OpenID for Verifiable Credentials

A Shift in the Trust Model Brought by Verifiable Credentials

June 23, 2022

Version: 2nd Editor’s Draft

Lead Editors: Kristina Yasuda, Torsten Lodderstedt, David Chadwick, Kenichi Nakamura, Jo Vercammen

Contents

Executive Summary

OpenID Connect, a protocol that enables deployment of federated Identity at scale, was built with User-Centricity in mind. The protocol is designed so that the Identity Provider releases the claims about the End-User to the Relying Party after obtaining consent directly from an End-User. This enables Identity Providers to enforce consent as the lawful basis for the presentation based on the Relying Party's privacy notice. The protocol also enables two kinds of Identity Providers, those controlled by the End-Users and those provided by the third parties.

Now, User-Centricity is evolving to grant the End-Users more control, privacy and portability over their identity information. Using OpenID for Verifiable Credentials protocols, the End-Users can now directly present identity information to the Relying Parties. This empowers the End-Users to retain more control over the critical decisions when and what information they are sharing. Furthermore, the End-Users' privacy is preserved since Identity Providers no longer know what activity the End-Users are performing at which Relying Party. End-Users also gain portability of their identity information because it can now be presented to the Relying Parties who do not have a federated relationship with the Credential Issuer.

The goal of this whitepaper is to inform and educate the readers about the work on the OpenID for Verifiable Credentials (OpenID4VC) specification family. It addresses use-cases referred to as Self-Sovereign Identity, Decentralized Identity, and User-Centric Identity. The work is being conducted in the OpenID Foundation, in liaison with the Decentralized Identity Foundation (DIF) and with working groups in International Organization for Standardization (ISO), namely ISO/IEC JTC 1/SC 17 Cards and security devices for personal identification[1]. This has enabled working towards aligning ISO-compliant mobile driving licences with W3C verifiable credentials data model, which has been one area of particular interest for the ecosystem.

The whitepaper targets private and public sector decision-makers, architects and implementers interested in the concepts, use-cases, and architecture where the End-User directly receives credentials from the Issuer and directly presents them to the Verifier using verifiable credentials. It is important to note that verifiable credentials are not only limited to credentials expressed using W3C Verifiable Credentials Data Model, but also verifiable credentials expressed using other data models.

First, the basics of the concept of verifiable credentials are described focusing on the shift in the trust model that they bring, their advantages, and clarifying commonly misunderstood concepts.

Next, we elaborate on the benefits and business drivers of transforming Physical Credentials into digital ones in terms of cost, time, and security. This is followed by a section elaborating how the use-cases of verifiable credentials are being realized today, highlighting their value, flexibility, and applicability to a wide range of scenarios and credential formats.

[1] https://www.iso.org/committee/45144.html

Then the technical details of OpenID4VC are presented, alongside an explanation of certain decision choices that were made, such as why OpenID Connect, and OAuth 2.0 are well-suited as basis for presentation and issuance protocols for verifiable credentials.

Finally, the whitepaper concludes by reiterating the importance of making choices for standards that meet certain use-cases in order to realize a globally interoperable verifiable credentials ecosystem.

Achieving large-scale adoption of verifiable credentials will be "by Evolution, not by Revolution". The identity community can more swiftly empower people, and government authorities developing identity infrastructure and policies, by adopting standards like OpenID4VC that facilitate convergence and interoperation of existing and emerging standards.

We welcome feedback on this First Implementor's Draft. We also welcome participation in the Working Group to help progress the standards and align with other standards bodies. Last, we also encourage implementor's to consider joining the GAIN Proof of Concept Community Group to consider testing your implementations. Please find more about these opportunities in the "Conclusion" section of this paper.

Terminology

This specification defines the following terms.

Terms that have already been defined in OpenID Connect Core are used as-is, with modifications if needed.

- **Identifier:** Value that uniquely points to an Entity in a specific context.
- **Identity:** A set of attributes that uniquely identifies a person in a defined context[2].
- **Verifiable Credential (VC)[3]:** A verifiable credential is a tamper-evident credential that has authorship that can be cryptographically verified. This definition is borrowed from W3C Verifiable Credentials Data Model specification, but it is used more broadly, including other data models such as ISO/IEC 18013-5 mDL[4], etc.
- **Entity:** Something that has a separate and distinct existence and that can be identified in a context. An End-User is one example of an Entity.
- **End-User**: Human participant.
- **Wallet**: Entity that receives, stores, presents, and manages credentials and key material of the End-User. There is no single deployment model of a wallet: credentials and keys

[2] https://icma.com/physical-credentials-still-necessary-in-the-age-of-digital-transformation/

[3] Note that this is a different definition than how a term 'credential' has traditionally been used in the identity industry to mean "data presented as evidence of the right to use an identity or other resources" (OpenID.Core) such as passwords, biometrics, etc.

[4] https://www.iso.org/standard/69084.html

can both be stored/managed locally by the end-user, or by using a remote self-hosted service, or a remote third party service.

- **Verifier**: Entity that verifies the credential to make a decision regarding providing a service to the End-User. Also called Relying Party (RP) or Client.
- **Credential Issuer:** Entity that issues verifiable credentials.
- **Identity Provider (IdP) / OpenID Provider (OP):** OAuth 2.0 Authorization Server that is capable of Authenticating the End-User and providing Claims to a Relying Party about the Authentication event and the End-User, given the End-User’s consent. Identity Providers can be government entities, identity service providers, digital platforms, mobile networks, wallet providers, financial institutions or any entity a user trusts to set-up and present identity claims.
- **Self-Issued OpenID Provider:** An OpenID Provider (OP) used by the End-users to prove control over a cryptographically verifiable identifier.
- **Relying Party (RP):** OAuth 2.0 Client application requiring End-User Authentication and Claims from an OpenID Provider.
- **Client**: Application making protected resource requests on behalf of the End-User and with its authorization. The term "client" does not imply any particular implementation characteristics (e.g., whether the application executes on a server, a desktop, or other devices).
- **Wallet Provider:** Entity that is responsible for building, deploying, and running a wallet implementation.

End-User, Wallet, Verifier, and Credential Issuer are entities that can be intermittently assumed by the same actor dependent upon the use-case and the business transaction. For example, the same actor can act as a Verifier to one Wallet in the presentation protocol and as a Credential Issuer to another Wallet in the issuance flow.

Client, Relying Party, OpenID Provider, and Self-Issued OpenID Providers are roles assumed by the entities in the protocol. For example, in the presentation protocol, Wallet assumes the role of the Self-Issued OpenID Provider, and in the issuance protocol, the role of the Client.

Key Takeaways

- Transforming physical credentials into digital credentials have significant benefits in terms of cost, time, security, and End-User-experience of digital services. Verifiable credentials solve the problem that the End-Users currently lack control, privacy, and portability over their identity information, and facilitates the establishment of cross-domain trust among organizations. Using verifiable credentials, the End-Users, whether they are a citizen, employee, or customer, can:
  - retain control over when to disclose which credential to which Verifier;
  - retain control over from which Credential Issuer to obtain what credential;
  - present credentials to the Verifier without the credential issuer knowing, strengthening privacy protection for the End User;

    - present credentials to the Relying Parties who do not have a federated relationship with the Credential Issuer;
    - present multiple credentials issued by different credential issuers in a single presentation; and
    - control their relationship with the Verifiers independent from third party identity providers' decisions or lifespan
- Verifiable credentials resemble the pattern of existing physical credentials where the End-User can present a credential (e.g., a driving licence or passport) to a verifier, without either the End-User nor the verifier directly interacting with the issuer. This approach is also consistent with global trends to enable user consent-based policies and architectures, such as found in privacy legislation (GDPR, CCPA) and the Open Banking/Open Data movement.[5]
- Using the OpenID4VC specification family allows the implementation of a verifiable credentials ecosystem in a secure, interoperable and trusted manner. The specifications are flexible enough to accommodate various use-cases by allowing implementers to make their own choices among other components of the verifiable credentials technical stack: entity identifier types (including DID methods), credential formats, revocation schemes, crypto suites, and trust mechanisms, etc.
- OpenID4VC enables reuse of some of the existing infrastructure and provides a wide availability of the code and libraries, because it is built upon the OpenID Core specification.

