

Self-Sovereign Identity Using WalletConnect for Issuing, Verifying, and Revoking Academic Credentials

Ardi Birawinata¹⁾, Siti Sauda²⁾

^{1,2)}Universitas Bina Darma, Indonesia

Jalan Jenderal Ahmad Yani No. 3, Palembang, South Sumatra 30111, Indonesia

¹⁾natagskt97@gmail.com²⁾, siti_sauda@binadarma.ac.id

Submitted : Aug 21, 2026 | **Accepted :** Sep 11, 2026 | **Published :** Oct 1, 2026

Abstract: Student data at vocational high schools is managed solely by the schools, so external parties must contact the school to confirm a record, while account access still relies on passwords vulnerable to credential theft. This study implements a Self-Sovereign Identity system for student academic data at SMK Bina Sriwijaya Indonesia Palembang that combines passwordless WalletConnect authentication with credential issuance, verification, and revocation anchored on a public blockchain. Following the Design Science Research Methodology, the artefact was built as a web prototype using React.js, Express.js, and PostgreSQL, connected to MetaMask and to a CredentialRegistry smart contract on Ethereum Sepolia. Users authenticate through an EIP-191 challenge-response signature, student identity is expressed as a did:ethr Decentralized Identifier, and academic data is issued as W3C Verifiable Credentials whose canonicalised Keccak-256 hash and status are recorded on chain, while the complete record remains off chain. Evaluation covered four levels: eighteen black-box functional cases, thirty smart contract unit tests, seventeen backend tests, and fourteen security cases at the authentication and verification boundary, plus gas measurement of every state-changing function on Sepolia. All seventy-nine cases produced the expected output, which demonstrates functional conformance and correct access control rather than a general guarantee of security, since independent penetration testing and a formal contract audit were not performed. Within these limits, a verifier can check authenticity, issuer, and validity independently, without confirmation from the school and without exposing personal data on a public blockchain.

Keywords: Blockchain; Decentralized Identifier; Passwordless Authentication; Self-Sovereign Identity; Verifiable Credentials; WalletConnect

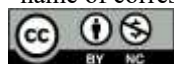
INTRODUCTION

At SMK Bina Sriwijaya Indonesia Palembang, student identity data such as the national student identification number, study programme, and class are the primary information used for administration and for proving academic status. These data are managed centrally by the school, so students have no digital means of accessing and sharing their own academic identity.

A second problem appears on the verification side. When an internship provider, a university, a competition organiser, or a government agency needs proof of a student record, the check is still performed through printed documents and direct confirmation to the school, so it depends on the availability of school staff and cannot be carried out independently by the verifier. Access to academic services also still relies on usernames and passwords, which are vulnerable to credential theft and phishing. Periodic vulnerability assessment of academic web applications remains necessary to find the gaps that attackers may exploit (Dasmen, Rasmila, Widodo, Kundari, & Farizky, 2023).

Digital identity supports three main processes, namely identification, authentication, and authorisation (Robles-Carrillo, 2024). Self-Sovereign Identity (SSI) is an identity management paradigm that places the individual as the holder of full control over their own identity without depending on a central authority (Grech, Sood, & Ariño, 2021). In an SSI ecosystem the issuer publishes the credential, the holder keeps and presents it,

*name of corresponding author



This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

and the verifier checks it, a structure that matches the relationship between a school, its students and external parties.

Blockchain has been applied to academic documents, and, in a separate line of work, wallet-based decentralised authentication has been demonstrated as a replacement for conventional login. These two contributions have so far remained apart. Studies on academic certificates anchored on a blockchain generally keep the institution as the sole custodian of the document and do not give the student a cryptographic identity of their own, while studies on decentralised authentication stop at the login step and do not address credential issuance, verification, or revocation. The novelty of this study therefore does not lie in Self-Sovereign Identity or in blockchain as technologies, since both are established, but in their integration into a single operational flow for the vocational school context. In this flow, the wallet address used for passwordless authentication through WalletConnect is the same address that deterministically becomes a did:ethr identifier. That identifier becomes the credential subject of a Verifiable Credential document issued by the school, and only the canonicalised hash and the status of that document are anchored on a public blockchain, so that an external party can verify authenticity and detect revocation without contacting the school and without any student personal data leaving the school database. To the best of the authors' knowledge, this integration has not been reported for the Indonesian vocational high school setting, where the observation conducted in this study found a high volume of verification requests alongside limited administrative capacity to answer them.

This study therefore aims to implement a prototype of a student academic identity system based on Self-Sovereign Identity on the Ethereum Sepolia network, covering passwordless challenge-response authentication with WalletConnect and MetaMask, the deterministic generation of a did:ethr identifier, the issuance of academic credentials following the W3C Verifiable Credentials Data Model 2.0 (Sporny, Longley, Chadwick, & Herman, 2025), independent public verification, and credential revocation. The contributions are threefold. First, an end-to-end SSI artefact for vocational school academic data in which the school acts as issuer, the student as holder, and any external party as verifier. Second, a design that binds passwordless wallet authentication and the credential subject identifier to a single wallet address, so that authentication and identity are not two separate registrations. Third, a hash-only anchoring scheme with a public QR code that carries no personal data, allowing verification and revocation checking on a public blockchain while keeping the academic record off chain.

LITERATURE REVIEW

An electronic identity may contain only the attributes needed to perform a particular task, so it does not have to be a complete representation of the physical identity of its owner (Schardong & Custódio, 2022). This property is the basis for disclosing only the data a specific purpose requires.

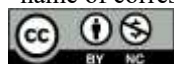
Čučko & Turkanović (2021) explain that the core of SSI lies in minimising dependence on a central authority, which reduces the risk of a single point of failure. Technically, this sovereignty is achieved through Decentralized Identifiers (DID) that follow the World Wide Web Consortium specification (Sporny et al., 2022) and Verifiable Credentials (VC) defined in the data model of the same organisation (Sporny et al., 2025). A DID is a globally unique identifier that can be verified through cryptographic proof and does not depend on a centralised registry (Butincu & Alexandrescu, 2024). The general structure of a DID consists of the did prefix, the method name, and the method-specific identifier (Hoops, Muhle, Matthes, & Meinel, 2023).

