Securing the Semantic Supply Chain

Meaning as a Foundation of Trust Infrastructure

Research article

Dr. Carsten Stöcker
Spherity GmbH
carsten.stoecker@spherity.com
ORCID 0009-0003-7417-8872

Version 1.0, 16 November 2026 doi:10.5281/zenodo.24293613



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

Supplements S1–S5 and the open harness v0.4.0 (doi:10.5281/zenodo.24281131) accompany this article.

Abstract

Verifiable credentials are becoming legal and economic infrastructure. With a trusted binding between the signing key and the issuer, a signature authenticates the signed content; it does not by itself fix what that content means to the party that reads it. Meaning reaches a verifier through a supply chain of its own: JSON-LD contexts, vocabularies, ontologies, validation shapes, crosswalks, rulebooks, Type Metadata and codec tables. Each can be substituted, rolled back, frozen, shown differently to different verifiers, or withdrawn without verifiers noticing. W3C Data Integrity already requires approved contexts, or equivalent protections, at each verifier. We show what this leaves open and propose mechanisms to close it.

In laboratory tests with an open harness on production libraries, changing mutable, decision-relevant context bytes that the verifier did not authenticate independently changed a credential's meaning without invalidating signatures over its JSON text (JCS) or over an envelope. On one CBOR-LD library, a synthetic compact label became unreadable after a routine vocabulary update and, under a profile that signs the compressed bytes, decoded to a different meaning with a valid signature. Unauthenticated input imposed seconds of computation before the signature check when canonicalisation had no work limit; the tested canonicalisation library's default limit stops such input but rejects some legitimate credentials. A formal model of the semantic supply chain states the core results with proof sketches and model-checks the stateful mechanisms in 66 cases on small instances.

From these results we derive 30 mechanisms, among them release engineering for meaning, lexical-first proofs that commit to the semantic graph, codec binding for long-lived carriers, receipts that record the meaning a decision applied, attested decision engines whose receipts the proposed profile requires to be attributable to an accountable legal person, and a profile for compliance evidence. We propose 91 requirements as a baseline for a high-assurance profile of W3C VCDM 2.0; no W3C group has adopted them. The problem is not specific to W3C VCDM: SD-JWT VC and ISO mdoc take their meaning from rulebooks and Type Metadata, and we give a first mapping, by design reasoning, of which mechanisms would transfer; none is specified or tested for those formats. The results bind meaning, not its truth; hardware isolation is simulated, not demonstrated, and enrolment for attribution is specified, not implemented.

Keywords. verifiable credentials; semantic integrity; JSON-LD; RDF canonicalisation; Data Integrity; supply-chain security; formal model; European Digital Identity Wallet; digital product passport; attested decisions.

Contents

1 Introduction **5**

1.1 Signed is not the same as meant . . . 5
1.2 Why now . . . 6
1.3 Scope and relation to earlier work . . . 6
1.4 Contributions . . . 7
1.5 Organisation of the article . . . 7

2 Background **8**

2.1 Credentials, JSON-LD and Data Integrity . . . 8
2.2 What the standards already require, and what they leave open . . . 8
2.3 Every credential format has a semantic supply chain . . . 10
2.4 Meaning has layers . . . 11

3 Threat model **12**

3.1 The lifecycle of meaning . . . 12
3.2 Objectives . . . 13
3.3 Actors and adversaries . . . 14
3.4 Attack classes . . . 15

4 Empirical findings **16**

4.1 Harness and method . . . 16
4.2 Five findings . . . 17
4.3 Results per worked example . . . 19
4.4 Historical v0.3.0 evidence and the v0.4.0 implementation . . . 21

5 Formal model and results **22**

5.1 SSC-FM in brief . . . 22
5.2 Results . . . 22
5.3 Model checking . . . 25
5.4 What the model checker is and is not . . . 26

6 Mechanisms **26**

6.1 Produce: meaning as a build output . . . 26
6.2 Release: policy-gated keys . . . 28
6.3 Distribute: release engineering for meaning . . . 28
6.4 Verify: authenticate first, then interpret . . . 29

6.5 Decide and monitor: decisions remember their meaning . . . 29
6.6 Mechanisms added by the use cases . . . 30
6.7 Evidence for compliance decisions . . . 31

7 Use cases **31**
7.1 UC1: battery passport, market-role access and the battery label . . . 32
7.2 UC2: e-commerce shipment at the border . . . 32
7.3 UC3: edge verifier . . . 32
7.4 UC4: AI agent acting on a mandate and an evidence graph . . . 32
7.5 UC5: verifiable knowledge graphs for AI (open problem) . . . 33
7.6 UC6: verifiable business-partner records (KYBP) . . . 33

8 Standards and policy **34**
8.1 Regulation to architecture . . . 34
8.2 Path to a W3C high-assurance profile . . . 35
8.3 Requests to other bodies . . . 35
8.4 The requirements in summary . . . 36

9 Related work **36**
9.1 Our earlier work . . . 36
9.2 Data Integrity and canonicalisation security . . . 36
9.3 Software supply-chain security and transparency . . . 37
9.4 Semantic versioning and ontology governance . . . 37
9.5 Attestation and decision receipts . . . 37
9.6 Compliance evidence and selective disclosure . . . 38
9.7 Agent and AI security . . . 38

10 Limitations and threats to validity **38**
10.1 What this article does not establish . . . 38
10.2 Threats to validity . . . 39

11 Conclusion and research agenda **39**

Statements **40**

References **41**

A Compact claim register **43**

1 Introduction

1.1 Signed is not the same as meant

With a trusted binding between the signing key and the issuer, signature verification authenticates the signed content. It does not by itself fix every external definition used to interpret that content. Meaning reaches the reader of a credential through a supply chain of its own: contexts, vocabularies, ontologies, validation shapes, Type Metadata, codec tables, crosswalks, rulebooks and status-list profiles. Each is published by someone, changes over time, and can be substituted, rolled back, frozen, shown differently to different verifiers, or silently ignored. This holds for every credential format, including SD-JWT VC and ISO mdoc (§2.3).

A short example shows the gap. An energy-sector credential states that a grid operator is a *distribution system operator* with access to *public* Battery Passport data. A battery manufacturer's portal checks the credential and grants access. The credential's JSON names its vocabulary by a URL. If the document behind that URL is changed, for instance by a compromised server, a cache or a careless update, the same signed bytes now say *transmission system operator* and *restricted* data. The signature check still says "valid". Nobody broke a key or revoked anything, yet authority changed (Figure 1).

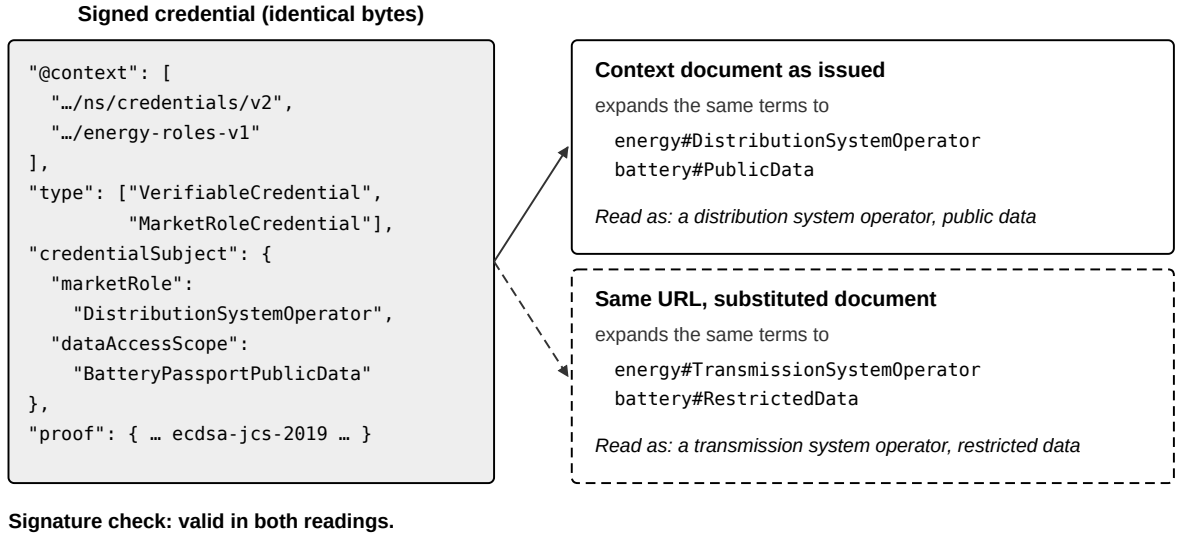


Figure 1: Same signature, two meanings. One credential signed once (JCS or envelope) verifies under the issuer's context as a distribution system operator with public data access, and under a substituted context at the same URL as a transmission system operator with restricted access (WE-1). This requires context bytes the verifier does not authenticate independently; RDFC proofs and checked pins detect it.

This follows from how JSON-LD separates data from its definitions; it was argued by [1] and proved as Lemma L1 in [2]. The result is conditional: it requires mutable, decision-relevant context bytes that the consuming verifier does not authenticate independently, and the demonstrated paths are JCS-based Data Integrity and envelope signatures. We reproduce it on production libraries, on two independent implementations including Digital Bazaar's `ecdsa-jcs-2019`, in one use case (WE-1); it does not imply that deployments with authenticated external context pins are affected. The same class of failure reaches compact product labels: after a routine vocabulary update, a signed CBOR-LD label became unreadable, and under a profile that signs the compressed bytes it decoded into a different document with a valid signature (WE-2). That a label read in 2040 may no longer mean what it meant in 2026 is the consequence we project, not an observation. An attacker without any key can make a verifier that canonicalises without

a work limit spend seconds before the signature is checked; current `rdf-canonize` versions abort by default, and that default in turn rejects some legitimate credentials (F4, F5).

Meaning therefore needs the discipline that software and keys already have: authorised publishers, reproducible builds, signed release, freshness, rollback protection, transparency, withdrawal, attestation and audit. We propose lifecycle controls and test selected implementations; the remaining governance, deployment and assurance obligations are specified as requirements. The results bind meaning, not its truth. Hardware isolation of decision engines is simulated, not demonstrated, and enrolment for attribution is specified, not implemented.

1.2 Why now

- **Credentials are becoming legal and economic infrastructure.** The European Business Wallet proposal, COM(2025) 838, is in first reading with interinstitutional negotiations open: the Council agreed its general approach on 9 June 2026, and Parliament's plenary confirmed the committee's mandate to negotiate on 7 October 2026 (procedure 2025/0358(COD)). Under Regulation (EU) 2023/1542, Article 77, battery passports are required from 18 February 2027 for light means of transport (LMT) batteries, industrial batteries above 2 kWh and electric-vehicle (EV) batteries. The semantic data artefacts of the Commission's digital product passport (DPP) Registry (30 October 2026) contain JSON-LD contexts, RDF/SKOS vocabularies and SHACL shapes.
- **Business-partner records are being piloted as attestations.** WE BUILD, the Large Scale Pilot on business and payment use cases of the EU Digital Identity Wallet, pilots know-your-business-partner due diligence, and its semantics group has published a vocabulary for business records, including legal representatives, powers of attorney, control structure and beneficial owners [3, 4]. In our analysis (UC6), such records would feed onboarding, master data, anti-money-laundering checks and export controls.
- **Meaning is moving into compact, long-lived carriers.** CBOR-LD 1.0 (W3C Working Draft, 9 November 2026) compresses JSON-LD for storage- and bandwidth-constrained uses, and VC Barcodes (Working Draft, 22 August 2026) embeds credentials in optical barcodes such as QR codes. On products, such carriers can stay in the field for 10–20 years (our assumption from product lifetimes; UC1). In the Working Draft of 9 November 2026, CBOR-LD's Security Considerations section is empty.
- **Customs and market surveillance want machine-readable meaning.** A 2026 white paper we co-authored with TU Delft and the PARSEC, CIRPASS-2 and BORDERLINK projects notes that DPP data helps customs only if it is verifiable and available "in truly machine-readable form and with semantic meaning" for automated risk analysis [5].
- **AI agents act on credentials and graphs.** An agent that uses mutable definitions without checking their approved versions may act on changed meaning without detecting the change; this is a conjectured risk, not a measured agent attack.
- **The attack surface has been measured.** Fujie and Suzuki demonstrate context injection, server-side request forgery (SSRF) and poison-graph denial of service against verification pipelines, and leave the JCS meaning gap as future work [1].

Legal and standards status is given as of 14–15 November 2026.

1.3 Scope and relation to earlier work

The article covers every artefact whose content changes how a verifier, wallet, policy engine or agent interprets a credential, a status entry or a graph, without changing its signed bytes. We call these **semantic artefacts**, and the set a decision depended on its **semantic closure**. Out

of scope are comparing credential formats, legal equivalence of meanings across jurisdictions, whether a vocabulary models the world correctly, and post-quantum migration in general.

Proofs over Statements [2] shows what a verifier must hold to trust a statement in an evidence graph. It *assumes* pinned contexts, an authenticated context closure and admitted datasets. This article supplies those assumptions: it shows how the meaning a verifier holds is produced, released, distributed, verified, attested and withdrawn, and where today's stacks fail. [2] proposes a W3C high-assurance profile; a planned companion paper (Paper B) will draft it, with our proposed requirements as its baseline.

1.4 Contributions

- **C1 Threat model of the semantic supply chain:** the lifecycle of meaning in seven stages (govern, produce, release, distribute, verify and use, decide, monitor, including withdrawal), the layers L0–L4 with typed digests, eleven objectives OBJ-1 to OBJ-11 and the attack classes (§2.4, §3).
- **C2 Empirical findings F1–F5** on production libraries, reproducible in an open harness: signed is not meant (F1), undefined signed terms (F2), process-wide context caching (F3), pre-authentication canonicalisation cost (F4), and RDF lists with repeated interior values (F5). WE-1, WE-2, WE-4 and WE-5 and every timing and payload size are results of harness v0.3.0 [6]; WE-3, WE-3 R, WE-6 and WE-7 are results of harness v0.4.0 [7], which also re-runs the earlier examples and whose evidence is functional tests of the corrected paths (§4).
- **C3 A formal model (SSC-FM)** with theorems T1–T19 and T11′ stated with proof sketches, and model checking of the stateful mechanisms in 66 cases on small instances (§5; S3).
- **C4 Thirty mechanisms** across the lifecycle, among them semantic release engineering (M1–M9), codec binding for long-lived carriers (M15), lexical-first, semantic-committed proofs (M17), receipts that record the meaning a decision applied and attested or replay-validated decision engines (M22, M24), and a compliance-evidence profile (M31), each with its evidence level (§6; S1).
- **C5 Ninety-one requirements** (31 new, 40 refined, 20 baseline) as a baseline for a high-assurance profile of W3C VCDM 2.0, with profile classes (§8; S2). No W3C group has adopted them.
- **C6 Scope beyond W3C VCDM:** a first mapping of which mechanisms transfer to SD-JWT VC, ISO mdoc and EUDI attestation rulebooks (design reasoning, E1; §2.3), and a research agenda, including verifiable knowledge graphs for AI (§11).

1.5 Organisation of the article

Section 2 gives the background: what the W3C standards require and leave open, the semantic supply chain of every credential format, and the layers L0–L4. Section 3 sets out the threat model, Section 4 the empirical findings and worked examples, and Section 5 the formal model, theorems and model-checking results. Section 6 presents the mechanisms by lifecycle stage; Section 7 the use cases UC1–UC4 that ground the requirements, the open problem UC5, and UC6, verifiable business-partner records, which applies the requirements. Section 8 covers standards and policy, Section 9 related work, Section 10 limitations and threats to validity, and Section 11 concludes with the research agenda. The appendix gives a compact claim register with evidence levels (Table A1). The supplements carry the full record: mechanism disclosures and claim register (S1), requirements (S2), formal model with proof sketches and model checking (S3), prior-art ledger (S4), and method and review record (S5).

