Categorical Semantics for the Civic Exchange Protocol (CEP)

Denise Case[1,2]

[1]Northwest Missouri State University, Computer Science and Information Systems, Maryville, MO, USA
[2]Civic Interconnect, Ely, MN, USA

December 8, 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 the core primitives of civic information systems. This paper develops a categorical semantics for CEP, providing a mathematical foundation for its compositional structure, identity guarantees, and interoperability claims. We model CEP records as typed objects in a finitely complete category, with morphisms representing provenance-preserving transformations. Envelopes, attestations, and context tags are formalized as natural transformations between functors encoding system-level views of the data. We show that canonicalization corresponds to a monoidal functor that preserves identity and equivalence classes, and that jurisdictional adapters arise as oplax functors mediating between local schemas and global vocabularies. This categorical semantics clarifies CEP's invariants, sketches correctness properties for its identifier construction (SNFEI), and establishes a basis for future extensions in data fusion, federation, 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.

Our contributions are:

1. A construction of a category **CEP** whose objects are well-typed record states and whose morphisms represent valid, provenance-preserving transformations.

2. A functorial account of envelopes, attestations, and context tags as

natural transformations between appropriate record functors.

3. A characterization of canonicalization as a monoidal functor that induces stable identifiers (SNFEI) and preserves equivalence classes of civic actors and events.

4. A semantics for jurisdictional adapters as oplax functors mediating between local schemas and global vocabularies.

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 between 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 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:

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

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`.

3.2 Morphisms: Provenance-Preserving Transformations

Morphisms in **CEP** represent admissible, provenance-preserving transformations between record states. A morphism $f : R \to R'$ 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**: $\mathrm{Schema}(R) \to \mathrm{Schema}(R')$ must be respected, meaning R' remains valid under the same or a successor schema.
2. **Revision monotonicity**: $\mathrm{Revision}(R) \leq \mathrm{Revision}(R')$. 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.

3.3 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 $f : R \to A$ and $g : R' \to A$, their pullback is an object P equipped with morphisms $p_1 : P \to R$ and $p_2 : P \to R'$ 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 $f : R \to A$ and $g : R' \to A$ 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.

3.4 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 un-enveloped 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*

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

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 $f : P \to P'$ 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

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

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

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

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 $f : P \to P'$ in $\mathbf{P}$, the following square commutes:

$$\begin{tikzcd}
\mathcal{E}(P) \arrow[r, "\mathcal{E}(f)"] \arrow[d, "\alpha_P"'] & \mathcal{E}(P') \arrow[d, "\alpha_{P'}"] \\
\mathcal{E}'(P) \arrow[r, "\mathcal{E}'(f)"'] & \mathcal{E}'(P')
\end{tikzcd}$$

Attestations as Natural Transformations (Provenance Commutativity)

The naturality of α ensures that the act of attesting (α_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:

$$\mathrm{Rev} : \mathbf{E} \to (\mathbb{N}, \leq),$$

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

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: $\mathcal{E}$ behaves as a *comonad* on the category of payloads [Mac Lane, 1971, Awodey, 2010].

A comonad $(\mathcal{E}, \varepsilon, \delta)$ consists of:

- a functor $\mathcal{E} : \mathbf{P} \to \mathbf{P}$,
- a counit $\varepsilon : \mathcal{E} \Rightarrow \mathrm{Id}$,
- a comultiplication $\delta : \mathcal{E} \Rightarrow \mathcal{E}\mathcal{E}$,

satisfying standard coassociativity and counitality laws.

In CEP:

- $\mathcal{E}(P)$ is the "contextualized" version of P,
- ε extracts the underlying payload,
- δ enriches a record with its own envelope again, representing recursive contextualization.

