Passwordless Login with EUDI Wallets

Leveraging Verifiable Credentials for Secure, User-Centric Authentication

A Position Paper by iGrant.io

Authors: Benjamin Hansson, George Padayatti, Lal Chandran

Date: March 2026

Abstract

Traditional password-based authentication remains the dominant method for accessing online services despite well-documented security weaknesses, including credential reuse, susceptibility to phishing, password fatigue, and the operational burden of password reset flows. The European Digital Identity (EUDI) Wallet ecosystem, mandated by eIDAS 2.0 (Regulation (EU) 2024/1183), presents a compelling opportunity to replace passwords entirely by leveraging cryptographically verifiable credentials stored in users' secure, government-backed digital wallets.

This paper presents a practical architecture and reference implementation for passwordless login using EUDI Wallets, demonstrated using the iGrant.io Organisation Wallet Suite integrated with Keycloak as the identity and access management (IAM) platform. By bridging the EUDI Wallet ecosystem with established IAM infrastructure through the OpenID Connect protocol, organisations can offer a “Sign in with EUDI Wallet” capability supporting three credential types: Person Identification Data (PID), Photo ID, and a dedicated Strong Customer Authentication (SCA) Authenticator as specified in TS12 of the EU Architecture and Reference Framework (ARF).

This architecture establishes a foundational authentication layer with broad applicability. In particular, it provides the high-assurance identity verification required before any form of delegated authority can be issued, whether to human representatives or autonomous software agents. The cryptographic trust chain from government-issued wallet credentials, through verified presentation, to authenticated session creates the assurance guarantees needed for sensitive domains, including financial services, healthcare, and government services.

Table of Contents

1. The password problem and the EUDI opportunity

1.1 Background

The digital economy depends on authentication, yet the mechanisms most widely deployed are fundamentally broken. Passwords are the weakest link in the security chain: they can be guessed, phished, stuffed, and breached at scale. The operational costs of password management, including help desk resets, breach notification, and regulatory penalties, represent a significant and growing burden on organisations.

The European Digital Identity Framework, established by Regulation (EU) 2024/1183 (eIDAS 2.0), mandates that each EU Member State provide at least one EUDI Wallet to its citizens by late 2026 [1]. By late 2027, regulated private-sector services that already require strong user authentication, as well as very large online platforms, must accept wallet-based identification at the user's request [2]. This regulatory landscape creates an immediate, practical need for standards-based approaches to integrate EUDI Wallet credentials into existing authentication infrastructures.

The EUDI Wallet is not merely a digital ID card. It is a verifiable credential ecosystem supporting Person Identification Data (PID), Electronic Attestations of Attributes (EAAs), and Qualified Electronic Attestations of Attributes (QEAAs), built on open protocols including OpenID for Verifiable Credential Issuance (OpenID4VCI) and OpenID for Verifiable Presentations (OpenID4VP) [3][4]. Critically, the wallet supports selective disclosure: the ability to prove a specific attribute without revealing the underlying data, aligned with GDPR principles of data minimisation.

1.2 Purpose and scope

This paper presents a practical architecture and reference implementation for passwordless login using EUDI Wallets. The approach bridges the EUDI Wallet ecosystem with established IAM infrastructure (specifically Keycloak) via OpenID Connect, enabling organisations to offer a “Sign in with EUDI Wallet” capability alongside or in place of traditional login methods.

The EUDI Wallet ecosystem supports a range of verifiable credentials that can serve as the basis for passwordless authentication. Examples of credential types that can be used for login include:

- **Person Identification Data (PID):** The foundational identity credential issued by Member States, containing attributes such as legal name, date of birth, and national identifiers. A PID-based login provides high-assurance government-verified authentication.

- **Photo ID:** A credential containing a verifiable photograph alongside identity attributes, enabling visual verification where required. Photo ID-based login supports KYC-grade authentication scenarios.

- **SCA Authenticator:** A dedicated Strong Customer Authentication attestation, as specified in TS12 of the EU Architecture and Reference Framework (ARF) [8], designed for high-assurance authentication in payment and financial services contexts.

