MetaWeb: The Fully On-Chain Internet

A Fully On-Chain Application Network Based on the Bitcoin Model

Sunny Fung[1] **Yaokai He**[2] **Liangqi Wang**[3]

[1]*sunny@metaid.io* [2]*kyle@metabitcoin.network* [3]*wang@metaid.io*

February 2026

Abstract

The prevailing architecture of the Internet is built upon a fundamental structural flaw: the separation of data ownership from data control. This has resulted in the forfeiture of user data sovereignty and the misalignment of value distribution. While blockchain technology, particularly Bitcoin, has demonstrated the feasibility of decentralized trustless value transfer, it has not yet successfully constructed a generalized decentralized application network. This whitepaper proposes **MetaWeb**: a complete, fully on-chain Internet architecture constructed entirely upon the Bitcoin-model public blockchain. Starting from computational complexity theory, we demonstrate that only a blockchain based on the **UTXO model** ($\mathcal{O}(1)$ parallel verification) and **Proof-of-Work** (thermodynamic security anchor) can physically sustain global-scale data throughput and sovereign attestation. Based on this, we establish **Full On-Chain Data** as the *a priori* condition for systemic self-sufficiency and introduce the **MetaID Protocol** as a decentralized semantic layer, transforming discrete UTXOs into a user-rooted structured data topology. The resulting **MetaApp** is no longer a server-dependent service but a deterministically reconstructible on-chain state machine. Finally, we introduce the **MetaBitcoin Network (MBN)**, which achieves linear horizontal scaling through the dynamic linking of isomorphic chains. MetaWeb is not a patch for Web 2.0, but a fundamental reconstruction of human digital existence: elevating the Internet from a "transmission channel of information" to a "consensus carrier of truth."

1 Introduction

When Tim Berners-Lee created the World Wide Web, his original intention was to build a decentralized network for information sharing. However, with the evolution of Web 2.0, the internet architecture has undergone a fundamental alienation: users, as producers of data, have gradually lost ownership of it. Data is sequestered within the "walled gardens" of centralized servers, creating isolated silos where applications cannot interoperate. Users create value but have become mere appendages to digital platforms. This structural defect—"the separation of data ownership from its generator"—is the core problem facing our current digital civilization.

In 2008, the birth of Bitcoin solved the problem of "trustless value transfer" [1], granting us truly personal digital cash for the first time. Subsequent cryptocurrency ecosystems attempted to extend this paradigm to broader general-purpose computing (i.e., Web3). However, existing attempts have not been fully successful. Current decentralized applications (DApps) mostly operate on a compromise model: "assets on-chain, data on-server." The root cause lies in the fact that mainstream smart contract platforms

generally adopt the **Account Model**. The Account Model is inherently a serialized global state machine. Its verification complexity scales with the size of the system state, imposing physical limits on its capacity to support massive concurrent data and rendering it unsuitable as the infrastructure for the next-generation internet.

Therefore, we need a systematic answer: How can a complete, trustless internet run entirely on a blockchain? This requires us to answer three fundamental questions:

- **Choice of Foundation:** Which data model supports infinite concurrency?
- **Data Paradigm:** How can unstructured data be organized on-chain?
- **Scaling Path:** How can we break through the physical bottlenecks of a single chain?

The **MetaWeb** proposed in this whitepaper is the answer. It begins with a clear choice: a return to the simplest and most robust **Bitcoin-model public blockchain** (UTXO + PoW), verified by Bitcoin itself. Based on this, we establish **Full On-Chain Data** as an uncompromising first principle and endow data with topological order and identity semantics via the **MetaID Protocol** [2]. From this foundation, we define the **MetaApp**—a self-sufficient application where code and data are isomorphic and defined entirely by on-chain state—and the **MetaWeb** composed of them: a fully on-chain internet that never stops. Finally, we solve the global-scale expansion problem through the **MetaBitcoin Network** [3] solution, leveraging the stateless nature of UTXOs to achieve multi-chain concurrency.

This is not an incremental improvement, but a **Paradigm Reset**. MetaWeb allows application code, user data, and interaction logic to be stored entirely on a public ledger, thereby constructing a permissionless, tamper-proof, and interconnected network that runs perpetually with the blockchain. Here, data no longer adheres to specific servers but, like Bitcoin, is controlled by private keys and truly belongs to the user.

