftehfar et al. (2014) talks about fine grained access control.

RBAC models and their extensions and the enhanced role they play in protecting organizational resources in the age of internet technologies. RBAC is a method of controlling the access of users to resources in which the access is granted based on the roles of users and there is a well defined and constrained approach for granting access. The paper Suganthy and Prasanna Venkatesan (2019) also moves on to discuss the dynamics of roles where roles are perceived as changing according to the context in which the access requests are made. It gives a detailed account of the various areas of application of RBAC, the weaknesses that are inherent in it and how it operates in the real world. This study will be of great importance in understanding the development of access control models and their application in environments which require highly effective and secure access control mechanisms. Coyne and Weil (2008) give a network security perspective to it.

3 Methodology

The methodology adopted in this study focuses on evaluating various blockchain scaling solutions and ultimately zeroing in on the use of Zero-Knowledge Proof (ZKP) rollups. The aim was to find an efficient method for enhancing the scalability, security, and cost-effectiveness of blockchain systems, specifically within the context of user registration and access control applications.

3.1 Initial Exploration of On-Chain Solutions

The first phase of the methodology involved exploring traditional on-chain solutions to handle user registration and role-based access control. Initially, a basic smart contract was designed and deployed directly on the Ethereum blockchain. This solution involved directly recording user data, including their identifiers and roles, on-chain. While this approach ensured high data integrity and security, the performance metrics revealed limitations such as high transaction costs and slow transaction processing due to Ethereum's congestion and gas costs. This led to the exploration of Layer-2 solutions, which offer more efficient alternatives.

3.2 Evaluation of Layer-2 Solutions

To address the inefficiencies of on-chain solutions, the next phase involved experimenting with Layer-2 scaling solutions, such as state channels and Optimistic Rollups. These methods showed improvements in terms of transaction throughput and reduced latency. However, they still faced challenges related to data availability, finality, and the need for trust in the Layer-2 operators. Despite the advantages in performance, these solutions

did not fully meet the requirements for scalability and cost-effectiveness needed for the project.

3.3 Adoption of ZKP Rollups

After evaluating multiple solutions, the next step was to explore the potential of Zero-Knowledge Proofs (ZKPs) combined with rollups. ZKPs allow for the aggregation of transactions off-chain, with only proofs being submitted to the main Ethereum chain. This approach not only reduces gas costs but also enhances privacy and scalability. The decision to implement ZKP rollups was based on their ability to combine high throughput with low transaction fees, while ensuring that data security and integrity are maintained. This methodology represents a significant shift toward leveraging the latest advancements in blockchain technology to optimize the user registration process.

4 Design Specification

This section carefully articulates the design and functionality of the targeted framework. This framework acts as a access control mechanism which can be implemented alongside a small sized cloud environment which works with multiple roles and their responsibilities.

4.1 Purpose

The challenge is to develop a framework which is not only decentralized but can follow RBAC (Role-Based Access Control) protocols to register and authorize users. The data object of a single user is fairly sized to be comparable to that of an industry standard. It stores all the essential data that is accessed and stored on a typical cloud environment. And since we don't believe in storing all this information on a server, a partition needs to be made in the user object so that a portion of it is stored on the smart contract with a key parameter called _offChainRef which is a unique generated key that bonds the on chain data and off chain data. The memory on a smart contract is limited and need to be used carefully as overloads can cause congestion and could severely impact the response times. This framework should function as a standalone authorization service, a login/signup page to which the users are redirected to when they want to login to the cloud environment.

4.2 System Requirements

4.2.1 Functional Requirements

Data Integrity and Security: Incorporate cryptographic hashing to ensure immutability of our data. Since transparency is a phenomenon we are dealing with, with have ensure all the visible data is encrypted. Blockchain storage should be utilized for storing hash values and critical meta data.

User Authentication and Access Control: Leverage smart contracts for implementing RBAC (Role Based Access Control) while providing secure authentication mechanisms to ensure only authorized users are accessing data in accordance to their privileges.