These are illustrative examples; the architecture is credential-agnostic by design. Any verifiable credential held in an EUDI Wallet that contains attributes suitable for identity matching can be used for passwordless login, provided the relying party configures an appropriate presentation definition. This extensibility means that as the EUDI ecosystem matures and new credential types emerge (such as professional qualifications, organisational roles, or sector-specific attestations), they can be incorporated into the authentication flow without changes to the underlying infrastructure.

2. Regulatory and standards context

2.1 eIDAS 2.0 and the EUDI Wallet ecosystem

The amended European Digital Identity Framework, eIDAS Regulation (commonly called “eIDAS 2.0”, Regulation (EU) 2024/1183) entered into force on 20 May 2024. The Commission’s Architecture and Reference Framework (ARF) provides the technical blueprint for wallet implementations, specifying data models, cryptographic requirements, trust frameworks, and interoperability profiles [5]. The ARF explicitly favours OpenID for Verifiable Presentations (OpenID4VP) as the presentation protocol, ensuring alignment between the regulatory framework and the open standards ecosystem.

Key regulatory timelines: Member States must make at least one EUDI Wallet available within 24 months of the implementing acts entering into force (targeting late 2026). Private-sector relying parties requiring strong authentication must accept the wallet within 36 months (targeting late 2027). The Commission published the first implementing acts on 4 December 2024, with a second round adopted on 7 May 2025.

2.2 OpenID for Verifiable Credentials (OpenID4VC)

The OpenID for Verifiable Credentials family comprises two principal specifications: OpenID for Verifiable Presentations (OpenID4VP) [3] and OpenID for Verifiable Credential Issuance (OpenID4VCI) [4].

OpenID4VP defines how a verifier (relying party) requests and receives verifiable presentations from a wallet holder. The protocol supports both same-device flows (wallet and verifier on the same device) and cross-device flows (user scans a QR code from a browser using their mobile wallet). It supports multiple credential formats, including SD-JWT VC and ISO mdoc, and includes the Digital Credentials Query Language (DCQL) for expressive credential requests [6]. The OpenID4VP specification was on track for final publication by September 2025, having completed pairwise interoperability testing across multiple implementations worldwide.

2.3 Strong Customer Authentication (SCA) and TS12

Strong Customer Authentication is mandated under PSD2 Article 97(1) for electronic payment initiation and online access to payment accounts [7]. The EUDI Wallet supports all three authentication factor categories required by PSD2: knowledge, possession, and inherence. TS12 of the EU ARF specifies how SCA is implemented using the wallet, introducing the concept of SCA Attestations issued by Account Servicing Payment Service Providers (ASPSPs) into the user's wallet unit [8].

The SCA Attestation model is directly relevant to passwordless login: an SCA Attestation stored in the wallet can serve as a high-assurance authenticator, satisfying regulatory requirements for strong authentication while eliminating password dependency entirely. The architectural pattern, verify identity first, then issue a scoped, purpose-specific credential that can be presented to relying parties, is broadly applicable beyond authentication to scenarios involving delegation of authority, transaction authorisation, and agent credential issuance.

2.4 OpenID Connect as the integration bridge

OpenID Connect (OIDC) is the de facto standard for federated authentication on the web [9]. By positioning the EUDI Wallet verification service as an OIDC-compliant identity provider, organisations can integrate wallet-based authentication into their existing IAM infrastructure without re-engineering their application layer. Claims derived from the verifiable presentation (such as email, name, or national identifier) are mapped to standard OIDC claims, enabling seamless interoperability with Keycloak, Auth0, Ping Identity, Okta, Azure AD, and other identity platforms.

2.5 Passkeys and EUDI Wallet-based login: Complementary approaches

Passkeys, based on the FIDO2/WebAuthn standards, have emerged as the leading passwordless authentication technology for consumer applications. Passkeys use public-key cryptography, bound to a user's device, to provide phishing-resistant authentication without shared secrets. Major platform vendors (Apple, Google, Microsoft) now support passkeys natively, and adoption is accelerating across the web.

