Why business wallet evidence graphs need a qualified W3C VCDM profile

Executive brief for wallet, identity, legal and standardisation decision-makers

Dr. Carsten Stöcker
Spherity GmbH, Düsseldorf

Prepared: 9 November 2026



Except where otherwise noted, this work is licensed under the Creative Commons Attribution 4.0 International License (CC BY 4.0). © 2026 Carsten Stöcker, Spherity GmbH.

Dr. Carsten Stöcker, Spherity GmbH, Düsseldorf. AI assurance graph edition, 9 November 2026. Based on the paper Proofs over Statements in Verifiable Evidence Graphs for the European Business Wallet (reviewed version 1.3, CC BY 4.0) [1].

Why a common evidence stack matters

Knowledge graphs and verifiable evidence and assurance graphs should form a core evidence layer of high-performance, safe and compliant industrial AI infrastructures. Knowledge graphs connect meaning; evidence graphs preserve authenticated sources; assurance graphs connect claims and evidence to requirements and the reasons for accepting a decision. Business wallets can supply the credentials for this infrastructure. Most relationships may be supported by ordinary electronic attestations of attributes (EAAs); selected relationships need qualified attestations (QEAAs) where the applicable policy requires them. For retained, repeatedly verified graph evidence, we recommend one W3C VCDM 2.0 JSON-LD profile with RDF-based Data Integrity proofs for both levels, so a shared graph and verification pipeline can preserve their different assurance properties. This requires an eligible qualified profile and qualified issuance; using the same format does not itself confer qualification. The result would support reusable, machine-checkable evidence for Trusted AI, trusted industrial AI and trusted physical AI. It would not, by itself, prove that evidence is true or an AI system is safe or compliant.

1 Citizen presentations and industrial evidence graphs

A transaction presentation can establish a person's eligibility, company identity or an employee's mandate. SD-JWT VC and mdoc support these exchanges under the applicable disclosure and holder-binding profile [2–4]. Retention and reuse depend on the workflow; neither format is restricted to a citizen presenting a credential once.

An industrial evidence graph has a further lifecycle. A battery passport links supplier, carbon-footprint, recycling, test and verification records from different issuers. Customs and aerospace designs likewise combine evidence for a shipment or machine decision. The relying party must

preserve sources, meanings and versions. Selected identity or representation relationships may require qualified assurance; ordinary product and process evidence can remain non-qualified. Both should coexist in the graph, with assurance visible to policy.

Figure 1 illustrates this distinction: a citizen presents selected credentials, while organisations combine attributable evidence in a graph. Qualified electronic attestations of attributes (QEAAs) support selected assurance-critical relationships alongside ordinary electronic attestations of attributes (EAAs) within one evidence stack.

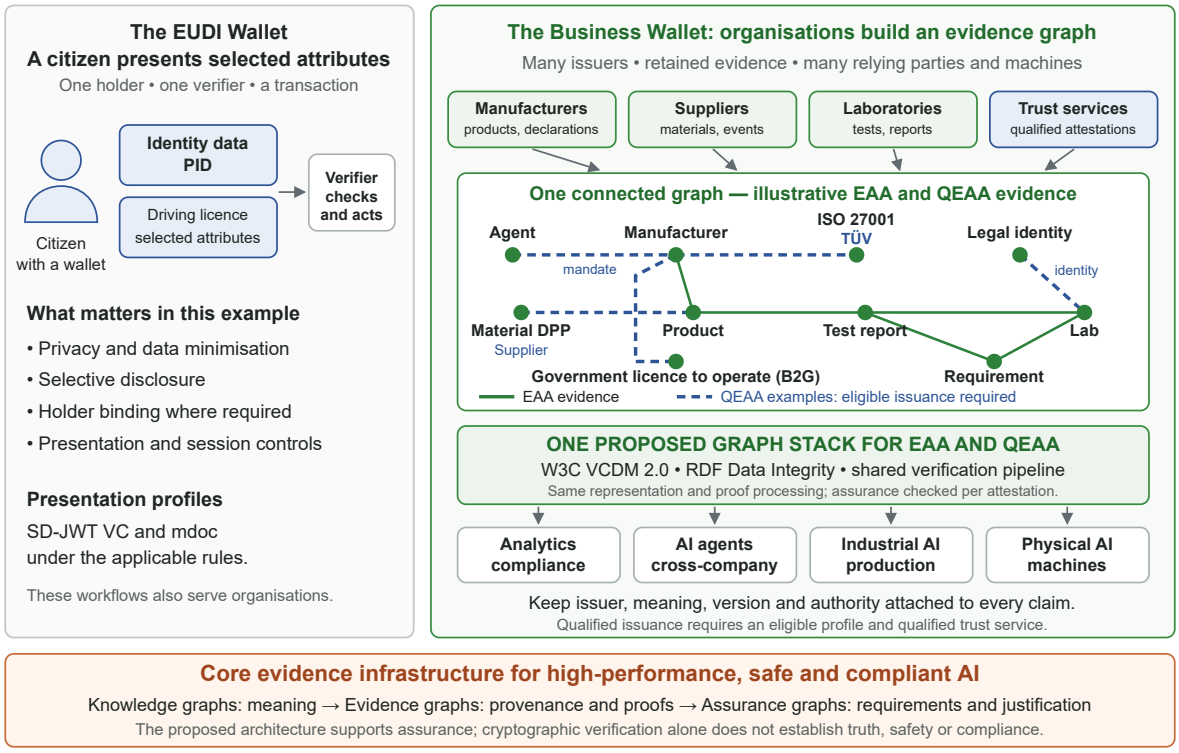


Figure 1: Citizen presentation and retained business evidence. Green relationships are EAA-backed; blue dashed relationships require QEAAs under the decision policy. The proposed common graph stack preserves issuer, scope, status and assurance level. Colour denotes supporting attestations, not independently signed edges. The examples illustrate workloads, not exclusive wallet capabilities.

Four requirements follow:

- **Meaning after combination.** Shared identifiers and agreed semantics distinguish a battery model from a batch, or a test result from legal status. Other formats can provide these through explicit identifiers and governed mappings; merging field names without agreed meaning is insufficient.
- **Attribution after combination.** Each claim must remain bound to the issuer-authenticated source. A graph-native design preserves a recoverable signed snapshot. A source-backed design retains the original signed credential and the evidence needed to verify its mapping.
- **Repeated machine verification.** Suppliers, manufacturers, auditors and automated systems may reuse evidence. The verification schedule is a policy decision based on trust boundaries, integrity, status and freshness; every local read need not repeat every signature check.

- **Authority under the decision policy.** Identity, mandates and accreditation scope establish who may make a claim. Policy and the qualified issuance scheme determine which attestations require and can obtain qualification. The proposed common EAA/QEAA stack shares a data model, graph-proof profile and pipeline, while checking each qualified provider, trust information, scope, status and issuance conditions. Seamless processing preserves the different legal assurance levels.