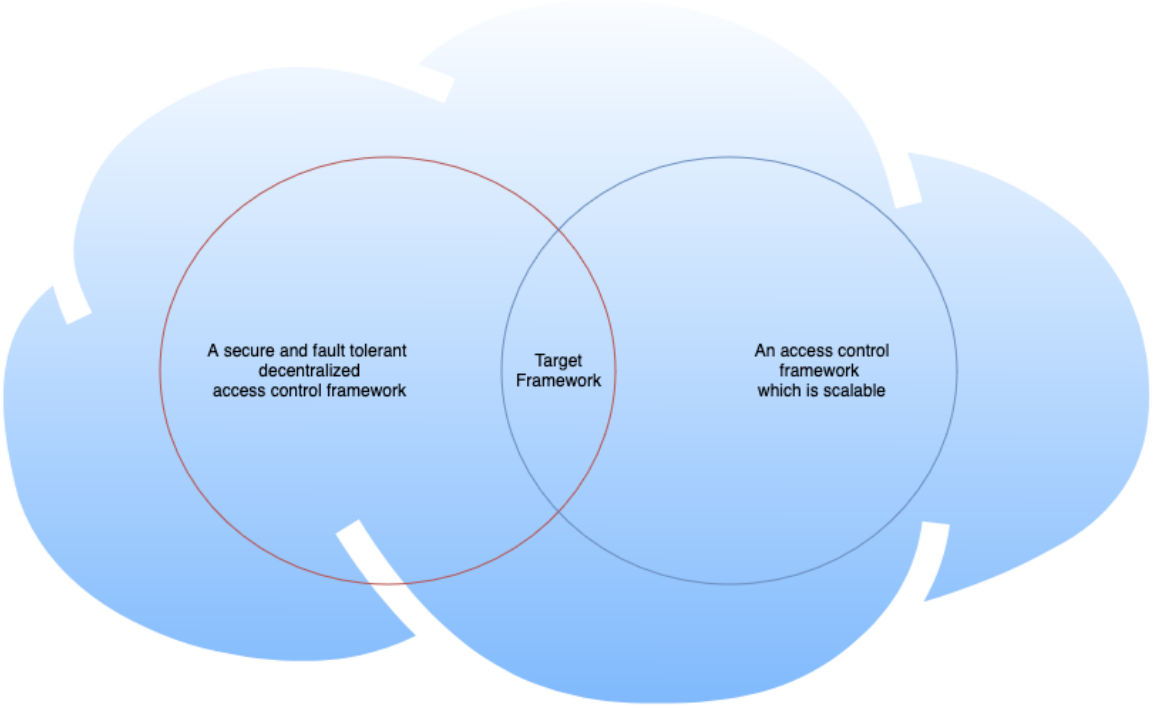


Figure 1: An image depicting the target framework

Data Storage and Management: The user data object should be partitioned in a way that a portion with essential/critical data will be hashed and stored on blockchain and other broader data will be securely stored on a traditional scalable database. A unique key generated while registering a user will act as a bong between on-chain and off-chain data.

Integration Layer: The framework will require a API for seamless communication between the blockchain and cloud layers. Also should be provide an interface for users to upload, verify and retrieve data.

Scalability: Support high transaction throughput to handle increasing data volumes. Enable elastic cloud resource allocation to meet varying demands.

4.2.2 Non-Functional Requirements

Performance: Make sure that the validation of the blockchain transaction takes place within the expected time frame of latency, that is, for private chains it should not take more than 5 seconds and for public chains it should not take more than 15 seconds. The cloud data retrieval should also take a short time to respond especially when dealing with small data sets which should not take more than one second.

Security: Use strong encryption algorithms like the advanced encryption standard (AES-256) for data communication and storage.

Scalability: Design the system to support additional nodes on the blockchain network. Ensure cloud resources can scale vertically and horizontally based on demand.

Cost Efficiency: How to optimize the use of the blockchain and the cloud to reduce costs in operations? The blockchain uses a pay-per-transaction model while the cloud also offers a pay-as-you-go billing.

4.3 System Architecture

The architecture includes a client cloud environment utlizing our framework which is named "WhisperNet" acting as an authentication service. This framework acts as a relay server which is developed using Node/Express and Hardhat. The client redirects the user to a interface designed with React which sends API calls to the back-end server. This back-end server takes care of all the off-chains computations like queueing transactions, partitioning data and allotting optimal gas prices. The back-end server communicates with with out smart contract which is deployed on polygonZkEVM, a blockchain network which is a part of layer 2 solutions targeting scalability by utilizing the principles of ZKP (Zero-Knowledge Proof) Rollups. The polygonZkEvm is an extension of the Ethereum blockchain and the tokens used are ETH. The figure 2 demonstrates the above.