A VC is a structured digital credential that carries one or more claims about a subject and can be verified using cryptographic mechanisms, so that any modification of the credential content is detectable and the identity of the issuer can be examined (Mazzocca et al., 2025). A VC may also support selective disclosure, which allows a holder to present only the attributes required by the verifier (Flamini et al., 2024). In education, Chotikakamthorn, Mi San, & Sathitwiriawong (2024) embedded a compact credential fingerprint in a QR code printed on a paper credential, which shows that a QR code can connect a physical document to its digital verification process.

Blockchain security relies on cryptography: a hash function maps data of any size to a fixed-size output, so any change produces a different value, while a digital signature proves the authenticity and integrity of a message (Sarwar, Maghrabi, Khan, Naith, & Nisar, 2023). Several studies have applied these properties to academic documents. Fartitchou, Lamaakal, Makkaoui, Allali, & Maleh (2025) developed a smart contract system for securing higher education digital certificates, and Silaghi, Artenie, & Popescu (2025) showed that Ethereum and a cost-effective smart contract can support the integrity and traceability of teacher portfolio documents issued by schools. Sultana, Rupa, Malleswari, & Gadekallu (2023) proposed a hybrid on-chain and off-chain framework based on IPFS to reduce certificate fraud, although the framework does not adopt SSI principles. Chan, Gai, Yu, & Zhu (2025) surveyed blockchain-assisted SSI in the education sector and reported that interoperability, scalability, security, and regulatory compliance still require further development before comprehensive deployment in the academic sector.

On the authentication side, Petcu, Pahontu, Frunzete, & Stoichescu (2023) demonstrated that decentralised authentication based on Web 3.0 and the Ethereum blockchain can replace conventional login with a more secure digital signature, although the work was not applied to academic data and did not address credential issuance.

*name of corresponding author



This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

WalletConnect is a protocol that connects a decentralised application to a user wallet through a QR code or a connection link, so that the user can interact with the application without entering credentials directly (Román-García, Hernández-Serrano, & Esparza, 2024). The same study notes that WalletConnect depends on external relay infrastructure and proposes an alternative pairing protocol. The protocol was nevertheless selected here because it works with the wallet applications that students already run on their own devices. The wallet itself stores the private key in dedicated storage for sensitive cryptographic material, which keeps the key confidential and available (Kersic, Vidovic, Vrecko, Domajnko, & Turkanovic, 2023).

Table 1. Position of the present study against related work

Study	Wallet auth	DID	VC	On-chain anchor	Off-chain data	Revocation	Public verification
Petcu et al. (2023)	Yes	No	No	No	n.a.	No	No
Sultana et al. (2023)	No	No	No	Yes	Partly	Yes	Yes
Fartitchou et al. (2025)	No	No	No	Yes	Yes	Yes	Yes
Silaghi et al. (2025)	Yes	No	No	Yes	Yes	Not reported	Yes
Chotikakamthorn et al. (2024)	No	Yes	Yes	Yes	Partly	No (future work)	Yes
This study	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Yes = reported and implemented; Partly = partially addressed; No = not provided by the design; future work = named by the authors as planned but not implemented; Not reported = not stated in the source; n.a. = not applicable.

Table 1 shows that each reviewed implementation covers only part of the credential life cycle. Several anchor certificate integrity on chain, keep the document off chain and expose a public check, two provide a revocation function, and two authenticate through a wallet. None, however, combines wallet authentication with a decentralised identifier: the works that authenticate through a wallet issue neither a decentralised identifier nor a Verifiable Credential, while the work that issues Verifiable Credentials to decentralised identifiers authenticates by other means. The address a user signs in with is therefore never the address that becomes the credential subject, so authentication and credential identity remain two separate registrations. This is consistent with the finding of Chan et al. (2025) that interoperability, scalability, security, and regulatory compliance still require further development before blockchain-assisted Self-Sovereign Identity can be deployed comprehensively in education. The gap addressed by the present study is therefore the absence of a single artefact in which wallet authentication, deterministic did:ethr generation from that same wallet address, school-issued Verifiable Credentials, public verification without the issuer, and verifiable revocation form one continuous flow, demonstrated in a vocational high school setting.

METHOD

This study uses the Design Science Research Methodology (DSRM) formulated by Peffers, Tuunanen, Rothenberger, & Chatterjee (2007) because the research is oriented towards the design, development, demonstration, and evaluation of an artefact that solves a real problem. The organisation of each stage follows the echeloned Design Science Research approach of Tuunanen, Winter, & vom Brocke (2024), so that the output of one stage becomes the input of the next. The first stage, problem identification and motivation, analyses the academic data management and verification process at SMK Bina Sriwijaya Indonesia Palembang and reviews the literature on SSI, blockchain, DID, and Verifiable Credentials. The second stage defines the objectives of the solution, namely the system requirements, actor roles, and the authentication, issuance, and verification flows. The third stage designs and develops the artefact, covering the system architecture, use case model, database, smart contract and interface design, and then builds the prototype. The fourth stage demonstrates the artefact by running the complete flow against the Ethereum Sepolia network. The fifth stage evaluates the artefact on four complementary levels, described below, because functional testing alone is not sufficient for a system that handles identity data. The sixth stage communicates the results as a research report and a scientific article.

Data were collected through observation of the academic data management process, interviews with teachers and staff, and a literature study of journals and technical standards (Ardiansyah, Risnita, & M.Syahrani Jailani, 2023).

The identification stage produced six problems, which correspond to the verification, authentication, document integrity, credential status, privacy and actor-relationship issues set out in the introduction.

The system architecture separates data stored off chain from integrity proofs recorded on chain. Student data, user sessions, credential documents, public verification codes, and activity history are stored in PostgreSQL. The

*name of corresponding author



This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

blockchain stores neither the national student identification number nor the name, study programme, class, or any other personal data; it only stores the credential identity hash, the credential content hash, the issuer address, the issuance time, and the active status. The system consists of three layers as shown in Fig. 1.