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 $f : P \to P'$,

$$\begin{tikzcd} \mathcal{E}(P) \arrow[r, "\mathcal{E}(f)"] \arrow[d, "\alpha_P"'] & \mathcal{E}(P') \arrow[d, "\alpha_{P'}"] \\ \mathcal{E}'(P) \arrow[r, "\mathcal{E}'(f)"'] & \mathcal{E}'(P') \end{tikzcd}$$

is required to be a pullback.

Cartesianness ensures:

- attestations preserve joins,
- no new contradictions can arise under α,
- 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 vocabulary URIs. Each CEP record is typed by a schema element, giving rise to a functor:

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

We interpret π as a *fibration*: for any morphism $s : S \to S'$ 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 evolution propagates 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
`attestorId`	`prov:Agent`
`attestationTimestamp`	`prov:Generation` / `prov:Start`
`proofType`	`prov:Activity` type
`proofPurpose`	`prov:Plan` or `prov:Role`
`proofValue`	`prov:wasGeneratedBy`

Categorically, PROV's "wasGeneratedBy" and "wasAttributedTo" become morphisms in a provenance category **Prov**. The attestation natural transformation α factors through a functor:

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

thereby embedding CEP provenance into the PROV lineage.

5 Canonicalization as a Monoidal Functor

Canonicalization is the mathematical step that guarantees the stability and uniqueness of the SNFEI identifier. In this section we model the normalization and assembly pipeline as a composition of functors with a strict monoidal structure.

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

Let $\mathbf{R}_{\text{raw}}$ denote raw, noisy input states and $\mathbf{R}_{\text{canon}}$ denote canonical states in which Unicode normalization, abbreviation expansion, and filtering have been applied. The deterministic cleanup step is a functor

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

By functoriality, any valid transformation of raw records $f : R \to R'$ in $\mathbf{R}_{\text{raw}}$ induces a corresponding transformation of canonical records $\mathcal{F}_{\text{normalize}}(f) : \mathcal{F}_{\text{normalize}}(R) \to \mathcal{F}_{\text{normalize}}(R')$, ensuring that normalization behaves consistently under updates and repairs.

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

The canonicalization process orders and concatenates normalized components (e.g., name, address, date, jurisdiction) into a single string. Both $\mathbf{R}_{\text{canon}}$ and the category of canonical strings $\mathbf{C}$ carry a monoidal structure $(\otimes, I)$ given by concatenation and an appropriate unit (e.g., the empty string).

The overall assembly process is a strict monoidal functor

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

satisfying

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

Intuitively, $\mathcal{C}$ is the disciplined concatenation procedure: it respects both the component-wise structure and the fixed ordering prescribed by CEP (for example, name first, then address, then date, then country code).

5.3 From Canonical Strings to SNFEI

Let $H : \mathbf{C} \to \mathbf{ID}$ be the hashing functor that maps canonical strings to 256-bit identifiers using SHA–256. For any canonicalized record R_{canon} we write

$$\mathrm{SNFEI}(R) = H(\mathcal{C}(R_{\text{canon}})), \tag{6}$$

where $R_{\text{canon}} = \mathcal{F}_{\text{normalize}}(R_{\text{raw}})$.

The composite

$$H \circ \mathcal{C} \circ \mathcal{F}_{\text{normalize}} : \mathbf{R}_{\text{raw}} \to \mathbf{ID} \tag{7}$$

is the full canonicalization pipeline from raw inputs to stable identifiers.

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

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

5.4 Identifier Invariance and Equivalence Classes

The central invariant of CEP can now be stated categorically.

Invariance Principle. If $f : R \to R'$ is a valid identity-preserving morphism in **CEP**, then

$$H \circ \mathcal{C} \circ \mathcal{F}_{\text{normalize}}(R) = H \circ \mathcal{C} \circ \mathcal{F}_{\text{normalize}}(R'). \tag{8}$$

In other words, identity-preserving updates do not change the SNFEI. Two record states R and R' belong to the same equivalence class precisely when they share the same canonical string and hence the same SNFEI. The strict monoidality of $\mathcal{C}$ guarantees that reordering or omitting required components is not permitted by the schema, and thus cannot occur along valid morphisms in **CEP**.

