On the Structure of Value

Value, Information, and the Cartesian Degeneracy

A Categorical Separation of Value Systems and Information Processing

Anantha Krishnan
Sarva Labs, Princeton, NJ, USA
`ak@sarva.ai`

March 2026

*Computation over value requires participant-indexed linear structure,
while classical information processing arises as its Cartesian special case.*

Abstract

This paper analyzes a categorical distinction between value systems and information processing systems. We establish the precise categorical structure of each domain, prove that no structure-preserving embedding can map value into information, and show that information processing is a strict special case of value management — not the reverse.

We show that value is a non-separable entanglement of Participant, Context, and Content. It cannot be decomposed into its constituent parts and reassembled without structural destruction. This is not merely an economic observation; it is a categorical one. We interpret value systems as computational domains whose categorical semantics correspond to symmetric monoidal categories without natural diagonals. Information inhabits Cartesian monoidal categories where copying and discarding are foundational primitives. By Fox's theorem, Cartesian structure is monoidal structure with additional constraints. The general case is monoidal. The special case — obtained by imposing a natural comonoid on every object — is Cartesian.

We term the imposition of this comonoid the *Cartesian Degeneracy*: the condition under which the general structure of value collapses into the special case of information processing. Parametrically, this occurs when the participant dimension reduces to the trivial unit object — where WHO disappears from the computational structure.

Three results follow. First, every Cartesian system is a symmetric monoidal system with strict additional structure. Second, the general monoidal machine subsumes the special case: information can be processed on a value machine, but value cannot be processed on an information machine. Third, the structural conditions that produce the coordination and consistency challenges endemic to value systems built on Cartesian foundations are not engineering failures. They are mandatory consequences of the degeneracy — each requires the natural comonoid that the value domain categorically excludes, and each is therefore absent by construction in the general case.

Keywords: category theory, linear logic, symmetric monoidal categories, value theory, participant-indexed structure, domain separation.

MSC (2020): 18M05, 03G30, 68Q55.

Contributions. (1) Formal domain identification of value as a non-Cartesian, participant-indexed structure. (2) Proof of categorical domain separation: no essentially surjective comonoid-preserving functor exists from **Info** to **Value**. (3) Three computational consequences for distributed systems, established as structural necessities rather than engineering choices.

Series Positioning. This paper is CC2 in the Contextual Compute series. CC1 [1] establishes the philosophical and quantum-theoretic foundation of participant-indexed value. The present paper (CC2) establishes the categorical separation of value from information and proves the Domain Separation Theorem. CC3 [2] introduces the K-Machine — the abstract computational model whose categorical foundation is established here. CC4 [3] derives the operational semantics of the K-Machine through the PIC model. The results of this paper are the mathematical foundation from which CC3 and CC4 derive their core claims.

1 Introduction

Contemporary computing infrastructures are expected to manage two fundamentally different classes of entities: information and value. Information can be copied freely — files, messages, and data structures may be duplicated and discarded without altering their meaning. Value-bearing objects behave differently. Financial assets, permissions, identity claims, and ownership rights cannot be duplicated without breaking the semantics of the system.

Using the same computational primitives for both creates structural friction: systems must add layers of coordination and validation to prevent inconsistencies that the underlying model actively permits. This paper shows that this friction is not an engineering deficiency. It is a categorical necessity — a consequence of modeling value using primitives designed for freely replicable information. Value requires a computational structure centered on participants rather than anonymous global state. The computational consequences of this structure, including a participant-indexed model of computation, are developed in subsequent work.

1.1 Related Work

The resource interpretation of computation originates in Girard's linear logic [11], which distinguishes propositions that may be used arbitrarily often from those consumed exactly once. Categorical semantics of linear logic have been extensively developed: Seely [16] established the correspondence between linear logic and $*$-autonomous categories; Benton [6] introduced mixed linear and non-linear logic; Melliès [13] provides a comprehensive survey of categorical semantics in this setting.

Categorical models of quantum mechanics provide a parallel instantiation of non-Cartesian structure. Abramsky and Coecke [4] developed a categorical semantics of quantum protocols in which $(\mathbf{FHilb}, \otimes)$ is the canonical example of a symmetric monoidal category without natural diagonals. The quantum no-cloning theorem — that an unknown quantum state cannot be duplicated — is precisely the absence of a natural comonoid in **FHilb**, established formally by Wootters and Zurek [17].

While linear logic and categorical resource semantics describe systems in which duplication is restricted, these frameworks are typically presented in terms of abstract logical propositions or physical resources. The perspective adopted here is domain-theoretic: we ask which existing categorical structure correctly characterizes value-bearing computational systems, and show the answer is participant-indexed linear resource theories. This is an interpretive contribution, not a new categorical theorem — but the identification is not merely interpretive: it yields a provable structural result establishing that no comonoid-preserving functor can bridge the two domains.

