Web4: A Peer-to-Peer Δ-Based Electronic Cash System

KIM BYEONGYEON

November 2025

Abstract

This paper proposes Web4, a new digital economic model that operates without consensus mechanisms, ledgers, issuance, or inflation, while maximizing anonymity. The model defines interpersonal debt and claims—i.e, that is, the imbalance (Δ)—as the unit of value, functioning equivalently to conventional currencies such as the dollar. The network maintains equilibrium by enforcing the PDE-based Laplacian condition $Lx = 0$, ensuring that all imbalance automatically converges to a stable state. All transactions are negotiation-based and are validated only if they satisfy first-, second-, and third-order derivative constraints as well as integral constraints tracing back to prior states. Furthermore, the cryptographic stack—consisting of an extended XChaCha20, SHA-3, HMAC-SHA3, ZKP, Multivariate and TLS—neutralizes forgery, replay, tracing, interference, and most known attack models at their source. As a result, Web4 enables a fully functional digital reserve currency that operates without issuance, inflation, records, ledgers, or consensus.

1 Introduction

Most digital payment systems depend on ledgers, issuance, and consensus to maintain a shared state. While functional, these models inherit structural weaknesses: they require global agreement, expose transaction history, enable inflation, and rely on trusted infrastructure that limits anonymity and scalability.

A system that stores no history and requires no consensus avoids these limitations, but it requires a new method to ensure consistency without shared records. We propose Web4, a model in which value is defined as imbalance (Δ) between users, and network-wide consistency is enforced by a PDE-based Laplacian equilibrium $Lx = 0$. Transactions are validated only if they pass derivative and integral constraints that prevent artificial creation of Δ.

With a cryptographic stack incorporating Multivariate ,XChaCha20, SHA-3, HMAC-SHA3, ZKP, and encrypted transport, Web4 enables anonymous, issuance-free, inflation-free digital payments without ledgers, without consensus, and without historical records.

2 Background

Modern digital payment systems-whether traditional banking networks, blockchain-based ledgers, or zero-knowledge rollups-share a fundamental architectural assumption: they all rely on permanent records. Every model maintains a growing history of all prior actions, because the system can't determine its current state without referencing its past.

This assumption introduces structural limitations that no amount of optimization can eliminate.

First, record retention creates storage inflation. As systems grow, their historical data grows without bound, forcing participants to replicate or trust intermediaries who store that data on their behalf.

Second, any record-based model requires global agreement about the meaning of that record. Blockchains use consensus; banks use centralized reconciliation; rollups rely on sequencers or provers. Regardless of implementation, a system built on history must coordinate around its history, making consensus a mandatory cost.

Third, history inherently compromises anonymity. Even with mixers or ZKP-based aggregations, the existence of a traceable sequence creates a structural information leak. Metadata accumulates. Patterns emerge. Perfect anonymity is unattainable in any model that keeps records.

Fourth, record-based systems are slow to scale. As history increases, validation becomes more expensive, consensus becomes slower, and throughput plateaus. Improvements such as sharding or proof compression delay the problem but can't remove its root cause: the system is carrying its entire past.

Finally, history enables external manipulation. Censorship, reordering, MEV, front-running, and protocol-level interference all originate from the fact that transactions form a visible timeline.

In short: The limitations of modern digital money arise not from cryptography, bandwidth, or computation but from the very existence of history itself.

Web4 removes this dependency entirely. A system without records requires no global agreement, exposes no past actions, accumulates no metadata, and scales independently of its ago. Eliminating history is the only way to achieve long-term scalability, perfect anonymity, and a non-inflationary digital currency that does not rely on issuance.

Web4 therefore begins from a different premise: value is not stored as historical transactions, but as instantaneous inbalance (Δ) between users, enforced by a PDE-based equilibrium rather than a ledger.

3 Conceptual Overview

Most electronic payment systems depend on a foundational assumption: the current state of money can only be understood by referencing a permanent history of past transactions. Banking networks rely on ledgers, blockchains maintain immutable chains of records, and rollup systems store ordered traces. All of these architectures inherit the same structural dependency on history.

