

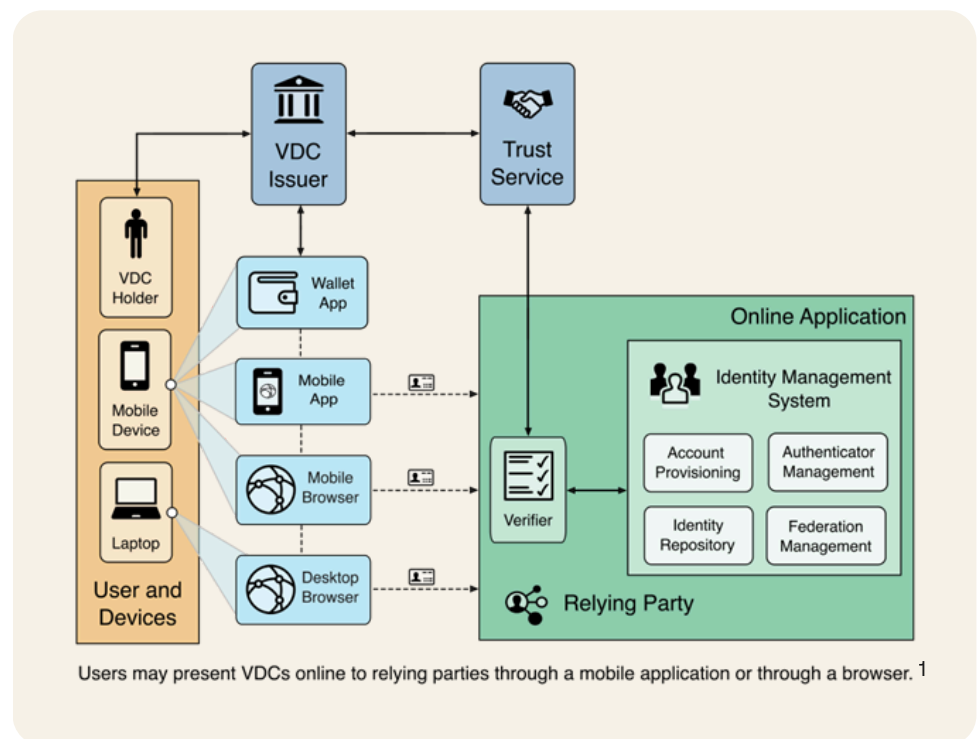
Verifiable digital credentials: A technical overview

Learn the technical advantages,
supported protocols, and why
to trust VDCs



The VDC ecosystem

Verifiable digital credentials (VDCs) operate within a defined ecosystem of participants who work together to ensure secure, portable, and privacy-preserving sharing of identity data. Holders have control over their information and what they share with whom. When a holder shares information with a relying party, the verifier can perform real-time verification and check for authenticity without needing to communicate with the issuer. The VDC ecosystem comprises issuers, wallets, verifiers, and a verifiable data registry



Issuer

An entity trusted to assert claims about a subject, such as age, state of residence, employment status, or skill qualifications. The issuer issues and cryptographically signs credentials with its private key.

Credential Manager (Wallet)

A secure digital wallet controlled by the individual (the **holder**), storing credentials and responding to verifier requests. Wallets and specific credential formats may support selective disclosure, enabling holders to share only the claims needed to perform a specific action, and not the entire document. Wallets primarily facilitate a cryptographic chain of custody between the issuer and verifier. Wallet provenance can be shared with the issuer on credential issuance and with the verifier when a credential is ultimately presented.

[1] Credit: NIST diagram
<https://www.nist.gov/blogs/cybersecurity-insights/digital-identities-getting-know-verifiable-digital-credential-ecosystem>

Cont'd

Verifier

An entity that acts on behalf of a **relying party** — the organization or service, e.g. bank, employer -- that wants to accept a credential and requests proof of specific claims from a holder. It cryptographically validates the credential's signatures to confirm authenticity and integrity. Verifiers are configured by the relying party to enforce policy, including which issuers are trusted and which claims are required. This allows verifiers to ensure that only credentials from approved issuers are accepted, while still preserving holder control over which claims are shared.

