

# Categorical Semantics for the Civic Exchange Protocol (CEP)

Denise Case<sup>1,2</sup>

<sup>1</sup>Northwest Missouri State University, Computer Science and  
Information Systems, Maryville, MO, USA

<sup>2</sup>Civic Interconnect, Ely, MN, USA

December 10, 2025

**Preprint Notice.** This preprint has not undergone peer review. It is shared to support community discussion, transparency research, and early technical evaluation.

## Abstract

The Civic Exchange Protocol (CEP) defines entities, relationships, exchanges, and context tags as foundational primitives for civic information systems. This paper develops a categorical semantics for CEP, providing a rigorous mathematical basis for its compositional structure, identity guarantees, and interoperability claims. CEP records are modeled as typed objects in a finitely complete category, with morphisms representing provenance-preserving transformations. Envelopes, attestations, and context tags arise as natural transformations between functors encoding system-level views of the data. Canonicalization is formulated as a strict monoidal functor that preserves identity and equivalence classes, while jurisdictional adapters appear as oplax functors mediating between local schemas and global vocabularies.

Together, these constructions clarify CEP’s invariants, formalize correctness properties of its identifier mechanism (SNFEI), and establish a robust foundation for future work in verification, data fusion, and cross-jurisdiction interoperability.

**Keywords:** Civic data; category theory; functorial data modeling; interoperability; canonicalization; identifiers; provenance.

## 1 Introduction

Civic information ecosystems contain heterogeneous data about entities, relationships, and exchanges. These data are typically fragmented across jurisdictions, systems, formats, and organizational boundaries. The Civic Exchange Protocol (CEP) provides a unified, verifiable, schema-based framework for representing such data in an interoperable manner. CEP specifies four record families: entities, relationships, exchanges, and context tags, all wrapped by a shared record envelope with stable identifiers, attestations, and lifecycle metadata.

While CEP has a concrete implementation in JSON Schema, Rust, and Python, its underlying structure is fundamentally compositional. This paper develops a *categorical semantics* for CEP. Our motivation is threefold: (1) to formalize the invariants that CEP relies on for interoperability and identity stability; (2) to express its record structures in a precise mathematical language; and (3) to establish a foundation for evaluating transformations, adapters, and identifier derivations across heterogeneous civic data sources.

We treat CEP canonicalization as a structured rewriting system, which can be viewed categorically as a  $2\bar{\text{category}}$  or polygraph whose  $1\bar{\text{morphisms}}$  are rewrite steps and whose  $2\bar{\text{morphisms}}$  capture coherence between rewrite paths. As in standard rewriting frameworks, not all rewrite rules commute: information-preserving rewrites must precede information-reducing ones. CEP therefore adopts an ordered, stratified rewrite strategy to ensure

determinism and semantic fidelity. This perspective is developed more fully in Section 5.

CEP applies rewriting-theoretic ideas beyond name canonicalization, including adapter chains, entity-merge logic, canonical-string construction for hashing, and vocabulary evolution. We unify several layers of rewriting:

- lexical and orthographic normalization,
- semantic expansion (corporate forms, jurisdiction forms),
- schema alignment and jurisdictional adapter chains,
- provenance-graph normalization,
- vocabulary evolution and version migration,
- entity-merge rewriting with trust and timestamp rules,
- cryptographic canonicalization of canonical strings.

Conceptually, this draws on ideas from term-rewriting theory and related areas, including compiler intermediate representations, natural-language tokenization and normalization, linked-data graph canonicalization, blockchain determinism, and provenance alignment [Baader and Nipkow \[1998\]](#), [Terese \[2003\]](#), [Spivak \[2014\]](#).

Given the heterogeneity and path-dependence of civic data pipelines, the global rewrite system is *not* designed to be:

- commutative or order-independent,
- symmetric,
- context-free.

Instead, CEP is engineered as a *strategy-governed* rewrite system whose evaluation order is explicitly specified, versioned, and enforced. Within this framework, CEP aims for:

- termination (rewriting pipelines always complete),
- determinism (a given input and strategy yield a unique output),

- consistency (strategies respect CEP invariants), and
- documentation (strategies and rules are inspectable and auditable).

This paper makes three main contributions:

1. We formalize the Civic Exchange Protocol as a category **CEP** whose objects are well-typed record states and whose morphisms represent valid, provenance-preserving transformations between them.
2. We introduce *CEP canonicalization* as a structured rewriting system: a stratified, strategy-governed rewriting pipeline that produces deterministic canonical forms suitable for hashing, comparison, and linkage. Categorically, we characterize canonicalization as a monoidal functor that induces stable identifiers (SNFEI) and preserves equivalence classes of civic actors and events.
3. We give a categorical semantics for jurisdictional adapters, vocabularies, and envelopes: adapters are modeled as (op)lax functors mediating between local schema categories and the global CEP category; envelopes and context tags arise as natural transformations between record functors. We demonstrate that this framework extends uniformly across domains (campaign finance, environmental regulation, education), each equipped with its own controlled vocabularies and adapters, showing that a single compositional rewriting framework can support heterogeneous civic data while preserving determinism and provenance.

The goal is not to introduce new implementation mechanisms, but to clarify the mathematical structure implicit in CEP and justify its identity, integrity, and interoperability guarantees. A categorical perspective exposes CEP’s compositionality, makes explicit the conditions under which identifiers and transformations remain stable, and provides a principled basis for future extensions in federation, data fusion, and cross-jurisdiction civic analytics.

## 2 Preliminaries

This section reviews the mathematical tools used throughout the paper and summarizes the structural elements of the Civic Exchange Protocol (CEP). We assume only standard familiarity with category theory, drawing primarily from Mac Lane [Mac Lane \[1971\]](#) and Spivak [Spivak \[2014\]](#) for the treatment of categories as models of data and structure-preserving transformations.

### 2.1 Category-Theoretic Background

We recall only the categorical notions required for the development of CEP semantics:

- **Categories:** collections of objects and morphisms equipped with associative composition and identity arrows.
- **Functors:** structure-preserving mappings between categories, sending objects to objects and morphisms to morphisms in a way that respects identities and composition.
- **Natural transformations:** morphisms between functors, providing coherent comparisons between system-level views of the same underlying data.
- **Monoidal categories:** categories equipped with a tensor product  $\otimes$  and unit object  $I$ , allowing formal reasoning about concatenation, aggregation, and combination of structured data streams.
- **Oplax functors:** functors that preserve monoidal or structural properties up to a controlled relaxation, used here to model schema adapters that preserve meaning even when strict equivalence between jurisdictions cannot be enforced.

These constructions provide a natural language for expressing CEP’s compositional structure: canonicalization becomes a monoidal functor, envelopes become natural transformations, and adapters become oplax mediators be-

tween local and global schema categories.

## 2.2 CEP Structural Recap

CEP defines four record families:

- **Entities:** civic actors (organizations, agencies, districts, individuals).
- **Relationships:** directed or undirected structural links between entities (membership, control, affiliation, reporting lines).
- **Exchanges:** flows of value, information, or action between entities (payments, transfers, notifications).
- **Context tags:** optional, non-canonical annotations that express interpretive, analytic, or contextual facts about a record.

All record families are wrapped in a shared *record envelope* that provides:

- schema and vocabulary references,
- revision numbers and lifecycle status,
- attestation metadata,
- timestamps describing observation and validity intervals,
- stable CEP identifiers (verifiable IDs).

The envelope separates canonical, identity-bearing components of a record from contextual or analytic metadata, ensuring both stability and extensibility.

## 2.3 Canonicalization and Identifiers

For entity records, CEP constructs a canonical string from jurisdiction-normalized components such as name, address, and formation date. A SHA-256 hash of this canonical string yields a stable identifier known as the *Structured Non-Fungible Entity Identifier* (SNFEI).

In this paper, we treat canonicalization as a deterministic, strictly monoidal process: concatenation of components corresponds to a tensor-like operation, and hashing corresponds to an endofunctor that collapses equivalence classes of canonical strings to identity objects. This perspective will be formalized in Section 5.

### 3 The Category CEP

This section formalizes the structural semantics of the Civic Exchange Protocol in categorical terms. We introduce the category **CEP**, whose objects are attested record states and whose morphisms capture provenance-preserving transformations. The purpose is not to replace CEP’s operational semantics, but to express its invariants and data-integration behavior in a compositional language.

#### 3.1 CEP as a unified rewriting framework

We treat CEP as a family of rewriting systems glued together by a common category-theoretic backbone. At the lowest level, canonicalization rewrites raw strings into canonical forms. At the schema level, adapters rewrite source-specific records into CEP entities and relationships. At the graph level, provenance and entity-merge operations rewrite collections of records into coherent, versioned entity graphs. Each layer uses a stratified, strategy-governed set of rewrite rules rather than assuming that all transformations commute.

