ZONAsi
Jurnal Sistem Informasi
ZONAsi VOL. 7 NO. 2 Mei 2025
Page: 395 - 407
ISSN: 2656-7407 (Online)
2656-7393 (Print)

SMART CONTRACTS FOR DECENTRALIZED IDENTITY: EMPOWERING CITIZENS IN GOVERNMENTAL ECOSYSTEMS

Yustus Eko Oktian[1], Kei Patrick Hilton[1], Jeffri Lieca Handoyo[1]
([1]Program Studi Sistem Informasi, Fakultas Teknologi Informasi, Universitas Ciputra Surabaya)
[1]CitraLand CBD Boulevard, Made, Kec. Sambikerep, Kota Surabaya, 0317451699
e-mail: [1] yustus.oktian@ciputra.ac.id, [2]kpatrick@student.ciputra.ac.id,
[3]jhandoyo@student.ciputra.ac.id

Abstract

The rapid digitization of governmental services has underscored the urgent need for secure and efficient identity management systems, particularly within e-government ecosystems. Traditional centralized systems suffer from vulnerabilities such as data breaches, lack of user autonomy, and dependence on single points of control. This study proposes a decentralized identity system leveraging blockchain technology to address these limitations. Our framework employs factory-based smart contracts, hierarchical authority structures, and robust validation mechanisms to ensure trust, security, and interoperability among government entities. Through detailed implementation and testing, the system demonstrates its capability to enhance privacy, streamline identity verification, and maintain data integrity. This paper provides a practical, scalable solution for modern e-governance, setting a foundation for future research and adoption.

Keywords: *blockchain, smart contract, decentralized identity, e-governance.*

1. PRELIMINARY

The rapid advancement of digital technologies has transformed how individuals and organizations interact, creating an urgent need for efficient and secure identity management systems. Traditional centralized identity systems are increasingly seen as inadequate due to their vulnerability to security risks, including data breaches and unauthorized access, which expose sensitive personal information to misuse [1]. These systems often limit user control over personal data, relying heavily on central authorities for verification and updates. As governments digitize their operations, the demand for robust, secure, and efficient identity systems is particularly pressing in e-government scenarios, where trust, security, and interoperability are critical for seamless operations. In this context, the global expansion of digital interactions has underscored the necessity for secure, user-controlled, and reliable identity management solutions.

Blockchain technology [2] has emerged as a promising foundation for decentralized identity (DID) systems [3], offering inherent benefits such as decentralization, immutability, and transparency. Unlike traditional systems, blockchain enables entities to own and control their identities while maintaining a tamper-proof and verifiable ledger of transactions. This ensures that identity data remains secure and accessible only to authorized parties, reducing the risks of unauthorized access and fraud. Additionally, blockchain's support for smart contracts [4] facilitates automated workflows, streamlining identity verification processes and enhancing operational efficiency. Its suitability for e-government is particularly compelling, enabling seamless collaboration between citizens and government entities while ensuring that identity records are secure, interoperable, and accessible across departments.

Despite the growing interest in blockchain-based DID systems, significant research gaps persist. Many studies focus on isolated aspects of DID without addressing the need for comprehensive, integrated e-government solutions. For instance, Rodionov highlights the potential of self-sovereign identity (SSI) models to enhance privacy and user control but notes regulatory challenges that hinder widespread adoption [5]. Kersic et al. propose integrating on-chain and off-chain DID systems using universal digital wallets to bridge technological gaps and improve interoperability [6]. Thorbecke

demonstrates the use of blockchain in Swiss federal identity management, focusing on secure and verifiable citizen registries [7]. Meanwhile, Pleger and Guirguis examine public acceptance of government identity systems, revealing factors critical to e-government adoption [8]. These studies emphasize the potential of DID systems but often address either technical advancements or policy considerations in isolation, leaving a gap in holistic approaches tailored for specific e-government applications.

This paper aims to bridge this gap by proposing a DID system specifically designed for e-government scenarios. By leveraging blockchain technology, the system integrates factory-based smart contract designs, hierarchical authority management, and robust validation mechanisms to streamline identity verification and management processes for both citizens and government entities. This paper offers a comprehensive framework that ensures seamless interoperability across government departments while safeguarding user privacy and data integrity. Through detailed implementation methodologies and feasibility testing, this paper provides a scalable, practical, and secure solution that meets the unique requirements of e-government identity management.

The rest of this paper is organized as follows: Section 2 discusses the research methodology, detailing the system model and the inner workings of the proposed system. Section 3 presents the prototype and evaluation results, showcasing the user interface and feasibility analysis to validate the system's effectiveness. Finally, Section 4 concludes the paper and highlights potential avenues for future research.

2. RESEARCH METHODS

This section outlines the design and functionality of the proposed system. It begins with the problem statement, followed by a description of the actors involved and the data structures used. Finally, it explains the system's inner workings, including smart contract creation, identity management for citizens and government, and secure identity verification processes.

2.1. Problem Statement

Traditional identity management systems face significant challenges in ensuring trust, data integrity, and security. One major issue is the absence of a centralized mechanism to verify the authenticity of entities. Without a single root of trust, it is difficult to ensure that identities are genuine, leaving systems vulnerable to the creation of fake identities and unauthorized entities gaining access.

Data integrity is another critical challenge, as unauthorized modifications can compromise the reliability of stored information. Existing systems often lack stringent approval mechanisms and effective tracking of changes, making them susceptible to identity fraud and malicious tampering. This is particularly concerning for government-related data, where insufficient oversight increases the risk of unauthorized edits and misuse.

Identity verification processes in conventional systems often fail to ensure that only legitimate users can access or modify sensitive information. Without robust cryptographic protections, these systems are prone to vulnerabilities that can compromise the security of user identities.