2 Background

2.1 Credentials, JSON-LD and Data Integrity

A W3C VCDM 2.0 credential is JSON whose terms are defined by JSON-LD contexts named by URL [8]; expansion turns it into an RDF dataset, so data and definitions travel separately. RDF Dataset Canonicalization (RDFC) gives an RDF dataset a deterministic form, so that equivalent datasets can be signed and compared; RDFC-based Data Integrity proofs (`ecdsa-rdfc-2019`) sign that form. Canonicalisation can cost more than proportionally to input size when unnamed graph nodes (blank nodes) are connected symmetrically (F4, F5). JSON Canonicalization Scheme (JCS) proofs (`ecdsa-jcs-2019`) sign the canonical JSON text, and JOSE and COSE envelopes (`vc+jwt`) sign the encoded bytes.

2.2 What the standards already require, and what they leave open

The W3C standards already treat context documents as security-relevant.

- **Data Integrity 1.0** [8] fixes its own contexts by SHA-256 digest. For documents secured with Data Integrity, applications MUST run its Context Validation algorithm (DI §4.6), "or one that achieves equivalent protections", after verifying a proof; a context that matches no known good value or cryptographic hash is an error unless the document is recompacted with known contexts (DI §2.4.1, §4.6). Production systems SHOULD permanently cache the context files they use (DI §2.4). Any recoverable loss of data during JSON-LD processing MUST be an error (DI §2.4.3).
- **VCDM 2.0** [8] requires application developers to understand every JSON-LD context their application uses, at least as far as it affects the meaning of the terms used. It allows other specifications to require integrity protection of contexts (VCDM §4.3) and advises stable URLs for contexts and vocabularies (VCDM §4.1).

Approved, stable and versioned contexts, loaded only from controlled copies, stop the most direct attack: a field that looks unchanged but is interpreted differently because the context behind the same URL changed (F1, WE-1).

Other W3C specifications sit on the same path of meaning (Table 1; status of 15 November 2026). Each fixes one part locally; none governs how meaning is released, updated or withdrawn across verifiers. Working Drafts (CBOR-LD, VC Barcodes, Rendering Methods) are listed in S4.

Table 1: Other W3C specifications on the path of meaning.

Specification	Status	What it fixes	What it leaves open
JSON-LD 1.1 and its Processing Algorithms and API	Recommendations, 16 Jul 2020	The processing model: remote and scoped contexts, `@protected` term definitions	Which context versions are approved, and the loader policy

Specification	Status	What it fixes	What it leaves open
RDF Dataset Canonicalization (RDFC-1.0)	Recommendation, 21 May 2024	A canonical form of an RDF dataset; implementations MUST defend against denial of service by raising exceptions and terminating early (§4.4.3)	Specific bounds are left to implementations (F4, F5)
VC-JOSE-COSE	Recommendation since 15 May 2025; current edition 13 Oct 2026	Securing credentials with JOSE and COSE envelopes	The signature covers bytes; the specification defines no context validation of its own and points to JSON-LD's guidance on remote contexts (F1)
VC JSON Schema	Candidate Recommendation Draft, 4 Feb 2025	`credentialSchema`; a schema MAY carry `digestSRI`	"Versioning is not provided as an explicit feature"
Bitstring Status List v1.0	Recommendation, 15 May 2025	Status purposes (`revocation`, `suspension`, `refresh`, `message`) and the mapping of status values to messages	That mapping is itself a semantic artefact, defined by the issuer in each status entry; external status-reference material and its interpretation may still need pinning (M21)
Controlled Identifiers v1.0	Recommendation, 15 May 2025	Verification methods and relationships such as `assertionMethod`	Binding a key to the issuer a credential names is a verifier check (VER-15)
SHACL	Recommendation, 20 Jul 2017	Validation shapes	Shapes are semantic artefacts too; conformance is not truth

Together, these rules protect **one verifier at one moment**. They leave open the questions a supply chain has to answer:

- **Who approves meaning, and how does approval reach every verifier?** The known good values are local configuration. Nothing defines who may publish or change them, how verifiers learn of a new version, a correction or a withdrawal, or how a stale, rolled-back or verifier-specific list is detected.
- **Meaning beyond contexts.** Vocabularies, ontologies, validation shapes, crosswalks, code lists, rulebooks, Type Metadata and codec tables define meaning too (L3). Context validation does not cover them.
- **Implementations.** None of the libraries tested by Fujie and Suzuki applies context validation by default [1, §IV-C2]. With JCS-based or envelope signatures, the signature covers the context URL, not the meaning behind it (WE-1).
- **Time and memory.** A decision cannot later show which meaning it applied. A label read in 2040 must still decode as it was meant in 2026. A withdrawn meaning must stop authorising actions.

The first is ecosystem governance that the standards leave open; the third concerns the tested libraries' defaults, not what the standards require of a conforming implementation. Software met the same questions once integrity checks of single packages proved insufficient, and answered them with provenance, reproducible and signed builds, update frameworks, transparency logs and advisories. Sections 3 and 6 apply that discipline to meaning.

2.3 Every credential format has a semantic supply chain

In every format, a signature covers claim names and values whose meaning is defined elsewhere.

- **SD-JWT VC** [9, 10]. The `vct` type identifier and the claim names are signed. Their meaning is set by Type Metadata and by the rulebook the type belongs to. An integrity hash (`vct#integrity`) can fix the Type Metadata bytes, but not their freshness, rollback, withdrawal, or which version a verifier applied (S4).
- **ISO mdoc** [11]. The document type, namespaces and data-element identifiers are signed. Their meaning is defined in the standard or, for other attestations, in a rulebook.
- **EUDI Wallet attestations** [12–14]. Attestation rulebooks define the structure, semantics and validation of each attestation type. CIR (EU) 2025/1569 requires the Commission to establish and publish a catalogue of attributes (Art. 7) and a sealed catalogue of schemes for the attestation of attributes; a request to include or modify a scheme must state its status and version (Art. 8). We did not confirm what the catalogues hold today.

What is the same. A rulebook, a Type Metadata document or a doctype definition is a semantic artefact (L3), authored, versioned, distributed, cached and replaced outside the signed bytes. The attack classes of §3 that do not depend on JSON-LD processing can affect it too, depending on format and profile: an enforced integrity hash prevents substitution of the pinned bytes, while lifecycle authority, withdrawal and evidence of actual use require additional controls. Which meaning a decision used (S1, M22) is equally open.

Where these formats are harder. Beyond person identification data, rulebooks grow into long prose and schema documents, written per attestation type and changed by people. The same attribute (a birth date, a role, a battery chemistry) appears in many attestation types, each defined by its own rulebook. EUDI TS11 §2.1 already gives each catalogued attribute a URI identifier containing its namespace, local identifier and version, with an optional link to a structured semantic specification. The narrower gap addressed here is a verifier receipt binding the definitions actually used, with governed checks of cross-type equivalence after a rulebook changes. We do not establish the absence of such methods across the literature; M28 is our proposed control. SD-JWT VC and mdoc prescribe no common RDF interpretation; their ecosystems can govern identifiers, publish semantic definitions and automate checks, but cross-type equivalence still needs explicit, governed mappings.

Where these formats are easier. No context is dereferenced during verification, so context substitution through the processor (F1) and canonicalisation cost (F4, F5) do not arise in this form [1]. The attack surface moves to the rulebook, the Type Metadata and the verifier's configuration. Fujie and Suzuki leave the resolution of credential type metadata outside their evaluation [1, §IV-C1]; that resolution, and the lifecycle of the definitions behind it, is what concerns us here. Parts are already addressed: integrity hashes for Type Metadata [10], and a review observing that a stale cached copy can admit a weaker presentation [15].

Why we work on W3C VCDM. Authorisation chains (mandates, roles, delegation) and evidence and knowledge graphs need statements that refer to one another by identifier, and meanings that machines can compose and check. Linked, graph-based data provides this, and with it the processing attack surface studied here. SD-JWT VC supports nested JSON, and either format can carry signed identifiers and relationship values under an application profile; what neither supplies by itself is the shared RDF interpretation, so a verifier must validate the application-defined links and their semantics.

Table 2 is a design-level mapping (E1) for 15 of the 30 mechanisms, not an implemented cross-format result. In it, codec binding (M15) transfers only in part, the RDF-specific M17 and M18

do not apply directly, and M28 appears as a candidate control. M5–M7, M10–M14, M19–M21, M23, M24, M26 and M31 are not assessed, and no mapped mechanism is specified or tested for these formats.

Table 2: Transfer of mechanisms to SD-JWT VC and mdoc (design reasoning, E1).

Mechanism	SD-JWT VC and mdoc
M1–M4 build pipeline, semantic diff, closure manifests, cross-serialisation (Type Metadata is already a build output)	transfer: rulebooks, Type Metadata and doctype definitions become build outputs with a regression corpus
M8, M9 update roles and transparency; M29 trusted lists for publishers of meaning	transfer: to rulebooks, Type Metadata and catalogue entries
M22 semantic-closure digest in receipts	transfers: the rulebook or Type Metadata digest and version in every receipt
M25 withdrawal, M27 time-scoped meaning, M30 advisories	transfer unchanged
M28 governed crosswalks	candidate control for checking cross-type mappings and their changes, using existing governed identifiers where available
M15 codec binding	partly: mdoc is CBOR without context-driven compression; binding the namespace definitions applies
M17, M18 and T2, T9; T10 profile I-RDFC	RDF-specific; do not apply directly. T10 profile B's authenticated deterministic-decoding argument can inform a format-specific binding profile, which this article does not specify

Specifying these mechanisms for SD-JWT VC, ISO mdoc or EUDI rulebooks is outside this article and outside the planned work, which concentrates on W3C VCDM (§11).

2.4 Meaning has layers

A credential is meant at five layers (S3):

- **L0 bytes**, fixed by the transmission;
- **L1 the JSON value**, fixed by the parser profile (an I-JSON/JCS value domain);
- **L2 the RDF dataset**, produced by JSON-LD expansion under the processor's semantics and options, using exactly the resources in its dependency closure;
- **L3 the interpretation** of the IRIs through the vocabulary closure (ontologies, shapes, rulebooks, code lists, crosswalks, selected by time, M27) **and the interpretation engine** with its configuration and built-in rules;
- **L4 the decision** under a policy and a decision context: authority epoch, time, status evidence and the other inputs it reads.

Each securing method protects some layers and not others; much of the confusion about "semantic integrity" disappears once the layer is named. The **dependency closure** of L2 is processing-aware: every resource the processor dereferences (top-level, embedded, type- and property-scoped contexts, recursive `@import`) plus the loader metadata and options that affect the result. It is recorded from the loader's requests under the pinned processor and authenticated by a manifest; a verifier rejects any resolution outside it (VER-16). A closure built from top-level contexts alone misses embedded contexts that change the RDF (S3, X12). For formats without an RDF

layer, L2 is the format's own authenticated claim structure, and the RDF-specific results (T2, T9, T10 I-RDFC) do not apply.

Four domain-separated **typed digests** identify different things (S3): a **content digest** over one artefact's bytes (pinning, archives); a **context-manifest digest** over the URLs and content digests of the dependency closure plus loader metadata and processor (semantic commitments, codec manifests); a **semantic-closure digest** that adds the vocabulary closure and the interpretation engine (receipts, admission, withdrawal tracking; M22, M25); and a **decision-context digest** over every member of the decision context (replay, T14). A context-manifest digest alone does not see the vocabulary closure, so withdrawing a rulebook must be tracked through the semantic-closure digest.

A graph commitment does not authenticate the truth of a vocabulary or an L4 decision (Figure 2).

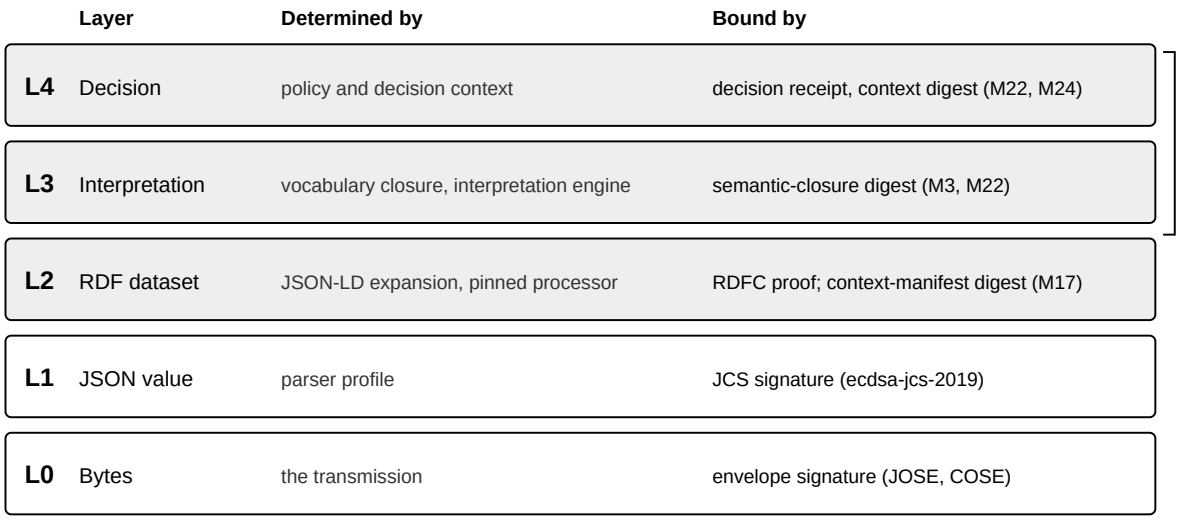


Figure 2: Meaning layers L0–L4 and what binds each: a signature over the bytes (with JCS, also the JSON value), an RDFC proof over the RDF dataset, a semantic-closure digest over the interpretation's artefacts and engine, and a decision receipt over the decision. Binding a lower layer does not by itself bind the layers above it.

3 Threat model

3.1 The lifecycle of meaning

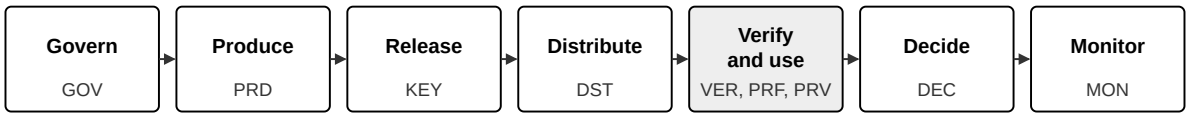
Following the software supply chain, meaning passes through seven stages, from governance to monitoring (Table 3). Each has its own failure modes and its own requirement group (S2). The context validation of §2.2 covers part of one stage, *Verify and use*, at one verifier.

Table 3: Lifecycle stages and requirement groups.

Stage	What happens	Requirement group
Govern	Who may define meaning for a namespace, type or scheme	GOV
Produce	Authoring, building, testing and versioning of contexts, vocabularies, shapes, tables, crosswalks	PRD
Release	Signing keys, signing policy, randomness	KEY

Stage	What happens	Requirement group
Distribute	Registries, mirrors, caches, edge bundles, data carriers	DST
Verify and use	Proof checking, expansion, canonicalisation, interpretation	VER, PRF, PRV
Decide	Verification and decision receipts, agent admission	DEC
Monitor	Drift detection, attestation, advisories, withdrawal	MON

Governance frames every other stage. Withdrawal belongs to the Monitor group, but it ends the life of a meaning and several objectives and theorems turn on it (OBJ-3, OBJ-4, OBJ-11; T4–T6, T12). Figure 3 shows the seven stages with their requirement groups, and the attack classes of §3.4 with the objectives each violates.



Shaded: partly covered by context validation in W3C Data Integrity, at one verifier.