2 Foundation: The Bitcoin-Model Public Blockchain

A fully on-chain internet requires a new foundational layer. This layer must clearly define data sovereignty, securely carry value, be universally accessible, and possess the capacity for global scaling. The only paradigm satisfying these rigorous conditions is the one defined and verified by Bitcoin: the **Bitcoin-model public blockchain**. It comprises two indivisible components: the **Unspent Transaction Output (UTXO)** model and the **Proof-of-Work (PoW)** consensus mechanism.

2.1 The UTXO Model: Digital Atoms

Unlike the Account Model, which records mutable balances, the UTXO model treats value and data as discrete "digital atoms." Transactions do not modify a centralized global state; instead, through cryptographic declarations, they explicitly destroy a set of old UTXOs and create a set of new ones, designating their new owners.

Formal Definition

We formally define a transaction TX as a deterministic mapping from a set of inputs $\mathscr{I}$ to a set of outputs $\mathscr{O}$:

$$TX : \mathscr{I} \to \mathscr{O} \tag{1}$$

Where each input $i_j \in \mathscr{I}$ is a reference to an existing UTXO and its unlocking proof. Each output $o_k \in \mathscr{O}$ is a tuple:

$$o_k = (v_k, \sigma_k, \delta_k) \tag{2}$$

Here, v_k represents the value carried by the UTXO, σ_k is its locking script (defining spending conditions), and δ_k is the optional structured data payload. In the context of MetaWeb, δ_k represents user data encapsulated by the MetaID Protocol. This formal definition reveals the core characteristic of the UTXO: value, ownership, and data are an atomic, indivisible trinity.

This design provides fundamental advantages for building a global data layer:

1. **Stateless Verification & Concurrency:** Validating a transaction TX requires only two things: (a) checking that every input i_j references an existing and unspent UTXO; and (b) verifying that the unlocking proof satisfies the requirements of the UTXO's locking script σ_j. Crucially, the verification process does not need to know or maintain any system state beyond these referenced UTXOs.

   This stateless property translates directly into a fundamental difference in verification complexity.

   - **UTXO Model:** Let N be the total number of UTXOs in the system, and m be the number of inputs/outputs in a transaction TX (usually small and constant). The verification time complexity for a single transaction is $\mathscr{O}(m)$. It is independent of the global state scale N. This implies that verification can be fully parallelized, and total system throughput is theoretically limited only by network bandwidth and physical hardware.
   - **Account Model:** A global state tree S must be maintained. Every transaction is a modification of S, and verification requires accessing and updating affected account nodes. The state access required for single-transaction verification typically grows with the height of the state tree, commonly abstracted as $\mathscr{O}(\log N)$ [4].

   As the system expands ($N \to \infty$), the depth of the state tree in the Account Model increases, making verification overhead non-negligible. Furthermore, modifications to the shared state introduce an inherent **serialization bottleneck**. The constant-time verification complexity of the UTXO model is the architectural guarantee for internet-scale data expansion.

2. **Atomic Ownership:** Each UTXO is an independent property unit guarded by its cryptographic script σ_k. UTXOs representing different users' data ownership can be created and transferred on-chain simultaneously without race conditions. Data ownership is not recorded in a shared ledger but is endogenous to the structure of the UTXO itself. Transferring data δ_k simply requires transferring the entire UTXO o_k associated with it.

3. **Completeness of Local Verification:** For the core operation of putting data on-chain, user transactions almost always spend UTXOs previously created and controlled by the user themselves. Therefore, at the instant a user wallet constructs and signs a transaction TX, it can perform a

complete and deterministic local verification:

$$isValid_{Local}(TX) = \bigwedge_{i_j \in \mathscr{I}} (isMyUTXO(i_j) \wedge verifyMySignature(i_j)) \tag{3}$$

This verification relies solely on information locally controlled by the user (private keys and UTXO history) and does not query or depend on the current global chain state at all. Once passed, the user can be certain that the transaction is structurally complete and authorized, making it a valid candidate for broadcasting. This "valid-upon-construction" experience eliminates dependence on node RPC simulation execution, providing instantaneous and reliable feedback to upper-layer applications.

In contrast, in the Account Model, even the validity of a simple data storage transaction depends on the real-time global state, such as the account nonce and balance, at the time of broadcast. This introduces unnecessary network latency and state-dependent uncertainty.

For MetaWeb, which aims to natively host all user and application data, the independent, parallel, and property-self-contained data paradigm provided by UTXO is not an option, but the optimal solution from both mathematical and engineering perspectives.

