


Article

Self-Sovereign Identity-Based E-Portfolio Ecosystem

Yu-Heng Hsieh, Jun-Yu Yan, Chia-Hung Liao  and Shyan-Ming Yuan * 

Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan; k28998989.cs11@nycu.edu.tw (Y.-H.H.); yen.cs09@nycu.edu.tw (J.-Y.Y.); aiallen.cs07@nycu.edu.tw (C.-H.L.)

* Correspondence: smyuan@nycu.edu.tw

Abstract: In Taiwan, traditional student assessments, covering academic and extracurricular achievements, have shifted from paper to electronic portfolios (e-portfolios). However, limited trust among institutions restricts students from freely sharing and using their educational data. This paper introduces a self-sovereign identity-based infrastructure aimed at enhancing personal data security within the e-portfolio ecosystem. The proposed system includes two core components: (1) a decentralized identity chain, aligning user identities across platforms and granting users full self-sovereign control; and (2) an e-portfolio application chain to manage user interactions and access permissions within the ecosystem. A trusted educational authority also audits data sources, ensuring data integrity and reliability. This infrastructure empowers users to control who can access their data, safeguarding their security, with the identity chain preventing unauthorized access and the application chain recording authorization statuses to restrict data visibility to approved parties only.

Keywords: smart contract; e-portfolio; self-sovereign identity; blockchain

1. Introduction

Since 2019, significant changes have unfolded in university admissions processes, pivoting from a predominant focus on entrance exam scores to a heightened consideration of a student's enduring academic performance. This shift has highlighted the limitations of conventional paper documents in the eyes of university reviewers. In response, the proposal of an e-portfolio system [1] has emerged as a solution, offering reviewers a more comprehensive understanding of a student's academic and extracurricular achievements.

By embracing the e-portfolio system, reviewers gain access to a detailed breakdown of a student's final scores across various academic subjects, coupled with insights into their engagement in extracurricular activities. This comprehensive approach provides deeper insights into a student's passions and achievements, aiding reviewers in making informed decisions. Significantly, the e-portfolio system has taken the place of conventional paper records, leading to a more streamlined admissions process, greatly enhancing overall efficiency.

The traditional e-portfolio system encounters several challenges. First, students find it cumbersome to manage multiple text-based password authentication systems, as high schools and activity organizations are often reluctant to adopt social logins like Google or Facebook. This reliance on text-based passwords hinders the creation of a unified digital identity. Additionally, the lack of a trusted connection between institutions and reviewers disrupts direct data sharing, leading to admissions delays that can negatively affect acceptance rates. The absence of a robust authentication method also raises concerns about the reliability of submitted data, making it difficult for reviewers to quickly verify the accuracy and authenticity of information due to limited review time. This can result in inaccuracies in admissions decisions, disadvantaging deserving students.

Moreover, students may require greater control over who can access their data. By implementing a decentralized identity management system, students can selectively share their learning history, allowing them to avoid disclosing unfavorable academic records or



Citation: Hsieh, Y.-H.; Yan, J.-Y.; Liao, C.-H.; Yuan, S.-M. Self-Sovereign Identity-Based E-Portfolio Ecosystem. *Appl. Sci.* **2024**, *14*, 10361. <https://doi.org/10.3390/app142210361>

Academic Editor: Gianluca Lax

Received: 14 October 2024

Revised: 4 November 2024

Accepted: 8 November 2024

Published: 11 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

incomplete assessments. Consequently, there is a pressing need for a system that ensures the credibility and authenticity of student data. Such a system would streamline authentication, facilitate direct data sharing, and provide reviewers with accurate information, ultimately enhancing the admissions process for all stakeholders.

To address the issues discussed earlier, we propose a solution based on a self-sovereign identity-based personal information security control infrastructure for the e-portfolio ecosystem, using blockchain technology. This infrastructure consists of two parts:

Decentralized identity blockchain: The decentralized identity blockchain enables users to have a self-sovereign identity, allowing them to fully control their own identity without relying on centralized services.

Application blockchain: A blockchain specifically designed for the e-portfolio scenario enables students to share their documents, allows institutions to authorize awards, helps teachers upload student e-portfolio documents, and provides university reviewers with access to review students' documents.

Using blockchain and smart contracts, users can log in with a digital signature, eliminating the need for passwords, which are often vulnerable to hacking or lost. The e-portfolio application blockchain allows individuals to grant their consent and provide their information to reviewers, while also enabling trusted educational bodies to audit the information provided by activity groups. Access to any student data is permitted only with the student's explicit authorization, ensuring that the data's credibility and authenticity are maintained. This provides reviewers with reliable information to make informed decisions. Overall, this self-sovereign identity-based data protection framework for the e-portfolio ecosystem offers a secure and efficient solution to challenges faced by both reviewers and students, safeguarding the integrity of the admissions process and enhancing opportunities for deserving candidates.

The upcoming sections are structured as follows: Section 2 offers an in-depth review of blockchain's background, alongside an exploration of related work on e-portfolio ecosystem and self-sovereign identity. In Section 3, the operational model, smart contract design, and the key software components are introduced, offering insights into the foundational elements of our system. Section 4 outlines the proposed workflow within our system, elucidating the sequential steps and interactions that define its functionality. A demonstration of the proposed system is detailed in Section 5, showcasing its practical application and functionality. In Section 6, we delve into the evaluation of our system's performance, providing an in-depth analysis of its effectiveness and efficiency. The final section wraps up the paper by summarizing the key findings and insights derived from the study. To facilitate readability, some abbreviations used in this study are defined in Appendix A.

2. Background and Related Work

2.1. Blockchain

Blockchain serves as an unalterable distributed ledger, enabling the exchange of information within a network of participating nodes. Its applications extend across diverse domains, notably in the realm of cryptocurrencies like Bitcoin [2]. The architecture of blockchain involves organizing transaction records into blocks, linked by a cryptographic mechanism. This not only ensures data confidentiality but also establishes a decentralized trust foundation, eliminating the reliance on centralized authorities. Blockchain's decentralized structure ensures that all involved nodes uphold a consistent ledger, thereby providing a dependable method for sharing information.

Blockchain systems can be classified according to node type: public, consortium, or private. In a private blockchain, one entity retains control, and transaction visibility is limited to its members. Public blockchains, in contrast, are accessible to everyone, allowing users to read the ledger, conduct transactions, and participate as nodes in maintaining the network. Consortium blockchains involve multiple entities collaborating within a semi-centralized ecosystem. These frameworks facilitate cooperative transactions through smart contracts, fostering trust among the participating parties.

Ethereum [3] was proposed in 2014 by V. Buterin. Programmable code on the blockchain, known as a smart contract, initiates actions automatically when specific conditions are satisfied. By combining with blockchain, smart contracts establish a trusted program execution environment that does not rely on any centralized server. Programmers can develop their own decentralized applications (dapps) by creating smart contracts and implementing them on the blockchain, which has been applied to several fields, such as healthcare [4–7], the Internet of Things (IoT) [5,8,9], contract production [10], product traceability [11], supply chains [12,13], and open banking [14].

Consensus mechanisms [15] in blockchain are methods to achieve agreement among participants, ensuring that all have a consistent view of the data's state. These mechanisms resolve how nodes in a decentralized network can trust each other and maintain the integrity and consistency of the blockchain data. Here are some commonly used consensus mechanisms:

Proof of Work (PoW): PoW is the earliest and one of the most widely used consensus mechanisms, central to the Bitcoin blockchain. In PoW, miners must solve a complex mathematical puzzle to add a new block to the blockchain. This process requires substantial computational power and energy but is effective in preventing tampering, ensuring security.

Proof of Authority (PoA): PoA is a reputation-based consensus mechanism, primarily used in private or consortium blockchains. Unlike PoW, PoA does not rely on high levels of computational resources or token holdings. Instead, it depends on verified authority nodes to generate and validate blocks. Only authorized nodes are able to add new blocks, making PoA highly efficient and well suited for scenarios requiring fast validation and stability, such as supply chain management or enterprise consortium blockchains.

Practical Byzantine Fault Tolerance (PBFT): PBFT is a consensus mechanism specifically designed for fault tolerance, often used in private or consortium blockchains. PBFT can achieve a consensus even when some nodes display inconsistent behavior, ensuring data consistency. It is commonly applied in blockchain applications requiring rapid validation.

These consensus mechanisms aim to enhance blockchain security, efficiency, and decentralization. Blockchains choose mechanisms based on specific application requirements to achieve optimal performance for their use cases.

2.2. Hyperledger Fabric (HLF)

Hyperledger Fabric [16] is a blockchain framework tailored for enterprise applications. It operates as a permissioned blockchain, restricting access to authorized participants—a departure from public blockchains like Bitcoin [2], in which anyone can participate.

Hyperledger Fabric boasts a modular architecture as one of its key features. This allows enterprises to customize the framework to fit their specific needs, making it more flexible to use. For example, Hyperledger Fabric provides functional components like Fabric CA (Certificate Authority), which is used for identity management, and channel and offline sign, which provides privacy and security features.

2.2.1. Fabric CA

Fabric CA serves as a public key infrastructure (PKI) system for managing identities within HLF, a permissioned blockchain network. Unlike public blockchains that are accessible to all, participation in HLF requires users to register with Fabric CA and acquire an X.509 certificate signed by Fabric CA. Obtaining a certificate from Fabric CA involves two steps: registration and enrollment. During the registration step, a Fabric CA administrator adds a new user and assigns attributes based on their role and permissions within the network.

In the enrollment step, the user submits a certificate signing request (CSR) to Fabric CA using their private key (P, k). Fabric CA uses the CSR to generate an X.509 certificate containing the assigned attributes, which is then signed by Fabric CA. The certificate is returned to the user, who can then use it to prove their identity and access the network based on their assigned attributes and permissions. Fabric CA enables organizations

to assign details and permissions to users, linking them to their public key ($P_u k$). This provides more granular control over network access and permissions. Users use their $P_r k$ to verify ownership of their $P_u k$, ensuring a secure method for managing their identity within the network.

2.2.2. Channel

Supporting multiple ledgers, HLF utilizes channels—private blockchains shared exclusively among participating organizations. These channels ensure transaction privacy, enable the separation of business logic, and empower organizations to define their governance rules and policies. Outside parties cannot access information about transactions on channels, ensuring transaction confidentiality.

2.2.3. Offline Signing

HLF facilitates the development of services through software development kits (SDKs) tailored for various programming languages. These SDKs empower service providers in constructing their offerings. While web services provide convenience with their user-friendly interface and minimal need for additional installations, the present SDK faces constraints when utilized in browsers due to compatibility challenges.

Typically, service providers furnish an SDK environment for users to execute transactions and serve as identity managers. This involves storing the user’s X.509 certificate and $P_r k$, as depicted in Figure 1. In this situation, as a user triggers a request, the service provider employs the stored $P_r k$ to sign transactions. While this method is convenient, it may fall short of meeting stringent privacy requirements. The storage of $P_r k$ s in services introduces vulnerabilities, potentially leading to identity fraud and unauthorized transactions.

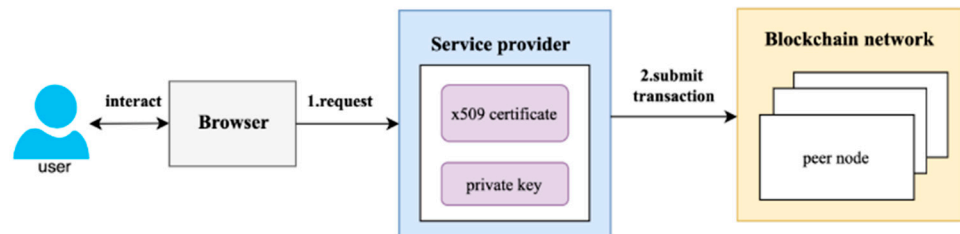


Figure 1. Service providers overseeing a user’s identity management.

Offline signing emerges as a solution to address user concerns about $P_r k$ security. Illustrated in Figure 2, this approach keeps the $P_r k$ securely with the user. When a user wants to create a transaction, the service provider generates a transaction draft based on the user’s certificate and returns it. The user then signs the draft with their $P_r k$ and submits the signed transaction to the blockchain network through the service provider.

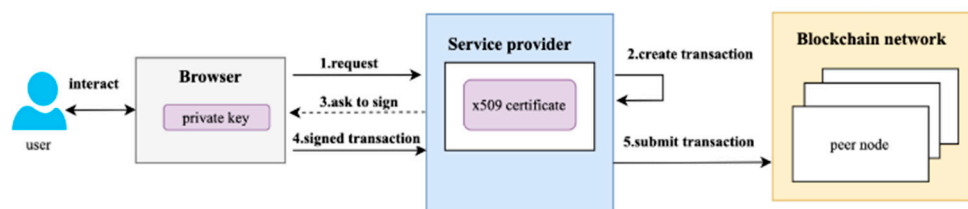


Figure 2. Offline signing flow.