Verifiable Credentials: A Paradigm Shift

Benefit of Transforming Physical Credentials into Digital Ones

Transforming physical credentials into digital ones offer the opportunity to make identity verification cheaper, faster, and more secure than current paper-based or semi-online processes, as described in the "Business Drivers of Digitizing Physical Credentials" section of this paper.

Using physical credentials as-is in a digital world has led to processes such as sending a pdf of a passport attached to an email that are susceptible to weaknesses and vulnerabilities. We are at cross-roads to transform our physical documents such as drivers' licences, audited tax returns, birth and marriage certificates, and diplomas into digital verifiable credentials and enable us to present them over the Internet in a trusted, secure, privacy preserving, and interoperable manner.

Verifiable credentials are a promising tool in this transformation journey.

[5] *"Open Banking, Open Data and Financial-Grade APIs,"* lead editor Dave Tonge: https://openid.net/2022/03/18/openid-foundation-publishes-whitepaper-on-open-banking/

Shift in the Trust Model Brought by Verifiable Credentials

The verifiable credentials[6] architecture constitutes a paradigm shift that puts the End-User in the centre of the exchange between the verifier and the credential issuer. It provides greater privacy, portability and control for the End-Users, by enabling End-Users to:

- Present credentials to the verifiers without verifiers directly contacting the credential issuer of that credential
- The use of End-User identifiers that are not namespaced to a certain third-party Identity Provider
- Control their relationship with the Verifiers independent from third party identity providers' decisions or lifespan

This is a big shift in the trust relationships between the actors in the identity ecosystem, i.e., the End-User who possesses the credential, the verifier who verifies the credential, and the credential issuer who issues the credential.

In the currently widely adopted federated identity model, whenever a user wants to access an RP, they are required to visit an IdP and request just in time issuance of the necessary credentials (in the case of OpenID Connect, an ID token). Those credentials are then presented to the RP by some user agent, typically a web browser. A verifier chooses to utilize those claims (or not) based on its relationship with the Identity Provider and its knowledge of the claims verification procedures used by that IdP; it's up to the RP to determine whether the claims provided are suitable for use for its purposes.

This flow enables a lot of the traditional scenarios where the IdP is required to retain the knowledge with which RP the user has interacted with in the past. Such traditional scenarios occur whenever the transaction occurs within the boundaries of the terms of federation agreements between organizations, or whenever business logic needs to be executed depending on the identity of the RP being accessed. However, there are also scenarios where the IdP has no legitimate reason to know which RPs the user wants to access resources from and when they do so. The "traditional" flows cannot realize those scenarios because it is impossible to hide from the IdP when and with which RPs the user interacts with.

It is important to note that using verifiable credentials enables these new scenarios but does not invalidate traditional scenarios.

In verifiable credentials, the verifier receives credentials directly from a wallet under the control of the End-User. A verifier can make a decision to accept those credentials because it can cryptographically verify that 1) they have been issued the credential by an issuer trusted by the

[6] Varying terminology exists, and similar or related concepts are sometimes also referred as Self-Sovereign Identity, self-managed identity, direct presentation model, Decentralised Identity, User-Centric Identity, etc.

verifier; and 2) the wallet used by the End-User to present the credentials is the same wallet to which the credentials have been issued. The most notable feature is that the verifier can receive and validate presented credentials without directly interacting with the Issuer. Note that to prove that it is the same End-User controlling the wallet both during issuance and presentation, additional strong authentication mechanisms are required.

It is important to note that to unleash the power and benefits of utilizing verifiable credentials, the challenge remains how to effectively establish trust in the new ecosystem itself, where the verifiable credentials flow between the issuers and the verifiers through the End-Users. Is the ability to verify the Credential Issuer's signature on a credential enough for the Verifier to accept the verifiable credential and grant the End-User access to its service? Can End-Users use any wallet to manage their verifiable credentials? This new model that involves verifiable credentials is resulting in the emergence of new trust frameworks.

Further Advantages of Verifiable Credentials

As a further advantage, End-Users are able to present multiple credentials issued by different credential issuers in a single presentation, e.g., a COVID pass, and a ticket for access to a conference, which allows for a more seamless End-User experience.

It is important to note that even when using verifiable credentials, Verifiers need to trust the respective Credential Issuer, just like in a federated identity model, where the RPs need to trust Identity Providers. Enabling such trust would require regulatory or contractual relationships on top of technical interoperability.

Using verifiable credentials makes technical implementation easier and simpler, especially in the use-cases where there are hundreds of Credential Issuers, numerous wallets and millions of Verifiers authenticating the End-Users and granting them access to the services. The Verifier, Wallet Provider, and the Credential Issuer establish a relationship once and do not need to contact endpoints of all Credential Issuers.

This has led to the following use-cases that require collaboration between a number of entities being implemented in production:

- Onboarding (e.g., onboarding of employees, customers, suppliers, partners, etc.)
- Entitlement management (e.g., managing access to employee's applications, partner organization's applications, third-party applications on the Internet, physical buildings, etc.)
- Issuance of digital identity credentials by government issuing authorities
- A supply chain process or where traceability of goods handled by multiple entities across national borders is a legal requirement

We will elaborate on these in the "Use-Cases" section of this paper.

Demystifying Myths about Verifiable Credentials

Among many myths, there are four important ones to demystify.

First, verifiable credentials are not equivalent to self-asserted or self-issued claims. The protocols used in verifiable credentials certainly can enable End-Users to present self-asserted or self-issued claims to verifiers. But they can also include verifiable credentials issued by the third-party entities that offer trust services online today, or government entities that issue physical identity credentials today. In short, verifiable credentials are much broader, or a superset of both types of credentials.

Verifiable credentials are not equivalent to self-sovereignty, when it implies the End-User's autonomy and freedom from Issuers and Verifiers, which is hard to achieve in real-life use-cases, especially in the regulated use-cases. Even when the Verifier has obtained the claims directly from the End-User, it is still up to the Verifier to decide whether to accept those credentials and provide the service to the End-User (or not). Regardless of where the End-Use is planning to use a credential, it is still up to the Issuer to decide whether to issue the credential to the End-User in the first place. Even after the issuance, in most cases, the Issuer retains the right to revoke and invalidate the credential.

Second, verifiable credentials are not analogous to the usage of distributed ledger technology (DLT), or blockchains. For the End-Users to directly receive credentials from the Issuers and directly present them to the Verifiers, a mechanism for how the Verifiers obtain the public keys controlled by the Issuers becomes crucial. Decentralized Identifiers (DIDs) leveraging a DLT or blockchain is one useful mechanism to do so. However, not all DIDs rely on DLT or blockchain, and there are other mechanisms as well, such as: obtaining public keys via a PKI or web pages accessible under the domain name controlled by that entity[7]. Other decentralized implementation techniques have their role to play, but they are neither necessary nor sufficient to achieve a verifiable credentials ecosystem.

Third, verifiable credentials are not analogous to use of the W3C Verifiable Credentials data model. Other data models can be used, for example the ISO-compliant mDL model.

Fourth, verifiable credentials can have varying degrees of openness in terms of participation. Some ecosystems, like the ones managed by the government, may require certain permissions or certifications for the wallet application providers, credential issuers and verifiers to join their ecosystem, while others may be completely open to anyone to participate. This is just like federated identity management systems, which allow for various governance and participation models.

[7] e.g. /.well-known/ locations

Various Scopes of “Decentralization”

“Decentralization” is often praised when discussing the benefits of verifiable credentials ecosystems. It is very important to define the scope of “Decentralization” for the purpose of this whitepaper.

One scope of Decentralization is decentralization from a location perspective, which means there is no dependance on one central computer system, i.e., it is a distributed system. This could be a peer-to-peer or a client-server distributed system, both of which have been operational for decades. The OpenID Connect ecosystem can be viewed as a decentralized system from this perspective, with thousands of OpenID Connect servers (IdPs) and millions of clients (RPs) running. However, each individual OP and verifier in this ecosystem is itself typically under the control of a particular party, with the exception of Self-Issued OpenID Providers (Self-Issued OPs), where the OP is under control of the End-User.

Another scope of Decentralization is the End-Users’ ability to bring their own identifiers to the Credential Issuers and Verifiers instead of having identifiers assigned to them by the third-party Identity Providers. W3C Decentralized Identifiers (DIDs) are mentioned the most in this context of identifier decentralization. However, other types of identifiers that are not DIDs can also be used.