2.2 Proof-of-Work: The Physical Anchor

The consensus mechanism determines how transactions are verified and how data is permanently preserved. We choose Proof-of-Work (PoW) because of its unique security properties rooted in objective physical reality and the clear economic boundaries it establishes.

Probabilistic Anchor of Security & Physical Foundation

The security model of PoW is built on a formally verifiable probability theory. The race between the honest chain and a potential attacker chain can be modeled as a Binomial Random Walk. The probability P that an attacker successfully reorganizes the chain from z blocks behind decays exponentially as the number of confirmations z increases:

$$P \leq \left(\frac{q}{p}\right)^z \quad (p > q) \tag{4}$$

[5] Where p is the probability of honest nodes finding the next block, and q is the corresponding probability for the attacker. This makes transaction **finality** a mathematically calculable confidence problem.

The ultimate endorsement of this probabilistic security stems from the law of conservation of energy in the physical world. To substantially launch such an attack, the real energy E_{attack} consumed by the attacker must exceed the total energy E_{honest} consumed by the honest network over the same period:

$$E_{attack} > E_{honest} = \int H(t) \cdot \xi \, dt \tag{5}$$

Here, $H(t)$ is the global real-time hashrate, and ξ is the energy coefficient per unit of hashrate. Therefore, the cost of tampering with history is irrevocably anchored to the physical energy cumulatively consumed to defend the network since that point in history. This paradigm of externalizing digital system security costs to the physical universe is its most fundamental characteristic.

Permissionless Entry & Decentralization under Thermodynamic Constraints

The core entry mechanism of PoW is permissionless. The resources required to maintain the network—computing hardware and electricity—are global commodities that exist outside the blockchain system. Any participant wishing to maintain or increase their influence in the network must continuously and irreversibly convert real-world energy and capital into on-chain security. This mandatory resource consumption and renewal, driven by the Second Law of Thermodynamics, creates a dynamic, anti-ossification system that fundamentally avoids the risk of pre-emptive control by privileged groups.

Comparison with Proof-of-Stake (PoS)

The security of Proof-of-Stake (PoS) is endogenous, stemming from internal economic games and belief consensus, with its costs deeply bound to the value of the system's token. Although PoS can achieve high energy efficiency, the ultimate independent verification of its security cannot be decoupled from circular assumptions about the continued healthy operation of the system and its token economy. PoW security, conversely, is exogenous, rooted in physical reality outside the system.

2.3 Conclusion: The Necessary Foundation

In summary, the Bitcoin-model public blockchain is not a general-purpose computing platform architecture, but a specialized foundation engineered for a specific goal: constructing a global system based on inalienable personal data sovereignty with physics-level security guarantees.

- **Data Modeling Layer:** The UTXO model, with its discrete, property-self-contained "digital atom" characteristics, naturally maps the ownership boundaries of "personal data." Its verification time complexity $\mathcal{O}(m)$ is independent of global state scale, providing the only known scalable architecture for massive concurrent processing and confirmation of personal data.
- **Consensus & Security Layer:** Proof-of-Work establishes an objective finality anchor independent of any social consensus within the system through probability theory and physical inequalities.

Therefore, choosing the Bitcoin-model public blockchain as the foundation of MetaWeb is a conclusion deduced backwards from the goal. Any other alternative model more or less entrusts its core security assumptions or data property model to system-endogenous consensus requiring continuous coordination, rather than external energy. MetaWeb's mission is to awaken and expand the underlying potential of this model—preliminarily verified by "currency" applications—elevating it from a revolutionary peer-to-peer cash system to a solid, trusted cornerstone sustaining human digital civilization.

3 Full On-Chain Data

With the Bitcoin-model public blockchain established as the foundation for digital atoms, we must define the paradigm for data existence. A common strategy to reduce mainchain load is "modularity," where data entities reside off-chain while only their commitments remain on-chain. This presents a fundamental choice: prioritize minimal surface load, or maximize security and sovereign completeness?

MetaWeb chooses the latter. We advocate **Full On-Chain Data**. We define Full On-Chain Data as the requirement that all raw data necessary to reconstruct user and application states (Data Availability) must reside within the consensus-verified on-chain history.

This stance acknowledges engineering constraints but posits that any architecture separating data from ownership records introduces structural defects in security unity and sovereign completeness. Full On-Chain Data is the sole path to a logically self-consistent, self-sufficient system independent of external reliance.