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, not of any particular representation, which is exactly the identity guarantee CEP requires.

6 Jurisdictional Adapters as Oplax Functors

6.1 Motivation

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

Any adapter must reconcile these two perspectives. However, the mapping from local to global structure is typically *weak*: optional fields may not appear locally, local compositions may collapse distinctions preserved globally, and some structural information may be lost.

This motivates the use of *oplax functors*, which formalize structure-weakening translations between categories.

6.2 Oplax Functor $\mathcal{A}$

An adapter is modeled as an oplax functor

$$\mathcal{A} : \mathbf{J}_{\text{local}} \longrightarrow \mathbf{J}_{\text{global}}. \tag{9}$$

For every pair of composable morphisms $f : X \to Y$ and $g : Y \to Z$ in $\mathbf{J}_{\text{local}}$, oplaxity means that we have a *coherence morphism*

$$\phi_{f,g} : \mathcal{A}(g) \circ \mathcal{A}(f) \implies \mathcal{A}(g \circ f), \tag{10}$$

which need not be invertible. This captures the possibility of *lossy harmonization*: two distinct local transformations may be indistinguishable at the global level.

$$\begin{tikzcd} X \arrow[r, "f"] & Y \arrow[r, "g"] & Z \end{tikzcd}$$

$$\begin{tikzcd} \mathcal{A}(X) \arrow[r, "\mathcal{A}(f)"] \arrow[rr, bend left, dashed, "\mathcal{A}(g \circ f)"] & \mathcal{A}(Y) \arrow[r, "\mathcal{A}(g)"] \arrow[r, phantom, "\phi_{f,g}"] & \mathcal{A}(Z) \end{tikzcd}$$

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

This structure formalizes **Jurisdictional Autonomy**: local schemas may preserve distinctions or collapse details not present in the global vocabulary, while still participating coherently in the CEP ecosystem.

Adapters as Oplax Functors (Jurisdictional Autonomy)

An oplax functor models the fact that the translation from a local schema to the global CEP vocabulary may weaken structure. Local compositions $g \circ f$ may only match their global images up to a coherence morphism $\phi_{f,g}$, reflecting partial, lossy, or jurisdiction-specific mappings. This accommodates local flexibility without violating global interoperability.

6.3 Correctness Criteria for Adapters

A jurisdictional adapter $\mathcal{A}$ is valid for CEP if and only if it satisfies the following criteria:

1. **Required-field preservation.** For every required local field, $\mathcal{A}$ must map it to a required term in the CEP vocabulary. Formally, required morphisms in $\mathbf{J}_{\text{local}}$ must map to required morphisms in $\mathbf{J}_{\text{global}}$.

2. **Optional weakening via coherence.** Optional fields or partial structures must be captured by the oplax coherence morphisms $\phi_{f,g}$, ensuring that weakened structure is still well-typed globally.

3. **Canonicalization compatibility.** The canonical identifier must be computable on the image of the adapter:

$$\mathcal{C}(\mathcal{A}(R)) \quad \text{is defined for all admissible local records } R. \tag{11}$$

   This guarantees that local jurisdictions can produce stable and globally comparable SNFEI identifiers.

4. **Provenance monotonicity.** Adapter-induced transformations must not contradict envelope attestations or revision-order invariants established in **CEP**.

5. **Functoriality on valid updates.** For any valid local update $u : R \to R'$, the image $\mathcal{A}(u)$ must remain a valid update in the CEP record system, modulo the required oplax coherence.

A valid adapter therefore acts as a structure-respecting mediator between local civic systems and the global CEP representation while allowing precisely the amount of structural weakening necessary for jurisdictional independence.