In this view, adding a new civic domain (e.g., campaign finance, environmental permits, educational programs) means specifying: (i) a domain vocabulary, (ii) a set of rewrite rules for codes, names, and identifiers, and (iii) adapters that connect domain-specific sources into the CEP category. Sections 5 to 7 make these rewriting layers explicit, and Section 9 outlines several domain-specific instantiations.

### 3.2 Objects: Attested Record States

Objects of **CEP** are *well-typed, time-stamped record states*. Each record consists of a payload (entity, relationship, or exchange) together with its envelope, which includes schema references, revision numbers, attestations, timestamps, and stable identifiers.

Formally:

$$\text{Ob}(\mathbf{CEP}) = \{ R \mid R = (\text{Payload}, \text{Envelope}) \text{ is a valid CEP record} \}. \quad (1)$$

Each object  $R$  carries a fixed record kind (entity, relationship, or exchange) and is assumed to satisfy JSON Schema validity with respect to its declared `recordSchemaUri`.

### 3.3 Morphisms: Provenance-Preserving Transformations

Morphisms in **CEP** represent admissible, provenance-preserving transformations between record states. A morphism  $f : R \rightarrow R'$  models a transition such as an amendment, attestation update, relationship creation, or audit step. To qualify as a morphism,  $f$  must preserve CEP’s core invariants:

1. **Schema validity:**  $\text{Schema}(R) \rightarrow \text{Schema}(R')$  must be respected, meaning  $R'$  remains valid under the same or a successor schema.
2. **Revision monotonicity:**  $\text{Revision}(R) \leq \text{Revision}(R')$ . Updates must advance (not rewind) the revision counter.
3. **Identity invariance:** The canonical identifier (SNFEI) associated with the canonical payload of  $R$  must match that of  $R'$ . This ensures that morphisms do not alter identity-bearing fields.

Intuitively, morphisms encode permissible “ways a record can change” while respecting immutability of identity and the integrity constraints of the



envelope.

### 3.4 Finite Limits and Consistent Joins

We assume **CEP** is *finitely complete*, meaning it admits all finite limits. Among these, the most important for data integration is the *pullback*, which formalizes the notion of a consistent join between heterogeneous record fragments.

Given two morphisms  $f : R \rightarrow A$  and  $g : R' \rightarrow A$ , their pullback is an object  $P$  equipped with morphisms  $p_1 : P \rightarrow R$  and  $p_2 : P \rightarrow R'$  satisfying the usual universal property. We interpret  $A$  as a common semantic target (e.g., an entity type), and  $P$  as the maximal set of facts jointly consistent between  $R$  and  $R'$ .

#### The Categorical Pullback (Consistent Join)

A pullback of  $f : R \rightarrow A$  and  $g : R' \rightarrow A$  is an object  $P$  that represents the most specific record state compatible with both  $R$  and  $R'$  when they assert information about the same underlying entity or relationship  $A$ . For example, joining a vendor registry record  $R$  with a contracting record  $R'$  along a shared entity type  $A$  yields  $P$ , the unique maximal consistent integration of both fact sets.

This provides a principled mechanism for data fusion, deduplication, and consistency checking within and across jurisdictions.

### 3.5 Subobjects

Subobjects in **CEP** correspond to *partial, typed views* of records. Examples include:

- extracting only the attestation sequence of a record,
- projecting a relationship record onto one of its participating entities,

- isolating the canonical payload while omitting context tags.

Because subobjects inherit morphisms from the ambient category, substructure analysis, such as reasoning about what remains invariant under updates, can be expressed cleanly and compositionally.

## 4 Envelopes and Attestations as Natural Transformations

CEP record envelopes, including schema references, revision metadata, timestamps, and cryptographic attestations, are not merely auxiliary fields. They impose a structural layer that is functorial: every payload can be given an envelope, and every valid update to a payload induces a corresponding update to its envelope. This section formalizes that layer using functors and natural transformations.

### 4.1 Envelopes as Functors

Let  $\mathbf{P}$  denote the category of *raw payloads*, whose objects are unenveloped CEP payloads (entities, relationships, and exchanges), and whose morphisms are valid payload-level transformations (e.g., updates, amendments, or partial recomputations) that do not yet touch attestation or revision metadata.

Let  $\mathbf{E}$  denote the category of *enveloped records*, in which each object consists of a payload together with its envelope, and morphisms are the provenance-preserving transformations defined in Section 3.

We define the *enveloping functor*

$$\mathcal{E} : \mathbf{P} \rightarrow \mathbf{E} \tag{2}$$

that assigns to each payload  $P$  an enveloped record  $\mathcal{E}(P)$ . The functor:

- injects the schema reference,
- initializes revision and lifecycle fields,
- creates timestamps for observation and validity,
- embeds the canonical (identity-bearing) fields.

On morphisms,  $\mathcal{E}(f)$  augments a payload-level update  $f : P \rightarrow P'$  with corresponding envelope updates that preserve schema validity, revision monotonicity, and identifier invariance.

Thus, the enveloping process is not an ad hoc transformation but a strictly functorial lift from payload space to the full CEP record space.

## 4.2 Attestations as Natural Transformations

Let  $\mathcal{E}'$  be a functor

$$\mathcal{E}' : \mathbf{P} \rightarrow \mathbf{E}$$

representing the construction of *attested envelopes*, where each payload  $P$  receives an envelope that incorporates a cryptographic validation step (digital signature, proof-of-origin, or other attestation mechanism). As with  $\mathcal{E}$ , this functor acts on both objects and morphisms.

An attestation operation is then modeled as a *natural transformation*

$$\alpha : \mathcal{E} \Longrightarrow \mathcal{E}'.$$

For each payload  $P$ , the component

$$\alpha_P : \mathcal{E}(P) \rightarrow \mathcal{E}'(P)$$

corresponds to the application of a cryptographic attestation method (e.g., signing, sealing, notarizing) to the envelope of  $P$ .

Naturality requires that for every payload morphism  $f : P \rightarrow P'$  in  $\mathbf{P}$ , the

following square commutes:

$$\begin{array}{ccc}
 \mathcal{E}(P) & \xrightarrow{\mathcal{E}(f)} & \mathcal{E}(P') \\
 \alpha_P \downarrow & & \downarrow \alpha_{P'} \\
 \mathcal{E}'(P) & \xrightarrow{\mathcal{E}'(f)} & \mathcal{E}'(P')
 \end{array}$$

#### Attestations as Natural Transformations (Provenance Commutativity)

The naturality of  $\alpha$  ensures that the act of attesting ( $\alpha_P$ ) commutes with the act of transforming the payload ( $\mathcal{E}(f)$ ). This formalizes the requirement that provenance chains remain consistent: an attestation applied before or after an update must yield envelopes related in a predictable, structure-preserving manner.

This formalizes CEP's design principle that attestations do not break transformations and transformations do not invalidate attestations.

### 4.3 Revision Monotonicity as a Functorial Constraint

Let  $(\mathbb{N}, \leq)$  denote the natural numbers ordered by the usual non-decreasing relation. Revision numbers in CEP form a functor:

$$\text{Rev} : \mathbf{E} \rightarrow (\mathbb{N}, \leq),$$

which assigns to each enveloped record its revision number and to each morphism  $R \rightarrow R'$  the corresponding order-preserving mapping  $\text{Rev}(R) \leq \text{Rev}(R')$ .

Thus, revision monotonicity is not merely a rule but a categorical invariant:

all CEP morphisms must map to monotone arrows in  $(\mathbb{N}, \leq)$ . This captures the immutability and forward-only progression of the revision field as a semantic constraint enforced at the categorical level.

#### 4.4 Envelopes as a Comonad

The enveloping process admits an alternative and illuminating interpretation: it behaves comonadically when viewed as a context constructor on payloads [Mac Lane, 1971, Awodey, 2010].

Concretely, consider a category  $\mathbf{P}_{\text{env}}$  whose objects are payloads together with their envelopes, but where we regard the payload component as primary and the envelope as context. On this category, an endofunctor

$$C : \mathbf{P}_{\text{env}} \rightarrow \mathbf{P}_{\text{env}}$$

can be defined that “adds context” in the sense of enriching a record with its envelope structure.

A comonad  $(C, \varepsilon, \delta)$  consists of:

- a functor  $C : \mathbf{P}_{\text{env}} \rightarrow \mathbf{P}_{\text{env}}$ ,
- a counit  $\varepsilon : C \Rightarrow \text{Id}$ ,
- a comultiplication  $\delta : C \Rightarrow CC$ ,

satisfying standard coassociativity and counitality laws.

In CEP:

- $C(P)$  is the “contextualized” version of  $P$ , carrying its envelope,
- $\varepsilon$  extracts the underlying payload from its envelope,
- $\delta$  enriches a record with its own envelope again, representing recursive contextualization (e.g., envelopes of envelopes).

