Extending EUDI Wallet Credential Flows with x402-Based Payments

Vid Keršič[1,*], Martin Domajnko[1] and Muhamed Turkanović[1]

[1]Faculty of Electrical Engineering and Computer Science, University of Maribor, Institute of Informatics, Blockchain Lab:UM, Maribor, Slovenia

Abstract

The European Digital Identity Wallet (EUDIW) is expected to become a key infrastructure component for issuing, storing, and presenting digital credentials across the European Union. However, current EUDIW credential flows do not provide a protocol-native mechanism for payment-gated credential issuance or verification. This paper proposes a reference architecture for extending EUDI Wallet credential flows with x402-based payments. The architecture introduces an x402 payment layer in front of OpenID4VCI issuance and OpenID4VP verification services, enabling payment authorization before credential transactions are created while preserving standardized wallet-facing protocol interactions. We describe the architectural components, design principles, and payment-gated issuance and verification workflows, and validate the approach through a proof-of-concept implementation. The validation demonstrates that unpaid requests are rejected, successful payments unlock the intended credential service operation, and existing credential wallets can complete the subsequent OpenID4VCI or OpenID4VP interaction without x402-specific modifications. The results indicate that protocol-native payments can be integrated into EUDI Wallet credential services while maintaining interoperability, separation of concerns, and clear trust boundaries between payment and credential processing.

Keywords

digital identity, EUDIW, verifiable credentials, payments, x402

1. Introduction

European Digital Identity Wallets (EUDIW) are expected to become a key component of the European Union's future digital infrastructure, with each Member State (EU MS) required to provide wallets to its citizens by the end of 2026 [1]. EUDIW is intended to make digital identity universally accessible by providing every citizen with a secure and interoperable digital wallet. While citizens may already possess digital identities through existing national systems, EUDIW enables them to seamlessly receive, store, and share personal digital documents across digital platforms and real-life scenarios, including government IDs, driving licences, diplomas, concert tickets, and other verifiable credentials. To enable cross-border adoption, EUDIW are being released and developed according to the Architecture and Reference Framework (ARF), which defines a common technical foundation for interoperable wallet implementations, including ecosystem roles, interfaces, trust relationships, data models, and credential issuance and presentation flows [2]. This enables interoperable adoption across all EU MS.

Although the ARF defines technical details for issuance, verification, trust establishment, and many other flows and models, payment handling remains outside the core framework. This is notable because enabling payments is one of the fundamental functions traditionally associated with both physical and digital wallets, alongside the storage and presentation of credentials. This is furthermore important because many credential-related services can be monetizable, including the issuance of specialized attestations or access to certain verification services. The ARF does address the use of wallet units for strong user authentication (SCA) in electronic payments, where the wallet can authenticate transaction information such as the amount and payee [2]; however, this is different from defining a protocol-native

SQAMIA 2026: 13th Workshop on Software Quality Analysis, Monitoring, Improvement, and Applications, September 16–18, 2026, Niš, Serbia

vid.kersic@um.si (V. Keršič); martin.domajnko1@um.si (M. Domajnko); muhamed.turkanovic@um.si (M. Turkanović)
https://ii.feri.um.si/en/person/vid-kersic-2/ (V. Keršič); https://ii.feri.um.si/sl/person/martin-domajnko/ (M. Domajnko); https://ii.feri.um.si/en/person/muhamed-turkanovic-2/ (M. Turkanović)
0000-0002-7340-7501 (V. Keršič); 0009-0005-6640-5822 (M. Domajnko); 0000-0002-5079-5468 (M. Turkanović)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



published 2026-11-19

CEUR-WS.org/Vol-4336/paper18.pdf

mechanism for pricing, requesting, and settling payments inside credential issuance or verification flows. As a result, implementers may rely on external payment systems, such as card payments, account-based billing, invoices, or proprietary APIs, which can introduce additional integration complexity, user friction, and transaction costs [3].

Recent developments in HTTP-native payment protocols create an opportunity to address this gap. The *HTTP 402 Payment Required* status code was originally reserved to support digital cash or micropayment systems from the beginning of the internet, but it was never used in practice due to the inefficiency of micropayments at the time [4]. Protocols such as x402 and the Machine Payments Protocol (MPP) revisit this unused design space by embedding payment negotiation into HTTP-based request–response flows [5, 6]. x402 defines an open payment standard that enables services to charge for access to APIs or digital resources directly over HTTP, using the aforementioned status code to communicate payment requirements and allowing clients to submit a payment payload before receiving the requested resource. These protocols are enabled by newer financial infrastructures, including stablecoins, blockchains, and distributed ledger technologies (DLTs), which make low-value, programmatic, and pay-per-request payments practical and usable [7, 8, 9].

1.1. Aim and Contribution

The aim of this paper is to bridge the identified gap by introducing a protocol-native payment layer into EUDIW credential flows. Specifically, the paper investigates how HTTP-native payment mechanisms, particularly x402, can be integrated directly into EUDIW issuance and verification flows, rather than relying on external, application-specific payment systems.

The main contribution of this paper is the proposal of a reference architecture for x402-enabled payments in the EUDIW ecosystem. The proposed architecture defines how payment requests, payment proofs, and payment validation can be incorporated into credential issuance and presentation workflows. In addition to the conceptual model, the paper presents a reference implementation that demonstrates the feasibility of integrating x402 into EUDIW-compatible flows. To the best of our knowledge, this is the first reference architecture and implementation combining EUDIW and x402.

