DSR PROTOCOL · CRYPTOGRAPHIC INTEGRITY

How DSR proves authenticity — without a blockchain

A frequent misreading is that a decoupled, own-domain passport must be “just a database” with no real tamper-evidence. It is not. Every EU-Ready passport DSR issues is a **cryptographically signed Verifiable Credential** that anyone can check — offline, from the credential itself — with a tamper-evident hash chain and independent counter-signatures. The result is **independently witnessed, tamper-evident and equivocation-resistant — with no blockchain**: the trust a ledger promises, without its latency, cost or permanence.

1 What we sign

Each passport is issued as a **W3C Verifiable Credential** and signed with the brand’s own **Ed25519** key (an `Ed25519Signature2020` data-integrity proof). The credential is serialised to the **UN Transparency Protocol (UNTP, UN/CEFACT)** data model, so it is a recognisable **DigitalProductPassport** to any UNTP-aware verifier, customs authority or inspection body.

2 Verify offline, from the credential alone

> **The key travels with the credential**
>
> The issuer identity is a self-describing **did:key** — the public key *is* the identifier, carried inside the credential. A customs officer or auditor can verify the Ed25519 signature **with no call back to us and no database lookup**. That is the “don’t trust the brand’s server” property usually claimed only for blockchains — here with none of the consensus overhead.

3 A hash chain, not a ledger

Every new version of a passport carries the **hash of the version before it** into a running **chain hash** (`priorCredentialHash → chainHash`). The history is therefore **tamper-evident and provable** — alter any past version and the chain no longer reconciles — yet the live passport remains **correctable and erasable** when a repair, a resale or the law requires. Tamper-*evident*, not tamper-*frozen*.

4 Independently counter-signed & revocable

> **Third-party attestation**
>
> An independent validator can add a signed **DigitalConformityCredential** (UNTP) — a counter-signature from a party that is not the brand. That is trust **between unrelated actors**, the use-case usually reserved for a shared ledger.

> **Revocation built in**
>
> Status is published as a signed **W3C StatusList2021** credential, so a credential can be marked revoked or reinstated and checked live — without rewriting history.

5 See it for yourself

Live proof — generated for this page view

We just minted a fresh Ed25519-signed sample passport in memory, verified it from the credential alone, then changed one field to show the signature breaks. No database, no ledger.

✓ Signature verified — valid ✗ After tampering — signature_mismatch

Issuer (did:key)	`did:key:z6MkrBYnHHuexS…nxqfr6`
Proof type	`Ed25519Signature2020`
Key fingerprint	`fca0a2f69b126c177f0758f69b3c9f827f880aff3fd6c86aa957e8ab05fff8b7`
Credential hash (SHA-256)	`0cc018c5b03f7195e8b437917be4c022ce9b216720d87908b20344c9b4913516`
Chain hash	`8d63f3d157201846a0a9a42379ee6c2321574d573315d2c7bfef9101fcc05231`

6 Independently witnessed — no blockchain

A tamper-evident history still sits on infrastructure we operate. To remove even the need to trust our clock, we periodically fold every passport version into a single **Merkle checkpoint** and have its root hash **counter-signed by several independent timestamp authorities** (RFC 3161 — the mechanism behind eIDAS qualified timestamps). To back-date the record, or show two different versions of history to two people, **every independent authority would have to collude** — the “don’t trust one server” property a ledger promises, reached without one. Crucially, **only the 32-byte root hash is ever sent** — never product or personal data — so the live passport stays correctable and erasable (GDPR), while its history stays provable.

Live witness — generated for this page view

We folded a freshly signed sample passport into a Merkle checkpoint and anchored its root to several independent timestamp authorities in real time. Each returns a signed proof of the exact moment it saw this root.

Checkpoint root (SHA-256)	`beeaac2c11922cb5bd7c4ad3f87b412c99a7fa3e4f93e9a5ba636580c25c52d2`
Your passport leaf	`c7adb645e76d27691b7bf4dcbf12ae39d40446faca65a2fc14fa2dd75f326aa9`
Inclusion proof	`✓ folds to the checkpoint root`

INDEPENDENT AUTHORITY	SIGNED TIME	SERIAL
✓ FreeTSA	2026-11-24 19:06:59 UTC	`0a86dde7…`
✓ DigiCert	2026-11-24 19:06:59 UTC	`39cc8c925b21…`
✓ Sectigo	2026-11-24 19:06:59 UTC	`00a864b33422…`
✓ GlobalSign	2026-11-24 19:06:59 UTC	`4c55c14b7e0f…`
✓ Certum	2026-11-24 19:06:59 UTC	`0aa87c7775f0…`

5 of 5 independent authorities witnessed this exact root. Each token is independently verifiable; a legally *qualified* eIDAS timestamp authority can be used via the same interface.

Signed, offline-verifiable, tamper-evident and independently witnessed — the trust of a ledger, without the ledger.

Verify a real passport

Every signed DSR passport exposes the same public checks used above: `/api/public/passport/<slug>/verify` returns the verification result, `/versions` re-checks every link in the hash chain, and `/credential.jsonld` hands over the signed credential for offline verification in your own tools. The human-readable verifier lives at `/verify/passport/<slug>`.

Document seal (SHA-256): `3c72b6cdad4af060b551ff7f424cf1a63bcc2b9826af89332baa9243d8c6ae67` Prepared: 24 November 2026