This perspective aligns CEP envelopes with the comonadic interpretation of

contextual data in database theory [Spivak, 2014].

## 4.5 Attestations as a Cartesian Natural Transformation

Not all natural transformations preserve the limit structure needed for record-level consistency. CEP attestations must preserve pullbacks: they cannot break joins or invalidate prior provenance.

Thus we refine  $\alpha : \mathcal{E} \Rightarrow \mathcal{E}'$  to be a *cartesian natural transformation*, meaning that each naturality square is a pullback square.

Formally, for every  $f : P \rightarrow P'$ ,

$$\begin{array}{ccc} \mathcal{E}(P) & \xrightarrow{\mathcal{E}(f)} & \mathcal{E}(P') \\ \alpha_P \downarrow & & \downarrow \alpha_{P'} \\ \mathcal{E}'(P) & \xrightarrow{\mathcal{E}'(f)} & \mathcal{E}'(P') \end{array}$$

is required to be a pullback.

Cartesianness ensures:

- attestations preserve joins,
- no new contradictions arise under  $\alpha$ ,
- provenance behaves consistently across merges.

This matches CEP’s requirement that attestations cannot “detach” from the record state they certify.

## 4.6 CEP as a Fibred Category

Let **Sch** denote the category of CEP schemas and controlled vocabulary URIs. Each CEP record is typed by a schema element, giving rise to a functor:

$$\pi : \mathbf{CEP} \rightarrow \mathbf{Sch}.$$

We interpret  $\pi$  as a *fibration*: for any morphism  $s : S \rightarrow S'$  in **Sch** and any record  $R'$  over  $S'$ , there exists a *cartesian lifting* describing the induced transformation on record instances.

This validates two core CEP invariants:

- type-consistent updates always exist,
- vocabulary and schema evolution propagate along cartesian liftings.

The fibred structure provides the mathematical foundation for version migration and schema evolution.

## 4.7 Relation to W3C PROV

CEP attestation components align directly with PROV-DM constructs:

CEP Component	PROV Construct
<code>attestorId</code>	<code>prov:Agent</code>
<code>attestationTimestamp</code>	<code>prov:Generation</code> / <code>prov:Start</code>
<code>proofType</code>	<code>prov:Activity</code> type
<code>proofPurpose</code>	<code>prov:Plan</code> or <code>prov:Role</code>
<code>proofValue</code>	<code>prov:Entity</code> / <code>prov:wasGeneratedBy</code>

Categorically, PROV's `wasGeneratedBy` and `wasAttributedTo` become morphisms in a provenance category **Prov**. The attestation natural transforma-

tion  $\alpha$  factors through a functor:

$$A : \mathbf{CEP} \rightarrow \mathbf{Prov},$$

thereby embedding CEP provenance into the PROV lineage.

This establishes a formal bridge between CEP’s categorical semantics and the W3C PROV standard for provenance representation.

## 5 Canonicalization as a Monoidal Functor

Canonicalization is the mathematical process that ensures the stability and uniqueness of the SNFEI identifier. In the Civic Exchange Protocol (CEP), normalization and assembly are expressed as a structured rewriting system and modeled categorically as a composite of functors equipped with a strict monoidal structure.

**Ordering of Rewrite Rules.** We regard normalization as a total function

$$\text{norm} : \text{RawString} \rightarrow \text{CanonicalString}.$$

Rewrite operations occur at three levels: (1) semantic expansions, (2) structural simplifications, and (3) surface-level normalizations. Some rewrites preserve information while others reduce it, so their order is essential. For example, the corporate form “S.A.” must be expanded before punctuation is removed; otherwise the pattern disappears and the expansion cannot apply.

CEP adopts a *stratified* rewriting system with a fixed ordering of rule strata. Rewrite rules compose associatively within each stratum. Across strata, the ordering is not assumed to commute and is treated as part of the definition of the normalization pipeline.

This behavior is representative rather than exceptional: whenever a semantic rewrite depends on surface structure, rewrite order becomes semantically



meaningful and cannot be treated as commutative.

### 5.0.1 Stratified Canonicalization as a Functor

Let **Raw** be the set of raw strings and **Canon** the set of canonical forms, viewed as a quotient of **Raw** under an equivalence relation that identifies strings representing the same civic entity. Let

$$q : \mathbf{Raw} \rightarrow \mathbf{Canon}$$

be the quotient map selecting canonical representatives.

Normalization is implemented by a finite sequence of rewrite functions

$$n_1, n_2, \dots, n_k : \mathbf{Raw} \rightarrow \mathbf{Raw},$$

each belonging to a particular stratum. Within a stratum composition is associative; across strata the evaluation order is fixed. The overall normalization map is

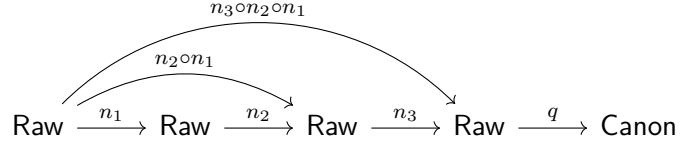
$$\text{norm} = q \circ n_k \circ \dots \circ n_2 \circ n_1 : \mathbf{Raw} \rightarrow \mathbf{Canon}.$$

Categorically, each  $n_i$  is an endomorphism of **Raw** in **Set**. The rule strata form a thin (preordered) category  $\mathcal{S}$ . The normalization pipeline is a functor

$$F : \mathcal{S} \rightarrow \mathbf{End}(\mathbf{Raw}),$$

where  $\mathbf{End}(\mathbf{Raw})$  is the monoidal category of endofunctions on **Raw** under composition. The fixed order of  $\mathcal{S}$  determines the evaluation of the composite; commutativity is not required.

### Diagrammatic View.



Associativity of composition gives

$$(n_3 \circ n_2) \circ n_1 = n_3 \circ (n_2 \circ n_1), \quad \text{norm} = q \circ n_3 \circ n_2 \circ n_1.$$

Stratification appears as the requirement that the arrows  $n_i$  occur in a fixed left-to-right order.

**Relation to Rewriting Systems.** Many established rewriting systems use ordered or strategy-driven rule application. Classical completion procedures, modern term-rewriting frameworks, compiler pipelines, natural-language normalization workflows, and Unicode normalization all apply structured, multi-stage rewriting in which information-preserving rewrites precede information-reducing ones.

CEP follows this standard pattern. Each stratum contains rewrite rules that compose cleanly within their domain, while the fixed ordering across strata functions as the evaluation strategy that guarantees determinism of the composite normalization function.

This strategy-controlled evaluation means that CEP canonicalization is not intended to be confluent or transitive with respect to individual rewrite rules. Only the fully evaluated composite function `norm` defines the canonical equivalence relation.

### 5.0.2 Rewriting Systems as 2-Categories

Rewriting systems may be described in terms of 2-categories or polygraphs: objects are syntactic states, 1-morphisms are rewrite steps, and 2-morphisms represent coherence between rewrite paths. Normalization corresponds to selecting a distinguished representative of each connected component of the 1-skeleton.

CEP fits this model. Each stratum consists of 1-morphisms that are closed under associative composition, while the ordering of strata specifies a deterministic rewriting strategy. The normalization map

$$\text{norm} : \text{Raw} \rightarrow \text{Canon}$$

is therefore a strategy-controlled composite followed by quotienting.

### 5.1 The Normalizing Functor $\mathcal{F}_{\text{normalize}}$

Let  $\mathbf{R}_{\text{raw}}$  be the category whose objects are raw record states and whose morphisms are valid transformations between them. Let  $\mathbf{R}_{\text{canon}}$  be the corresponding category of canonical states.

Normalization is a functor

$$\mathcal{F}_{\text{normalize}} : \mathbf{R}_{\text{raw}} \rightarrow \mathbf{R}_{\text{canon}}. \tag{3}$$

For any morphism  $f : R \rightarrow R'$  in  $\mathbf{R}_{\text{raw}}$ , functoriality ensures that

$$\mathcal{F}_{\text{normalize}}(f) : \mathcal{F}_{\text{normalize}}(R) \rightarrow \mathcal{F}_{\text{normalize}}(R')$$

is the corresponding update on canonical records.

## 5.2 Monoidal Structure and the Canonicalization Functor $\mathcal{C}$

Let  $\mathbf{R}_{\text{canon}}$  carry a monoidal structure in which objects represent normalized components and  $\otimes$  denotes their ordered concatenation. Let  $\mathbf{C}$  be the category of canonical strings with monoidal product also given by concatenation and unit  $I$ .