2. Preliminaries and Related Work

2.1. User-centric identity, eIDAS 2.0 and European Digital Identity (EUDI)

The idea of user-centric digital identity has been developed for more than a decade in scientific communities and industry under different labels, such as *user-centric identity*, *self-sovereign identity* (SSI), and *decentralized identity* [10, 11]. Despite differences in governance and technical implementation, these approaches share a common objective: users should hold identity credentials within their own software or hardware environment, selectively disclose them, and decide when, with whom, and for what purpose their personal data is shared. Several technical standards and protocols have been developed to support this objective, including decentralized identifiers (DIDs) and verifiable credentials (VCs) [12, 13].

Within this general model, each participating actor takes one of three roles: an *issuer* creates and signs an attestation about a subject; a *holder* (i.e., the subject) stores credentials and presents them under user control; and a *verifier*, often also called a *relying party*, requests and validates presentations in order to provide a service or make an authorization decision.

In the European Union, these user-centric identity ideas were not adopted as a pure SSI model, but rather as a regulated and interoperable public-market deployment under the revised eIDAS framework. The original eIDAS Regulation was adopted in 2014 and established a legal framework for electronic identification and trust services for electronic transactions [14]. However, the original framework was limited in its ability to support more advanced and user-controlled forms of digital identity. For this reason, the revised European Digital Identity (EUDI) framework, commonly referred to as eIDAS 2.0, was introduced, adopted in 2024, and entered into force in the same year. eIDAS 2.0 establishes the framework, while the EUDIW represents its concrete technical realization for end users. Under this

framework, each EU MS must provide at least one EUDIW to citizens by the end of 2026 [1]. The Architecture and Reference Framework (ARF) provides the common technical basis for interoperable wallet development and operationalizes the legal requirements of eIDAS 2.0 into ecosystem roles, interfaces, trust relationships, credential formats, and interoperable issuance and presentation flows [2].

In the current EUDI technical stack, several technical standards and protocols are defined rather than a single one. The official EUDI standards catalogue lists ISO/IEC 18013-5 and ISO/IEC 18013-7 for *mdoc/mDL* credentials, Selective Disclosure for JSON Web Tokens (SD-JWT) for VCs, OpenID for Verifiable Credential Issuance (OpenID4VCI) 1.0, OpenID for Verifiable Presentations (OpenID4VP) 1.0, as well as specific profiles for OpenID4VC, such as the High Assurance Interoperability Profile (HAIP) [2]. At the trust layer, EUDI also relies on European Telecommunications Standards Institute (ETSI) certificate profiles and trusted lists for wallets, Person Identification Data (PID) / Electronic Attestations of Attributes (EAA) providers, and relying parties. This represents the eIDAS certificate-based Public Key Infrastructure (PKI) layer, commonly realized with X.509-compatible (qualified electronic) certificate profiles and trust anchors [2].

2.2. OpenID for Verifiable Credentials

OpenID for Verifiable Credentials defines how a wallet obtains credentials from issuers and shares them with verifiers or relying parties. It builds upon OAuth 2.0 and OpenID-style discovery [15]. OpenID4VCI defines how a (holder) wallet obtains a credential from an issuer. For example, in the authorization-code flow, the wallet either initiates issuance itself or receives a *Credential Offer*, fetches issuer metadata, sends an authorization request, receives an authorization code, exchanges it for an access token, and finally calls the credential endpoint with the token and proof-of-possession information to obtain the credential [16]. The structure of the protocol cleanly separates discovery, user authorization, token issuance, and credential issuance.

Similarly, OpenID4VP defines the complementary presentation protocol used when a verifier requests one or more credentials from a (holder) wallet [17]. In this flow, the verifier first creates an authorization request that specifies which credentials or claims are required. In a same-device flow, this request is sent directly to the wallet, while in a cross-device flow it is usually transferred through a QR code containing a client identifier and a *request_uri*. The wallet then retrieves and evaluates the request, identifies matching credentials, and asks the user for authentication and consent. After approval, the wallet generates the requested verifiable presentation and returns it to the verifier. The presentation is bound to the verifier and the specific transaction context through specific parameters, which help prevent replay attacks and presentation injection.

2.3. x402 Payment Protocol

The HTTP status code `402 Payment Required` has existed since the original HTTP/1.1 specification from 1997, where it was reserved for payment purposes but was never widely adopted due to inefficient digital payment infrastructure until 2025. x402 revisits this dormant design space by making payment negotiation native to standard HTTP request-response flows and by using newer financial primitives and improvements developed over the last decade, especially stablecoins and blockchains. The protocol was primarily developed by Coinbase and was later moved to the Linux Foundation [5].

The typical x402 control flow is as follows. First, a client requests a protected resource. If no acceptable payment is attached, the resource server returns *402 Payment Required* together with machine-readable payment requirements. The client then chooses one payment option, constructs a payment payload, and retries the HTTP request with a payment header. The server verifies the payload locally or through a facilitator, and, if verification succeeds, returns the originally requested resource together with a payment response header or receipt. What makes this revived approach practically viable is the emergence of low-cost and programmable digital asset infrastructure, which enables micropayments as low as a few cents on high-performance blockchains and layer 2s, such as Solana and Base, as well as pay-per-request flows [5].

Another payment protocol that extends x402 is MPP. Unlike x402's custom-header approach, MPP is standardized as the HTTP *Payment* authentication scheme in an IETF Internet-Draft and uses the standard *WWW-Authenticate: Payment* challenge and *Authorization: Payment* credentials [6].