7 Context Tags as Indexed Families

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

Context tags (CTags) are interpretive annotations attached to a record without altering its canonical identity. Their semantics is most naturally expressed via a *fibered category*:

$$\pi : \mathbf{CT} \longrightarrow \mathbf{CEP}, \tag{12}$$

where:

- **CEP** is the base category of identity-bearing records,
- **CT** is the category whose objects are *pairs* (R, T) of a record and a context tag attached to it,
- π is the projection functor $\pi(R, T) = R$.

For each base object $R \in \mathbf{CEP}$, the *fiber* $\mathbf{CT}_R$ is the collection of all valid context tags that may be attached to R. These fibers encode interpretive or analytic information without modifying the canonical structure of the underlying record.

Context Tags in a Fibered Category (Separation of Concerns)

The fibration $\pi : \mathbf{CT} \to \mathbf{CEP}$ cleanly separates identity (base objects in **CEP**) from interpretation (fibers $\mathbf{CT}_R$). Any morphism in the base category induces a structured reindexing between fibers, guaranteeing that context tags track valid record evolutions without affecting the canonical identifier. This formalizes the principle that CTags annotate a record but never influence its SNFEI identity.

7.2 Functoriality and Reindexing

Given a morphism

$$f : R \longrightarrow R' \tag{13}$$

in the base category **CEP** (representing a valid record evolution or update), the fibration supplies a corresponding *reindexing functor*:

$$f^* : \mathbf{CT}_{R'} \longrightarrow \mathbf{CT}_R. \tag{14}$$

Intuitively, f^* describes how tags on a revised record R' may be pulled back to valid tags on the earlier state R, preserving their interpretive meaning.

$$\begin{tikzcd}
\mathbf{CT}_R \arrow[d, "\pi"'] & \mathbf{CT}_{R'} \arrow[l, "f^*"'] \arrow[d, "\pi"] \\
R \arrow[r, "f"'] & R'
\end{tikzcd}$$

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

The reindexing condition ensures:

$$\pi \circ f^* = f \circ \pi, \tag{15}$$

i.e., the diagram in Figure 3 commutes.

This formalizes an essential CEP design principle:

> **Context tags move with the record's evolution but do not change its identity.**

7.3 Identity Preservation and Canonical Invariance

A tag object $T \in \mathbf{CT}_R$ must not alter any canonical data used in identifier generation. Formally:

$$\mathcal{C}(R) = \mathcal{C}(R, T), \tag{16}$$

where $\mathcal{C}$ is the canonicalization functor defined previously. Equivalently, the projection functor π is *identity-reflecting*: if (R, T) and (R', T') share the same canonical identifier, then R and R' must already do so in **CEP**.

Thus:

- CTags are *pure annotations*;
- they cannot introduce new canonical structure;
- they cannot affect the SNFEI or equivalence classes of records.

This guarantees a strict separation between:

$$\text{identity} \quad \text{(base category)} \qquad \text{and} \qquad \text{interpretation} \quad \text{(fibers).}$$

8 Interoperability Results

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

8.1 Functorial Consistency

We first show that the interaction between record evolution and attestation is coherent under composition.

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

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

Proof. By naturality of α, each attestation map satisfies $\alpha_{R'} \circ \mathcal{E}(f) = \mathcal{E}'(f) \circ \alpha_R$. Associativity of morphism composition in **CEP** gives the result. □

This establishes that CEP's update–attestation workflow is compositionally well-behaved, enabling reliable audit chains and provable immutability of provenance.

8.2 Identifier Preservation

We now formalize the guarantee that canonical identifiers remain stable under all valid record evolutions.

Theorem 8.2 (Identifier Preservation)**.** *Let $f : R \to R'$ 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. By definition of a valid **CEP** morphism, f preserves the canonical components of the record. By the strict monoidality of $\mathcal{C}$, we have $\mathcal{C}(R) = \mathcal{C}(R')$. Since H is a pure function, the SNFEI values coincide. □

This result provides the mathematical foundation for CEP's claim of *identity stability across revisions*, ensuring that record evolution does not alter canonical identity.