Attack classes and the objectives (OBJ-n) each violates

A	Substitution	1, 2	**J**	Must-understand downgrade	8
B	Uncontrolled resolution	2, 9	**K**	Status lifecycle inference	9
C	Processing exhaustion	7	**L**	Display deception	1, 6
D	Term redefinition	2, 6	**M**	Definition-level poisoning of agents	1, 6, 10
E	Rollback	4	**N**	Publisher key compromise	1
F	Freeze	3	**O**	Resolution tracking	9
G	Split view	5	**P**	Disclosure-induced meaning shift	6, 8
H	Hidden drift	6, 10	**Q**	Codec drift	2, 6, 10
I	Validator tampering	1, 6			

Figure 3: The seven stages of the semantic supply chain with their requirement groups (S2), and the attack classes A–Q with the objectives each violates. The shaded stage, verify and use, is the one that context validation in W3C Data Integrity partly covers, at one verifier.

3.2 Objectives

A high-assurance semantic supply chain must achieve eleven objectives (Table 4).

Table 4: Objectives OBJ-1 to OBJ-11.

#	Objective	Meaning
OBJ-1	Authority	Only a party authorised for the namespace, type and operation, under a currently valid delegation, can publish meaning that verifiers accept
OBJ-2	Integrity	A verifier uses exactly the bytes its publisher released; its RDF graph matches what the issuer signed; and it interprets IRIs only through pinned artefacts
OBJ-3	Freshness	A verifier uses meaning only while an honest status assertion of bounded age says it is current, and otherwise returns *unknown*
OBJ-4	Rollback resistance	A verifier never accepts an older version than one it has accepted, unless the publisher signed a downgrade

#	Objective	Meaning
OBJ-5	Consistency	Accepted log checkpoints lie on one history, and conflicting releases are eventually detected by a monitor
OBJ-6	Completeness	Everything that changes interpretation is inside the pinned closure: imports, parents, shapes, profiles, tables
OBJ-7	Bounded processing	Resolution, canonicalisation and validation finish within declared, deterministic bounds
OBJ-8	Safe interpretation	A verifier that does not understand a meaning-changing feature rejects the input instead of ignoring it
OBJ-9	Privacy	Resolving or updating meaning does not reveal which holder, credential or transaction is processed
OBJ-10	Accountability	An auditor can establish afterwards which meaning a decision used, and replay it
OBJ-11	Conditional availability	Within a declared envelope (admitted inputs, workload, resources, reachable dependencies, retention period), every admitted input gets a definitive result. A verifier that rejects everything meets OBJ-1 to OBJ-10; OBJ-11 rules it out. Nothing is promised while the network is suppressed indefinitely

Pipeline hardening (static loaders, allowlists, call limits) addresses parts of OBJ-2, OBJ-7 and OBJ-9. The other objectives need lifecycle and governance controls, which is where our contributions lie. In the formal model each objective becomes a predicate over executions, so that each requirement can be traced to the theorem that needs it (§5; S3). OBJ-2 thereby splits into publisher bytes, the issuer's graph at L2 and interpretation at L3.

3.3 Actors and adversaries

The model's actors are the **issuer**, whose key is authorised for assertion by the controller named in the credential; the **publisher of a namespace**, with roles in the style of The Update Framework (a threshold root, targets with delegations, snapshot and an online timestamp role); **registries, transparency logs, witnesses and monitors**; an **archive custodian**; the **verifier**, whose state comprises metadata and target high-water marks, a permanent withdrawal set, caches and a clock; the **relying party**; an **edge device** with a measured boot chain, an isolated decision engine, an attestation key and registers; an **agent kernel** with an authority epoch; an **auditor**; and a **network observer** (for OBJ-9).

Adversaries are explicit (Table 5). They range over the network, input authors and malicious issuers, authorised insiders, compromised publisher roles and equivocating log operators, benign drift, root on an edge device with or without isolation of the decision engine, and a network observer.

Table 5: Adversary classes (condensed; S3).

Class	Capabilities	Cannot
A-net	Read, delay, drop or replay any published object; serve any bytes for a URL	Forge signatures; control a trusted clock
A-clock	A-net plus set the verifier's clock	Forge signatures
A-input	Send arbitrary input	Sign as an allowed issuer
A-iss	An allowed issuer that signs arbitrary configurations and commitments	Sign as another issuer

Class	Capabilities	Cannot
A-ins	An **authorised insider** publisher or threshold that releases harmful but authorised meaning	Nothing cryptographic: only review, testing and transparency limit it
A-pub	One compromised publisher role key, for example the online timestamp key	Meet a threshold it does not hold
A-log	An equivocating log or registry operator	Forge honest witnesses' cosignatures
A-drift	None: the same URL serves updated bytes	—
D0	Honest device (may reboot)	—
D1	Root on the device outside the measured boot chain: runs modified code, chooses bundles, reboots, resets resettable registers, extends registers, and invokes any trusted platform module (TPM) command the key policy permits, including using the engine key while the register shows the approved engine	Alter the measured boot chain; break isolation of an isolated engine runtime
O	Network observer of resolution and update traffic	See inside the verifier

Three threats lie outside the model and are stated as limits: a compromised measured boot chain; compromised libraries or compilers, that is, supply-chain compromise of the verifier itself; and hardware side channels. The A-ins row is also a limit of every control proposed here: the mechanisms establish authenticity, freshness and accountability of meaning, but cannot stop an authorised publisher from releasing harmful but authorised meaning.

3.4 Attack classes

Table 6 lists seventeen attack classes, A to Q. Pipeline-level classes (A–C) have been demonstrated against verification pipelines [1]. Rollback, freeze and split view (E–G) are known for software updates and certificates; for meaning we found no earlier treatment of them as attack classes. "New" in this section and in Table 6 is search-relative (searches of 12–13 and 16 November 2026, no systematic patent search; S4). Several classes are shown in our worked examples (§4); others rest on published issues or the literature. We do not claim that the taxonomy is complete.

Table 6: Attack classes, the objectives they violate and the status of the evidence (condensed).

Class	Mechanism	OBJ	Evidence
A Substitution	Another artefact is served for the same identifier	1, 2	[1]; WE-1 B, D
B Uncontrolled resolution	The verifier fetches attacker-chosen URLs (SSRF)	2, 9	[1]; WE-1 I
C Processing exhaustion	Inputs whose cost grows non-linearly, before authentication	7	[1]; WE-4 (new family: symmetric acyclic trees); F5 (some lists with repeated interior values)
D Term redefinition	A local context or `@vocab` rebinds a term	2, 6	[16]; JSON-LD `@protected`
E Rollback	An older, validly signed version is replayed	4	Known for software (TUF); new class for meaning
F Freeze	A verifier is kept on stale meaning	3	Known for software; new class for meaning

Class	Mechanism	OBJ	Evidence
G Split view	A registry shows different versions to different verifiers	5	Known for certificates; new class for meaning
H Hidden drift	Hard-coded meaning diverges across vendors or caches	6, 10	WE-1 H (process-wide cache keyed by URL)
I Validator tampering	Shapes or severities change while the data stays the same	1, 6	SHACL security literature
J Must-understand downgrade	An older verifier ignores a meaning-changing field	8	W3C `vc-bitstring-status-list` issue #218 (masked lists)
K Status lifecycle inference	Published status histories reveal events	9	Revocation series; issue #218
L Display deception	Display or render metadata treated as authoritative	1, 6	VC Render Method Working Draft
M Definition-level poisoning of agents	A definition, not a data instance, changes an agent's interpretation	1, 6, 10	Instance-level KG/RAG poisoning is known; definition-level context poisoning of agents is described conceptually [17]; no general or empirical treatment found (search of 16 November 2026)
N Publisher key compromise	A stolen or quantum-broken publisher key signs bad meaning	1	Threat-level only, in our earlier work on quantum resilience
O Resolution tracking	Lookups of meaning act as usage signals	9	[16]; SD-JWT VC privacy considerations
P Disclosure-induced meaning shift	Withholding a member that carries a type-scoped context changes the expansion of what remains	6, 8	[2] P2
Q Codec drift	A compressed carrier decodes differently, or not at all, after a table or context update	2, 6, 10	WE-2 S2, S3 (new)

Class M is a conjectured attack: M26 states it, with closure digests as the control, at evidence level E1, and no attack is measured (S1, M26). It is partially anticipated by context poisoning of shared agent contexts [17] and content-hash pinning of agent configurations [18] (S4). The classes that do not depend on JSON-LD processing can also affect the rulebooks and Type Metadata of other credential formats, depending on format and profile (§2.3).

4 Empirical findings

4.1 Harness and method

All empirical results come from an open harness released in two versions, kept apart throughout.

- **Harness v0.3.0** [6], Apache-2.0, released 12 November 2026, contains worked examples WE-1 to WE-5. Every timing and payload size in this section is a historical v0.3.0 measurement, from main runs on one 4-vCPU cloud container with Node 22 and the cross-checks stated per finding; two figures come from review experiments, not from archived runs (X5, X11; S3).
- **Harness v0.4.0** [7], Apache-2.0, released 16 November 2026 from commit `2ee3f96` (https://doi.org/10.5281/zenodo.24281131), contains the corrected software paths for M15,

M17 and M18, the M24 receipt protocols under an idealised protected-engine boundary, WE-3's 20 software-TPM scenarios, WE-3 R's 21 replay-receipt scenarios, the WE-6 list matrix and WE-7's 23 synthetic compliance scenarios. From a clean extract, with a software TPM, it passes 136 tests with no skips and runs the 66-case model checker (§5.3) as a test. All four vendored W3C vector files are exercised. The release record identifies the released source archive and its clean-extract verification run.

A measurement of a design later found defective is not evidence for its corrected successor. The v0.3.0 timings therefore do not benchmark v0.4.0, whose evidence consists of functional tests of the corrected paths.

The model, the harness and the manuscript from which this article derives were subjected to seven rounds of adversarial review by AI models, the last by two reviewers in parallel, and to a cross-check of the last round's integration. Findings were checked by the method appropriate to each: the decisive code defects and counterexamples by execution before acceptance, standards and legal statements at the primary source (S5). The reviews found critical defects, including in our own corrections; §4.4 and Table 8 state what was corrected. The article, its supplements and the Executive Brief were reviewed once more before publication, by two AI models working independently (S5.11).

Evidence levels are not interchangeable: E1 design rationale, E2 worked example, E3 executable model, E4 prototype measurement, E5 field trial or independent replication. Where a design was corrected, the level is given per version.

4.2 Five findings

The theory behind F1 to F5 is not new in every case; what is new is the demonstration on production libraries, in a regulated use case, with cross-checks.

F1: signed is not meant. Under `ecdsa-jcs-2019` and `vc+jwt`, substituting the document behind a context URL keeps the signature valid and changes the credential's meaning: in WE-1 B and D, *distribution system operator, public* becomes *transmission system operator, restricted*. We reproduced this on two independent implementations, including Digital Bazaar's `ecdsa-jcs-2019` with `jsonld` 9.0.0. The result is conditional: it arises where verification does not read the contexts (JCS-based Data Integrity, JOSE/COSE envelopes), and the L2 meaning must actually depend on mutable, decision-relevant context bytes that the consuming verifier does not authenticate (T1, §5.2). It does not imply that deployments with authenticated external context pins are affected. RDFC proofs, a checked `relatedResource` digest and `ecdsa-lfsc-2026`, the experimental lexical-first proof suite proposed in §6.4 (M17, abbreviated LFSC), detect the substitution; pinning helps only if it covers every context the processor dereferences, including contexts embedded in the document (T1.1; X2, X12). The theory was argued in [1, §VI-A] and proved as Lemma L1(a) in [2]; ours is a laboratory demonstration in one use case, not an observation in a deployed system. Evidence: E4 and proof (T1).

F2: undefined signed terms. A JCS-signed member that no context defines stays validly signed and is silently dropped from the RDF view (WE-1 G). The signature authenticates the JSON member. It concerns the tested JCS-to-RDF transformation: Data Integrity 1.0 §2.4.3 [8] requires implementations that use JSON-LD processing to raise an error on dropped data, so a conforming consuming transformation must reject the loss. Evidence: E4.

F3: process-wide context caching. `jsonld.js` caches resolved contexts process-wide by URL, so a second loader holding a different copy is never asked (WE-1 H). This is the security framing of a known functional issue (jsonld.js issues #502 and #537, pull request #552, unmerged); the

shared cache applies to loader results tagged `static`. It concerns one library and one cache path; not every per-loader pin is bypassable. Evidence: E4.

F4: pre-authentication canonicalisation cost. Symmetric *acyclic* blank-node trees make RDFC-1.0 canonicalisation super-linear without any key. In the v0.3.0 measurement, a 22 kB tree cost a naive verifier, one canonicalising without a work bound, more than 20 s before any signature check (WE-4; Figure 4); the result was cross-checked on PyLD and `rdf-canonize` 5.0.0. As far as our searches found, this is a new attack family alongside the cyclic and complete graphs of [1]; no general complexity novelty is claimed. The finding is version-qualified: `rdf-canonize` 4.0.0 and later abort such inputs by default, and that default bound has its own cost (F5). Evidence: E4, with versions and bound settings stated.

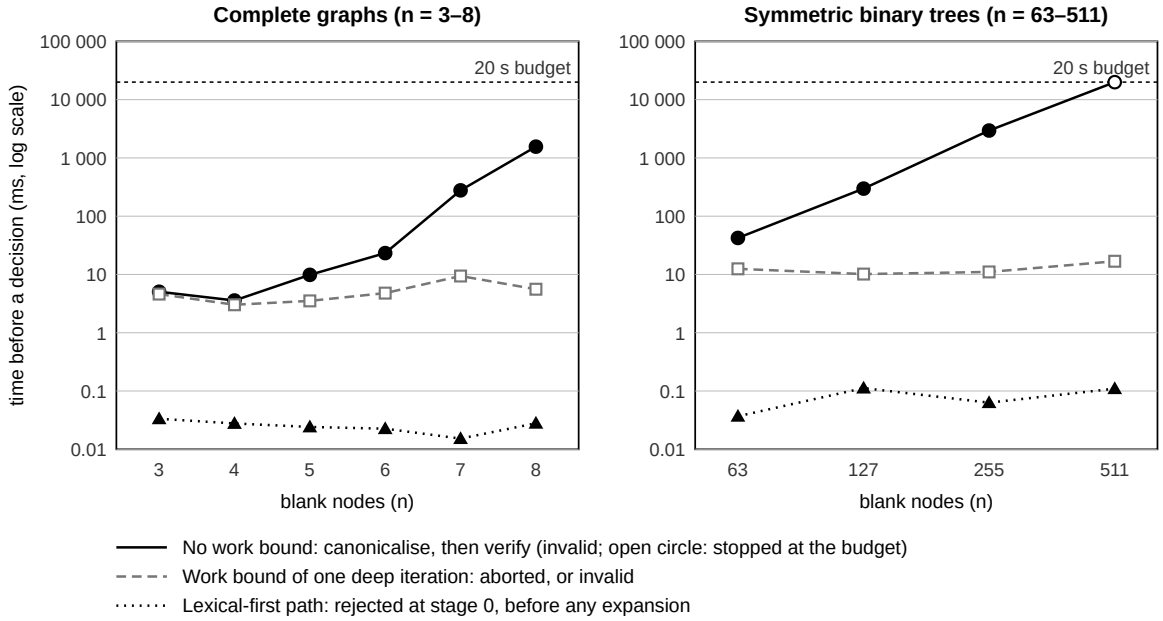


Figure 4: Authenticate-first cost in WE-4, as measured in harness v0.3.0 on one machine: verifier work on the attack documents for a conventional verifier canonicalising without a work bound (1.6 s for a 1.7 kB complete graph, 3.0 s for a 255-node symmetric tree, more than 20 s for a 511-node, 22 kB tree, in the run archived with the release), for a verifier with a work bound of one deep iteration, which aborts or rejects, and for the lexical-first `ecdsa-lfsc-2026` path, which rejects each in 0.02–0.11 ms. These v0.3.0 timings do not benchmark the corrected v0.4.0 implementation.