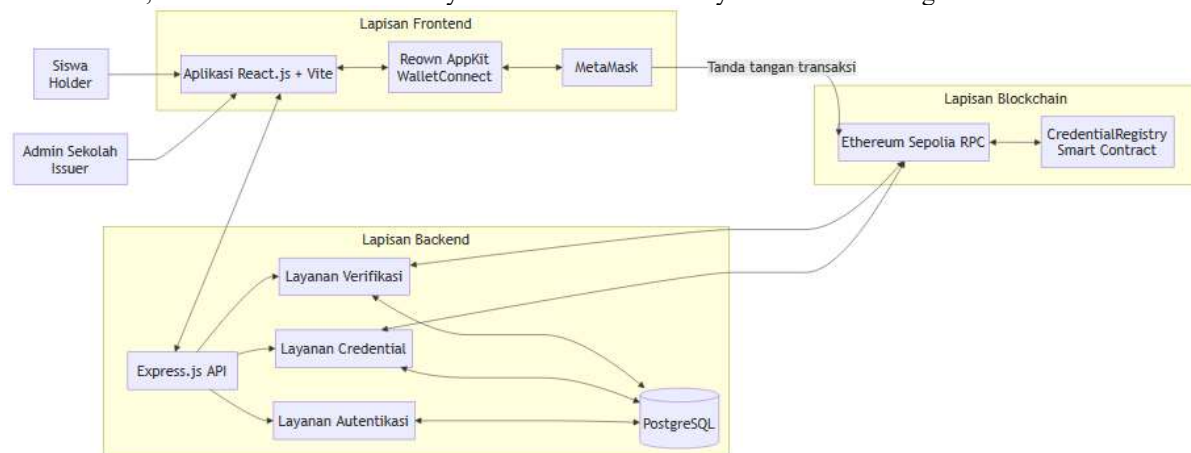


Fig. 1 System architecture of the SSI-based student academic identity

The frontend layer is developed with React.js and Vite and is connected to MetaMask through WalletConnect integrated using Reown AppKit. This layer provides three main spaces, namely the administrator dashboard, the student portal, and the public verification page. The backend layer is developed with Node.js, Express.js, and TypeScript, with database access through Drizzle ORM. The backend generates the authentication challenge, verifies the wallet signature, forms the DID, composes the Verifiable Credential document, canonicalises the JSON, computes the hash, and validates the transaction after block confirmation. The use of Node.js and PostgreSQL as a backend that provides a REST API for the frontend has been shown to be applicable in web application development (Barokah & Sauda, 2022). The blockchain layer consists of the Ethereum Sepolia network and the CredentialRegistry smart contract.

Authentication is designed as a passwordless challenge-response mechanism. The wallet acts as the instrument that proves identity ownership, while the backend verifies the signature without ever obtaining the private key. The process starts when the user connects MetaMask through WalletConnect. The frontend obtains the wallet address, confirms that the wallet uses the Ethereum Sepolia network with chain ID 11155111, and sends the address to the backend to request a challenge. The backend generates a random nonce, the issuance time, the expiry time, the application domain, and the network information, and stores them in the `auth_challenges` table with a limited lifetime. The user signs the message with MetaMask according to the EIP-191 format. The backend then recovers the wallet address from the signature and compares it with the address recorded in the challenge, and also checks the message, domain, network, expiry time, and the usage status of the challenge. When all checks succeed and the wallet is registered, the backend marks the challenge as used and creates a user session. The unique nonce, the limited lifetime, and the single-use status are intended to prevent replay attacks. The authentication flow is shown in Fig. 2.

The administrator records student data consisting of the national student identification number, full name, study programme, and class, and then links the MetaMask address of the student. One wallet address can be linked to only one active student account. After the wallet address is stored, the system forms the DID deterministically using the pattern `did:ethr:sepolia:<student-wallet-address>`, which follows the DID syntax defined by the World Wide Web Consortium (Sporny et al., 2022) and the `did:ethr` method described by Hoops et al. (2023). The DID is stored as the subject identifier of the Verifiable Credential, so the same address always produces the same DID even when the student accesses the system from a different device.

Credential issuance uses a two-phase transaction pattern. In the first phase, the backend composes a document that follows the W3C Verifiable Credentials Data Model 2.0 (Sporny et al., 2025), containing the credential identity, credential type, issuer DID, validity date, student DID, academic data, credential status, and schema reference. The document is canonicalised using the JSON Canonicalization Scheme, converted to UTF-8 bytes, and hashed with Keccak-256, producing `credentialIdHash` as the mapping key in the contract and `contentHash` as the integrity proof of the credential content. In the second phase, the issuer wallet signs the `registerCredential` transaction and sends it to the Ethereum Sepolia network. After two block confirmations, the backend inspects the transaction receipt, the target contract address, the calling function, the `CredentialRegistered` event, and the consistency of `credentialIdHash` and `contentHash`. The credential status changes from PENDING to ACTIVE only when all of these checks succeed. The issuance flow is shown in Fig. 3.

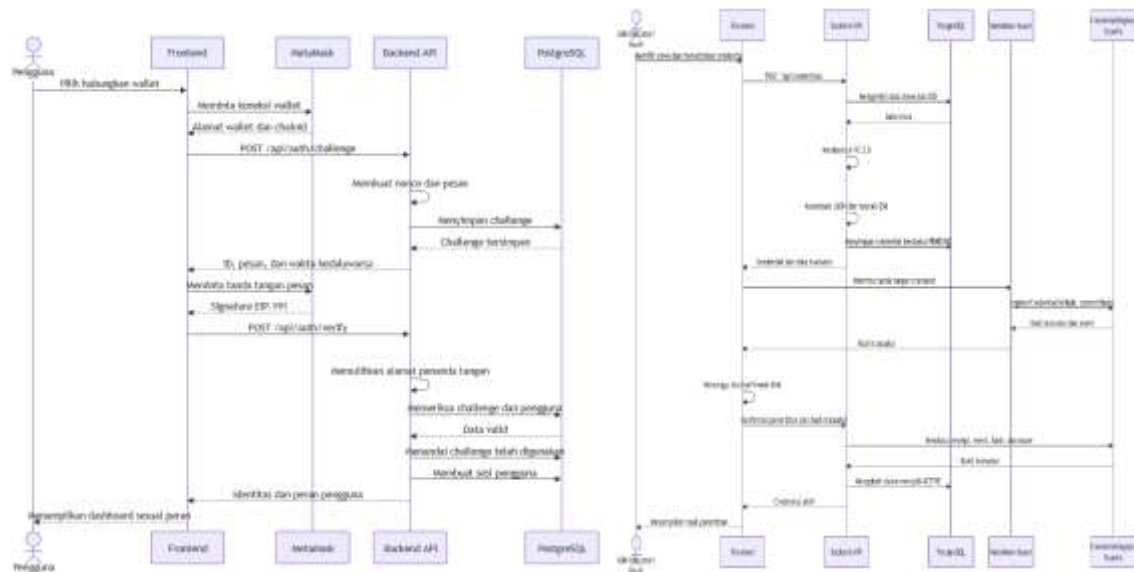


Fig. 2 Wallet-based challenge-response authentication flow (left) and Fig. 3 Verifiable Credential issuance flow (right)