Ordinary attestations can far outnumber qualified attestations. In many industrial and AI evidence designs, ordinary EAAs can greatly exceed QEAAs: batch declarations, test results, logistics events and signed AI-evaluation records multiply per asset, event or model version, while qualified identity, mandates and operating permissions can be reused. The paper's battery scenario assumes 10–100 million self-asserted credentials against about 10,000 qualified credentials, one qualified credential per 1,000–10,000 ordinary credentials; this is an illustrative workload, not a measured market ratio [1]. The mix depends on the use case: Spherity's author-provided research estimate is 1 QEAA per 5 ordinary EAAs in high-risk use cases, critical infrastructure, regulated industries and sensitive processes [1]. This is a scenario estimate; the supplied package contains no supporting dataset or methodology.

Industry already builds on linked-data standards: GS1 EPCIS 2.0 supports JSON-LD and RDF integration [5]; UNTP's pre-production profile uses W3C VCDM 2.0 and JSON-LD for product passports, conformity and traceability credentials, with enveloping proofs [6]; Siemens Graph Studio offers RDF-based enterprise knowledge graphs for AI agents [7]. Graph facts and EPCIS events are not automatically EAAs: they must be issued as attestations to count as such. As AI knowledge graphs expand, we expect substantially greater volumes and reuse of attestable evidence; these examples establish technical uptake, not a measured EAA/QEAA growth rate. QEAA credential-format standardisation should therefore accommodate RDF, JSON-LD and W3C VCDM interoperability, shared identifiers, preservation of source proofs, mixed assurance levels and repeated machine verification, so ordinary and qualified evidence can interoperate across workloads with different assurance mixes.

Foundational assurance is a dependency requirement. If policy requires a QEAA for an identity, mandate or licence-to-operate relationship, an ordinary EAA at that point cannot meet the requirement. A decision path requiring that link lacks the target assurance unless policy supplies an admissible alternative. If all admissible paths share the deficient foundation, the graph cannot justify that decision at the required assurance level. This neither requires every edge to be qualified nor invalidates independent evidence. Intersecting relationships make a common credential profile useful for expressing dependencies and checking assurance consistently; the format cannot repair missing assurance.

Hybrid signatures and post-quantum cryptography (PQC). A cryptographic hybrid differs from nesting credential formats. Combining a traditional signature with a post-quantum signature can increase proof size, key-management work and verification steps; mixed and nested credential formats add more combinations to implement and test. W3C Data Integrity already supports multiple digital signatures through unordered proof sets and ordered proof chains [8, 9]. These can carry multiple proofs without nesting credential formats, but do not alone constitute a secure hybrid scheme or a post-quantum guarantee. A hybrid profile must define supported cryptosuites, key and credential bindings, which components must verify, and protection against proof removal or downgrade. Where policy requires both signatures, it must detect a missing proof; accepting either is insufficient [10]. A common format reduces integration variation; cryptographic costs, post-quantum suite validation and qualified-profile eligibility still need separate treatment. The paper's existing timings do not measure these hybrid paths.

2 What a graph proof covers

W3C VCDM 2.0 provides a credential data model; JSON-LD can express its linked-data claims. The recommendation here concerns an RDF-based Data Integrity suite, not every W3C VC or Data Integrity mechanism [8, 11]. At ingestion, the verifier expands under approved contexts, canonicalises the credential and proof configuration, and verifies the issuer's signature. For later verification from a graph store, it reconstructs the same proof input from the preserved source snapshot and original graph positions. An arbitrary isolated statement cannot be verified with a full-credential proof without that snapshot.

SD-JWT VC authenticates a protected JWT and commitments to disclosed claims; mdoc authenticates document elements through its issuer-signed structure and digests. Those proofs remain useful for retained evidence if their source inputs and interpretation are preserved. They do not automatically authenticate an application's subsequent graph projection. Two distinctions matter:

1. **The source-to-graph binding.** A source-backed verifier authenticates the original credential and validates the mapping to the stored statement. If an unpinned context changes, an unchanged envelope signature can still verify while a fresh expansion changes. A governed profile can prevent acceptance by fixing and authenticating the interpretation inputs. The application must use the verified view in either architecture.
2. **Nested processing and cost.** Putting JSON-LD inside an SD-JWT requires envelope processing and a controlled route from disclosed claims to graph statements. Figure 2 shows the practical steps; Figures 3 and 4 compare the measured and modelled costs. In the paper's 146-quad fixture, the stored graph-proof check took about 0.28 ms; envelope processing with fresh expansion took 1.06 ms. Including issuance and ingestion, the graph path became cheaper from the fifth additional check. A protected cached mapping is another valid design; the envelope-only timing of 0.25 ms excluded checks of that mapping and its graph binding.

Complete cryptographic examples. Appendix A supplies a fully signed VCDM credential and the same credential inside an SD-JWT, with exact signatures, public keys and runnable checks. The plain path verifies one Ed25519 proof. This nested example requires the outer ES256 signature, a disclosure commitment and the inner proof to pass. It therefore adds work before the graph can be used. This is a two-proof construction; the paper's envelope-only architecture has no inner Data Integrity check. These new tests establish verification behaviour, not the timings quoted above.

Disclosure and confidentiality still matter. Industrial evidence may contain sensitive process information, personal data or commercially confidential measurements. Access control, selective disclosure and appropriately scoped references remain relevant. The separate library comparison found faster SD-JWT presentation verification than the tested Data Integrity selective-disclosure path, but the implementations and checks differ. It is not a universal ranking of formats.