By adopting offline signing, $P_r k$ s are not stored with external organizations, granting users a self-sovereign identity. This approach enhances security by preventing unauthorized access to $P_r k$ s, giving users full authority over their transactions. In summary, offline signing is an effective solution for safeguarding the security and privacy of user transactions on the HLF.

2.3. E-Portfolio

The e-portfolio system was established in Taiwan in 2019, coinciding with a shift in focus for college admissions from entrance exam results to a greater emphasis on long-term learning performance. The e-portfolio records students' academic performance in high school and enables students to regularly record and edit their files to more accurately and realistically present their characteristics, professional interests, and learning records. Through these files, university reviewers can understand a student's learning performance, which cannot be ascertained through the entrance examination.

Figure 3 depicts the procedure for uploading e-portfolio files. Currently, the Ministry of Education of Taiwan has created an e-portfolio central database to integrate high school students' e-portfolio files, school grades, and activity records. Students' data are uploaded layer by layer and centralized in a specific organization, which also makes it an attractive target for attackers. Additionally, students are not free to use uploaded files, which can only be used for university admissions reviews.

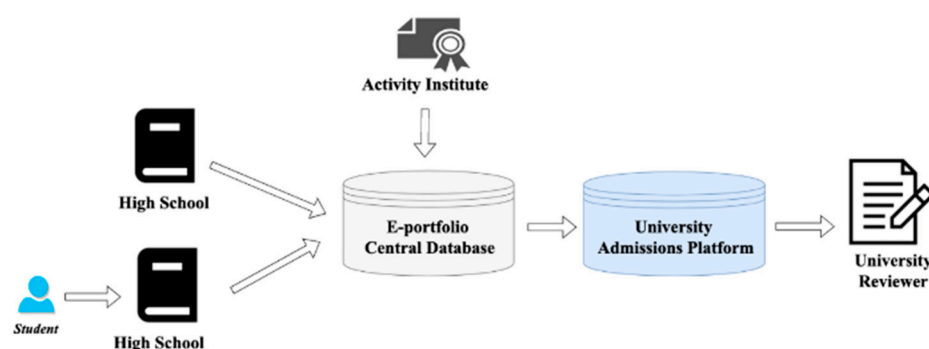


Figure 3. The process of uploading e-portfolio files [17].

In [18], the author delineated three types of traditional e-portfolio systems: The Developmental Portfolio highlights a student's growing skills over time, serving as a developmental tool that incorporates self-assessment and reflection and promotes communication with academic staff. The Assessment Portfolio, on the other hand, demonstrates a student's proficiency in specific areas, employed for continuous or summative evaluations, and assesses their performance based on program standards. The Showcase Portfolio highlights a student's skills and work examples, typically created at the end of a program for potential employers to assess the quality of their work. However, these traditional e-portfolio systems come with challenges, such as the need for constant system activity to prevent student uploads or the necessity for teachers to provide timely feedback to students.

Some research suggests that the decentralized and immutable features of blockchain are helpful in building a more open and trustworthy educational field [19]. Chuyang Li et al. [20] proposed a blockchain system that combines public and private blockchains for online learning evaluation and certification. This architecture not only reduces the complexity of the public blockchain but also maintains the flexibility of the application. Junho Jeong et al. [21] proposed a blockchain-based personal portfolio authentication system to improve the centralized storage of student and teacher portfolios in Korean educational institutions (NEIS).

There are several commercial applications. Turing Certs [22] established a third-party authentication authority, which creates an anti-counterfeiting e-wallet for students to store their certificates. Netizen [23] proposed an electronic certificate infrastructure based on a private blockchain, which stores the hash of the certificate in the blockchain to ensure the integrity of the certificate. These applications [22,23] use blockchain technology as their solution, but the services are managed by a single entity, which still has a single point of failure and raises concerns about companies going out of service.

2.4. Identity Management

Text passwords are commonly used for authentication, but they rely on the trustworthiness of service providers. Users may reuse the same password for multiple services for convenience [24], but if a malicious provider gains access to the password, they can use the user's identity to log in to other services. Additionally, managing too many passwords can be difficult. To address these issues, some providers offer social login through tech giants like Google and Facebook. Social login allows users to use their digital identity to log in to multiple services without additional registration, but it still relies on centralized platforms, which raises concerns about the fraudulent use of user identities.

Web-based digital identities (DIs) have gone through four stages of evolution [25]: centralized identity, federated identity [26], user-centric identity [27], and self-sovereign identity (SSI) [28,29]. The traditional centralized approach to digital identity has been gradually losing its dominance as users demand greater control and autonomy over their identities.

SSI [29] offers an innovative method for digital identity, enabling users to take full control of their identity management. SSI allows users to have distributed identities across multiple locations that are interoperable and portable, which means that they are not limited to specific websites or services.

In [29], the Self-Sovereign Identity (SSI) framework is defined according to ten features: control, existence, transparency, access, persistence, consent, interoperability, portability, protection, and minimalization. The government utilizes a Decentralized Identifier (DID) chain to create user identities in the e-portfolio application chain, ensuring alignment with these SSI features. Notably, within the ten aspects, our identity in the e-portfolio application chain adheres to nine, excluding "control". The exclusion of the "control" attribute is justified as it is considered unnecessary for our system. It is noteworthy that a parallel research approach, as documented in [30], also omits this attribute.

By adhering to these principles, agencies can help ensure that SSI is secure, transparent, and protects users' privacy. SSI has the capacity to revolutionize digital identity systems by giving users enhanced control and autonomy over their identities.

Recently, blockchain technology has enabled the concept of SSI [28], according to which users have complete control of their own identities anchored in blockchain. Several studies have proposed self-sovereign identity-based digital identity platforms, such as "Casper" by Eranga Bandara et al. [31], which provides users with one identification stored in a mobile identity wallet to log in to different organizations. Nitin Naik et al. [32] proposed the open-source "uport" identity management system to realize SSI and offer application developers a general authentication option. With blockchain technology, users' identities cannot be used fraudulently, providing users with more control and security.

In [7], the authors propose a physiological data sharing platform via blockchain technology, incorporating both a decentralized identity chain (DID-chain) and a physiological data sharing chain. Their DID-chain meets nine out of the ten requirements of SSI [29], including existence, transparency, access, persistence, consent, interoperability, portability, protection, and minimalization.

This consortium blockchain, developed in collaboration with government and regulatory authorities, aims to unify user identities across various ecosystems. Users and organizations must register using their real names through official government channels and undergo a verification process. Upon successful verification, they receive a personal identity contract (*PIcon*) that is exclusively controlled by the individual or organization. This contract is secured by a primary identity represented by a private key, which serves as a verification method for the authentic entity. Once users acquire their identity, they can enroll with App-chains and access services offered by those registered App-chains. This system establishes a secure and verified identity framework, promoting trust and reliability within the ecosystem, as illustrated in Figure 4.

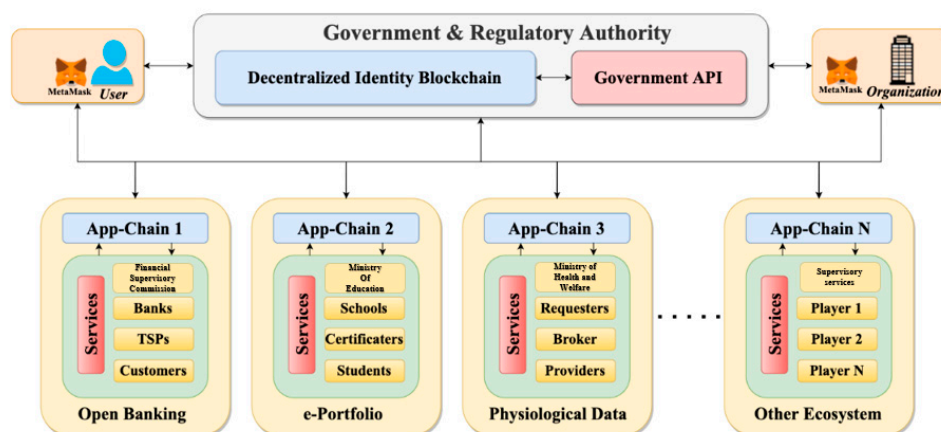


Figure 4. Integrate DID-chain and App-chains.

Organizations can use the primary identity to verify a user's existence and create an App-chain identity for them. Users can add registration materials, such as encrypted Certificate Signing Requests (CSRs) and encrypted App-chain private keys, to the event logs of their personal identity. These event logs record identity-related information and can be used as evidence in the event of a dispute to prove the user's identity.

2.5. Access Control

In the development of applications, the choice of an access control model is crucial and should align with the specific requirements of the scenario. Two common models are Role-Based Access Control (RBAC) [33] and Attribute-Based Access Control (ABAC) [29,30]. RBAC simplifies permission assignment by predefined roles, facilitating swift access control for new users. However, its effectiveness diminishes when dealing with dynamic attributes such as temporal and spatial parameters. Conversely, ABAC offers increased adaptability in administering access control, accommodating various organizational roles and job responsibilities, and can include dynamic attributes in access control policies.

Existing access control management systems, such as PKI [34], face scalability and granularity issues and are susceptible to attacks targeting certificate authorities. Blockchain-based solutions, with their transparency, nonrepudiation, and security features, emerge as attractive alternatives for access control. Many decentralized access control methods use blockchain to share access control policies or employ a multilayer blockchain structure to ensure reliability and efficient operations.

For example, in [35], they suggested using blockchain to manage access control policies. Ref. [36] employed AuthPrivacyChain to secure cloud services against unauthorized access. The author of [37] proposed a user rights management system using blockchain and smart contracts to facilitate relationships between users, data providers, and regulatory bodies. Ref. [38] introduced a distributed ABAC system that employs blockchain to audit access attempts in digital libraries. Lastly, Ref. [39] discussed a multi-stakeholder ABAC system in which blockchain smart contracts support relationships between users, data providers, and regulatory bodies. These various approaches highlight the versatility of blockchain technology in improving access control in different applications.

2.6. Related Work

Stuchain [40] employs ABAC and RBAC on Hyperledger Fabric (HLF) to enable teachers to manage academic records and students to control access. However, its reliance on traditional databases limits its ability to fully represent student interests. In a similar vein, an HLF-based e-portfolio system [41] effectively manages evaluations and course data but lacks the capability to track extracurricular activities, hindering a complete view of student progress.

Merlec et al. [42] introduce a four-layer e-portfolio system on Ethereum that facilitates secure data sharing, though high smart contract fees pose scalability issues. PETS [43] improves governance in higher education by connecting on-chain and off-chain data through standardized APIs. EduRSS [44] and MOOCsChain [45] emphasize data privacy, with EduRSS balancing costs through off-chain storage, while MOOCsChain secures MOOC data using Hyperledger Fabric and IPFS. While these systems enhance educational data security, they still grapple with challenges related to scalability, cost-effectiveness, and comprehensive data representation.

This paper proposes an advanced blockchain-based educational data system that addresses the limitations of previous models. By integrating ABAC and RBAC, our system facilitates detailed access control and combines on-chain and off-chain storage for enhanced scalability and cost efficiency. Notably, it incorporates extracurricular activity tracking and supports parallel execution to efficiently manage multiple user requests, presenting a more comprehensive and effective approach to educational data management.

2.7. Preliminary Work

Comparing this study to the preliminary work in [46], a distinct framework for an e-portfolio ecosystem has been proposed. While the previous work focused on a single blockchain utilizing Hyperledger Fabric for the application chain, several notable differences characterize our approach. Firstly, our system architecture diverges significantly. To facilitate the exchange of learning history information, we adopted a dual-blockchain setup utilizing both Ethereum and Hyperledger Fabric frameworks. This novel configuration contributes to enhanced system functionality. Secondly, the system features themselves are markedly dissimilar. Unlike [46], in which user accounts are stored within registered institutions, introducing the risk of potential loss due to hacking or human error, our paper empowers users to manage their private keys. With this user-centric approach, individuals retain control over their private keys, which serves as their means of accessing the ecosystem. This effectively minimizes vulnerabilities associated with centralized storage. Thirdly, our system boasts broader applicability and user convenience compared to that in [46]. Our system not only builds upon the advantages of the preliminary work, but also offers the streamlined capability for users to log in using a single account. Lastly, the performance of our proposed system has been exhaustively analyzed, and these findings are extensively detailed in Section 6, providing valuable insights into its operational efficiency and effectiveness.