Both passkeys and EUDI Wallet-based login eliminate passwords and resist phishing attacks through public-key cryptography. However, they differ fundamentally in what they prove and the trust guarantees they provide:

Property	Passkeys (FIDO2/WebAuthn)	EUDI Wallet-Based Login
What is proven	Device-user binding: the same person who registered is returning	Verified identity: the person is who they claim to be, as attested by a trusted issuer

Identity attributes	None; passkeys carry no identity data	Rich, selectively disclosable attributes (name, date of birth, nationality, etc.)
Issuer trust chain	Self-asserted; no third-party attestation of identity	Government or regulated-sector issuer; cryptographically verifiable trust chain
Assurance level	Device-bound authentication; assurance depends on the initial registration method	High assurance from credential issuance; suitable for eIDAS “high” level of assurance
Cross-service linkability	Unlinkable by design (unique key pair per relying party)	Unlinkable with SD-JWT selective disclosure; linkable if the same attributes are presented
Regulatory acceptance	Not recognised as an eIDAS identification means	Legally mandated under eIDAS 2.0 for public and regulated private services
User experience	Seamless; single biometric gesture on the same device	QR code scan (cross-device) or deep link (same-device); consent screen for attribute disclosure

The two technologies are complementary rather than competing. In a mature deployment, an organisation might use EUDI Wallet-based login for initial identity verification and account binding (establishing who the user is with high assurance), and subsequently offer passkeys for returning-user authentication (providing fast, frictionless re-authentication on trusted devices). EUDI Wallet-based login can also serve as a step-up mechanism when a passkey-authenticated session requires elevated assurance for a sensitive operation, such as initiating a high-value payment or authorising a delegation.

Within the EUDI Wallet itself, device-level authentication (biometrics, device PIN) functions analogously to passkey-style device binding: the wallet uses possession and inherence factors to unlock before any credential presentation occurs. The critical distinction is that the wallet then presents a verifiable credential from a trusted issuer, adding the identity attestation layer that passkeys alone cannot provide. As the FIDO Alliance has noted, public key cryptography—the foundation of both passkeys and wallet credential presentations—is resistant to deepfake-driven impersonation attacks that increasingly threaten knowledge-based and document-upload verification methods.

3. Architecture

3.1 System overview

The passwordless login architecture consists of four principal components interacting through standards-based protocols:

- **EUDI Wallet (User Agent):** The user’s mobile wallet application, holding verifiable credentials (PID, Photo ID, or SCA Attestation) issued by trusted providers. The wallet handles credential storage, user consent, biometric verification, and cryptographic signing of presentations.

- **iGrant.io Organisation Wallet Suite (Verification Service / OIDC Provider):** Acts as the verifier and OpenID Connect identity provider. It generates presentation requests, verifies credential presentations, extracts claims, and issues OIDC tokens [10].

- **Keycloak (Identity and Access Management):** The enterprise IAM platform configured as a relying party to the iGrant.io OIDC provider. It handles session management, user provisioning, attribute mapping, and federation with downstream applications [11].
- **Application (Service Provider):** The end-user application that delegates authentication to Keycloak. It receives standard OIDC tokens and session information, unaware of the underlying wallet-based authentication mechanism.

3.2 Authentication flow

The end-to-end passwordless login flow proceeds through four stages:

Step 1 - Flow Initiation: The user clicks “Sign in with EUDI Wallet” on the application’s login page. The application redirects to Keycloak, which in turn redirects to the iGrant.io OIDC authorisation endpoint. The iGrant.io service generates an OpenID4VP authorisation request that includes a presentation definition specifying which credential attributes are required.

Step 2 - Credential Presentation: The authorisation request is rendered as a QR code (cross-device flow) or delivered via a deep link (same-device flow). The user’s EUDI Wallet receives the request, identifies matching credentials, and presents the user with a consent screen showing exactly which attributes will be shared. The user authenticates locally (e.g., via biometrics) and approves the presentation.