There is also a Decentralization scope of the End-User’s ability to present credentials to the verifier, without the verifier having to contact the Credential Issuer directly. W3C Verifiable Credentials (VCs) are being mentioned in this context of decentralization of the credential presentation.

Finally, there is Decentralization from a control perspective, which means not depending on one single body controlling access to the ecosystem. The “NASCAR problem” and invocation of the wallets are usually mentioned in this context. It depends on the use-case and level of assurance of the required credentials and whether any entity is allowed access to the ecosystem, or only certified entities can access the ecosystem. OpenID Connect Core already enables this range of access control within the available technology, but many RPs do not enable complete user choice of OPs. Realizing a completely open and fully decentralized ecosystem might require some technical changes to certain software components such as browsers and mobile Operating Systems.

This paper mainly focuses on the third scope of decentralization as described above, while other three scopes are also touched upon.

Business Drivers of Digitizing Physical Credentials

This section will elaborate on the benefits and business drivers of transforming our physical documents into digital credentials.

There are many problems with physical credentials, whether they are paper-based, or plastic-card based. First, they can be costly to produce[8], especially if they are combined with security features such as watermarks, embossing, holograms, embedded chips, etc. While plastic cards can easily run from low dollars, more secure certificates can run up to €150 per card in production costs[9]. Secondly, they are easy to lose or have stolen, and the owner may not be aware of the loss for hours, days or even weeks. Third they are relatively easy to forge, notwithstanding their security features. Most professional and educational qualification certificates can be openly bought on the Internet[10], whilst fake IDs, passports and COVID-19 certificates[11] can be purchased on the black market. Fourth they are almost impossible to verify reliably without contacting the credential issuer to check their veracity.

One of the more intangible drivers is that both citizens and issuers consider digital verifiable credentials an “inevitable evolution,” a continuation of digitizing other credentials in the physical wallet. Today, credit and debit cards, transit cards, loyalty cards, airline boarding passes, building access credentials, “cash” and crypto currencies are commonly available to consumers to be stored in the applications on their smartphones. In fact, some government issuing authorities have already announced the launch of, or the intention to launch, digital versions of their physical credentials, e.g., States in the United States launching mobile driving licences in partnership with the Department of Homeland Security’s Transportation Security Administration, or the EU Digital Wallet (eIDAS 2.0) request for proposals.

For these reasons many credential issuers are moving towards a digital credentialing system of some sort. In the simplest of cases this might only be for verification of existing physical credentials, whilst in others it will be to supplement physical credentials with electronic ones in order to both issue and verify them electronically. Electronic credentialing systems could solve the aforementioned problems of physical credentials. They are very cheap to issue, with a marginal cost of almost zero to low dollars after initial installation of the issuance system (depending on the required level of security and other considerations). They also increase the trust in and the reputation of the credential issuer, as they can be validated cryptographically, instead of relying on the various physical protection mechanisms that can get lost, faked, or

[8] For example, one UK university pays more than £1.40 per student ID card and issues nearly twenty thousand per year. It cost the UK government £10.79 to produce a UK passport in 2011 (https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/118636/17949-breakdown-costs-passport.pdf)

[9] https://publications.jrc.ec.europa.eu/repository/bitstream/JRC108255/jrc108255_blockchain_in_education(1).pdf

[10] Fake degree certificates can be openly purchased online from https://www.buydiydiploma.com/

[11] https://www.dailymail.co.uk/news/article-8871923/Passengers-use-fake-negative-Covid-test-certificates.html

disregarded. Digital Credentials can be verified almost instantaneously thereby simultaneously reducing the cost to both the verifier and the credential issuer, whilst virtually eliminating the incidence of fraud[12].

Let us look at the case of validating paper-based degree certificates issued by UK universities as an example. The British Council in Oman offers a degree validation service for 52 US dollars[13].
The process is as follows:

1. Visit our office with your original education qualification certificate and ID.
2. Complete the verification request form and submit a consent letter. We will then email a copy of the consent letter along with a copy of your document to the awarding UK institution for confirmation.
3. We will contact you once we receive confirmation of the authenticity of your qualification.
4. You may then visit us with your documents to have it stamped.

The verification process is both expensive and time consuming. Not only that, but adding a stamp to a physical document does not provide any extra security, as the degree certificate with a stamp can be reproduced just as easily as a degree certificate without the stamp.

A service called HEDD provides a semi-online service to UK businesses who want to verify the university degree of a consented individual[14]. The process is as follows:

1. The business Verifier creates an account with HEDD. This involves providing a business Verifier name, organization name, organization address and End-User's email address.
2. Manual validation of these details then takes place, and the email address is verified by sending a confirmation email and secret URL to the provided email address. In the test case conducted by the writers of this whitepaper, the manual validation of the End-User's details took 3 hours.
3. The business collects from the individual whose degree it wants to validate details of the university degree, and a completed and signed degree verification consent form.
4. The business Verifier completes an application form on the HEDD web site, enters details of the graduate (including date of birth) and their degree (classification etc), uploads a copy of the signed consent form, and pays the required fee. The cost is between £12 and £37 (+20% VAT) per verification depending upon the university, although most universities charge £12.

[12] Fraudulent electronic COVID-19 certificates have been on sale, but these have been due to corrupt employees issuing cryptographically genuine certificates from genuine laboratories, for example see: https://www.bbc.co.uk/news/world-europe-54839434

[13] https://www.britishcouncil.om/en/study-uk/verification-uk-education-services

[14] https://hedd.ac.uk/

5. After paying the fee and submitting the consent form, the business Verifier is then advised to wait for an email notification.In the test case conducted by the writers of this whitepaper, this took 16 days.

Again, one can see how these semi-online processes can be very expensive and time consuming for the business Verifier.

Now let's compare and contrast this with a digital credentialing issuance and verification system. Technical details of how to deploy such a system using W3C Verifiable Credentials with OpenID Connect are explained in the section "Technical 101 of OpenID Connect and OpenID4VC".

As a prerequisite, the University needs to issue a digital degree certificate so that the End-User can store it in a digital wallet on their smartphone, which could be the University's official app or a general-purpose wallet app. The issuing University would be responsible for establishing a process where only the subject of that credential can receive a digital credential to a device owned by that individual.

To consume a digital degree certificate, any business Verifier can cryptographically verify the certificate presented by the End-User and obtain information such as the name of the university that issued the degree; the graduate's name at the time of graduation; and subject, classification, and date of award of the degree.

The cost the verifier has to pay, which might even be free in some cases, to obtain a public key or a certificate to verify the credential's cryptography is much cheaper than the price paid for the above-mentioned physical/semi-online verification services. This makes digital credentials a very cost effective alternative or a supplement. Furthermore, verification is almost instantaneous and saves the verifier a large amount of time and effort.

It is worth noting that verifiable credentials might enable novel business models such as the verifiers paying the credential issuers via the wallet or via a marketplace.

The importance of digitizing physical credentials became apparent when COVID-19 struck. Countries saw that international travel could not safely be restarted again until passengers could securely provide proof that they had been vaccinated against COVID-19, had recently tested negative to the virus, or had recently recovered from the virus. Paper certificates were initially introduced, but countries had difficulty in knowing who the correct credential issuers of these certificates were, and which certificates were valid and which were not. Not only that, but forged paper certificates soon became available for purchase on the black market. The business need for electronic certificates was obvious.

Use-Cases

This section showcases the value of verifiable credentials through exploring use-cases where credentials of various formats are issued and presented using the OpenID4VC specification family.

It is important to differentiate between enterprise (company to employee), government and consumer (company to customer) use-cases since the value proposition is different.

The credential formats supported by OpenID for Verifiable Credentials family include W3C Verifiable Credentials Data Model[15], ISO/IEC 18013-5 mobile Driving Licence (mDL)[16], ISO/IEC 23220-2 electronic Identification (eID)[17], Anonymous Credentials[18], and FHIR (Fast Healthcare Interoperability Resources) data model[19] used in SMART Health Cards Framework[20].

VC Data Model: Employee Onboarding (Enterprise Use-Case)

As has been mentioned, verifiable credentials become very advantageous in the use-cases where the End-User is asked to present multiple credentials issued by different credential issuers in a single presentation, without involving the credential issuers.

One such use-case is onboarding employees.