Closest in spirit is the mathematical theory of resources developed by Coecke, Fritz, and Spekkens [7], which formalizes resource theories as symmetric monoidal categories in which certain state transitions are forbidden by the categorical structure. The present work extends this line of inquiry by identifying value systems specifically as participant-indexed linear resource structures and deriving the architectural consequences of that identification for distributed computation.

Session types for communication protocols provide a further parallel. Honda [14] and Wadler [15] developed linear type disciplines for communication channels that must be used exactly once. This linear discipline for communication is directly analogous to the treatment of value transfer as linear agreement between participants — though session types operate at the language level on a Cartesian substrate, whereas the present work concerns the structural linearity of the semantic domain itself. The distinction is between linearity as a constraint on programs and linearity as a property of the computational primitive.

1.2 Paper Organization

Section 2	Establishes that value is intrinsically participant-indexed and non-separable.
Section 3	Formalizes this categorically: value inhabits SMCs; information inhabits their Cartesian specialization.
Section 4	Proves the Domain Separation Theorem and the Semantic Independence Proposition.
Section 5	Establishes strict subsumption, recovery of classical computation, and structural impossibility.
Sections 6–8	Develops architectural consequences: parallelism, replication, and hierarchy inversion.
Section 9	States explicit scope limitations: what this paper does not do.
Section 10	Open problems in formalization, resource theory, complexity, and quantum extensions.
Appendix A	Anticipated objections: this is just linear logic; value is not conserved; type systems enforce linearity; essential surjectivity is too strong.

2 What Value Is

We use ‘value’ in a structural, not economic sense. A value state is any state whose identity is constitutively dependent on a participant — one that cannot be decomposed into participant-independent content without semantic destruction. Financial assets are the motivating example, but the structure applies equally to permissions, identity claims, delegated authority, and other domains where who holds something is inseparable from what is held.

Before introducing any category theory, we must be formally precise about what value is. This precision is not pedantry. The failure to be precise at this exact juncture is the source of the modeling error this paper addresses. We do not assume our conclusion; we demonstrate it.

2.1 The Three Components

Every instance of value involves three dimensions.

Participant (P). The entity for whom value exists: a person, an institution, an autonomous agent. The participant is not an external observer of the system. The participant is a constitutive dimension of the value state itself.

Context (C). The situation in which the participant holds the content: legal status, relational proximity, temporal bounds, circumstantial constraints. Context is not background metadata. It defines what the content is for this participant at this moment.

Content (X). The object, token, claim, or asset at the center of the value relationship.

Classical economics and classical computation treat these as three independent inputs to a function, with X as the primary bearer of value and P and C as modifiers. We now demonstrate this is structurally false.

2.2 The Rs. 100 Note

The following example is illustrative rather than formal; the categorical argument does not depend on it. Consider a Rs. 100 note with a unique serial number — materially identical to millions of others. The same content (X) across varying combinations of Participant (P) and Context (C):

Participant (P)	Context (C)	Value State
Me	In my wallet	Spendable liquidity
Me	Pledged as collateral	Locked, inaccessible
Bank	My account balance	Claim on reserves
Deceased person	In probate	Frozen, legal resolution pending
Criminal	Proceeds of crime	Seizable, illicit
Child	Birthday gift	Precious beyond purchasing power
Counterfeiter	Forged note	Worthless, structurally dangerous

The content (X) is identical in every row. The value state is fundamentally distinct in every row. The note does not possess value intrinsically. The joint state (P, C, X) constitutes the value.

2.3 Why Not a Cartesian Product?

The classical response models value as a mapping from the Cartesian product of three independent sets:

$$V : P \times C \times X \to \mathbb{R}$$

This structure assumes separability: each component contributes independently. We now show this fails — not as an approximation, but structurally.

2.4 The Separability Test

A function $f(x, y)$ is separable if $f(x, y) = g(x) \cdot h(y)$: the contribution of x can be isolated from the contribution of y. If value were separable over a Cartesian product, the following equation would hold for any two participants and two content objects under a fixed context:

$$V(\text{Alice}, \text{Book}) + V(\text{Bob}, \text{Car}) = V(\text{Alice}, \text{Car}) + V(\text{Bob}, \text{Book})$$

The example. Fix the context: a single-afternoon barter market. Alice urgently needs a car and has no use for books. Bob urgently needs the specific book Alice has and already owns a car he no longer uses.

	Book	Car
Alice	≈ 0	$\approx$ \$50,000
Bob	$\approx$ \$50,000	≈ 0

Applying the separability test:

$$\begin{aligned} \text{Left:} \quad & V(\text{Alice, Book}) + V(\text{Bob, Car}) = 0 + 0 = 0 \\ \text{Right:} \quad & V(\text{Alice, Car}) + V(\text{Bob, Book}) = 50{,}000 + 50{,}000 = 100{,}000 \end{aligned}$$

$$0 \neq 100{,}000$$

The equation fails by $100,000. The content has no value independent of who holds it in what context. The joint state (P, C, X) does not factor. **Value is not separable.**

2.5 Entanglement, Not Correlation

Correlation means a definite value exists and knowing P gives information about it. Entanglement means no value exists independent of the joint state. P and C do not reveal value — they constitute it.