Public verification is provided through a QR code that stores no personal data and carries only a URL containing a random public code. When the QR code is scanned, the backend retrieves the credential document from PostgreSQL, removes the transaction proof from the hash computation, canonicalises the document again, and recomputes contentHash. The backend then reads the credential record from the smart contract using credentialIdHash. The verification result is determined as valid when the hash matches, the issuer matches, the record exists, and the status is active both in the database and on the blockchain; as inactive when the hash and the issuer match but the status has been revoked; and as invalid when the hash differs, the issuer does not match, the record is absent from the contract, or the status is inconsistent.

Revocation can be performed only by an authenticated administrator holding an authorised issuer wallet. The backend prepares the call to the deactivateCredential function, and the issuer confirms the transaction through MetaMask. After the transaction obtains two block confirmations and the CredentialDeactivated event is read, the credential status in the database becomes INACTIVE. Revocation deletes neither the document nor the issuance history, so traceability is preserved.

The CredentialRegistry smart contract is developed in Solidity version 0.8.19. The contract holds an owner variable, a list of issuer addresses, and a credential mapping. Each mapping entry stores contentHash, the issuer address, the issuance time, the active status, and an existence flag. The mapping key is credentialIdHash, so neither the credential identity nor personal data is written directly on the blockchain. The functions provided by the contract are shown in Table 2.

Table 2. Functions of the CredentialRegistry smart contract

Function	Access right	Description
setIssuer	Owner	Grants or revokes the issuer right
transferOwnership	Owner	Transfers ownership of the contract
registerCredential	Issuer	Records credentialIdHash and contentHash
deactivateCredential	Issuing issuer	Changes the credential status to inactive
getCredential	Public	Reads credential data by identity hash
verifyCredential	Public	Checks contentHash consistency and active status

The contract emits the CredentialRegistered event after issuance and the CredentialDeactivated event after revocation. The backend uses these events to confirm that a transaction really called the intended contract, function, and data. A registered credential identity cannot be overwritten, and a revoked credential cannot be reactivated in this prototype version.

Evaluation design

The first level is black-box functional testing of eighteen test cases at the application level, which examines whether each function of the artefact produces the output defined in the design specification for the three actors involved.

*name of corresponding author



The second level is unit testing of the CredentialRegistry smart contract with the Hardhat framework, comprising thirty test cases across the six contract functions. This level is directed at the trust boundary of the contract rather than at its normal operation.

The third level is unit and integration testing of the backend with the Vitest framework, comprising seventeen test cases on canonicalisation, hashing and identifier formation.

The fourth level is a set of fourteen targeted security test cases at the authentication and verification boundary of the application. Twelve cases attempt to defeat a specific control, and two are control cases that confirm the same endpoint behaves correctly for a legitimate request, so that a rejection can be attributed to the security mechanism rather than to a general failure of the system. The twelve attack cases were derived from the threat surface of the two boundaries that the artefact exposes to an untrusted party. At the authentication boundary these are challenge replay, use of an expired challenge, substitution of the signing wallet, tampering with the signed message, a request for a chain other than the configured one, and the invocation of issuer functions from a holder session or without any session. At the verification boundary these are the submission of a credential document whose content has been altered, the substitution of the registry contract named in the proof, a verification code that was never issued, and a login from a wallet that is not registered. Gas consumption of every state-changing function of the contract was measured separately on the Sepolia network using a test instance of the contract. These four levels are deliberately not presented as a complete security evaluation, because independent penetration testing of the REST API and a formal audit of the smart contract are outside the scope of this study.

RESULT

Implementation and testing were carried out in a local environment connected to the Ethereum Sepolia network, with PostgreSQL storing student data, user accounts, authentication challenges, credentials, sessions and activity history, and the ethers.js library handling interaction with the network. The smart contract deployment configuration is shown in Table 3.

Table 3. Smart contract deployment configuration

Attribute	Value
Network	Ethereum Sepolia
Chain ID	11155111
Smart contract address	0x98D93456A9cb32B831E3149f3BdBdF26B7978313
Issuer address	0x64a1153BCa7065707E3AB6833428dC9eC9EF3Fd3
Deployment block	11445808
Deployment transaction	0x812d999ed2609a60e3bb58b360f0d5b26e754be6082eb72aa757bb950dda1b42
Transaction confirmation	2 blocks

Authentication for both the issuer and the student was implemented with the challenge-response mechanism designed in the method section. The implementation shows that the issuer wallet ending in 3Fd3 obtained the ISSUER_ADMIN role, while the student wallet ending in 1Eb5 obtained the HOLDER role. Sessions are created without a username and a password, so no static credential data is stored on the server.

The student data page is used by the issuer to add, update, search, activate, or deactivate student records. After the wallet address is stored, the backend forms the student DID using the pattern `did:ethr:sepolia:<student-wallet-address>`. In the demonstration flow, the student wallet ending in 1Eb5 produced the DID `did:ethr:sepolia:0x62eabcD7ca2A847222B6dA748E24e13b11061Eb5`, and the interface displayed a DID ready indicator, showing that the student record met the issuance requirements. The system restricts changes to academic data and to the wallet address while a credential is PENDING or ACTIVE, preventing inconsistency with the hash already recorded on the blockchain.

Credential issuance produced a document that adopts the structure of the W3C Verifiable Credentials Data Model 2.0 (Sporny et al., 2025). The document was canonicalised with the JSON Canonicalization Scheme and hashed with Keccak-256 to obtain `credentialIdHash` and `contentHash`. The credential was stored with the initial status PENDING and changed to ACTIVE after the `registerCredential` transaction obtained two block confirmations and all receipt and event checks succeeded. The transaction information was then added to the document as registry proof containing the transaction hash, the smart contract address, the chain ID, the block number, `credentialIdHash`, and `contentHash`. Testing recorded three confirmed issuance transactions on the Sepolia network.