F5: RDF lists with repeated interior values. Such lists are a canonicalisation hazard in both directions. *Without a bound*, a list of 200 equal values cost 1.1 s in experiment X5 against v0.3.0 (S3) and 0.93 s in the single run archived with v0.4.0. *Under the default bound* of `jsonld` 9.0.0 and `rdf-canonize` 5.0.0, canonicalisation fails for some lists with repeated interior values, such as `[1,2,2,3]`, `[0,0,0,1]` and `[1,2,1,2,1,2]`, while others complete, such as `[1,2,3,1]` and `[0,1,0,1,0]`; on the pinned v0.3 stack (per-node bound 1), lists of up to four equal values complete and five fail. Legitimate credentials containing failing lists cannot be verified with RDFC under default settings. In the tested graphs, what decides is how many list cells share a first-degree hash and whether their neighbours are already distinguished, not repetition as such: at the first hashing round an interior cell sees its neighbours only as `_:z` (T9.3, §5.2). These are tested outcomes for one list family on two stacks, not a general criterion, universal boundary or asymptotic law. Rejection is an availability problem wherever such lists are legitimate; exploitability depends on the use. The finding is partially anticipated: a credential SDK refuses more than 102 identical `@list` values without measuring or explaining the cost [19], and the library default is reported to reject some acyclic inputs and test-suite cases [1, §III-C7]. The

search of 16 November 2026 found no prior report of the list cause or of legitimate credentials failing under the default (S4). Fedify issue #824 reports a production failure with the same error but does not identify the cause; it is a related symptom, not corroboration. Evidence: E4, independently reproduced on both stacks (X14).

4.3 Results per worked example

Table 7 summarises each worked example: the question it answers, the result, its evidence and release, and what it does not show.

Table 7: Worked examples. WE-1, WE-2, WE-4 and WE-5 belong to harness v0.3.0 [6]; WE-3 and WE-3 R as described and WE-7 are v0.4.0 results [7]; WE-6 originates in experiment X5 against v0.3.0 and is archived as a list matrix in v0.4.0. Model checking is reported in §5.3.

WE	Question	Result	Evidence and release	What it does not show
WE-1	Does a valid signature prove what a credential means?	No (F1–F3). Additionally, an attacker's context URL `http://169.254.169.254/...` is requested by naive processing and rejected by the LFSC pre-scan	E4, v0.3.0; two implementations	That deployments with authenticated context pins are affected; any deployed incident
WE-2	Does a CBOR-LD label still mean what was printed after a routine vocabulary update?	A signed label becomes **unreadable**, with an error that does not name the cause. A label signed over its CBOR-LD bytes still verifies but **decodes into a different document** (subject becomes status, scope becomes *restricted*). The M15 envelope rejects before decoding; the lifetime archive restores the meaning. WE-2's RDFC inner proof resists a rewritten envelope; with a byte-level signature (X4) or a JCS inner proof (X8) a rewritten envelope, and with an envelope signature a substituted transitive context (X7), yield *restricted* data under a valid signature. v0.4.0 signs the envelope and pins the transitive closure	E4, v0.3.0; E3 (T10, MC-4); E4, v0.4.0 functional tests	A production lifetime archive; the 2040 reading, which is a projection; the cost of the recommended binding (the measured 23 bytes are one 128-bit configuration; 256-bit bindings and signatures cost more)
WE-3	Can a relying party tell, remotely, which meaning an edge verifier ran?	The v0.3.0 software-TPM receipt protocol held only under its restricted adversary model and is defeated by reset and replay with TPM access (X6). v0.4.0 exercises 20 corrected engine-protocol scenarios, the engine a separate operating-system process, including the counterexamples of a key policy that requires only a platform configuration register (PCR) value and of a host-created key	v0.3.0: E4, defeated; v0.4.0: E4 for the protocol; E3 for execution provenance (T11, MC-3 V5)	Host-root isolation; hardware trust; MON-8 enrolment. The test adversary has no access to authorisation secrets, protected memory or bypass paths

WE	Question	Result	Evidence and release	What it does not show
WE-3 R	Can co-signers that replay a decision validate its receipt without isolation?	21 scenarios exercise replay and acceptance rules; the relying party requires two signatures over identical receipt bytes from two separately generated ECDSA P-256 software keys	E4, v0.4.0, two-signature stand-in; E3 for the outcome property (T11′; MC-3 V6, V6x, V6c; MC-6)	What the device executed; FROST, distributed key generation or nonce security; separate failure domains. The keys share one fixture and are not enrolled under MON-8
WE-4	What can an attacker with no key make a verifier spend?	Naive verifier: 1.6 s (1.7 kB complete graph), 3.0 s (255-node symmetric tree), more than 20 s (511-node tree); LFSC rejects each in 0.02–0.11 ms. A legitimate credential verifies in 1.09 ms (first), 0.25 ms (cached) and 0.18 ms graph-native from stored quads, against 1.94 ms for `ecdsa-rdfc-2019` (Figure 4)	E4, historical v0.3.0 measurement only	Performance of the corrected v0.4.0 verifier; other machines or stacks; that every rejection is cheap (T3 bounds stages 0 and 1)
WE-5	Is a verification decision the same tomorrow, on a busier machine?	Not with time limits: a legitimate credential passes a 1.5× idle-median limit 21 of 21 times idle and 0 of 21 under contention. An operation bound accepts 21 of 21 in both	E4, v0.3.0, one configuration; proof (T8)	That operation counting alone gives determinism (T8 also needs fixed executable and bound semantics)
WE-6 (from X5)	Do lists with repeated elements canonicalise safely?	No (F5)	E4; two stacks; independently reproduced; matrix in v0.4.0	A general criterion; exploitability
WE-7	Can a compliance decision rest on a fixed evidence set, certified and replayed?	23 synthetic scenarios as predicted: fixed declaration fields and source maps, signed complete-list and status responses, an evidence-graph commitment, a signed certificate core, local scope, status and policy checks, deposit and replay by a separately configured instance of the same evaluator	E4, v0.4.0, bounded synthetic example; M31 itself E1 with proof sketches (T19)	Independent provisioning; full issuer and dependency lifecycle; complete context archive; periodic recovery; external effect gate; qualified timestamping; conformity assessment; legal admissibility; truth or completeness of evidence

WE-7 enforces a narrow input domain. Its extraction/v2 commits each signed statement with its container role and Map key in one canonical `sourceEntry` tuple; it accepts only nonempty, well-formed string Map keys whose values are statement objects, and reserves the field name

`sourceEntry`. Preserving every association used by evaluation also requires that the inputs be plain JSON data, a caller premise the harness does not enforce recursively. Older certificates need their original extraction implementation. WE-7 is synthetic, with a toy coverage profile; nothing was deployed.

4.4 Historical v0.3.0 evidence and the v0.4.0 implementation

Experiments X1–X14 against harness v0.3.0 (S3; Table 8) showed where it fell short of the model: pre-scan bypasses admitted blank nodes (M18; X1, X9); another issuer's key was accepted, contexts beyond the top level and contexts embedded in the data went unlisted, a warm cache accepted substituted context bytes, and an oversized `proofValue` cost 528 ms at 60,000 characters in experiment X11 (S3) before the signature check (M17; X2, X10–X13); envelope rewrites gave *restricted* data under valid signatures (M15; X4, X7, X8); and PCR 23 reset and replay was accepted (M24; X6).

Table 8: Evidence per version. The v0.3.0 receipt protocol is defeated (S3, T11).

Mechanism	Harness v0.3.0	Corrected design and v0.4.0
M15 codec binding	E4: drift detected, archive restores meaning (I-RDFC); defeated for byte-signed and JCS profiles and transitive drift	E3 (MC-4); E4 for signed-envelope or inner-RDFC binding, injective tables, pinned transitive contexts and processing profile. No production archive
M17 `ecdsa-lfsc-2026`	E4: detection, cost, graph-native timing; no issuer binding, bounded decoding, full closure or cache contract	E1 and proof sketches (T2, T3); E4 for the tested issuer binding, strict parsing, bounded decoding, closure, cache and graph-native paths. No v0.3.0 timing measures this verifier
M18 admission	E4 for the measured families; lexical scan unsound in general	E1 and proof of sufficiency (T9.2), trusted certifier; E4 for finite tests of `ssc-admission/v1`. No general soundness checker
M24 attested decisions	E4 on a software TPM; defeated by reset and replay	E3 for execution provenance (MC-3 V5) under platform assumptions; E3 for the replay-validated outcome only; E4 for the software protocols. Neither isolation nor threshold signatures demonstrated
M8, M25	Not implemented (M25: relying-party rejection in WE-3 E only)	E3 (MC-1, MC-2, MC-5)
M31 compliance evidence	Not implemented	E1 and proof sketches (T19); E4 for WE-7's synthetic tests

Limits. The main runs used one machine and one JavaScript stack, so every timing is a single-environment result. Model checking is bounded (§5.3); the general statements are the theorems. A software TPM tests the protocol, not hardware trust; hardware validation on a gateway and on a battery management system or charge-point controller is planned. `ecdsa-lfsc-2026` and the M15 envelope are experimental formats, not standards.

5 Formal model and results

We model the semantic supply chain formally (SSC-FM) to check that the findings, requirements and mechanisms are right, not only plausible; the results are proof sketches and bounded model checks. The full model, proof sketches and traceability are in S3.

5.1 SSC-FM in brief

Objects. Byte strings; JSON values in an I-JSON/JCS profile domain $\mathbb{J}$; a context environment Ctx mapping URLs to the bytes a party holds; loader metadata Λ; processor semantics P; a collision-resistant hash; an EUF-CMA-secure signature scheme.

Layers. L0 is the bytes; L1 the JSON value under the parser profile; L2 the RDF dataset produced by expansion under P from exactly the resources in the processing-aware dependency closure dep(j, Ctx, Λ, P); L3 the interpretation through the vocabulary closure V and the interpretation engine E with its configuration; L4 the decision under a policy and a decision context Σ_{ctx}. The closure dep contains every resource the processor dereferences, including contexts embedded anywhere in the document, scoped contexts and `@import`, and is authenticated by a manifest. For formats without an RDF layer, L2 is the format's own authenticated claim structure, and the RDF-specific results (T2, T9, profile I-RDFC of T10) do not apply.

Typed digests. Four digests, domain-separated, identify different things: the *content digest* of one artefact's bytes; the *context-manifest digest* of dep with Λ and P; the *semantic-closure digest*, which adds V and E; and the *decision-context digest* over every member of Σ_{ctx}. A context-manifest digest alone does not see V, so a withdrawn rulebook must advance the epoch of every admission whose semantic-closure digest covers it.

Adversary classes. A-net replays, delays, withholds and serves any bytes for a URL; A-clock also sets the verifier's clock; A-input sends arbitrary input; A-iss is a malicious allowed issuer; A-ins an authorised insider releasing harmful but authorised meaning; A-pub holds one compromised publisher role key; A-log equivocates; A-drift is benign drift. D0 is an honest device; D1 is root outside the measured boot chain, able to reset resettable registers and invoke any TPM command the key policy permits; O observes resolution traffic. A compromised boot chain, compromised libraries or compilers, and hardware side channels are outside the model.

Objectives as predicates. OBJ-1 to OBJ-11 become properties of every execution, traceable to theorems and requirements (S3). A verifier that rejects everything satisfies every safety objective; OBJ-11, conditional availability, excludes it.

5.2 Results

Each result has explicit premises and a proof sketch in S3; none is a mechanised proof, for which S3 gives a plan.

Binding meaning.

Theorem T1 (Signed is not meant). If verification does not depend on Ctx (JCS-based Data Integrity, JOSE/COSE envelopes, SD-JWT) and an accepted document's L2 output actually depends on mutable bytes at some u in dep, then there are context environments differing only at u under which the same signed document is accepted with non-isomorphic meanings; membership of an unused context in a conservative closure does not establish dependence (S3, T1). *Corollary T1.1:* content-digest pins over a set U guarantee L2 agreement against A-net if U covers dep and the verifier uses the same P, Λ and options.

Theorem T2 (Agreement for `ecdsa-lfsc-2026`). If the verifier accepts, its dataset is isomorphic to the one an honest, uncompromised issuer computed, whichever JSON-LD implementation each side uses; against a malicious issuer, to the dataset whose canonical hash that issuer signed, with nothing implied about its truth. Premises: unforgeable signatures, collision-resistant hashing, canonical RDFC, input in $\mathbb{J}$, and a key authorised by the named issuer. T2 binds L2, not L3. *Corollary T2.1:* graph-native re-verification from stored quads, with the verification method checked against the stored, authenticated issuer, covers the whole committed dataset, not single facts.

S3 distinguishes four binding profiles for compact carriers: A, a signature over the CBOR-LD bytes; B, an issuer signature over the envelope; I-RDFC and I-JCS, an inner RDFC (or LFSC) or JCS proof signed before encoding.

Theorem T10 (Codec binding). Under profile B, if the manifest binds every decoding input and the issuer signs the envelope, identical inputs to the deterministic decoder yield identical output. Under profile I-RDFC, a graph-binding inner proof gives L2 agreement when the verifier uses the authenticated graph, even if decoding provenance is incomplete. Neither profile binds L3, guarantees successful decoding, or authenticates the other's material.

Theorem T16 (Parser agreement). One raw-text and parsed-value profile, with the authenticated value handed on immutably, removes the parser differences listed in S3.

Cost and determinism.

Theorem T3 (Authenticate-first cost). Verification has three stages: the pre-scan without JSON-LD processing (stage 0); the issuer-binding, lexical-digest and signature checks (stage 1); and canonicalisation under a bound with comparison against the commitment (stage 2). If every unauthenticated field decoded before the signature check has constant-bounded size and key resolution is local, stages 0 and 1 cost O(n log n), O(n) with radix ordering, under the stated cost model. This does not bound every rejection: valid replays and malicious issuers reach stage 2, which only the operation bound B limits.

Theorem T8 (Deterministic verdicts). For a deterministic verifier with fixed executable and metering semantics, and *unknown* on wall-clock or resource interruption, completed runs on the same input and decision context agree. Operation counting alone does not create determinism.

Theorem T9 (Blank nodes, admission and lists). Blank-node-free input canonicalises in O(m log m) (T9.1). A lexical scan is not sound; admission needs a restricted grammar and a certified closure, with the certifier trusted until a checker is proved sound (T9.2). Repeated interior list values share first-degree hashes, explaining F5 for the tested family only (T9.3).

Theorem T14 (Replay). A deterministic verifier whose decision-relevant Σ_{ctx} is reconstructible from receipt and archive has every definitive verdict reproduced by replay.

Lifecycle and authority.

Theorem T4 (Freshness versus availability). Under indefinite withholding, no verifier both keeps accepting and stops using withdrawn meaning; a trusted clock lets it stop, and with pins only it cannot even stop.

Theorem T5 (Freshness bound). With authenticated status assertions from an honest signer, clock error at most ε and fail-closed use, freshness is bounded by $\Delta_{\text{status}} + \varepsilon$, the lifetime of the honest status-bearing object, not of a timestamp that may wrap it.

Theorem T6 (Rollback). With durable metadata and target high-water marks, a permanent withdrawal set and lower versions accepted only under a current, scoped downgrade authorisation, versions are monotone except at authorised downgrades, and an artefact is never used for current authorisation after the verifier has learned of its withdrawal.

Theorem T7 (Fork exclusion). Under unforgeable signatures and a common witness policy, quorums that intersect in an honest, stateful witness signing only consistent extensions exclude forks. Detecting conflicting releases additionally requires a declared conflict relation and an honest monitor that eventually reads every leaf; detection is after the fact.

Theorem T12 (Withdrawal as an epoch event). With an atomic check of admission, epoch, fresh status and withdrawal at the commit point, and fenced external effects, no stale admission commits after a withdrawal is learned.

Theorem T13 (Extension downgrade). If critical-extension declarations are authenticated and verifiers reject unsupported critical extensions, no verifier accepts while ignoring a declared meaning-changing extension; a closed-world verifier achieves this without marking, at the cost of rejecting harmless extensions.