Test: *What is the value of this note, independent of any participant?* The correct answer is that the question is malformed. The value is not unknown. It is structurally non-existent in that configuration. The note participates in value. It does not possess value.

2.6 The Formal Definition

Definition 2.1 (Value State)**.** A value state is an element of a state space indexed by participant:

$$|V\rangle_P \in \mathcal{H}_P$$

Each participant P has their own state space $\mathcal{H}_P$ (denoted suggestively in Hilbert-space notation) of possible value states.

Definition 2.2 (Value Entanglement)**.** A value entanglement is a joint state of Participant, Context, and Content that cannot be factored into independent component states:

$$|V\rangle_{P,C,X} \neq |P\rangle \otimes |C\rangle \otimes |X\rangle$$

for any independent states $|P\rangle$, $|C\rangle$, $|X\rangle$. All three are bound. The binding is the value. The notation is structural, not physical: the point is non-factorizability, not quantum mechanics.

2.7 Summary of Section 2

Three things established: (1) value is a joint state depending simultaneously on P, C, and X; (2) value is non-separable — the separability test fails by $100,000; (3) value is entangled, not correlated — the question “what is the value independent of any participant?” is malformed, not merely difficult to answer.

The correct mathematical structure for a non-separable joint state is the tensor product $\otimes$ in a category without natural projections back to its components — a symmetric monoidal category. The gap between $\otimes$ and $\times$ is the categorical gap between value and information. Everything that follows from that gap is the subject of the rest of this paper.

3 The Categorical Framework

Section 2 established that value is a non-separable joint state. We now identify the correct mathematical structure, show why value must inhabit it, and demonstrate that Cartesian structure is a strict special case obtained by imposing additional constraints on the general structure. All definitions below are standard. The contribution is domain identification.

3.1 Symmetric Monoidal Categories: The General Structure

Definition 3.1 (Symmetric Monoidal Category)**.** A *symmetric monoidal category* (SMC) $(\mathcal{C}, \otimes, I)$ consists of a category $\mathcal{C}$, a bifunctor $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$, and a unit object I, together with natural isomorphisms:

$$\begin{aligned} \alpha_{A,B,C} &: (A \otimes B) \otimes C \xrightarrow{\sim} A \otimes (B \otimes C) && \text{(associator)} \\ \lambda_A &: I \otimes A \xrightarrow{\sim} A && \text{(left unitor)} \\ \rho_A &: A \otimes I \xrightarrow{\sim} A && \text{(right unitor)} \\ \sigma_{A,B} &: A \otimes B \xrightarrow{\sim} B \otimes A && \text{(symmetry)} \end{aligned}$$

satisfying Mac Lane’s coherence conditions. An SMC is *not* required to have natural projections $A \otimes B \to A$ or $A \otimes B \to B$.

The canonical example of a non-Cartesian SMC is $(\mathbf{FHilb}, \otimes)$ — finite-dimensional Hilbert spaces under tensor product. **FHilb** has no natural projections, which is precisely why quantum states cannot be cloned. This connection is made precise in Section 3.

3.2 Comonoids: The Categorical Form of Copying

Definition 3.2 (Commutative Comonoid)**.** A *commutative comonoid* in an SMC $(\mathcal{C}, \otimes, I)$ is an object A equipped with morphisms

$$\Delta_A : A \to A \otimes A \qquad \varepsilon_A : A \to I$$

satisfying coassociativity, counitality, and cocommutativity. A *natural* family of comonoids is one where Δ and ε are natural transformations.

The morphism Δ_A is the categorical form of copying. The morphism ε_A is the categorical form of discarding. An SMC without a natural family of comonoids is one in which copying and discarding are not uniformly available.

3.3 Fox's Theorem

Theorem 3.3 (Fox, 1976)**.** *A symmetric monoidal category* $(\mathcal{C}, \otimes, I)$ *is Cartesian monoidal if and only if every object carries a natural commutative comonoid structure. Formally:* $\mathbf{CMC} \subsetneq \mathbf{SMC}$ *(strict inclusion).*

Remark 3.4. The strict inclusion $\mathbf{CMC} \subsetneq \mathbf{SMC}$ means there exist SMCs that are not Cartesian. **FHilb**, **FinVect** under $\otimes$, and the value category **Value** of Definition 3.9 are all examples. The natural comonoid is a non-trivial additional constraint, not a consequence of monoidal structure alone.