Verifiable data registry

A trust list or trust registry that maintains metadata about issuers, importantly their public key or reference thereto, and optionally status information regarding credential revocation.

What's a VDC technically?

Modern VDCs are built on industry standards to ensure interoperability and a cohesive security profile.

A VDC is a cryptographically signed digital document issued by a trusted authority (issuer) to a holder. It contains verified claims about the holder (e.g. age, employment status, etc.) and can be stored in a digital wallet.

These claims can represent anything ranging from a ticket ID and seat number to a full driving license or national identity. The primary driving use cases for VDCs are surrounding human-centric identity verification. What is a user's legal name? Is a user over the age of 21? The focus is on proving claims about a real person, which involves support for selective disclosure, and includes biometric indicators that can be used when stronger assurance is required. Key features of a VDC:

- **Cryptographically signed:** Ensures tamper-proof integrity and authenticity.
- **Supports selective disclosure:** Holders share only the claims required for a transaction
- **Is bound to a holder:** Verifiers can confirm that the credential is being presented by the authorized holder.
- **Adheres to standards and protocols:** Supports multiple formats (e.g. SD-JWT, mdoc) and protocols (e.g. OID4VP³ for presentation, OID4VCI⁴ for issuance).
- **Held in a wallet:** Credentials are stored client-side in the holder's wallet which can respond to verifier requests using the DCQL and DC_API². The holder controls what information is shared and when.

[2] DC_API
<https://www.w3.org/TR/digital-credentials/>

[3] OID4VCI
https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html

[4] OID4VCI [httSD-JWT VC](https://www.ietf.org/archive/id/draft-ietf-oauth-sd-jwt-vc-10.html)
<https://www.ietf.org/archive/id/draft-ietf-oauth-sd-jwt-vc-10.html>

A word on trust

A ton of work has been done to ensure VDCs are more capable and secure compared to their traditional physical document counterparts. In addition, the VDC ecosystem is designed to mitigate the type of problems that digital systems bring exposure to, like monitoring of communications individually and at scale, and collection of privacy invasive personally identifying information. VDCs use asymmetric cryptography ensuring secrets are never transmitted over the wire. And VDCs support selective disclosure and zero knowledge proofs which limit the information a user needs to disclose to pass verification checks. The technology works to better protect both online services and their users conferring privacy and security benefits to both.

Issuance

VDCs are issued by entities trusted to make claims about a subject. This includes public institutions like government agencies (driver's licenses, national IDs) and private organizations like employers (employee badges, role certifications), financial institutions (account verification), and professional bodies (qualifications, memberships). If you would trust a state DMV to make claims about a person's name and age, or an employer to make claims about an employee's role and department, then there is no functional difference when it comes to VDCs — they are issued by the same authority. The key technical difference is that the digital document is secured using the same battle-tested cryptography used to keep prying eyes off communications with your bank or email provider, ensuring the VDC cannot be forged or stolen by a hacker.

Verification

VDCs are trusted by relying parties because verifiers can cryptographically confirm their authenticity. The digital nature of the document means this is done with certainty without relying on fuzzy mechanisms like OCR or manual human reviews. If you would trust a document scan of a driver's license today, you should trust a digital presentation of a driver's license in VDC format. But you would be better off with the VDC presentation because it cannot be forged and because the user's wallet protects them against drive-by phishing attacks since it cryptographically binds credentials to the user.

Tying it together

When the holder needs to prove a claim, they present the VDC to a relying party or their verifier. The verifier:

(1) **Cryptographically validates** the credential is signed by the issuer's private key. This (2) **ensures the credential hasn't been tampered** with. The verifier (3) checks whether the credential **has been revoked or is expired** and (4) ensures the credential is being **presented by the same holder to whom it was issued** (holder binding).

The verifier can confirm the integrity and authenticity of the information all without contacting the issuer directly, a boon for user privacy.

VDCs compared to existing technology

VDCs don't aim to compete with existing authentication technology. A general rule of thumb is, A VDC should never be used for authentication. VDCs should be used to verify information about users during low volume events like registration or step-up checks, and any required verification checks should simply be flagged on the user account and re-checked as [in]frequently as required. All authentication should happen using existing protocols and standards designed for authentication, like Passkeys and OAuth. VDCs aim to replace physical analog credentialing systems like plastic cards and badges.