3. System Design

3.1. Operation Model

The proposed self-sovereign identity system includes two main elements: the decentralized identity blockchain (DID-chain) and the e-portfolio application blockchain (EApp-chain). These two blockchains are maintained by different organizations to handle different types of transactions. The DID-chain integrates the identities of users across various ecosystems, while the EApp-chain is responsible for designing access control authorization based on specific ecosystem requirements.

3.1.1. Decentralized Identity Blockchain (DID-Chain)

In the DID-chain, we utilize the decentralized identity chain developed in [7] to manage user identities, ensuring compliance with the four key requirements of General Data Protection Regulation (GDPR) [47], with the government serving as the administrator. Users register their identities in the DID by providing personal information, which the government processes to generate a unique Decentralized Identifier (DID) for each individual. Once issued, the DID enables users to securely access all services within the App-chain.

3.1.2. E-Portfolio Application Blockchain (EApp-Chain)

In the EApp-chain, we utilize HLF to construct a multi-channel ecosystem. Our EApp-chain comprises two channels to tackle two key issues—data credibility and user authorization. By leveraging the multi-channel mechanism provided by HLF, we effectively segment the business logic and improve ledger privacy.

Before utilizing the App-chain, users must first create an identity on the DID-chain and upload the necessary registration material to their *PIcon* on the DID-chain based on the blockchain framework utilized in the ecosystem. Upon registration at any high school, users are bestowed with a self-sovereign e-portfolio identity (SSEI). This primary identity allows users to easily log in to any participating entity, in which they can use their SSEI to manage access within the e-portfolio ecosystem. The intricacies of the login and registration workflow will be expounded upon in Section 4, and an in-depth overview of the user journey and authentication processes within this system is provided.

Figure 5 presents the structure of our proposed e-portfolio ecosystem. In this system, students are equipped with a self-sovereign identity, allowing them to govern data authorization without resorting to conventional text password logins. Reviewers gain direct access to reliable review data from high schools and activity organizations, eliminating the need for these institutions to upload student data to a centralized database. Data from various sources, including awards, exam scores, and extracurricular activities, are subject to verification by educational units. The system involves multiple entities, such as the certificate and award channel (*CaAch*), access control channel (*ACch*), central education unit (*CEU*), local education unit (*LEU*), high school (*HS*), activity organization (*AO*), student (*STU*), and reviewer (*RE*). Each entity plays a specific role, contributing to a decentralized and secure framework for the management and access of educational data. Further details on the roles and interactions of these entities are discussed in subsequent sections. The roles of these parties are explained in detail as follows:

- **Certificate and Award Channel (*CaAch*):** In the *CaAch*, *CEU* and *LEU* serve as peer nodes to share the ledger1. Organizations that are not part of this channel, such as *HSs*, *REs*, and *AOs*, must obtain client credentials from peer node organizations. This channel stores information about *AOs* and *REs* that have passed the education unit's audit, as well as each user's award information. The *CEU* and *LEUs* form an audit alliance to jointly maintain the reliability and credibility of users' awards.
- **Access Control Channel (*ACch*):** The *ACch* plays an important role in the e-portfolio ecosystem by allowing the *CEU*, *LEUs*, and *HSs* to act as peer nodes and share the ledger2 as shown in Figure 5. This channel stores the authorization status of users, with each user having a unique access control instance (*ACins*) that can only be updated by themselves. The access control manager contract is tasked with managing access control in the ecosystem, ensuring that only permitted users can access their respective data. By segmenting this functionality into a separate channel, the e-portfolio ecosystem can effectively manage user authorization and access control, which helps keep user data secure and private.
- **Central Education Unit (*CEU*):** The role of the *CEU* is crucial in the e-portfolio ecosystem, as it is responsible for organizing college entrance exams and supervising the overall system. As part of its responsibilities, the *CEU* maintains a root-ca server, which issues identity certificates to reviewers. A root-ca server is a trusted entity that issues digital credentials to authenticate the identity of users or devices in a network. The certificates issued by the root-ca server are considered trustworthy because they are signed by the root-ca's *Prk*. In the context of the e-portfolio ecosystem, the root-ca server issued identity certificates to reviewers, which are used to verify their identities when accessing the system. This helps to ensure that only authorized reviewers can access user data and provide feedback.
- **Local Education Unit (*LEU*):** In the proposed e-portfolio ecosystem, *LEUs* are responsible for auditing activity organizations. Once an *AO* passes the audit, they are granted the right to create awards for students and receive a client credential to confirm user

consent on the *ACch*. Additionally, the *LEU* has a root-ca server responsible for issuing identity certificates to the local *HS* to create an intermediate *CA*.

- High School (*HS*): *HS*s are responsible for managing and storing students' school grades and e-portfolio files in the proposed system. They also monitor events on the *CaAch* to add award attributes for their students. To achieve this, *HS*s use an intermediate *CA* to create an identity certificate for each student, enabling them to participate in the e-portfolio ecosystem.
- Activity Organization (*AO*): The *AO* is an organization that manages students' extracurricular activity data, such as TOEFL, APCS, and online learning platforms. These institutes can apply for verification from local education units to improve the credibility of the data they manage.
- Student (*STU*): In different ecosystems, a user may have multiple identities. In the e-portfolio ecosystem, a user may identify as a 'student'. Each user also possesses a self-sovereign identity, which they can utilize to log in and sign up for *AO*s in the e-portfolio ecosystem. *STU*s retain full control over their data authorization and can determine which data are used for reviews.
- Reviewer (*RE*): In this scenario, the *RE* represents the university. *RE*s are responsible for reviewing *HS* student profiles and have direct access to trusted data that have been authorized by students from their *HS* and *AO*.

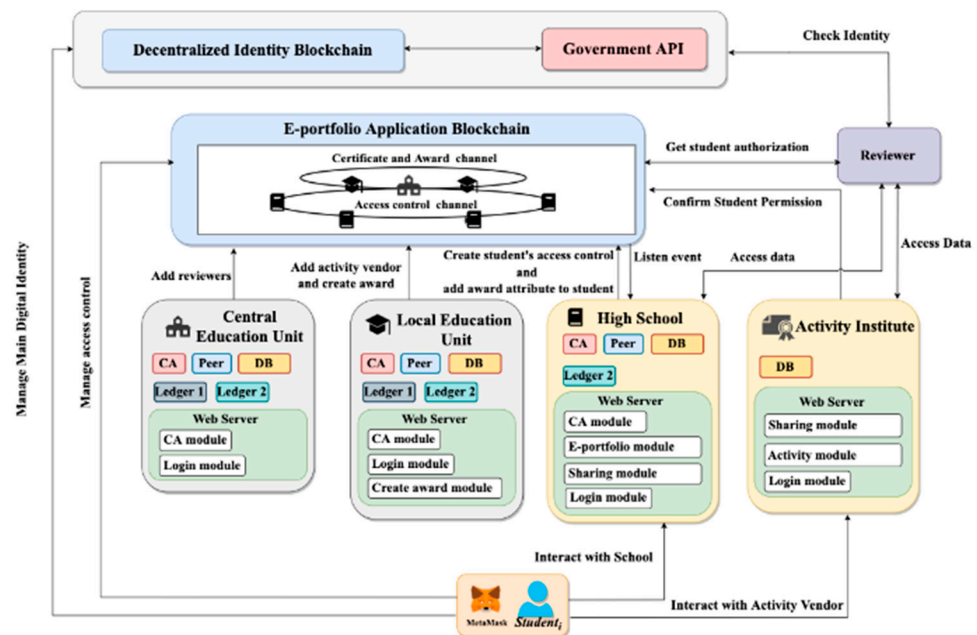


Figure 5. E-portfolio ecosystem architecture.

3.2. Smart Contract Design

Smart Contract in EApp-Chain

Figure 6 illustrates the relationship diagram featuring roles and smart contracts within the EApp-chain. The EApp-chain hosts three active smart contracts: the certificate authority manager contract (*CAMcon*), award manager contract (*AMcon*), and access control contract (*ACcon*). Specifically, the *CAMcon* and *AMcon* are deployed in the *CaAch*, while the *ACcon* takes residence in the *ACch*. This deployment configuration ensures a structured and efficient distribution of functionalities within the EApp-chain, delineating the roles and interactions of each smart contract in facilitating the operations of the e-portfolio ecosystem.

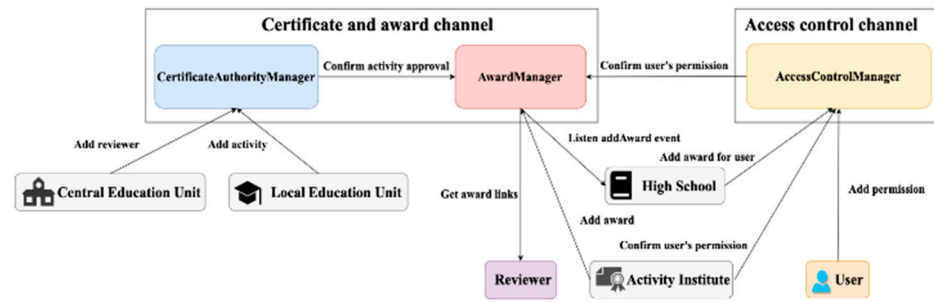


Figure 6. Smart contracts and role relationships in App-chain.

In the HLF framework, when an organization becomes a participant in two channels, it gains the capability to store the ledger data of both channels concurrently. This capability allows smart contracts within the organization to execute cross-channel read operations. For instance, when both the LEU and the CEU join two channels simultaneously, the AMch deployed in both entities can access and read the permissions of STUs stored in the ACcon. This cross-channel read functionality enhances the flexibility and interoperability of smart contracts, allowing them to seamlessly access data across multiple channels within the HLF network.

- Certificate Authority Manager Contract (CAMcon)

CAMcon is responsible for managing information related to REs and approved activities. The diagram of CAMcon is presented in Figure 7, and it stores the structure of reviewer and activityInfo in key-value pairs. When invoking smart contracts, X.509 certificates can be used to extract user identities and attributes, such as $P_{u}ks$ and roles. These attributes can be utilized to restrict contract function execution to specific users.

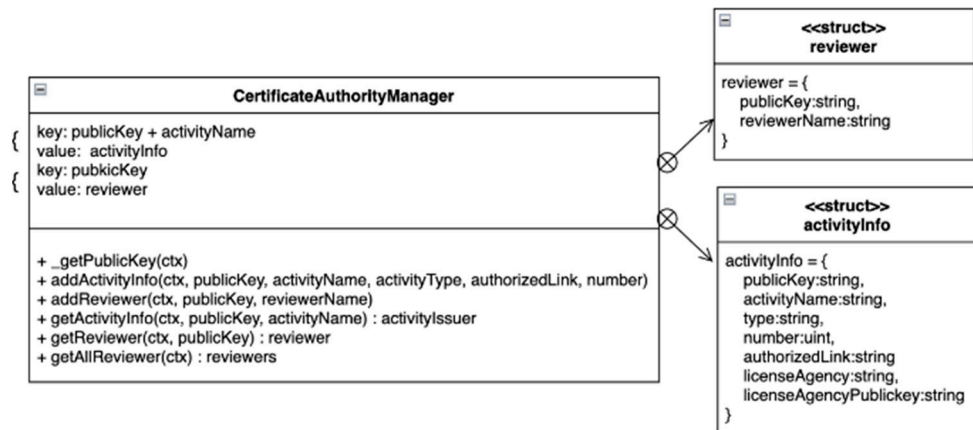


Figure 7. CAMcon diagram.

The addActivityInfo function allows the administrator of the LEU and CEU to add approved AOs to CAMcon, which can then be used by AMcon to grant awards to users. On the other hand, the addReviewer function is used by the CEU to add an RE. HSs can access the $P_{u}ks$ of all REs through the getAllReviewer function, which can help users generate authorization for REs.

- Award Manager Contract (AMcon)

AMcon is used to manage award records for users. The diagram of ACcon is presented in Figure 8 and the structure of awardInfo is stored in AMcon, as shown in Table 1. The award function can be invoked by the AO, and then the award function will invoke the getActivityInfo function in CAMcon to check whether the AO has the right to give an award. After a successful invocation, the addAward event is generated to notify the HS where

the user is registered. The `getAccessLink` function is used to enable the reviewer to obtain the data access link authorized by the user. First, *AMcon* reads the world state about the user’s awards and calls `getPermission` in *ACcon* to obtain the user’s permission. Finally, authorized access links are returned to the *RE*.

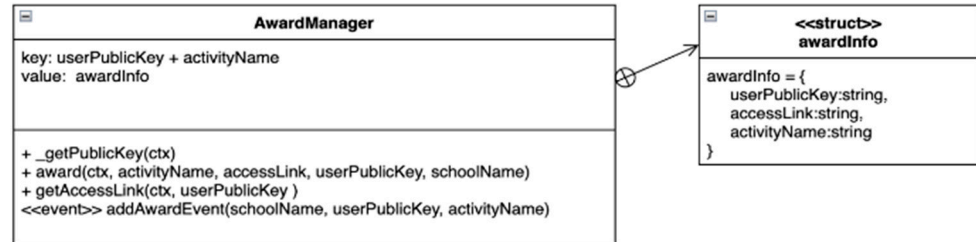


Figure 8. AMcon Diagram.