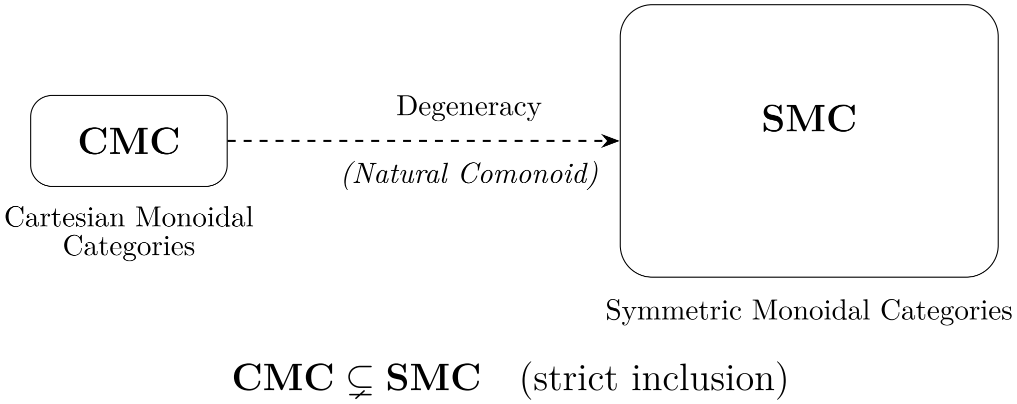

Figure 1. Information systems inhabit the Cartesian specialization (CMC); value systems inhabit the more general symmetric monoidal structure (SMC). The dashed arrow denotes the Cartesian Degeneracy — the imposition of a natural comonoid on every object. The inclusion is strict: there exist SMCs that are not Cartesian, and **Value** is one of them.

3.4 The Cartesian Degeneracy

Definition 3.5 (Cartesian Degeneracy)**.** The *Cartesian Degeneracy* is the specialization of a symmetric monoidal category into a Cartesian monoidal category through the imposition of a natural commutative comonoid on every object.

This specialization can be expressed parametrically. Let P denote the participant index set. We model participant-indexed structure as:

$$P \to \mathbf{Cat}, \quad P \mapsto C_P$$

a functor from the discrete category on participants to the category of small categories. For the trivial participant I (the unit object):

$$\begin{aligned} C_I &= C_{\text{Cartesian}} && \text{(participant trivial: specializes to Cartesian)} \\ C_P &= C_{\text{Monoidal}} && \text{(participant non-trivial: remains genuinely monoidal)} \end{aligned}$$

When $P = I$, the participant index vanishes. The fiberwise monoidal structure collapses to a global Cartesian product. Objects become context-free, and a natural comonoid becomes globally definable.

Remark 3.6 (Irreversibility). Once the monoidal tensor is fixed as a categorical product, its projection structure is determined functorially and cannot be forgotten without altering the monoidal structure itself.

Remark 3.7 (Morphisms and the Grothendieck Construction). The indexing $P \mapsto C_P$ organizes *objects* over participants. A full treatment requires specifying morphisms between participant-indexed contexts and proving these morphisms are linear. This is captured via the Grothendieck construction $\int_P C_P$ over the participant category $\mathbf{P}$, where fiberwise morphisms represent witnessed value transfers. The full fibered structure of **Value**, and the proof that its morphisms do not admit a natural comonoid, are developed in the companion paper [2].

3.5 Two Domains Formally Defined

Definition 3.8 (Information Category)**.** An *information category* $(\mathbf{Info}, \times, 1)$ is a Cartesian monoidal category. Every object carries a natural commutative comonoid:

$$\Delta_D : D \to D \times D \qquad \varepsilon_D : D \to 1$$

State is anonymous, globally readable, and freely duplicable.

Definition 3.9 (Value Category)**.** A *value category* $(\mathbf{Value}, \otimes, I)$ is a linear symmetric monoidal category: an SMC in which no natural family of commutative comonoids exists on all objects. (We use *linear symmetric monoidal category*, *participant-indexed SMC*, and *value category* interchangeably henceforth; the first emphasises structure, the second emphasises indexing, the third emphasises domain.) This term denotes a well-known class of SMCs, interpreted here as the correct categorical domain for value-bearing computation; no new categorical structure is introduced. Objects model participant-indexed ownership states.

Property	**Info**	**Value**
Monoidal structure	Cartesian ($\times$)	Linear ($\otimes$)
Natural comonoid	Present on every object	Absent
Copying	Δ_D natural	No natural Δ
Discarding	ε_D natural	No natural ε
State	Anonymous, global	Participant-indexed
Fox's theorem	Applies	Fails (by design)

3.6 Bridge to Section 4

Every system that models value as replicated anonymous state implicitly claims a structure-preserving bridge between **Info** and **Value** exists. Section 4 proves it does not.

4 The Domain Separation Theorem

4.1 The Theorem

Intuitively, the theorem states that systems whose primitives assume copyable state cannot faithfully represent domains where duplication is structurally forbidden.