Additionally, the fragmentation of login data across multiple government services presents a significant usability challenge for users. Individuals are often required to manage numerous accounts and credentials for verification, leading to increased complexity and a higher likelihood of errors or forgotten credentials. This fragmentation undermines the efficiency and user experience of identity systems.

To address these challenges, our system design incorporates a unified and secure framework emphasizing trust, data integrity, and usability. By employing a single root of trust through a factory-based smart contract creation model, the authenticity of all entities is guaranteed. Data integrity is preserved through a strict approval mechanism, ensuring that only authorized changes are allowed and transparently tracked on the blockchain. Identity verification is streamlined with a cryptographically secure Single Sign-On (SSO) mechanism using temporary One-Time Passwords (OTP), enhancing security and preventing unauthorized access. This unified approach also resolves the fragmentation of login data by providing a seamless and efficient verification process across multiple government services, improving user experience while maintaining robust security.

2.2. Actors

The following are the actors and entities present in our system:

- **Super Admin**: This actor holds the highest authority in the system and has the privilege to create smart contract templates on the blockchain. Other entities create smart contract objects based on these predefined templates. Ideally, the entity with the highest privilege in the country should assume the role of the Super Admin, as all deployed smart contracts are anchored to the main contract initiated by this entity.
- **Citizen**: Refers to individuals who reside in a country or district.
- **Government**: Refers to a single entity representing one department within the broader government structure.
- **Department Leader**: A person authorized to lead a specific government department.
- **Government Employee**: An individual working for the government and belonging to a particular department.
- **Kominfo**: Represents Indonesia's Ministry of Communication and Informatics, the government body responsible for managing and overseeing information technology (IT) systems across national and regional government entities. This is just a mere example and it can be changed to any department authorized to manage the operational aspects of information technology (IT) in a country or district.
- **Validation Seeker**: An entity or individual seeking verification of their originality or authenticity.
- **Prover**: An entity or individual aiming to validate the authenticity of another entity.

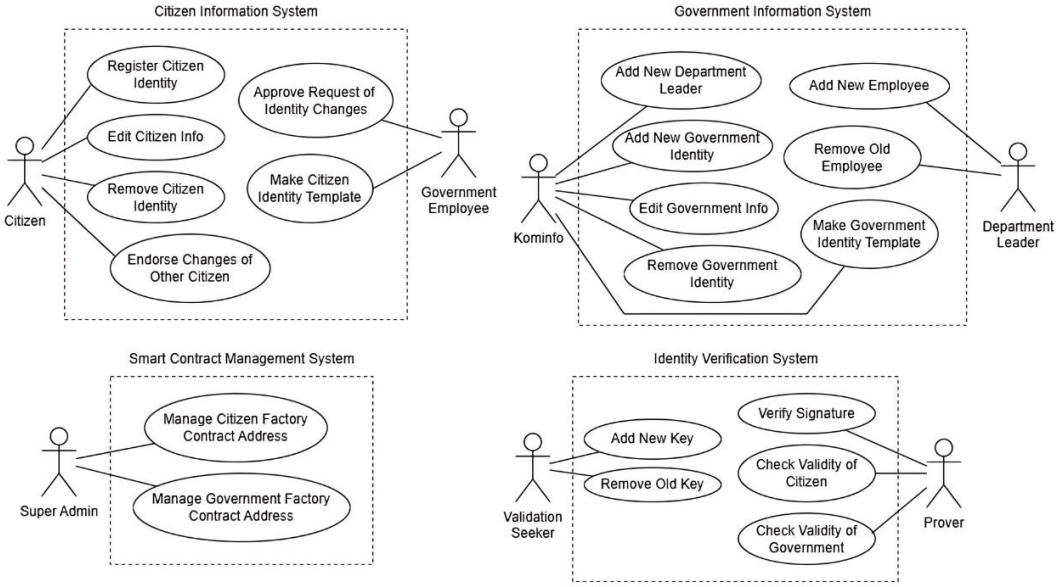

Image 1. Use case diagram of our proposed system

The use case diagram, shown in Image 1, summarizes the actions each actor can perform within the system. Our proposed system is divided into four subsystems, which integrate seamlessly to form a cohesive and efficient distributed identity management framework.

2.3. Data Structure

The class diagram, shown in Image 2, provides an overview of the data stored within our system. We implement six main smart contracts in total. Given the two primary entities in the system (i.e., the citizen and the government) we develop two factory smart contracts to standardize the creation of citizen and government objects. These factories are anchored to the **BaseFactory** smart contract, which serves as the single root of trust. This design ensures the authenticity of citizen and government objects by verifying that they originate from the **BaseFactory** instance. Additionally, both **Citizen** and

Government entities manage their private and public keys to establish and prove their identities. To support this functionality, both entities inherit the **Key** smart contract.