2.4. Related Work

Several existing works in academic, industry, and open-source communities have explored aspects of payments, incentives, digital identity, and verifiable credentials. The authors in [18] propose multi-agent economies that integrate ledger-anchored identities with x402-based micropayments, and argue that autonomous agents require verifiable identities and payment mechanisms to discover, authenticate, and compensate one another. In the broader self-sovereign identity ecosystem, prior work emphasizes that issuers, holders, and verifiers must be properly incentivized for decentralized credential systems to achieve adoption [19]. The OpenID Foundation also identifies payments and business models as relevant considerations for verifiable credential ecosystems [20]. In the education domain, studies on microcredentials show that users may be willing to pay for credential-related services and that pricing can influence learning behavior [21]. In addition, open-source projects such as [22] by RankOneLabs demonstrate early experimentation in combining credential issuance with x402-based payment flows. However, to the best of our knowledge, existing work has not yet combined x402 with EUDIW credential flows or proposed a reference architecture and implementation for protocol-native payments in the EUDIW ecosystem.

3. Proposed architecture

This section presents the proposed architecture for extending EUDI Wallet credential issuance and verification flows with protocol-native x402 payments. The architecture preserves the standardized OpenID4VCI and OpenID4VP interactions defined by the EUDI Wallet ecosystem while introducing payment authorization before protected credential service operations. Rather than modifying wallet-facing credential exchanges, x402 is applied at the service boundary, enabling payment-gated credential services without affecting interoperability between EUDI Wallet implementations.

The section first presents the overall architecture and its main components, followed by the architectural design principles that guided the proposed solution. It then describes the payment-gated credential issuance and verification workflows.

3.1. Architecture Overview

The proposed architecture comprises six logical elements: the **credential holder**, **integrating applications**, a **payment wallet**, a **credential wallet**, the **OIDCx402 service**, and the **external payment infrastructure**. Figure 1 presents these elements and their interactions as a C4 container diagram, while Table 1 summarizes the primary responsibility of each architectural component.

Table 1
Primary responsibilities of the components in the proposed architecture.

Component	Primary responsibility
Payment wallet	Authorize x402 payments.
Credential wallet	Receive, store, and present credentials.
x402 payment gate	Enforce payment before protected service operations are executed.
Issuer service	Create credential offers and issue credentials using OpenID4VCI.
Verification service	Create verification sessions and validate credential presentations using OpenID4VP.
Payment facilitator	Verify and settle x402 payments.

The credential holder interacts with integrating applications that initiate either credential issuance or credential verification. During issuance, an application requests the creation of a credential offer,

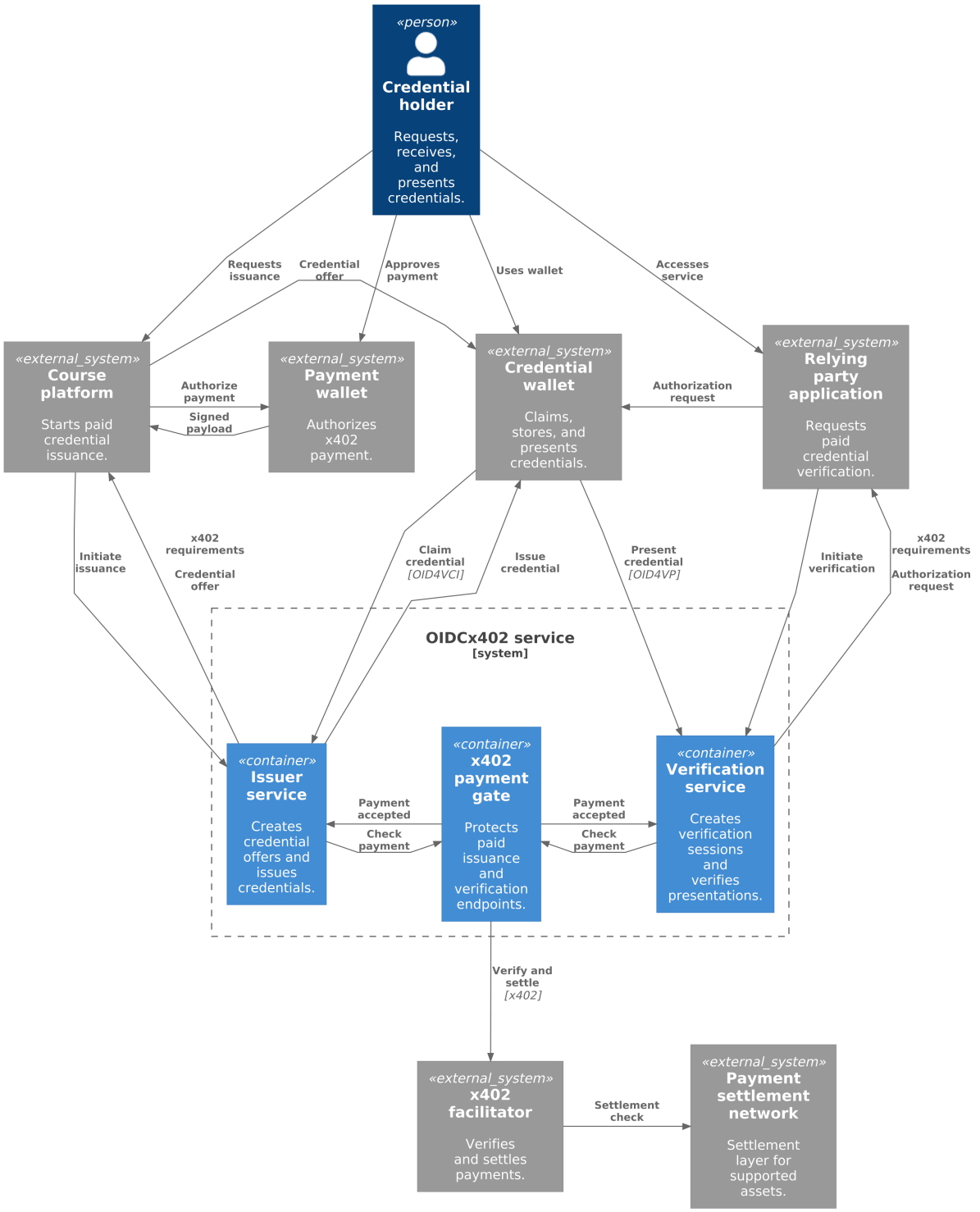


Figure 1: C4 container view of the proposed architecture.

whereas during verification, a relying-party application requests the creation of a credential verification transaction. Credential storage and presentation are handled by an EUDI Wallet-compatible credential wallet implementing the OpenID4VC protocols, while payment authorization is performed by the payment wallet. These roles are logically independent and may therefore be implemented either by the same wallet application or by separate applications.