Theorem 4.1 (Domain Separation)**.** *Let* $(\mathbf{Info}, \times, 1)$ *be an information category and* $(\mathbf{Value}, \otimes, I)$ *be a value category. There does not exist an essentially surjective strong symmetric monoidal functor*

$$F : (\mathbf{Info}, \times, 1) \longrightarrow (\mathbf{Value}, \otimes, I)$$

that preserves the natural commutative comonoid structure of **Info***.*

Equivalently: if such F *exists and is essentially surjective, then* $(\mathbf{Value}, \otimes, I)$ *must be Cartesian monoidal — contradicting the assumption that* **Value** *is linear.*

Definition 4.2 (Comonoid-Preserving Functor)**.** A strong symmetric monoidal functor $F : \mathbf{Info} \to \mathbf{Value}$ is *comonoid-preserving* if it sends the canonical natural commutative comonoid induced by the Cartesian structure on each $D \in \mathbf{Info}$ to a natural commutative comonoid on $F(D) \in \mathbf{Value}$, up to the coherence isomorphisms of the strong monoidal structure:

$$F(\Delta_D) \cong \Delta_{F(D)} \qquad F(\varepsilon_D) \cong \varepsilon_{F(D)}$$

Definition 4.3 (Essential Surjectivity)**.** A functor $F : \mathbf{Info} \to \mathbf{Value}$ is *essentially surjective* if every object $V \in \mathbf{Value}$ is isomorphic to $F(D)$ for some $D \in \mathbf{Info}$. This is the formal requirement for a full domain representation.

4.2 Proof

Proposition 4.4 (Strong Monoidal Functors Preserve Comonoids)**.** *Let* $F : \mathbf{Info} \to \mathbf{Value}$ *be a strong symmetric monoidal functor. Then for each* $D \in \mathbf{Info}$ *with comonoid* $(\Delta_D, \varepsilon_D)$*:*

$$F(\Delta_D) : F(D) \to F(D) \otimes F(D) \qquad F(\varepsilon_D) : F(D) \to I$$

equip $F(D)$ *with a commutative comonoid structure in* **Value***, natural with respect to morphisms in the replete image of* F*.*

Proof. Strong monoidal functors preserve the comonoid axioms because these axioms are commutative diagrams involving $\otimes$ and the unit, which strong monoidal functors preserve up to coherence isomorphisms. Naturality transports along the replete image by functoriality of F. □

Definition 4.5 (Replete Image)**.** The *replete image* $\mathcal{C}$ of $F : \mathbf{Info} \to \mathbf{Value}$ is the full replete subcategory of **Value** whose objects are those isomorphic to $F(D)$ for some $D \in \mathbf{Info}$.

Proof of Theorem 4.1. Let $\mathcal{C} \subseteq \mathbf{Value}$ be the replete image of F.

Since F is strong symmetric monoidal, $\mathcal{C}$ inherits a symmetric monoidal structure $(\mathcal{C}, \otimes, I)$ from **Value**.

By Proposition 4.4, F transports the canonical comonoid $(\Delta_D, \varepsilon_D)$ of each $D \in \mathbf{Info}$ to morphisms

$$F(\Delta_D) : F(D) \to F(D) \otimes F(D) \qquad F(\varepsilon_D) : F(D) \to I$$

equipping each $F(D)$ with a commutative comonoid in $\mathcal{C}$.

Because $\mathcal{C}$ is replete, every object is isomorphic to some $F(D)$, and the comonoid structure transports along these isomorphisms. Hence every object of $\mathcal{C}$ carries a commutative comonoid natural with respect to morphisms in $\mathcal{C}$.

By Fox's theorem (Theorem 3.3), $(\mathcal{C}, \otimes, I)$ is Cartesian monoidal.

If F is essentially surjective, $\mathcal{C} \simeq \mathbf{Value}$, forcing **Value** to be Cartesian monoidal — contradicting its linearity. □

Remark 4.6. The proof dependencies are: comonoid preservation (Proposition 4.4) → natural comonoids on every object of the replete image → Cartesianity (Fox's theorem) → contradiction.

4.3 The No-Cloning Correspondence

Theorem 4.7 (No-Cloning, Wootters–Zurek, 1982)**.** *There is no unitary* U *such that for all* $|\psi\rangle$*:* $U(|\psi\rangle \otimes |0\rangle) = |\psi\rangle \otimes |\psi\rangle$*.*

The no-cloning theorem states that no natural diagonal $\Delta : A \to A \otimes A$ exists in **FHilb**. The categorical obstruction is the same in both theorems:

	Quantum Mechanics	Value Theory
Category	**FHilb**	**Value**
Structure	SMC, not Cartesian	SMC, not Cartesian
Categorical reason	No natural $\Delta : A \to A \otimes A$	No natural $\Delta : A \to A \otimes A$
Forbidden operation	Cloning quantum states	Duplicating value
Theorem	No-cloning theorem	Domain Separation Theorem
Constitutive role	Observer in measurement	Participant in valuation

Remark 4.8. The correspondence is structural, not metaphorical. Both domains exhibit non-separable joint structure, and non-separability categorically precludes natural copying.