Theorem T15 (Certification time). Trusted timestamps certify when evidence existed, not when events occurred.

Theorem T17 (Scoped publisher authority). Under authenticated roots, scoped delegations, revocation, durable state, bounded update freshness (T5) and uncompromised thresholds, every accepted release has a valid authorisation derivation for exactly its namespace, type, operation and time: authority, not truth or benign insider behaviour. Revocation takes effect within the bound of T5, not instantly.

Theorem T18 (Non-interference). If the verification cell has no egress and cannot signal into resolution scheduling, a separate fetcher uses only public state on credential-independent schedules, and cache contention, errors and retries add no undeclared dependence, then resolution and update traffic is independent of the credentials processed, up to a declared leakage. Presentation traffic and host side channels are excluded.

Decision attestation.

Theorem T11 (Execution-bound attestation). Suppose an accepted device identity binds a certified key K to an approved isolated decision engine E, and: (i) attestation roots are intact; (ii-a) the relying party has authenticated and pinned K, and accepted evidence the host cannot produce establishes that E exclusively controls K's use, since a TPM key certificate shows K's policy, not who holds its authorisation secret; (iii) runtime isolation protects E's code, state and input snapshot; (iv) E signs exactly its outcome and context digests; (v) receipts carry a fresh challenge, audience and decision identifier; (vi) engine and key are authorised when the receipt is produced. Then an accepted receipt was produced by E over the named context, except with negligible probability. Measuring what was loaded, or a PCR-only key policy, is not enough (S3, T11).

Corollary (governance attribution). If also (ii-b) holds, namely a current enrolment record from an authority recognised under the ecosystem's governance binds K and its key epoch to the device, approved engine and key policy, responsible legal person L and authorised scope S, with L's identity and mandate established independently and the receipt's actual use and relevant time checked under the profile's validity, freshness, compromise and revocation rules, then the receipt is bound to L's enrolled authorisation for that use within S under that governance and compromise and revocation policy. This is an evidentiary attribution claim: it proves neither correctness, nor that L approved this computation, nor legal liability. Exclusive use and attribution do not imply each other.

Theorem T11′ (Replay-validated receipts). With a t-of-n threshold scheme unforgeable against at most $f < t$ statically corrupt participants (the device counting as corrupt), an independently authenticated group key, and honest co-signers that authenticate the request, check admissibility against independently anchored roots and replay the decision before signing

the exact receipt message, an accepted receipt certifies only that its **outcome** equals deterministic evaluation of its committed context and that this context was admissible at signing. It does not establish what the device executed, the truth of device-only observations, or liveness.

Evidence for decisions.

Theorem T19 (Disclosure, sufficiency and the cost of assurance). Exact decision preservation, grant soundness and deny soundness are separate; *unknown* is a third outcome.

- *T19.1 Optionality.* Two admissible views with different decisions defeat exact preservation; b optional units give at most 2^b views.
- *T19.2 Information.* If two complete evidence graphs consistent with the verifier's observation give different decisions, no procedure decides correctly in both; a safe procedure returns *unknown*.
- *T19.3 Cost.* Exhaustive checking of a black-box policy needs up to N evaluations; full disclosure of a sufficient set needs one complete evaluation, and commit-and-certify one at certification plus the required deposit replay. The view count is not a lower bound on assurance cost.
- *T19.4 Opening and replay.* A matching replay shows computation consistency, not true or complete evidence; which snapshot is authoritative needs a separate rule (DEC-15).
- *T19.5 Coverage.* If the sources are sound for their declared responsibilities, coverage is validated under a contract independent of the holder, and the accepted outcome is the correct evaluation of the evidence (the relying party's own evaluation, honest certification, a sound computation proof or a proved sufficient projection), then the accepted decision equals the decision on the policy-relevant authoritative evidence. The lemma is conditional and does not make sources truthful; neither full disclosure nor certification makes incomplete evidence complete.

5.3 Model checking

The checker explores M8, M25, M24 and freshness under a compromised timestamp role exhaustively on small instances; M15 and the premises of replay co-signing are analysed case by case. All 66 cases, each with a stable identifier, come out as predicted [7] (Table 9).

Table 9: Model-checking results: per model, the properties that hold and the variants that violate them (66 cases; S3).

Model	Holds	Violated
MC-1 M8 (12)	corrected design; window $\Delta + \varepsilon$; three redundant removals	pin only; no expiry; stateless; version counter only; adversarial clock; window Δ; mutant
MC-2 M25 (10)	corrected design; two redundant removals	counter and memory removed; no check; check-then-commit; push only; no fail-closed; mutants
MC-3 M24 (14)	honest device; V5 isolated engine against root; V6 threshold key, outcome only	V1–V3; V4 PCR-only key policy; V6x (execution identity); V6c (co-signer compromised)
MC-6 T11′ premises (14)	each premise in place; effects deferred while offline	unauthenticated input; counterfeit status; older status under a current-status property; device-chosen closure; effects on provisional decisions; bounded completion while offline
MC-4 M15 (12)	profile B with full decoding manifest; I-RDFC (assumed from T2)	profile A; B with top-level or transitive manifest; I-JCS

Model	Holds	Violated
MC-5 M8/T5 (4)	honest timestamp key with timestamp bound; compromised timestamp key with status-object expiry checked	compromised key with timestamp-only check or long-lived status object

The complete designs hold on the checked instances, the threshold alternative only for the outcome property it claims, and the designated attack variants fail. For each case reported as holding, the checker also requires that an acceptance, use or commit is reachable, so that no property holds vacuously. **MC6.offline.deferEffect** is the deliberate exception: while the co-signer is offline no acceptance occurs, demonstrating safety through deferral rather than availability.

5.4 What the model checker is and is not

A "holds" covers the instance checked, not every size; the general statements are the theorems. Besides non-vacuity, the checker reports *redundant* rows (single-guard removals that still hold), tightness mutants that must be violated, and a strict check that every case appears exactly once with its expected verdict. Redundant rows show what the instance cannot demonstrate, not that a guard is unnecessary; where we keep such guards, the reason is in the theorems. Instances are bounded, cryptography abstract, time in ticks; MC-1 and MC-2 have one artefact; MC-3 abstracts isolation and key policy as guards; MC-4 and MC-6 are case analyses, MC-4 assuming I-RDFC soundness. T17–T19 have no model check, and T11′ is checked only with the threshold scheme abstracted.

6 Mechanisms

We propose thirty mechanisms, M1 to M31 (M16 is merged into M15), ordered by the lifecycle of §3 (Figure 5). One pattern links them: no key signs, and no device loads, meaning that has not passed an attested, reproducible and logged pipeline, and no decision is made without recording which meaning it used. The controls establish authenticity, freshness and accountability of meaning; they cannot stop an authorised publisher from releasing harmful but authorised meaning, which only review, testing and transparency limit.

Unless stated otherwise, a mechanism is at evidence level E1. E4 evidence for a defeated or incomplete predecessor is not evidence for its corrected successor. "Not found" means not found in our searches of November 2026, which included no systematic patent search (S4). The experimental formats below (`ecdsa-lfsc-2026`, the M15 envelope, the admission certificate, replay receipts, the evidence-graph certificate) are not standards. Full disclosures are in S1.

6.1 Produce: meaning as a build output

M1 builds every published form of meaning from one source model on a hermetic builder, with declared processing bounds as a release gate. M2 classifies each release as patch, minor or major by comparing the RDFC-1.0 expansions and SHACL outcomes of a regression corpus, and blocks a major change under an existing identifier; it detects only changes the corpus exercises. M3 digests the complete closure, including OWL imports, shapes and the interpretation engine, into the semantic-closure digest, distinct from the context-manifest digest. M4 attests corpus-level equivalence across serialisations (test results, not universal equivalence), M5 signs compiled

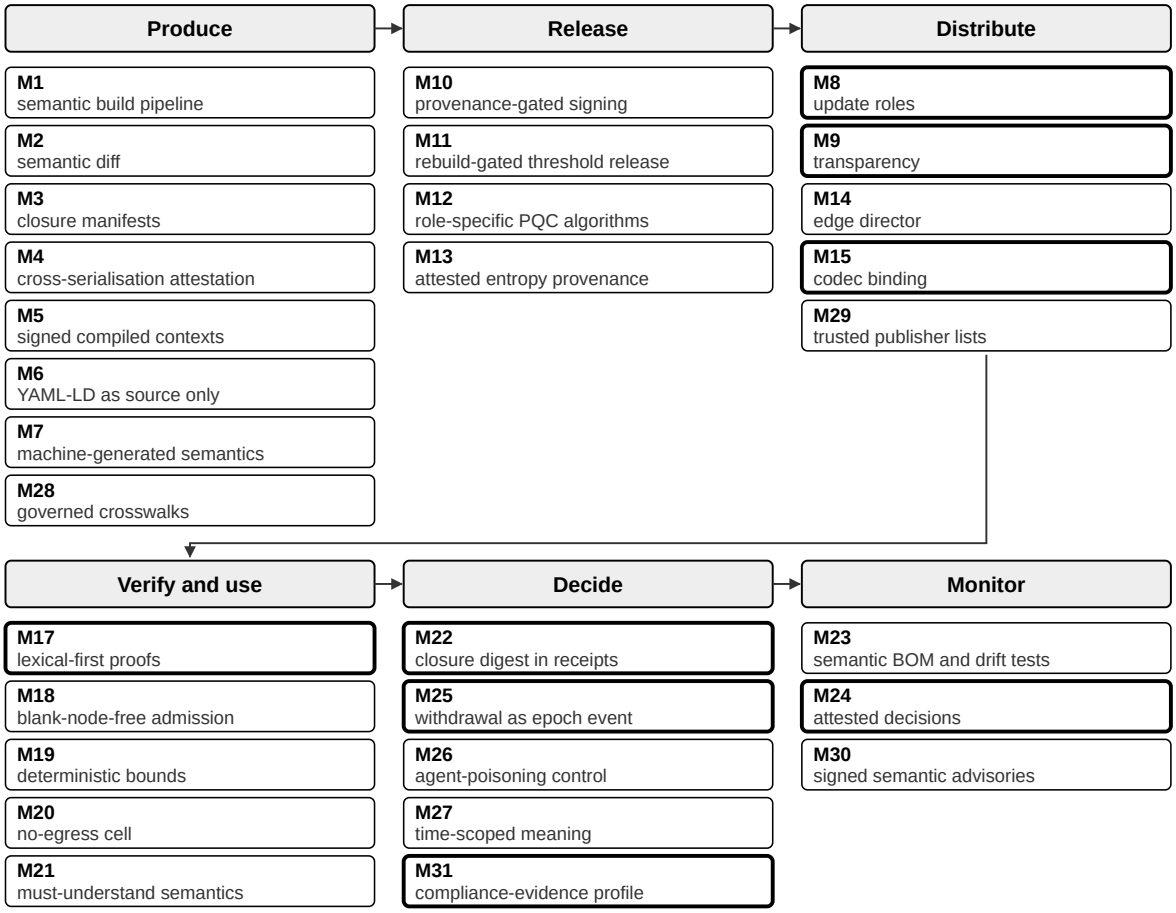


Figure 5: Mechanism map: the thirty mechanisms on the lifecycle stages to which the mechanism disclosures assign them (S1): produce M1–M7 and M28; release M10–M13; distribute M8, M9, M14, M15 and M29; verify and use M17–M21; decide M22, M25, M26, M27 and M31; monitor M23, M24 and M30. The eight treated in detail in this section are marked with a thick border.

contexts for a restricted language, M6 admits YAML-LD as source only, and M7 treats machine-generated semantics as untrusted source (S1, M1–M7).

6.2 Release: policy-gated keys

M10 verifies build provenance, test attestations and the semantic-diff class inside the hardware security module (HSM) or trusted execution environment (TEE) boundary before the key signs. M11 releases threshold key shares only from independently administered builders whose rebuilds match, so that a signature shows at least $t - f$ honest reproductions when at most f participants are corrupt. M12 assigns post-quantum algorithms by role. M13 attests entropy sources and assigns status-list indexes by a keyed permutation; it is incomplete, because a key commitment alone proves neither correct assignment nor absence of bias (S1, M10–M13).

6.3 Distribute: release engineering for meaning

M8, update roles for semantic artefacts. Adapted from TUF and Uptane, M8 defines threshold root, delegation, targets, snapshot and timestamp roles. Verifiers fail closed on expired metadata, keep durable high-water marks for both the metadata and each artefact's version, accept a lower version only under a current, scoped downgrade authorisation, and treat a signed withdrawal as permanent. A withdrawal takes effect within the lifetime of the named status-bearing object plus clock error; a fresh timestamp around an older status object does not refresh it (T5, T6). With a counter on artefact versions only, as in an earlier form of the design, replayed metadata can undo a withdrawal. Indefinite withholding defeats availability (T4). Evidence: E3 for the modelled sub-protocol (T4–T6; MC-1, MC-5); the full TUF delegation structure is not modelled, and some single guards are redundant in the checked instances (S1, M8).

M9, transparency for semantic artefacts. Releases are logged as signed statements following SCITT [20]. Verifiers accept only receipts under consistent checkpoints cosigned by witness quorums that intersect in an honest witness, which excludes forks (T7). Detecting conflicting releases, after the fact, additionally needs a declared conflict relation and a monitor that reads every leaf (DST-13). An eIDAS qualified electronic ledger (Arts. 45k–45l; CIR (EU) 2025/2531, in force since 6 January 2026) can serve as the backend [21]. Evidence: E1 with a proof sketch (T7); the wire-level profile is not yet specified (S1, M9).

M14 lets an edge director fix each device class's bundle and maximum offline window, with hardware rollback counters (not claimed across a factory reset).

M15, codec binding outside the compressed payload, with a lifetime archive. Because CBOR-LD derives term codes from the contexts, a commitment inside the compressed payload would decode with the same wrong codes (WE-2). M15 carries, in an envelope, the digest of a decoding manifest over every decoding input: decoder and processing model, codec semantics, type tables, the context manifests of all contexts used, transitively, and the graph-processing profile (DST-12). The decoder checks the digest before decoding and fails closed; an archive restores old manifests. For profile B, a complete decoding manifest and an issuer signature over the whole envelope bind the inputs to the deterministic decoder. For profile I-RDFC, a graph-binding proof signed before encoding establishes L2 agreement when the verifier uses the authenticated graph, even if decoding provenance is incomplete (T10). Neither profile binds L3 or guarantees successful decoding. Unsigned envelopes under byte-level signatures or JCS inner proofs, and manifests without transitive contexts or decoder semantics are defeated (X4, X7, X8, MC-4). The historical v0.3.0 overhead of 23 bytes is one 128-bit configuration; the recommended 256-bit binding and an envelope signature cost more (PRF-3). Evidence: E3 (T10;

MC-4 case analysis) and E4 for the tested v0.4.0 software paths; no production lifetime archive is demonstrated. It was not found in the CBOR-LD or VC Barcodes drafts (S1, M15).

6.4 Verify: authenticate first, then interpret

M17, lexical-first, semantic-committed proofs (`ecdsa-lfsc-2026`, experimental). The proof signs a configuration that carries a lexical digest of the JCS form and a semantic commitment to the RDFC-1.0 canonical N-Quads, the context manifest of everything the processor dereferenced, and the processor and its options. Verification pre-scans without JSON-LD processing (M18), checks issuer binding, lexical digest and signature, and only then canonicalises under a bound and compares with the commitment; no semantic action is authorised earlier, and a "light" mode that stopped after the signature is withdrawn. Caches return the authenticated graph under a scoped contract (PRF-5). Graph-native re-verification from stored quads covers the whole committed dataset; standalone proofs for single facts are future work. On acceptance, the verifier's RDF dataset is isomorphic to the one an honest issuer computed, or, for a malicious issuer, to what it signed (T2), given a key bound to the named issuer, strict JSON input and use of the authenticated view; pre-signature cost stays quasi-linear only if fields decoded before the signature check are bounded (T3). M17 binds L2, not L3. Evidence: E4 for the v0.3.0 paths, which lacked issuer binding, bounded decoding, the full closure and the cache contract (X2, X10–X13); for the corrected verifier, E1 with proof sketches (T2, T2.1, T3) and E4 for the tested v0.4.0 paths (S1, M17).