The OIDCx402 service provides the integration layer between protocol-native payments and OpenID for Verifiable Credentials. It consists of three components: an *x402 payment gate*, an *issuer service*, and

a *verification service*. The payment gate protects selected service operations by requiring successful x402 payment authorization before forwarding requests to the appropriate credential component. The OIDCx402 service provides the integration layer between protocol-native payments and the OpenID4VC protocols used by the EUDI Wallet ecosystem.

A key architectural decision is the separation of payment processing from credential processing. The external payment infrastructure, comprising the x402 facilitator and settlement network, is responsible only for payment verification and settlement. Credential issuance and presentation continue to follow the standard OpenID4VCI and OpenID4VP protocols without modification [5, 16, 17]. Consequently, existing credential wallets do not require x402 support, as payment is completed before the wallet receives either a credential offer or an authorization request.

This separation also preserves clear trust boundaries between the payment and credential domains. The payment layer determines only whether a protected service operation may proceed. Issuance policies, credential validation, issuer trust decisions, requested attributes, validity conditions, and holder consent remain the responsibility of the corresponding issuer and verification services.

The proposed architecture primarily targets application-specific deployments of non-qualified EAAs and therefore abstracts the complete EUDIW relying-party and intermediary trust infrastructure. In production EUDIW deployments, the managed verification service and its customer applications would additionally require relying-party registration, intended-use and attribute registration, access certificates, intermediary registration, and the associated governance controls defined by the EUDIW ARF to ensure that wallet interactions remain consistent with the registered relying-party context [2].

Both EUDI Wallet credential issuance and verification follow the same architectural pattern. The integrating application requests a protected service operation, the OIDCx402 service authorizes its execution through x402, and the corresponding credential service executes the standard EUDI Wallet protocol flow. The detailed issuance and verification workflows are presented in the following subsections.

3.2. Architectural Design Principles

The proposed architecture is guided by four design principles that extend EUDI Wallet credential flows with protocol-native payments while preserving interoperability with the standardized OpenID4VC protocols.

Separation of concerns. Payment processing and credential processing are treated as independent responsibilities. x402 determines whether a protected service operation may proceed, while credential issuance and verification continue to enforce their own policies, trust decisions, and protocol requirements.

Protocol preservation. Wallet-facing interactions remain fully compliant with OpenID4VCI and OpenID4VP. The architecture does not modify either protocol or require additional wallet capabilities. Existing credential wallets therefore continue to participate using standard protocol exchanges.

Service-level payment enforcement. Payments are enforced at the service boundary rather than within the credential protocols themselves. Issuance and verification services expose protected operations whose execution depends on successful payment, while the resulting credential transactions remain unchanged.

Component independence. The payment wallet and credential wallet are treated as independent components. This allows deployments to combine both functions within a single application or distribute them across separate applications without affecting the architecture or protocol flows.

3.3. Payment-Gated Credential Issuance

During credential issuance, the protected operation is the creation of the credential offer rather than the OpenID4VCI protocol itself. An integrating application first requests the creation of an issuance transaction for a holder who is eligible to receive a credential. Before the transaction is created, the OIDCx402 service requires successful x402 payment authorization. Once payment has been verified

and settled, the issuer service creates the credential offer and returns it to the integrating application. The holder then opens the credential offer using an EUDI Wallet-compatible credential wallet, which completes the standard OpenID4VCI issuance flow to obtain and store the credential [5, 16].

Separating payment authorization from credential issuance preserves the responsibilities of the participating components. The application or issuer determines whether the holder satisfies the eligibility requirements for receiving a credential, whereas x402 authorizes only the execution of the protected issuance operation. Consequently, payment does not replace existing enrollment, authentication, or issuance policies.

The architecture also allows the payer and credential holder to be different entities. For example, an organization or an authorized software agent may pay for credential issuance on behalf of an eligible holder. Regardless of who authorizes the payment, the issued credential remains bound to the intended holder through the standard OpenID4VCI interaction.

Each verified payment authorizes exactly one issuance transaction. A production implementation should therefore associate payment evidence with a single credential issuance request, ensuring that repeated requests resolve to the same transaction instead of generating duplicate payments or duplicate credentials. Figure 2 illustrates how x402 extends the standard EUDI Wallet issuance flow by introducing payment authorization before creation of the credential offer, while leaving the subsequent OpenID4VCI interaction unchanged.