4.4 Corollaries

Corollary 4.9 (Structural Overhead)**.** *Any computational architecture modeling value transfer using information-system primitives must simulate linearity by auxiliary constraints external to the primitive categorical structure.*

Corollary 4.10 (Global-Replication Artifacts)**.** *Coordination constraints arising in architectures built on replicated anonymous global state are structural consequences of the Cartesian specialization.*[1]

Corollary 4.11 (Absence of Natural Diagonals)**.** *There is no natural family of morphisms* $\Delta_V : V \to V \otimes V$ *in* **Value** *that duplicates value objects while preserving participant-indexing.*

Proposition 4.12 (Semantic Independence)**.** *The incompatibility between information and value semantics established in Theorem 4.1 persists independently of assumptions about:*

(i) computational speed (including infinitely fast computation),
(ii) network reliability (including perfectly reliable message delivery), and
(iii) participant behaviour (including universally honest participants).

---

[1] A specific instance is the CAP theorem [8, 10], which presupposes replicated global state as an architectural primitive.

Proof. Theorem 4.1 is a statement about categorical structure. It makes no assumptions about computation speed, network reliability, or participant behaviour. The non-existence of a copy-preserving, essentially surjective functor $F : \mathbf{Info} \to \mathbf{Value}$ is a structural property of the categories involved — not a limitation of any particular computational model or engineering context. Therefore, even under the idealised assumptions of infinite speed, perfect reliability, and universal honesty, replicated information does not determine value. The obstruction is semantic, not operational. No improvement in infrastructure eliminates the category error. □

5 Three Computational Consequences

5.1 Result 1: Subsumption

Theorem 5.1 (Strict Subsumption)**.** *The forgetful functor* $U : \mathbf{CMC} \to \mathbf{SMC}$ *is faithful. The inclusion* $\mathbf{CMC} \subsetneq \mathbf{SMC}$ *is strict.*

Remark 5.2. Cartesian computation is value transfer with the assumption that participants are trivial baked into the primitive operations. The assumption is valid for information. It is not valid for value.

5.2 Result 2: Recovery

Theorem 5.3 (Recovery)**.** *Let* $(\mathcal{C}, \otimes, I)$ *be an SMC with exponential comonad* $! : \mathcal{C} \to \mathcal{C}$*. Then the co-Kleisli category* $\mathbf{Kl}(!)$ *is Cartesian monoidal. Classical Cartesian computation embeds into the linear setting via the exponential [11, 6].*

Corollary 5.4. *A participant-indexed machine can execute every computation that an anonymous machine can execute. The converse is false.*

Remark 5.5. No expressive power is lost in the move from Cartesian to linear. Classical computation is the special case C_I of C_P.

5.3 Result 3: Structural Impossibility

Lemma 5.6. *Non-witness ordering over interactions, observable intermediate state, and atomic cross-context composition each require a natural* Δ *on state objects.*

Theorem 5.7 (Structural Impossibility)**.** *Let* C_P *be a participant-indexed SMC with* P *non-trivial. Then:*

(a) Non-witness ordering over participant interactions is structurally absent in C_P*.*

(b) Intermediate state is not observable during interaction execution in C_P.

(c) Atomic composition over states of independent participant contexts is structurally impossible in C_P.

These conditions arise in systems whose categorical semantics admit natural comonoids.

Proof. Each condition requires a natural Δ on state objects. By Fox's theorem, this forces C_P Cartesian — contradicting linearity. ☐

6 Emergent Property: Parallelism from Linearity

Theorem 6.1 (Parallelism from Linearity)**.** *In a participant-indexed SMC C_P, interactions whose participant sets are disjoint are categorically independent and admit parallel execution without global coordination.*

Proof sketch. In C_P, state is participant-indexed. Interactions t_1 on $\{P_1, Q_1\}$ and t_2 on $\{P_2, Q_2\}$ with disjoint participant sets share no state objects and have no categorical dependency. In **Info**, global state S introduces a shared dependency through Δ_S; parallelism requires partitioning S, reintroducing coordination at partition boundaries. ☐

7 The Replication Bridge

The objection: replication is copying; copying is Δ; distributed systems require replication; therefore distributed systems require Cartesian structure. The error lies in equating replication with duplication of value state.

Theorem 7.1 (Replication without Δ)**.** *Fault-tolerant agreement on participant-indexed state can be achieved by replicating attestations of agreement, not the state itself. State remains linearly governed. Attestations — being information — are freely copyable and live in C_I.*

The value layer is linear. The attestation layer is Cartesian. The Recovery theorem guarantees the linear machine can operate both simultaneously.