Step 3 - Verification and Token Issuance: The wallet constructs a verifiable presentation, cryptographically signs it, and sends it to the iGrant.io verification endpoint. The service validates the presentation signature, checks credential status (revocation), verifies the trust chain back to the credential issuer, and extracts the requested claims. Upon successful verification, it issues an OIDC authorisation code to Keycloak.

Step 4 - Session Establishment: Keycloak exchanges the authorisation code for ID and access tokens, applies configured attribute mappers to map presentation claims (e.g., presentation.email) to local user attributes (e.g., username), and runs the custom authentication flow. The “Detect existing broker user” step matches the incoming identity to existing local accounts, and the “Automatically set existing user” step completes the login without further interaction. The user is now authenticated and redirected to the application with a valid session.

3.3 Credential types for authentication

The architecture supports three credential types, each suited to different assurance levels and use cases:

Credential Type	Login Attributes	Assurance Level	Typical Use Case
PID	Legal name, date of birth, email, national identifier	High (government-issued)	Government services, banking, and the regulated sector login
Photo ID	Photo, name, date of birth, document number	High (biometric match)	KYC-required services, age-gated access
SCA Authenticator	Authentication factors (possession + inherence), dynamic link data	High (PSD2-compliant)	Payment initiation, account access, and financial services

The choice of credential type is controlled entirely by the presentation definition associated with the OIDC client. Organisations can configure presentation definitions using the Presentation Exchange or DCQL syntax to request specific credential types, attribute sets, and trust anchors [6]. This flexibility allows the same infrastructure to support multiple authentication scenarios: basic login (PID with email only), KYC-grade login (Photo ID with full identity attributes), payment-grade login (SCA Attestation with dynamic linking), or composite login (multiple credentials for role-based access).

4. Implementation with iGrant.io Organisation Wallet Suite

The reference implementation has been built and demonstrated using the iGrant.io Organisation Wallet Suite integrated with Keycloak [10][11]. The implementation follows a seven-step configuration process that can be completed by an organisation's IT administrator without custom software development. The complete implementation guide is available at [12].

4.1 Configuration overview

Step 1 - API Key Provisioning: The organisation obtains an API key from iGrant.io to authenticate all subsequent API calls to the Organisation Wallet Suite.

Step 2 - Enable the OpenID Connect Extension: The OpenID Connect extension is activated on the Organisation Wallet Suite via API, exposing the standard OIDC metadata discovery endpoint (/.well-known/openid-configuration).

Step 3 - Create an OpenID Client: An OIDC client is registered with the iGrant.io service, specifying the presentation definition ID (which determines which credential attributes are requested), the redirect URI (pointing to Keycloak's broker callback endpoint), and the allowed origins. The service returns a client ID and client secret. The callback URI and secret serve a dual purpose: they are also used to receive the verifiable presentation and derive the subject identifier (sub) value that Keycloak uses for user matching.

Step 4 - Configure Keycloak Identity Provider: In Keycloak, create a new OpenID Connect v1.0 identity provider using the client credentials and discovery endpoint from the previous steps. Keycloak automatically discovers all required endpoints (authorisation, token, userinfo, JWKS) from the discovery document.