Queensland Government, ConnectID by eftpos and Meeco conducted a production Pilot to improve the efficiency in onboarding new workforce employees. The goal was to automate and orchestrate business flows where employees present their identity, qualifications, or certifications credentials to prove their skill set and experiences.

Usually, these credentials are provided through a manual process. The employees are asked to submit licences, certifications, etc. by attaching scanned or photographed copies of paper documentation to an email.

By building a verifiable credentials solution leveraging the SIOP v2 specification family and W3C Verifiable Credentials, the PoC demonstrated significant efficiency gains[21]:

- Time to provide the required credentials **reduced from 48 hours to 30 minutes**.
- Time spent on identity and credentials validation **reduced from 3 days to 30 minutes**.

OpenID4VC was a technology driver in this PoC. Using OIDC4VP, existing Verifiers who already have an OpenID Connect infrastructure, became verifiers of verifiable credentials with minimal impact to their existing infrastructure. A process to ensure the validity of the credentials

[15] https://www.w3.org/TR/vc-data-model/

[16] https://www.iso.org/standard/69084.html

[17] https://www.iso.org/standard/79124.html

[18] https://www.hyperledger.org/use/hyperledger-indy

[19] https://ecqi.healthit.gov/fhir

[20] https://spec.smarthealth.cards/

[21] https://www.meeco.me/resources/case-study-digital-identity-verifiable-credentials

was also established, since these credentials require renewal on a perpetual basis. Employees need to be up to date with evolving practices and knowledge around a certain skill set.

It was also very straightforward to create an orchestration process guiding employees to retrieve the required credentials from different identity providers and to present them in a single presentation to an employer.

W3C Verifiable Credentials Data Model defines a flexible, general data model that can be used to express any type of credentials, including workplace credentials.

VC Data Model: Entitlement Management with Workplace Credentials (Enterprise Use-Case)

The employee onboarding use-case can be generalized into an entitlement management use-case with workplace credentials. The credential is issued by a workplace organization and the End-User could be an employee, student, staff member, contractor, or a vendor. In addition to onboarding, it supports the following End-User journeys:

- Access to workplace applications – e.g., Verified Employee accessing their work email
- Access to workplace applications by partners – e.g., Verified Employee at Woodgrove collaborating at Fabrikam
- Access to applications on the Internet – e.g., Verified Employee at Woodgrove, unlocking a travel discount with an airline
- Access to physical buildings - e.g., Verified Employee accesses company premises anywhere in the country.

One basic scenario for entitlement management using workplace credentials that Microsoft has been focusing on looks like the following:

1. Woodgrove contracted Fabricam for a marketing campaign. Woodgrove has a policy set up that only employee of Woodgrove or contracted companies can access its resources. Woodgrove accepts entitlement credentials via a digital wallet.
2. Fabricam sends out an email to its employees that offers to issue an employee credential to a wallet pre-selected by the employer.
3. When Alice, an employee of Fabricam tries to access the resources of Woodgrove, she is prompted to present a credential proving her employer. Alice scans a QR code displayed on Fabricam's website, and when prompted, confirms presentation of the credential to Woodgrove. Woodgrove verifies the received credential and allows Alice to access the resources accessible to contractors.

ISO/IEC 18013-5 Data Model: Mobile Driving Licence (Government Use-Case)

Verifiable credentials are also useful for government-issued credentials. One notable use-case is mobile Driving Licences.

A physical driving licence is a data set of driving permits which is subject to each jurisdiction and is effective only for domestic use. Historically, a main venue for the work to design domestic driving licences which can then be used as international driving licences has been the International Organization for Standardization (ISO).

Recently, various regions including North and South America, Europe, Asia-Pacific have been considering introducing a mobile driving licence, well known as mDL, with many Proof of Concepts and implementations at-scale. An mDL is a digital representation of driving permits which includes identity information of the driver and his/her driving privileges.

Legal jurisdictions can use driving licence information to verify the identity of the holder as well as his/her driving privileges. For a successful verification, the following conditions must be met:

- Proof of possession (PoP); the driving licence information is bound to a unique cryptographic key so that only the holder of the licence can use it,
- User consent: the driving licence information should be presented only after the holder has explicitly requested this.

It should be noted that ID documents can be used for both “intended use” and “secondary use”. The scope of “intended use” of a driving licence is defined by the issuing authority, in many cases, to verify driving privileges (by a legal jurisdiction). The use of a mobile driving licence for personal identification is a secondary use and an issuing authority is not responsible for such use and the licence holder is considered to be using it at his/her own risk. OpenID4VC is suitable for issuing and presenting mDLs expressed in a data model defined by ISO/IEC 18013-5, in particular for the “secondary use”

Self-Issued OP is referred to in the following two ISO technical specifications:

- ISO/IEC TS 23220-4 Cards and security devices for personal identification — Building blocks for identity management via mobile devices — Part 4: Protocols and services for operational phase
- ISO/IEC TS 18013-7 Personal identification – ISO compliant driving licence – Part 7: Mobile driving licence (mDL) add-on functions

Note that the mDL use-case can be expanded to government-issued ID Documents such as Citizen ID, Birth Certificate, Marriage Certificate, etc.

FHIR Data Model: SMART Health Cards (Consumer Use-Case)

Another use-case is Healthcare data. One of the widely used data models in the Healthcare industry is Fast Healthcare Interoperability Resources (FHIR).

SMART Health Cards are paper or digital versions of the End-User’s clinical information, mainly known for expressing vaccination credentials. It is a great example of the value of standards that leveraged both the FHIR data model to express "clinical semantics" (what vaccine was given, where, by whom, what were the demographics of the recipient) and the W3C Verifiable Credentials Data Model to express "assertion semantics" (who said what, when did they say it, how do you know).

SMART Health Cards is also an example of a credential that is not cryptographically bound to the identifier of the End-User possessing it, in contrast to the previous examples of workplace VCs and mDLs. Instead, SMART Health Cards include End-User identity claims that enable claim-based binding. The verifier must verify the binding of the credential by requesting the presentation of existing forms of physical or digital identification that include the same identity claims (e.g., name, address, date of birth, from a driver's licence or other ID card, either in person or via an online ID verification service) alongside the SMART Health Card.

SMART Health Cards do not necessitate usage of the presentation specifications of OpenID4VC, but they can greatly benefit from the OpenID for Credential issuance specification, since the FHIR community already uses OAuth 2.0 to issue credentials.

Technical 101 of OpenID Connect and OpenID4VC

Demystifying OpenID Connect

In contrast to some existing myths, OpenID Connect is not only built for centralized big Identity Providers (IdPs). It supports a wide range of architectures, not just centralized IdPs, but also large-scale networks of IdPs (e.g. universities, mobile operators, or financial institutions), IdPs run by End-Users on their own Web sites, and IdPs running on End-Users’ mobile devices. The latter model has dedicated support in the OpenID Connect Core specification in the form of the Self-Issued OpenID Provider (Self-Issued OP).

OpenID Connect was built with User-Centricity in mind from the ground up. Identity data can be requested and provided at the granularity of the individual End-User claims thus applying the data minimization principle. The protocol flow is also designed to provide the capabilities for the IdP to talk directly to the End-User and obtain End-User consent before releasing End-User claims to the Verifier. The IdP presents to the End-User the requested data and the ultimate data recipient, which allows the End-User to make an informed decision and to approve or refuse the sharing of the requested claims.