8 Hierarchy Inversion

The conventional view: value transfer is a constrained special case of information processing. The results of this paper establish an alternative hierarchy.

Information processing is a special case of value transfer. By Theorem 4.1, **CMC** $\subsetneq$ **SMC**. Adding comonoids specializes. Removing comonoids generalizes. C_I is the degenerate case. C_P is the general structure.

Copying is additional structure, not a natural right. In the general machine, Δ is absent. In the Cartesian special case, Δ is imposed on every object.

Linearity is the general condition. The overhead of Cartesian value systems is not the price of building value systems. It is the price of building value systems on a mismatched categorical primitive. Optimization within the Cartesian framework cannot eliminate this overhead within the same categorical framework.

9 What This Paper Does Not Do

This paper is intentionally limited in scope. The following explicit non-claims separate the semantic results from questions of implementation, design, or performance.

No protocol specification. This paper does not propose or specify any protocol for value transfer. No assumptions are made about network topology, message passing, ordering, or coordination mechanisms. The systems-level realisation of these semantics is the subject of CC3 [2] and CC4 [3].

No language or execution model. This paper does not introduce a programming language, type system, or execution model. While linear and participant-indexed semantics have implications for language design, such implications are not developed here.

No performance claims. This paper does not address efficiency, scalability, throughput, latency, or resource consumption. The results hold independently of computational cost.

No security analysis. This paper does not provide security guarantees or adversarial models. Byzantine behaviour, fault tolerance, and resistance to strategic manipulation are not analysed here.

No implementation claims. This paper does not claim that any existing system faithfully implements the semantics described. Concrete systems may approximate or realise these semantics in various ways, but their specification and analysis lie outside the scope of this work.

The goal of this paper is to identify and formalise the semantic requirements that distinguish value from information. Systems-level consequences follow from these requirements, but their exploration belongs to the companion papers in this series.

Computational Consequence. The categorical structure identified in this paper specifies the form of value states and establishes that any computation over such states must preserve participant-indexed linearity. Classical computational models, defined over anonymous state, do not satisfy this requirement. This reveals a structural gap between the domain of value and the existing models of computation.

A computational model addressing this gap — by treating the participant as a primitive component of the state and defining transitions over participant-indexed configurations — is developed in a companion work [2]. The present paper establishes the semantic domain and its structural constraints; the computational realisation follows from these constraints. In categorical terms, this work characterises the domain of value-bearing states, while CC3 develops the corresponding morphisms and their composition.

10 Open Questions

10.1 Formal Verification. Formalization in Agda or Lean using homotopy type theory, where linear and Cartesian types are native.

10.2 Quantitative Resource Theory. Modeling P as a graded monad or quantitative type theory [5, 12] for fine-grained resource tracking indexed by participant authority.

10.3 Complexity-Theoretic Implications. If parallelism is structurally unconstrained in the linear case, does participant-indexed computation admit efficient parallel algorithms for problems inherently sequential in the Cartesian case?

10.4 Quantum Computation. FHilb and **Value** both instantiate non-Cartesian SMCs. The connection between participant-indexed computation and quantum computational models deserves formal investigation [4].

10.5 The General Theory of Computation. The companion paper develops morphisms of **Value** and their linearity via $\int_P C_P$ over $\mathbf{P}$. The open question: does participant-indexed computation constitute a genuinely new complexity-theoretic primitive?

11 Conclusion

This paper establishes a precise categorical distinction between value and information, and proves that information is the degenerate special case of value — not its generalization.

Value is a non-separable joint state of Participant, Context, and Content. The binding is constitutive. Remove the participant and the value state is not incomplete — it is malformed. This places value objects in an SMC without natural comonoids, and information objects in a Cartesian monoidal category where every object carries one.

The Cartesian Degeneracy (Definition 3.5) makes the relationship precise. Information systems are value systems with $P = I$. Fox's theorem closes this: the comonoid is the mathematical form of the participant's absence.

The Domain Separation Theorem (Theorem 4.1) proves that no essentially surjective comonoid-preserving functor exists from **Info** to **Value**. Any system modeling value using information primitives must simulate linearity externally — not because of engineering deficiency, but because the categorical structure of those primitives actively requires what the value domain forbids.

This establishes that Cartesian structure is the strict specialization, and monoidal structure the general case. Copying is additional structure imposed on the general machine. Linearity is the general condition. The conventional framing — value transfer as a constrained special case of information processing — is the category error this paper addresses.

The present paper establishes the objects of **Value**. The morphisms — interactions between participant-indexed contexts, and the proof that these morphisms are linear — are developed in CC3 [2] via $\int_P C_P$ over $\mathbf{P}$. Together, objects and morphisms constitute the full categorical structure from which classical computation is recovered as the degenerate case $P = I$.