8.3 Cross-Jurisdiction Reconciliation

Jurisdictional adapters mediate between heterogeneous local structures and the global CEP vocabulary. We show that their oplax semantics preserves canonical equivalence classes.

Theorem 8.3 (Canonical Equivalence Preservation Under Adapters)**.** *Let* $\mathcal{A}_1, \mathcal{A}_2$ *be valid jurisdictional adapters (as defined in 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. Each adapter $\mathcal{A}_i$ is oplax, meaning it may weaken local structure but cannot introduce new canonical content (Section 6.3). Thus $\mathcal{C}(\mathcal{A}_i(R))$ is defined and stable. The strict monoidality of $\mathcal{C}$ ensures that canonical strings are preserved under composition of such structure-weakening functors. Applying H yields the claimed equality. □

This theorem establishes that cross-jurisdiction data pipelines cannot violate CEP's canonical identity invariants, provided the adapters conform to the correctness criteria of Section 6. It provides the formal backbone for CEP's claim of *globally consistent, locally autonomous interoperability*.

9 Applications

This section illustrates how the categorical semantics developed above manifest in real-world civic data workflows. Each example highlights a core CEP invariant and shows how the categorical structure enforces it.

9.1 Civic Entity Records: Identity Stability

A municipal entity's lifecycle is modeled as a chain of morphisms in **CEP**:

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

where each E_i is a revision of the same entity. Identity Invariance (Section 3.2) ensures that any identity-preserving amendment f_i leaves the canonical form unchanged:

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

Thus the SNFEI remains stable even as auxiliary data (e.g., address, classification, or contact fields) evolves over time.

9.2 Campaign Finance: Compositional Provenance

A donation from a donor D to a committee C is modeled as an exchange morphism $f : D \to C$. A subsequent transfer from C to a subcommittee S is a morphism $g : C \to S$. The composed morphism $g \circ f : D \to S$ represents the derived provenance lineage. Because composition in **CEP** is associative, this lineage is canonical and immutable, enabling consistent financial traceability across multiple reporting systems.

9.3 Public Contracting: Data Fusion via Pullbacks

Consider a Vendor Registry record R_V and a Contract Record R_C that both reference the same underlying legal entity A. The pullback

$$P = R_V \times_A R_C$$

constructs the maximal consistent set of facts compatible with both records. This is exactly the categorical mechanism that guarantees sound cross-system data integration (Section 3.3) and supports audits that reconcile public procurement data across heterogeneous jurisdictional sources.

10 Limitations and Future Work

Several aspects of CEP lie outside the present categorical core, including probabilistic data, continuous-time flows, and schema evolution. Higher categorical structures may be relevant for nested multi-stage exchanges.

11 Conclusion

We presented a categorical semantics for the Civic Exchange Protocol. This semantics clarifies identity, invariants, and interoperability, and provides a foundation for future work in formal verification and cross-jurisdiction data reconciliation.

Appendix A. Category-Theoretic Background

This appendix summarizes the categorical notions used in the paper. It is intended for readers with a background in data modeling, databases, or formal methods who may not use 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;
- an associative composition operation and an identity arrow for each object.

Intuition:

- Objects represent states or structured data (e.g., CEP records).
- Morphisms represent valid transformations (e.g., amendments).
- Composition corresponds to applying transformations sequentially.

Example: A sequence of record updates in CEP corresponds to a chain of morphisms

$$R_0 \to R_1 \to R_2 \to \cdots .$$

A.2 Functors

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

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

such that identities and composition are preserved.

Intuition:

- A functor is a structure-preserving translation.
- CEP uses functors to model processes such as envelope construction or canonicalization.

A.3 Natural Transformations

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

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

such that each square

$$\begin{tikzcd}
F(X) \arrow[r, "\eta_X"] \arrow[d, "F(f)"'] & G(X) \arrow[d, "G(f)"] \\
F(Y) \arrow[r, "\eta_Y"'] & G(Y)
\end{tikzcd}$$

commutes for all morphisms $f : X \to Y$.