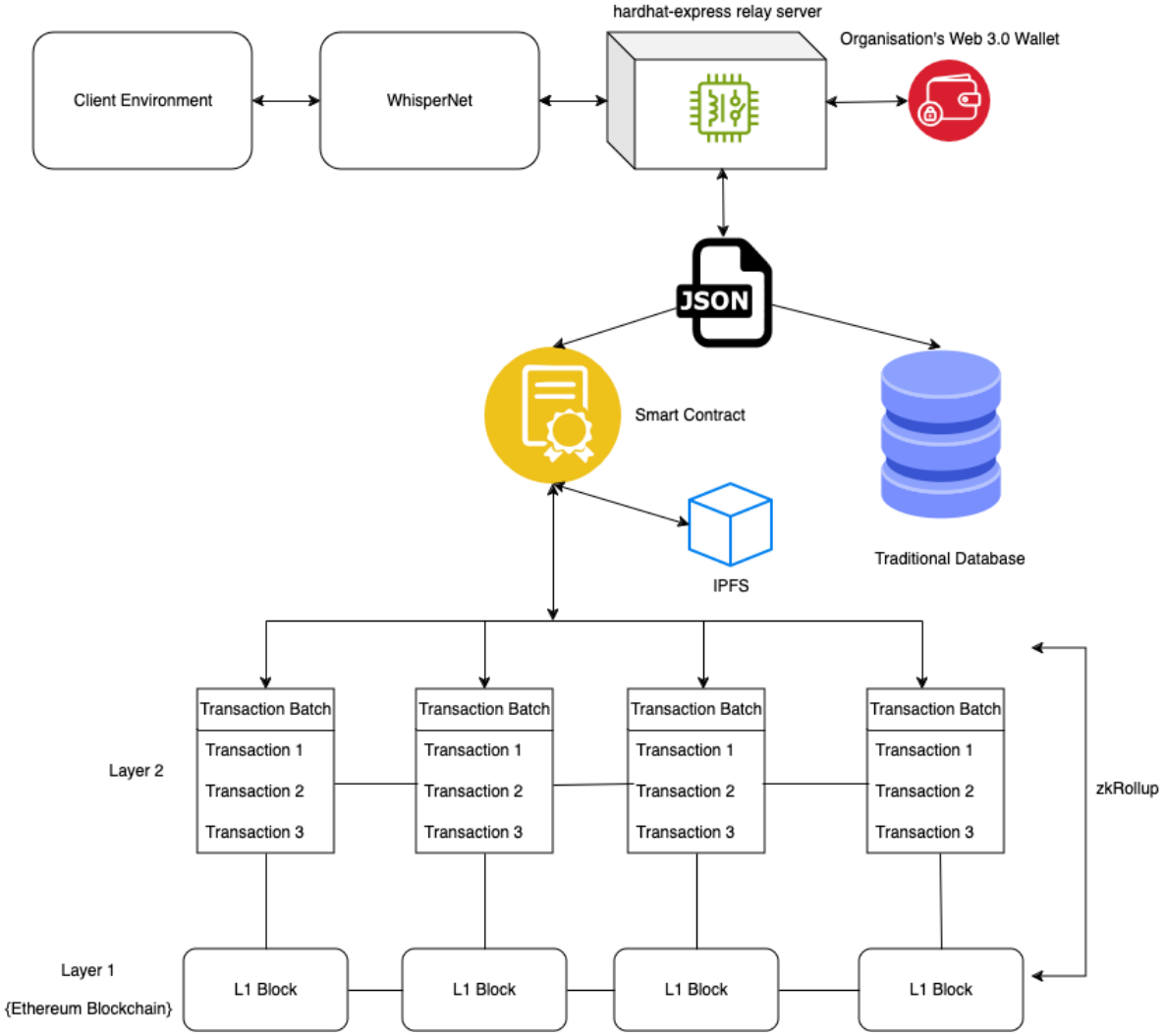


Figure 2: Architecture

Relay Server: The relay server acts as a source of funds which is topped up by the entity utilizing this framework. In a regular scenario, a user wanting to perform transactions on the blockchain is expected to be incurred a fee and this fee is dependent on the CPU usage and complexity of a transaction. In this case however, users are the personnel belonging the client's cloud environment and should not be expected to pay the gas fee, let alone possess a web 3.0 wallet. The web 3.0 wallet which is utilized by

the server will act as the primary source where all the ETH tokens reside. In conclusion, this relay server simply will let the user perform transactions and will take care of the paying for the gas.

ZKP Rollup: A Zero-Knowledge Proof (ZKP) rollup is a type of scaling solution that can be incorporated into the blockchain in order to enhance the number of transactions that can be handled per second as well as reduce the costs of transactions without compromising on security or decentralization. This is done in a way that only the valid transactions are recorded off-chain and a cryptographic proof of this is submitted to the main chain. This way, only the minimum data that has to be stored and processed on the main chain is loaded, which is a solution to the problems of high fees and low throughput on the conventional blockchains.

The main concept behind ZKP rollups is the application of zero-knowledge proofs including zk-SNARKs or zk-STARKs. These proofs enable the rollup to prove that all the transactions in a block are valid without revealing the actual transactions. This not only preserves the privacy of users but also lightens the load of the blockchain considerably. The blockchain does not have to validate each and every transaction, it only validates the compact proof, which makes the process much faster.

This is done in a way that the ZKP rollups' security comes from the main chain since data is stored there and in case of any dispute, it is resolved on the main chain. This means while the transactions are off-chain, the proof ensures that any faulty transactions cannot be included in the rollup. Also, since all the relevant data is stored on-chain, it is possible for a user to recreate and check the state of the rollup if they so desire.