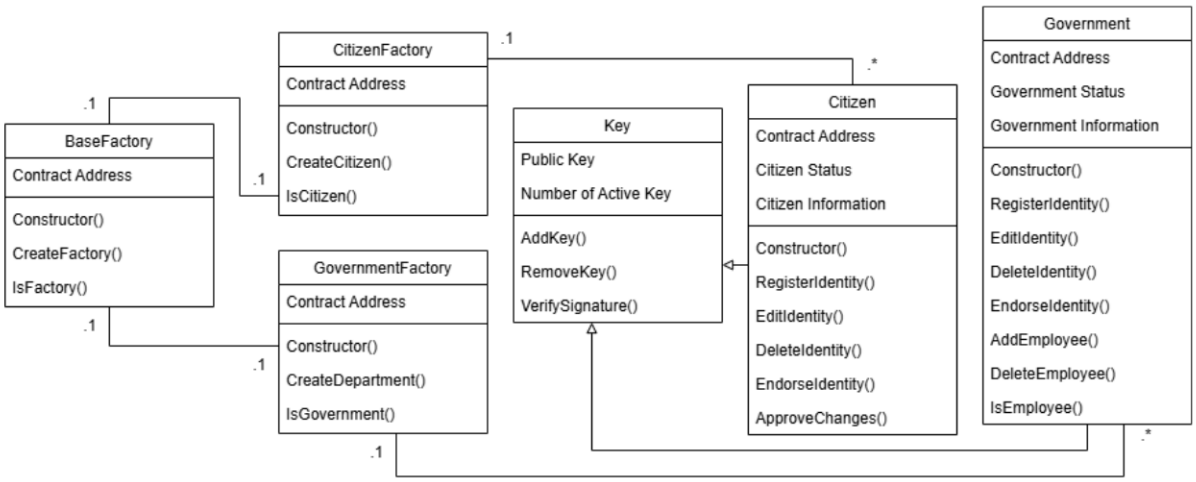

Image 2. Class diagram of our proposed system

2.4. Innerworking

Our proposed system operates through four core processes: smart contract creation, government identity management, citizen identity management, and identity validation and key management. Each process is designed to ensure security, trust, and seamless operation within the system. The following paragraphs provide detailed insights into how these processes function and interact to maintain the integrity of the decentralized identity framework.

2.4.1. Smart Contract Creation

The initial step, as highlighted in Image 3, involves the creation of smart contracts. In this process, the Super Admin deploys the **BaseFactory** instance on the blockchain. Following this, the Super Admin creates the **CitizenFactory** and **GovernmentFactory**, linking both factories to the **BaseFactory**. This linkage establishes the authenticity of the factories, ensuring they are rooted in the trusted **BaseFactory**. By design, the system prevents the inclusion of malicious or unauthorized factories, maintaining the integrity and trustworthiness of the entire framework.

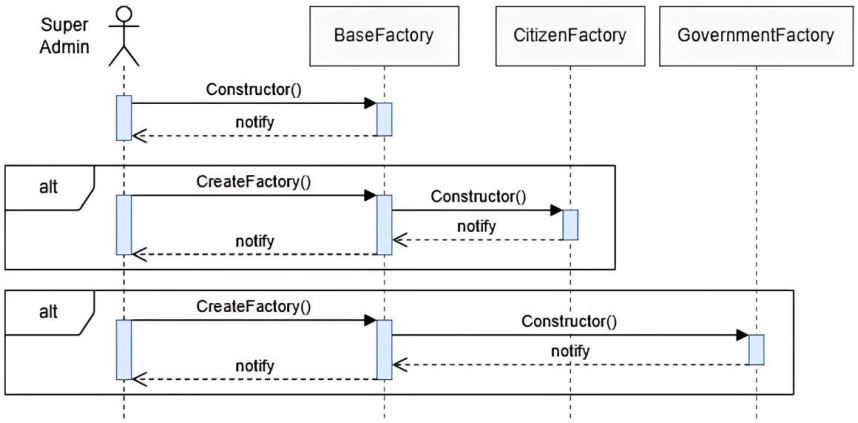

Image 3. Sequence diagram of creating citizen and government factory

Using the deployed **CitizenFactory**, government employees can create new citizen identities, as illustrated in Image 4. This process is carried out exclusively when new citizens are born and require registration. Once a citizen is registered, their validity status can be verified by anyone directly through the smart contract, ensuring transparency and trust in the system.

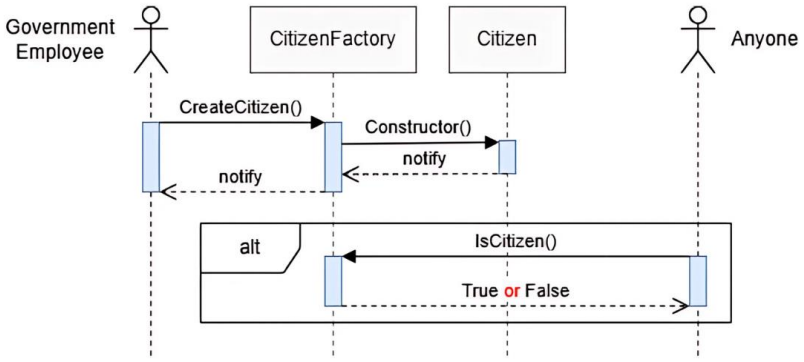

Image 4. Sequence diagram of initiating citizen identity

Similar to the citizen case, once the **GovernmentFactory** is deployed, Kominfo can create new government instances, as shown in Image 5. Each instance represents a specific department within the government. For example, the Department of Finance is one instance, while the Department of Transportation is another. The creation of government instances is a one-time process and must be unique; a specific department should not have more than one instance. Once registered, the validity status of each government instance can be verified by anyone directly through the smart contract, ensuring transparency and authenticity.

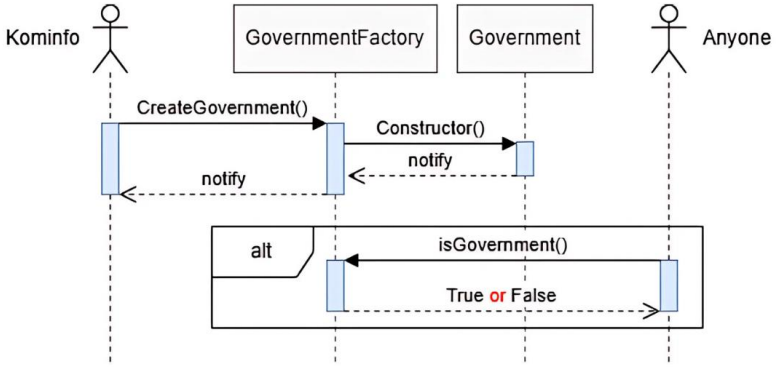

Image 5. Sequence diagram of initiating government identity

2.4.2. Managing Government Identity

When the government establishes a new division, they create a new government instance and register a unique identity for that division, as depicted in Image 6. Behind the scenes, this process automatically generates a new pair of private and public keys. The public key is then registered in the **Key** instance, which serves as the parent class of the **Government** instance. This design ensures that the public key can be used to verify the originality and authenticity of the government entity in the future, reinforcing the system's integrity and trustworthiness.