The assembly step is a strict monoidal functor

$$\mathcal{C} : (\mathbf{R}_{\text{canon}}, \otimes, I) \rightarrow (\mathbf{C}, \otimes, I) \quad (4)$$

satisfying

$$\mathcal{C}(x \otimes y) = \mathcal{C}(x) \otimes \mathcal{C}(y), \quad \mathcal{C}(I) = I. \quad (5)$$

This functor performs the disciplined concatenation that assembles canonical parts (name, address, date, jurisdiction) into a single string in the fixed order prescribed by CEP.

## 5.3 From Canonical Strings to SNFEI

Let

$$H : \mathbf{C} \rightarrow \mathbf{ID}$$

be the hashing functor that maps canonical strings to 256-bit identifiers via the SHA-256 cryptographic hash function. For a record  $R$ ,

$$\text{SNFEI}(R) = H(\mathcal{C}(R_{\text{canon}})), \quad R_{\text{canon}} = \mathcal{F}_{\text{normalize}}(R_{\text{raw}}).$$

The full pipeline is the composite functor

$$H \circ \mathcal{C} \circ \mathcal{F}_{\text{normalize}} : \mathbf{R}_{\text{raw}} \rightarrow \mathbf{ID}. \quad (6)$$

$$\mathbf{R}_{\text{raw}} \xrightarrow{\mathcal{F}_{\text{normalize}}} \mathbf{R}_{\text{canon}} \xrightarrow{\mathcal{C}} \mathbf{C} \xrightarrow{H} \mathbf{ID}$$

Figure 1: Canonicalization pipeline from raw records to stable identifiers.

## 5.4 Identifier Invariance and Equivalence Classes

**Invariance Principle.** If  $f : R \rightarrow R'$  is an identity-preserving morphism in the category **CEP** of valid CEP record states and updates, then

$$H \circ \mathcal{C} \circ \mathcal{F}_{\text{normalize}}(R) = H \circ \mathcal{C} \circ \mathcal{F}_{\text{normalize}}(R').$$

Identity-preserving updates therefore do not change the SNFEI. Two record states lie in the same equivalence class precisely when they produce the same canonical string and hence the same identifier. Strict monoidality of  $\mathcal{C}$  ensures that required components appear in the correct order and cannot be reordered or omitted along valid CEP morphisms.

### Canonicalization as a Monoidal Functor and Identity Guarantee

Because  $\mathcal{F}_{\text{normalize}}$  and  $\mathcal{C}$  are functorial, and  $\mathcal{C}$  is strictly monoidal, the canonical string associated with an entity is uniquely determined by its identity-bearing fields. The SNFEI is therefore a function of an equivalence class of records rather than any particular representation.

This invariance principle underpins CEP’s claims of stable, provenance-respecting identifiers suitable for cross-jurisdiction linkage and data fusion.

## 6 Jurisdictional Adapters as Oplax Functors

### 6.1 Motivation

Jurisdictions frequently maintain their own data schemas, codes, and structural conventions. Let  $\mathbf{J}_{\text{local}}$  denote the category generated by a jurisdiction’s native schema and update rules, and let  $\mathbf{J}_{\text{global}}$  denote the category induced by CEP vocabularies and canonical record structures.

An adapter must relate these two perspectives. In practice, the mapping from local to global structure is often *weak*: optional fields may be absent locally, locally grouped elements may correspond to several globally distinct components, and some structural detail may not be recoverable. These characteristics motivate treating adapters as *oplax functors*, which model structure-preserving translations that allow controlled weakening.

### 6.2 Oplax Functor $\mathcal{A}$

A jurisdictional adapter is modeled as an oplax functor

$$\mathcal{A} : \mathbf{J}_{\text{local}} \longrightarrow \mathbf{J}_{\text{global}}. \quad (7)$$

For any composable morphisms  $f : X \rightarrow Y$  and  $g : Y \rightarrow Z$  in  $\mathbf{J}_{\text{local}}$ , oplaxity provides a coherence morphism

$$\phi_{f,g} : \mathcal{A}(g) \circ \mathcal{A}(f) \Longrightarrow \mathcal{A}(g \circ f), \quad (8)$$

which need not be invertible. This captures the possibility that two distinct local transformations may correspond to a single global transformation, or that information preserved locally becomes conflated at the global level.

This formalism expresses **jurisdictional autonomy**: local schemas may preserve distinctions or collapse details differently than the global schema while still participating coherently in the CEP ecosystem.

$$\begin{array}{c}
X \xrightarrow{f} Y \xrightarrow{g} Z \\
\\
\begin{array}{ccc}
& \text{---} \mathcal{A}(g \circ f) \text{---} & \\
\mathcal{A}(X) \xrightarrow{\mathcal{A}(f)} \mathcal{A}(Y) \xrightarrow{\mathcal{A}(g)} \mathcal{A}(Z) & & \\
& \xrightarrow{\phi_{f,g}} & 
\end{array}
\end{array}$$

Figure 2: An oplax adapter: composition is preserved up to a coherence morphism  $\phi_{f,g}$ .

#### Adapters as Oplax Functors (Jurisdictional Autonomy)

An oplax functor represents the fact that translation from a local schema to the global CEP vocabulary may weaken structure. Local compositions  $g \circ f$  may correspond to their global images only up to a coherence morphism  $\phi_{f,g}$ , allowing partial or jurisdiction-specific mappings while maintaining global consistency.

### 6.3 Correctness Criteria for Adapters

A jurisdictional adapter  $\mathcal{A}$  is valid precisely when it satisfies the following criteria:

1. **Required-field preservation.** Every required field or transformation in  $\mathbf{J}_{\text{local}}$  must map to a required global term or transformation in  $\mathbf{J}_{\text{global}}$ .
2. **Optional weakening via coherence.** Optional or partially represented structures must be mediated by the oplax coherence morphisms  $\phi_{f,g}$  so that weakened structure remains well typed in the global system.

3. **Canonicalization compatibility.** The canonical identifier must be computable on all translated records:

$$\mathcal{C}(\mathcal{A}(R)) \text{ is defined for all admissible local records } R. \quad (9)$$

This ensures that each jurisdiction can produce stable and globally comparable SNFEI identifiers.

4. **Provenance monotonicity.** Adapter-induced transformations must remain consistent with CEP’s provenance envelopes, attestation rules, and revision ordering.
5. **Functoriality on valid updates.** For every valid local update  $u : R \rightarrow R'$ , the induced  $\mathcal{A}(u)$  must constitute a valid update in the global CEP system, modulo the necessary oplax coherence.

A valid adapter thus functions as a structure-respecting mediator between local civic systems and the global CEP representation, providing the flexibility required for jurisdictional independence while maintaining global interoperability and identifier stability.

## 7 Context Tags as Indexed Families

### 7.1 Fibered Category $\pi : \mathbf{CT} \rightarrow \mathbf{CEP}$

Context tags (CTags) are interpretive annotations attached to a record without altering its canonical identity. Their semantics is captured by a *fibered category*

$$\pi : \mathbf{CT} \longrightarrow \mathbf{CEP}, \quad (10)$$

where:

- **CEP** is the base category of identity-bearing records,



- **CT** consists of pairs  $(R, T)$  of a CEP record  $R$  and an attached context tag  $T$ ,
- $\pi$  is the projection functor  $\pi(R, T) = R$ .

For a base object  $R \in \mathbf{CEP}$ , the fiber

$$\mathbf{CT}_R = \{(R, T) \in \mathbf{CT} \mid \pi(R, T) = R\}$$

collects all valid context tags that may be associated with  $R$ . These fibers encode analytic or interpretive information while leaving the canonical structure and identifier of  $R$  unchanged.

#### Context Tags in a Fibered Category (Separation of Concerns)

The projection  $\pi : \mathbf{CT} \rightarrow \mathbf{CEP}$  cleanly separates *identity* (records in **CEP**) from *interpretation* (tags in the fibers  $\mathbf{CT}_R$ ). Any morphism in **CEP** induces a coherent reindexing across fibers, ensuring that context tags evolve with records while never influencing their SNFEI identity.

## 7.2 Functoriality and Reindexing

Let

$$f : R \longrightarrow R'$$

be a morphism in **CEP** representing a valid evolution or update of a record. The fibration provides a corresponding *reindexing functor*

$$f^* : \mathbf{CT}_{R'} \longrightarrow \mathbf{CT}_R. \tag{11}$$

Intuitively,  $f^*$  specifies how tags attached to the updated record  $R'$  may be pulled back to valid tags on the earlier state  $R$ , preserving interpretive meaning along the update path.

$$\begin{array}{ccc}
\mathbf{CT}_R & \xleftarrow{f^*} & \mathbf{CT}_{R'} \\
\pi \downarrow & & \downarrow \pi \\
R & \xrightarrow{f} & R'
\end{array}$$

Figure 3: Reindexing in the fibration  $\pi : \mathbf{CT} \rightarrow \mathbf{CEP}$ .