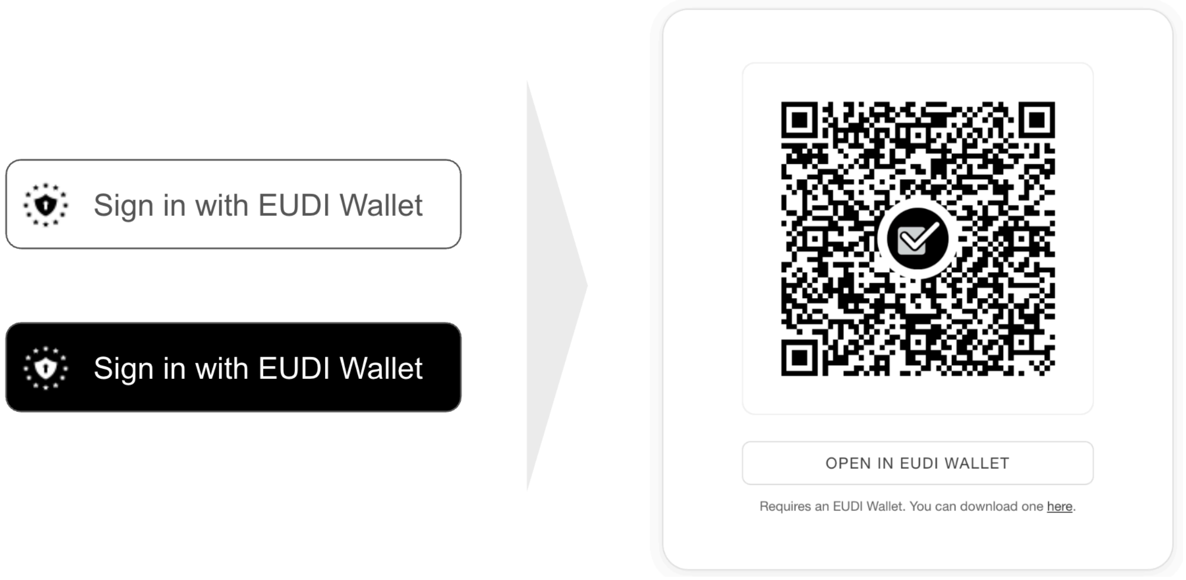
Step 5 - Configure Attribute Mappers: Attribute mappers in Keycloak map claims from the verifiable presentation to local user attributes. For example, the claim “presentation.email” from a PID credential is mapped to the local “username” attribute.

Step 6 - Create a Custom Authentication Flow: A dedicated Keycloak authentication flow (e.g., “Login with EUDI Wallet”) is created with two execution steps: “Detect existing broker user” (checks whether a local account matches the wallet identity) and “Automatically set existing user” (completes login without password entry or email confirmation).

Step 7 - Assign the Authentication Flow: The custom flow is assigned as the post-login flow for the iGrant.io identity provider. The sync mode is set to “force” to ensure attribute mappers are re-applied on every login.

4.2 Frontend integration

A reference React application demonstrating the “Sign in with EUDI Wallet” button is available as open source [13]. The application integrates with Keycloak using standard OIDC libraries. When the user clicks the wallet login button, the application initiates the Keycloak OIDC flow with a hint parameter directing Keycloak to use the iGrant.io identity provider, bypassing the default Keycloak login page and presenting the wallet QR code directly.



Requires an EUDI Wallet. You can download one here.

Figure 1: The passwordless login user experience. The application presents a "Sign in with EUDI Wallet" button (shown in light and dark variants). Upon clicking, the user

is presented with a QR code that can be scanned with their EUDI Wallet, or a deep link ("Open in EUDI Wallet") for same-device flows.

5. Security analysis

5.1 Threat model comparison

The passwordless wallet-based login addresses several categories of authentication threats that plague traditional password-based systems:

Threat	Password-Based	Wallet-Based (This Paper)
Credential phishing	High risk; users tricked into entering passwords on fake sites	Eliminated; cryptographic binding to verifier origin; no credentials transmitted
Credential stuffing	High risk from reused passwords across services	Not applicable; no shared secrets; each presentation is unique
Brute force attacks	Mitigated by rate limiting and complexity rules	Not applicable; device biometrics and cryptographic proofs
Server-side breach	Catastrophic; hashed passwords may be cracked	No credentials stored server-side; only verified claims for the session
Man-in-the-middle	Mitigated by TLS but vulnerable to session hijacking	Cryptographic binding of presentation to verifier; nonce-based replay protection
Social engineering	Users can be tricked into revealing passwords	Wallet consent screen shows exact data shared; device authentication required

5.2 Data minimisation and privacy

The architecture inherently supports the GDPR principle of data minimisation through selective disclosure. The verifiable presentation contains only the attributes specified in the presentation definition. SD-JWT and mdoc credential formats support cryptographic selective disclosure, ensuring that undisclosed attributes cannot be reconstructed from the presentation [14][15]. Presentations are not linkable across different relying parties when using appropriate credential formats, protecting users from cross-service tracking.