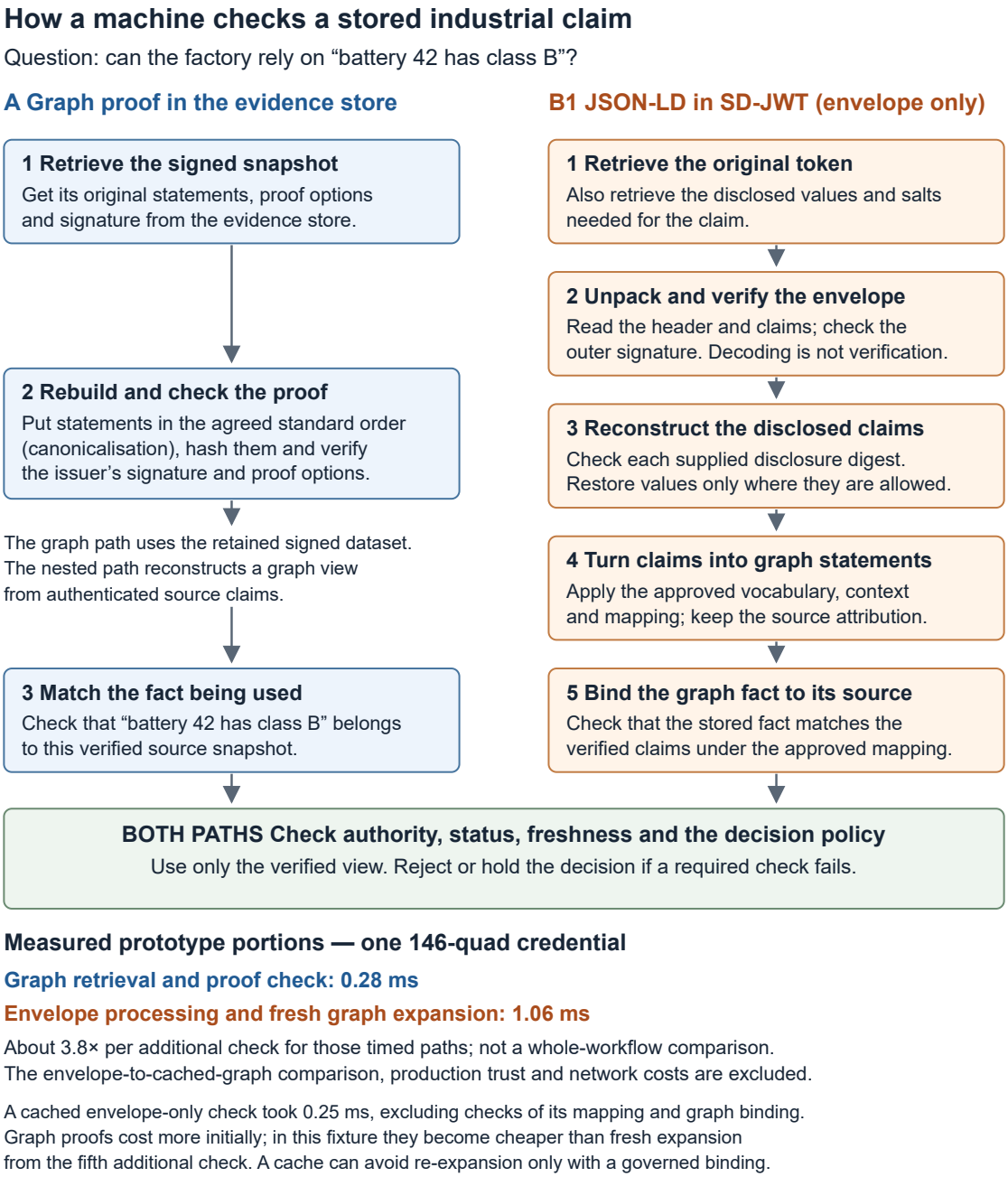


Figure 2: A machine checking the same stored battery claim through A and envelope-only B1. The dual-proof nested example in Appendix A additionally verifies the inner graph proof. Unnesting includes authentication, disclosure reconstruction and controlled interpretation; decoding alone proves nothing. The displayed timings cover prototype portions of the process, not all steps or complete security checks. A cached mapping can replace fresh expansion only under a policy protecting its source binding, freshness and status. Source: paper [1], Sections 5.2–5.5; explanatory diagram added in this edition.

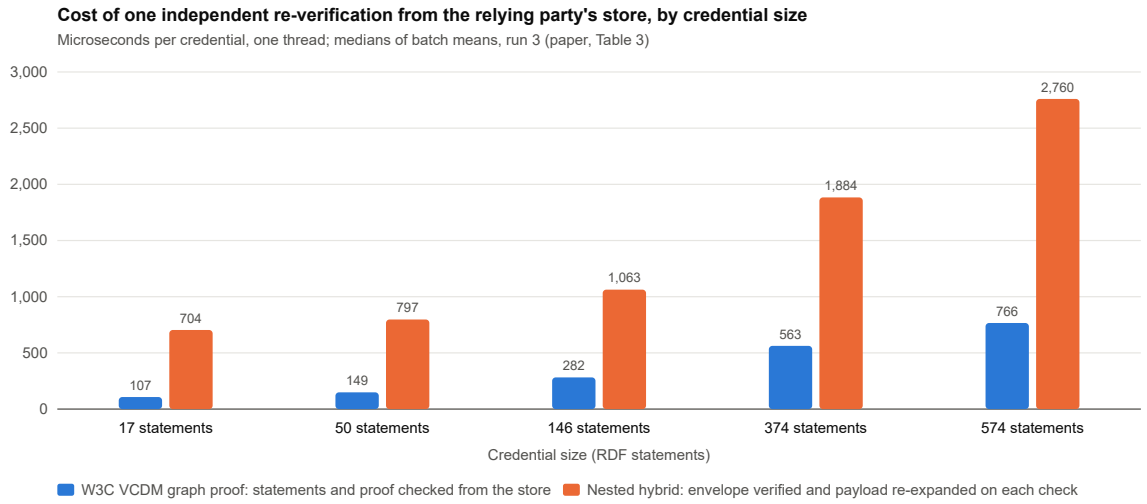


Figure 3: Prototype times per additional verification, by credential size. The orange path verifies the retained envelope and freshly expands its payload; comparison with a cached graph is outside that timed path. The blue path retrieves and verifies the stored credential and proof inputs. At 146 quads, the separately measured envelope-only path took 0.25 ms, excluding validation of the cached view and source binding. Source: paper [1], Table 3.

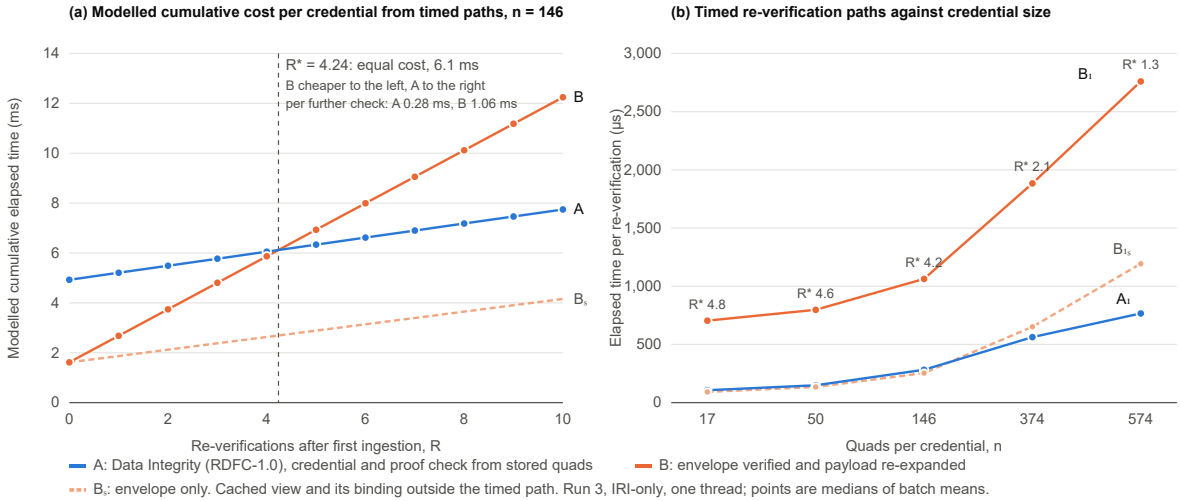


Figure 4: Cumulative processing and per-check costs from the full paper, including the envelope-only alternative. The graph path costs more initially and crosses over against fresh expansion at 4.24 additional checks for the 146-quad fixture. The dashed envelope-only curve assumes a trusted cached mapping and excludes its validation. These are prototype processing costs, not complete service latencies. Source: paper [1], Figure 4.