A student who authenticates successfully is directed to the student portal. The portal displays the DID formed from the wallet address together with every credential linked to that account, so a student cannot access credentials belonging to another wallet. The portal provides buttons to copy the verification link, download the QR code,

download the credential document in W3C VC JSON format, build a Verifiable Presentation, and open the issuance transaction on Etherscan Sepolia. The student portal is shown in Fig. 4.

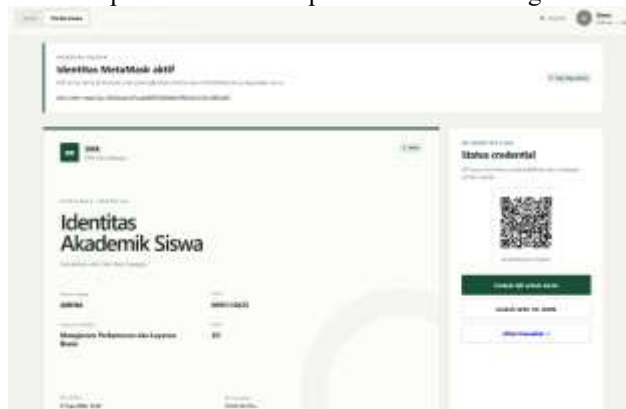


Fig. 4 Student portal and credential QR code

The QR code is generated from a public verification URL that uses a random code. The QR code carries neither the name nor the identification number, the DID, or any academic data, so personal data is placed neither in the QR code nor on the blockchain.

Public verification is available on two paths, neither of which requires authentication or a connected wallet. The first is the verification page hosted by the school, which accepts a QR code or a public verification code and retrieves the corresponding credential document from PostgreSQL before recomputing its hash and reading the registry record from the smart contract. The second is an independent verifier distributed as a static page that runs entirely in the browser of the verifying party and holds no connection to the school API or database. It loads a trusted configuration containing the network identifier, the registry contract address and the issuer DID, and never takes those values from the credential under examination. Given a credential file downloaded by the student, it removes the proof block, canonicalises the document with the JSON Canonicalization Scheme, recomputes the Keccak-256 content hash, and reads the registry record directly from the smart contract over a public Sepolia endpoint, confirming that the issuance transaction reached the official contract with at least two confirmations at the block number named in the proof, and that its `CredentialRegistered` event holds the same hashes and the trusted issuer address. Both paths report the same six checks. Fig. 5 shows the independent verifier returning the status `Credential valid` with all six checks passed for a credential downloaded from the student portal.

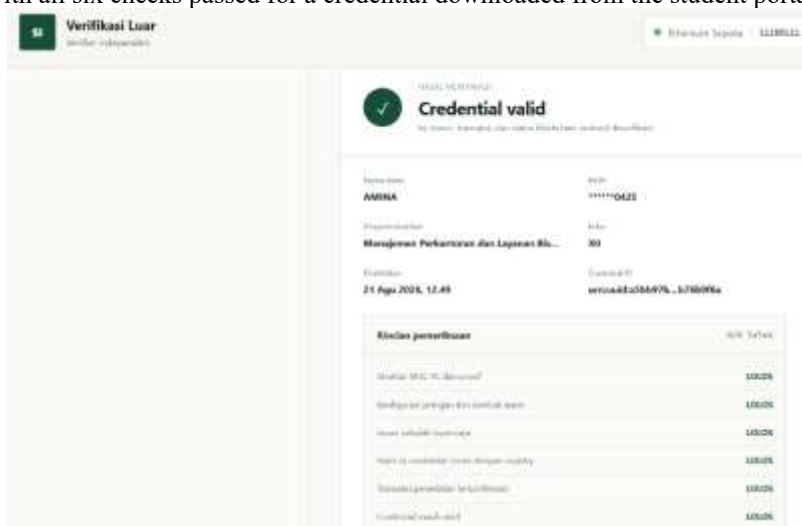


Fig. 5 Verification of a downloaded credential by an independent verifier that runs without the school backend

The student information displayed on the verification page is limited to the data required for the check, and the identification number is shown in a masked form.

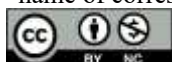
Black-box testing covered eighteen test cases involving wallet authentication, access restriction, student data management, DID generation, credential issuance, the student portal, the QR code, public verification, and credential revocation. The testing involved three actors, namely the school administrator as issuer, the student as holder, and the external party as verifier. A case was recorded as successful when the actual output matched the expected output. The test results are summarised in Table 4. The revocation transaction referred to in cases 15 and

16 is 0x41c253c75ddd0ce8fa66837c1ba84ac6ed9d1282ed29dbc86ed43e3e5c71b71, confirmed in block 11674968 on the Ethereum Sepolia network.

Table 4. Black-box testing results

No	Test case	Expected result	Actual result	Status
1	Issuer authentication with WalletConnect and MetaMask	ISSUER_ADMIN session created after signature validation	Wallet 3Fd3 recognised; issuer dashboard opened without a password	Passed
2	Student authentication with a registered wallet	HOLDER session created and student portal displayed	Wallet 1Eb5 recognised; holder DID shown and portal opened	Passed
3	Access restriction by role	Holder cannot reach issuer functions and the reverse	Access rejected; registered role of the wallet displayed	Passed
4	Identification number in an invalid format	Data rejected and no student record stored	Value 123 rejected; record not stored	Passed
5	DID generation from the student wallet	DID formed from the Sepolia network and the wallet address	Two students with a wallet shown with DID ready status	Passed
6	Editing a student holding a PENDING or ACTIVE credential	Change restricted to keep data consistent with the credential	Editable fields disabled; restriction notice displayed	Passed
7	Opening the credential issuance form	Only students meeting the issuance requirements listed	Form listed active students with a wallet and a DID	Passed
8	Checking the issuance transaction on Sepolia	Confirmed issuance transactions displayed	Issuance transactions shown with status Confirmed	Passed
9	Accessing credentials through the student portal	Only credentials linked to the signed-in wallet visible	Portal displayed only credentials of the signed-in account	Passed
10	Displaying the credential QR code	QR code pointing to the public verification service	QR code displayed in the instant verification panel	Passed
11	Downloading the W3C VC JSON credential	Credential document generated and downloaded in JSON	System reported the W3C VC credential downloaded	Passed
12	Verifying an ACTIVE credential with the public code	Hash, issuer, record and on-chain status checked; result valid	Credential valid returned with the masked identification number	Passed
13	Verifying an unregistered credential code	Code rejected and credential reported as not found	Verification failed; Credential not found	Passed
14	Uploading a JSON credential whose content was modified	Change detected and credential rejected	Result Credential invalid returned	Passed
15	Revoking an ACTIVE credential with the issuer wallet	Wallet approval requested, deactivateCredential sent, status set to INACTIVE	Status changed from Active to Inactive after the transaction was confirmed in block 11674968	Passed
16	Checking the revocation transaction	Revocation transaction recorded after backend confirmation	History showed the transaction Confirmed, matching the CredentialDeactivated event in block 11674968	Passed
17	Re-verifying a revoked credential	Credential reported as inactive	The same public code returned Credential inactive	Passed