5.3 Assurance level considerations

The assurance level of the authentication depends on the credentials presented. PID credentials, issued by Member State authorities, inherently provide high assurance. SCA Attestations provide PSD2-grade authentication with multi-factor verification. Organisations can configure their presentation definitions to require credentials from specific trusted issuers and at specific assurance levels, ensuring that the authentication strength matches the service's risk profile.

6. Applicability to delegated authority and AI Agent credentials

6.1 Authentication as the foundation for delegation

As AI agents increasingly participate in the digital economy as autonomous actors browsing, negotiating, purchasing, and transacting on behalf of humans and organisations, a fundamental question arises: how can an agent's authority be cryptographically bound to a verified human or organisational identity? The answer begins with authentication.

Passwordless wallet-based login provides the trust anchor required for any delegation model. When a user authenticates via their EUDI Wallet, the verifier obtains cryptographic proof of the user's identity directly from a government-issued or regulated-sector credential, with no intermediary credentials (passwords) that could be compromised or repudiated. This high-assurance authentication is a necessary precondition for any form of delegated authority to be issued.

6.2 From authentication to delegation

Once authenticated via the architecture described in this paper, the user's session can serve as the context for issuing delegation attestations. The same OpenID Connect / OpenID4VP infrastructure that powers passwordless human login can be extended to support:

- **Agent credential issuance:** After wallet-based authentication, the user authorises the issuance of a time-limited, scope-restricted credential to an agent (human or software) that can act on their behalf.
- **Delegation chain verification:** When an agent presents a delegated credential, the service can verify both the agent's credential and the original delegation, tracing the chain back to the user's wallet-authenticated identity.
- **Scope management:** The presentation definition mechanism used for login can also express the scope of a delegation, ensuring agents receive only the minimum required privileges.
- **Revocation:** The same credential status infrastructure used to check PID or SCA Attestation validity during login can be reused to check delegation credential status.

6.3 The SCA authenticator as a delegation enabler

The SCA Authenticator credential type is particularly relevant to delegated-agent scenarios in financial services. An SCA Attestation can serve dual purposes: authenticating the user (passwordless login) and subsequently authorising specific transactions performed by a delegated agent. The dynamic linking capability of TS12 ensures that each transaction authorisation is cryptographically bound to its specific parameters, preventing agents from exceeding their delegated authority [8].

The concept of Delegated Agent Credentials, verifiable credentials issued within the EUDI Wallet ecosystem that formally authorise AI agents to act on behalf of verified humans or organisations, represents a natural extension of the authentication architecture presented here. iGrant.io is actively developing this concept as the next layer in the trust stack, building directly on the foundation of passwordless authentication.

7. Open questions and future work

Several areas of further development are identified:

First-time user provisioning: The current implementation requires users to have existing Keycloak accounts. Future work will explore automatic account creation (just-in-time provisioning) upon the first wallet-based login, with appropriate identity-verification workflows.

Multi-wallet interoperability: As different Member States deploy different wallet solutions, the architecture should be validated across multiple EUDI Wallet implementations to confirm interoperability. The OpenID4VP interoperability testing programme provides a framework for this validation [3].

Step-up authentication: Integration with Keycloak's step-up authentication capabilities to dynamically request higher-assurance credentials (e.g., SCA Attestation) for sensitive operations within an authenticated session.

Delegation credential issuance: Extending the architecture to issue OpenID4VCI-based delegation credentials upon successful wallet authentication, enabling the issuance of scoped, time-bound, revocable credentials to AI agents and human representatives acting on the authenticated user's behalf.

Offline and proximity scenarios: Exploring wallet-based authentication for offline or proximity scenarios using ISO 18013-5 mdoc presentation where internet connectivity is unavailable [15].