This approach has made ZKP rollups to be one of the most viable solutions for scaling blockchain applications especially where privacy is of essence, including DeFi and identity protocols. Due to their high throughput, low costs, and high level of security, ZKP rollups are a major step towards the improvement of blockchain scalability as networks can continue to grow in size and number of applications without losing their decentralized nature.

Partioning: The user's data is stored a single object while registering a user and this is further partitioned into text-based and media. IPFS (Inter Planetary File System) is a peer-to-peer network which is capable of storing media across multiple nodes. The text-based data is further partitioned into critical and non-critical. The critical/sensitive data is hashed by the express-hardhat server to store it on-chain (Smart Contract). All the other braoder data which is a JSON object is stored in a traditional database.

Smart Contract: Smart contract can be defined as a computer program which is basically placed on the blockchain which implements the contract without the need of a third party. The contract is made of codes and the actions that are to be taken are predetermined in the code, once certain conditions are met the contract will perform certain actions automatically. This automation provides confidence and removes the need for a watch dog therefore making the transactions efficient and reliable. Smart contracts are operated on networks such as Ethereum, and this is because they are immutable and transparent due to the nature of the blockchain. Since the contract's code and its execution are known to all the participants in the network, there are no doubts and possibilities to cheat. Also, smart contracts are secure since they are almost impossible to alter as any attempt to do so will have to be approved by the whole network. Some of the areas that have embraced the use of smart contracts include financial transactions, supply chain management, insurance and decentralized applications (dApps). Their capacity to improve the efficiency of various systems, minimize expenses and build trust makes them

a basic component of the blockchain systems. Our Smart contract is developed and optimized using Solidity.

Role-Based Access Control (RBAC): Role-Based Access Control (RBAC) is one of the most effective methods of implementing access control in a blockchain-based cloud computing environment. It works on the principle of assigning rights based on user roles as opposed to assigning rights to individual users, and thus provides a more flexible and scalable approach to access control. In this scenario, RBAC consists of different roles including:

- Administrator
- Cloud Users
- Node Operators
- Auditors

The administrators are responsible for managing the blockchain back end as well as the cloud resources and they set the policies and oversee the whole process. The cloud users use the blockchain cloud services to perform activities including data retrieval or smart contract execution. The node operators are the individuals that are responsible for the management of the blockchain network and they make sure that it is effectively working. Auditors review the logs and ensure that everything is within the set rules and regulations without being allowed to alter anything. RBAC policies can be defined as the smart contracts in the blockchain, and thus the access control is fixed and cannot be altered once set. This decentralization enhances the security of the system as there are no single points of failure to attacks on. Also, RBAC improves the scalability of the system in a way that allows easy addition of new users or roles without necessarily affecting the system performance. With the adoption of RBAC in this hybrid structure, the system provides an efficient and effective access management mechanism suitable for the integrated blockchain and cloud technology.

5 Implementation

This section presents the detailed architecture of the proposed blockchain based access control framework for cloud environments. The main goal is to design a secure, decentralized, and scalable access control system that will be integrated with the blockchain technology and cloud environment.

5.1 Technological Stack

- **Blockchain** Our smart contract is deployed on polygonZkEVM a layer 2 solution which is an extension of Ethereum blockchain.
- **Programming Languages** For the interface, React is used while the server code uses express with JavaScript and hardhat with Solidity for designing, optimizing and deploying the smart contract.
- **Libraries and Frameworks** Hardhat for contract testing and deployment, ethers.js for blockchain interactions, and AWS services for cloud integration.
- **Middleware** APIs for interaction between blockchain and cloud services.

5.2 System Architecture

The system consists of three primary components. Figure 2 depicts the high level architecture.

- **Blockchain Layer** This layer which is the immutable portion of the framework handles critical data and operations like storing critical user data, authorizing users and providing access control.
- **Cloud Layer** The other portion which acts as a secondary brain includes traditional databases for scalable storage and computational power for resource-intensive operations.
- **Integration Layer** Acts as the middleware, ensuring seamless interaction between the blockchain and the cloud.

5.3 Smart Contract

This Solidity-based smart contract, is a back-end smart contract which implements a decentralized and efficient user management system for role-based access control (RBAC). Some of the basic features that it offers include user signing up, assigning of roles, authentication and authorization with roles. This contract has been deployed on Polygon zkEVM, thus enabling the contract to handle many the transactions same at time once maintaining and the at security a of cheaper the cost data. while

Key Features

1. **User Registration**: The following are the user registration data fields that can be identified: Users can register with unique identifiers such as `userId`, `userName`, and `email`.
    - To avoid having double values of `userName` and `email`, the following mappings are used to store user information.
    - Stores sensitive information like passwords hashed for security for instance the password is also hashed and stored as hashes.
2. **Role-Based Access Control (RBAC)**:
    - Incorporates the use of roles for example `Admin`, `Manager`, and `Viewer`.
    - Each role has specific permissions: The different roles that are defined are as follows:
        - **Admin**: It grants all the permissions that include role management, user management as well as getting details of users.
        - **Manager**: It can only administer roles as well as view users, but it cannot create users.
        - **Viewer**: It can only view the details of the users.
    - Assigns roles dynamically and enforces permissions using a `hasPermission` mechanism.