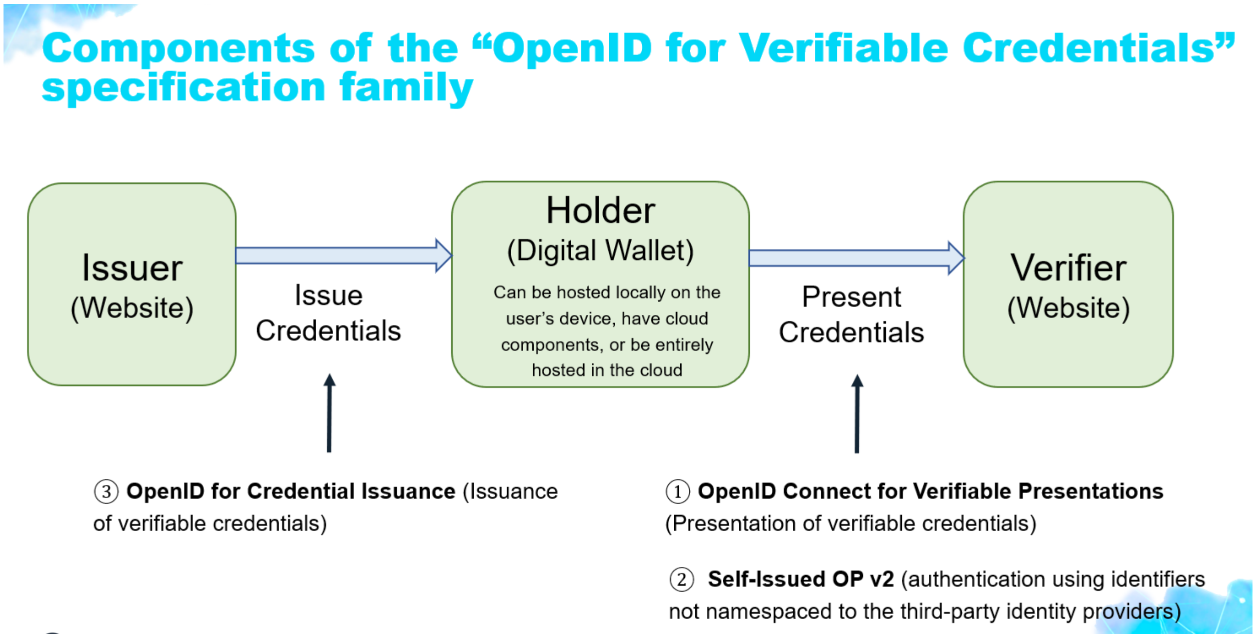
This direct End-User interaction capability also allows OpenID Connect implementations to utilize modern origin-based authentication mechanisms like WebAuthn.

Also, OpenID Connect is well-known for its proven security (see [1], [2], and [3]).

Extending OpenID for Verifiable Credentials Applications

OpenID for Verifiable Credentials leverages the above-mentioned characteristics and capabilities of OpenID Connect to be used with verifiable credentials applications.

The next figure shows the actors in a verifiable credentials ecosystem along with their interfaces and the respective sub-protocols of OpenID for Verifiable Credentials.



Components of the "OpenID for Verifiable Credentials" specification family

Issuer (Website)

Issue Credentials

Holder (Digital Wallet)

Can be hosted locally on the user's device, have cloud components, or be entirely hosted in the cloud

Present Credentials

Verifier (Website)

③ **OpenID for Credential Issuance** (Issuance of verifiable credentials)

① **OpenID Connect for Verifiable Presentations** (Presentation of verifiable credentials)

② **Self-Issued OP v2** (authentication using identifiers not namespaced to the third-party identity providers)

- SIOP v2 (SIOP v2): the protocol to exchange cryptographically verifiable identifiers and authenticate using the key material controlled by the End-User.
- OpenID Connect for Verifiable Presentations: an extension of OpenID Connect allowing verifiers to request and receive verifiable presentations.
- OpenID Connect for Credentials Issuance: An API and corresponding authorization flows to request issuance of verifiable credentials.

It is important to note that OpenID4VC is not tied to JWTs or JWS as the only supported credential formats. Examples of other credential formats successfully implemented with OpenID4VC include Linked Data Proofs and Anonymous Credentials[22] (AnonCreds). Moreover, OpenID4VC is designed to work with identifier types (e.g. DID methods), cryptographic schemes, and revocation schemes of the implementer's choice.

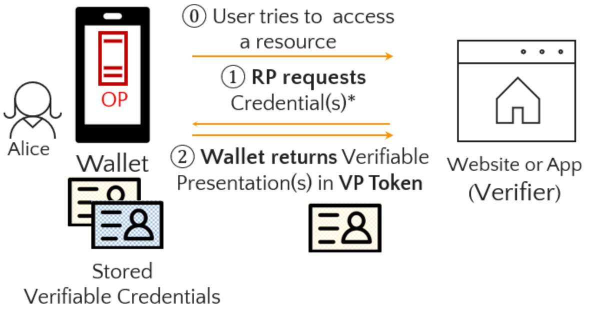
[22] https://openid.net/specs/openid-connect-4-verifiable-presentations-1_0.html#Hyperledger.Indy

OIDC4VP 101

OpenID Connect for Verifiable Presentations (OIDC4VP) extends OpenID Connect with the ability to request and present verifiable credentials. It therefore introduces the new “VP Token” to convey verifiable presentations and integrates the DIF Presentation Exchange[23] into the “claims” request parameter to specify the RP’s requirements regarding the credentials to be presented as well as to help the verifier process the result.

This is illustrated by the following figure.

OpenID for Verifiable Presentations



⓪ User tries to access a resource
① RP requests Credential(s)*
② Wallet returns Verifiable Presentation(s) in VP Token
OP
Alice
Wallet
Stored Verifiable Credentials
Website or App
(Verifier)

- Query language to granularly specify what kind of credential Verifier wants. (utilizes DIF Presentation Exchange 2.0)
- Verifiable Presentations* are returned in a newly defined VP Token
- Simple overall architecture, e.g. device local communication when same device flow is used

In this flow, the RP requests presentation of a verifiable credential on top of a Self-Issued OP request. The Self-Issued OP (also being a wallet in this case), interacts with the holder to select the credential to be presented and creates a verifiable presentation that is sent back to the RP in the VP token, along with the ID Token.

The RP verifies the holder binding and the integrity and authenticity of the credential before it is processed. The concrete steps to be taken depend on the credential format, crypto scheme, and revocation mechanism, which are out of scope of OIDC4VPs.

Note that version 2 of the Presentation Exchange (PEv2) specification that is used as the query language in OIDC4VP is under active development and the example in the OIDC4VP specification might not always reflect the latest changes in the PEv2 specification.

[23] https://identity.foundation/presentation-exchange/

For an extensive example please reference the Appendix of the specification.

SIOP v2 101

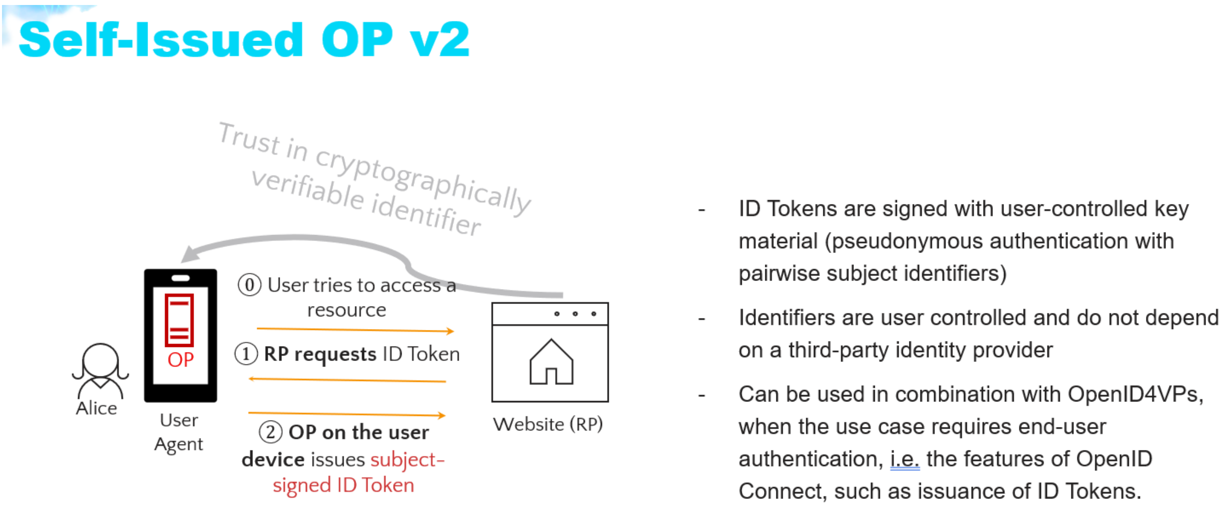
The Self-Issued OpenID Provider (Self-Issued OP) was already part of the OpenID Core specification (this version is designated as SIOP v1). It enabled End-Users to be in control of the identity information and signing keys. Using the Self-Issued OP, an End-User could authenticate using a self-signed ID Token that was signed using the key material controlled by the End-User.