8. Conclusion

This paper presents a practical, standards-based architecture for passwordless login using EUDI Wallets, implemented and demonstrable with the iGrant.io Organisation Wallet Suite and Keycloak. By leveraging OpenID4VP for credential presentation, OpenID Connect for IAM integration, and the EUDI Wallet ecosystem for credential issuance and management, the architecture eliminates passwords while providing high-assurance, privacy-preserving authentication.

The approach supports three illustrative credential types, namely PID, Photo ID, and SCA Authenticator, while remaining credential-agnostic by design. Any verifiable credential in the EUDI Wallet ecosystem can serve as the basis for passwordless authentication, and the architecture accommodates new credential types as the ecosystem matures. The implementation requires very little to almost no software development, consisting almost entirely of configuration steps that bridge the EUDI Wallet verification service with existing enterprise IAM infrastructure.

Critically, this work is not theoretical. The architecture has been built, deployed, and demonstrated. The same iGrant.io Organisation Wallet Suite that powered the first live EUDI Wallet payment in production [16], in collaboration with Visa, Banca Transilvania, and Worldline, provides the infrastructure for passwordless login described here.

Beyond authentication, this architecture establishes the foundational trust layer upon which delegation, authorisation, and agent credential management can be built. The cryptographic trust chain from government-issued wallet credentials, through verified presentation, to authenticated session provides the assurance required to issue delegated authority to human representatives and AI agents alike, a capability that will be essential as autonomous agents become active participants in the European digital economy.

The infrastructure exists. The standards are mature. The regulation mandates it. What remains is adoption.

References

[1] 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.

[2] European Commission, “EUDI Wallet Architecture and Reference Framework (ARF),” Section on Cross-border Reliance (Article 5f).

[3] OpenID Foundation, “OpenID for Verifiable Presentations 1.0,” openid.net/specs/openid-4-verifiable-presentations-1_0.html.

[4] OpenID Foundation, “OpenID for Verifiable Credential Issuance 1.0,” openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html.

[5] European Commission, “EUDI Wallet Architecture and Reference Framework (ARF),” GitHub: eu-digital-identity-wallet/eudi-doc-architecture-and-reference-framework.

[6] Digital Credentials Query Language (DCQL), as defined in OpenID4VP 1.0. See also: iGrant.io, “Understanding DCQL in the EUDI Wallet Ecosystem,” docs.igrant.io/concepts/eudi-wallet-dcql-openid4vp-business-wallet-payments.

[7] Directive (EU) 2015/2366 (PSD2), Article 97 on Strong Customer Authentication.

[8] European Commission, "TS12: Electronic Payments SCA Implementation with Wallet," EUDI Wallet Standards and Technical Specifications, GitHub: eu-digital-identity-wallet/eudi-doc-standards-and-technical-specifications.

[9] OpenID Foundation, "OpenID Connect Core 1.0," openid.net/specs/openid-connect-core-1_0.html.

[10] iGrant.io, "Organisation Wallet Overview," docs.igrant.io/docs/organisation-wallet-overview.

[11] Keycloak Identity and Access Management, keycloak.org.

[12] iGrant.io, "Implement Passwordless Login with the EUDI Wallet," docs.igrant.io/docs/eudi-passwordless-login-flow.

[13] iGrant.io, "Passwordless Login Playground (Reference Implementation)," GitHub: github.com/L3-iGrant/passwordless-login-playground.

[14] IETF, "SD-JWT-based Verifiable Credentials (SD-JWT VC)."

[15] ISO/IEC 18013-5:2021, Personal identification – ISO-compliant driving licence – Part 5: Mobile driving licence (mDL) application.

[16] iGrant.io powers the first live EUDI Wallet payment in production https://www.igrant.io/news/press-releases/digital-id-gets-real-igrantio-powers-first-live-eudi-wallet-payment-in-production.html

iGrant.io builds European Digital Identity and Business Wallet infrastructure SW. The Organisation Wallet Suite enables issuers, holders, and verifiers of verifiable credentials built on eIDAS 2.0 and EUDI standards. Learn more at iGrant.io and https://docs.igrant.io/docs/organisation-wallet-overview/.