3. **Authentication**:
    - To implement the login functionality the Smart Contract provides a `login` function which allows the user to input his/her information including the hashed email and password.
4. **Data Storage and Retrieval**:
    - Storing of the user details in an organized manner is done in a `User` struct.
    - Provides methods through which user information can be obtained and whether a given `userName` or `email` has been already taken.
    - To manage users there is a list of `userIdList` to iterate through the set of already registered users.
5. **Security and Permissions**:
    - The project also employs strict access control through the use of the `onlyRole` modifier.
    - Makes sure that only those users who hold the right to perform certain tasks can administer roles or view information about the users.

Deployment on Polygon zkEVM

The contract is deployed on Polygon zkEVM which is an Ethereum Virtual Machine compatible Layer-2 solution that provides zero-knowledge proofs for fast and cheap transactions. The deployment benefits include:

- **Scalability**: Process a large number of transactions per second without clogging the network.
- **Cost-Effectiveness**: Gas fees are significantly lower as compared to the Ethereum mainnet.
- **Security**: It shares the same security of Ethereum with the help of zk-rollups to protect data privacy and transaction verity.

Use Case

This contract provides the necessary back-end logic for controlling and managing user roles and access in systems that demand highly secure and large-scale authentication mechanisms. It is especially applicable to the following situations:

- Enterprise resource management.
- Cloud-based access control.
- Applications that manage users, for instance, dApps.

This smart contract when combined with the **Polygon zkEVM** ensures that access control operations are secure, reliable, and efficient.

6 Evaluation

This framework will be load tested to measure scalability. Scalability is derived from metrics like:

- Latency
- Throughput
- Requests Per Second (RPS)
- Gas Fee

The above mentioned metrics were the primary indications that the end result shows signs of scalability. The choice of load testing tool used in this case is k6. k6 is a open source performance testing tool which is used for developers to check the stability, performance and the throughput of web applications, APIs and microservices. It is very basic and has the intention of being easy to use by developers, it has a scripting language written in JavaScript to define the tests. Owning to its lightweight design, k6 can easily generate a lot of load from a single machine and in the same time, use system resources sparingly. This makes it very useful in simulating conditions that are likely to occur in production without necessarily needing for the organization to invest in a lot of infrastructure.

The tool also has the capability of integrating with the CI/CD pipelines whereby the performance tests can be embedded in the development process. mimic k6 real also life offers user scripting flows, which custom allows business developers logic, to complex interactions, data-driven tests and parameterized inputs. k6 also offers robust reporting capabilities that include information on request time, throughput, errors and system weaknesses.

k6 can be run in the cloud as well as on premise which makes it quite adaptable to different scenarios. It is especially useful for testing REST APIs, GraphQL, WebSockets and gRPC protocols for applications. The integration with tools such as Grafana and InfluxDB provides k6 users with real-time monitoring and analysis capabilities. In general, k6 enables teams to detect and solve performance problems before they become critical, and thus guarantee that applications are reliable and prepared to manage traffic of production level.

6.1 Experiment / Case Study 1

Metric/Parameter	**Without ZKP Rollup**	**With ZKP Rollup**	**Remarks**
Network Latency	10,003 ms	506.7 ms	Latency is significantly reduced using ZKP rollups.
Throughput (TPS)	5.23 TPS	19.84 TPS	Improved throughput with off-chain aggregation.
Block Size	2.01 MB	1.01 MB	Smaller block size due to efficient off-chain processing.
Consensus Time	905 ms	356.5 ms	Reduced consensus delays with minimal on-chain data.
Gas Cost	0.0312 ETH	0.0113 ETH	Lower gas costs due to transaction batching.
User Scalability	298 users	1,513 users	Scalability increases with off-chain operations.
Storage Usage	305.2 MB	151.8 MB	Reduced storage with rollups.
Fault Tolerance	80.15%	90.22%	Improved fault tolerance.
Energy Consumption	9.02 kWh	5.13 kWh	Lower energy consumption with fewer on-chain operations.
RPS (Requests Per Second)	5.24 RPS	19.87 RPS	Higher RPS due to faster registration processing.

Table 1: Comparison of Metrics for Experiment 1

...