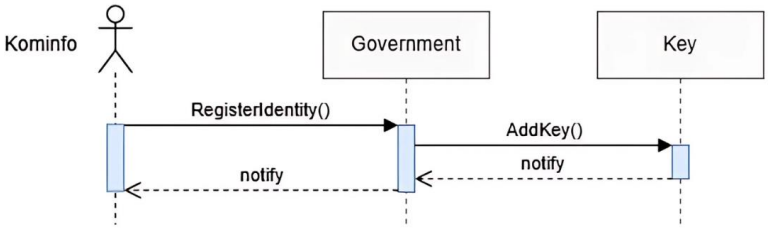

Image 6. Sequence diagram of creating a new department of government

If modifications or deletions of government information (e.g., name, address, or phone number) are needed, these can be performed directly within the smart contract, as illustrated in Image 7. To enhance the trustworthiness of this editing process, other entities can optionally act as endorsers for the changes. When multiple entities endorse a change, it indicates broader agreement and consensus, reducing the risk of bias or unilateral edits by a single entity. This mechanism ensures that changes reflect the collective will of involved stakeholders, adding an extra layer of reliability and transparency.

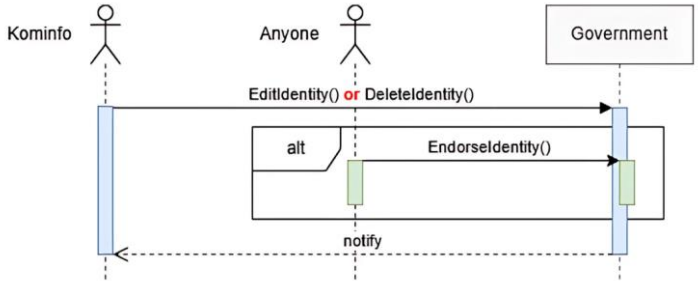

Image 7. Sequence diagram of editing or deleting government information in a department

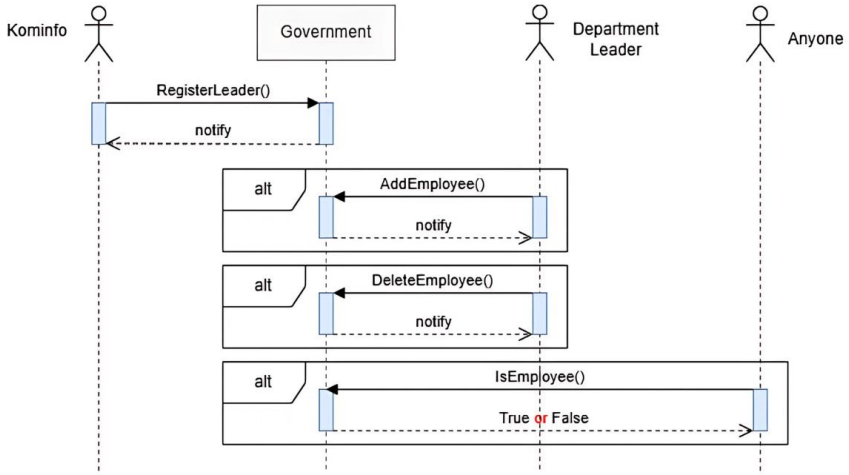

Image 8. Sequence diagram of registering new leader, adding or deleting employee in a department of government

Each government department is managed by a leader, who can be elected by registering a new leader in the smart contract, as shown in Image 8. Once registered, the leader has the authority to add new employees to their department or remove existing ones. Government employees, under the leadership of the department head, are empowered to approve changes to citizen identities, a process that will be elaborated upon in subsequent paragraphs.

2.4.3. Managing Citizen Identity

When a new citizen wants to register in the system, they must propose their registration through the smart contract, as summarized in Image 9. During this process, the system generates a pair of private and public keys. The public key is then stored in the newly created citizen smart contract instance. Similar to the government setup, all public keys are managed within the **Key** class, which serves as the parent class of the **Citizen** class. To complete the registration, a government employee must approve the proposal via the smart contract. This approval process ensures that all citizens are validated, effectively preventing the creation of fake citizen identities.