References

[1] Krishnan, A. (2026). On the Truth of Value: The Quantum Value Paper. *Contextual Compute Series, CC1. Sarva Labs Working Paper*, 2026.

[2] Krishnan, A. (2026). Contextual Compute: The General Theory of Computation. Zenodo. `https://doi.org/10.5281/zenodo.19177337`

[3] Krishnan, A. (2026). PIC: Operational Semantics of Participant-Indexed Computation. Zenodo. `https://doi.org/10.5281/zenodo.19189687`

[4] Abramsky, S., & Coecke, B. (2004). A categorical semantics of quantum protocols. *Proc. 19th Annual IEEE Symposium on Logic in Computer Science.*

[5] Atkey, R. (2018). Syntax and semantics of quantitative type theory. *Proc. LICS 2018.*

[6] Benton, P.N. (1995). A mixed linear and non-linear logic. *Proc. CSL 1994*, Springer LNCS 933.

[7] Coecke, B., Fritz, T., & Spekkens, R.W. (2016). A mathematical theory of resources. *Information and Computation*, 250, 59–86.

[8] Brewer, E.A. (2000). Towards robust distributed systems. *Proc. 19th Annual ACM PODC.*

[9] Fox, T. (1976). Coalgebras and Cartesian categories. *Communications in Algebra*, 4(7), 665–667.

[10] Gilbert, S., & Lynch, N. (2002). Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News*, 33(2), 51–59.

[11] Girard, J.-Y. (1987). Linear logic. *Theoretical Computer Science*, 50(1), 1–102.

[12] McBride, C. (2016). I got plenty o' nuttin'. *A List of Successes That Can Change the World*, Springer LNCS 9600.

[13] Melliès, P.-A. (2009). Categorical semantics of linear logic. *Panoramas et Synthèses*, 27, 1–196.

[14] Honda, K. (1993). Types for dyadic interaction. *CONCUR'93*, Lecture Notes in Computer Science, 715, 509–523.

[15] Wadler, P. (2012). Propositions as sessions. *ACM SIGPLAN Notices*, 47(9), 273–286.

[16] Seely, R.A.G. (1989). Linear logic, ∗-autonomous categories, and cofree coalgebras. *Categories in Computer Science and Logic*, AMS Contemporary Mathematics 92.

[17] Wootters, W.K., & Zurek, W.H. (1982). A single quantum cannot be cloned. *Nature*, 299, 802–803.

A Anticipated Objections

The following addresses objections a careful reader is likely to raise. The purpose is to sharpen the scope of the Domain Separation Theorem and clarify what it does and does not assert.

A. "This is just linear logic."

Objection. The incompatibility between Cartesian and linear categories is well-known from the semantics of linear logic. What is new here?

Response. The categorical machinery is standard. The contribution of this paper is not a new theorem in category theory but the identification of value as a semantic domain that necessarily inhabits linear structure. Linear logic provides the formal tools; this paper provides the domain identification. The observation that information is

Cartesian and value is linear — and that this distinction is semantic rather than a modelling choice — is the novel claim.

B. "Value is not conserved."

Objection. Value can be created through labour, lending, or monetary expansion. If value were truly linear it could never be created or destroyed.

Response. This conflates creation with copying. Linearity forbids the diagonal $\Delta : V \to V \otimes V$, which duplicates an existing value object. It does not forbid morphisms $f : \mathbf{I} \to V$ that introduce new value into the system. Creation (introducing new value through appropriate morphisms) is compatible with linearity. Copying (duplicating existing value without corresponding agreement) is not. Monetary expansion creates new value; it does not clone existing currency.

C. "Type systems can enforce linearity."

Objection. Linear type systems such as Rust's ownership model already enforce use-once semantics. The problem is solved at the language level.

Response. Linear types constrain programs; they do not change the semantic structure of the underlying computational substrate. A language with linear types running on a Cartesian machine still operates within Cartesian semantics at the substrate level. The type system prevents certain ill-formed programs but does not eliminate the underlying copyability of the runtime representation. This paper concerns structural linearity of the semantic domain, not type-level linearity of the programming language.

D. "Essential surjectivity is too strong."

Objection. Why require essential surjectivity? Partial embeddings may suffice for practical purposes.

Response. Relaxing essential surjectivity means accepting that some value objects have no representation in the information domain. These unrepresented objects must then be handled outside the computational model — precisely the "external enforcement" identified as symptomatic of semantic mismatch. Partial embeddings do not eliminate the problem; they relocate it.

E. Summary.

The Domain Separation Theorem asserts:

1. Information, as a semantic domain, is necessarily Cartesian.
2. Value, as a semantic domain, is necessarily linear and participant-indexed.
3. Modelling the latter with the former induces structural overhead (enforcement).

These claims survive the objections above without modification. The theorem is a statement about semantic structure, not about specific systems, protocols, or implementation strategies.