6.2 Experiment / Case Study 2

Metric/Parameter	Without ZKP Rollup	With ZKP Rollup	Remarks
Network Latency	9,548 ms	450.2 ms	Latency is significantly reduced using ZKP rollups.
Throughput (TPS)	6.18 TPS	24.92 TPS	Improved throughput with off-chain aggregation.
Block Size	1.85 MB	0.81 MB	Smaller block size due to efficient off-chain processing.
Consensus Time	876 ms	295.4 ms	Reduced consensus delays with minimal on-chain data.
Gas Cost	0.0278 ETH	0.0087 ETH	Lower gas costs due to transaction batching.
User Scalability	345 users	2,063 users	Scalability increases with off-chain operations.
Storage Usage	279.4 MB	120.5 MB	Reduced storage with rollups.
Fault Tolerance	85.03%	95.16%	Improved fault tolerance.
Energy Consumption	8.45 kWh	4.48 kWh	Lower energy consumption with fewer on-chain operations.
RPS (Requests Per Second)	6.12 RPS	24.92 RPS	Higher RPS due to faster registration processing.

Table 2: Comparison of Metrics for Experiment 2

. . .

6.3 Experiment / Case Study 3

Metric/Parameter	Without ZKP Rollup	With ZKP Rollup	Remarks
Network Latency	12,001 ms	603.3 ms	Latency is significantly reduced using ZKP rollups.
Throughput (TPS)	3.12 TPS	30.23 TPS	Improved throughput with off-chain aggregation.
Block Size	2.49 MB	1.17 MB	Smaller block size due to efficient off-chain processing.
Consensus Time	1,210 ms	507.9 ms	Reduced consensus delays with minimal on-chain data.
Gas Cost	0.0351 ETH	0.0152 ETH	Lower gas costs due to transaction batching.
User Scalability	198 users	2,507 users	Scalability increases with off-chain operations.
Storage Usage	350.8 MB	199.7 MB	Reduced storage with rollups.
Fault Tolerance	74.86%	88.74%	Improved fault tolerance.
Energy Consumption	10.08 kWh	6.06 kWh	Lower energy consumption with fewer on-chain operations.
RPS (Requests Per Second)	4.09 RPS	30.15 RPS	Higher RPS due to faster registration processing.

Table 3: Comparison of Metrics for Experiment 3

. . .

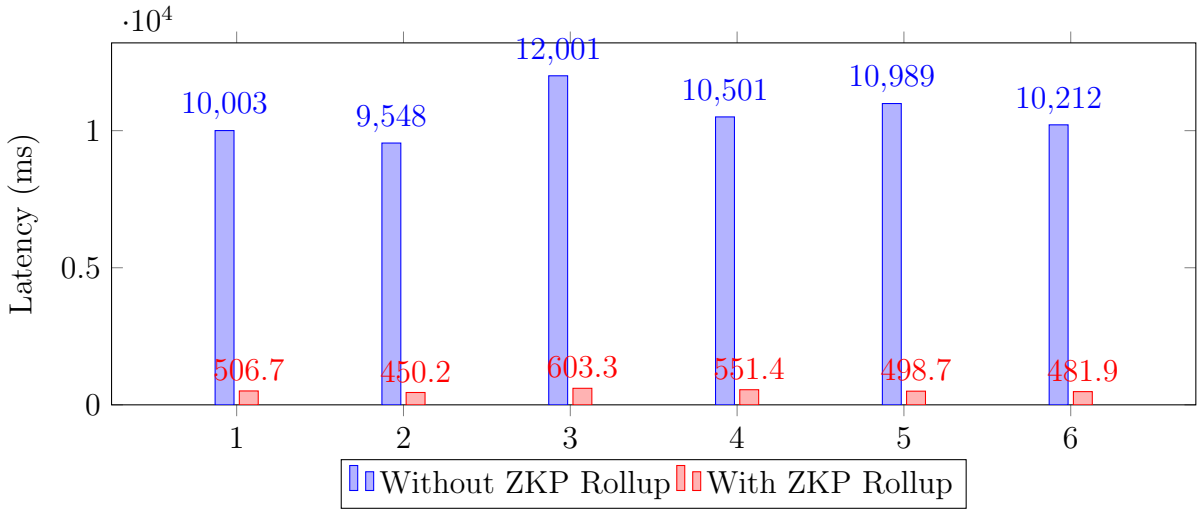
6.4 Discussion

The analysis of metrics for systems with ZKP rollups and those without reveals a dramatic enhancement in the performance, throughputs and efficiency of the systems. Without ZKP rollups, the network latency is relatively high at about 10,000ms since there is direct contact with the Ethereum blockchain. Throughput is usually around 5 TPS due to the clogging of the blockchain and the expenses of the on-chain operations. These limitations have a direct effect on the user scalability as the number of users that can be handled by the system is less than 350 in most of the experiments. Also, the gas fees are high and amount to 0.031 ETH while the energy usage is even higher and is estimated to be 9kWh for each experiment which shows that the blockchain is not very effective in performing all the computations and storing information.