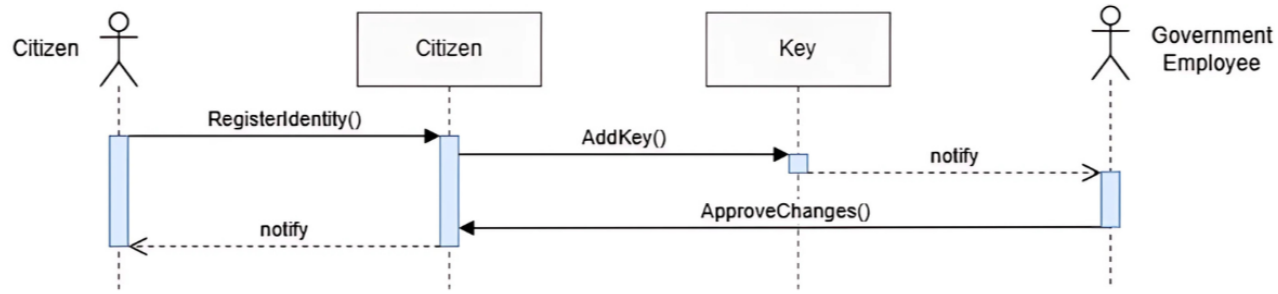

Image 9. Sequence diagram of creating new citizen identity

If a citizen needs to edit or delete information, such as their name or living address, they can request the changes through the smart contract, as shown in Image 10. To enhance the credibility of these changes, the citizen can optionally gather endorsers. These endorsers validate the proposed changes, ensuring that the updated information is accurate and genuine. The idea is that the more endorsers a change has, the more trustworthy it becomes. Ultimately, the government employee reviews and approves the changes via the smart contract if they deem the updates valid, ensuring a transparent and reliable process.

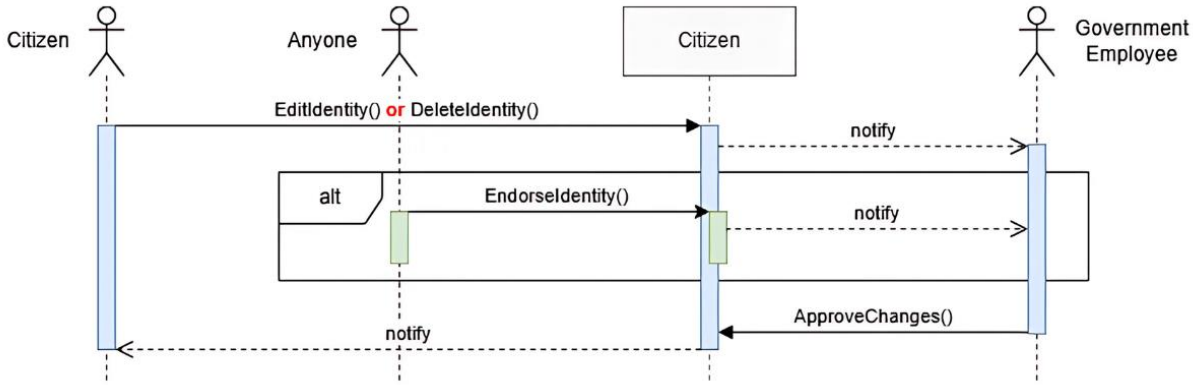

Image 10. Sequence diagram of editing or deleting citizen information

2.4.4. Identity Validation and Key Management

Our proposed system enables identity verification through OTP challenges, as illustrated in Image 11. In this process, the validation seeker (which can be a **Citizen** or a **Government** entity) requests an OTP from the prover (which can be another **Citizen** or another **Government** entity). After receiving the OTP, the validation seeker signs it using their private key and sends the resulting signature back to the prover. The prover then verifies the signature by referencing the corresponding identity instance in the smart contract (either a **Citizen** or **Government** instance). A valid signature, submitted within the OTP's time limit, is recognized as a genuine request. This ensures that the prover can confidently confirm the authenticity of the entity they are interacting with.

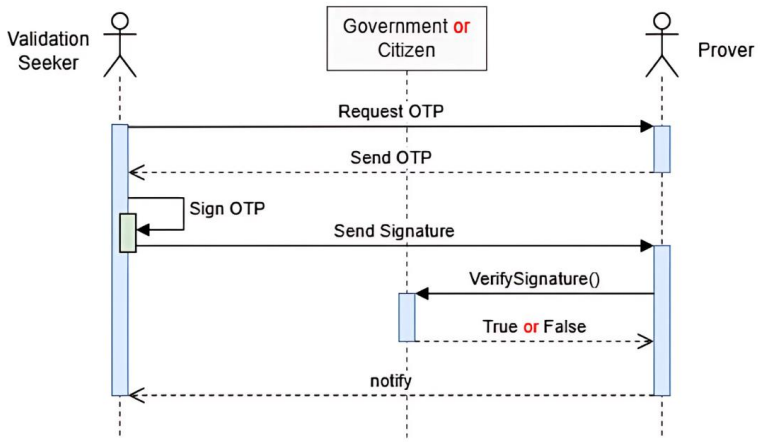

Image 11. Sequence diagram of verifying the authenticity of an identity whether they are citizen or government

Last but not least, in the event of a lost or updated key, both citizens and government entities can manage their respective keys by adding or removing them through their own **Citizen** or **Government** smart contract instance, as summarized in Image 12. This feature ensures that keys can be updated over time, providing flexibility and enhancing the security of the identities in the system.

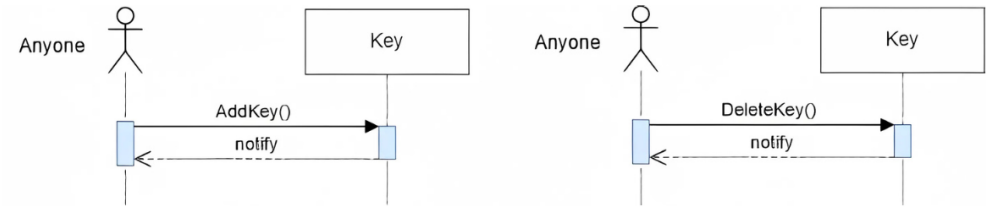

Image 12. Sequence diagram of managing keys in the smart contract.

3. RESULT AND DISCUSSION

This section presents the implementation details, user interface design, and evaluation of the proposed system. First, we describe the technical setup and tools used in developing the prototype, including both the backend and frontend components. Next, we analyze the user interfaces developed for citizens and government users, showcasing the design and functionality tailored to their specific roles. Finally, we assess the feasibility of the system through functional testing, followed by a comprehensive security analysis to evaluate its ability to address trust, integrity, and data transparency challenges.