Table 1. Table of awardInfo structure.

Variable	Type	Description
userPublicKey	string	The public key of the user
accessLink	string	Link to access the award
activityName	string	Name of activity

- Access Control Contract (*ACcon*)

ACcon is primarily responsible for managing the access control of users’ data. The diagram of *ACcon* is presented in Figure 9. Each user is associated with an `accessControl` structure in *ACcon*, as shown in Table 2. This structure comprises two objects, namely `awardAttributes` and `permission`, which only the user can modify. The `unconfirmAwards` object stores the awards added by organizations but not yet confirmed by the user. When an organization receives the `addAward` event, it confirms that the user is a member of the organization and then invokes the `addAwardForUser` function in *ACcon* to add the award to the user’s `unconfirmAwards` object. The user can subsequently call the “`confirmAward`” function to transfer the award from the `unconfirmAwards` object to the `awardAttributes` object.

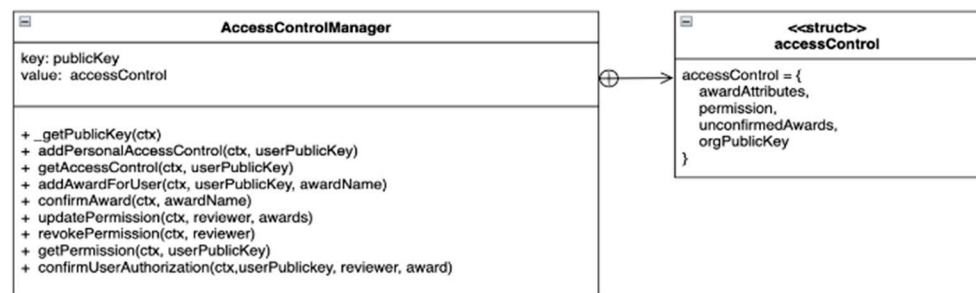


Figure 9. ACcon diagram.

Table 2. Table of accessControl structure.

Variable	Type	Description
awardAttributes	list	Award is confirmed by the user
permission	dict	User’s permission
unconfirmedAwards	list	Award is added by the organization
orgPublickey	string	The public key of the organization

Once the `confirmAward` function is executed successfully, the user can manage the authorization of their award using the `updatePermission` and `revokePermission` functions.

The permission object, which has a dictionary-like structure, stores the authorization state. With blockchain technology and smart contracts, the authorization of a user's award cannot be tampered with by anyone else, ensuring that each user's award can be traced and trusted.

3.3. Software Component

3.3.1. Metamask

Metamask [48] enables effortless interaction with decentralized applications (Dapps) on the Ethereum network. It safely manages users' P_rk s within the browser and provides an API that allows websites to access Ethereum account information and related blockchain data. Furthermore, Metamask allows websites to request user actions, such as signing messages and generating digital signatures, all while upholding strong security measures. This functionality enhances the user experience when engaging with various Dapps across the Ethereum network.

Figure 10 illustrates the process of signing transactions using Metamask. It is crucial for the user to trust the smart contract provided by the visited website and grant permission to use the Metamask API for actions involving their account and P_rk . With this approach, Dapp developers can focus on smart contract and webpage development without worrying about connection issues between users and the Ethereum blockchain nodes.

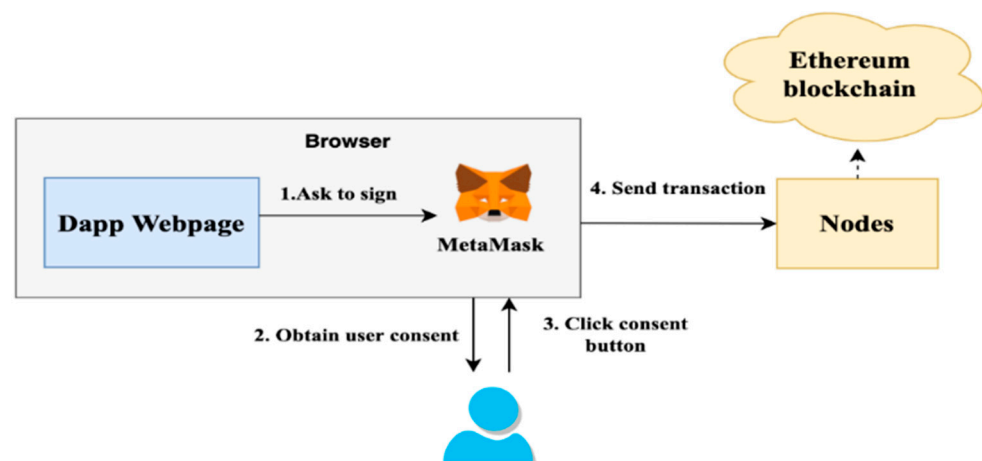


Figure 10. Signing the Ethereum transaction with Metamask.

In our proposed self-sovereign identity infrastructure, users utilize Metamask to store P_rk s that manage their primary identity in the DID-chain. Since Metamask does not support signing transactions in HLF, we utilize offline signing to carry out transactions on the EApp-chain.

3.3.2. JSON Web Token (JWT)

The JWT is employed as a secure method for validating data exchange between parties via APIs in our proposed system. JWTs utilize a digital signature, signed with a password (HMAC) or a P_rk/P_rk (RSA/ECDSA), ensuring data authenticity. Unlike traditional stateless HTTP protocol requests, in which each operation is independent, JWT encapsulates the user's status for self-preservation.

In our system, JWTs are fundamental to data sharing. RE s, prior to accessing a user's data, seek an access identity token from resource servers such as educational institutions and extracurricular groups. This access identity token, presented as a JWT, represents an authorized identity granted by the resource servers. By validating the token's expiration and digital signature, the server can effortlessly access the user's current status embedded in the token, streamlining the authentication and authorization process for data retrieval.

4. Implementation

4.1. DID-Chain

We followed the method of [7] to implement our DID-chain, which encompasses three main workflows: registering for a decentralized identity, uploading the EApp-chain P_rk , and uploading registration materials.

4.2. EApp-Chain

4.2.1. Creating Access Control Instance

Figure 11 illustrates the workflow of establishing an $ACin$ within the $ACch$, particularly when a user logs in at an HS within the e-portfolio ecosystem. In this context, the user is comparable to a student. The workflow unfolds as follows:

1. The user creates a digital signature using their primary identity and sends it to the HS for identity verification;
2. Upon receiving the digital signature, the HS verifies it and retrieves the user's $PIcon$ address and registration details from the event logs in the DID-chain;
3. The HS decrypts the user's registration materials to access their information, encompassing the EApp-chain P_{ik} and their common name. It utilizes a Certificate Signing Request (CSR) to enroll the user with Fabric CA;
4. The HS maintains the user's X.509 certificate, which serves as the basis for creating transaction proposals on behalf of the user;
5. The HS then establishes an $ACin$ for the user in the $ACch$, empowering them to use their primary identity to log in to the HS and manage access to their EApp-chain identity. By adhering to this systematic workflow, the HS ensures a secure authentication process, granting the user access to the requisite resources within the e-portfolio ecosystem.

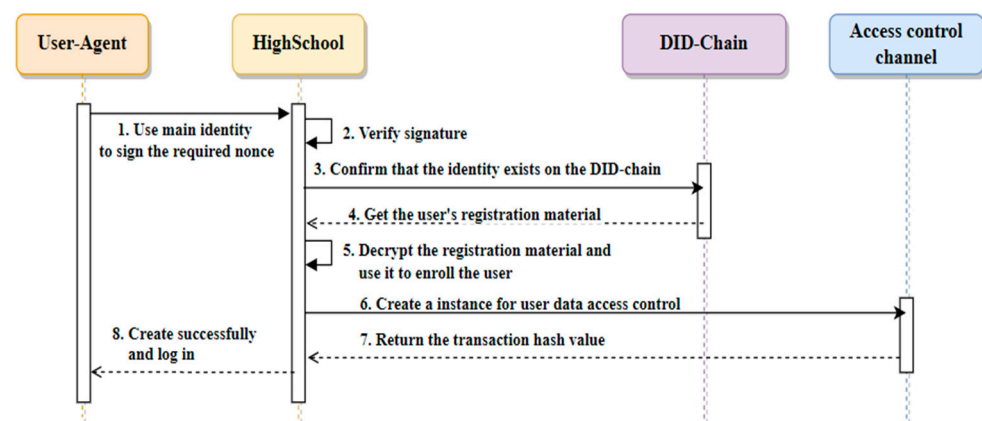


Figure 11. Creating access control instance flow.

4.2.2. Login with Digital Signature

Once the $ACin$ is successfully created, the user gains an identity that can be utilized within the e-portfolio ecosystem. The login process encompasses two distinct instances: the initial login and general login.

Figure 12 illustrates the login flow when the user first accesses another organization. In the e-portfolio ecosystem, this workflow takes place when a student participates in an activity.

The login process involves the following steps:

1. The user uses their primary identity to sign a required nonce and generate a digital signature, which is sent to the AO ;
2. The AO verifies the signature and queries the database to determine whether the user is logging in for the first time;

3. If the user’s EApp-chain $P_{u,k}$ does not exist in the database, the AO creates another nonce and asks the user to sign it with their EApp-chain P_r,k ;
4. The AO uses the EApp-chain $P_{u,k}$ to verify the signature and find the $ACin$ in the app-chain;
5. The AO records the relationship between the primary identity and the EApp-chain $P_{u,k}$ in the database to record that the user has finished the initial login.

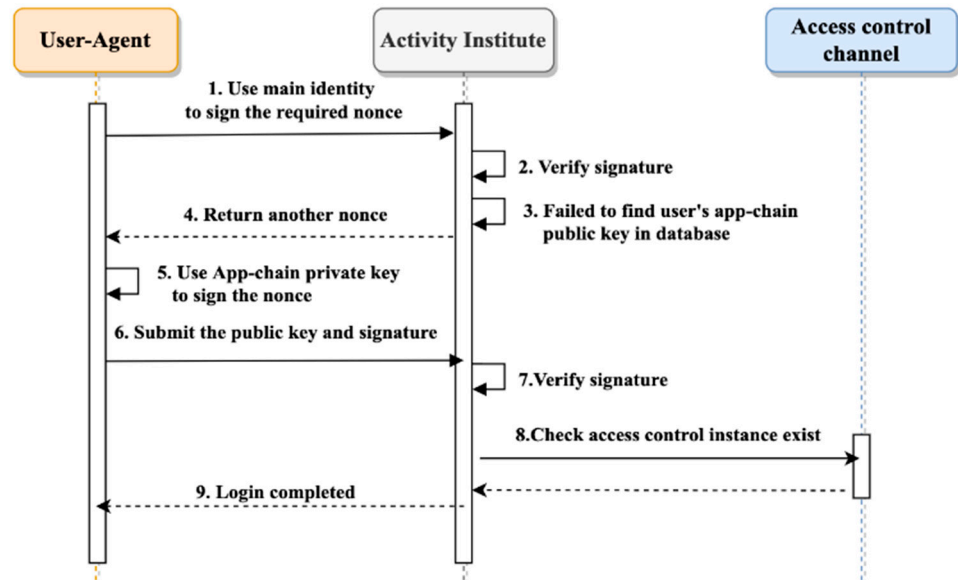


Figure 12. Initial login flow.

It is crucial to understand that until the initial login is completed, the AO does not know the user’s identity within the EApp-chain.

This login flow ensures the security of the e-portfolio ecosystem by verifying the user’s identity and recording their access to the EApp-chain.

Figure 13 shows the general login flow for the e-portfolio ecosystem. Once the user has completed the initial login process and the organization has recorded their identity within the EApp-chain, they can use their primary identity to generate a digital signature for general login.

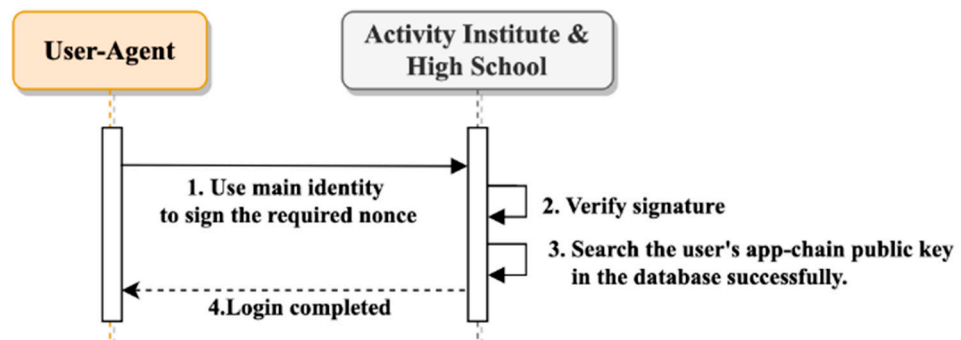


Figure 13. General login flow.