Intuition:

- Naturality means: *"attest first then transform" = "transform first then attest"*.
- This is exactly the coherence needed for CEP attestation chains.

A.4 Monoidal Categories

A *monoidal category* is a category equipped with:

- a tensor product $\otimes$ combining objects,

- a unit object I,
- coherence laws ensuring associativity and proper unit behavior.

In CEP, the relevant monoidal structure is **string concatenation**:

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

This allows the SNFEI to be treated as a universal construction.

A.5 Oplax Functors

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

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

that need not be invertible.

Intuition:

- Oplax functors allow **structure weakening**.
- This directly models jurisdictional adapters: some structure from the local schema may be incomplete or only partially mappable to the global vocabulary.

A.6 Indexed Families and Fibrations

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

- a base category $\mathbf{B}$,
- a total category $\mathbf{E}$,
- a projection functor π,
- 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 plus context tags),
- the fiber over R is the set of all allowed context tags for R,
- fibers reindex naturally when R evolves.

This formalizes the idea that context tags do not affect identity.

A.7 Universal Properties (Informal)

A universal property specifies an object uniquely up to isomorphism by the role it plays in relation to others.

In CEP:

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

This concludes the appendix. Readers seeking more detail may consult Mac Lane Mac Lane [1971], Awodey Awodey [2010], and Spivak Spivak [2014].

Appendix B. Worked Examples

This appendix provides concrete examples illustrating the categorical interpretations developed in the paper. The goal is to anchor the abstract definitions (functors, fibrations, oplax structure) in realistic CEP workflows.

B.1 Normalization and Canonicalization Example

Consider an entity with the following unnormalized fields:

Name: "City of Springfield" **Address:** "123 Lincoln Ave" **Country:** "US" **Date:** (none)

After running the normalization functor

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

we obtain:

Name: "city of springfield" (lowercased) **Address:** "123 lincoln ave" **Country:** "US" **Date:** "1900-01-01" (jurisdiction-default)

The canonicalization functor

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

then produces the canonical string:

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

Applying the hashing endofunctor H yields:

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

This example demonstrates the strict monoidal behavior of $\mathcal{C}$: each component is concatenated in a fixed order, independent of non-identity updates such as revision increments.

B.2 Envelope Functor and Attestation Naturality

Let P be the payload from Example B.1.

$$\mathcal{E}(P) = \text{Envelope with schema version, timestamps, and revision 1,}$$

$$\mathcal{E}'(P) = \text{Envelope with cryptographic attestation.}$$

The natural transformation $\alpha : \mathcal{E} \Rightarrow \mathcal{E}'$ assigns to P a morphism

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

representing the application of an attestation (e.g., a digital signature).

If the payload evolves by a valid morphism $f : P \to P'$, the square

$$\begin{tikzcd}
\mathcal{E}(P) \arrow[r, "\alpha_P"] \arrow[d, "\mathcal{E}(f)"'] & \mathcal{E}'(P) \arrow[d, "\mathcal{E}'(f)"] \\
\mathcal{E}(P') \arrow[r, "\alpha_{P'}"'] & \mathcal{E}'(P')
\end{tikzcd}$$

commutes, meaning provenance is consistent.

B.3 Jurisdictional Adapters as Oplax Functors

Consider a local schema $\mathbf{J_{local}}$:

name, **state**, **address**, **source-system**

and a global CEP vocabulary $\mathbf{J_{global}}$:

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

An adapter $\mathcal{A}$ maps:

$$\mathcal{A}(\texttt{name}) = \texttt{legalName}, \qquad \mathcal{A}(\texttt{state}) = \texttt{jurisdictionIso},$$

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

while `source-system` may be mapped into an optional metadata field.

Because optional fields may be dropped or weakened, the functor is oplax:

$$\mathcal{A}(x) \otimes \mathcal{A}(y) \to \mathcal{A}(x \otimes y)$$

need not be an isomorphism.