*name of corresponding author



This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

No	Test case	Expected result	Actual result	Status
18	Attempting to revoke an INACTIVE credential again	Repeated revocation prevented	Revoke action unavailable for an inactive credential	Passed

Unit testing of the CredentialRegistry contract and of the backend was executed on an isolated environment that replicates the artefact stack, using Hardhat with solc 0.8.19 for the contract and Vitest for the backend. The thirty contract cases are distributed across the six contract functions and are directed at the trust boundary rather than at normal operation. They cover calls from a non-owner and from an address without issuer rights, zero addresses and zero identity hashes, duplicate credential identities, repeated deactivation and deactivation attempted by an issuer other than the issuing one, and the response of verifyCredential to an altered content hash, to a revoked credential and to an identity that was never registered. Two cases confirm that credentials already issued remain readable and verifiable after the issuer authorisation is revoked and after contract ownership is transferred. The seventeen backend cases cover canonicalisation and hash stability, the detection of a modified payload, the exclusion of the proof block, the masking of the identification number, the published JSON-LD context, credential schema and issuer DID Document, and the deterministic formation of did:ethr identifiers. All forty-seven cases produced the expected output, as summarised in Table 5.

Table 5. Unit and integration testing results

Level	Target and focus of the test group	Cases	Passed
Smart contract	setIssuer: issuer rights granted and revoked, non-owner and zero address rejected	8	8
Smart contract	transferOwnership: ownership transfer, non-owner and zero address rejected	5	5
Smart contract	registerCredential: authorised issuance, unauthorised wallet, duplicate identity, zero hash	4	4
Smart contract	deactivateCredential: issuing issuer, repeated revocation, unregistered credential, another issuer	5	5
Smart contract	getCredential: record retrieval, empty record, public read access	3	3
Smart contract	verifyCredential: active, altered content, revoked, unregistered, persistence after issuer and owner change	5	5
Backend	Service endpoints and the published JSON-LD context, schema and DID Document	4	4
Backend	Credential canonicalisation, hashing and identification number masking	5	5
Backend	Verifiable Credential structure, credential offer and verification request	3	3
Backend	Credential status encoding and key representation	3	3
Backend	Wallet normalisation and did:ethr identifier formation	2	2
Total		47	47

The results at this level show that the access control rules of the contract hold at its trust boundary and that the hash computation of the backend is deterministic and sensitive to any change of credential content. These are the two properties on which independent public verification depends.

Security testing at the authentication and verification boundary produced the results summarised in Table 6. All fourteen cases produced the expected output. Every attempt to reuse a consumed challenge, to sign after the expiry time, to substitute a different signing wallet, to alter the signed message, to request a challenge for another network, to reach an issuer function from a holder session or without a session, to submit an altered credential document, or to point the registry proof at another contract was rejected by the control intended to reject it, while the two control cases confirmed that legitimate requests on the same endpoints succeeded.

Table 6. Security testing results

No	Security test case	Actual result	Status
S1	Signature from a challenge that has already been used is resent	First login returned 200; the replay returned 409 CHALLENGE_ALREADY_USED	Passed
S2	Challenge signed after its expiry time	Rejected with 410 CHALLENGE_EXPIRED	Passed
S3	Valid signature produced by a wallet other than the one named in the challenge	Rejected with 401 SIGNER_MISMATCH	Passed
S4	Challenge message altered before it is signed	Rejected with 400 MESSAGE_MISMATCH	Passed
S5	Challenge requested for a chain other than Sepolia	Rejected with 400 WRONG_NETWORK	Passed
S6	Holder session calling an issuer-only endpoint	Access denied with 403 FORBIDDEN	Passed
S7	Issuer-only endpoint called without any session	Access denied with 401 UNAUTHENTICATED	Passed
S8	Revocation requested with a holder session	Access denied with 403 FORBIDDEN; no transaction prepared	Passed
S9	Credential document whose academic field has been altered	Verifier returned status INVALID with hashMatched false	Passed
S10	Registry proof pointing to a contract other than the official one	Verifier returned status INVALID	Passed
S11	Verification with a well-formed public code that was never issued	Returned 404 CREDENTIAL_NOT_FOUND	Passed
S12	Login with a valid signature from a wallet that is not registered	Rejected with 403 USER_NOT_REGISTERED	Passed
C1	Control: issuer session on the endpoint used in S6	Access granted with 200 and the role ISSUER_ADMIN	Passed
C2	Control: unmodified credential document submitted to the public verifier	hashMatched, issuerAuthorized and transactionVerified all true; status INACTIVE because the credential had been revoked on chain	Passed

Gas consumption was measured directly on the Sepolia network using a separate test instance of the contract, so that the production contract remained untouched. Contract deployment consumed 637,124 gas, registerCredential 71,692 gas, and deactivateCredential 31,419 gas. Five repetitions of transferOwnership consumed an identical 28,578 gas each, while setIssuer ranged from 25,891 to 47,803 gas depending on whether the storage slot was empty, cleared, or rewritten with the same value. Thirteen transactions consumed 1,080,328 gas in total, equivalent to 0.001146 ETH at the gas prices in effect during the measurement. Credential issuance, the recurring operation of the system, is therefore the dominant cost driver, while the administrative functions are executed only occasionally.