This dependency introduces three fundamental limitations. First, the size of the historical record grows without bound, creating permanent storage inflation. Second, the need for a shared interpretation of history forces the system to use consensus, central reconciliation, or trusted sequencing. Third, any model that embeds history inherently leaks metadata; anonymity becomes impossible as patterns, linkages, and sequences accumulate over time.

Web4 removes the dependency on history entirely.

Instead of viewing money as a chronological record of transactions, Web4 defines value as instantaneous imbalance, denoted Δ, between participants. When user A transfers value v to user B, the system updates:

$$\Delta_A \leftarrow \Delta_A - v, \qquad \Delta_B \leftarrow \Delta_B + v.$$

The global sum of imbalances remains zero:

$$\sum_{i=1}^{N} \Delta_i = 0,$$

representing a conservation law rather than a historical record. In this view, money is not "stored history" but the current distribution of imbalance across the network.

However, eliminating history raises a critical question: how does the system prevent forgery or double-spending without referencing past states? In Web4, consistency is enforced through a PDE-based equilibrium condition on the global imbalance vector $x = (\Delta_1, \ldots, \Delta_N)$:

$$Lx = 0,$$

where L is the network Laplacian. This condition guarantees that any valid update to x must be compatible with a global equilibrium. Artificial creation of value, replay of prior updates, or manipulation of Δ results in a vector that can't satisfy the Laplacian constraint, making forgery mathematically impossible.

Because consistency is enforced by differential structure rather than historical agreement, Web4 requires no consensus. Transactions become local negotiations that update only the relevant components of x, while the Laplacian ensures global compatibility. No participant must agree on an ordering, timeline, or shared record.

This framework also eliminates the need for issuance. Traditional systems define money as an issued unit that accumulates through historical balances. Web4 defines money as the shape of the imbalance vector itself; the sum of all

Δ is always zero, so supply is neither created nor destroyed. Value is not issued but emerges from relative imbalance between participants.

The cryptographic layer reinforces these guaranties. A stack combining XChaCha20, SHA-3, HMAC-SHA3, ZKP(zero-knowledge proofs), and multivariate signatures prevents replay, impersonation, and unauthorized updates to Δ. Even though Web4 stores no history, the system remains secure under all conventional attack models.

By removing history, avoiding issuance, eliminating consensus, and enforcing consistency through a PDE-based equilibrium, Web4 establishes a new paradigm for digital money: a system where value is not a record of the past but the current, constraint-governed state of a network.

4 Model

4.1 Value as Imbalance (Δ)

Web4 doesn't maintain balances, UTXOs, or historical account states. Instead, value is defined as instantaneous imbalance Δ across users.

When user A transfers a balue v to user B, the system applies:

$$\Delta_A \leftarrow \Delta_A - v, \qquad \Delta_B \leftarrow \Delta_B + v,$$

The global sum of imbalances is always zero:

$$\sum_{i=1}^{N} \Delta_i = 0.$$

This expresses a conservation law: no new value is created or destroyed. Since Δ represents only the current state (Not any historical record), Web4 requires no ledger, no timestamps, and no persistent transaction log.

Value in Web4 is therefore not stored history, but a momentary state vector representing the current distribution of imbalance across the network.

4.2 Negotiation-Based Updates

Web4 doesn't rely on global consensus. Each interaction is a local negotiation that updates only relevant entries in the imbalance vector.

If users A and B negotiate a transfer v, the protocol applies:

$$\Delta_i^{\text{new}} = \begin{cases} \Delta_i - v, & \text{if } i = A, \\ \Delta_i + v, & \text{if } i = B, \\ \Delta_i, & \text{otherwise.} \end{cases}$$

No global synchronization is required. Only the updated imbalance values matter; no information about ordering or history is recorded.

4.3 State Representation

The system maintains a state vector:

$$\mathbf{x} = (\Delta_1, \Delta_2, \ldots, \Delta_N)$$

subject to the conservation constraint:

$$\mathbf{1}^\top \mathbf{x} = 0.$$

Thus Web4 behaves not as a ledger but as a PDE-constrained state machine where the current imbalance vector fully defines the monetary state.