Reindexing satisfies the standard fibration condition:

$$\pi \circ f^* = f \circ \pi, \quad (12)$$

so the diagram in Figure 3 commutes. This expresses the guiding CEP principle:

**Context tags move with the record's evolution but never change its identity.**

### 7.3 Identity Preservation and Canonical Invariance

A tag object  $T \in \mathbf{CT}_R$  must not modify any canonical data used in identifier generation. Formally, if  $\mathcal{C}$  denotes the canonicalization functor, then

$$\mathcal{C}(R) = \mathcal{C}(R, T), \quad (13)$$

so attaching tags does not affect the canonical form of  $R$ .

Equivalently, the projection functor  $\pi$  is *identity-reflecting*: if two tagged objects  $(R, T)$  and  $(R', T')$  yield the same canonical identifier, then  $R$  and  $R'$  must already be identified in the base category  $\mathbf{CEP}$ .

Consequently:

- CTags are *pure annotations*;
- they introduce no new canonical structure;
- they cannot alter SNFEI values or canonical equivalence classes.

This establishes a strict separation between

identity  $(\mathbf{CEP})$  and interpretation  $(\mathbf{CT}_R)$ .

#### Identity Preservation via Context Tags

Context tags in the fibered category  $\pi : \mathbf{CT} \rightarrow \mathbf{CEP}$  are strictly non-influential on canonical identity. The projection  $\pi$  ensures that no tagged object  $(R, T)$  can alter the canonical form or SNFEI of its base record  $R$ , preserving CEP's invariants.

## 8 Interoperability Results

The categorical semantics developed in the preceding sections yield formal guarantees for CEP's core interoperability claims. These results follow from functoriality, naturality, monoidal structure, and the oplax semantics of jurisdictional adapters.

### 8.1 Functorial Consistency

We first establish coherence between record evolution and attestation.

**Lemma 8.1** (Functorial Consistency of Attested Evolutions). *Let  $f : R \rightarrow R'$  and  $g : R' \rightarrow R''$  be morphisms in  $\mathbf{CEP}$ , and let  $\alpha : \mathcal{E} \Rightarrow \mathcal{E}'$  be the natural transformation representing attestation. Then:*

$$\mathcal{E}'(g \circ f) = \alpha_{R''} \circ \mathcal{E}(g \circ f) = \mathcal{E}'(g) \circ \mathcal{E}'(f).$$

*Proof.* Naturality of  $\alpha$  gives  $\alpha_{R'} \circ \mathcal{E}(f) = \mathcal{E}'(f) \circ \alpha_R$  for all  $f : R \rightarrow R'$ .

Composing these constraints and using associativity of morphisms in **CEP** yields the claim.  $\square$

This shows that CEP’s update–attestation workflow behaves coherently under composition, supporting reliable audit chains and immutability of provenance traces.

## 8.2 Identifier Preservation

We now formalize the stability of canonical identifiers under all valid record evolutions.

**Theorem 8.2** (Identifier Preservation). *Let  $f : R \rightarrow R'$  be any morphism in **CEP**. Let  $\mathcal{C}$  denote the canonicalization functor and  $H$  the deterministic hash used to compute the SNFEI. Then:*

$$H(\mathcal{C}(R)) = H(\mathcal{C}(R')).$$

*Proof.* A valid **CEP** morphism  $f$  preserves all canonical components of the record. Strict monoidality of  $\mathcal{C}$  therefore implies  $\mathcal{C}(R) = \mathcal{C}(R')$ . Since  $H$  is a pure function, the resulting SNFEI values coincide.  $\square$

This theorem provides the mathematical foundation for CEP’s claim of *identity stability across revisions*.

## 8.3 Cross-Jurisdiction Reconciliation

Jurisdictional adapters reconcile heterogeneous local schemas with the global CEP vocabulary. We show that their oplax semantics preserves canonical identity across compositions.

**Theorem 8.3** (Canonical Equivalence Preservation Under Adapters). *Let  $\mathcal{A}_1$  and  $\mathcal{A}_2$  be valid jurisdictional adapters (Section 6). If two local records*

$R$  and  $R'$  satisfy

$$H(\mathcal{C}(\mathcal{A}_1(R))) = H(\mathcal{C}(\mathcal{A}_1(R'))),$$

then after composition with a second adapter,

$$H(\mathcal{C}(\mathcal{A}_2(\mathcal{A}_1(R)))) = H(\mathcal{C}(\mathcal{A}_2(\mathcal{A}_1(R')))).$$

*Proof.* A valid adapter is oplax: it may weaken structure but cannot introduce new canonical content. Thus each  $\mathcal{C}(\mathcal{A}_i(R))$  is defined and invariant under further valid transformations. Strict monoidality of  $\mathcal{C}$  ensures that canonical strings are preserved under composition of such structure-weakening functors. Applying  $H$  yields the result.  $\square$

This theorem shows that multi-stage, cross-jurisdiction pipelines respect CEP’s canonical identity invariants. It provides the formal backbone for *globally consistent, locally autonomous interoperability*.

## 9 Applications

The categorical semantics developed above manifest directly in real civic data workflows. Domain schemas provide typed structures for entities and exchanges, while CEP vocabularies furnish the controlled classifications and relationship types used throughout these workflows. The following examples illustrate how functoriality, monoidality, pullbacks, and oplax adapter semantics enforce CEP’s core invariants in practice.

### 9.1 Civic Entity Records: Identity Stability

Every domain schema (e.g., municipal, educational, environmental) defines a category of entity revisions whose morphisms correspond to admissible

updates. A municipal entity’s lifecycle is therefore a chain in **CEP**:

$$E_1 \xrightarrow{f_1} E_2 \xrightarrow{f_2} \dots$$

where each  $E_i$  conforms to the relevant domain schema and vocabulary.

Identity Invariance ensures that any identity-preserving update  $f_i$  leaves the canonical form unchanged:

$$\mathcal{C}(E_i) = \mathcal{C}(E_{i+1}).$$

Thus the SNFEI remains stable even as auxiliary fields (address, classification, jurisdictional codes, vocabulary-driven status fields) evolve over time. The vocabulary layer guarantees that enumerated fields (e.g. **ACTIVE**, **DISSOLVED**) remain consistent across revisions and jurisdictions.

This yields a strong operational principle:

**Domain schemas define the structure of change; canonicalization guarantees that such change never affects identity.**

## 9.2 Campaign Finance: Compositional Provenance

Domain schemas for campaign finance define typed exchange morphisms, where controlled vocabulary terms specify the semantics of each transfer (e.g. **donation**, **in-kind**, **reallocation**).

A donation from a donor  $D$  to a committee  $C$  is a morphism  $f : D \rightarrow C$ . A downstream transfer from  $C$  to a subcommittee  $S$  is a morphism  $g : C \rightarrow S$ . Their composition

$$g \circ f : D \rightarrow S$$

represents the derived provenance lineage.

Because composition in **CEP** is associative and is compatible with the

vocabulary-governed domain schema, this lineage is canonical across all reporting systems. It enables coherent financial traceability even when jurisdictions differ in local reporting conventions—oplax adapters (Section 6) ensure that structure-weakening translations still preserve this compositional provenance.

### 9.3 Public Contracting: Data Fusion via Pullbacks

Public contracting data arises from multiple domain schemas:

- Vendor Registry schema (legal entities, certifications, status),
- Contract schema (awards, amendments, payments),
- Jurisdiction-specific extensions and vocabularies.

Suppose a Vendor Registry record  $R_V$  and a Contract Record  $R_C$  both reference the same underlying legal entity  $A$ . The fiber product

$$P = R_V \times_A R_C$$

constructs the maximal consistent set of jointly satisfiable facts. This pullback operation merges information along the shared canonical identity specified by the SNFEI and vocabulary-controlled relationship types.

This is the categorical mechanism that guarantees sound cross-system integration. It ensures that:

- canonical identity aligns disparate domain schemas,
- vocabulary-governed code systems reconcile classification fields,
- provenance constraints remain intact across fused datasets.

As a result, auditors can reconcile procurement data across heterogeneous jurisdictional sources, even when local schemas differ substantially, while maintaining CEP’s global invariants.

## 10 Limitations and Future Work

The categorical core presented in this paper captures identity, canonicalization, provenance, and interoperability for discrete, revision-based civic records. Several important extensions remain outside the present scope.

Many civic systems also generate data that is statistical, uncertain, or continuously updated (e.g., longitudinal indicators, evolving aggregates). Incorporating such information may require probabilistic semantics or temporal indexing, which are not modeled in the current framework.

Jurisdictions evolve data models over time. Although CEP supports adapters for structural variation, a full account of schema evolution, including additions, deprecations, and long-term migration paths, remains future work.

**Multi-Stage and Nested Exchanges.** Some workflows involve layered or multi-party processes (e.g., multi-level budget allocations, nested reporting pipelines). These may benefit from higher-structured categorical tools, but such extensions lie beyond the scope of this foundational treatment.