The emerging SIOP v2 aims at adjusting SIOP v1 to the challenges of modern verifiable credentials applications. It introduces the following capabilities:

- Support for DIDs in addition to the raw JSON Web Keys as End-User identifiers
- Support for Dynamic Self-Issued OP discovery
- Support for invoking Self-Issued OP via HTTPS URLs in addition to the custom schemes such as “openid://”. This enables the use of deep/app/universal links on modern smartphone operating systems and web wallets.
- Support for all OpenID Connect Flows, e.g. authorization code flow, which allows cloud/web wallets to leverage the advanced security features and capabilities in comparison to the traditional “OIDC implicit” flow utilized by SIOP v1.
- Support for “cross device” flows, where the End-User can start the presentation on a different device than where the credentials will be accessed from, in addition to the “same device” flows
- Support for OpenID Connect Registration metadata[24] for the management of wallets. This enables interactions among pre-registered and verified RPs and Self-Issued OPs, which is an important enabler for regulated verifiable credentials schemes (e.g. eIDAS 2), in addition to ad-hoc interactions.

The following figures shows the basic message flow

[24] https://openid.net/specs/openid-connect-registration-1_0.html



Self-Issued OP v2

Trust in cryptographically verifiable identifier
⓪ User tries to access a resource
① RP requests ID Token
② OP on the user device issues subject-signed ID Token
OP
Alice
User Agent
Website (RP)

- ID Tokens are signed with user-controlled key material (pseudonymous authentication with pairwise subject identifiers)
- Identifiers are user controlled and do not depend on a third-party identity provider
- Can be used in combination with OpenID4VPs, when the use case requires end-user authentication, i.e. the features of OpenID Connect, such as issuance of ID Tokens.

0) The End-User tries to login to a site (or app).

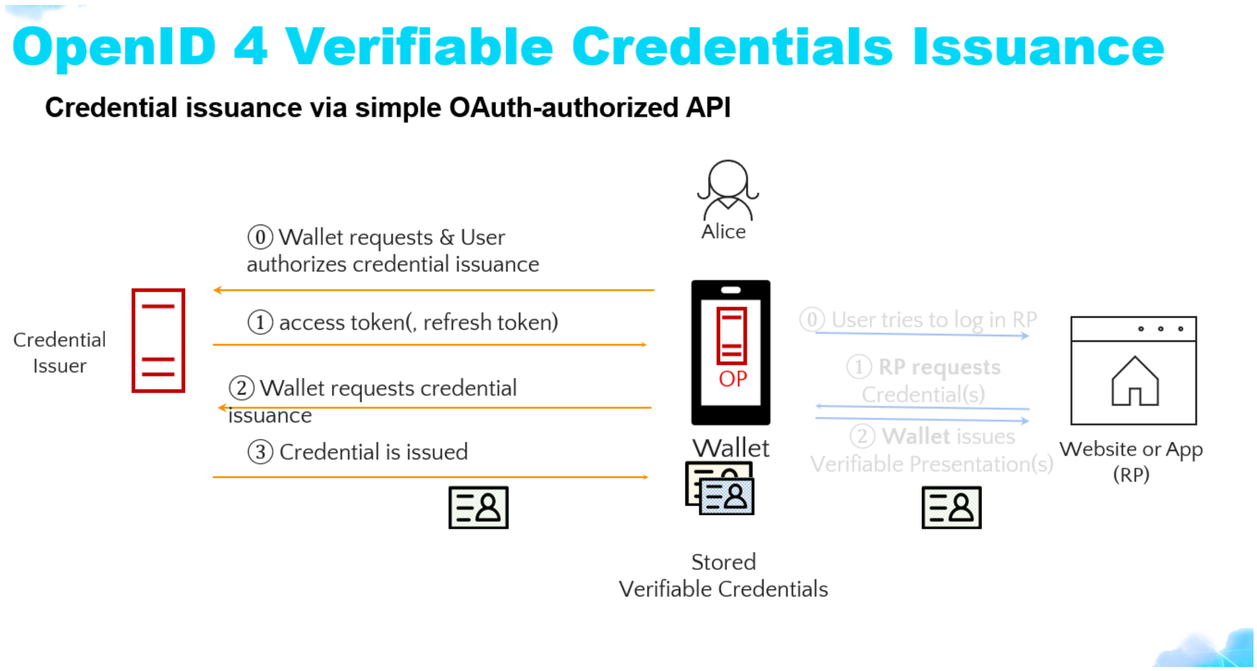
1) The site sends an authentication request to the End-User's Self-Issued OP.

2) The Self-Issued OP (on behalf of the End-User) issues an ID Token that is signed with a key under the End-User's control. Depending on the Self-Issued OP's architecture, such a key resides on the mobile device or is stored in a cloud (custodial wallet). The Self-Issued OP will typically use a certain key for a particular Verifier in order to prevent collation across RPs. If the Self-Issued OP wishes to prevent correlation of requests to the same RP then it may use an ephemeral key.

The Self-Issued OP enables a wide variety of wallet architectures. For example, wallets can be run on the user's device but they can also be hosted in the cloud. The user experience can be provided through a mobile app or a web application. From a protocol perspective, they can utilise all OpenID Connect flows, i.e. there is plenty of choice to meet the needs of the respective deployment and use-case(s).

OpenID4CI 101

OpenID Connect for Credential Issuance (OpenID4CI) allows the issuance of verifiable credentials. Issuance is performed using the Credential Issuance API. In order to be able to retrieve a credential from this API, the client (typically the wallet), needs an OAuth access token, which is obtained in the course of an authorization process. The specification currently describes two types of authorization processes covering the needs of different scenarios.



OpenID 4 Verifiable Credentials Issuance

Credential issuance via simple OAuth-authorized API

Alice
Credential Issuer
⓪ Wallet requests & User authorizes credential issuance
① access token(, refresh token)
② Wallet requests credential issuance
③ Credential is issued
OP
Wallet
Stored Verifiable Credentials
⓪ User tries to log in RP
① RP requests Credential(s)
② Wallet issues Verifiable Presentation(s)
Website or App (RP)

Authorization Code Flow

In this flow, the wallet starts the process by sending an OIDC/OAuth authorization request to the Credential Issuer, i.e. the End-User is sent to the Credential Issuer's authorization endpoint. During the authorization process, the credential issuer authenticates and/or identifies the End-User and may obtain consent to issue one or more credentials. After successful completion of the authorization request, the wallet is issued an authorization code that it redeems at the token endpoint for an access token.

The credential issuer may also decide to issue a refresh token, e.g. to allow on-demand issuance or credential refreshes without further End-User interactions.

As a security countermeasure against replay, all tokens may be sender constrained, i.e. they are bound to key material whose possession the wallet must prove when using the respective token.

This flow is especially suited if the End-User journey starts in the wallet, e.g. the End-User just installed the wallet app and wants to install the first set of credentials or if the End-User journey starts with a verifier and the End-User's wallet lacks the required credentials. In both cases, the wallet can offer the End-User a selection of suitable credential issuers and can directly enter a secure and convenient issuance process.

Pre-Authorized Code Flow

There are scenarios where the journey starts with a credential issuer's website or app and the wallet is involved at a later step. For example, the already authenticated End-User conducts an examination and, after successful completion, is offered to download a digital certificate. In such a case, the data that will go into the credential is more or less complete, the wallet "just" needs to pick the credential up.

The pre-authorized code flow is optimized for such scenarios.

It starts by the credential issuer sending a request to the End-User's wallet, which tells the wallet what kind of credential to request from which credential issuer. The request comes along with a pre-authorized code and the issuer's preference whether an additional security protection is needed, requiring the End-User to enter a PIN when using the pre-authorized code. Instead of sending a request, the credential issuer may also render a QR code with the same data, which allows the End-User to utilise a wallet residing on a different device.

In this flow, the wallet does not need to send the End-User to the credential issuer's authorization endpoint. Instead, it can directly redeem the pre-authorized code for an access token at the credential issuer's token endpoint. This streamlines the End-User experience but it also poses a security risk since the code cannot be bound to a certain device. In order to mitigate this risk, the credential issuer may establish an End-User PIN with the End-User during a user authentication flow that results in the issuance of a pre-authorization code and require this PIN to be present when pre-authorized code is being used at the token endpoint. How this PIN is communicated to the End-User is up to the implementation as long as it is sent using a channel different from the one used to send the pre-authorized code.