3 Qualified authority and the claims it supports

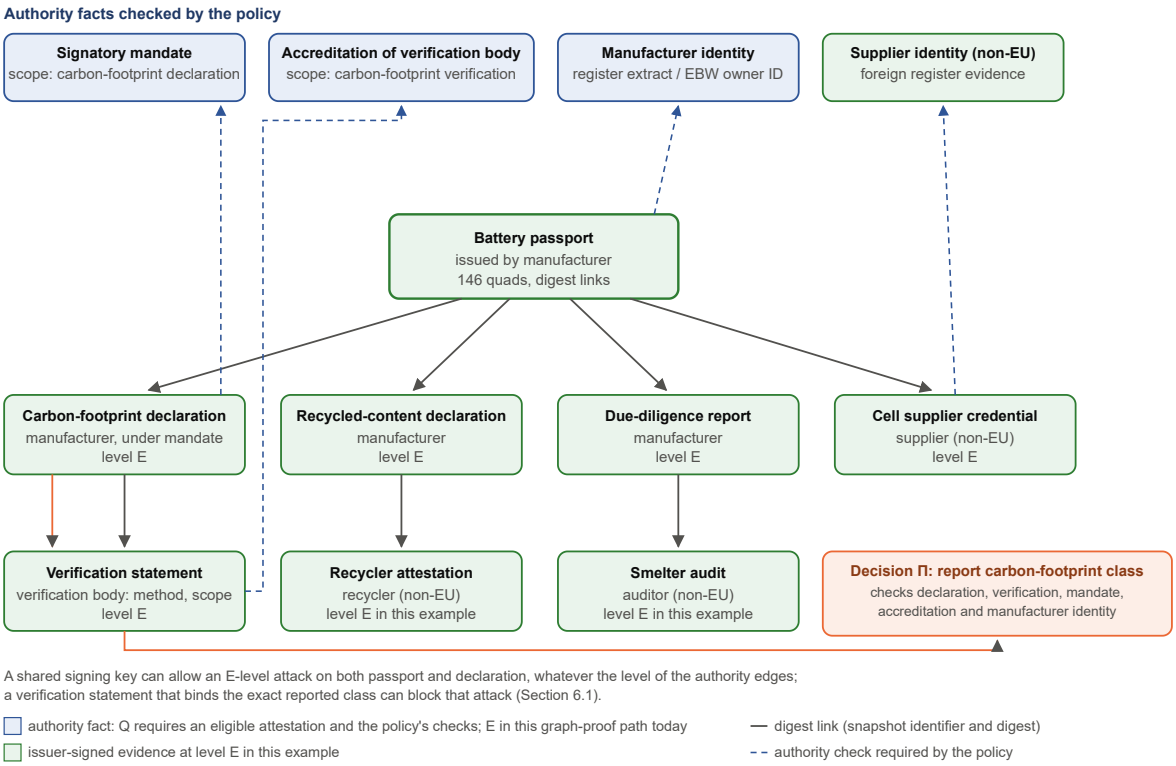


Figure 5: The battery-passport design distinguishes content, evidence links and authority checks. A valid mandate and accredited issuer do not by themselves authenticate the exact reported carbon-footprint class. The diagram includes a content attack that can remain possible even when authority attestations are qualified. Source: paper [1], Figure 6.

As Figure 5 illustrates, qualifying selected authority attestations can support many ordinary product and process credentials. The paper's battery scenario assumes about 10,000 authority credentials supporting 10–100 million content credentials annually; these are illustrative workload volumes. Its attack model also includes content substitutions and joint attacks. A genuine manufacturer with a valid mandate can issue an incorrect class, and a verification statement covering only the method may not bind that class. Qualification does not propagate automatically through a link. Article 45b gives the specified attestations legal effect under its conditions; it does not make every downstream claim correct [12].

The format request is narrower: make an appropriate graph-proof profile eligible for qualified issuance, so ordinary EAAs and required QEAAs can use one representation, graph store and verification pipeline. This avoids making the application switch credential interpretation solely because one relationship needs a higher assurance level. It does not remove qualified-provider or lifecycle checks. Alternatively, an organisation can retain compliant source credentials, operate their verifiers and govern the mappings into its graph. That introduces additional integration and audit obligations, rather than an automatic loss of assurance.

4 Three uses and a common evidence requirement

- **Battery passports and supply-chain compliance.** From 18 February 2027, Article 77 requires a battery passport for each electric-vehicle battery, light-means-of-transport

battery and industrial battery above 2 kWh placed on the market or put into service [13]. The paper's million-unit manufacturer scenario assumes 10–100 million linked credentials annually. The reporting decision needs evidence for the reported class as well as the applicable issuer, mandate and accreditation checks.

- **E-commerce customs review.** The paper reconstructs an evidence-graph design informed by published customs material, a public DHS SVIP demonstration in 2021 and trade identity specifications [14–18]. It links product, order, invoice, packing and transport evidence, then adds a worked conformity chain for children's sleepwear [19]. The complete design is not asserted to be an agency's production deployment. Verified attribution helps detect impersonation; a genuinely issued false declaration still needs corroboration, content checks or examination.
- **An AI safety supervisor in aerospace.** A deterministic supervisor evaluates proposed maintenance actions against a verified requirements graph. It can check evidence at load and again when policy requires. The cited 3 ms versus 16 ms for 20 supporting credentials is an arithmetic projection from individual checks, not a measured end-to-end supervisor result, worst-case execution bound or certified safety deadline. Policy evaluation and status checks need separate budgets; immediate protective functions remain outside the graph-dependent path.

The common requirement is a verifiable connection from a decision input to its source, interpretation, version and authority. Graph-native verification provides one way to achieve it. A properly governed source-backed architecture provides another. Their costs depend on what is retained, checked and reused.

5 What is being asked

1. **W3C.** Develop a high-assurance VCDM 2.0 profile with RDF-based Data Integrity proofs: fixed contexts and processing, rejection of lost terms, source-snapshot preservation, graph admission rules, work and execution limits, permitted cryptosuites, status, organisational key binding, and conformance tests.
2. **ETSI ESI.** Extend the relevant issuance and presentation profiles, including TS 119 472-1 clause 7 [20], with precise graph-proof, certificate and key-binding, status, context and security requirements.
3. **European Commission.** Consider referencing the completed profile for the qualified and public-source attestations in scope and in future European Business Wallet rules, with scheme-specific adaptations. Company identity and representation are relevant; accreditation and autonomous-agent mandates require their applicable authority and scheme.
4. **Sector bodies.** Test the profile in supply-chain and industrial workflows, including confidential evidence, retained-source alternatives, status changes and cross-border trust requirements.

The proposal adds a graph-proof option alongside existing profiles. Format eligibility, issuer qualification and successful validation are separate conditions.

6 The formats side by side

The table compares verification architectures and their conditions. Its B1 column uses an envelope-only proof, unlike the dual-proof nested example in Appendix A. The graph-store timings are the paper's run-3 medians on one thread. The separate one-shot results and attack tests use different implementations and checks [21, 22]. They do not establish a universal performance or security ranking.