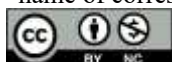
DISCUSSIONS

The test results show that every main function of the artefact produced the expected output. Three findings can be drawn from these results. First, authentication based on a digital signature can replace the username and password on an academic portal. The private key never leaves the user wallet, and the server stores only a single-use challenge with a limited lifetime. This finding extends the work of Petcu et al. (2023), which showed that decentralised authentication based on Ethereum can serve as an alternative to conventional login, into the context of vocational school academic data that the earlier study did not address.

Second, the separation between off-chain data and on-chain integrity proofs allows authenticity to be checked without exposing student personal data to the public. The blockchain stores only the credential identity hash, the credential content hash, the issuer address, the issuance time, and the active status. This approach is consistent with the hybrid storage idea proposed by Sultana et al. (2023), but the present study binds the credential to an identifier that the student controls through a private key the school never holds.

The degree to which the artefact satisfies Self-Sovereign Identity requires a precise statement, because the credential document is also stored in the school database. Control on the side of the student rests on three

*name of corresponding author



This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

properties. First, the credential subject is a did:ethr identifier derived from a wallet address whose private key resides only in the student wallet, so the school can neither authenticate as the student nor reassign the credential to another subject. Second, the portal allows the student to download the complete Verifiable Credential document in W3C JSON format, and that document can be checked by a verifier that has no connection to the school at all, as shown in Fig. 5. Third, presentation is initiated by the holder rather than by the school: the independent verifier issues a challenge and the student signs the resulting Verifiable Presentation in a wallet, so disclosure requires an explicit act of consent.

Two properties nevertheless remain outside full holder control. The primary copy of the credential document is retained by the issuer, so a student who has not downloaded the document depends on the school for its availability, and revocation is an issuer right, as the Verifiable Credentials data model prescribes. The artefact is therefore best described as an issuer-hosted Self-Sovereign Identity deployment in which the holder controls the identifier, the act of presentation, and a portable copy of the credential that remains verifiable without the school, rather than a full wallet-custody model. Moving credential storage into a student-side wallet agent is left to further work.

Third, this study responds to the need for further development before comprehensive deployment in the academic sector reported by Chan et al. (2025). The artefact combines passwordless authentication, DID generation, Verifiable Credential issuance, hash recording through a smart contract, public verification through a QR code, and a revocation mechanism into a single flow that has been demonstrated on the Ethereum Sepolia network. Limiting the verification page to the data required for the check follows the spirit of data minimisation, a principle carried further by the cryptographic selective disclosure mechanisms reviewed by Flamini et al. (2024).

This study has several limitations. The implementation covers one case study and runs on the Sepolia test network, so it has not been evaluated on a main network or with a larger number of users. The demonstration used a small number of student records and credentials, so performance and scalability were not measured quantitatively. The revocation flow was demonstrated end to end on a dedicated test credential whose issuance in block 11674965 and deactivation in block 11674968 are both recorded on the Sepolia network, while the credentials used in the issuance and verification demonstration were left untouched, so the revocation evidence rests on a single credential rather than on the whole sample. A further limitation concerns state synchronisation: the backend updates the credential status only after reading the confirmation event, so an interruption between a confirmed transaction and that update can leave the database behind the on-chain status. The prototype has no automatic reconciliation routine, and one such case was realigned manually during testing. In addition, the evaluation has not included user acceptance measurement, and the security testing reported here examines the controls the artefact implements rather than an adversarial assessment, so independent penetration testing of the API and a formal smart contract audit remain outside its scope.

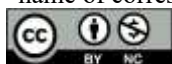
CONCLUSION

This study produced a prototype of a student academic identity system based on Self-Sovereign Identity that involves the school as issuer, the student as holder, and an external party as verifier. Passwordless challenge-response authentication using WalletConnect and MetaMask was implemented successfully, so a user obtains a session according to their role without entering a username and a password. Student identity is represented as a Decentralized Identifier with the did:ethr method on the Ethereum Sepolia network, and academic data is issued as a W3C Verifiable Credential 2.0 document whose hash and status are recorded through the CredentialRegistry smart contract. The complete academic data remains in PostgreSQL, so credential integrity and status can be checked without placing student personal data on a public blockchain. The student holds a downloadable copy of the credential that remains verifiable against the smart contract independently of the school database. Evaluation at four levels produced the expected output in all seventy-nine cases: eighteen black-box functional cases at the application level, thirty unit tests at the trust boundary of the smart contract, seventeen backend unit and integration tests on canonicalisation and hashing, and fourteen targeted security cases at the authentication and verification boundary. These include the rejection of unregistered verification codes, of modified credential documents, of replayed and expired authentication challenges, and of issuer functions called from a holder session.

These results should be read as the outcome of a prototype evaluation. The artefact was demonstrated on the Ethereum Sepolia test network with a limited number of student records and credentials, and the evaluation established functional conformance together with the correctness of the access control rules of the smart contract. It did not establish operational readiness for large-scale deployment, which would require validation on a main network with real transaction costs, an independent security audit, measurement under a realistic number of concurrent users, and an assessment of the institutional arrangements needed for a school to act as a credential issuer.

Further work should place the JSON-LD context, the credential schema, the issuer DID document, and the status service on a stable public domain, demonstrate the presentation and selective disclosure flows end to end with third-party wallets, carry out deeper security testing through penetration testing and a smart contract audit,

*name of corresponding author



This is anCreative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

and study the operational readiness of the blockchain in terms of transaction cost, confirmation speed, and the management of gas funds by the school.