3.1. Implementation Setup

We implemented our design as a web application using Node.js with TypeScript [9] as the backend web server. The frontend is built using basic HTML, CSS, and JavaScript. The smart contract is written in Solidity [10] and designed to be deployed on any blockchain compatible with the Ethereum Virtual Machine (EVM) [11]. For the blockchain network, we utilize Hardhat [12] as our local testnet and development tool. All system testing was conducted on a machine equipped with an Intel Core i5-8250U CPU running at 1.60 GHz and SK Hynix DDR4 RAM operating at 2400 MHz. During testing, we utilized a single CPU core and 16 GB of RAM. Image 13 illustrates our software component stack.

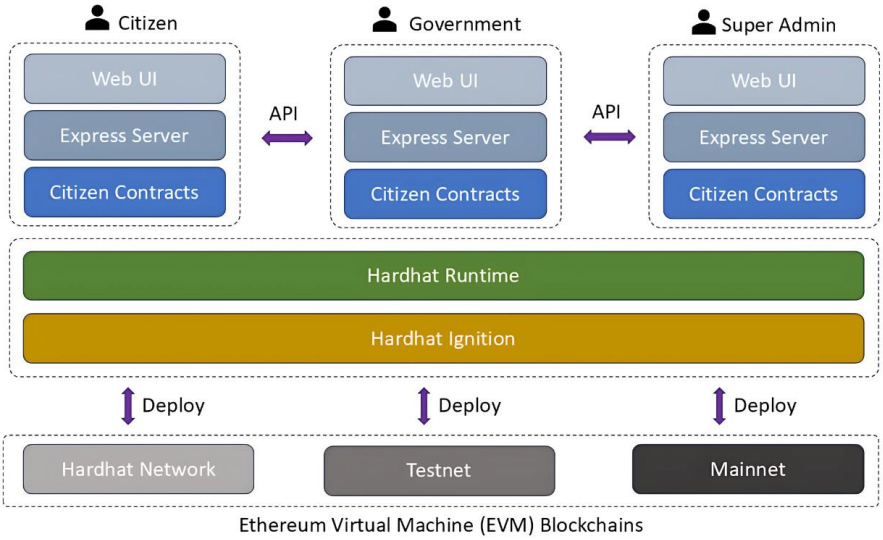

Image 13. The software architecture of our proposed system

3.2. User Interface Analysis

The user interface depicted in Image 14 showcases the user interface (UI) application designed for citizens. It consists of three primary features: a profile view displaying the citizen's information, including their smart contract address; a OTP generation feature, allowing the user to display a QR code for identity verification; and a QR code scanning feature that enables the user to scan OTP codes from others to verify their identities.

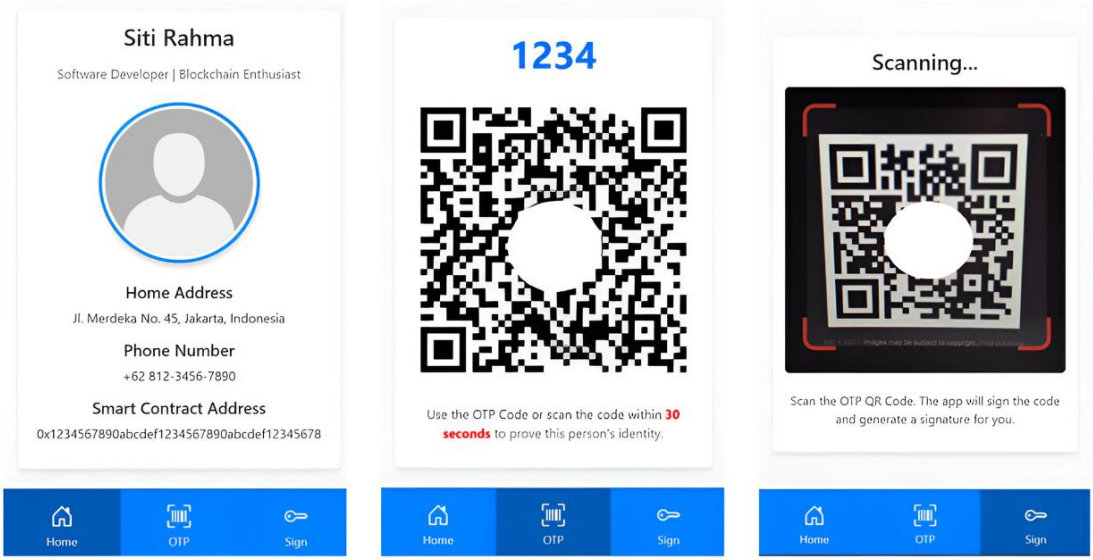

Image 14. Main user interface for Citizen

This user interface shown in Image 15 represents the government-side application. The dashboard provides an overview of critical information for government employees. The top section displays the total number of citizen contracts that have been deployed, along with the government's smart contract address, essential for managing decentralized interactions. Below that, a line chart visualizes the growth of citizen contracts over time, offering a clear trend of adoption and deployment. The bottom section lists citizen change requests, detailing the name of the citizen, the requested update (e.g., address or phone number), and an "Approve" button for the government employee to process these requests.