	Physical Document	VDC
Readability	Human	Machine
Speed	Human-in-the-loop	Instant
Integrity	Physical tamper resistant at best / Sometimes physically signed	Digitally signed / Originally encrypted at time of presentation
Disclosure	Full document	Full document / Selective / Zero knowledge
Issuer Trust	Fixed	Fixed / Discoverable
Holder Binding Options	Biometric	Cryptographic (Proof of possession) / Biometric / Claims based

Okta’s VDC capabilities

As a leading identity company, Okta is committed to supporting all digital identity verification needs spanning from verification of existing credentials, like mobile drivers licenses, to issuance of bespoke credentials that a company or service might have authority over, such as digital employee badges or concert tickets. Okta’s VDC platform is designed with the latest standards that govern the digital identity space to ensure security, integrity and interoperability between all parties. Okta's VDC platform supports the full credential lifecycle: issuance of custom credentials, verification of both custom and government-issued credentials, and credential revocation via Token Status Lists. The details below cover both verification and issuance capabilities.

Document format	Verification protocols ⁶	Issuance protocols	Verifiable data registry
mdoc <ul style="list-style-type: none"> ISO/IEC 18013-5 ⁵ 	ISO/IEC 18013-7 ⁶ <ul style="list-style-type: none"> Annex B (Custom schemes + OID4VP) Annex C (DC API + mdoc) Annex D (Open HAIP (DC API + OID4VP⁷ + mdoc)) 	Not currently supported	AAMVA Digital Trust Service (DTS) ⁸ or manually sourced issuer certs; supports custom trust anchor lists Bespoke state DMV trust anchors
SD-JWT VC <ul style="list-style-type: none"> IETF 	OpenID <ul style="list-style-type: none"> OID4VP (dc+sd-jwt) W3C <ul style="list-style-type: none"> DC API (dc+sd-jwt) 	OpenID <ul style="list-style-type: none"> OID4VCI (Pre-Authorized Code Flow with tx_code) 	URL-based issuer metadata discovery via .well-known endpoints; supports pre-baked issuer lists, custom URL lists, and offline mappings

[5] ISO/IEC 18013-5
<https://www.iso.org/standard/69084.html>

[6] ISO/IEC 18013-7
<https://www.iso.org/standard/91154.html>

[7] OID4VP
https://openid.net/specs/openid-4-verifiable-presentations-1_0.html

Glossary of terms

Document Format

The Okta VDC platform supports SD-JWT⁸ VC and mdoc document formats for verification. SD-JWT VC is additionally supported for issuance via OID4VCI.

Verification

Okta supports the verification of SD-JWT VCs and mdocs using the OpenID for Verifiable Presentations (OID4VP) protocol. Both formats can be verified via the W3C Digital Credentials API (dc_api.jwt), which is the primary verification mode and supports both same-device and cross-device flows. When the DC API is unavailable, a fallback flow using custom URI schemes (direct_post.jwt) is supported via QR code or deep link.

All wallet responses are encrypted using JARM (JWT Secured Authorization Response Mode) with ECDH-ES + A256GCM, ensuring payload confidentiality over the wire. Verification requests are signed by the verifier's X.509 certificate, allowing wallets to confirm the verifier's identity before sharing any credentials.

The verifier specifies which credentials and claims to request using DCQL (Digital Credentials Query Language), enabling precise, selective requests. Okta supports cryptographic holder binding at both issuance and presentation time — wallets must prove possession of the hardware-backed key bound to the credential before any claims are released.

Issuance

Okta supports the issuance of SD-JWT Verifiable Credentials using the OpenID for Verifiable Credential Issuance (OID4VCI) protocol. This enables organizations to mint custom credentials — such as digital employee badges, membership cards, or professional qualifications — and deliver them directly to a holder's wallet.