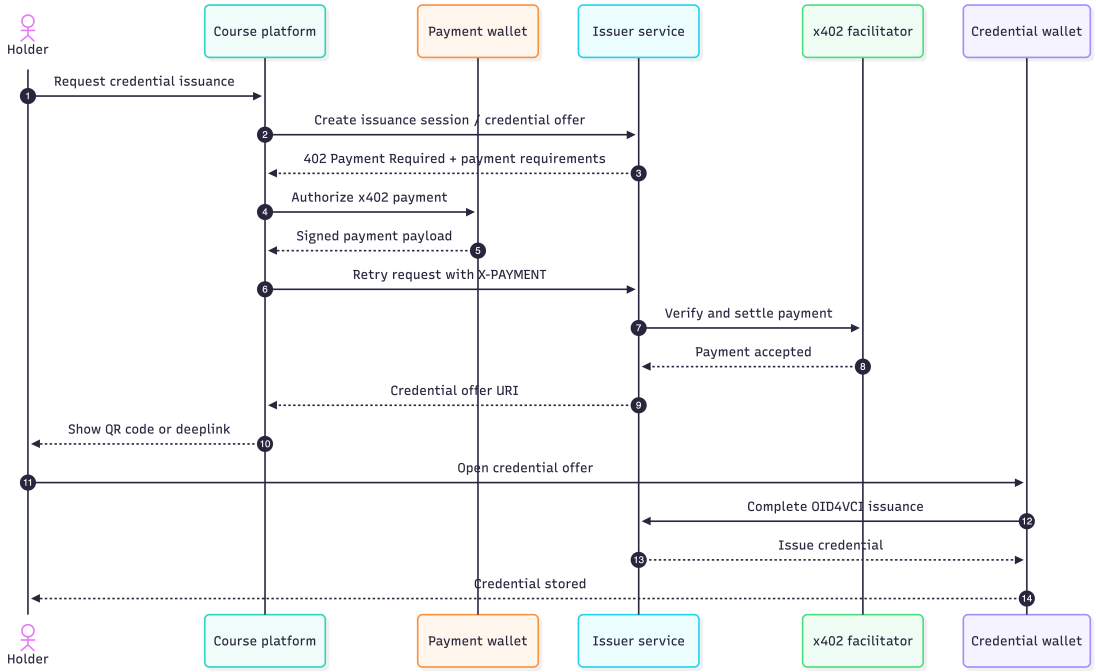


Figure 2: Payment-gated credential issuance using x402 and OID4VCI.

3.4. Payment-Gated Credential Verification

Credential verification follows the same architectural pattern as credential issuance. The protected operation is the creation and processing of a verification transaction rather than the OpenID4VP protocol itself. An integrating application requests a verification transaction by either selecting a predefined verification profile or specifying the credential requirements to be evaluated. After successful x402 payment authorization, the verification service creates the corresponding OpenID4VP authorization request and returns it to the integrating application together with the information required to retrieve the verification result.

The integrating application presents the authorization request to the holder as a QR code or deep link. The holder opens the request in an EUDI Wallet-compatible credential wallet, reviews the requested credentials or attributes, and decides whether to approve the presentation. If consent is granted, the wallet completes the standard OpenID4VP interaction by submitting the presentation to the verification service, which validates it according to the configured verification policy [17].

Payment authorizes only the managed verification service and does not influence the verification outcome. A verification transaction may still fail because the holder rejects the presentation request, the transaction expires, or the submitted credentials do not satisfy the requested verification policy. The integrating application uses the resulting verification outcome to determine whether the protected service operation should proceed.

As with issuance, the payment and credential domains remain independent. The payment infrastructure authorizes execution of the verification service but does not process credential presentations or disclosed attributes. Those remain visible only to the managed verification service responsible for evaluating the presentation.

Each verified payment authorizes one verification transaction, including retrieval of its corresponding verification result. Production implementations should therefore maintain a durable association between payment evidence and the resulting verification transaction to prevent duplicate processing while allowing safe retries. Figure 3 illustrates how x402 extends the standard EUDI Wallet verification flow by introducing payment authorization before creation of the OpenID4VP authorization request while preserving the standardized wallet interaction.