The general login flow involves the following steps:

1. The user creates a digital signature with their primary identity and forwards it to the AO;
2. The AO verifies the signature and uses the EApp-chain $P_{u,k}$ to find the $ACin$ in the EApp-chain;

- The *ACin* determines if the user possesses the required access rights to access the requested resources.

If the user is authorized, the *AO* grants them access to the requested resources.

By using a digital signature for login, the e-portfolio ecosystem ensures the security of user data while providing a convenient and efficient login experience.

4.2.3. Activity Organization Creates an Award for User

Figure 14 outlines the process for an *AO* to create awards for a user in the e-portfolio ecosystem. As *AOs* are not peer nodes in the EApp-chain, they must apply to *LEUs* to gain eligibility for award creation. The *LEU* conducts an audit of the application information submitted by the *AO*, including details such as activity name, type, number of awards to be created, and authorization API. Upon approval, the *AO* is included in the *CaAch*. Subsequently, it can invoke the createAward function in the *AMcon*, facilitating the generation of credible award records stored securely in the blockchain ledger.

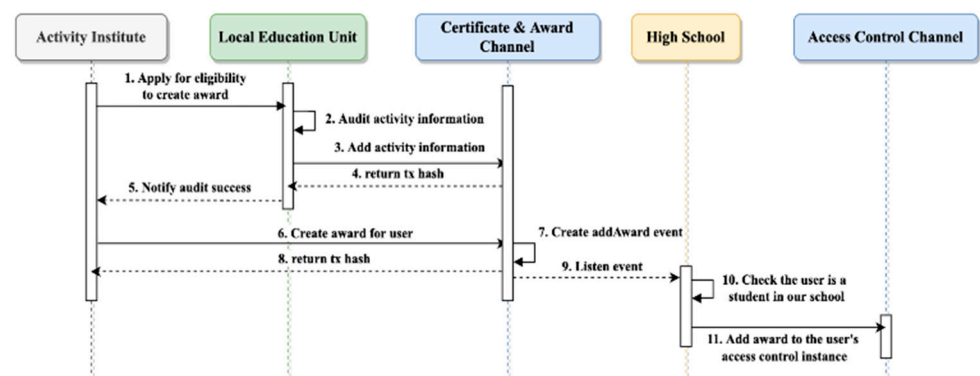


Figure 14. Creating award flow.

The award creation process involves the following steps to ensure the credibility of awards within the e-portfolio ecosystem:

- The *AO* applies to the *LEU* for eligibility to create awards;
- The *LEU* reviews the submitted details and grants approval to the *AO* to create awards;
- The *AO* triggers the createAward function in the *AMcon*, generating credible award records securely stored in the blockchain ledger;
- An addAward event is triggered to notify the *HS* attended by the user about the recently created award;
- The *HS* retrieves award information from the event, including the activity name and the user's P_{uk} ;
- The *HS* invokes the addAwardForUser function in the *ACcon*, adding the award to the user's unconfirmedAward list;
- The *HS* notifies the user through email, prompting them to confirm the newly added award.

This robust process ensures that only credible awards are added to a user's e-portfolio, and users are promptly notified of any new awards. Establishing a secure and reliable system for award creation and verification enhances the overall credibility of awards within the e-portfolio ecosystem, fostering increased user participation in activities.

4.2.4. Access Control

Within each EApp-chain, an *ACin* functions as a repository for the user's authorization actions. The user, utilizing their ecosystem identity, has control over the *ACin* stored within the EApp-chain. Leveraging the decentralization inherent in blockchain technology, the user's authorization status remains secure from tampering or forgery, provided the

user safely safeguards their P_rk . This ensures the integrity of the authorization system, enhancing trust in the user’s control over their access permissions within the EApp-chain.

In Figure 15, the diagram illustrates how users manage access permissions within the EApp-chain. To fulfill the SSI requirement, offline signing is utilized instead of storing the P_rk in the organization or mandating users to install the SDK. This process encompasses various user operations for managing their $ACins$, including functions like `updatePermission`, `revokePermission`, and `confirmAward` within the $ACcon$. For clarity, let us delve into the workings of the `updatePermission` function as an illustrative example.

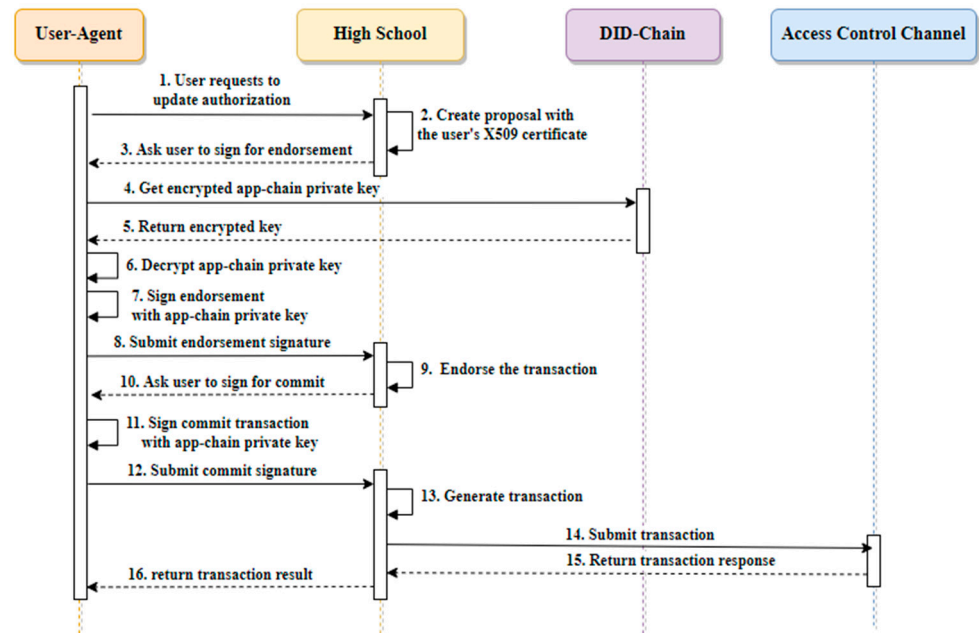


Figure 15. Authorization flow.

The user initiates the authorization process by determining who is granted access and specifying the data consented for review. Upon receiving the user’s request, the HS generates a transaction proposal utilizing both the user’s X.509 certificate and the pertinent request parameters. Following this, the user is prompted to sign the proposal to facilitate the endorsement process.

Afterward, the user acquires the EApp-chain P_rk , encrypted by the primary identity from the $PIcon$. Decrypting the key, the user signs the proposal. Upon receiving the signed proposal, the endorsement node at the HS evaluates and endorses it. During the simulation of the proposal’s execution, the endorsement node verifies the signature against the P_rk in the X.509 certificate and confirms the user’s right to invoke the smart contract.

In the endorsement process, if the user executes the evaluate operation without necessitating changes to the world state in the ledger, the result is obtained after the endorsement. On the other hand, if the transaction involves changes to the world state, the user needs to sign the proposal again to commit the transaction.

Finally, the HS submits the transaction, incorporating the commit signature, to the orderer node. Upon receiving the orderer response, the transaction is recorded in the ledger, thereby concluding the authorization process. This meticulous approach ensures the integrity and security of the authorization actions within the EApp-chain.

4.2.5. Data Sharing

In Figure 16, the flow illustrates how an RE can access review data authorized by the user. Initially, the RE initiates the process by invoking the `getAccessLink` function in the $AMcon$ to retrieve the access link for the awards. Within the execution of the `getAccessLink` function, the $AMcon$ internally calls the `getPermission` function. This function facilitates

obtaining the user’s authorization for the *RE* and subsequently returns the relevant access link for the awards, adhering to the permissions granted by the user.

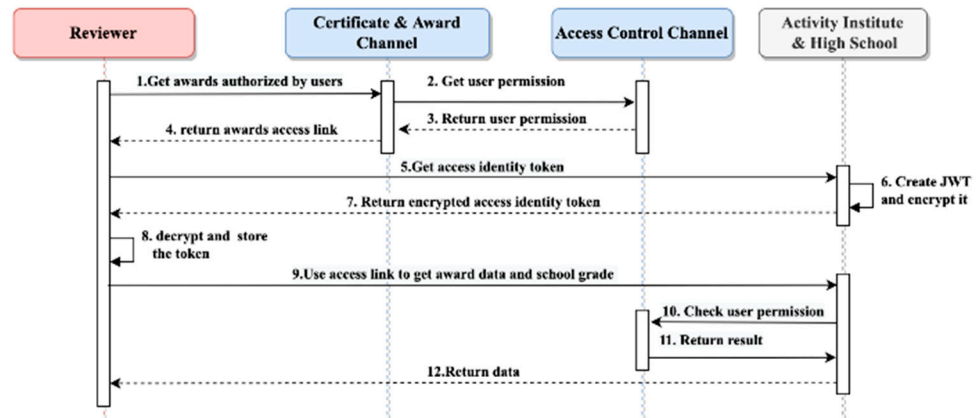


Figure 16. Data sharing flow.

Following this, the *RE* provides their EApp-chain P_{uk} to the resource server for authentication, aiming to acquire an access identity token. The resource server, in response, generates the access identity token and encrypts it using the applicant’s P_{uk} . Only the applicant who holds the corresponding P_{rk} can decrypt and access the token.

Table 3 outlines the structure of the access identity token, featuring a crucial parameter known as expiresIn. This parameter is vital for the system’s security, determining the duration during which the authorized identity remains valid. It is imperative for the applicant to securely safeguard the token to prevent identity theft. In scenarios demanding high data security, data providers can set a brief expiration time for frequent authentication, thereby minimizing the risk of token theft.

Table 3. The structure of access identity token.

Name	Type	Description
issueAt	timestamp	The issuance time of the token.
expireIn	timestamp	The expiration time of the token.
subject	string	The subject of the token refers to the identity represented by the token, for example, the public key of the reviewer.
issuer	string	The issuer of the token, for example, the public key of the activity organization.

With the access identity token and access link in hand, the reviewer can employ these credentials to access the data authorized by the user from the data providers. Prior to releasing the data, the data providers validate the user’s permission status in the *ACch*. This meticulous process ensures that the *RE* efficiently obtains the user’s review data in a secure and reliable manner.

5. DApp Demonstration

5.1. DID-Chain

Creating a Self-Sovereign Identity

In the DID-chain, two identity types—users and entities—register with real-name authentication, each submitting forms to apply for an SSI. Once registered, they bind their Ethereum accounts as primary identities to manage their *PIcons*. Entities also require additional encryption of user data using P_{uk} via the Metamask API.

After binding, users and entities receive a *PIcon* for managing their identity across ecosystems. For example, in the e-portfolio ecosystem, users encrypt their CSR with the organization’s P_{uk} and record it on the blockchain. Since HLF lacks browser wallet support,

the EApp-chain Prk is encrypted with the primary identity's $P_{u,k}$ and stored in the $PIcon$. For a detailed demonstration, see [7].

5.2. EApp-Chain

5.2.1. Creating Access Control Instance

After submitting their CSR, users can log into the chosen organization using their primary identity. The organization verifies the user's digital signature, enrolls them using the stored CSR, and creates $ACins$ with profile attributes like rank, grade, and e-portfolio files. This allows users to manage their data authorization.

5.2.2. Users Login for the First Time to Another Organization

For their initial login to another organization, users provide the ecosystem's $P_{u,k}$, decrypt the EApp-chain P,k with their primary identity, and sign a nonce for identity validation. After this, their subsequent logins only require the primary identity.

5.2.3. Activity Organizations Create Awards for Users

Activity organizations must submit applications to the local education unit to obtain qualifications for award creation, as shown in Figure 17. After the organization requests consent, the local educational unit evaluates and approves the application. This approval increases the award's value and credibility, as illustrated in Figure 18. Once the local educational unit grants approval for the certificate, the organization can create and issue awards to users, as depicted in Figure 19.

The screenshot shows a web interface for the 'Local Ministry of Education'. At the top, there is a blue navigation bar with the text 'Local Ministry of Education' on the left and three icons with labels: 'issue', 'apply', and 'logout'. Below the navigation bar is a white box titled 'Apply for award issuance'. Inside this box, there are four input fields: 'Activity Name' with the placeholder 'Enter Activity Name', 'Authorized link' with the placeholder 'Enter authorized link', 'Type' with a dropdown menu showing 'Select activity type', and 'Number' with the placeholder 'Enter Number of issued'. At the bottom of the white box is a blue 'Submit' button.