**Rule Sensitivity in Canonicalization.** Certain linguistic cases (such as expansions of abbreviations like “S.A.”) require stratified rule ordering within the normalization pipeline rather than treating all rewrite rules as freely permutable. This refinement does not affect the well-definedness of the canonicalization function but does highlight the need for continued empirical tuning of rule strata.

The present semantics establish a robust core for identity and interoperability. Extending CEP to the richer data practices found across governments and civic ecosystems is a key direction for future work.



## 11 Conclusion

We presented a categorical semantics for the Civic Exchange Protocol that unifies canonicalization, provenance, adapters, and context tags into a coherent mathematical framework. This perspective makes explicit the invariants that govern identity, record evolution, and interoperability across heterogeneous civic data systems.

Canonicalization was formulated as a deterministic monoidal functor, ensuring stable identifiers and well-defined equivalence classes. Jurisdictional adapters were modeled as oplax functors, capturing how local structure may be weakened while preserving global identity. Context tags were expressed via a fibered category, isolating interpretive annotations from canonical record content. Together, these structures yield formal guarantees for identifier preservation, compositional provenance, and cross-jurisdiction reconciliation.

The resulting semantics provides a rigorous foundation for validation, verification, and future extensions of CEP. It enables principled design of domain schemas, vocabularies, and interoperability standards, while remaining extensible to evolving civic workflows. As civic data ecosystems continue to grow in scale and complexity, categorical methods offer a durable and expressive language for ensuring that shared identities and exchanges remain consistent, transparent, and reliable.

## Acknowledgements

Generative AI tools were used to assist with editing and formatting during manuscript preparation. All ideas, analyses, and conclusions are the authors' own. All AI-assisted edits were reviewed and validated, and the authors take full responsibility for the accuracy and integrity of this manuscript.

## Appendix A. Category-Theoretic Background

This appendix summarizes the categorical notions used in the paper. It is intended for readers familiar with structured data, provenance, or formal methods who may not work with category theory daily. The goal is conceptual clarity rather than mathematical depth.

### A.1 Categories

A *category* consists of:

- a collection of **objects**;
- a collection of **morphisms** (arrows) between objects;
- identity morphisms and an associative composition operation.

Intuition:

- Objects represent structured states (e.g., CEP records).
- Morphisms represent valid transformations (e.g., amendments).
- Composition corresponds to performing transformations in sequence.

Example: a sequence of CEP record updates is a chain of morphisms

$$R_0 \rightarrow R_1 \rightarrow R_2 \rightarrow \cdots .$$

### A.2 Functors

A *functor*  $F : \mathbf{C} \rightarrow \mathbf{D}$  maps:

- each object of  $\mathbf{C}$  to an object of  $\mathbf{D}$ ;
- each morphism of  $\mathbf{C}$  to a morphism of  $\mathbf{D}$ ;

while preserving identities and composition.

Intuition:

- A functor expresses a structure-preserving translation.
- CEP uses functors to model envelope construction, canonicalization, and provenance propagation.

### A.3 Natural Transformations

Given functors  $F, G : \mathbf{C} \rightarrow \mathbf{D}$ , a *natural transformation*  $\eta : F \Rightarrow G$  assigns to each object  $X$  a morphism

$$\eta_X : F(X) \rightarrow G(X)$$

such that for every  $f : X \rightarrow Y$  the square

$$\begin{array}{ccc} F(X) & \xrightarrow{\eta_X} & G(X) \\ F(f) \downarrow & & \downarrow G(f) \\ F(Y) & \xrightarrow{\eta_Y} & G(Y) \end{array}$$

commutes.

Intuition:

- Naturality expresses the compatibility condition: “*attest then update*” = “*update then attest*”.
- This coherence underlies CEP envelope and attestation chains.

The categorical tools summarized in this appendix correspond directly to the rewriting-based normalization pipeline developed in the main text. Categories and functors model record states and their admissible transformations; monoidal structure captures the ordered concatenation of canonical components; oplax functors express structure-weakening translations such as jurisdictional adapters; and fibrations formalize how context tags vary over

identity-bearing records. The stratified ordering of rewrite rules in canonicalization aligns with these structures: each stratum corresponds to a functorial transformation, and the monoidal product enforces the fixed sequencing required for deterministic normalization.

## A.4 Monoidal Categories and Stratified Rewriting

A *monoidal category* is a category equipped with:

- a tensor product  $\otimes$ ,
- a unit object  $I$ ,
- coherence laws for associativity and unit behavior.

In CEP, the monoidal structure is string concatenation:

- canonical components (name, address, date, jurisdiction) combine via  $\otimes$ ;
- the canonicalization functor preserves this structure strictly.

This yields determinism in canonical string construction and hence in SNFEI identifiers.

In CEP, the monoidal product reflects the ordered application of rewrite strata in the normalization pipeline, ensuring that semantic expansions, structural simplifications, and surface-level normalizations compose deterministically.

## A.5 Oplax Functors

Given monoidal categories  $(\mathbf{C}, \otimes)$  and  $(\mathbf{D}, \otimes)$ , an *oplax monoidal functor*  $F : \mathbf{C} \rightarrow \mathbf{D}$  includes coherence maps

$$F(X) \otimes F(Y) \rightarrow F(X \otimes Y)$$

that are not required to be isomorphisms.

Intuition:

- Oplax functors allow **structure weakening**.
- This models jurisdictional adapters: local structure may map incompletely or partially into the global vocabulary.

## A.6 Indexed Families and Fibrations

A *fibration*  $\pi : \mathbf{E} \rightarrow \mathbf{B}$  consists of:

- a base category  $\mathbf{B}$ ,
- a total category  $\mathbf{E}$ ,
- a projection functor  $\pi$ ,
- satisfying certain lifting properties.

The fiber over  $B \in \mathbf{B}$  is the category

$$\mathbf{E}_B = \{E \in \mathbf{E} \mid \pi(E) = B\}.$$

Intuition for CEP:

- $\mathbf{B} = \mathbf{CEP}$  (identity-bearing records),
- $\mathbf{E} = \mathbf{CT}$  (records with context tags),
- the fiber over  $R$  contains all valid tags for  $R$ ,
- reindexing tracks tag evolution without altering identity.

## A.7 Universal Properties (Informal)

A universal property characterizes an object uniquely (up to isomorphism) by its relationships to other objects.

In CEP:

- the canonical string is universal for its class of normalized components,
- the SNFEI arises from applying a hashing endofunctor,
- identity preservation follows from the uniqueness of this construction.

Further reading: Mac Lane [Mac Lane \[1971\]](#), Awodey [Awodey \[2010\]](#), Spivak [Spivak \[2014\]](#).

## Appendix B. Worked Examples

This appendix provides concrete examples illustrating how the categorical structures in the paper (functors, fibrations, oplax adapters) manifest in realistic CEP workflows.

### B.1 Normalization and Canonicalization

Consider an entity with the following unnormalized fields:

<b>Name:</b> “City of Springfield” <b>Address:</b> “123 Lincoln Ave” <b>Country:</b> “US” <b>Date:</b> (none)
--

Applying the normalization functor

$$\mathcal{F}_{normalize} : \mathbf{R}_{raw} \rightarrow \mathbf{R}_{canon},$$

yields:

<b>Name:</b> “city of springfield” <b>Address:</b> “123 lincoln ave” <b>Country:</b> “US” <b>Date:</b> “1900-01-01”
--

The canonicalization functor

$$\mathcal{C} : \mathbf{R}_{canon} \rightarrow \mathbf{C}$$

produces the canonical string:

$$\mathcal{C}(R) = \text{city of springfield|123 lincoln ave|US|1900-01-01.}$$

The hashing endofunctor  $H$  computes the identifier:

$$\text{SNFEI}(R) = H(\mathcal{C}(R)).$$

This example illustrates strict monoidality: components are concatenated in a fixed order, independent of non-identity updates.

## B.2 Envelope Functor and Attestation Naturality

Let  $P$  be the canonical payload from Example B.1.

$\mathcal{E}(P)$  = Envelope with schema version and revision metadata,

$\mathcal{E}'(P)$  = Envelope additionally carrying a digital attestation.

The natural transformation  $\alpha : \mathcal{E} \Rightarrow \mathcal{E}'$  assigns

$$\alpha_P : \mathcal{E}(P) \rightarrow \mathcal{E}'(P),$$

representing the application of attestation.

If  $f : P \rightarrow P'$  is a valid record evolution, naturality ensures the square

$$\begin{array}{ccc} \mathcal{E}(P) & \xrightarrow{\alpha_P} & \mathcal{E}'(P) \\ \mathcal{E}(f) \downarrow & & \downarrow \mathcal{E}'(f) \\ \mathcal{E}(P') & \xrightarrow{\alpha_{P'}} & \mathcal{E}'(P') \end{array}$$

commutes. This expresses provenance consistency: attestation interacts correctly with record evolution.