M18 admits only input without blank nodes, before expansion. A lexical scan against arbitrary contexts cannot be both simple and sound (T9; X1, X9), so a restricted grammar, a publisher certificate over the closure (PRD-10) and verifier enforcement (VER-4) share the work, and the certifier remains trusted until a checker is proved sound; per-credential HMAC skolem IRIs do not make presentations unlinkable (E1 with T9.2; E4 for finite tests of `ssc-admission/v1`). M19 lets operation bounds decide *invalid* and timeouts yield only *unknown*; replay needs fixed executable and bound semantics, not operation counts alone (E4 for one configuration in WE-5; T8). M20 verifies in a no-egress cell (not built); M21 makes meaning-changing extensions must-understand (T13).

6.5 Decide and monitor: decisions remember their meaning

M22, semantic-closure digest in receipts and admission. Receipts and agent admission carry the semantic-closure digest and a decision-context digest over the reconstructible decision context. With deterministic verification (T8), an auditor can later replay which meaning applied (T14, OBJ-10). Evidence: E2 for partial receipts (WE-3); full decision-context receipts E1 with a proof sketch (T14) (S1, M22).

M24, attested decisions. A relying party needs to know which meaning a remote device applied. Measuring the loaded bundle is not enough: a compromised application can decide in its own code, and a reboot clears the log; nor is a key whose policy only requires a PCR value, which root can invoke (T11; MC-3). The v0.3.0 protocol, which appraised a resettable PCR, is a defeated predecessor (X6).

Execution-bound receipts (T11). A measured, isolated engine that alone can use its non-exportable key decides on a snapshot of input and closure and signs a receipt with a fresh challenge, the outcome and the context digests. The platform must enforce isolation and exclusive key use (a TEE, or an OS-enforced isolated process). Only this design holds against root on the device outside the boot chain (MC-3 V5), whose integrity the model assumes. A TPM key certificate

shows the key's policy, not who holds its authorisation secret (WE-3, the scenario labelled E3 in the harness), so exclusive use needs evidence the host cannot produce.

Replay-validated receipts (T11′). Without a suitable TEE, the key is shared t-of-n between the device and independent co-signers, by MPC or FROST (RFC 9591). Each co-signer authenticates input and context, checks that the context is admissible for this request (authorised closure, current epoch and status, clock tolerance), replays the decision and approves one exact receipt message. T11′ binds the receipt's outcome to its committed, admissible context, not the device's execution, assuming fewer than t compromised share-holders, the device counted among them. The outcome property holds with an honest co-signer (MC-3 V6), the execution property fails (V6x), and both fail with a compromised co-signer (V6c); MC-6 shows what each admissibility check prevents. While co-signers are offline, a protected action is unavailable, not provisionally allowed (DEC-7).

Attribution (MON-8). The high-assurance profile requires attribution of every execution-bound or replay-validated receipt to an accountable legal person (MON-8, Core (if execution-bound or replay-validated receipts)). Before current reliance, an authenticated enrolment record must bind the exact key or group configuration, key epoch, device, approved engine, responsible legal person and authorised scope; technical key origin and legal identity and mandate are validated separately. Enrolment is specified, not implemented by the harness, and this work does not demonstrate an operating authority that issues the specified enrolment records for decision-engine keys. The cited EU instruments (eIDAS 2.0 attestations and wallet-relying-party registration; the European Business Wallet proposal, still in the legislative procedure) are potential building blocks; meanwhile an ecosystem can designate an authority under its own governance. Attribution is evidence for accountability, not of correctness or liability, and a qualified seal on an enrolment record does not make later receipts qualified seals.

Evidence. E3 for execution provenance (T11) under platform assumptions and for the outcome property of T11′. E4 for the v0.4.0 software protocols [7]: 20 WE-3 scenarios on a software TPM under an idealised protected-engine boundary, and 21 WE-3 R scenarios with two ordinary signatures standing in for a threshold key. Isolation from host root, FROST and separate failure domains are not demonstrated (S1, M24; S3, T11, T11′).

M25, withdrawal of meaning as an authority-epoch event. Withdrawal of a meaning, or a detected split view, advances the authority epoch of every mandate or rule whose closure contains it, so that admissions under the old epoch cannot commit. Epoch, freshness and withdrawal are checked atomically at the commit point, external effects are fenced (DEC-7), and push notifications are only hints (T12). Evidence: E4 for relying-party rejection (WE-3 E) and E3 (MC-2); some single guards are redundant in the checked instance. The authority epoch itself comes from our earlier work (S1, M25).

M23 monitors loaded meaning for drift. M30 issues signed semantic advisories that name the required verifier action and feed M25 (E2 for withdrawal behaviour in WE-3 E; advisory format E1; a profile control, not a statutory duty). M26 names a conjectured attack, definition-level poisoning of agents, with closure digests as the control (§7).

6.6 Mechanisms added by the use cases

M27 records which meaning applied at certification time and at decision time, and the selection rule, using trusted time, which shows when evidence existed, not when the event happened (T15). M28 releases crosswalks between vocabularies as governed artefacts. M29 lists authorised publishers of meaning in an eIDAS-style trusted list; it authenticates publishers, not the correctness of meanings (T17).

6.7 Evidence for compliance decisions

M31, a compliance-evidence profile. Compliance decisions, such as customs release, must be reproducible and reopenable in a dispute. Three things can go wrong. First, holder choice: compliance policies are rarely monotone, so if the holder chooses among policy-relevant statements, withholding an authentic adverse one can turn a refusal into a grant; evidence graphs that look alike to the verifier but decide differently cannot be told apart by any computation, which can only return *unknown* (T19.2). This is a limit of information, not of computation. Second, coverage: evidence can be incomplete before anyone sees it: coverage, not the disclosure format, decides whether omitted adverse evidence is caught. Completeness needs a coverage contract the holder does not control: authoritative sources, subject resolution, accepted absence attestations, and *unknown* for missing or stale coverage (T19.5, DEC-9). Third, replay: the evaluation must be complete and replayable under the pinned closure (T14).

M31 fixes a sufficient key-claim set K per purpose and decision in the governing profile, with its coverage contract, and leaves the holder no choice over policy-relevant evidence; selective disclosure remains usable outside K and for proofs of established sufficiency (DEC-8, VER-8). K is either fully disclosed and evaluated once, or committed and certified. Under commit-and-certify, a salted, hiding commitment binds the canonicalised evidence graph, so that public digests cannot reveal it to guessing. A certifier validates sources and coverage (DEC-10), evaluates the policy and signs a core that binds the commitment, closure digest, policy, outcome, scope, coverage reference and retention. The relying party checks authority, scope against the live request, status, closure and policy version (DEC-11), and DEC-15 selects the authoritative snapshot across certificates for a scope and epoch. A custodian acknowledges only a verified deposit, after replaying the decision (DEC-13); invalidating changes, including policy-only corrections, stop current reliance (DEC-12); a dispute replays under the archived closure (DEC-14).

M31 is not justified by a computational lower bound. Checking that every admissible view leads to the same decision can take up to $N \leq 2^b$ evaluations for b optional disclosure units, but that count is not a universal cost of assurance: M31 evaluates one fixed set, plus the deposit replay and any dispute replay (T19.3). Grant soundness with fail-closed *unknown* can suffice for safe authorisation, not for reproducible three-way verdicts; selective disclosure suits grant-monotone checks such as an age threshold (T19.1).

A matching replay shows a consistent computation, not that the evidence was true or complete. Without a sound computation proof, the relying party trusts certification and custodian replay; verifiable computation is neither implemented nor benchmarked here. Stable identifiers are linkable; legal admissibility is not established. Evidence: E1 with proof sketches (T19.1–T19.5); E4 for the synthetic example WE-7, which enforces a restricted input domain, does not demonstrate independent administration or complete archival, and whose older certificates need their original extraction implementation. Novelty lies in the integration (S1, M31; S3, T19; S4).

7 Use cases

Four use cases ground the requirements (S2). They motivate the requirements; they do not validate them, since none was deployed or trialled for this work. A fifth case, UC5, states open problems and adds no requirements. A sixth, UC6, applies the requirements to verifiable business-partner records and adds none.

7.1 UC1: battery passport, market-role access and the battery label

From 18 February 2027, LMT batteries, industrial batteries above 2 kWh and EV batteries placed on the EU market or put into service carry a battery passport, accessible through a QR code on the battery (Regulation (EU) 2023/1542, Art. 77). Parts of the passport are restricted to persons with a legitimate interest. We propose that access be decided with market-role credentials, with which a distribution system operator, a recycler or a market surveillance authority proves its role to a passport provider; the Regulation does not prescribe this architecture. The case stresses authority meaning: a role credential is only as good as the meaning of "distribution system operator" and "restricted data" at the moment of the decision (F1). It stresses long-lived carriers: the label is read for 15 years or more by readers with other software and vocabulary versions, so a compact signed label must decode in 2040 exactly as it was meant in 2026 (WE-2, M15). And a role governance framework for persons of legitimate interest does not yet exist [5]; a draft implementing regulation notified to the WTO on 26 October 2026 (G/TBT/N/EU/1248) defines who these persons are and what they may access, not how they prove their role.

7.2 UC2: e-commerce shipment at the border

Low-value e-commerce consignments arrive in huge numbers: 4.6 billion entered the EU in 2024, almost twice the 2023 number (Commission figures, as cited in [5]). Authorities must decide quickly which to inspect. US Customs and Border Protection has tested verifiable credentials (the Global Interoperability Standards test of August 2023; a one-year test for crude-oil pipeline data from 2 July 2026, subject to change, announced in the Federal Register on 2 June 2026, 91 FR 32985). We found no public evidence of production use.

Some submitters gain from slowing or confusing risk targeting, so pre-authentication denial of service (F4) becomes an attack on border processing. Shipment data use many vocabularies (the W3C CCG traceability vocabulary, the Harmonized System (HS), product-safety rules), and the mappings between them are semantic artefacts with their own supply chain (M28). Meaning changes on a date: HS 2028 enters into force on 1 January 2028 (accepted, as the WCO announced on 21 January 2026), so a code classified under HS 2022 and assessed after that date needs a recorded rule for which meaning applies (M27). Risk-selection models consume the shipment graph, so a changed definition changes what the model sees (attack class M, §3). [2] uses this case for evidence-graph cost and assurance; this article uses it for meaning.

7.3 UC3: edge verifier

Industrial gateways, battery management systems and charge points decide offline or with intermittent connectivity, on constrained hardware, often with a JCS-only stack because RDFC is too costly. The case stresses offline windows for meaning, rollback protection, constrained cryptography, and a question a relying party cannot otherwise answer: which meaning did that device actually apply? WE-3 exercises the receipt protocol with a software TPM under an assumed protected-engine boundary; it does not establish which code ordinary host root executed. Execution provenance needs the further isolated-engine and exclusive-key premises of T11 (§6, M24).

7.4 UC4: AI agent acting on a mandate and an evidence graph

An AI agent acts for an organisation under a mandate credential and reads evidence graphs (test results, certificates, roles) before each action. The case stresses per-action verification cost

and caching, replayable decisions, and authority that must change when meaning is withdrawn. Our earlier revocation work introduced the authority epoch for mandates; this article extends it so that withdrawing a meaning advances the epoch (M25). A companion paper on agent key management is in preparation.

7.5 UC5: verifiable knowledge graphs for AI (open problem)

Organisations increasingly feed AI systems from knowledge graphs that join product, supplier, compliance and safety data under shared ontologies, through graph-based retrieval (GraphRAG [22]) or as training and evaluation data. Unlike UC1–UC4, such graphs are large (millions of statements or more) and semantically rich.

The graph becomes the AI's view of the world: a substituted ontology, a stale mapping or a withdrawn term changes what the AI concludes, and no signature fails; this is F1 at the scale of a whole graph. Whoever selects the retrieved subgraph can shape the answer, the holder-choice problem of T19 in a new form, and one wrong or withdrawn fact can propagate to many answers and inferred facts. Where the high-risk-system provisions apply (§8), the EU AI Act sets data and data-governance requirements for training, validation and testing datasets (Art. 10) and automatic logging requirements (Art. 12). Recording the facts and definitions used for an output can support traceability; it is not by itself evidence that those datasets meet Article 10, nor a complete demonstration of Article 12 compliance.

Open problems remain, for which we propose no mechanism and no requirement: incremental commitments whose cost bounds (F4, F5, M19) hold per update; closures (M22) that include the entailment regime, with inferred facts bound to their premises; proofs for single facts and retrieved subgraphs, with a sufficiency rule, where M17 re-verifies whole datasets only; per-fact issuer binding (VER-15), status and time scope (M27); withdrawal (M25) that reaches every derived fact, cached retrieval and index; and binding embeddings, vector indexes and fine-tuned models to the closure they were built from.

Definition-level poisoning of agents (M26), in which an attacker changes a definition rather than a data instance, is a conjectured attack at E1; none has been measured. It is partially anticipated by conceptual work: context poisoning defended by signed, versioned contexts [17], and content-hash pinning of agent prompts, tools and configurations [18] (S4). The proposed control, closure digests pinned in retrieval and receipts, stops unapproved changes only, not an approved poisoned definition. §11 outlines the research these problems need.

7.6 UC6: verifiable business-partner records (KYBP)

Before a company onboards a customer or a supplier, it must know its business partner (WE BUILD: know your business partner, KYBP, covering know your customer, KYC, know your supplier, KYS, and know your business, KYB): the partner's legal identity and register entries, its legal representatives and their powers, its control structure and its ultimate beneficial owners (UBOs). Today each company assembles these records from registers, documents and questionnaires, and every partner repeats the work. WE BUILD, the Large Scale Pilot on business and payment use cases of the EU Digital Identity Wallet, pilots know-your-business-partner due diligence [3]. Its WP4 Semantics group has published the European Business Wallet Vocabulary v0.1 (2 November 2026, in Turtle and JSON-LD), an RDF vocabulary for electronic attestations of attributes. Its terms include legal representatives and corporate roles, powers of attorney and their scope, equity holders and participations, beneficial owners, including those determined by control exercised in other ways, transparency-register extracts and KYC checks [4].

Issued as verifiable credentials over such a vocabulary, business-partner records could be processed straight through, without re-keying: onboarding, the creation and update of business-partner master data, anti-money-laundering (AML) due diligence and export-control screening would consume the same signed statements. That is why the case is about meaning. The definitions used to interpret representation powers, beneficial ownership and ownership or control shares are part of the semantic closure (L3); signed case-specific facts and the applicable policy are also needed for a decision (L4). If those definitions change without the verifier authenticating the intended version, unchanged signed statements may be interpreted differently (F1), and processing without manual handling can remove an opportunity to notice such a change. The case stresses authority meaning for representation and powers, as in UC1, with issuer binding (VER-15); time scope, because a record certified under one vocabulary version is relied on long afterwards (M27); withdrawal of meaning when a term or definition is deprecated, which stops new decisions under the old meaning (M25, M30); governed mappings between the vocabulary, national registers and other vocabularies (M28); decision receipts that record the closure used, so that an onboarding, AML or export-control decision can be replayed in an audit, given deterministic verification and a reconstructible decision context (M22, T14); and evidence the presenting party cannot select, because a due-diligence decision must not depend on which ownership or control statements that party chooses to disclose (M31, T19). Two limits apply: the end of a power is credential status, not a withdrawal of meaning, and carrying either through to master data already derived from it is not covered by these mechanisms (compare the open problem of derived state in UC5); and recording the closure in master-data records would extend M22. A vocabulary at version 0.1 that changes while a pilot uses it is the situation that release engineering for meaning addresses (M1–M3, M8, M9). We analysed this case on paper only; we neither tested nor deployed it, and it adds no requirements to S2.