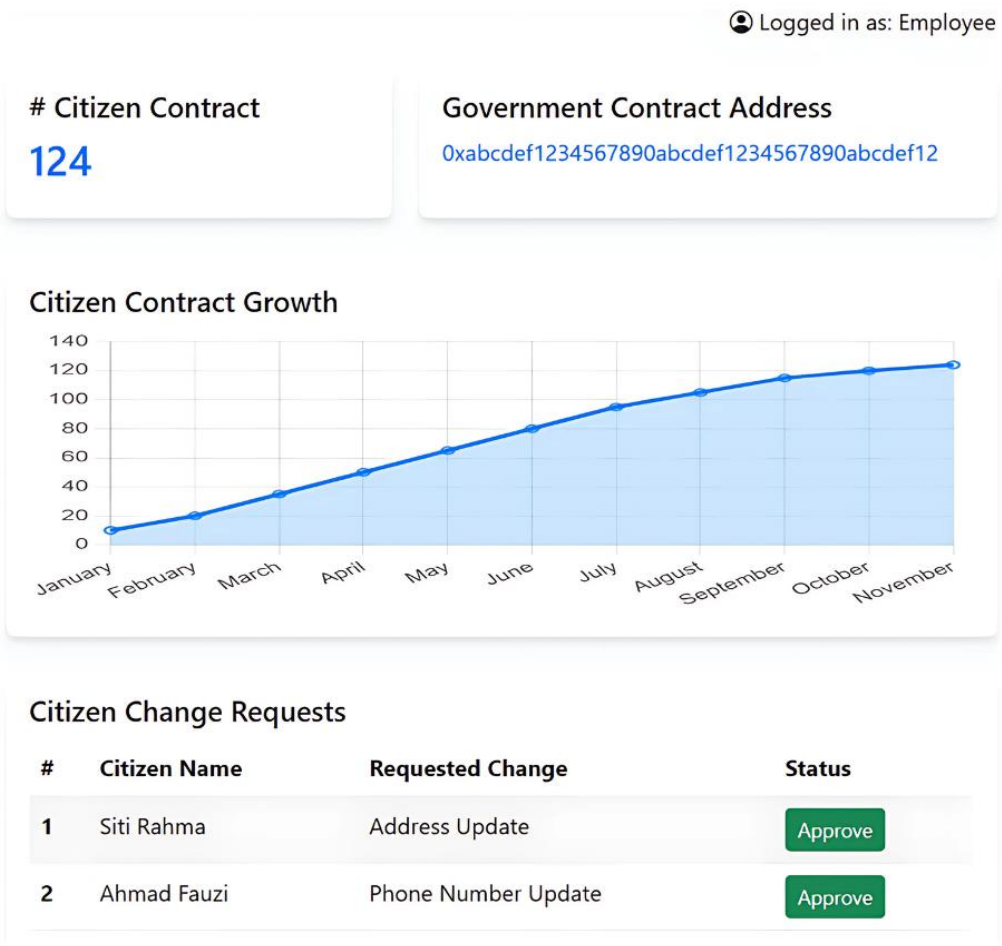

Image 15. Main user interface for Government

3.3. Feasibility Analysis

Smart contract development differs significantly from traditional application development. Once a smart contract is deployed on the blockchain network, its code becomes immutable and cannot be modified. This restriction arises from the immutable nature of blockchain technology. As a result, it is critical to ensure the correctness of the smart contract before deploying it to the mainnet (the production-ready blockchain network). To achieve this, we create unit tests for the smart contract.

In our system, we utilize the Mocha and Chai libraries for unit testing [13]. These tests verify the functionality of each method to ensure they behave as intended. Specifically, we assess not only valid inputs but also invalid inputs. For example, we test scenarios where a user with an unauthorized role attempts to access restricted methods or where a string is provided instead of a number when the contract expects a number. Rigorous testing is conducted to identify and eliminate potential bugs in the smart contract.

To maintain clarity and conciseness, we summarize the unit testing results and present the functional testing in Table 1. Listing all individual unit tests would be overwhelming and consume excessive space in the paper. From the summarized table, it is evident that all critical scenarios in our system have been thoroughly tested and are functioning correctly.

Table 1. Result of functional testing from our proposed system

Test Case	Result
All contracts are deployed succesfully	✓
User with the role of Employee can initiate citizen contract	✓
User with the role of Kominfo can initiate government contract	✓
User with the role of Kominfo can edit or remove government contract	✓
User with the role of Kominfo can add new leader to the government contract	✓
User with the role of Leader can add new employee	✓
User with the role of Leader can remove existing employee	✓
User with the role of Citizen can register identity in citizen contract	✓
User with the role of Citizen can edit citizen contract	✓
User with the role of Citizen can delete citizen contract	✓
User with the role of Citizen can endorse changes made by other citizen	✓
All users can add new key to their respective contract	✓
All users can remove existing key to their respective contract	✓
All users can verify signature in their respective contract	✓

3.4. Security Analysis

Our proposal guarantees the following security properties.

The system ensures trust and authenticity through a single root of trust. Only the super admin is authorized to create the BaseFactory, from which all citizen and government contracts are derived. This design guarantees that all entities originate from a verified source, making it easy to detect and prevent the creation of fake identities. By centralizing the creation process within a secure and controlled framework, the system maintains the integrity of all participants.

Data integrity is preserved through a strict approval mechanism for modifications. Any changes to citizen data require explicit government approval, effectively preventing unauthorized edits and identity fraud. Similarly, modifications to government data are tracked on the blockchain and supervised by Kominfo. This oversight ensures that all data changes are transparent and that malicious actions can be identified and addressed swiftly.

Identity verification is highly secure, relying on cryptographic methods to prevent impersonation and replay attacks. Entities must sign a temporary, one-time code that is ephemeral and resistant to guesswork. This mechanism ensures that only legitimate users can verify their identities, adding a robust layer of protection against unauthorized access and fraudulent activities.

Finally, **the system leverages blockchain technology to achieve transparency and tamper resistance in data storage**. All data is stored immutably on the blockchain, enabling multiple nodes to audit the accuracy and integrity of the stored information. This decentralized auditing capability ensures that any attempts to manipulate data, whether by governments or citizens, are quickly detected. The visibility provided by blockchain technology fosters accountability and trust among all stakeholders.