Issuance uses the Pre-Authorized Code Flow: the relying party's backend creates a credential offer via the Okta VDC API, specifying a credential configuration and the holder's claims. The wallet resolves the offer, authenticates the holder via a TX code (PIN), and submits a proof-of-possession to bind the credential to a hardware-backed key on the holder's device. Okta signs the credential and returns it to the wallet as a dc+sd-jwt with selective disclosure support.

[8] SD-JWT
<https://www.ietf.org/archive/id/draft-ietf-oauth-selective-disclosure-jwt-22.html>

[9] AAMVA-VICAL
<https://vical.dts.aamva.org/>

Glossary of terms

Claims provided during issuance are held ephemerally and purged automatically — they are never persisted in the Okta platform. Only non-PII credential metadata (status, configuration reference, timestamps) is retained for revocation management.

Wallet Attestation

Before issuing a credential, the Okta VDC platform can verify that the requesting wallet is a genuine, untampered application running on a secure device. The wallet provides a Client Attestation JWT backed by platform integrity services (Apple App Attest, Google Play Integrity). This prevents credential issuance to cloned or modified wallet applications.

Revocation

Issued credentials include a status reference pointing to a Token Status List¹⁰ (RFC 9560) hosted by Okta. Verifiers and wallets can fetch this list to check whether a credential has been revoked. The status list is generated on-demand as a signed JWT, with each credential assigned a unique index in a compressed bit array. This approach allows revocation checks without contacting the issuer for each individual credential, preserving holder privacy.

Holder Binding

Okta supports cryptographic holder binding at both issuance and presentation time. During issuance, the wallet proves possession of a hardware-backed key via a proof JWT, and the credential is bound to that key (‘cnf’ claim for SD-JWT, device key in MSO for mdoc). During presentation, the wallet proves possession again via a Key Binding JWT (SD-JWT) or COSE Sign1 device signature (mdoc). This ensures only the authorized holder can present the credential. Okta currently supports cryptographic holder binding at presentation time.

Trust Model

The platform manages trust in two directions: issuer trust (how verifiers confirm credentials are authentic) and verifier trust (how wallets confirm verifiers are legitimate).

[10] Token Status Lists
<https://www.rfc-editor.org/rfc/rfc9560>

Glossary of terms

Issuer Trust

Trust is configured via individual verification templates. The allowed issuer trust anchors and metadata URLs are configured granularly for each verification the service is configured to perform.

- **mdoc:** Mdocs are supported generically by issuer CA or direct end-cert configuration. For each verification template, a list of allowed CAs/Issuers may be provided. For US use cases, we've integrated support for the American Association of Motor Vehicle Administrators' Verified Issuer Certificate Authority Lists (AAMVA VICAL). For states that don't participate in AAMVA VICAL, we source the issuer certificates manually. Other instances of mdoc verification will use the same model where the platform sources trust lists or the user can manually specify.
- **SD-JWT VC:** The IETF `dc+sd-jwt` format is specified to be URL-based. The issuer is indexed by a URL and the issuer metadata is discovered by querying `.well-known` endpoints. Note `dc+sd-jwt` does not adopt arbitrary distributed identifier (DID) support. Our platform supports pre-baked lists of issuers, like "all Okta verified workforce customers". Additionally, custom trusted issuer URL lists can be configured. And, for offline issuers, custom mappings between issuer URL and issuer key can be specified.

Verifier Trust

Each verifier is provisioned with an X.509 signing certificate issued by the Okta VDC intermediate CA. Signed verification requests include the full certificate chain in the `x5c` JWT header, enabling wallets to validate the verifier's identity before sharing credentials. This protects holders against phishing verifiers.

Glossary of terms

Verification Mechanisms

- **mdoc:** The platform validates the COSE Sign1 device signature against the device key embedded in the Mobile Security Object (MSO), ensuring the credential is presented by its authorized holder. Session transcript computation follows the OpenID4VP specification, incorporating the verifier's nonce, client ID, response URI, and encryption key thumbprint to prevent replay attacks.
- **SD-JWT VC:** The platform verifies the issuer's signature against the trusted certificate chain (extracted from the JWT's `x5c` header), validates the Key Binding JWT to confirm holder possession, and extracts only the selectively disclosed claims. The platform supports both `x5c` and URL-based (`iss`) issuer identification methods.