The token response is the same as for the authorization code flow.

Credential Endpoint

The access token obtained using one of the flows described above is then used to request credential issuance at the credential endpoint. The wallet must send a suitable proof of possession of the key material the credential shall be bound to along with the request.

Key Features

In summary, the following are the key features of OpenID4VC Family:

- Simplicity
- Developer familiarity and friendliness
- Leverages deployed OpenID Connect infrastructures (facilitates verifiable credentials adoption)

- Security
- Flexibility regarding identifier (e.g. DID methods), credential formats, cryptographic schemes, and revocation schemes

Conclusion

Standards that the verifiable credentials ecosystem consists of vary in their level of maturity and are being developed across different SDOs (Standards Development Organisations). The key to a globally interoperable verifiable credentials ecosystem is making choices for each component among existing and emerging options that enables a certain use-case. This is why interoperability profiles that have made these choices for certain use-cases have been emerging. Therefore, one should be aware that there are a number of ways to implement a verifiable credentials system, depending on the needs and business requirements of the use-case.

One of the notable strengths of using the OpenID4VC specification family as a credential transport protocol is that it allows implementers to make their own choices for other components of the verifiable credentials technical stack: entity identifier types (including DID methods), credential formats, revocation schemes, crypto suites, trust mechanisms, etc.

This is a very powerful extensibility point because implementers can add or remove support for a new credential format or a new identifier type while still being able to issue and present credentials over the same protocol.

Below we give an overview of choices for each of the components that can be combined with OpenID4VC issuance and response protocols.

Credential Data Model. One is the W3C Verifiable Credentials Data Model which is a generic multipurpose data model. Another is the ISO/IEC 18013-5 standard that defines an entire technical stack for mobile driving licences which includes a data model which could be exchanged over an OpenID4VC protocol. The ISO/IEC 23220-2 standard is a data model for mobile ID documents not limited to driving licences. The SMART Health Cards Framework uses FHIR data model to express "clinical semantics" (what vaccine was given, where, by whom, what were the demographics of the recipient) and the W3C Verifiable Credentials Data Model to express "assertion semantics" (who said what, when did they say it, how do you know). The first three data models enable the Wallet to cryptographically prove legitimate possession of a credential that it is presenting by proving that the device that obtained the credential is the same device that is presenting it. This is not currently utilised in the SMART Health Card framework. Instead, it relies on the Verifier matching the credentials of the End-User, which are obtained from another identity document, with those in the FHR credential.

Credential Schemas. While ISO/IEC 18013-5 and the SMART Health Cards Framework define namespaces and concrete claim names, other data model standards such as W3C Verifiable Credentials Data Model do not. Implementers would need to define namespaces and claim names for their use-cases if none of the existing ones apply. For example, use-cases utilizing

W3C Verifiable Credentials Data Model need to define schemas that extend the core VC schema and the corresponding vocabularies[25].

Trust Frameworks. OpenID4VC provides mechanisms that can be used to establish trust between Clients and Credential Issuers and between RPs and the Wallets, but it does not provide concrete criteria for the level trust that needs to be established. Trust Frameworks are required to fulfil this function. Various groups have been developing Trust Frameworks that support individual use-cases with varying requirements of the trust levels.

Cryptographic Suites. A large variety of digital signatures algorithms with different curves are being used, ranging from ECDSA and EdDSA to more advanced Camenisch-Lysyanskaya signatures and BBS signatures. There are also different options on how to express proofs of integrity: IETF JWS/JWT, W3C CCG Data Integrity (that used to be called Linked Data Proofs), and AnonCreds used in Hyperledger Indy SDK. Note that not all of the algorithm options are vetted or approved by relevant National institutions such as NIST, BSI, ENISA and others.

Credentials and signatures formats. The protocol supports (but is not limited to) verifiable credentials expressed as W3C Verifiable Credentials Data Model in JWT and JSON-LD format and as ISO/IEC 18013-5 mDL data model in JSON and CBOR encoding. It supports external signatures (compact serialized JWS) and embedded or enveloped signatures.

Entity Identifiers. Three entities - Credential Issuers, End-Users and Verifiers - need identifiers in the verifiable credentials ecosystem. Usually, an identifier is also used to obtain a cryptographic public key used to verify the signature on the credential. Options include W3C Decentralized Identifiers (DIDs) v1.0, HTTPS URLs, JWKs, X.509 certificates, etc.

Note that there is convergence around the utilized query language. When the verifier requests presentation of the digital credentials, there is a need to be able to specify aspects of the requested credential. OIDC4VP uses the Presentation Exchange (PE) v2.0 specification[26] defined by the Decentralized Identity Foundation.

For those interested in learning more, please refer to the Appendix for more technical details on the specifications, or to openid.net/wg/connect/status for access to the full documentation at no cost. We also warmly invite the readers to comment on this white paper and the standards via participation in the AB/Connect Working Group’s Special Calls dedicated to OpenID for Verifiable Credentials specifications family. Details on the Working Group meetings and requirements to participate are also available at openid.net/wg/connect.

For those looking for the opportunities to do interoperability testing of your implementations, consider joining the Global Assured Identity Network Proof of Concept Community Group, hosted by the OpenID Foundation, which seeks to deliver on the vision of enabling “networks of

[25] One example of such use-case is https://ec.europa.eu/digital-building-blocks/code/projects/EBSI/repos/json-schema/browse/schemas

[26] https://identity.foundation/presentation-exchange/

networks” to help people assert their identity (openid.net/gainpoc). Some participants are interoperability testing their implementations using the OpenID for Verifiable Credentials specifications family.

In closing, achieving large scale adoption of verifiable credentials will be by Evolution, not by Revolution. We hope that this whitepaper has illustrated how the work on the OpenID4VC specification family fits into the overall picture of verifiable credentials, when it should be considered as part of your verifiable credentials implementation and how it can facilitate adoption of verifiable credentials.

References

[1] Daniel Fett, Ralf Küsters, und Guido Schmitz, „The Web SSO Standard OpenID Connect: In-Depth Formal Security Analysis and Security Guidelines“, in IEEE 30th Computer Security Foundations Symposium (CSF 2017), 2017, S. 189--202. https://publ.sec.uni-stuttgart.de/fettkuestersschmitz-csf-2017.pdf

[2] Daniel Fett, Ralf Küsters, und Guido Schmitz, „A Comprehensive Formal Security Analysis of OAuth 2.0“, in Proceedings of the 23rd ACM SIGSAC Conference on Computer and Communications Security (CCS 2016), 2016, S. 1204--1215. https://publ.sec.uni-stuttgart.de/fettkuestersschmitz-ccs-2016.pdf

[3] Daniel Fett, Pedram Hosseyni, und Ralf Küsters, „An Extensive Formal Security Analysis of the OpenID Financial-grade API“, in 2019 IEEE Symposium on Security and Privacy (S&P 2019), 2019, Bd. 1, S. 1054–1072. https://publ.sec.uni-stuttgart.de/fetthosseynikuesters-fapi-sp-2019.pdf (edited)

[4] Nat Sakimura, John Bradley, Michael B. Jones, Breno de Medeiros, and Chuck Mortimore, OpenID Connect Core 1.0, November 2014.

Appendix

Examples OpenID Connect 4 Verifiable Presentations

ISO/IEC 18013-5 mDL

Below is a non-normative example of how claims parameter is used in the authorization request to request an mDL credential in ISO/IEC 18013-5:2021 format:

```
"claims": {
 "vp_token": {
  "presentation_definition": {
   "id": "mDL-sample-req",
   "input_descriptors": [
    {
     "id": "mDL",
     "format": {
      "mdl_iso_cbor": {
       "alg": ["EdDSA", "ES256"]
      },
     "constraints": {
      "limit_disclosure": "required",
      "fields": [
       {
        "path": ["$.mdoc.doctype"],
        "filter": {
         "type": "string",
         "const": "org.iso.18013.5.1.mDL"
        }
       },
       {
        "path": ["$.mdoc.namespace"],
        "filter": {
         "type": "string",
         "const": "org.iso.18013.5.1"
        }
       },
       {
        "path": ["$.mdoc.family_name"],
        "intent_to_retain": "false"
       },
       {
        "path": ["$.mdoc.portrait"],
        "intent_to_retain": "false"
       },
       {
        "path": ["$.mdoc.driving_privileges"],
        "intent_to_retain": "false"
        }
       ]
      }
     }
    }
   ]
  }
 }
}
```

For detailed explanation please see the Annex of OIDC4VP specification.

