

UZI certificates as a source of identity

UZI certificates from CIBG can back Verifiable Credentials. The credential is signed with the UZI private key, and its issuer is a `did:x509` DID derived from the certificate. This reuses CIBG's identity checks instead of setting up new issuers. It breaks a chicken-and-egg problem: wallets have no value without credentials, and issuers wait until wallets are in use.

Credential definitions are in the [credential catalog](#) (normative). For the server-certificate credential used in production, including examples, see [X509Credential with UZI server certificates](#).

Principles

- **Reuse CIBG's checks.** CIBG checks organizations (URA) and persons (UZI number, role) before it issues a certificate or card. A credential backed by that certificate needs no new issuer.
- **The key holder vouches for the claim.** Whoever holds the private key makes the claim: the care organization (or its vendor) for a server certificate, the person for a smartcard.
- **Binding.** Credentials are bound together by an identifier they share, such as the URA. Binding works within one Verifiable Presentation, or across presentations that belong together, e.g. one from the vendor and one from the care organization. All credentials are issued to the care organization's `did:web`; a server-certificate credential binds its URA to that `did:web`.
- **Trust the issuing UZI CA, check the pass type.** CIBG has a separate CA per pass type (UZI-register Zorgverlener CA, Medewerker op naam CA, Medewerker niet op naam CA, Server CA). A verifier trusts the CA that issues the pass type it accepts, and checks the pass type in `san.otherName`.

UZI types

UZI type	Pass type	Key held by	Proves	Credential	Status
Server certificate	<code>S</code>	Care organization, often managed by the vendor	The care organization and its URA	X509Credential ; HealthcareProviderCredential (next version)	Production; draft
Zorgverlenerspas	<code>Z</code>	Care professional, on a smartcard	A named care professional, with UZI number and role	HealthcareProfessionalDelegationCredential , PatientEnrollmentCredential	Draft, LSPxNuts pilot

UZI type	Pass type	Key held by	Proves	Credential	Status
Medewerkerspas op naam	N	Named employee, on a smartcard	A named employee of the care organization	PatientEnrollmentCredential	Draft, LSPxNuts pilot
Medewerkerspas niet op naam	M	Employee, on a smartcard	Only that the holder works for the care organization	None: the catalog's credentials reject this pass type	-

Pass type codes as defined by CIBG, see e.g. the [UZI reader of the Ministry of VWS](#).

How the credentials fit together

The URA binds the credentials together. Example from LSPxNuts, all presented by the care organization:

Credential	Signed with	States
HealthcareProviderCredential	Server certificate (S)	This URA belongs to this <code>did:web</code>
HealthcareProfessionalDelegationCredential	Zorgverlenerspas (Z)	This professional delegates these actions to this URA
PatientEnrollmentCredential	Zorgverlenerspas (Z) or medewerkerspas op naam (N)	This patient is enrolled at this URA

The verifier checks:

1. each credential's issuing CA and pass type;
2. that the `did:web` in the HealthcareProviderCredential is the presenter;
3. that all credentials name the same URA.

Issuance

UZI type	How the credential is created
Server certificate	With the certificate's key file, no smartcard needed: with the (Go) go-didx509-toolkit or the Java library . Done by someone at the care organization or its vendor, or automated.
Smartcard	The card holder signs with card and PIN, on a workstation with a card reader. In LSPxNuts this runs through the ZORG-ID SDK; see lspxnuts-pilots .

Governance

Decision	Who	Where
Which credential types exist, and which pass types and CAs they accept	Credential catalog	Credential catalog
Which credentials a use case requires	Use case owner	Discovery Service and policy definitions
Who holds the server certificate's private key	Care organization, or vendor by agreement	Contract between them
Who signs a person-backed credential	The card holder, personally	-
Who gets a certificate or card	CIBG	UZI register

Limits and dependencies

- **Server certificate key held by the vendor:** ideally only the care organization holds the private key of its server certificate. In practice the vendor often holds it, because the same certificate is used for TLS client authentication to other services. The vendor can then create the care organization's credential. This is why Nuts wants to move towards credentials that aren't backed by a TLS certificate (see Outlook).
- **CIBG and PKIoverheid:** a CA change affects every definition that trusts it, e.g. G1 to G4 for server certificates.
- **Workstations:** smartcards need a card reader and software on every workstation where people sign.
- **URA in a combined string:** credentials based on X.509 fields need a pattern to extract the URA from `san.otherName`. The HealthcareProviderCredential contains the URA as a separate field.
- **Personal data:** person-backed credentials contain a name and UZI number.

Outlook

- **Credentials not backed by a TLS certificate:** trust service providers (TSPs) could issue care organization credentials directly, so the organization's identity no longer depends on a key that is also used for TLS.
- **Dezi:** the credential catalog also has a [DeziUserCredential](#), which identifies healthcare workers through Dezi. Whether Dezi will replace UZI cards for this is still open.
- **GIS-VN:** the catalog names future GIS-VN intermediate CAs as trust anchors for personal passes.

Revision #8

Created 2024-10-11 13:11:46 UTC by Steven van der Vegt

Updated 2026-10-28 10:38:04 UTC by Rein Krul