### B.3 Jurisdictional Adapters as Oplax Functors

Consider a local schema  $\mathbf{J}_{\text{local}}$ :

`name, state, address, source-system`

and a global CEP vocabulary  $\mathbf{J}_{\text{global}}$ :

`legalName, jurisdictionIso, location.address,  
entityTypeUri, status`

An adapter  $\mathcal{A}$  maps:

$$\mathcal{A}(\text{name}) = \text{legalName}, \quad \mathcal{A}(\text{state}) = \text{jurisdictionIso},$$

$$\mathcal{A}(\text{address}) = \text{location.address},$$

while `source-system` may map to optional metadata.

Optionality induces oplaxity:

$$\mathcal{A}(x) \otimes \mathcal{A}(y) \rightarrow \mathcal{A}(x \otimes y),$$

which need not be invertible. This models structure weakening in cross-jurisdiction transformations.

### B.4 Context Tags and Fiber Reindexing

Let  $R$  be an entity record with SNFEI  $s$ .

The fiber  $\mathbf{CT}_R$  contains allowable context tags such as:

`analysis.cluster.membership: "cluster-17"  
quality.issue.incomplete_address: true`

If the record evolves via

$$f : R \rightarrow R',$$

then the fibration provides a reindexing functor

$$f^* : \mathbf{CT}_{R'} \rightarrow \mathbf{CT}_R$$

that keeps tags aligned with the record's evolution without altering the underlying identifier  $s$ .

## Appendix C. Proof Sketches

This appendix provides informal but rigorous proof sketches supporting the main theorems of the paper. Each argument reflects the categorical and rewriting-theoretic semantics developed above, including stratified canonicalization, functorial provenance, limit constructions, and oplax adapter behavior.

### C.1 Identifier Preservation Under CEP Morphisms

**Theorem.** If  $f : R \rightarrow R'$  is a morphism in **CEP**, then

$$\mathcal{C}(R) = \mathcal{C}(R') \quad \text{and hence} \quad \text{SNFEI}(R) = \text{SNFEI}(R').$$

**Sketch.** Morphisms in **CEP** preserve:

- all canonical fields required for identifier construction,
- their normalized values under the stratified rewriting system,
- revision monotonicity and schema validity.

Normalization uses a stratified rewriting system whose evaluation order is fixed; canonicalization is implemented as a strict monoidal functor. Thus for any canonical decomposition of record components  $x_i$ :

$$\mathcal{C}(R) = \bigotimes_i \mathcal{F}_{\text{normalize}}(x_i), \quad \mathcal{C}(R') = \bigotimes_i \mathcal{F}_{\text{normalize}}(x'_i),$$

and  $f$  preserves each canonical component  $x_i = x'_i$ . Therefore  $\mathcal{C}(R) = \mathcal{C}(R')$ .

Applying the hashing endofunctor  $H$  preserves equality.

## C.2 Naturality of Attestations Implies Provenance Consistency

**Theorem.** If  $\alpha : \mathcal{E} \Rightarrow \mathcal{E}'$  is the attestation transformation, then for any valid  $f : P \rightarrow P'$ , provenance remains consistent.

**Sketch.** Naturality gives:

$$\alpha_{P'} \circ \mathcal{E}(f) = \mathcal{E}'(f) \circ \alpha_P.$$

Interpreting  $\mathcal{E}$  and  $\mathcal{E}'$  as envelope-constructing functors and  $\alpha$  as an attestation step:

$$\text{attest then update} = \text{update then attest}.$$

Thus provenance chains commute with record evolution and cannot be invalidated by further updates.

## C.3 Pullbacks Guarantee Consistent Joins

**Theorem.** If two record fragments map to a common object  $A$  in **CEP** and the pullback exists, then the join is consistent with canonical identity and schema constraints.

**Sketch.** Given morphisms  $f : R \rightarrow A$  and  $g : R' \rightarrow A$ , the pullback  $P$  has projections

$$\pi_1 : P \rightarrow R, \quad \pi_2 : P \rightarrow R',$$

satisfying the universal property: any object mapping compatibly into  $R$  and  $R'$  factors uniquely through  $P$ .

Identity consistency follows because both  $R$  and  $R'$  map to the same canonical representative in  $A$ . Schema consistency follows from finite completeness of the category of CEP records: limit constructions preserve well-formedness and canonical identity invariants.

## C.4 Oplax Adapters Preserve Canonical Equivalence Classes

**Theorem.** If  $\mathcal{A}$  is a valid oplax jurisdictional adapter, then local records that map to the same global canonical string belong to the same SNFEI equivalence class.

**Sketch.** An oplax functor allows structure weakening but supplies coherence maps:

$$\mathcal{A}(x) \otimes \mathcal{A}(y) \rightarrow \mathcal{A}(x \otimes y),$$

which need not be invertible but remain functorial. This ensures that stratified normalization and strict monoidal canonicalization commute with  $\mathcal{A}$  up to coherence.

If two local records  $R_\ell, R'_\ell$  satisfy:

$$\mathcal{C}(\mathcal{A}(R_\ell)) = \mathcal{C}(\mathcal{A}(R'_\ell)),$$

then their canonical forms agree after adapter translation. Applying  $H$  yields identical SNFEI values, so canonical identity is preserved even under adapter weakening.

## Appendix D. Diagrammatic Intuition

This appendix provides informal diagrams for the categorical structures introduced in the main text. The goal is to support intuition rather than to introduce new formal content.

### D.1 Morphisms in CEP

Figure 4 depicts CEP objects as attested record states and morphisms as provenance-preserving transformations.

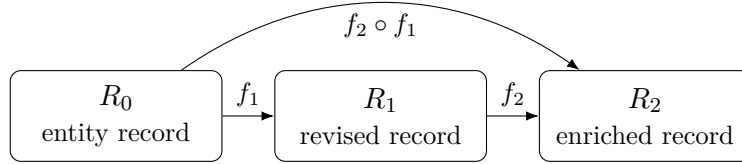


Figure 4: Morphisms in **CEP** as provenance-preserving transformations. Each arrow corresponds to a valid record evolution that preserves schema validity, revision monotonicity, and canonical identity.

### D.2 Naturality of Attestations

Figure 5 visualizes attestations as a natural transformation between two envelope functors  $\mathcal{E}, \mathcal{E}' : \mathbf{P} \rightarrow \mathbf{E}$ .

### D.3 Canonicalization Pipeline

Figure 6 summarizes the canonicalization pipeline: normalization, assembly, and hashing.

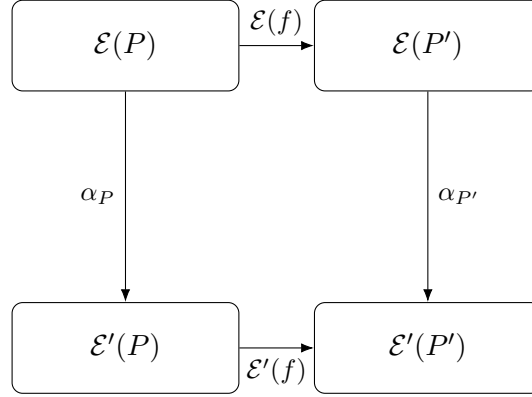


Figure 5: Naturality square for attestations. The equality  $\alpha_{P'} \circ \mathcal{E}(f) = \mathcal{E}'(f) \circ \alpha_P$  expresses that attestation commutes with valid transformations of payloads.

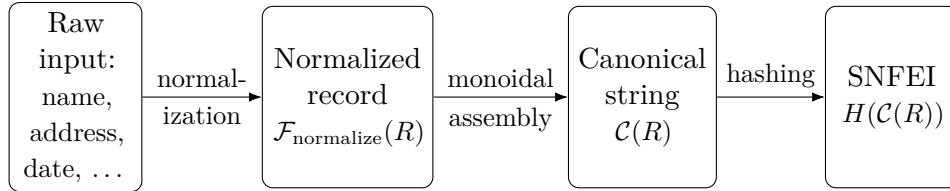


Figure 6: The canonicalization pipeline as a composition of a normalization functor, a strict monoidal assembly functor, and a hashing endofunctor that collapses canonical strings to identifiers.

## D.4 Jurisdictional Adapters as Oplax Functors

Figure 7 illustrates the weakened coherence condition for an oplax functor  $\mathcal{A} : \mathbf{J}_{\text{local}} \rightarrow \mathbf{J}_{\text{global}}$ .

## D.5 Context Tags as a Fibration

Figure 8 shows the projection  $\pi : \mathbf{CT} \rightarrow \mathbf{CEP}$  and the fibers of context tags above a base record.