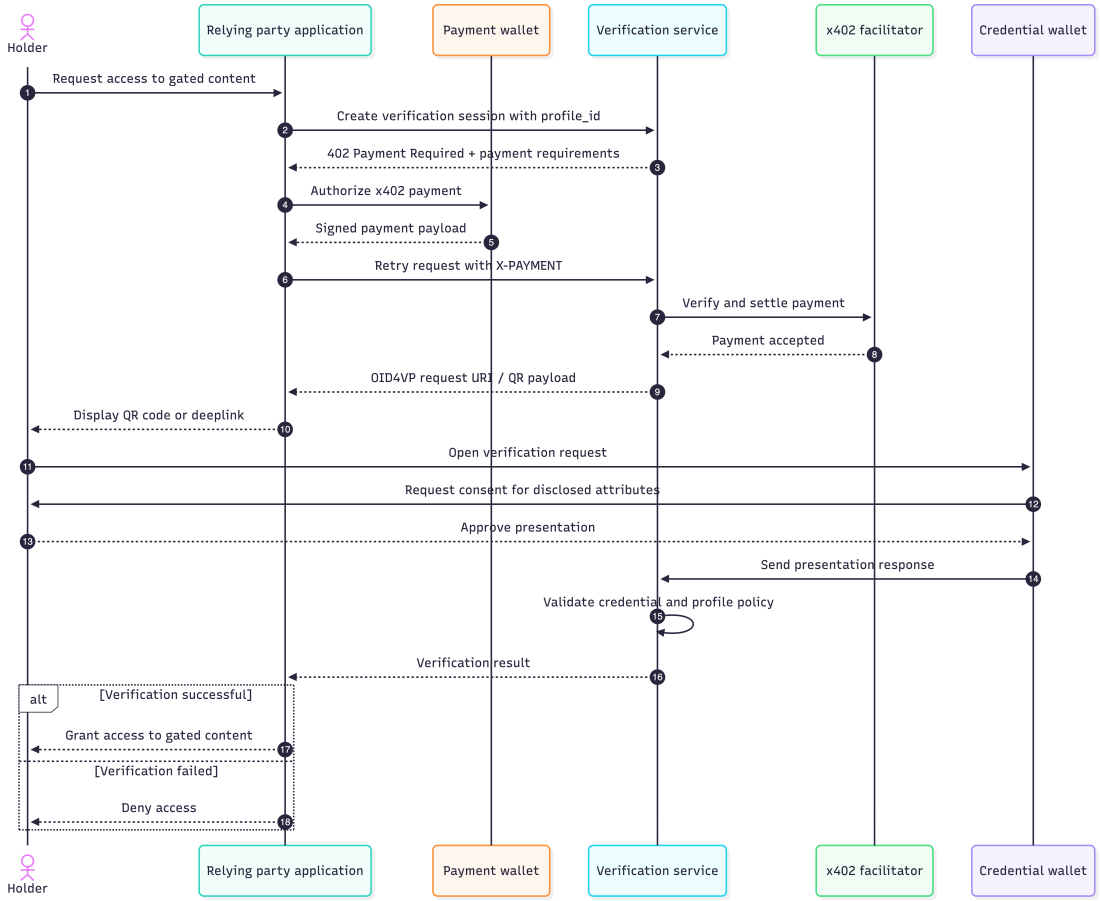


Figure 3: Pay-per-verification using x402 and OID4VP.

The integrating application makes the x402 payment, not the wallet holder. The payment facilitator does not need access to the credentials or attributes disclosed by the holder. The managed verification

service, however, receives and validates the presentation and can associate it with the corresponding paid transaction.

4. Validation

To validate the proposed architecture, we developed a proof-of-concept implementation demonstrating payment-gated credential issuance and verification within EUDI Wallet credential flows. The implementation follows the architecture presented in Figure 1 and realizes the workflows shown in Figures 2 and 3. The objective is to demonstrate that x402 can be introduced into EUDI Wallet credential services without modifying the standardized OpenID4VCI and OpenID4VP interactions implemented by credential wallets.

4.1. Reference Implementation

The proof of concept is implemented in TypeScript [23] using Next.js [24] and React [25]. Privy [26] provides access to embedded or externally connected payment wallets, `viem` [27] is used for blockchain interaction, and `x402-fetch` [28] implements the x402 payment exchange. The prototype uses USDC [29] on the Base Sepolia test network [30].

The implementation follows the layered architecture proposed in Section 3. The x402 payment gate protects the service operations responsible for creating EUDI Wallet issuance and verification transactions. After successful payment authorization and settlement, control is delegated to the issuer or verification service, which executes the standard OpenID4VCI or OpenID4VP interaction expected by EUDI Wallet implementations.

For credential issuance, the issuer service creates OpenID4VCI credential offers after successful payment authorization. For verification, the verification service creates OpenID4VP authorization requests and manages the corresponding presentation sessions. The implementation additionally exposes a protected status endpoint through which integrating applications retrieve verification results after presentation processing has completed.

The implementation intentionally maintains a strict separation between payment processing and credential processing. Payment verification and settlement are delegated to the x402 infrastructure, whereas credential issuance and presentation remain the responsibility of the issuer and verification services. This separation ensures that payment components never process credential contents while credential services remain independent of payment implementation details.

4.2. Functional Validation

The proof of concept was evaluated against the principal architectural properties introduced in Section 3. Table 2 summarizes the validated properties and the corresponding implementation outcomes.

Table 2
Functional validation of the proposed architecture.

Property	Result	Observation
Payment required before issuance	✓	Credential offer created only after payment
Payment required before verification	✓	Authorization request created only after payment
Standard OpenID4VCI preserved	✓	Wallet completed issuance unchanged
Standard OpenID4VP preserved	✓	Wallet completed presentation unchanged
Credential wallet independent of x402	✓	No wallet modifications required
Payment separated from credential processing	✓	x402 completed before protocol execution
One payment authorizes one transaction	✓	Duplicate transaction creation prevented

The results demonstrate that all architectural objectives were achieved. In both issuance and verification workflows, x402 successfully prevented execution of protected service operations until payment had been completed. After authorization, the credential wallet continued using standard OpenID4VCI and OpenID4VP interactions without modification, confirming that payment enforcement can be introduced without changing existing wallet implementations. The results demonstrate that x402 can be integrated into EUDI Wallet credential services while preserving standard wallet interactions and protocol interoperability.

5. Discussion

The proof of concept demonstrates that protocol-native x402 payments can be integrated with EUDI Wallet credential workflows without modifying the underlying OpenID4VCI and OpenID4VP protocols. Rather than embedding payment functionality within credential exchanges, the proposed architecture introduces payment as a prerequisite for protected service operations. This preserves interoperability among EUDI Wallet implementations while enabling issuers and verifiers to monetize credential-related services through a standardized payment mechanism.

The proposed architecture demonstrates one possible approach for introducing protocol-native payments into the EUDI Wallet ecosystem without modifying the standardized credential protocols. By enforcing payment before credential transactions are created, the architecture preserves interoperability among EUDI Wallet implementations while enabling payment-gated issuance, pay-per-verification, and other monetization models for credential services. This approach aligns with the layered architecture promoted by the EUDIW ARF, where payment functionality remains independent of credential exchange and trust establishment.

A key architectural decision is the separation of payment processing from credential processing. The x402 payment layer authorizes access to protected issuance and verification operations but remains independent of the trust decisions performed by the credential services. Consequently, payment does not influence eligibility for credential issuance, credential validation, issuer trust, or holder consent. These responsibilities remain within the existing OpenID4VC ecosystem. This separation of concerns reduces coupling between payment and credential protocols, allowing both to evolve independently while maintaining clear trust boundaries.