8 Standards and policy

8.1 Regulation to architecture

Several EU instruments create obligations that depend on stable, verifiable meaning. Table 10 maps each to the gap we see and to the mechanisms that would address it. The mapping is architectural: role credentials, receipts and advisories named here are our proposals, not statutory mechanisms. Statuses are those of 14–15 November 2026.

Table 10: EU instruments and one US test, the gap in meaning each leaves, and the mechanisms that would address it.

Instrument (status, date)	Gap	Mechanisms
Battery Regulation (EU) 2023/1542, Art. 77: passports for LMT, industrial (> 2 kWh) and EV batteries from 18 Feb 2027	Restricted data relies on role meaning (role credentials are our proposal, not prescribed); labels outlive vocabulary versions	M15, M17, M22
Ecodesign for Sustainable Products Regulation (ESPR) (EU) 2024/1781 (framework; requirements follow in delegated acts); DPP Registry semantic data artefacts (30 Oct 2026)	The published artefacts carry version metadata and a concept-level status, but no freshness or expiry metadata and no signed, artefact-level withdrawal (Registry protocol not audited)	M3, M8, M9

Instrument (status, date)	Gap	Mechanisms
eIDAS as amended; CIR (EU) 2025/1569 Art. 8 (scheme status and version, sealed catalogue; Arts. 6–9 apply from 19 Aug 2026); qualified ledgers, CIR (EU) 2025/2531 (in force since 6 Jan 2026)	Trusted lists cover who may *issue*, not who may *publish meaning*; Art. 8 does not itself specify catalogue-level withdrawal or deprecation transitions, or a maximum catalogue age (catalogue operations not audited)	M8, M9, M29
European Business Wallet proposal COM(2025) 838 (first reading; Council general approach 9 Jun 2026; Parliament's negotiating mandate confirmed 7 Oct 2026)	Legal-effect attestations rely on stable meaning	M22, M27
Cyber Resilience Act (EU) 2024/2847 (reporting from 11 Sep 2026; main obligations from 11 Dec 2027)	Semantic artefacts are undocumented components; no advisory process for meaning	M23, M30
AI Act (EU) 2024/1689 as amended by the AI Omnibus (in force 27 Jul 2026): Art. 12 record-keeping for high-risk systems, from 2 Dec 2027 (Annex III) and 2 Aug 2028 (Annex I)	Logs record inputs, not the meaning applied; closure-digest receipts are our proposed implementation, not text of Art. 12	M22, M25
Customs reform, Regulation (EU) 2026/2108 (adopted 16 Sep 2026; phased implementation); US CBP tests of DIDs and VCs for crude-oil pipeline data (from 2 Jul 2026)	Holder-chosen disclosure of policy-relevant data would make the risk decision depend on the holder's choice	M17, M18, M27, M28, M31

For attributing decision receipts (MON-8), eIDAS 2.0 attestations, wallet-relying-party registration and, if adopted, the European Business Wallet are potential building blocks; none provides enrolment records for decision-engine keys (§6).

8.2 Path to a W3C high-assurance profile

A W3C Community Group for a *High Assurance Profile for W3C VCDM 2.0* was proposed on 11 November 2026 and launched on 12 November 2026. Our requirements (S2) are a proposal to that group; it has not adopted them, nor has any other W3C group. They would feed four documents: *Use Cases and Requirements* (UC1–UC4, the objectives and all 91 requirements); the *Profile* itself (the 82 Core, Core (if ...) and Opt requirements as conformance requirements per class, drafted in a planned companion paper); a *securing-mechanisms note* (`ecdsa-lfsc-2026` and the M15 envelope, both experimental formats, not standards); and a *test suite* (the harness's negative vectors per attack class, W3C vectors and model checker [7]). Open points include the short name, since the group's mission statement uses "HAIP", which is also OpenID's High Assurance Interoperability Profile, and the royalty-free terms of the W3C community agreements, which suit a defensive publication but license every contribution royalty-free.

8.3 Requests to other bodies

These are our proposals, not positions of any body (claim P1):

- **W3C JSON-LD WG:** codec drift, registry substitution (WE-2) and M15's binding pattern in CBOR-LD 1.0's Security Considerations.
- **W3C VCWG:** pre-authentication canonicalisation, benign schema cost and network-dependent verification as denial-of-service threats in the VCDM threat model; mandatory `relatedResource` digests for non-base contexts under JCS or envelope securing.

- **`jsonld.js`, `rdf-canonize` and RDFC-1.0 implementers:** treat the URL-keyed process-wide cache (F3) as a security issue; document that some lists with repeated interior values fail under default work bounds (F5).
- **ETSI ESI:** publisher entries in trusted lists (M29); semantic status in scheme catalogues.
- **Those who define compliance data** (customs, ESPR delegated acts, EBW implementing acts): define per decision the sufficient evidence and how its completeness is established; recognise full disclosure and commit-and-certify; leave the holder no choice over policy-relevant evidence (M31).
- **IETF SCITT:** a statement profile for semantic artefacts (M9).

8.4 The requirements in summary

The 91 requirements (S2) comprise 31 new, 40 refined and 20 baseline requirements in ten groups: governance (GOV, 7), production (PRD, 10), keys (KEY, 6), distribution (DST, 13), verification (VER, 16), decision (DEC, 15), monitoring (MON, 8), interoperability (INT, 4), performance and availability (PRF, 6) and privacy (PRV, 6). "New" is search-relative: not found in the prior-art searches of 12–13 and 16 November 2026 (S4). The requirements are written as testable obligations, most of them single, each owned by an issuer, holder, verifier, publisher, distribution operator or governance authority. The profile split is 29 **Core**; 26 **Core (if ...)**, normative when the named feature is used or claimed; 27 **Opt: module**, binding on implementations that claim the module; 6 **Ref** to other bodies or frameworks (ETSI ESI, IETF, SLSA, TUF); and 3 **Dep**, informative without MUST. Selecting an optional mechanism can bring compulsory conditions: attested engines and replay-validated receipts are optional modules, but their attribution (MON-8) is Core whenever such receipts are used.

9 Related work

We condense the prior-art ledger (S4) by theme and state what this work adds. Novelty is search-relative: "not found" means not found in a web and standards search of 12–13 November 2026 or a focused search of 16 November 2026, without a systematic patent search; it does not mean proven absent.

9.1 Our earlier work

Proofs over Statements [2] shows what a verifier must hold to trust a statement in an evidence graph; its Lemma L1 shows that a context change alters meaning under a valid envelope signature. It *assumes* pinned contexts, an authenticated closure and admitted datasets. This article supplies them: how pinned meaning is produced, released, distributed, verified, attested and withdrawn; demonstrations on production stacks; and a concrete construction for case (iii) of Lemma L3 in [2] (M17). Our working paper [16] lacked an attack model, lifecycle controls and evidence. M25 makes withdrawal of meaning an authority-epoch event, extending our agent-revocation work. Qualified ledgers [21] become a transparency backend (M9).

9.2 Data Integrity and canonicalisation security

Fujie and Suzuki [1] demonstrate context injection, SSRF and poison-graph denial of service, and argue the JCS meaning gap; F1 demonstrates that gap on two implementations, and F4 adds symmetric acyclic trees to their cyclic and complete families. Data Integrity 1.0 already

requires context validation or equivalent protections, and an error on dropped data [8]; F2 demonstrates the latter rule on the JCS path, and F3 frames known `jsonld.js` cache issues as a security issue. The discontinued `vc-jws-2020`, nearest to M17, has no lexical-first ordering or context commitment. F5 is partially anticipated (search of 16 November 2026): a credential SDK's guard refuses more than 102 identical `@list` values without measuring or explaining the cost [19], and the `rdf-canonize` 5.0.0 default is reported to reject some acyclic inputs and test-suite cases [1, §III-C7]. We found no prior report that lists with repeated interior values make RDFC-1.0 super-linear without a bound or make legitimate credentials fail under the default. For compact carriers, CBOR-LD 1.0, VC Barcodes and SUIT manifests are closest to M15, which adds a complete decoding manifest, a pre-decode check and a lifetime archive.

9.3 Software supply-chain security and transparency

Software supply-chain controls precede several mechanisms: SLSA and in-toto (M1 adds semantic gates; M10 checks semantic provenance inside the signing boundary), TUF and Uptane (M8, M14), SCITT [20] and ledger-backed transparency (M9 adds a semantic statement profile with witness quorums and conflict monitoring), client-side rebuild checks (M11 moves them producer-side) and static inventories (M23 adds runtime inventories of loaded meaning). For advisories (M30), the search of 16 November 2026 found term obsoletion (`owl:deprecated`, OBO Foundry Principle 19), signed retraction of assertions (nanopublications) and trust-list status "withdrawn" (ETSI TS 119 612), but no publisher-signed advisory for terms, definitions and rulebooks checked at decision time.

9.4 Semantic versioning and ontology governance

Ontology and schema diff tools (ROBOT, Bubastis, schema registries) underlie M2, which adds instance-corpus expansion and enforcement at release; similar deltas hold for M5, M27 and M28 (S4). For other formats, SD-JWT VC protects Type Metadata by integrity hashes; a review of its draft observed that a stale cached copy can admit a weaker presentation [15]. Fujie and Suzuki treat SD-JWT VC and mdoc as needing no external references, which holds at verification time, but leave type-metadata resolution out of scope [1, §IV-C1]. We found no work that treats Type Metadata, mdoc namespace definitions, rulebooks and catalogues as one semantic supply chain with JSON-LD contexts (search of 16 November 2026).

9.5 Attestation and decision receipts

RATS and EAT, Linux IMA, TPM key policies and TEE-bound keys attest platforms and keys; M24 binds each decision to its semantic closure and decision context. Receipts committing to policy, schema and canonicalisation digests, with offline replay, are prior art [23]; M22 adds the semantic-closure digest as receipt content and admission input. For T11′, execute-endorse (Hyperledger Fabric) and witness cosigning (CoSi) establish independent execution before signing, and the search of 16 November 2026 found threshold-authorised agent actions [24], BFT-replicated threshold-signed access tokens [25] and quorum-signed oracle reports (S4). We found no scheme in which each co-signer replays a credential or policy decision from its committed, admissible context and closure before threshold-signing that exact receipt, nor one that scopes the guarantee to that context rather than the device's execution. For MON-8, agent keys bound to an accountable owner exist [26], but we found no enrolment record binding a receipt key, key epoch, engine version, legal person and decision scope that verifiers must check before reliance.

9.6 Compliance evidence and selective disclosure

SD-JWT [9], BBS and redactable signatures give holder-chosen disclosure; zk-creds, commit-and-prove proofs and verifiable computation prove statements over committed data. The omission problem was raised earlier [27], [20, §9.3], and in-toto layouts and SLSA verification summary attestations bind required evidence and input digests. M31's delta is the integration, not found in the search of 16 November 2026: an independent coverage contract over a decision's sufficient evidence set, a salted certifier commitment bound to the semantic closure and decision context, custodian replay before acknowledgement, and T19's result that coverage, not the disclosure format, decides whether omitted adverse evidence is caught. The decision-preservation lemma is elementary.

9.7 Agent and AI security

Instance- and tool-schema poisoning of knowledge graphs, retrieval and agents is known. The search of 16 November 2026 found conceptual context poisoning of shared agent contexts, defended by signed, versioned contexts [17], and content-hash pinning of agent prompts, tools and configurations [18]; attack class M and M26 are therefore partially anticipated. We found no work that generalises this to vocabularies, ontologies, contexts, shapes and crosswalks, demonstrates it against agents interpreting unchanged signed data, or pins an agent's full semantic closure by digest. M26 remains a conjectured attack (E1).

10 Limitations and threats to validity

10.1 What this article does not establish

- **Truth of meaning.** The results bind meaning, not its truth. A graph commitment authenticates neither the correctness of a vocabulary nor an L4 decision; a well-governed artefact can still model the world wrongly. M17 binds L2 to an honest issuer's computation, or to a malicious issuer's signed commitment, not L3.
- **Completeness.** Neither that the taxonomy of attack classes is complete nor that the 91 requirements suffice for any particular ecosystem.
- **Hardware trust.** Hardware isolation is simulated, not demonstrated. WE-3 runs on a software TPM under an idealised protected-engine boundary; the integrity of the measured boot chain is assumed; replay receipts use two ordinary signatures, not a threshold scheme, and no threshold co-signing service is deployed. Hardware validation on a gateway and on a battery management system or charge-point controller is planned. T11′ binds a receipt's outcome to its committed, admissible context, not what the device executed.
- **Attribution.** Enrolment for attribution is specified, not implemented. MON-8 is a proposal. Enrolment is specified, not implemented by the harness, and this work does not demonstrate an operating authority that issues the specified enrolment records for decision-engine keys. The cited EU instruments are potential building blocks, not engine-key enrolment services. Attribution is evidence for accountability; it does not by itself establish correctness or liability.
- **Compliance evidence.** M31 makes completeness a profile obligation resting on authoritative sources. Without a sound computation proof, the relying party trusts certification and custodian replay; a matching deposit or dispute replay shows a consistent computation, not true or complete evidence. Verifiable computation is neither implemented nor benchmarked, and legal admissibility is not established.

- **Further non-claims.** Presentation unlinkability; private auditability of status-index assignment (M13); standalone proofs for single facts; protection against an authorised insider who releases harmful but authorised meaning; availability beyond a declared envelope (under suppression the profile fails closed); legal recognition of meanings across jurisdictions; and verifiable knowledge graphs at AI scale (UC5).

10.2 Threats to validity

- **Measurement environment.** Measurements come from single environments: the main runs used one machine and one JavaScript stack, and performance on other stacks, hardware or production loads is not established. The v0.3.0 timings are evidence for v0.3.0 only; they do not benchmark the corrected v0.4.0 paths, whose evidence is functional tests (§4).
- **Scope of the findings.** F1 is a laboratory demonstration on production libraries in one use case, not an observation in a deployed system, and does not imply that deployments with authenticated context pins are affected. F2 and F3 concern the tested transformation and cache path; F4 is mitigated by default in `rdf-canonize` $\geq$ 4.0.0; F5 holds for the tested list family and stacks, without a general criterion.
- **Synthetic compliance example.** WE-7 is synthetic, with a toy coverage profile and 23 scenarios. It does not implement independent administration, complete context archival, periodic restore or production escrow, and a legally and operationally adequate coverage profile for a real deployment remains open. The use cases ground the requirements; none was deployed.
- **Formal results.** The theorems are proof sketches with explicit premises; mechanised verification is planned (S3). Model checking covers small instances: 66 cases with abstract cryptography, some of them case analyses rather than state exploration. Some single safeguards are redundant in these instances, so the checker does not show that each is necessary; the offline deferral case shows safety, not availability. T17–T19 have no model check, and T11′ is checked with the threshold scheme abstracted.
- **Experimental formats.** `ecdsa-lfsc-2026`, the M15 envelope, the admission certificate, replay receipts and the evidence-graph certificate are experimental formats, not standards.
- **Other formats.** The mechanism-transfer table (§2) is design reasoning (E1) for 15 of the 30 mechanisms; none is specified or tested for SD-JWT VC, ISO mdoc or EUDI rulebooks.
- **Novelty and review.** Novelty is search-relative (searches of 12–13 and 16 November 2026, no systematic patent search). The adversarial reviews (§4; S5) were carried out by AI models and are not peer review.