The rows distinguish graph semantics, source retention, repeated verification, presentation features and qualification. A positive result depends on the stated profile and operating conditions.

Property	**SD-JWT VC**	**ISO mdoc**	**B1: envelope-only JSON-LD in SD-JWT**	**VCDM 2.0 with RDF-based Data Integrity**
Meaning in a graph	Needs shared identifiers and an agreed mapping from claims	Needs a mapping from typed document elements	Linked-data content plus governed context and disclosure rules	Shared RDF semantics; profile must reject undefined or lost terms
What is authenticated	Protected JWS input and disclosure commitments	Issuer-signed structure and element digests	Envelope input; graph interpretation needs a controlled binding	Canonical credential dataset and proof configuration
Verification after storage	Retain the source and required disclosures; validate the graph mapping	Retain source verification evidence; validate the graph mapping	Retain token, disclosures and interpretation inputs; check the source-to-view binding	Recover the full source snapshot, graph positions, proof options and signature
Additional check at 146 quads in the paper	This graph-store path not separately measured	This graph-store path not measured	1.06 ms with fresh expansion; 0.25 ms envelope-only, excluding cached-view checks	0.28 ms; higher initial cost; cheaper than fresh expansion from check 5
Qualified authority on the path	Eligible under the applicable issuance profile; map and validate as required	Eligible under the applicable issuance profile; map and validate as required	Eligibility depends on the complete profile and scheme, not nesting alone	Common graph-proof path possible if a qualified profile becomes eligible
Selective disclosure	Claim disclosures with issuer commitments	Selected data elements and their issuer-authenticated digests	Claim disclosure plus rules protecting graph interpretation	Separate suites such as ecdsa-sd-2023 or BBS; not provided by every graph-proof suite

Continued on next page

Property	SD-JWT VC	ISO mdoc	B1: envelope-only JSON-LD in SD-JWT	VCDM 2.0 with RDF-based Data Integrity
Holder binding and linkability	Key binding where required; reused issuer JWT can link presentations	Device binding and linkability depend on the profile	Inherits the envelope's bindings; mapping must preserve them where relevant	Presentation and suite dependent; full-credential proofs do not imply unlinkability
Separate one-shot study [21]	0.30 ms, tested exchange	0.27 ms, tested exchange	Nested path not measured in that study	0.53 ms; subset re-signed in that experiment, so authentication guarantees differ
Security controls	Allowed algorithms, key and disclosure checks, status and policy	Issuer structure, element digests, device/session checks and policy	Envelope checks plus fixed interpretation and verified source-to-graph binding	Controlled loader, context checks, graph admission, work and execution limits at relevant processing steps
Current qualified issuance baseline	Referenced profiles, with applicable conditions	Referenced profiles, with applicable conditions	No automatic eligibility from using an SD-JWT wrapper	No graph-proof profile referenced in the examined baseline
Architectural fit	Presentations or governed source-backed graph use	Presentations or governed source-backed graph use	Useful when interoperability justifies its retention and mapping obligations	Supports graph-native reuse under the specified storage and processing conditions

Why a graph-proof profile is useful for repeated industrial verification. The benefit is a common issuer-authenticated representation for retained graph evidence, subject to explicit storage, processing and legal conditions:

- **The proof authenticates the graph statements and proof configuration.** A preserved signed snapshot can be recovered from the store for verification. This requires lossless source boundaries and original graph positions, not an arbitrary union of statements.
- **The tested reuse path avoids fresh JSON-LD expansion.** It still retrieves the snapshot, canonicalises, hashes and verifies. Source-backed systems can also cache verified mappings under a policy that protects the view, binding, freshness and status. The measured envelope-only check does real signature verification; its timing omits the cached-view checks.
- **One profile can simplify a chosen verification path.** Making it eligible for qualified attestations would let an organisation apply the same graph-proof rules to those attestations and ordinary evidence. Mixed-format systems remain possible, with additional verifiers and governed bindings.
- **Nesting needs a justified purpose.** An envelope can serve presentation or interoperability requirements. Envelope-only B1 needs a governed binding to the stored graph. Retaining a signed inner credential preserves its graph-proof route, while both layers must be checked whenever policy relies on both. A presentation-only wrapper need not be rechecked for every later use of independently verified inner evidence.
- **Resource bounds must be enforced.** Graph canonicalisation occurs during verification as well as issuance. Input restrictions, fixed loaders, work budgets and interruptible execution limits are needed. Library defaults or a valid-looking issuer identifier do not establish a safe execution deadline.

For repeated verification of retained industrial graph evidence, prefer a common W3C VCDM 2.0 JSON-LD profile with RDF-based Data Integrity as the persistent evidence format, with an eligible profile for any required qualified attestations. Shared semantics, preserved source boundaries and compatible validation rules are essential; JSON-LD syntax alone is insufficient. Add envelopes where their presentation or interoperability function justifies the extra bindings. Governed mixed architectures remain valid, and the complete processing path determines their cost.

7 What the evidence is

The manuscript supplies formal models, worked designs and prototype benchmarks against an in-memory quad store: 500 credentials per size, 17–574 quads, with selected reference-library and adversarial-input checks. Results are conditional on the stated processing paths, one machine and synthetic workloads. The supplied publication package contains the manuscript and figures, but not the separately referenced benchmark code or raw data. This review checks consistency with the reviewed version 1.3; it does not independently reproduce the measurements or validate a production deployment. The complete signed examples in Appendix A have, separately, been cryptographically tested; their public fixtures and verification scripts are supplied with this edition.

8 Why now

The EUDI issuance baseline references the relevant SD-JWT VC and mdoc profiles through clauses 2–6 of ETSI TS 119 472-1; clause 7's W3C VCDM material is outside that reference. The rules anticipate W3C support when new profiles are available [23–25]. The European Business Wallet remains a legislative proposal [26]. These developments create a standards opportunity to define a graph-proof option for organisational evidence. This is a specific profile-eligibility issue, not a rule that EUDI credentials are only for natural persons or that W3C credentials are prohibited in business.

9 Graphs as core evidence infrastructure for Trusted AI

Our architectural position is that knowledge graphs and verifiable evidence and assurance graphs belong at the core of high-performance, safe and compliant AI infrastructures. A knowledge graph expresses what entities and relationships mean. A verifiable evidence graph binds assertions to authenticated issuers, source snapshots and versions. An assurance graph links a system claim to the evidence, requirements, assumptions and reasoning that justify accepting it. These can be complementary views over a shared information architecture; a graph edge does not automatically establish an assurance conclusion.

This connects the paper to Spherity's research on **Trusted AI, trusted industrial AI and trusted physical AI**. An AI agent can trace a proposed action to its supporting evidence and authority. Industrial AI can link products, process conditions, tests and compliance requirements across organisations. Physical AI, such as robots or autonomous machines, can link permitted actions and operating limits to current evidence and authorised actors. Credential checks support this architecture alongside the safety engineering, monitoring and controls needed for physical action.