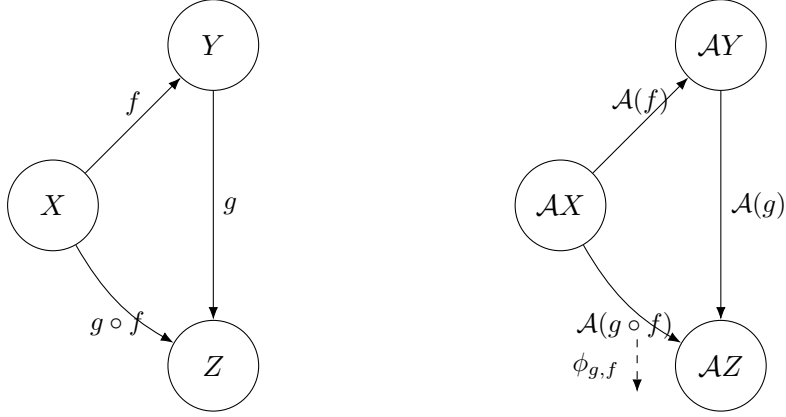


Figure 7: Oplax coherence for a jurisdictional adapter  $\mathcal{A} : \mathbf{J}_{\text{local}} \rightarrow \mathbf{J}_{\text{global}}$ . The dashed 2-cell  $\phi_{g,f}$  witnesses that  $\mathcal{A}(g \circ f)$  and  $\mathcal{A}(g) \circ \mathcal{A}(f)$  need not coincide strictly, reflecting possible lossy or partial mappings.

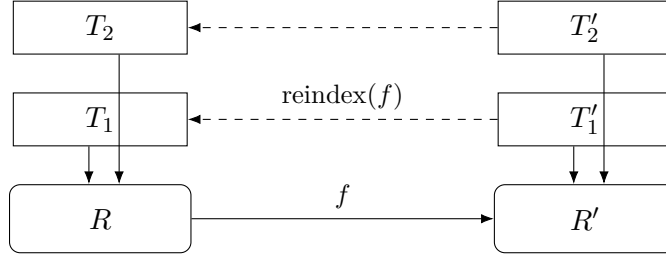


Figure 8: Context tags as a fibration  $\pi : \mathbf{CT} \rightarrow \mathbf{CEP}$ . Each base record  $R$  has a fiber of permitted tags above it. A morphism  $f : R \rightarrow R'$  induces a reindexing between fibers, while the underlying canonical identity remains anchored in the base.



## Appendix E. Glossary for Non-Category Theory Readers

This appendix provides short, non-technical explanations of the categorical concepts used in the paper. The intention is to make the mathematical structure of CEP more accessible to readers from computer science, data engineering, public administration, and civic-technology communities.

### E.1 Categories

A *category* is a mathematical setting that describes:

- some **objects** (things), and
- some **morphisms** (arrows) between them.

A category is like a directed graph with rules: every arrow (morphism) has a source and target, arrows can be composed, and each object has an identity arrow that starts from the object, points back to the object, and does nothing.

The latin root *morph* means "form" or "shape", so a morphism is a way of changing the form of one object into another.

In CEP:

- objects = record states,
- morphisms = valid record transformations (updates, amendments, joins).

Defining CEP as a category formalizes the idea of "things and the allowed changes between them".

## E.2 Functors

A *functor* is a mapping between categories that preserves structure. It sends:

- each object to another object, and
- each morphism to another morphism,

in a way that respects composition and identity.

In CEP, a functor often represents a pipeline stage, such as:

- wrapping a payload in an envelope,
- normalizing noisy text into canonical components,
- assembling canonical strings.

Functors ensure that if a record evolves legally, its transformed version evolves legally too.

*Function* and *functor* have the same root and similarities, but different meanings: functions map elements within sets, while functors map objects and morphisms between categories.

## E.3 Natural Transformations

A *natural transformation* is a structured way of comparing two functors. If functors are "processing stages", a natural transformation is a systematic way to convert the output of one stage into the output of another.

In CEP, attestations are modeled as natural transformations:

- the envelope functor produces a plain metadata wrapper,
- the attested-envelope functor produces a cryptographically validated wrapper.

Naturality expresses the idea:

"Whether you process then attest, or attest then process, you end up with consistent provenance."

## E.4 Monoidal Categories

A *monoidal category* is a category equipped with a notion of "combining things".

Examples:

- strings combine by concatenation,
- datasets combine by joining,
- workflows combine by sequencing.

Canonicalization is the process of converting data into a standard format, most commonly by selecting a single, preferred output to represent a piece of content that could have multiple versions.

Canonicalization in CEP is *monoidal* because it combines individual normalized components into a single canonical string in a strictly deterministic order.

## E.5 Strict Monoidal Functors

A *strict monoidal functor* is a functor that preserves the combination structure *exactly*.

In CEP:

- the order of pieces (name, address, date, jurisdiction) must always be preserved,
- no additional symbols or whitespace are introduced,
- the final output is the canonical string fed to the SHA-256 cryptographic hash function.

This strictness is what guarantees the stability of the SNFEI identifier.

## E.6 Oplax Functors

An *oplax functor* preserves structure in a weakened, direction-sensitive way. The *op* prefix indicates *opposite* directionality and *lax* indicates looseness. An oplax functor therefore "loosens" structure in a specific direction and allows "preservation up to a coherence map" rather than strict equality. A coherence map is a controlled way of relating two structures that are not strictly equal.

It allows:

- missing fields,
- lossy interpretations,
- mappings that preserve meaning but not full structure.

Jurisdictional adapters in CEP behave oplaxly because:

- local data models may omit fields,
- global vocabularies may have stricter typing,
- some equivalences hold only "up to" a coherence rule.

Oplax behavior models "local autonomy with global convergence". It means that local jurisdictions can adapt data flexibly while still ensuring that the global system remains coherent and consistent.

## E.7 Pullbacks (Consistent Joins)

A *pullback* is the categorical notion of a *consistent join*.

If two data sources both refer to the *same entity or event*, the pullback constructs the most precise version of their agreement.

This formalizes CEP's guarantee that:

Records may be joined only when they assert compatible facts.

A pullback enables us to formally define the data fusion operation that combines records from different schemas while preserving:

- canonical identity,
- vocabulary-governed semantics,
- provenance constraints.

*Data fusion* refers to the process of integrating multiple data sources to produce more consistent, accurate, and useful information than that provided by any individual source.

## E.8 Fibered Categories

A *fibered category* describes a setting where each object has a family of additional structures "above it".

CEP is a fibered category. In CEP:

- the base category is **CEP**,
- the fibers contain context tags (CTags).

This cleanly separates:

- **identity** (in the base category), and
- **interpretation or annotation** (in the fiber).

CTags provide information about a record, without affecting the canonical SNFEI identifier.

## E.9 Universal Properties

A *universal property* describes an object that is "best" or "most canonical" for a specific purpose.

SNFEI behaves like a universal property construction because:

- it is determined by a canonical string,
- it is invariant under allowed morphisms,
- any other identifier consistent with CEP's invariants must factor uniquely through this construction.

This is the mathematical justification enabling us to treat SNFEI as a stable, verifiable, compositional global identifier.

## E.10 Summary Table

Concept	Intuition	CEP Role
Category	Things, allowed changes	Record states and updates
Functor	Structure-preserving map	Normalization, envelopes
Natural transformation	Coherent comparison	Attestations
Monoidal category	Combine things	Canonical assembly
Strict monoidal functor	Combine exactly	SNFEI stability
Oplax functor	Weak structure map	Jurisdiction adapters
Pullback	Consistent join	Merging record fragments
Fibered category	Object annotations	CTags above records
Universal property	Optimal construction	Identifier uniqueness

## E.11 Closing Note

These notions are not introduced for abstraction’s sake; they express precisely and formally the structural guarantees CEP requires to support interoperability, trust, and cross-jurisdiction governance.

These mathematical definitions allow the protocol to be:

- modular in its design,
- formally verifiable in its behavior,
- capable of operating consistently even when sources use different schemas, vocabularies, or jurisdictional rules,
- and extensible to future domains.

By grounding CEP in category theory, we provide a rigorous foundation for its design principles and operational claims.

## References

- Steve Awodey. *Category Theory*, volume 52 of *Oxford Logic Guides*. Oxford University Press, 2nd edition, 2010. ISBN 978-0199237180. URL <https://global.oup.com/academic/product/category-theory-9780199237180>.
- Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998. doi: <https://doi.org/10.1017/CBO9781139172752>. URL <https://www.cambridge.org/core/books/term-rewriting-and-all-that/>.
- Saunders Mac Lane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics*. Springer, 1st edition, 1971. ISBN 978-0387900353.
- David I. Spivak. *Category Theory for the Sciences*. MIT Press, 2014. ISBN 978-0262028136.

Terese. *Term Rewriting Systems*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2003.