B.4 Context Tags and Fiber Reindexing

Let R be an entity record with SNFEI s.

The fiber $\mathbf{CT}_R$ contains allowable tags, e.g.:

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

If the base record evolves

$$f : R \to R',$$

then the fibration guarantees a reindexing

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

that keeps tags aligned with the record's identity but never alters s.

Appendix C. Proof Sketches

This appendix provides informal but rigorous proof sketches supporting the main theorems in the paper. They are intended to clarify correctness arguments for CEP canonicalization, provenance, and interoperability.

C.1 Identifier Preservation Under CEP Morphisms

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

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

Sketch. Morphisms in **CEP** preserve:

- canonical fields,
- their normalized values,
- revision monotonicity,
- schema validity.

The canonicalization functor $\mathcal{C}$ is strict monoidal. Thus for any decomposition of fields into components x_i,

$$\mathcal{C}(R) = \bigotimes_i \mathcal{F}_{normalize}(x_i)$$

and the same decomposition holds for R'. Since f does not alter canonical components, the equality follows immediately.

Hashing with H preserves equality of inputs.

C.2 Naturality of Attestations Implies Provenance Consistency

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

Sketch. By naturality,

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

Interpreting $\mathcal{E}(f)$ and $\mathcal{E}'(f)$ as envelope-level updates and α as a cryptographic attestation, the equation says:

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

ensuring an immutable provenance chain.

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 guaranteed not to violate identity or schema constraints.

Sketch. Let $f : R \to A$ and $g : R' \to A$. The pullback P satisfies:

$$\pi_1 : P \to R, \qquad \pi_2 : P \to R',$$

with universal property:

any object that maps into both R and R' in a way compatible with f and g factors uniquely through P.

Identity preservation follows because both R and R' share the same canonical image in A. Schema consistency follows from stability under limits (finite completeness).

C.4 Oplax Adapters Preserve Canonical Equivalence Classes

Theorem. If $\mathcal{A}$ is an oplax jurisdictional adapter satisfying the criteria in §6.3, then two local records that map to the same global canonical string must belong to the same SNFEI equivalence class.

Sketch. The oplax coherence map

$$\mathcal{A}(x) \otimes \mathcal{A}(y) \to \mathcal{A}(x \otimes y)$$

may not be invertible, but it is still functorial. Thus the normalized canonical fragments commute with the adapter.

If two local records R_ℓ, R'_ℓ satisfy:

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

then their SNFEI values match. Thus canonical identity is preserved 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.

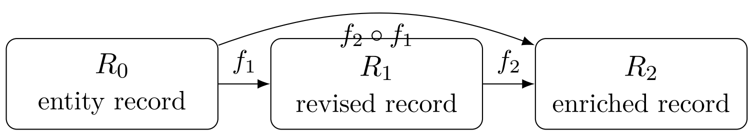

Figure 4: Morphisms in **CEP** as provenance-preserving transformations. Each arrow corresponds to a valid record evolution that preserves schema validity, revision monotonicity, and canonical identity.

D.2 Naturality of Attestations

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

D.3 Canonicalization Pipeline

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

$$
\begin{tikzcd}
\mathcal{E}(P) \arrow[r, "\mathcal{E}(f)"] \arrow[d, "\alpha_P"'] & \mathcal{E}(P') \arrow[d, "\alpha_{P'}"] \\
\mathcal{E}'(P) \arrow[r, "\mathcal{E}'(f)"'] & \mathcal{E}'(P')
\end{tikzcd}
$$

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.