An alternative design would integrate payment functionality directly into credential protocols or require EUDI Wallet implementations to support payment capabilities. Such approaches would tightly couple payment and credential processing, reduce compatibility with existing OpenID4VC deployments, and increase implementation complexity for wallet providers. By introducing payment at the service boundary instead of within wallet-facing protocol exchanges, the proposed architecture preserves standardized OpenID4VC interactions while enabling payment-gated credential services with minimal impact on existing implementations.

The proposed architecture also provides deployment flexibility. Since payment and credential wallets are treated as independent components, deployments may either use separate wallet applications or integrate both capabilities into a single user-controlled wallet. Likewise, payment authorization can be performed by the credential holder or by an authorized third party, enabling organizational payment models such as employer-funded credential issuance or application-funded credential verification.

Payment–credential correlation. Because the x402 payment gate and the payment facilitator operate at the service boundary, they process only payment authorization and never observe credential contents or the attributes disclosed during the OpenID4VP presentation exchange. Consequently, the payer's payment identity and the holder's credential identity remain, in principle, separable, a property further reinforced by the architecture's support for distinct payer and holder entities (Section 3). Nevertheless, a residual privacy risk arises when a single operator controls both the payment gate and the corresponding credential service, since such an operator could correlate a settled payment with the credential transaction that it authorizes. This risk can be mitigated by enforcing logical and operational separation between the payment and credential domains, by binding payment evidence to a single

transaction without persisting the linkage beyond its intended lifetime, and by employing one-time or unlinkable payment references so that repeated interactions cannot be aggregated into a behavioral profile.

Payment–service atomicity. A related consideration concerns the failure case in which a payment is verified and settled but the subsequent credential operation does not complete, for example because issuance is aborted or a verification transaction expires. As stated in Section 3, each verified payment authorizes exactly one credential transaction; to preserve this invariant under failure, a production deployment should treat the association between payment evidence and the authorized transaction as durable and idempotent, so that a retry resolves to the same transaction rather than triggering a second settlement. Where the credential operation cannot be completed at all, the same association enables a compensating action, such as crediting or refunding the payer, thereby ensuring that a settled payment results neither in a double charge nor in a silently lost service.

Regulatory alignment. Beyond the technical architecture, a production deployment of protocol-native payments within the EUDI ecosystem would need to align with the evolving European regulatory framework for digital payments. Because x402 settles payments over blockchain and distributed-ledger primitives, typically denominated in stablecoins, its use for identity-linked transactions intersects with the Markets in Crypto-Assets Regulation (MiCA), which governs the issuance of asset-referenced and e-money tokens in the European Union, as well as with the forthcoming third Payment Services Directive (PSD3) and the accompanying Payment Services Regulation (PSR). A compliant deployment would therefore likely rely on regulated, euro-denominated e-money tokens and on a payment facilitator operating under the appropriate authorization, rather than on the testnet assets used in our proof of concept. We regard a systematic analysis of this regulatory alignment, together with its implications for the trust and governance model defined by the EUDI ARF, as an important direction for future work.

The current work nevertheless has several limitations. First, the proof of concept evaluates functional feasibility rather than performance, scalability, or security characteristics. Second, the implementation currently relies on USDC payments on the Base Sepolia test network. Production deployments within the European Digital Identity ecosystem would require mature euro-denominated stablecoins or other payment infrastructures aligned with European regulatory and operational requirements. Finally, the prototype intentionally abstracts the complete EUDIW relying-party and intermediary trust infrastructure. A production deployment would additionally require compliance with the governance, registration, certificate, and interoperability requirements defined by the EUDIW Architecture and Reference Framework.

Overall, the proposed architecture demonstrates that protocol-native payments can be introduced as an independent architectural layer in front of credential services while preserving existing OpenID4VC interactions. This approach enables new business models for credential issuance and verification without changing the standardized protocols that govern wallet interoperability.

6. Conclusion

This paper presented a reference architecture for extending EUDI Wallet credential issuance and verification flows with protocol-native x402 payments. The proposed approach introduces payment-gated credential services while preserving the standardized OpenID4VCI and OpenID4VP interactions defined by the EUDI Wallet ecosystem. Consequently, existing EUDI Wallet implementations can participate in payment-enabled credential workflows without requiring protocol modifications.

The proof of concept demonstrates the functional feasibility of the proposed architecture. Successful x402 payments authorize the creation of credential issuance and verification transactions, while subsequent wallet interactions remain fully compliant with the corresponding OpenID protocols. The implementation further confirms that payment processing and credential processing can remain architecturally independent, preserving existing trust relationships and enabling protocol-native payments without changing standardized credential exchanges.

Future work will evaluate the proposed architecture on production blockchain networks to assess

performance, scalability, and operational costs under realistic workloads. In particular, we plan to quantify the latency introduced by the payment-authorization step relative to a baseline non-gated OpenID4VCI or OpenID4VP flow, decomposing the added overhead into payment-gate processing, facilitator-side verification, and on-chain settlement. We also plan to investigate additional protocol-native payment mechanisms, including Machine Payment Protocol (MPP), and explore unified wallet architectures that combine credential and payment capabilities while maintaining security, privacy, interoperability, and a seamless user experience.

Acknowledgments

The authors acknowledge the financial support from the Slovenian Research Agency (Research Core Funding No. P2-0057).

Declaration on Generative AI

The authors declare that AI tools were used solely for translation purposes and to improve language clarity. All content, ideas, and conclusions are the authors' own, and the translated material has been reviewed for accuracy.