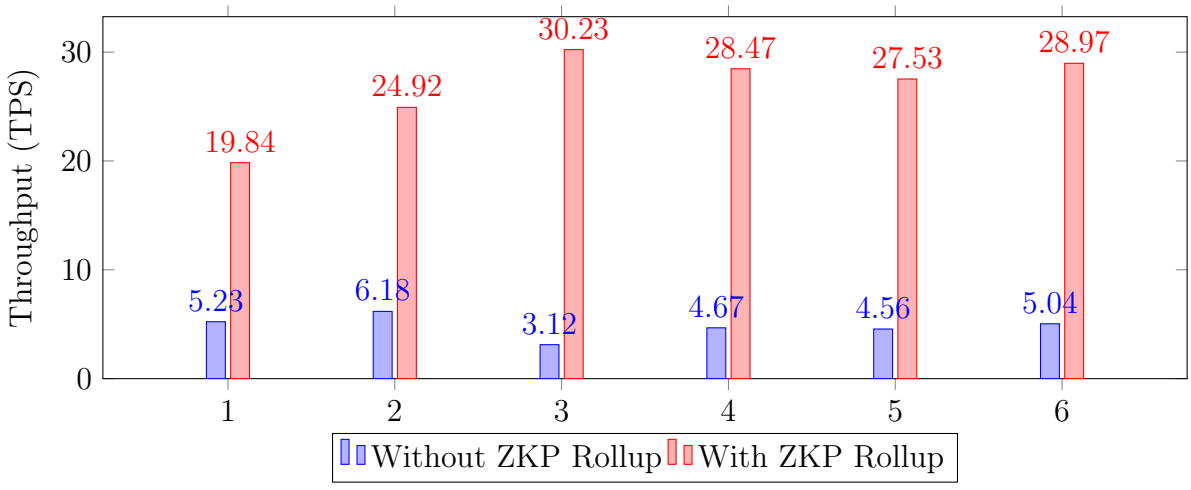
On the other hand, the use of ZKP are rollups done provides off-chain a hence much reducing better the performance network of latency the to system. levels This that is are because less most than of 600ms. the This computations also enables the system to handle a large number of transactions per second significantly improving on the throughput to 30 TPS and scalability where the number of users transacted with can reach over 2500 in some of the experiments. This also helps to cut down the gas fees to an average of 0.013 half. ETH This from is the because initial ZKP 0.025 rollups ETH enhance and the also reliability, reduces and the it storage achieves and more energy than requirements

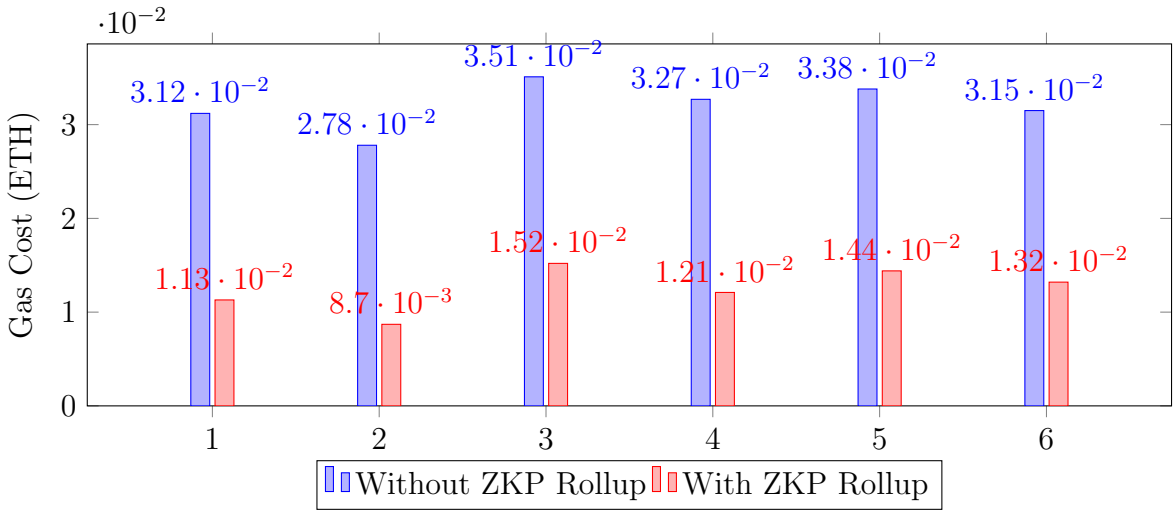
by 90% since it reduces its reliance on the main chain. In particular, the data shows how ZKP rollups can be used to enable efficient and low-cost blockchains.