1. **Use one graph while preserving different assurance levels.** An ordinary EAA may report a component's test result, while policy requires a QEAA for an identity or representation relationship used to accept it. Both should pass seamlessly through the proposed common profile. The decision engine still checks the assertion covered, the issuer's authority and current status. Qualified authority does not qualify the test result, other graph edges or the AI output. Full-credential graph-proof verification still needs recoverable signed source snapshots.
2. **Make the justification for a decision inspectable.** Connect a proposed action to the relevant requirement, the evidence supporting it and any assumptions or unresolved conflicts. For a factory decision, this might mean showing which test supports a component's acceptance and which mandate authorises the accepting agent. Cryptographic verification authenticates attribution and integrity; technical validation and the decision policy determine whether the evidence is sufficient. Record derived conclusions as derived, with their rule or model version and supporting sources, rather than presenting them as issuer-signed facts.
3. **Design performance and safety together.** Shared semantics and reusable verified evidence can reduce repeated interpretation and support efficient queries. The paper measures specific verification paths, not AI throughput or operational safety. Set freshness rules, revocation handling and the verification schedule, then benchmark the complete decision path. Safety-critical consumers also need resource limits, runtime controls and a defined response to missing or stale evidence. Physical systems need a designed safe response when verification is unavailable.

The engineering recommendation is a common JSON-LD/VCDM 2.0 profile with RDF-based Data Integrity for persistent industrial evidence graphs, with envelopes serving defined additional requirements. Make an appropriate graph-proof profile eligible for the qualified attestations required on those paths. A common profile can simplify expression and checking of intersecting dependencies; it cannot supply missing assurance. A missing required QEAA defeats each path that requires it; if no admissible alternative exists, the target decision assurance is unavailable. Independent evidence remains verifiable.

Data Integrity already supports multiple signatures through proof sets and chains, but PQC hybrids still need validated suites, key and credential bindings, mandatory-component checks and downgrade protection [8–10]. More signatures do not elevate an EAA to a QEAA. One stack can simplify integration while preserving assurance levels; safety and compliance still require sound engineering and governance.

Contact: carsten.stoecker@spherity.com. The author is chief executive of Spherity GmbH, a verifiable-credential software provider that participated in the DHS pilot informing the customs example. The full paper and figures are licensed under CC BY 4.0. This edition aligns the PQC and foundational-assurance guidance with the reviewed full paper.

Appendix A. Complete signed credentials and verification

The following are complete, executable demonstration credentials, with actual signatures rather than placeholder proof values. They assert that battery 4711 has a recycled-cobalt share of 16 percentage points. The vocabulary is illustrative. The issuer, public keys and contexts are fixed local test inputs; the example does not establish a real recycler's authority, truth of the claim, or QEAA status. Both variants carry exactly the same signed inner credential. [8, 9, 11]

The supplied `examples/` folder contains the original JSON, exact token, pinned contexts, public verification scripts and results. Verification uses 9 November 2026, 12:00 UTC. No private key is needed or included. The local profile requires the stated issuer, credential types, algorithms and unit, the credential disclosure, matching inner and outer issuers, and both proofs in the nested case. It permits the absence of holder binding and credential status; neither a live revocation service nor a wallet protocol is tested.

A.1 The plain VCDM credential

Listing 1 contains the credential itself. The first context is the W3C VCDM 2.0 context; the inline context defines the material terms and an IRI-valued unit. Here `ex:percentagePoint` means percentage points. The `proofValue` is a real Ed25519 signature in multibase encoding. It covers the suite's canonical credential and proof-configuration inputs, not the indentation of this JSON. The complete-document suite used here, `eddsa-rdfc-2022`, does not provide selective disclosure of inner attributes.

Listing 1: Complete VCDM 2.0 credential with its Data Integrity proof

```
{
  "@context": [
    "https://www.w3.org/ns/credentials/v2",
    {
      "@protected": true,
      "ex": "https://example.org/",
      "MaterialCredential": "ex:MaterialCredential",
      "recycledCobaltShare": "ex:recycledCobaltShare",
      "unit": {
        "@id": "ex:unit",
        "@type": "@id"
      }
    }
  ],
  "id": "https://issuer.example/vc/1",
  "type": [
    "VerifiableCredential",
    "MaterialCredential"
  ],
  "issuer": "https://issuer.example",
  "validFrom": "2026-11-09T00:00:00Z",
  "credentialSubject": {
    "id": "https://id.example/battery/4711",
    "recycledCobaltShare": 16,
    "unit": "ex:percentagePoint"
  },
```

```
  "proof": {
    "type": "DataIntegrityProof",
    "created": "2026-11-09T00:00:00Z",
    "verificationMethod": "https://issuer.example#ed25519",
    "cryptosuite": "eddsa-rdfc-2022",
    "proofPurpose": "assertionMethod",
    "proofValue": "zGaCddi1PXdPWMukSZvRoRtpVMszwmuaSmSWgHRAjf3fXEkumMg2vWDejF
    83qnBqbvmDe4B1FtTQ8omH8CcC42MM"
  }
}
```


A.2 Public verification material

The verifier obtains Listing 2 from its pinned local issuer record, not from an unauthenticated claim in the credential. The controller authorises the embedded public Multikey for `assertionMethod`. Listing 3 is the separate P-256 key pinned for the same example issuer's outer envelope. These are public verification keys, not signing secrets. The standard context documents and their SHA-256 hashes are included in `examples/contexts/`; unknown remote document requests are rejected.

Listing 2: Pinned issuer controller and authorised Ed25519 verification method


```
{
  "@context": "https://w3id.org/security/v2",
  "id": "https://issuer.example",
  "assertionMethod": [
    {
      "@context": "https://w3id.org/security/multikey/v1",
      "id": "https://issuer.example#ed25519",
      "type": "Multikey",
      "controller": "https://issuer.example",
      "publicKeyMultibase": "z6MkjNfty9qLyRo9AgQh71j9QX4H8hYQCyUF8r2Z4W7tvfau
    "
    }
  ]
}
```


Listing 3: Pinned public key for the outer ES256 signature


```
{
  "kty": "EC",
  "x": "98iFb5IBJKje3iOVduWmGWSDXXoW6L4t9f2NqNw9VR0",
  "y": "fbonvgL9uaDKmK_ZlswV1GqqnLf5Xup-bw_Q1ZlSkjU",
  "crv": "P-256",
  "kid": "https://issuer.example#es256",
  "alg": "ES256",
  "use": "sig"
}
```

A.3 The complete nested credential

The outer format uses the `dc+sd-jwt` type and an example-specific `vct` profile. Listings 4 and 5 show its decoded protected header and signed payload. The single `_sd` digest commits to a disclosure containing a random salt, the claim name `credential`, and the entire signed object in Listing 1. It therefore reveals the inner credential as one unit; it does not selectively reveal individual battery attributes. [2, 3]

Listing 4: Decoded protected header of the outer SD-JWT

```
{
  "alg": "ES256",
  "typ": "dc+sd-jwt",
  "kid": "https://issuer.example#es256"
}
```

Listing 5: Decoded signed payload, before disclosure reconstruction

```
{
  "iss": "https://issuer.example",
  "iat": 1794182400,
  "vct": "https://issuer.example/type/nested-material",
  "_sd_alg": "sha-256",
  "_sd": [
    "TqMFz3Asrf4bO_jEX_avXVLNzLfZ-P-KSxU4FnAxcM0"
  ]
}
```

Listing 6 is the complete wire value: issuer-signed JWT, a tilde, the base64url disclosure, and a final tilde. It has no ellipses or omitted signature bytes. Long-value wrapping in these listings is typographical; use the supplied files for exact inputs, including `examples/nested.sd-jwt`. The decoded disclosure and fully reconstructed claims are also supplied as JSON files. The final empty segment means there is no key-binding JWT under this example profile.