11 Conclusion and research agenda

With a trusted binding between the signing key and the issuer, signature verification authenticates the signed content; it does not by itself fix every external definition used to interpret that content. Meaning reaches a verifier through a supply chain of its own, and that chain needs the discipline that software and keys already have: authorised publishers, signed release, freshness, rollback protection, transparency, withdrawal, attestation and audit. We modelled the chain and its threats (C1), demonstrated five findings on production libraries in an open harness (C2), stated the core results with proof sketches and bounded model checks (C3), and proposed thirty mechanisms (C4) and ninety-one requirements as a baseline for a high-assurance profile of W3C VCDM 2.0 (C5); no W3C group has adopted them. The problem is not specific to W3C VCDM, and a first mapping by design reasoning indicates which mechanisms would transfer to other formats (C6). The results bind meaning, not its truth; hardware isolation is simulated, not demonstrated, and enrolment for attribution is specified, not implemented.

The following work is planned. It would build on the formal model, the requirements and the harness of this article; none of it is part of this article's claims.

1. **A high-assurance profile for W3C VCDM 2.0 (Paper B).** Proposed to the Community Group, it would turn the requirements (S2) into conformance requirements per class. Its anchor would be the credential's RDF dataset: every serialisation the profile admits must bind to that one dataset, with JSON-LD as the form that is exchanged and verified, CBOR-LD for compact, long-lived carriers (M15), and YAML-LD as an authoring source only (M6), and with corpus-level equivalence across serialisations attested at release (M4).
2. **Verifiable knowledge graphs for AI (UC5).** It would define a verifiable knowledge graph: a committed graph with provenance per fact, a pinned semantic closure that includes the entailment regime, and versioned updates. It would make commitments incremental, with proofs for single facts and retrieved subgraphs and cost bounds per update; bind inferred facts to the reasoner, profile, ontology closure and premises that produced them; introduce retrieval receipts that extend M22 and T11′; extend withdrawal (M25) to derived facts, indexes, caches and embeddings; and evaluate all of this on a realistic graph of product passports and supply-chain evidence.
3. **Agent key management (UC4),** in preparation.

Statements

Data and code availability

The harness, its tests, the model checker and all recorded results are released as harness v0.4.0 under Apache-2.0 [7]: GitHub release `ssc-harness-v0.4.0` and Zenodo, https://doi.org/10.5281/zenodo.24281131. The historical timings and payload sizes reported in §4 are archived with harness v0.3.0 [6], https://doi.org/10.5281/zenodo.24168347; they are not the timing tables of v0.4.0. The supplements S1–S5 are part of this publication.

Publication and patents

This work is published as a defensive publication; no patent has been or will be filed on the mechanisms it describes.

Competing interests

The author is chief executive of Spherity GmbH, which works on verifiable credentials and digital product passports. The research was carried out at Spherity. No product of the company is described or evaluated. The author supported the creation of the W3C Community Group named in §8.2.

Use of AI tools

The author used AI tools in three ways. Drafting and editing: text, tables and the write-up of the formal model were drafted and revised with Claude (Anthropic) under the author's direction. Software: the open harness, its tests and the model checker were written with Claude Code (Anthropic) and reviewed by the author. Review: the formal model, the harness and the manuscript from which this article derives went through seven rounds of adversarial review

by AI models, ChatGPT (OpenAI) and Claude (Anthropic), the last round by two reviewers working independently, followed by a cross-check of its integration. Findings were checked by the method appropriate to each: the decisive code defects and counterexamples by execution before acceptance, standards and legal statements at the primary source (S5). The article, its supplements and the Executive Brief were then reviewed before publication by two AI models working independently (S5.11). AI tools did not decide what the article claims. The author checked every claim, source and result and is responsible for the content. No AI tool is an author.

Acknowledgements

The author thanks Ronald Koenig (Spherity), Mikael af Hällström (Finnish Tax Administration) and the entire WE BUILD WP4 Semantics group for their work in WE BUILD, including the European Business Wallet Vocabulary; Manu Sporny (Digital Bazaar) for his leadership in W3C on JSON-LD and the Verifiable Credentials Data Model, and for our personal exchanges; and all members of the W3C Verifiable Credentials Working Group. Those named have not reviewed this work, and the acknowledgement does not imply their endorsement. The views expressed, and any errors, are the author's own.

References

[1] N. Fujie, 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:2611.04461, v2, 8 November 2026.

[2] C. Stöcker, *Proofs over Statements in Verifiable Evidence Graphs for the European Business Wallet*, v1.3, Spherity Research, 9 November 2026. https://spherity.github.io/spherity-research/Stoecker-2026-Proofs-over-Statements-v1.3.pdf

[3] WE BUILD Consortium, *Wallet Ecosystem for Business and payments Use cases on Identification, Legal representation and Data sharing*, Large Scale Pilot on business and payment use cases of the EU Digital Identity Wallet; use case BU1, Know Your Business Partner (KYC/KYS/KYB due diligence). https://www.webuildconsortium.eu/ (accessed 16 November 2026)

[4] WE BUILD Consortium, WP4 Semantics group, *European Business Wallet Vocabulary*, v0.1, 2 November 2026 (Turtle and JSON-LD). https://w3id.org/ebwv

[5] B. Rukanova, F. Janssens, K. Chawla, . . . , C. Stöcker, Y.-H. Tan, et al., *Cross-Border E-commerce to Consumers, Digital Product Passports (DPP), and the Role of Authorities*, v1.0, PARSEC, CIRPASS-2, BORDERLINK, March 2026. https://doi.org/10.5281/zenodo.19239072

[6] C. Stöcker, *Semantic supply chain harness: worked examples for securing the meaning of verifiable credentials*, v0.3.0, Apache-2.0, 12 November 2026. https://doi.org/10.5281/zenodo.24168347

[7] C. Stöcker, *Semantic supply chain harness: worked examples for securing the meaning of verifiable credentials*, v0.4.0, Apache-2.0 with third-party notices, 16 November 2026, source commit `2ee3f96`. https://doi.org/10.5281/zenodo.24281131 (concept DOI, all versions: https://doi.org/10.5281/zenodo.24168346); GitHub release https://github.com/cstoecker/spherity-research/releases/tag/ssc-harness-v0.4.0

[8] W3C, *Verifiable Credentials Data Model v2.0*, *Verifiable Credential Data Integrity 1.0* and *Data Integrity ECDSA Cryptosuites v1.0*, Recommendations, 15 May 2025; *RDF Dataset Canonicalization (RDFC-1.0)*, Recommendation, 21 May 2024; *Bitstring Status List v1.0* and *Controlled Identifiers v1.0*, Recommendations, 15 May 2025; *Verifiable Credentials JSON Schema Specification*, Candidate Recommendation Draft, 4 February 2025; *JSON-LD 1.1* and *JSON-LD 1.1 Processing Algorithms and API*, Recommendations, 16 July 2020; *Securing Verifiable Credentials using JOSE and COSE*, Recommendation, 15 May 2025, current edition 13 October 2026; *Shapes Constraint Language (SHACL)*, Recommendation, 20 July 2017; *CBOR-LD 1.0*, Working Draft, 9 November 2026; *Verifiable Credential Barcodes v1.0*, Working Draft, 22 August 2026; *Verifiable Credential Rendering Methods v1.0*, Working Draft, 13 October 2026; *YAML-LD 1.0*, Working Draft, 24 October 2026.

[9] IETF RFC 9901, *Selective Disclosure for JSON Web Tokens* (SD-JWT), Proposed Standard, November 2025.

[10] IETF, *SD-JWT-based Verifiable Digital Credentials (SD-JWT VC)*, Internet-Draft -26, 6 November 2026 (revision -27 followed on 16 November 2026).

[11] ISO/IEC 18013-5:2021 (mobile driving licence application) and ISO/IEC TS 18013-7:2025 (mobile driving licence add-on functions); ISO/IEC 23220-1:2023 and ISO/IEC TS 23220-2:2026 (building blocks for identity management via mobile devices).

[12] Commission Implementing Regulation (EU) 2024/2977 of 28 November 2024 (person identification data and electronic attestations of attributes issued to European Digital Identity Wallets).

[13] European Commission, *EUDI Wallet Architecture and Reference Framework*, v3.0.0, 21 July 2026. https://github.com/eu-digital-identity-wallet/eudi-doc-architecture-and-reference-framework

[14] *EUDI Wallet attestation rulebooks catalogue*, https://github.com/eu-digital-identity-wallet/eudi-doc-attestation-rulebooks-catalog (accessed 15 November 2026).

[15] IETF OAuth Working Group mailing list, *SD-JWT-VC-13 Review Memo*, 19 November 2025. https://mailarchive.ietf.org/arch/msg/oauth/vbb7k7xGsmX7gL6rWIk___Udonww/

[16] C. Stöcker, *The Semantic Supply Chain: Toward a W3C VCDM Assurance Framework*, working paper, 7 May 2026. https://github.com/eu-business-wallet/haip4w3c/blob/main/semantic-supply-chain-analysis.md

[17] C. Fleming, L. Muscariello, V. Pandey, R. Kompella, *A Layered Protocol Architecture for the Internet of Agents*, arXiv:2511.19699, v1 24 November 2025, v3 20 January 2026.

[18] OWASP GenAI Security Project, *OWASP Top 10 for Agentic Applications*, version 2026, December 2025 (ASI04, ASI07). https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/

[19] pilacorp/go-credential-sdk, pull request #165, *fix(processor): bound JSON-LD loading and canonicalization of untrusted input*, merged 5 November 2026. https://github.com/pilacorp/go-credential-sdk/pull/165

[20] IETF RFC 9943, *An Architecture for Trustworthy and Transparent Digital Supply Chains*, Standards Track, June 2026.

[21] I. Alamillo, S. Schwalm, C. Stoecker, R. Thiermann, "Qualified Ledgers – Breakthrough for proven security and legal trust in DLT through eIDAS2 Regulation?", Open Identity Summit 2024, GI. https://doi.org/10.18420/oid2024_19

[22] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, D. Metropolitansky, R. O. Ness, J. Larson, *From Local to Global: A Graph RAG Approach to Query-Focused Summarization*, arXiv:2404.16130, v1 24 April 2024, v2 19 February 2025.

[23] Stone & Ink Archive LLC, *Systems and methods for deterministic conformance suites, validator implementation certification, and verifiable certificate registries for replay-verifiable conformance receipts*, US 12,645,555 B1, granted 2 June 2026 (priority 9 January 2026).

[24] Y. Zheng, Q. Zhang, *From Review to Authorization: Key-Isolated Threshold Signing for LLM Agents*, arXiv:2609.05901, v1 5 September 2026, v2 23 October 2026.

[25] VMware LLC, *Decentralized identity access management using byzantine fault tolerant state machine replication*, US 12,256,007 B2, granted 18 March 2025.

[26] Y. Ni, C. P. Liu, M. Richardson, B. Wang, *WIMSE Applicability for AI Agents*, Internet-Draft draft-ni-wimse-ai-agent-identity-04, 27 October 2026.

[27] W3C Credentials Community Group mailing list, *Re: Selective Disclosure of lists*, 8–9 June 2021. https://lists.w3.org/Archives/Public/public-credentials/2021Jun/0055.html

A Compact claim register

Claim IDs are stable; the full register is in S1. **Type:** Def definition; Emp empirical; Form formal; Arch architecture; Pol policy. **Novelty** is search-relative (searches of 12–13 November 2026, or of 16 November 2026 where marked; no systematic patent search; not proven absent): N novel; NC novel combination; Int integration; Ref refinement; App application; PA partially anticipated; Known demonstration or framing of a known result. **Evidence:** E1 design rationale; E2 worked example; E3 executable model; E4 prototype measurement; E5 field trial or independent replication. Measurements of defeated predecessors (harness v0.3.0; M8, M15, M17, M18 and M24 were corrected) are not evidence for the corrected designs (S1, S3).

Table A1: Compact claim register; the full register is in S1.

ID	Claim	Type	Novelty	Evidence	Does not show
D1	Semantic supply chain: stages, objectives, attack classes A–Q, model SSC-FM	Def, Form	NC	E3 for checked fragments; proof sketches	Complete taxonomy; checks of T17–T19
F1	JCS and envelope signatures leave meaning unbound under context substitution	Emp, Form	Known	E4 + T1	That deployments with authenticated pins are affected; more than one use case
F2	Undefined JCS-signed members vanish from the RDF view	Emp	Known	E4	A defect in conforming transformations, which must reject the loss
F3	A URL-keyed process-wide cache defeats per-loader pins	Emp	Known	E4	Other libraries or cache paths

ID	Claim	Type	Novelty	Evidence	Does not show
F4	Symmetric acyclic trees: super-linear pre-authentication cost	Emp	New family	E4	General complexity novelty; mitigated by default in `rdf-canonize` $\geq$ 4.0.0
F5	Some lists with repeated interior values: super-linear unbounded, rejected under default bounds	Emp	PA (16 Nov)	E4, independently reproduced	A universal criterion
M1	Semantic build pipeline	Arch	App	E1	—
M2	Expansion-level semantic diff	Arch	NC	E1	Changes outside the corpus
M3	Closure manifests and digest	Arch	Ref	E1	—
M4	Attested cross-serialisation equivalence	Arch	NC	E1	Universal equivalence
M5	Signed compiled contexts	Arch	NC	E1	Unrestricted languages
M6	YAML-LD as source only	Arch	App	E1	—
M7	Machine-generated semantics as untrusted source	Arch	App	E1	A statutory duty
M8	Update roles, freshness, high-water marks, withdrawal	Arch	NC	E3 (T4–T6)	Full TUF delegation
M9	Semantic transparency, witness quorums, monitors	Arch	NC	E1 + T7	Fork exclusion without quorum intersection; a wire-level profile
M10	Provenance-gated signing in HSM or TEE	Arch	NC	E1	—
M11	Threshold release after matching rebuilds	Arch	NC	E1	More than $t - f$ honest rebuilds
M12	PQC by role and lifetime	Arch	App	E1	A complete transition
M13	Attested entropy; keyed index permutation	Arch	NC	E1, incomplete	Correct, unbiased assignment
M14	Edge rollback counters, offline window	Arch	NC	E1	Rollback across factory reset
M15	Codec binding over a complete decoding manifest; lifetime archive	Arch	N	E3 (T10); E4 v0.4.0 software	L3; production archive; +23 bytes is one 128-bit configuration
M17	Lexical-first, semantic-committed proofs	Arch	NC	E1 + T2, T3; E4 v0.4.0 software	L3; per-fact proofs; a standard
M18	Blank-node-free admission	Arch	Ref	E1 + T9.2; finite E4	A general checker; unlinkability

ID	Claim	Type	Novelty	Evidence	Does not show
M19	Operation bounds decide *invalid*	Arch	Ref	E4 (one configuration) + T8	Determinism from counts alone
M20	No-egress verification cell	Arch	NC	E1 (not built)	Fetch privacy without T18
M21	Must-understand semantics	Arch	App	E1 + T13	Wire syntax
M22	Closure digests in receipts	Arch	NC	E2 partial receipts; E1 + T14	Replay without reconstructible state
M23	Semantic BOM	Arch	NC	E1	Which meaning a decision used
M24	Isolated engine with exclusive key, or replaying threshold co-signers	Arch	NC	E3 under platform assumptions (T11; T11′ outcome only); E4 simulation	Hardware isolation; threshold cryptography; device execution under T11′; attribution (MON-8 proposed)
M25	Withdrawal advances the authority epoch	Arch	NC	E4 (WE-3 E, relying-party rejection only) + E3 (T12, MC-2)	Safety without fresh status and fenced effects
M26	Definition-level poisoning of agents	Arch	Conjectured; PA (16 Nov)	E1	A measured attack; approved poisoned definitions
M27	Time-scoped meaning	Arch	NC	E1 + T15	Event time
M28	Governed crosswalks	Arch	NC	E1	—
M29	Trusted lists for publishers of meaning	Pol	NC	E1 + T17	Correct meanings
M30	Signed semantic advisories	Arch	NC (16 Nov)	E2 withdrawal; format E1	A specified format
M31	Compliance-evidence profile	Arch, Form	Int (16 Nov)	E1 + T19; E4 synthetic WE-7	True or complete evidence; correct evaluation without trusted certification
P1	Requests to standards bodies	Pol	—	E1	Positions of any body