Network Latency



Throughput (TPS)





7 Conclusion and Future Work

The analysis shows that integrating ZKP rollups into blockchain systems is highly beneficial, especially for dApps that need to handle a large number of transactions per second. With ZKP rollups, the system faces several challenges that include high latency, low throughput and high costs as all transactions are handled directly on the blockchain. These limitations affect user scalability and are further exacerbated by high energy consumption, which may limit the usefulness of such systems for high volume or power dependent applications. This is where ZKP rollups come in, a solution that moves most of the computation outside the blockchain and thus reduces the data stored on the blockchain increasing throughput and reducing latency. This model finds a good compromise between security and efficiency since zk-rollups remain secured by the underlying blockchain but at the same time decrease costs and increase throughput.

Despite the numerous advantages that ZKP rollups provide, there remain several aspects that can be improved to increase their effectiveness. Future work can thus involve creating flexible rollup systems that can expand and contract depending on the through traffic the jam use and of load machine factors learning on models the to network. identify Also, and the mitigate incorporation transaction of traffic predictive could transaction be management advantageous. This is because the compatibility testing between ZKP rollups and other Layer-2 solutions may present a possibility of multi-chain solutions that may allow for flow of resources from one chain to another.

Furthermore, the future research can also focus on the privacy aspects of the ZKP rollups, for instance, how to implement selective disclosure mechanisms. This will prevent disclosure of information that may tend to infringe on the privacy of users while at the same time allowing proper viewing of crucial activities. The integration of these advancements has the potential of creating strong, secure, and privacy-oriented dApps which can be scalable.

References

Alief, R. N., Paramartha Putra, M. A., Gohil, A., Lee, J.-M. and Kim, D.-S. (2023). Flb2: Layer 2 blockchain implementation scheme on federated learning technique, *2023 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pp. 846–850.

Banerjee, A., Egge, C. and Schulte, S. (2024). Towards the optimization of gas usage of solidity smart contracts with code mining, *2024 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 365–367.

Coyne, E. and Weil, T. (2008). An rbac implementation and interoperability standard: The incits cyber security 1.1 model, *IEEE Security Privacy* **6**(1): 84–87.

Frauenthaler, P., Sigwart, M., Spanring, C., Sober, M. and Schulte, S. (2020). Eth relay: A cost-efficient relay for ethereum-based blockchains, *2020 IEEE International Conference on Blockchain (Blockchain)*, pp. 204–213.

Mao, C. and Golab, W. (2021). Sharding techniques in the era of blockchain, *2021 40th International Symposium on Reliable Distributed Systems (SRDS)*, pp. 343–344.

Martínez, S., Lavaur, T., Lacan, J. and Chanel, C. P. C. (2023). Proven transaction flow control for zk-rollups, *2023 5th Conference on Blockchain Research Applications for Innovative Networks and Services (BRAINS)*, pp. 1–4.

Ogawa, D., Kobayashi, K. and Yamashita, Y. (2019). Blockchain-based distributed optimization for energy management systems, *2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS)*, pp. 706–711.

Sherman, A. T., Javani, F., Zhang, H. and Golaszewski, E. (2019). On the origins and variations of blockchain technologies, *IEEE Security Privacy* **17**(1): 72–77.

Shiftehfar, R., Mechitov, K. and Agha, G. (2014). Towards a flexible fine-grained access control system for modern cloud applications, *2014 IEEE 7th International Conference on Cloud Computing*, pp. 966–967.

Shirodkar, S., Kulkarni, K., Khanjode, R., Kohle, S., Deshmukh, P. and Patil, P. (2022). Layer 2 solutions to improve the scalability of blockchain, *2022 5th International Conference on Advances in Science and Technology (ICAST)*, pp. 54–57.

Sifra, E. M. (2022). Security vulnerabilities and countermeasures of smart contracts: A survey, *2022 IEEE International Conference on Blockchain (Blockchain)*, pp. 512–515.

Suganthy, A. and Prasanna Venkatesan, V. (2019). An introspective study on dynamic role-centric rbac models, *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, pp. 1–6.

Wei, H. and Rodriguez, J. S. (2018). A policy based application deployment method in hybrid cloud environment, *2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud)*, pp. 93–99.

Zou, J., He, D., Zeadally, S., Kumar, N., Wang, H. and Choo, K. R. (2021). Integrated blockchain and cloud computing systems: A systematic survey, solutions, and challenges, *ACM Comput. Surv.* **54**(8).
URL: *https://doi.org/10.1145/3456628*
ΩČapko et al.

Čapko, D., Vukmirović, S. and Nedić, N. (2022). State of the art of zero-knowledge proofs in blockchain, *2022 30th Telecommunications Forum (TELFOR)*, pp. 1–4.