Figure 17. Applying for certificate issuance.

The screenshot shows a web interface for the 'Local Ministry of Education'. At the top, there is a blue navigation bar with the text 'Local Ministry of Education' on the left and four icons with labels: 'audit', 'issue', 'apply', and 'logout'. Below the navigation bar is a white box titled 'Require for consent'. At the top of this box are two input fields: 'Enter Organization Name' and 'Enter Activity Name', followed by a grey 'Pass' button. Below these fields is a table with the following data:

Organization	Name of activity	Type	No.	State
0x889735777f51c84272a7feb0d763280179a529a9	Science Exhibition	contest	10	true
0xc3a86d7375ff7b690c065022bbbe114aee2320c7	Python Django	course	10	false

Figure 18. Auditing activity/organization application.

The screenshot shows a web form titled "Create Award for user" with a close button (X) in the top right corner. It contains three input fields: "User" with the value "04c38b5740dac98953ee727d3dfa6bb536d37a43i", "Award Name" with the value "National High School Debating Competition", and "Access Link" with the value "http://localhost:3002/E-portfolio/dataStorage/". Below the fields is a large green "submit" button.

Figure 19. Creating an award for user.

5.2.4. Confirming Award and Authorization

In Figure 20, we observe the Confirm Award Page. After an activity organization establishes an award for the user, the user has the option to confirm and receive the award. Only confirmed awards can be utilized for authorization reviews, as shown in Figure 21. To achieve this, the user decrypts the EApp-chain P_rk using their primary identity and uses it to confirm the award, overseeing the authorization of their own data, as depicted in Figure 22.

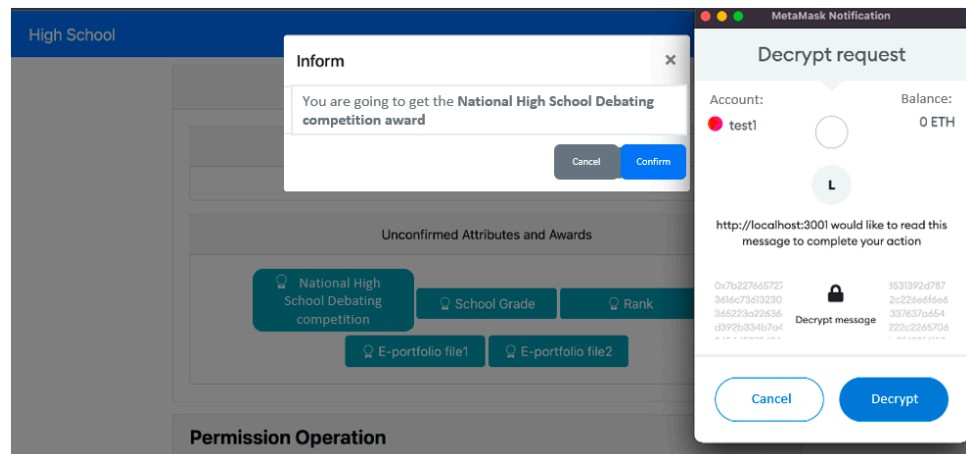


Figure 20. Confirming award.

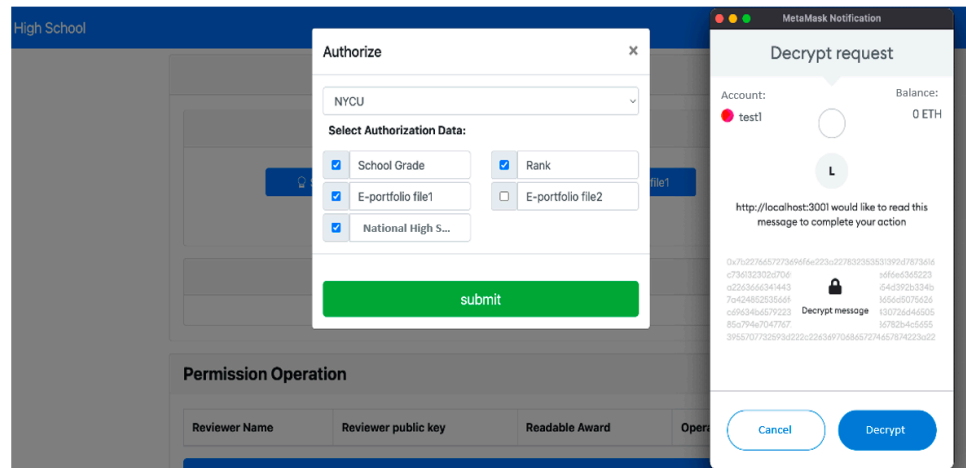


Figure 21. Authorization.

Permission Operation

Reviewer Name	Reviewer public key	Readable Award	Operation
NYCU	40d6a1b6903afac76b0d87b05dae5fca1a3d6f06e6	School Grade, Rank, E-portfolio file1, National High School Debating Competition	Update Revoke

Authorize

Figure 22. Authorization management.

5.2.5. Data Sharing

University reviewers receive a roster of students who have undergone evaluations by the Ministry of Education. They acquire a valid access link from the *AMcon*, ensuring that reviewers can exclusively access review data that have been consented to by the students, as illustrated in Figure 23.

Personal Information
from DCS

Student Name: **Xiao-Ming, Wang**

DID address: **0xc3a86D7375FF7b690C065022bbbe114aee2320c7**

High School: **DCS**

E-portfolio public key :
040d6a1b6903afac76b0d87b05dae5fca1a3d6f06e650d434c

School grade
from DCS

Subject grades:

NO	Academic Year Semester	Subject	Grade
1	1081	Math	87
2	1081	English	72
3	1081	Chinese	75
4	1081	Science	95
5	1081	Society	70
6	1082	Math	88
7	1082	English	82
8	1082	Chinese	92
9	1082	Science	94

Rank:

Academic Year Semester	Rank/ Total Students
1081	52/272
1082	26/272

Figure 23. Review page.

6. Experimental Evaluation

6.1. Test Environment

Our analysis was conducted utilizing the computer specifications outlined in Table 4. The deployment of the *DID-chain* and *EApp-chain* was achieved using Docker, with blockchain nodes managed through Docker Compose. Each Docker container was allocated four GB of RAM to use. These two chains operated in distinct subnets, virtualized by Docker Compose. For the performance assessment of the web applications, Apache JMeter [49], an open-source load testing software, was employed. The simulation involved a scenario in which numerous users sent requests and interacted with the blockchain network.

Table 4. Computer specification.

CPU	Cores	RAM
Intel i9-10900 2.80 GHz	10	64 GB

We select throughput and latency as our primary evaluation metric to assess the system's ability to manage high volumes of requests effectively. Taiwan has a total of 580,000 high school students, who are likely to generate the most activity on the platform. Assuming that each student uploads files once daily, the system would need to handle around seven requests per second on average. At peak hours, we assume the request rates will be ten times [50] more than the average, with 70 requests per second. This throughput measurement helps determine whether the system can support consistent, reliable access and processing for a large user base without performance degradation.

6.2. DID-Chain

In this section, we describe the network setting parameters of the *DID-chain* and evaluate the gas consumption of contract functions and throughput.

6.2.1. Network Setup

To build our proposed *DID-chain* using the Ethereum blockchain, we utilized the Geth docker image. The Ethereum blockchain's setting parameters are detailed in Table 5. Each government department responsible for identity-related matters acted as a node to form a public blockchain. Users and organizations could interact with Ethereum nodes using web3.js. For our *DID-chain*, we implemented the Proof of Authority (PoA) consensus algorithm. The PoA algorithm designates a set of accounts as authorities authorized to seal received transactions to generate a new block. Compared to the Proof of Work (PoW) algorithm, the PoA algorithm not only saves energy but also provides better TPS. Additionally, we used a lower mining difficulty and a shorter block time as all the blockchain nodes were trusted government departments.

Table 5. DID-chain setting parameters.

Parameter	Parameter Setting
Number of nodes	5
Number of miners	3
State database	LevelDB, key-value storage
Consensus mechanism	Proof of authority (PoA)
Difficulty	0×1

6.2.2. Gas Consumption

Gas refers to the resources required for computation on the Ethereum network. Gas consumption is related to modifying the state in the blockchain and the complexity of the contract function. Table 6 provides information on the usage frequency and gas consumption of deploying contracts and calling contract functions. It can be observed that operations

with higher gas consumption, such as deploying contracts and the `bindUser` function, are used less frequently. In contrast, functions with a higher usage frequency, such as `setEncryptMaterial` and `setEncryptAppPrivateKey`, have lower levels of gas consumption.

Table 6. Usage frequency and gas consumption.

Usage Frequency	Function/Contract	GAS USED
Once	New identity manager contract	1,744,419
Once per user	New personal identity contract	689,393
Once per user	<code>createIdentity</code>	51,024
Once per user	<code>bindUser</code>	667,914
Once per ecosystem	<code>setEncryptMaterial</code>	140,864
Once per ecosystem	<code>setEncryptAppPrivateKey</code>	65,367

6.2.3. Function Throughput

Figure 24 displays the throughput of smart functions at different send rates. We simulated sending 50, 100, 150, and 200 requests per second to the Ethereum blockchain node using JMeter. The throughput was found to be related to gas consumption, in which a higher level of gas consumption indicates more blockchain state changes and a lower throughput. The `bindUser` function had the lowest throughput among all the tested functions, as it could only be processed four times per second. The `bindUser` function involves creating and transferring the *PIcon* to the user, resulting in a large write operation to the blockchain. Despite its low throughput, the `bindUser` function is only executed once per user to obtain the *PIcon*.

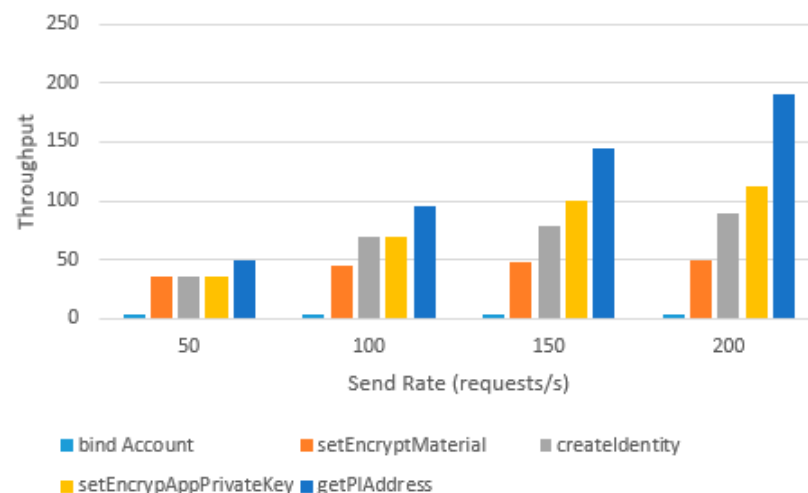


Figure 24. Throughput vs send rate.

6.3. EApp-Chain

In this section, we describe the network setting parameters of the EApp-chain and analyze the throughput of contract functions.

6.3.1. Network Setup

We utilized Fablo [51], an *HLF* network creation and management tool, to build our *HLF* network. Using Fablo, we created a test network and deployed smart contracts according to our requirements. Table 7 presents the configuration parameters for the *HLF* blockchain. The configuration involves a total of three nodes, with two nodes joining the *CaAch*, while all nodes participate in the *ACch*. To avoid single orderer node failure, we utilized the Raft consensus algorithm, ensuring the network remains available as long as more than half of the orderer nodes are alive. We also set the `BatchTimeout` to 0.5 s to pack transactions into blocks quickly.

Table 7. E-portfolio App-chain parameters.

Fabric Version	2.2
Database	CouchDB
Number Channel	2
Consensus algorithm, Number of instances	Raft, 3
Number of peer nodes in the <i>ACch</i> ,	3
Number of peer nodes in <i>CaAch</i> ,	2
BatchTimeout	0.5 s

6.3.2. Function Throughput

To provide clarification, Figures 25 and 26 depict the performance of smart contract functions on both the *CaAch* and the *ACch*. The measurements were taken under distinct sent rates. The results reveal that the read and write throughputs on both channels are similar. Functions requiring cross-channel read operations, such as *getAccessLink*, exhibit no substantial variance in throughput between the two channels. Additionally, the throughput of read operations, including *getAccessLink*, *getReviewer*, and *getAccessControl*, is greater than that of write operations. As the sending rate increases, read operations demonstrate a linear increase in throughput, while the throughput of write operations remains constant at around 250 transactions per second.