REFERENCES

- Ardiansyah, Risnita, & Jailani, M. S. (2023). Teknik pengumpulan data dan instrumen penelitian ilmiah pendidikan pada pendekatan kualitatif dan kuantitatif. *IHSAN: Jurnal Pendidikan Islam*, 1(2), 1–9. <https://doi.org/10.61104/ihsan.v1i2.57>
- Barokah, M., & Sauda, S. (2022). Penerapan Node.js dan PostgreSQL sebagai backend pada aplikasi ecommerce Localla. *INFOTECH Journal*, 8(2), 101–105. <https://doi.org/10.31949/infotech.v8i2.2944>
- Butincu, C. N., & Alexandrescu, A. (2024). Design Aspects of Decentralized Identifiers and Self-Sovereign Identity Systems. *IEEE Access*, 12, 60928–60942. <https://doi.org/10.1109/ACCESS.2024.3394537>
- Chan, W., Gai, K., Yu, J., & Zhu, L. (2025). Blockchain-Assisted Self-Sovereign Identities on Education: A Survey. *Blockchains*, 3(1), 3. <https://doi.org/10.3390/blockchains3010003>
- Chotikakamthorn, N., Mi San, A., & Sathitwiriawong, C. (2024). On-Chain Verifiable Credential with Applications in Education. *ECTI Transactions on Computer and Information Technology*, 18(3), 342–355. <https://doi.org/10.37936/ecti-cit.2024183.256091>
- Čučko, Š., & Turkanović, M. (2021). Decentralized and Self-Sovereign Identity: Systematic Mapping Study. *IEEE Access*, 9, 139009–139027. <https://doi.org/10.1109/ACCESS.2021.3117588>
- Dasmen, R. N., Rasmila, R., Widodo, T. L., Kundari, K., & Farizky, M. T. (2023). Pengujian penetrasi pada website elearning2.binadarma.ac.id dengan metode PTES (Penetration Testing Execution Standard). *Jurnal Komputer dan Informatika*, 11(1), 91–95. <https://doi.org/10.35508/jicon.v11i1.9809>
- Fartitchou, M., Lamaakal, I., El Makkaoui, K., El Allali, Z., & Maleh, Y. (2025). BlockMEDC: Blockchain Smart Contracts System for Securing Moroccan Higher Education Digital Certificates. *IEEE Access*, 13, 39152–39175. <https://doi.org/10.1109/ACCESS.2025.3546177>
- Flamini, A., Sciarretta, G., Scuro, M., Sharif, A., Tomasi, A., & Ranise, S. (2024). On cryptographic mechanisms for the selective disclosure of verifiable credentials. *Journal of Information Security and Applications*, 83. <https://doi.org/10.1016/j.jisa.2024.103789>
- Grech, A., Sood, I., & Ariño, L. (2021). Blockchain, Self-Sovereign Identity and Digital Credentials: Promise Versus Praxis in Education. *Frontiers in Blockchain*, 4. <https://doi.org/10.3389/fbloc.2021.616779>
- Hoops, F., Muhle, A., Matthes, F., & Meinel, C. (2023). A Taxonomy of Decentralized Identifier Methods for Practitioners. *Proceedings of the 2023 IEEE International Conference on Decentralized Applications and Infrastructures*, 57–65. <https://doi.org/10.1109/DAPPS57946.2023.00017>
- Kersic, V., Vidovic, U., Vrecko, A., Domajnko, M., & Turkanovic, M. (2023). Orchestrating Digital Wallets for On- and Off-Chain Decentralized Identity Management. *IEEE Access*, 11, 78135–78151. <https://doi.org/10.1109/ACCESS.2023.3299047>
- Mazzocca, C., Acar, A., Uluagac, S., Montanari, R., Bellavista, P., & Conti, M. (2025). A Survey on Decentralized Identifiers and Verifiable Credentials. *IEEE Communications Surveys and Tutorials*, 27(6), 3641–3671. <https://doi.org/10.1109/COMST.2025.3543197>
- Peffer, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45–77. <https://doi.org/10.2753/MIS0742-1222240302>
- Petcu, A., Pahontu, B., Frunzete, M., & Stoichescu, D. A. (2023). A Secure and Decentralized Authentication Mechanism Based on Web 3.0 and Ethereum Blockchain Technology. *Applied Sciences (Switzerland)*, 13(4). <https://doi.org/10.3390/app13042231>
- Robles-Carrillo, M. (2024). Digital identity: an approach to its nature, concept, and functionalities. *International Journal of Law and Information Technology*, 32, 19. <https://doi.org/10.1093/ijlit/eaee019>
- Román-García, F., Hernández-Serrano, J., & Esparza, O. (2024). DA2Wa: A secure pairing protocol between a DApp and a wallet for the blockchain scenario. *Computer Communications*, 224, 1–15. <https://doi.org/10.1016/j.comcom.2024.05.019>
- Sarwar, M. I., Maghrabi, L. A., Khan, I., Naith, Q. H., & Nisar, K. (2023). Blockchain: A Crypto-Intensive Technology - A Comprehensive Review. *IEEE Access*, 11, 141926–141955. <https://doi.org/10.1109/ACCESS.2023.3342079>
- Schardong, F., & Custódio, R. (2022). Self-Sovereign Identity: A Systematic Review, Mapping and Taxonomy. *Sensors*, 22(15), 5641. <https://doi.org/10.3390/s22155641>
- Silaghi, D. L., Artenie, A. C., & Popescu, D. E. (2025). Optimizing Teacher Portfolio Integrity with a Cost-Effective Smart Contract for School-Issued Teacher Documents. *Computers*, 14(9). <https://doi.org/10.3390/computers14090395>
- Sporny, M., Longley, D., Chadwick, D., & Herman, I. (2025). *Verifiable Credentials Data Model v2.0*. Retrieved from <https://www.w3.org/TR/2025/REC-vc-data-model-2.0-20250515/>

- Sporny, M., Longley, D., Sabadello, M., Reed, D., Steele, O., & Allen, C. (2022). *Decentralized Identifiers (DIDs) v1.0: Core Architecture, Data Model, and Representations*. Retrieved from <https://www.w3.org/TR/2022/REC-did-core-20220719/>
- Sultana, S. A., Rupa, C., Malleswari, R. P., & Gadekallu, T. R. (2023). IPFS-Blockchain Smart Contracts Based Conceptual Framework to Reduce Certificate Frauds in the Academic Field. *Information (Switzerland)*, 14(8). <https://doi.org/10.3390/info14080446>
- Tuunanen, T., Winter, R., & vom Brocke, J. (2024). Dealing with complexity in design science research: A methodology using design echelons. *MIS Quarterly: Management Information Systems*, 48(2), 427–458. <https://doi.org/10.25300/MISQ/2023/16700>