5 Examples

This section illustrates how Web4 updates the imbalance vector x under valid and invalid transaction scearios. All examples demonstrate that the system requires no history, no global ordering, and no consensus; correctness is determined solely by compatibility with the Laplacian equilibrium condition $Lx = 0$.

5.1 Basic Two-Party Transfer

Consider users A and B with initial imbalances Δ_A and Δ_B. If A transfers value v to B, the system updates only the two relevant components of the imbalance vector:

$$\Delta'_A = \Delta_A - v, \qquad \Delta'_B = \Delta_B + v.$$

The global conservation law holds:

$$\sum_{i=1}^{N} \Delta'_i = 0.$$

No history or ordering information is required. The updated vector $x' = (\ldots, \Delta'_A, \Delta'_B \ldots)$ must satisfy:

$$Lx' = 0,$$

ensuring compatibility with the global equilibrium.

5.2 Multi-Party Chain Transfer

Suppose A, B, and C form a chain of transfers:

$$A \xrightarrow{v_1} B \xrightarrow{v_2} C$$

The updates are:

$$\Delta'_A = \Delta_A - v_1, \qquad \Delta'_B = \Delta_B + v_1 - v_2, \qquad \Delta'_C = \Delta_C + v_2$$

Again the conservation law holds:

$$\sum_{i=1}^{N} \Delta'_i = 0$$

The Laplacian constraint $Lx' = 0$ ensures that no inconsistency arises even though no global ordering of events is defined. Only the final state vector matters.

5.3 Forgery Attempt

Assume an attacker attempts to introduce a forged imbalance $+v$ for user A:

$$\Delta'_A = \Delta_A + v$$

Let e_A denote the standard basis vector where the A-th component is 1. The updated vector is:

$$x' = x + ve_A$$

Applying the Laplacian gives:

$$Lx' = Lx + vLe_A.$$

Since $Lx = 0$ for any valid state, we obtain:

$$Lx' = vLe_A.$$

Because $Le_A \neq 0$ for all valid Laplacians, the forged update cannot satisfy $Lx' = 0$. The system rejects the transaction without requiring consensus or history.

5.4 Double-Spend Attempt

Suppose A attempts to spend the same value v twice, once to B and once to C, without decreasing Δ_A twice.

The attacker proposes:

$$\Delta'_B = \Delta_B + v, \qquad \Delta'_C = \Delta_C + v, \qquad \Delta'_A = \Delta_A - v.$$

The global sum would be:

$$\Delta'_A + \Delta'_B + \Delta'_C = (\Delta_A - v) + (\Delta_B + v) + (\Delta_C + v) = (\Delta_A + \Delta_B + \Delta_C) + v.$$

The conservation law is violated by $+v$. Therefore x' fails:

$$Lx' = 0.$$

The inconsistency is detected purely from the imbalance vector; no reference to any past state is needed.

5.5 PDE Violation Example

Consider a candidate update x' that satisfies conservation but violates the local continuity encoded by the Laplacian. For instance, assume a transfer creates a sharp discontinuity such that for some node i:

$$(Lx')_i \neq 0.$$

Because the Laplacian aggregates local second-order differences, any discontinuity indicates a non-physical imbalance injection. Thus x' cannot belong to the nullspace of L and is rejected.

These examples demonstrate that Web4 validates transactions through mathematical consistency rather than historical agreement. All invalid states produce Laplacian inconsistencies, while valid updates preserve the equilibrium $Lx = 0$ without requiring ordering, consensus, or history.

6 Security Analysis

Web4 achieves security without ledgers, timestamps, or consensus. All forms of validation arise from two components: (1) the conservation law

$$\sum_{i=1}^{N} \Delta_i = 0,$$

and (2) the Laplacian equilibrium constraint

$$Lx = 0.$$

Any update that violates either condition is rejected without reference to past states. This section demonstrates how Web4 prevents forgery, replay, double spending, manipulation, or interference through these intrinsic constraints.