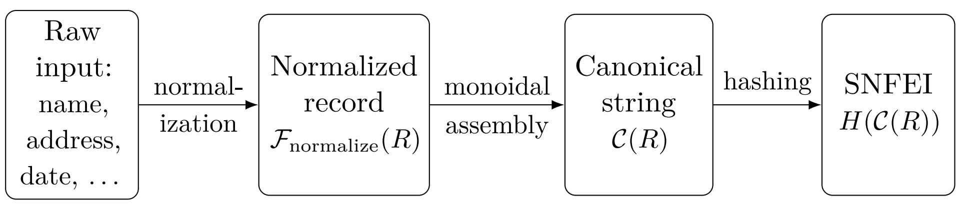

Figure 6: The canonicalization pipeline as a composition of a normalization functor, a strict monoidal assembly functor, and a hashing endofunctor that collapses canonical strings to identifiers.

D.4 Jurisdictional Adapters as Oplax Functors

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

D.5 Context Tags as a Fibration

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

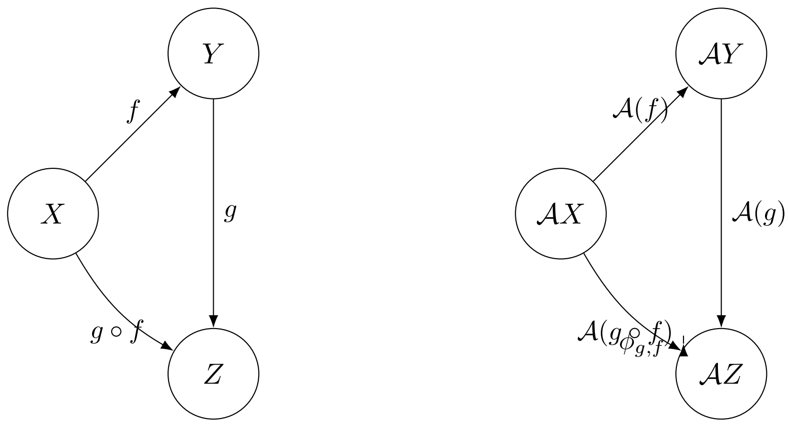

Figure 7: Oplax coherence for a jurisdictional adapter $\mathcal{A} : \mathbf{J_{local}} \to \mathbf{J_{global}}$. The dashed 2-cell $\phi_{g,f}$ witnesses that $\mathcal{A}(g \circ f)$ and $\mathcal{A}(g) \circ \mathcal{A}(f)$ need not coincide strictly, reflecting possible lossy or partial mappings.

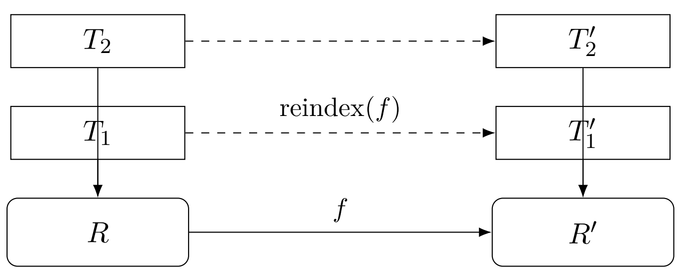

Figure 8: Context tags as a fibration $\pi : \mathbf{CT} \to \mathbf{CEP}$. Each base record R has a fiber of permitted tags above it. A morphism $f : R \to R'$ induces a reindexing between fibers, while the underlying canonical identity remains anchored in the base.

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.

You can think of a category as a directed graph with rules: every morphism has a source and target, arrows can be composed, and each object has an identity arrow that does nothing.

In CEP:

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

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.

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 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 SHA–256.

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

E.6 Oplax Functors

An *oplax functor* preserves structure in a weakened way.

Think of it as "oppositionally lax". It is "lax" as in loose, but the op- signals that the direction of the non-invertible natural transformation (α) is reversed or "opposite".

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."

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.

E.8 Fibered Categories

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

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).

This is why CTags do not affect the 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 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 for treating 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	Per-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 the structural guarantees CEP requires for interoperability, trust, and cross-jurisdiction governance.

They allow the protocol to be:

- modular,
- formally verifiable,
- robust to heterogeneity,
- 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.

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.