Listing 6: Complete nested SD-JWT serialization, including the disclosure

```
eyJhbGciOiJFUzI1NiIsInR5cCI6ImRjK3NkLWp3dCIsImtpZCI6Imh0dHBzOi8vaXNzdWVyLmV4Y
   W1wbGUjZXMyNTYifQ.eyJpc3MiOiJodHRwczovL2lzc3Vlci5leGFtcGxlIiwiaWF0IjoxNzk
   0MTgyNDAwLCJ2Y3QiOiJodHRwczovL2lzc3Vlci5leGFtcGxlL3R5cGUvbmVzdGVkLW1hdGVy
   aWFsIiwiX3NkX2FsZyI6InNoYS0yNTYiLCJfc2QiOlsiVHFNRnozQXNyZjRiT19qRVhfYXZYV
   kxOekxmWi1QLUtTeFU0Rm5BeGNNMCJdfQ.esVwHXL1gvrXOOLrYibjIORj-ZQbuVJXHvr2Iwc
   l9gilcEUfpyV_Lgbo9uyb4eRVlErBO2a6ke0Z0PN6O4WCKA~WyJ2LU5YUzFnQ19YTGZ5SFJNV
   UpfQnpRIiwiY3JlZGVudGlhbCIseyJAY29udGV4dCI6WyJodHRwczovL3d3dy53My5vcmcvbn
   MvY3JlZGVudGlhbHMvdjIiLHsiQHByb3RlY3RlZCI6dHJ1ZSwiZXgiOiJodHRwczovL2V4YW1
   wbGUub3JnLyIsIk1hdGVyaWFsQ3JlZGVudGlhbCI6ImV4Ok1hdGVyaWFsQ3JlZGVudGlhbCIs
   InJlY3ljbGVkQ29iYWx0U2hhcmUiOiJleDpyZWN5Y2xlZENvYmFsdFNoYXJlIiwidW5pdCI6e
   yJAaWQiOiJleDp1bml0IiwiQHR5cGUiOiJAaWQifX1dLCJpZCI6Imh0dHBzOi8vaXNzdWVyLm
   V4YW1wbGUvdmMvMSIsInR5cGUiOlsiVmVyaWZpYWJsZUNyZWRlbnRpYWwiLCJNYXRlcmlhbEN
   yZWRlbnRpYWwiXSwiaXNzdWVyIjoiaHR0cHM6Ly9pc3N1ZXIuZXhhbXBsZSIsInZhbGlkRnJv
   bSI6IjIwMjYtMTEtMDlUMDA6MDA6MDBaIiwiY3JlZGVudGlhbFN1YmplY3QiOnsiaWQiOiJod
   HRwczovL2lkLmV4YW1wbGUvYmF0dGVyeS80NzExIiwicmVjeWNsZWRDb2JhbHRaGFyZSI6MT
   YsInVuaXQiOiJleDpwZXJjZW50YWdlUG9pbnQifSwicHJvb2YiOnsidHlwZSI6IkRhdGFJbnR
   lZ3JpdHlQcm9vZiIsImNyZWF0ZWQiOiIyMDI2LTExLTA5VDAwOjAwOjAwWiIsInZlcmlmaWNh
```

```
dGlvbk1ldGhvZCI6Imh0dHBzOi8vaXNzdWVyLmV4YW1wbGUjZWQyNTUxOSIsImNyeXB0b3N1a
XRlIjoiZWRkc2EtcmRmYy0yMDIyIiwicHJvb2ZQdXJwb3NlIjoiYXNzZXJ0aW9uTWV0aG9kIi
wicHJvb2ZWYWx1ZSI6InpHYUNkZGkxUFhkUFdNdWtTWnZSb1J0cFZNc3p3bXVhU21TV2dIUkF
qZjNmWEVrdW1NZzJ2V0RlakY4M3FuQnFidm1EZTRCMUZ0VFE4b21IOENjQzQyTU0ifX1d~
```

A.4 What the verifier actually does

Plain credential. The machine follows these steps [8, 9]:

1. Parse the JSON and enforce the local profile: expected VCDM context and types, issuer, proof suite and purpose, subject fields, value range, unit and validity time. Load only the pinned contexts and issuer record. Check that the verification method is authorised for the issuer's `assertionMethod`.
2. Separate the unsigned credential from its proof. Prepare the proof configuration without `proofValue`, using the document context. Expand and canonicalise the credential and the proof configuration with the suite's RDF Dataset Canonicalization procedure. Reject processing that would lose undefined claims.
3. Hash both canonical byte strings with SHA-256. Concatenate the proof-configuration hash first and the credential hash second. Decode the multibase signature and public key and verify the Ed25519 signature over that concatenation.
4. Only after the required checks pass, expose the verified claim to the application. Preserve the source credential or its losslessly recoverable dataset, proof configuration and source identity if it will be checked again from a graph store. A merged set of triples with those boundaries erased is not, by itself, sufficient verification evidence.

Nested credential. The machine must first perform the following outer steps [3]:

1. Split the combined serialization, retaining the original JWT and disclosure bytes. Verify the ES256 signature with the pinned outer public key. Enforce the allowed algorithm, protected `typ` and key identifier, issuer, `vct`, issuance time and disclosure-hash algorithm.
2. Hash the original base64url disclosure string with SHA-256 and match its digest to the signed commitment. Do not hash a decoded and reserialised substitute. Reject uncommitted or duplicate disclosures and names that collide with clear claims.
3. Decode the authenticated disclosure and reconstruct its `credential` member. Enforce the profile's required disclosure and inner/outer issuer binding. Decoding alone authenticates nothing; a valid outer signature does not establish the inner proof's validity.
4. Apply every plain-credential step above to the recovered signed credential. Both layers must pass before this profile accepts it. If it is stored as evidence, retain the original envelope/disclosure and their binding to the verified inner source whenever later policy relies on the outer assertion.

A.5 Extra work, tested outcomes and limits

For this one-credential, one-disclosure example, the comparison is concrete. Both paths verify the same inner proof. The nested path adds an outer signature check, a disclosure hash/match, reconstruction and binding checks. The counts below describe these artifacts; they are not elapsed-time measurements.