6.1 Forgery Prevention

Assume an attacker attempts to create value by injecting a forged imbalance $+v$ for user A:

$$\Delta'_A = \Delta_A + v.$$

Let e_A denote the standard basis vector. The resulting state is

$$x' = x + ve_A.$$

Applying the Laplacian yields:

$$Lx' = Lx + vLe_A.$$

Since valid states satisfy $Lx = 0$, we obtain:

$$Lx' = vLe_A.$$

For all nontrivial Laplacians, $Le_A \neq 0$. Therefore $Lx' \neq 0$, meaning x' cannot lie in the nullspace of L. The system rejects the forged state without requiring consensus, historical comparison, or ordering.

6.2 Double-Spending Prevention

Suppose A attempts to spend the same value v twice, sending v to both B and C while reducing Δ_A only once:

$$\Delta'_B = \Delta_B + v, \qquad \Delta'_C = \Delta_C + v, \qquad \Delta'_A = \Delta_A - v.$$

The global sum becomes:

$$\Delta'_A + \Delta'_B + \Delta'_C = (\Delta_A - v) + (\Delta_B + v) + (\Delta_C + v) = (\Delta_A + \Delta_B + \Delta_C) + v.$$

The conservation law is violated by $+v$. Consequently, x' cannot satisfy $Lx' = 0$. Double spending is mathematically impossible: invalid updates are detected independently of time, history, or global agreement.

6.3 Replay Attack Resistance

A replay attack requires reusing a prior update to induce an inconsistent state transition. Web4 stores no history, so a replayed update is simply a new candidate vector x'.

If the replayed update corresponds to a valid transfer, then it must be reapplied to the current Δ values. However, applying a previously-valid update in a new context generally violates either:

$$\sum_i \Delta'_i = 0 \quad \text{or} \quad Lx' = 0.$$

Thus the system rejects it. If the update does not violate these constraints, it is indistinguishable from a legitimate new transfer. Since all transfers produce no metadata, replay provides no advantage to an attacker.

6.4 MITM Manipulation Failure

A man-in-the-middle attacker attempting to alter a message in transit must modify the proposed imbalance updates. Any change that alters the value of v or reassigns it to a different participant modifies the vector x'.

For any illegitimate modification:

$$Lx' \neq 0.$$

In addition, the cryptographic layer (XChaCha20, HMAC-SHA3, ZKP, and multivariate signatures) guarantees that the attacker cannot produce a valid signature for the modified update. MITM attacks therefore fail both cryptographically and mathematically.

6.5 Consensus Attacks Are Impossible

Consensus-based systems are vulnerable to majority or reordering attacks, including 51% attacks, bribery attacks, or sequencer censorship. Web4 does not define consensus or global ordering; each update is validated locally through $Lx' = 0$.

Thus there is no mechanism that resembles:

majority voting, block production, chain reorganization, or ordering.

Consensus attacks do not apply because consensus does not exist in Web4.

6.6 Network Partition Robustness

A network partition temporarily divides the set of users into two groups. Since Web4 stores no global ledger and does not require global agreement, the two partitions update their local Δ values independently.

When the network reconnects, the combined state x' must still satisfy:

$$\sum_i \Delta_i' = 0, \qquad Lx' = 0.$$

If any partition produced inconsistent updates, these updates fail the Laplacian test and are discarded. The system therefore tolerates partitions without requiring reorganization or rollback.

6.7 Cryptographic Integrity

The PDE constraints enforce mathematical consistency, while the cryptographic stack enforces participant authenticity. Web4 uses:

- XChaCha20: prevents state inference and replay.
- SHA-3 and HMAC-SHA3: prevents message alteration.
- Zero-knowledge proofs: validate correctness of updates without revealing identity.
- Multivariate signatures: resist classical and post-quantum forgery.

Although Web4 does not store history, cryptographic integrity ensures that only legitimate participants can propose updates to Δ. The Laplacian ensures that illegitimate updates cannot exist within the state space.