The response contains an ID Token and a VP Token. In the following example, a single ISO/IEC 18013-5:2021 mDL is returned as a VP Token. Note that a ISO/IEC 18013-5:2021 mDL could be encoded both in CBOR or JSON.

The following is a non-normative example of a successful authorization request when [SIOPv2] and this specification is used.

```
POST /callback HTTP/1.1
Host: client.example.org
Content-Type: application/x-www-form-urlencoded

id_token=<<Base64URL encoded ID Token>>
 &vp_token=<<Base64URL encoded ISO/IEC 18013-5:2021 mDL (CBOR in this example since the requested format was `mdl_iso`)>>
```

The following is an ID Token example. It shows how the `presentation_submission` helps the RP to locate in the VP Token an ISO/IEC 18013-5:2021 mDL expressed in CBOR:

```
{
 "aud": "https://client.example.org/callback",
 "sub": "9wgU5CR6PdgGmvBfgz_CqAtBxJ33ckMEwvij-gC6Bcw",
 "iss": "9wgU5CR6PdgGmvBfgz_CqAtBxJ33ckMEwvij-gC6Bcw",
 "sub_jwk": {
  "x": "cQ5fu5VmG...dA_5lTMGcoyQE78RrqQ6",
  "kty": "EC",
  "y": "XHpi27YMA...rnF_-f_ASULPTmUmTS",
  "crv": "P-384"
 },
 "exp": 1638483944,
 "iat": 1638483344,
 "nonce": "67473895393019470130",
 "_vp_token": {
  "presentation_submission": {
   "descriptor_map": [
    {
     "id": "mDL",
     "path": "$",
     "format": "mdl_iso"
    }
   ],
   "definition_id": "mDL-sample-req",
   "id": "mDL-sample-res"
  }
 }
}
```

A non-normative example of an ISO/IEC 18013-5:2021 mDL encoded as CBOR in diagnostic notation can be found in the Annex of OIDC4VP specification.

AnonCreds

Here is a simple presentation request, which asks the holder for a credential of type “EuropeanBankIdentity”:

```
{
   "response_type":"id_token",
   "client_id":"https://example.com/callback",
   "scope":"openid",
   "redirect_uri":"https://example.com/callback",
   "nonce":"67473895393019470130",
   ...
   "claims":{
      "vp_token":{
         "presentation_definition":{
            "id":"1",
            "input_descriptors":[
               {
                       "id": "1",
                       "constraints": {
                           "fields": [
                               {
                                   "path": [
                                       "$.credentialSchema.id"
                                   ],
                                   "filter": {
                                       "type": "string",
                                       "pattern":
                          "https://example.com/…/EuropeanBankIdentity.json"
                                   }
                               }
                           ]
                       }
                   }

            ]
         }
      }
   }
}
```

Since OpenID Connect embraces the principle of data minimization, verifiers can use “limit_disclosure” property to ask only for a subset of claims in a certain credential, as shown in the following:

```
{
  "response_type":"id_token",
  "client_id":"https://example.com/callback",
  "scope":"openid",
  "redirect_uri":"https://example.com/callback",
  "nonce":"67473895393019470130",
  ...
  "claims":{
     "vp_token":{
        "presentation_definition":{
           "id":"NextcloudLogin",
           "input_descriptors":[
              {
                 "id":"ref2",
                 "name":"NextcloudCredential",
                 "format": {
                   "ac_vc": {
                       "proof_type": ["CLSignature2019"]
                   }
                 },
                 "constraints":{
                    "limit_disclosure":"required",
                    "fields":[{
                               "path": [
                                   "$.schema_id"
                               ],
                               "filter": {
                                   "type": "string",
                                   "pattern":
                              "did:indy:idu:test:3QowxFtwciWceMFr7WbwnM:2:BasicScheme:0\\.1"
                               }
                              },
                              {"path":["$.values.email"]},
                              {"path":["$.values.first_name"]},
                              {"path":["$.values.last_name"]}]
                 }
              }
           ]
        }
     }
  }
}
```

Note that the “constraints” element along with “limit_disclosure” is used to request disclosure of the schema_id, email, first name and last name properties only. (Without “limit_disclosure” any credential that matched the “constraints” would have all its properties returned.)

This example also shows that the verifier can request credentials in formats beyond the scope of W3C verifiable credentials. In this example, the verifier asks for a so-called Anoncred (part of the Hyperledger Indy framework) by using the format of “ac_vc”.

The response from the Self-Issued OP consists of two artefacts, the ID Token and the VP token. Here are examples of those artefacts.

The ID Token contains a “presentation_submission”, which contains metadata about the place where the requested credential can be found in the result. In this case, the credential is contained in a verifiable presentation directly in the root of the VP token.

```
{
 "iss": "https://self-issued.me/v2",
 "aud": "https://example.com/callback",
 "sub": "did:key:z6MkqUDiu3MHxAm...mscLT8E9R5CKdbtr7gwR8",
 "exp": 1645469476,
 "iat": 1645465876,
 "nonce": "cdb97870-a3be-49b4-aa55-8c7c7122178a",
 "_vp_token": {
   "presentation_submission": {
     "descriptor_map": [
       {
         "path": "$",
         "format": "ldp_vp",
         "path_nested": {
           "path": "$.verifiableCredential[0]",
           "format": "ldp_vc"
       }
     ],
     "definition_id": "1",
     "id": "1"
   }
 }
}
```

And below is the corresponding VP Token:

```
{
  "@context":[
     "https://www.w3.org/2018/credentials/v1"
  ],
  "holder":"did:key:z6MkqUDiu3MHxAmuMQ8jjkLiUu1mscLT8E9R5CKdbtr7gwR8",
  "id":"urn:uuid:04816f2a-85f1-45d7-a66d-51764d39a569",
  "proof":{
     "domain":"https://example.com/callback",
     "jws":"...",
     "nonce":"cdb97870-a3be-49b4-aa55-8c7c7122178a",
     "proofPurpose":"authentication",
     "type":"Ed25519Signature2018",
     "verificationMethod":"did:key:z6MkqUDiu3..."
  },
  "type":[
     "VerifiablePresentation"
  ],
  "verifiableCredential":[
     {
        …
        "type":[
           "VerifiableCredential",
           "EuropeanBankIdentity"
        ],
        "credentialSubject":{
           "id":"did:key:z6MkqUDiu3MHxAmuMQ8jjkLiUu1mscLT8E9R5CKdbtr7gwR8",
           "familyName":"Family001",
           "givenName":"Given001",
           "birthDate":"1950-01-01",
           "placeOfBirth":{
              "country":"DE",
              "locality":"Berlin"
           }
        },
        "id":"identity#EuropeanBankIdentity#33665527-50d6-484e-a93a-283ecb8d660b",
        "credential issuer":"did:key:z6MkgF2pvVNEFXCksupWKrdPhL6ubecis3AWbWVsr9bNAbwC",
        "proof":{
           "created":"2022-02-18T19:08:27Z",
           "creator":"did:key:z6MkgF2pvVNEFXCksupWKrdPhL6ubecis3AWbWVsr9bNAbwC",
           "domain":"https://api.preprod.ebsi.eu",
           "jws":"eyJiNjQiOmZhbHNlLCJjcml0IjpbImI2NCJdLCJhbGciOiJFZERTQSJ9..jH-
6rgdVLsvbEE_g2RIDl_AQou4s3DwGg4VE06K3ngvSzm1SKppDvA3UuEfb0dfMrZ_ShTKThM-gxJaUSarwAA",
           "nonce":"7ca07921-26da-4a65-9a2e-781599cc1894",
           "type":"Ed25519Signature2018"
        },
        "validFrom":"2021-08-31T00:00:00Z",
        "issuanceDate":"2021-08-31T00:00:00Z",
             }
  ]
}
```

It directly contains a VP in LD Proof format.

Note that Version 1.0 of PE is published, with version 2.0 being under development (as of April 2022). Version 2.0 significantly simplifies Version 1.0, by introducing a core subset that is mandatory to implement, coupled with a set of optional extra query constructs.