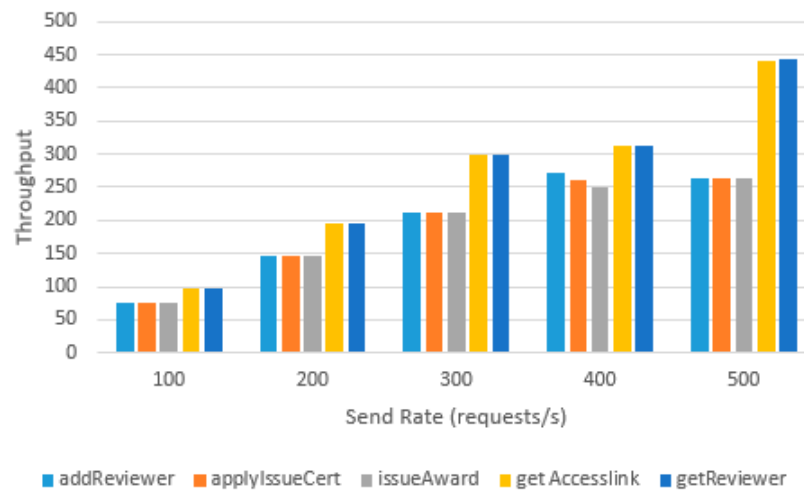


Figure 25. Function throughput on *CaAch*.

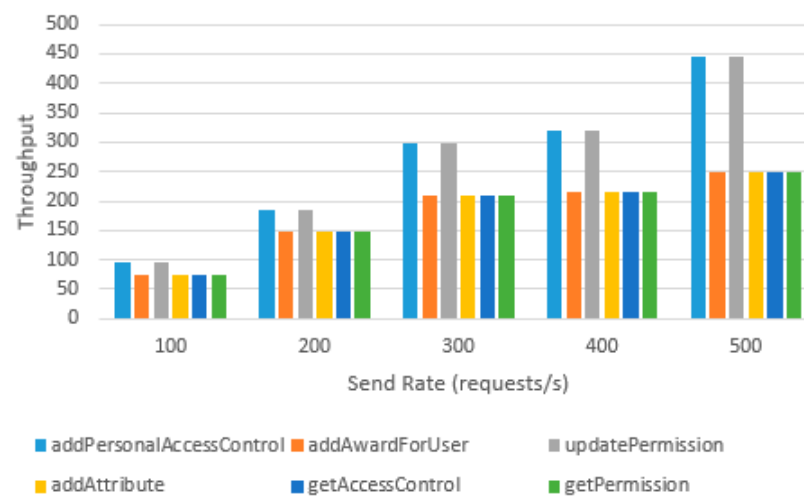


Figure 26. Function throughput on *ACch*.

6.3.3. Throughput and Latency

In the experiment, we gradually raised the sending rate from 100 to 1000 to assess user performance. As depicted in Figure 27, the data show that, for write operations at a sending rate of 400, the latency stays under two seconds, with a throughput close to 250 transactions. Likewise, for read operations at a sending rate of 600, the latency consistently remains under two seconds, achieving a throughput of around 450 transactions.

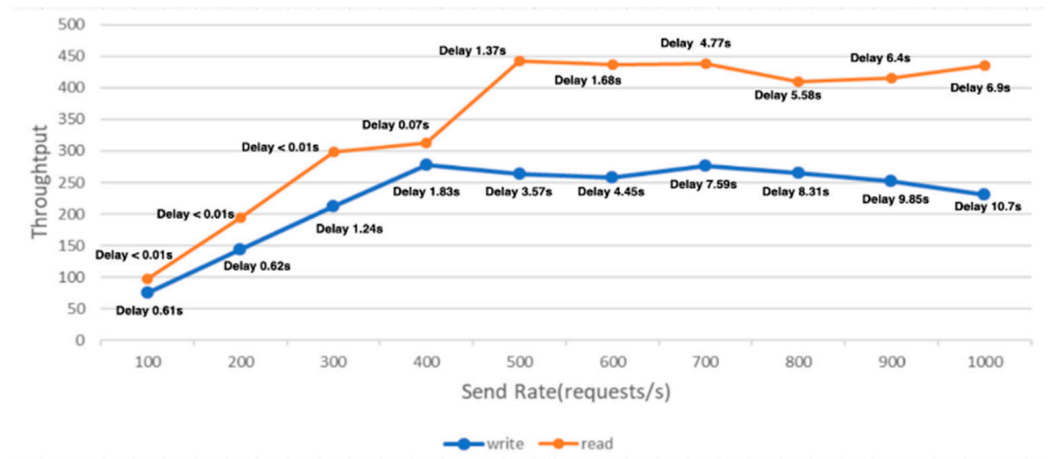


Figure 27. Throughput and latency.

6.3.4. Offline Sign Latency

We conducted a comparison of the latency between using offline signatures and escrow P_rks , by simulating users updating their permissions 1000 times using both methods. The results are presented in Table 8, which indicates that the difference in latency between the two methods is approximately 18 ms. This means that adopting offline signatures does not affect the user experience significantly and is more secure than escrow P_rk . It also helps achieve the goal of SSI.

Table 8. Latency of offline signing vs escrow private key.

Method	Function	Number	Average Latency (ms)
Offline sign	updatePermission	1000	607.85
Escrow private key	updatePermission	1000	625.76

6.4. System Comparison

In this section, we will conduct a comparative analysis with several research studies that share a common research interest. The outcomes of this comparison are presented in Table 9 for reference and evaluation. We employ six evaluation metrics to assess various schemes: access control model, data authorization, extracurricular activities, decentralized identity, parallel execution, and off-chain storage. The access control model elucidates data utilization methods. Data authorization reflects data trustworthiness. Extracurricular activities pertain to the system’s capacity to record additional activities. Decentralized identity evaluates system security. Parallel execution addresses the system’s ability to manage concurrent instructions, crucial for accommodating multiple users simultaneously. Lastly, off-chain storage enhances system efficiency by preventing prolonged execution times that may result from storing large data solely on-chain, necessitating a balance between on-chain and off-chain data storage.

Table 9. A comparison of the results between this paper and previous studies.

	This Paper	[40]	[41]	[42]	[43]	[44]	[45]
Access control model	ABAC	A-RBAC	ABAC	N/A	N/A	N/A	N/A
Data authorization	True	True	True	True	True	False	True
Extracurricular activities	True	False	False	True	False	False	False
Decentralized identity	True	True	False	True	False	True	True
Parallel Execution	True	False	False	False	False	False	False
Off-Chain Storage	True	True	False	True	True	True	True

Stuchain [40] is an innovative system that integrates two access control models, ABAC and RBAC, under the implementation framework of HLF. Within the Stuchain platform, teachers possess the capability to meticulously record and manage students' learning histories. Conversely, students have the authority to selectively grant access to their data to specific educational institutions that demonstrate an interest in their academic progress. It is important to highlight that students' learning histories are securely stored within a traditional database infrastructure, as opposed to being stored on a blockchain. Furthermore, given that these data originate from teachers, they may not encompass a comprehensive overview of all facets of students' interests. Consequently, educational institutions may not gain a complete understanding of each student's individual interests solely through these data.

In a paper [41], authors proposed an e-portfolio system that was implemented using HLF. The e-portfolio system employs seven chaincodes: CPEM_C manages teacher evaluations and student targets for department administrators. VAAMC_C handles visitor access and file update authorizations. AUMSEAC_C oversees student e-portfolio updates, while CMSEAE_C is dedicated to course management system evaluations. IMTCTP_C manages course teaching process data, ERM_C handles student evaluation results, and IMSCPP_C deals with student course participation information. While this system effectively enables students to manage access control rights for their data and allows teachers to evaluate student learning history, it does have limitations. Notably, it does not provide a mechanism for recording extracurricular activities. This limitation could potentially hinder reviewers' ability to gain a comprehensive understanding of a student's overall development and capabilities.

Merlec et al. [42] proposed a system that operates on the Ethereum blockchain and is structured around four layers, with a central secure e-portfolio management layer designed to provide dependable and secure management functionalities. These functionalities encompass activities such as the following: managing membership enrollments, user profiles, role assignments, and decentralized enrollment procedures; efficiently managing the issuance and revocation of membership credentials; overseeing personal information within user profiles; and effectively administering user role assignments. Furthermore, it facilitates seamless connections between learners and educators, as well as job seekers and employers, placing a strong emphasis on safeguarding user privacy and ensuring the secure and reliable exchange of educational data. However, it is important to note that implementing the entire system on the Ethereum blockchain may result in high operational costs due to smart contract execution fees.

The PETS architectural framework, introduced in [43], is tailored for blockchain-based solutions in higher education. It links on-chain and off-chain data through a cohesive data engine, ensuring security, governance, and visibility. The framework standardizes access via APIs, fostering developer collaboration. Its data logistics platform automates data transfers, simplifying management. This architecture aids software architects in evaluating blockchain technologies and supports research on decision-making frameworks for blockchain systems.

EduRSS [44], as detailed in a research paper, is a blockchain-based system for securely storing and sharing educational records. It integrates blockchain, storage servers, and cryptography to create a reliable platform. Blockchain ensures data security and

reliability, with smart contracts managing storage and sharing. Educational records are stored off-chain in encrypted form, while their hashes are maintained on the blockchain, ensuring data integrity through periodic anchoring. EduRSS benefits educational institutions and individuals by offering a secure, cost-effective solution that enables efficient, privacy-preserving record management and sharing. Experimental results confirm its cost-effectiveness compared to similar solutions.

MOOCsChain [45], a blockchain-based scheme for secure storage and sharing within MOOCs, addresses the need for the reliable management of Electronic Learning Records (ELRs). It uses blockchain to protect learner data privacy without complex cryptography. MOOCsChain provides efficient conditional anonymity, robust security, and streamlined content sharing via smart contracts. ELR components are stored on the blockchain, with original data in the InterPlanetary File System (IPFS). Evaluation on the HLF 1.4 platform shows MOOCsChain's superiority, offering online learners and educators secure ELR management with efficient anonymity and strong security assurances.

7. Conclusions and Future Avenues for Exploration

7.1. Conclusions

This research introduces a groundbreaking self-sovereign identity-based infrastructure tailored to enhance personal information security control within the dynamic e-portfolio ecosystem. Addressing three pivotal requirements, the system aims to revolutionize the landscape of user identity, data source credibility, and user-controlled data authorization. Firstly, users are endowed with self-sovereign identities, fostering seamless login experiences and identity management across diverse ecosystems. Anchored in a decentralized identity chain, these identities are resilient against tampering or denial. Secondly, a collective audit by educational institutions ensures the credibility of data sources, granting issuance rights to only those activity organizations that successfully pass the stringent evaluation. Thirdly, users wield absolute control over data authorization, with their dynamic access rights recorded securely in the blockchain ledger. This privacy-centric approach, utilizing multiple blockchains and channels, shields user-owned data attributes from prying eyes. The system's ability to process transactions concurrently is heightened through the parallel execution of transactions on different blockchains, with the App-chain handling more frequent transactions efficiently.

7.2. Future Avenues for Exploration

Looking ahead, the research identifies a lacuna in the realm of identity management—specifically, the absence of a wallet accommodating multiple blockchain frameworks. Future endeavors will be directed towards crafting a cross-blockchain wallet, a versatile solution enabling users to manage *Prks* seamlessly across different blockchains, ensuring a convenient and comprehensive self-sovereign identity management experience.

Expanding the ecosystem's versatility is another key focus area, with plans to integrate additional organizations such as universities and LinkedIn. This expansion will empower users to present credible data on their education and experience to prospective employers. Tailoring users' profile presentation page for college reviewers or interviewers will offer customization options, facilitating a rapid assessment of applicants' strengths. Through continuous refinement and expansion, the overarching goal is to establish a robust, trustworthy, and user-friendly platform for personal information management.

Author Contributions: For this research article, the authors contributed as follows: Conceptualization, Y.-H.H., J.-Y.Y., C.-H.L. and S.-M.Y.; Methodology, Y.-H.H., J.-Y.Y., C.-H.L. and S.-M.Y.; Software, Y.-H.H., J.-Y.Y., C.-H.L. and S.-M.Y.; Validation, Y.-H.H., J.-Y.Y., C.-H.L. and S.-M.Y.; Formal Analysis, Y.-H.H., J.-Y.Y., C.-H.L. and S.-M.Y.; Investigation, Y.-H.H., J.-Y.Y., C.-H.L. and S.-M.Y.; Data Curation, Y.-H.H., J.-Y.Y., C.-H.L. and S.-M.Y.; Writing—Original Draft Preparation, J.-Y.Y.; Writing—Review and Editing, Y.-H.H.; Visualization, Y.-H.H., J.-Y.Y., C.-H.L. and S.-M.Y.; Supervision, S.-M.Y.; Project Administration, S.-M.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Science and Technology council, grant number NSTC 113-2410-H-A49-053-.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

This table shows the terms and their abbreviations used in this paper.