6.8 Summary

Web4 achieves security through structural constraints, not historical records. Forgery, double spending, replay attacks, MITM interference, and consensus attacks all fail because invalid imbalance vectors cannot satisfy:

$$\sum_i \Delta_i = 0 \quad \text{and} \quad Lx = 0.$$

Security emerges from mathematical impossibility rather than trust, consensus, or global synchronization.

7 Information-Theoretic Anonymity

Web4 achieves anonymity not through obfuscation or mixing of historical records, but through the structural absence of history itself. Since the system maintains no ledger, no timestamps, and no ordering of events, an adversary cannot observe any information that links a user's identity to a specific update. This section presents an information-theoretic argument demonstrating that user identities are indistinguishable under all possible observation models.

7.1 Reductio Argument: Impossibility of Trace Construction

Assume that an adversary attempts to trace a user by observing the network-level messages exchanged during state updates. Let the adversary attempt to construct a mapping between a user identity I and some observed message sequence M.

1. a sequence of updates forming a trace,
2. an ordering or timestamp relation,
3. a persistent address or public identifier,
4. a structural pattern in the imbalance vector.

Web4 exposes none of these. All updates modify only the relevant components of the imbalance vector without producing ordering metadata or persistent identifiers. Since the system does not store history and does not define global time, any attempt to reconstruct a trace requires information that does not exist.

Thus, the assumption that a trace can be constructed leads to a contradiction. Therefore, user tracing is impossible.

7.2 Observation Model

Let I denote the random variable representing the user identity, and let M represent all possible messages an adversary can observe, including encrypted payloads, signatures, and imbalance updates after verification.

Since updates are validated solely through the equilibrium condition $Lx = 0$, messages contain no information about ordering or identity. Every valid message is a cryptographically authenticated proposal for updating Δ without revealing the origin of the update.

The adversary's observation M consists of:

$$M = \{\text{ciphertexts},\ \text{ZK-proofs},\ \text{imbalance increments}\}$$

None of these elements contains identity-dependent variation. The chiphertexts are uniformly distributed under XChaCha20, ZK-Proofs, reveal no metadata, and imbalance depend solely on transaction amounts, not on identities.

7.3 Mutual Information Analysis

Anonymity requires that observation of M yields no information about I. Formally:

$$I(I; M) = 0,$$

where $I(\cdot\,;\cdot)$ denotes mutual information.

Since Web4 does not expose persistent identifiers, timestamps, or historical dependencies, the conditional of M is identical for all users:

$$P(M \mid I = i) = P(M \mid I = j) \quad \forall\, i, j$$

Thus, M and I are statistically independent:

$$P(I \mid M) = P(I)$$

Substituting into the definition of mutual information:

$$I(I; M) = \sum_{i,m} P(i, m) \log \frac{P(i, m)}{P(i)P(m)} = 0$$

No observation can reduce uncertainty about a user's identity.

7.4 Unlinkability of Updates

Let updates be $u_1, u_2, \ldots, u_k$ and let the adversary attempt to determine whether two updates originate from the same user.

Since updates are defined solely as modifications to the components of x, and since each update is recomputed as a fresh, context-independent message:

$$P(u_a, u_b \text{ linked}) = P(u_a, u_b \text{ unlinked})$$

Because no deterministic or probabilistic structure connects updates to specific identities, the unlinkability is unconditional.

7.5 Summary

Web4 achieves information-theoretic anonymity because:

- no ledger, timestamp, or ordering information exists,
- imbalance updates contain no identity metadata,
- all cryptographic elements are uniformly distributed,
- the system exposes no persistent identifiers,
- the joint distribution of observable data is independent of identity.

Consequently:

$$I(I; M) = 0,$$

and anonymity holds under all adversarial models, including those with unbounded computational power. Web4 provides unconditional unlinkability and full identity indistinguishability as a structural property of the system.

8 References

Satoshi Nakamoto, 2008. "Bitcoin: A Peer-to-Peer Electronic Cash System." https://bitcoin.org/bitcoin.pdf