References

[1] European Parliament and Council of the European Union, Regulation (EU) 2024/1183 of the European Parliament and of the Council of 11 April 2024 amending Regulation (EU) No 910/2014 as regards establishing the European Digital Identity Framework, Official Journal of the European Union, L 2024/1183, 30 April 2024, 2024. URL: https://eur-lex.europa.eu/eli/reg/2024/1183/oj.

[2] European Union, Architecture and Reference Framework (ARF) of the European Digital Identity Wallet, European Digital Identity Wallet documentation, 2025. URL: https://eudi.dev/latest/, accessed: 2026-06-19.

[3] H. Schmiedel, G. Kostova, W. Ruttenberg, The social and private costs of retail payment instruments: a European perspective, ECB Occasional Paper 137, Frankfurt a. M., 2012. URL: https://hdl.handle.net/10419/154590.

[4] R. T. Fielding, M. Nottingham, J. Reschke, RFC 9110: HTTP Semantics, Internet Engineering Task Force (IETF), Request for Comments 9110, 2022. URL: https://www.rfc-editor.org/rfc/rfc9110.html.

[5] x402 Foundation, x402: A Payments Protocol for the Internet, Open-source protocol documentation and reference implementation, 2025. URL: https://github.com/x402-foundation/x402, accessed: 2026-06-19.

[6] Machine Payments Protocol, Protocol Overview: Standardizing HTTP 402 for Machine-to-Machine Payments, MPP Documentation, 2026. URL: https://mpp.dev/protocol, accessed: 2026-06-19.

[7] M. Rauchs, A. Glidden, B. Gordon, G. C. Pieters, M. Recanatini, F. Rostand, K. Vagneur, B. Z. Zhang, Distributed ledger technology systems: A conceptual framework, Available at SSRN 3230013 (2018).

[8] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, H. Wang, Blockchain challenges and opportunities: A survey, International journal of web and grid services 14 (2018) 352–375.

[9] V. Buterin, et al., Ethereum white paper, GitHub repository 1 (2013) 5–7.

[10] A. Preukschat, D. Reed, Self-Sovereign Identity: Decentralized Digital Identity and Verifiable Credentials, Manning, 2021. URL: https://books.google.si/books?id=Nh4uEAAAQBAJ.

[11] Š. Čučko, M. Turkanović, Decentralized and self-sovereign identity: Systematic mapping study, IEEE Access 9 (2021) 139009–139027.

[12] World Wide Web Consortium, Decentralized Identifiers (DIDs) v1.0: Core Architecture, Data Model, and Representations, W3C Recommendation, 2022. URL: https://www.w3.org/TR/did-1.0/, published 19 July 2022.

[13] World Wide Web Consortium, Verifiable Credentials Data Model v2.0, W3C Recommendation, 2025. URL: https://www.w3.org/TR/vc-data-model-2.0/, published 15 May 2025.
[14] European Parliament and Council of the European Union, Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC, Official Journal of the European Union, L 257, 28 August 2014, pp. 73–114, 2014. URL: https://eur-lex.europa.eu/eli/reg/2014/910/oj.
[15] N. Sakimura, J. Bradley, M. Jones, B. de Medeiros, C. Mortimore, OpenID Connect Core 1.0, OpenID Foundation Specification, 2023. URL: https://openid.net/specs/openid-connect-core-1_0.html, accessed: 2026-06-19.
[16] T. Lodderstedt, K. Yasuda, T. Looker, P. Bastian, OpenID for Verifiable Credential Issuance 1.0, OpenID Foundation Specification, 2026. URL: https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html, accessed: 2026-06-19.
[17] O. Terbu, T. Lodderstedt, K. Yasuda, D. Fett, J. Heenan, OpenID for Verifiable Presentations 1.0, OpenID Foundation Specification, 2026. URL: https://openid.net/specs/openid-4-verifiable-presentations-1_0.html, accessed: 2026-06-19.
[18] A. Vaziry, S. R. Garzon, A. Küpper, Towards multi-agent economies: Enhancing the A2A protocol with ledger-anchored identities and x402 micropayments for AI agents, arXiv preprint arXiv:2507.19550 (2025).
[19] M. C. Lacity, E. Carmel, Verifiable credentials in the token economy, in: Blockchains and the Token Economy: Theory and Practice, Springer, 2022, pp. 113–138.
[20] K. N. Chadwick, J. Vercammen, OpenID for verifiable credentials, OpenID Foundation (2022).
[21] P. K. Tee, T.-H. Cham, E. C.-X. Aw, A. Khudaykulov, X. Zhang, Marketing micro-credentials: understanding learners' engagement and willingness to pay more, International Journal of Educational Management 38 (2024) 1001–1020.
[22] RankOneLabs, x402 ZK Credential Demo, https://github.com/RankOneLabs/x402-zk-credential-demo, 2025. GitHub repository, proof-of-concept demonstration project; accessed 19 June 2026.
[23] Microsoft, TypeScript: JavaScript With Syntax For Types, https://www.typescriptlang.org/, 2025.
[24] Vercel, Next.js: The React Framework for the Web, https://nextjs.org/, 2025.
[25] Meta Open Source, React: The Library for Web and Native User Interfaces, https://react.dev/, 2025.
[26] Privy, Privy: Wallet Infrastructure and Authentication for Applications, https://www.privy.io/, 2025.
[27] Wagmi Developers, viem: A TypeScript Interface for Ethereum, https://viem.sh/, 2025.
[28] Coinbase, x402-fetch: An x402 Payment Client for the Fetch API, https://github.com/coinbase/x402, 2025.
[29] Circle, USDC: A Fully Reserved Digital Dollar Stablecoin, https://www.circle.com/usdc, 2025.
[30] Coinbase, Base: An Ethereum Layer 2 Network, https://www.base.org/, 2025.