3.5. Discussion

The primary contribution of this paper lies in the integration of blockchain technology with decentralized identity (DID) frameworks specifically tailored for governmental ecosystems. Unlike conventional centralized identity management systems, our proposed model addresses fundamental concerns such as data security, user privacy, and interoperability by leveraging blockchain's inherent attributes like decentralization, immutability, and transparency. By utilizing factory-based smart contracts and hierarchical authority structures, our model ensures robust identity verification and management processes suitable for complex governmental interactions. Furthermore, the practical implementation and testing of the framework provide clear evidence of its scalability, operational efficiency, and enhanced privacy mechanisms. Therefore, this study not only advances the theoretical understanding of decentralized identities but also provides a practical reference model for governmental entities considering blockchain-based digital identity adoption [14], [6].

Despite the advantages of blockchain-based decentralized identity systems presented in this paper, significant research gaps still persist when compared to previous work. Existing literature often addresses either technical dimensions or governance considerations independently, leaving comprehensive, integrative solutions relatively unexplored. For instance, earlier studies such as Rodionov [5] underline regulatory challenges and governance implications, while Kersic et al. [6] emphasize interoperability between on-chain and off-chain identity management through digital wallets. Moreover, Gans et al. [15] and Ishmaev [16] highlight governance, societal impacts, and ethical considerations of self-sovereign identities, yet these studies do not extensively cover the integration and systematic application of such models within specific governmental contexts. Therefore, this paper addresses this research gap by proposing a coherent and systematic framework explicitly designed to meet governmental requirements.

4. CONCLUSION

This paper introduces a decentralized identity system tailored for e-government ecosystems, addressing the inherent shortcomings of traditional centralized identity management. By utilizing blockchain technology, the system ensures trust, security, and transparency through factory-based smart contracts, hierarchical structures, and cryptographic validation protocols. The prototype demonstrates the feasibility of the approach, highlighting its ability to enhance privacy, operational efficiency, and data protection through the performed functional testings. Future research will focus on evaluating public acceptance and usability of the system, particularly among Indonesian citizens, and exploring the integration of incentives, such as credit and reward mechanisms, to encourage proactive citizenship and system adoption.

References

[1] M. Korir, S. Parkin, and P. Dunphy, "An empirical study of a decentralized identity wallet: Usability, security, and perspectives on user control," in *Proc. 18th Symp. Usable Privacy and Security (SOUPS 2022)*, 2022, pp. 195–211.

[2] A. M. Antonopoulos and D. A. Harding, *Mastering Bitcoin*. O'Reilly Media, Inc., 2023.

[3] O. Avellaneda, A. Bachmann, A. Barbir, J. Brenan, P. Dingle, K. H. Duffy, E. Maler, D. Reed, and M. Sporny, "Decentralized identity: Where did it come from and where is it going?," *IEEE Communications Standards Magazine*, vol. 3, no. 4, pp. 10–13, 2019.

[4] W. Metcalfe *et al.*, "Ethereum, smart contracts, DApps," *Blockchain and Crypt Currency*, vol. 77, pp. 77–93, 2020.

[5] R. A. Rodionov, "The Potential of Blockchain Technology for Creating Decentralized Identity Systems: Technical Capabilities and Legal Regulation," *International Journal of Law and Policy*, vol. 2, no. 4, pp. 19–24, 2024.

[6] V. Kersic et al., "Orchestrating Digital Wallets for On- and Off-Chain Decentralized Identity Management," *IEEE Access*, vol. 11, pp. 78135–78140, 2023.

[7] L. Thorbecke, "Decentralized Identity Management for Swiss Federalism," Master Thesis, University of Zurich, Mar. 2020.

[8] L. E. Pleger and K. Guirguis, "Public Acceptance of Government Information Systems: Evidence From the Popular Vote on an Electronic Identity (e-ID) in Switzerland," *Swiss Yearbook of Administrative Sciences*, vol. 15, no. 1, pp. 68–91, 2024.

[9] J. Bogner and M. Merkel, "To type or not to type? A systematic comparison of the software quality of JavaScript and TypeScript applications on GitHub," in *Proc. 19th Int. Conf. Mining Softw. Repositories*, 2022, pp. 658–669.

[10] D. P. Bauer, "Solidity," in *Getting Started with Ethereum: A Step-by-Step Guide to Becoming a Blockchain Developer*, Springer, 2022, pp. 13–16.

[11] W. Zhang and T. Anand, "Ethereum architecture and overview," in *Blockchain and Ethereum Smart Contract Solution Development: Dapp Programming with Solidity*, Springer, 2022, pp. 209–244.

[12] S. M. Jain, "Hardhat," in *A Brief Introduction to Web3: Decentralized Web Fundamentals for App Development*, Springer, 2022, pp. 167–179.

[13] A. Libby, "Unit Testing and Svelte," *Practical Svelte: Create Performant Applications with the Svelte Component Framework*, Springer, 2022, pp. 181–209.
[14] C. S. Sung and J. Y. Park, "Understanding of blockchain-based identity management system adoption in the public sector," *Journal of Enterprise Information Management*, vol. 34, no. 5, pp. 1481–1505, 2021.
[15] R. B. Gans, J. Ubacht, and M. Janssen, "Governance and societal impact of blockchain-based self-sovereign identities," *Policy and Society*, vol. 41, no. 3, pp. 402–413, 2022.
[16] G. Ishmaev, "Sovereignty, privacy, and ethics in blockchain-based identity management systems," *Ethics and Information Technology*, vol. 23, no. 3, pp. 239–252, 2021.