Work or representation	Plain credential	Nested credential
Signature verification	One Ed25519	One ES256 plus one Ed25519
RDF canonicalisation	Two suite inputs	Same two inner inputs
Explicit SHA-256 work	Two inner hashes	Two inner hashes plus one disclosure hash
Disclosure reconstruction	None	One whole-credential disclosure
Serialized bytes	797 compact JSON	1534 combined SD-JWT

The byte counts use minified UTF-8 JSON for the plain credential and the exact ASCII combined token, excluding external key and context documents in both cases. The explicit hash count excludes hashing internal to signature algorithms and canonicalisation. The whole-credential disclosure also means that removing an inner attribute generally breaks this Data Integrity proof. A different selective-disclosure suite or an envelope-only design would require a different analysis.

The public test runner verifies both complete examples and rejects changed claims, changed proof values, changed context meanings, loss of an unmapped term, removal of assertion-method authority, altered outer payloads or signatures, changed/uncommitted/duplicate/colliding disclosures, and withholding the required credential. A separately signed negative fixture has a valid outer signature and disclosure commitment but an invalid inner proof: outer-only verification passes and the required combined verification rejects it. This directly demonstrates why unnesting and verifying are different operations.

A second check verifies both signatures directly with Node/OpenSSL, reconstructs the signing input and checks the disclosure commitment; it shares the RDF processing dependency and is therefore an independent signature-path check, not an independent RDF implementation. The public scripts and test reports are in `examples/`. Run `npm ci -ignore-scripts` there once, then `node verify.mjs` and `node audit-spec.mjs`; verification uses local pinned inputs.

These additional stages create integration and testing obligations, not a proof that nesting is inherently insecure. In particular, this two-proof example differs from Architecture B1's envelope-only JSON-LD payload, which has no inner Data Integrity proof to verify. If an outer wrapper serves only a presentation step, policy can check it at ingestion and later verify the retained inner credential directly; the extra outer work is not necessarily repeated on every graph read. Reliance on outer assertions still requires their evidence and binding. The earlier benchmark timings cannot be assigned to this new example. Cryptographic success also does not establish legal qualification, production interoperability, truth of the assertion or AI safety.

References

[1] C. Stöcker, Proofs over Statements in Verifiable Evidence Graphs for the European Business Wallet, reviewed version 1.3, 9 November 2026, CC BY 4.0. Spherity Research repository, https://github.com/spherity/spherity-research

[2] IETF OAuth Working Group, SD-JWT-based Verifiable Digital Credentials (SD-JWT VC), draft-ietf-oauth-sd-jwt-vc-26, 6 November 2026 (Internet-Draft). https://datatracker.ietf.org/doc/draft-ietf-oauth-sd-jwt-vc/26/

[3] IETF RFC 9901, Selective Disclosure for JSON Web Tokens, November 2025. https://www.rfc-editor.org/rfc/rfc9901

[4] ISO/IEC 18013-5:2021, Mobile driving licence (mDL) application.

[5] GS1, EPCIS Standard, Release 2.0, June 2022, Section 10, JSON/JSON-LD bindings and RDF integration. https://ref.gs1.org/standards/epcis/2.0.1/

[6] UNECE, UN Transparency Protocol, Verifiable Credentials profile (work in progress; pre-production pilots), accessed 9 November 2026. https://untp.unece.org/docs/specification/VerifiableCredentials/

[7] Siemens, Rapidminer Graph Studio, open W3C standards and AI-agent context capabilities, accessed 9 November 2026. https://www.siemens.com/en-us/products/rapidminer/graph-studio/

[8] W3C, Verifiable Credential Data Integrity 1.0, Recommendation, 15 May 2025. https://www.w3.org/TR/vc-data-integrity/

[9] W3C, Data Integrity EdDSA Cryptosuites v1.0, Recommendation, 15 May 2025, proof-set and proof-chain test vectors. https://www.w3.org/TR/vc-di-eddsa/

[10] N. Bindel, B. Hale, D. Connolly and F. Driscoll, Hybrid Signature Spectrums, IETF RFC 9955, July 2026 (Informational). https://www.rfc-editor.org/rfc/rfc9955.html

[11] W3C, Verifiable Credentials Data Model v2.0, Recommendation, 15 May 2025. https://www.w3.org/TR/vc-data-model-2.0/

[12] Regulation (EU) No 910/2014 as amended by Regulation (EU) 2024/1183. https://eur-lex.europa.eu/eli/reg/2024/1183/oj

[13] Regulation (EU) 2023/1542 (Batteries). https://eur-lex.europa.eu/eli/reg/2023/1542/oj/eng

[14] U.S. Customs and Border Protection, E-Commerce, web page last modified 6 November 2026, accessed 8 November 2026. https://www.cbp.gov/trade/basic-import-export/e-commerce

[15] GLEIF, vLEI Ecosystem Governance Framework, version 4.0, accessed 8 November 2026. https://www.gleif.org/en/vlei/introducing-the-verifiable-lei-vlei

[16] DHS S&T SVIP, Plug-Fest report, W3C CCG list, 4 March 2021. https://lists.w3.org/Archives/Public/public-credentials/2021Mar/0022.html

[17] Spherity, DHS SVIP Plugfest Demo: E-commerce Digital Twin, video recordings of the DHS S&T SVIP Interoperability Showcase, March 2021. https://vimeo.com/540163632 ; https://vimeo.com/524276878

[18] W3C Credentials Community Group, Traceability Vocabulary v1.0, accessed 8 November 2026. https://w3c-ccg.github.io/traceability-vocab/

[19] U.S. Consumer Product Safety Commission, Third Party Testing Guidance, accessed 8 November 2026. https://www.cpsc.gov/Business--Manufacturing/Testing-Certification/Third-Party-Testing

[20] ETSI TS 119 472-1 V1.2.1 (2026-02). https://www.etsi.org/deliver/etsi_ts/119400_119499/11947201/01.02.01_60/ts_11947201v010201p.pdf

[21] N. Fujie and S. Suzuki, An Empirical Evaluation of Signing and Verification Performance of Verifiable Credential Formats: A Cross-Language Benchmark of SD-JWT VC, W3C VCDM, and mdoc, arXiv:2611.03493v1, 2 November 2026 (preprint). https://arxiv.org/abs/2611.03493v1

[22] N. Fujie and S. Suzuki, A Security Analysis of Verifiable Credential Verification Pipelines: Attack Surface Arising from JSON-LD/RDFC Canonicalization Compared with SD-JWT VC and mdoc, arXiv:2611.04461v1, 3 November 2026 (preprint). https://arxiv.org/abs/2611.04461v1

[23] Commission Implementing Regulation (EU) 2025/1569, consolidated text of 11 August 2026, Arts. 3–4 and Annex II. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:02025R1569-20260811

[24] Commission Implementing Regulation (EU) 2026/1731 of 15 July 2026 as regards applicable standards and specifications. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32026R1731

[25] Commission Implementing Regulation (EU) 2026/1735 of 15 July 2026 amending Implementing Regulation (EU) 2025/1569. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32026R1735

[26] European Commission, Proposal for a Regulation on European Business Wallets, COM(2025) 838. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52025PC0838