Table A1. Terms vs. abbreviations.

Term	Abbreviation
Blockchain	
Decentralized identity blockchain	DID-chain
E-portfolio application blockchain	EApp-chain
Channel	
Certificate and award channel	CaAch
Access control channel	ACch
Smart Contract	
Personal identity contract	PIcon
Certificate authority manager contract	CAMcon
Award manager contract	AMcon
Access control contract	ACcon
Stakeholder	
Central education unit	CEU
Local education unit	LEU
High school	HS
Activity organization	AO
Student	STU
Reviewer	RE
Other	
Self-Sovereign e-portfolio identity	SSeI
Certificate Signing Requests	CSRs
Access control instance	Acins
Hyperledger Fabric	HLF
Private key	P _r k
Public key	P _u k
JSON Web Token	JWT
Attribute-based access control model	ABAC
Role-based access control model	RBAC

References

1. Ministry of Education of Taiwan. Records of Learning Progress for Upper Secondary Education Stage Students 109 Academic Year Senior High School Information Guide. Available online: <https://www.hcvs.hc.edu.tw/resource/openfid.php?id=17272> (accessed on 1 August 2020).
2. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Bus. Rev.* **2008**, *4*, 21260.
3. Buterin, V. A next-generation smart contract and decentralized application platform. *White Pap.* **2014**, *3*, 2-1.
4. Daraghmi, E.Y.; Daraghmi, Y.A.; Yuan, S.M. MedChain: A Design of Blockchain-Based System for Medical Records Access and Permissions Management. *IEEE Access* **2019**, *7*, 164595–164613. [CrossRef]
5. Popoola, O.; Rodrigues, M.; Marchang, J.; Shenfield, A.; Ikpehia, A.; Popoola, J. A critical literature review of security and privacy in smart home healthcare schemes adopting IoT & blockchain: Problems, challenges and solutions. *Blockchain Res. Appl.* **2023**, *5*, 100178.

6. Antwi, M.; Adnane, A.; Ahmad, F.; Hussain, R.; Rehman, M.H.U.; Kerrache, C.A. The case of HyperLedger Fabric as a blockchain solution for healthcare applications. *Blockchain Res. Appl.* **2021**, *2*, 100012. [CrossRef]
7. Hsieh, Y.-H.; Guan, X.-Q.; Liao, C.-H.; Yuan, S.-M. Physiological-chain: A privacy preserving physiological data sharing ecosystem. *Inf. Process. Manag.* **2024**, *61*, 103761. [CrossRef]
8. Liu, X.; Muhammad, K.; Lloret, J.; Chen, Y.-W.; Yuan, S.-M. Elastic and cost-effective data carrier architecture for smart contract in blockchain. *Future Gener. Comput. Syst.* **2019**, *100*, 590–599. [CrossRef]
9. Kharche, A.; Badholia, S.; Upadhyay, R.K. Implementation of blockchain technology in integrated IoT networks for constructing scalable ITS systems in India. *Blockchain Res. Appl.* **2024**, *5*, 100188. [CrossRef]
10. Liao, C.H.; Lin, H.E.; Yuan, S.M. Blockchain-Enabled Integrated Market Platform for Contract Production. *IEEE Access* **2020**, *8*, 211007–211027. [CrossRef]
11. Xu, X.; Lu, Q.; Liu, Y.; Zhu, L.; Yao, H.; Vasilakos, A.V. Designing blockchain-based applications a case study for imported product traceability. *Future Gener. Comput. Syst.* **2019**, *92*, 399–406. [CrossRef]
12. Wamba, S.F.; Queiroz, M.M. Blockchain in the operations and supply chain management: Benefits, challenges and future research opportunities. *Int. J. Inf. Manag.* **2020**, *52*, 102064. [CrossRef]
13. Cao, S.; Foth, M.; Powell, W.; Miller, T.; Li, M. A blockchain-based multisignature approach for supply chain governance: A use case from the Australian beef industry. *Blockchain Res. Appl.* **2022**, *3*, 100091. [CrossRef]
14. Liao, C.-H.; Guan, X.-Q.; Cheng, J.-H.; Yuan, S.-M. Blockchain-based identity management and access control framework for open banking ecosystem. *Future Gener. Comput. Syst.* **2022**, *135*, 450–466. [CrossRef]
15. Zhang, P.; Schmidt, D.C.; White, J.; Dubey, A. Consensus mechanisms and information security technologies. *Adv. Comput.* **2019**, *115*, 181–209.
16. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; De Caro, A.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger fabric: A distributed operating system for permissioned blockchains. In Proceedings of the Thirteenth EuroSys Conference, Porto, Portugal, 23–26 April 2018; pp. 1–15.
17. The Historical Files of Students Learning in Upper Secondary Education. The Guidelines of Senior High Schools in 109 Academic Years. Available online: <https://www.hcvs.hc.edu.tw/resource/openfid.php?id=17272> (accessed on 30 October 2022).
18. Alam, F.; Chowdhury, H.; Kootsookos, A.; Hadgraft, R. Scoping e-portfolios to engineering and ICT education. *Procedia Eng.* **2015**, *105*, 852–857. [CrossRef]
19. Fedorova, E.P.; Skobleva, E.I. Application of blockchain technology in higher education. *Eur. J. Contemp. Educ.* **2020**, *9*, 552–571.
20. Li, C.; Guo, J.; Zhang, G.; Wang, Y.; Sun, Y.; Bie, R. A blockchain system for E-learning assessment and certification. In Proceedings of the 2019 IEEE International Conference on Smart Internet of Things (SmartIoT), Tianjin, China, 9–11 August 2019; pp. 212–219.
21. Jeong, J.; Kim, D.; Ihm, S.-Y.; Lee, Y.; Son, Y. Multilateral personal portfolio authentication system based on hyperledger fabric. *ACM Trans. Internet Technol. (TOIT)* **2021**, *21*, 1–17. [CrossRef]
22. Turing Certs. Available online: <https://certs.turingchain.tech/> (accessed on 30 May 2022).
23. Netizen Chain. Available online: <https://www.netizenbc.com/> (accessed on 25 May 2022).
24. Pearman, S.; Thomas, J.; Naeini, P.E.; Habib, H.; Bauer, L.; Christin, N.; Cranor, L.F.; Egelman, S.; Forget, A. Let’s go in for a closer look: Observing passwords in their natural habitat. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; pp. 295–310.
25. Lim, S.Y.; Fotsing, P.T.; Almasri, A.; Musa, O. Blockchain technology the identity management and authentication service disruptor: A survey. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2018**, *8*, 1735–1745. [CrossRef]
26. Naik, N.; Jenkins, P. Securing digital identities in the cloud by selecting an apposite Federated Identity Management from SAML, OAuth and OpenID Connect. In Proceedings of the 2017 11th International Conference on Research Challenges in Information Science (RCIS), Brighton, UK, 10–12 May 2017; pp. 163–174.
27. Kao, Y.W.; Huang, K.Y.; Gu, H.Z.; Yuan, S.M. Ucloud: A user-centric key management scheme for cloud data protection. *IET Inf. Secur.* **2013**, *7*, 144–154. [CrossRef]
28. Preukschat, A.; Reed, D. *Self-Sovereign Identity*; Manning Publications: Shelter Island, NY, USA, 2021.
29. Allen, C. The Path to Self-Sovereign Identity. Available online: <https://www.lifewithalacrity.com/article/the-path-to-self-sovereign-identity/> (accessed on 1 December 2023).
30. Soltani, R.; Nguyen, U.T.; An, A. A survey of self-sovereign identity ecosystem. *Secur. Commun. Netw.* **2021**, *2021*, 1–26. [CrossRef]
31. Bandara, E.; Liang, X.; Foytik, P.; Shetty, S.; De Zoysa, K. A blockchain and self-sovereign identity empowered digital identity platform. In Proceedings of the 2021 International Conference on Computer Communications and Networks (ICCCN), Athens, Greece, 19–22 July 2021; pp. 1–7.
32. Naik, N.; Jenkins, P. uPort open-source identity management system: An assessment of self-sovereign identity and user-centric data platform built on blockchain. In Proceedings of the 2020 IEEE International Symposium on Systems Engineering (ISSE), Vienna, Austria, 12 October–12 November 2020; pp. 1–7.
33. Sandhu, R.S. Role-based access control. In *Advances in Computers*; Elsevier: Amsterdam, The Netherlands, 1998; Volume 46, pp. 237–286.
34. Kiran, S.; Lareau, P.; Lloyd, S. PKI Basics—A Technical Perspective. PKI-Forum. 2002. Available online: https://people.cs.vt.edu/~kafura/cs6204/Readings/Context-Problems/PKI_Basics.pdf (accessed on 9 November 2024).

35. Paillisse, J.; Subira, J.; Lopez, A.; Rodriguez-Natal, A.; Ermagan, V.; Maino, F. Distributed access control with blockchain. In Proceedings of the ICC 2019–2019 IEEE International Conference on Communications (ICC), Shanghai, China, 21–23 May 2019; pp. 1–6.
36. Yang, C.; Tan, L.; Shi, N.; Xu, B.; Cao, Y.; Yu, K. AuthPrivacyChain: A blockchain-based access control framework with privacy protection in cloud. *IEEE Access* **2020**, *8*, 70604–70615. [[CrossRef](#)]
37. Fu, W.-K.; Lin, Y.-S.; Campagna, G.; Liu, C.-T. Soteria: A provably compliant user right manager using a novel two-layer blockchain technology. In Proceedings of the 2020 IEEE Infrastructure Conference, Pacific Grove, CA, USA, 7–8 October 2020; pp. 1–10.
38. Rouhani, S.; Belchior, R.; Cruz, R.S.; Deters, R. Distributed attribute-based access control system using permissioned blockchain. *World Wide Web* **2021**, *24*, 1617–1644. [[CrossRef](#)]
39. Guo, H.; Meamari, E.; Shen, C.-C. Multi-authority attribute-based access control with smart contract. In Proceedings of the 2019 International Conference on Blockchain Technology, Xi'an China, 9–11 December 2019; pp. 6–11.
40. Zhao, G.; He, H.; Di, B.; Chu, J. StuChain: An efficient blockchain-based student e-portfolio platform integrating hybrid access control approach. *Multimed. Tools Appl.* **2023**, *83*, 227–251. [[CrossRef](#)]
41. Zheng, Y. Design of a blockchain-based e-portfolio evaluation system to assess the education and teaching process. *Int. J. Emerg. Technol. Learn. (IJET)* **2021**, *16*, 261–280. [[CrossRef](#)]
42. Merlec, M.M.; Islam, M.M.; Lee, Y.K.; In, H.P. A consortium blockchain-based secure and trusted electronic portfolio management scheme. *Sensors* **2022**, *22*, 1271. [[CrossRef](#)]
43. Palanivel, K. Blockchain architecture to higher education systems. *Int. J. Latest Technol. Eng. Manag. Appl. Sci* **2019**, *8*, 124–138.
44. Li, H.; Han, D. EduRSS: A blockchain-based educational records secure storage and sharing scheme. *IEEE Access* **2019**, *7*, 179273–179289. [[CrossRef](#)]
45. Li, D.; Han, D.; Zheng, Z.; Weng, T.H.; Li, H.; Liu, H.; Castiglione, A.; Li, K.C. MOOCsChain: A blockchain-based secure storage and sharing scheme for MOOCs learning. *Comput. Stand. Interfaces* **2022**, *81*, 103597. [[CrossRef](#)]
46. Yan, J.-Y.; Hsieh, Y.-H.; Yuan, S.-M. Blockchain Based E-portfolio Ecosystem. In Proceedings of the 2023 IEEE 6th International Conference on Knowledge Innovation and Invention (ICKII), Hokkaido, Japan, 21–23 July 2023; pp. 264–269.
47. Voigt, P.; Von dem Bussche, A. The eu general data protection regulation (gdpr). In *A Practical Guide*, 1st ed.; Springer International Publishing: Cham, Switzerland, 2017; Volume 10, pp. 10–5555.
48. MetaMask docs. MetaMask. Available online: <https://docs.metamask.io/> (accessed on 15 December 2021).
49. Apache Jmeter. Available online: <https://jmeter.apache.org/> (accessed on 1 May 2022).
50. Adobe. Adobe Forecasts Record \$240.8 Billion U.S. Holiday Season Online with Black Friday Growth to Outpace Cyber Monday. Available online: <https://news.adobe.com/news/2024/09/092524-adi-holiday-forecast> (accessed on 1 November 2024).
51. Fablo. Available online: <https://github.com/hyperledger-labs/fablo> (accessed on 1 May 2022).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.