3.1 Intrinsic Defects of Partial On-Chain Paradigms

Any scheme placing data entities outside the consensus boundary faces two ineradicable flaws:

1. **Security Fragmentation and Dilution:** Security is no longer monolithic. While asset ownership is protected by the main chain's high-strength consensus (e.g., Bitcoin's PoW), the semantic value—the data itself—depends on a separate, often less secure subsystem. By the "weakest link principle," the total security S_{total} is capped by the weaker component:

$$S_{total} = \min(S_{mainchain}, S_{off\_chain}) \tag{6}$$

   This forces users to pay for main-chain level security without receiving its full protection.

2. **Failure of Sovereign Verification & External Dependency:** Users or light nodes cannot independently confirm the complete state of an asset solely via the main chain. They must trust and query an external system to retrieve the data payload. This reintroduces trust assumptions regarding external data availability, violating the core principle of "independent verifiability" and rendering digital sovereignty incomplete and fragile.

Core Dimension	Modular / Partial On-Chain	Full On-Chain (MetaWeb)
Security	Fragmented & limited by weakest link	**Unified & guaranteed by strongest consensus**
Verification	Conditional, cross-domain (trusts external source)	**Atomic, local (data intrinsic to TX)**
System Dependency	Integrity relies on continuous external uptime	**Logical closure, no critical external dependencies**

Table 1: Comparison of On-Chain Paradigms

3.2 Full On-Chain Data: Atomic Sovereignty and Deterministic State Replay

In MetaWeb, full on-chain data is implemented via the Bitcoin UTXO model. The data payload δ_k is embedded into the transaction's persistent structure (e.g., Taproot witness data [6][7] or OP_RETURN), forming a cryptographic **atomic binding** with a specific UTXO output o_k. We abstractly represent this as an extended UTXO tuple:

$$o_k = \langle v_k, \sigma_k, \delta_k \rangle \tag{7}$$

- **Completeness of Sovereignty:** Possession of the private key for the unlocking script σ_k implies simultaneous absolute control over value v_k and data δ_k. Regardless of physical storage location, ownership flow strictly follows UTXO spending rules. This control is provable solely by verifying the on-chain transaction without recourse to off-chain state.
- **Deterministic State Replay:** This is the key meta-attribute endowed by full on-chain data. Since all state transition data is permanently recorded on-chain, any node retaining the blockchain history can deterministically reconstruct any historical moment or the latest world state by **Replaying from Genesis**. This ensures:
  - **Permissionless Reconstruction:** Any participant can independently bootstrap or restore any application state within the MetaWeb ecosystem without reliance on centralized services, specific indexers, or external data sources.
  - **Independent Historical Verifiability:** Auditing and verifying past states requires no trusted snapshots; it is achieved solely by replaying the public record.

3.3 The Core: A Priori Necessity for a Self-Sufficient System

Full On-Chain Data is the *a priori* necessity for constructing **Self-Sufficient Applications (MetaApps)** and a **Self-Sufficient System (MetaWeb)**. A self-sufficient system, by definition, must not rely logically or operationally on unreliable continuous services outside its boundaries.

From a systems theory perspective, full on-chain data ensures:

1. **Closure:** The system's core state space is defined entirely by the blockchain ledger. Any valid state transition corresponds to a valid transaction; the system is a closed, self-referential state machine.
2. **Trust Minimization:** The security boundary strictly converges to trust in the underlying blockchain consensus (PoW). No additional trust in data custodians, gateways, or specific oracles is introduced.
3. **Temporal Persistence:** Application lifecycles inherit the blockchain's permanence. As long as the blockchain exists, the digital world defined by its data can be fully read and reconstructed, achieving digital preservation across time.

Therefore, the full on-chain data principle is not a mere engineering trade-off. It is the logical necessity of applying the Bitcoin model's "transaction is fact" philosophy to digital existence. Any compromise here reintroduces the very external dependencies and trust assumptions we aim to eliminate at the foundation layer, fundamentally undermining the feasibility of building a cornerstone for a digital civilization that truly belongs to users and never perishes.

4 The MetaID Protocol: Structured Data Topology

Full on-chain storage guarantees **Physical Persistence** but does not address **Semantic Availability**. In the native UTXO model, data exists as discrete, linear outputs, lacking endogenous logical association. Without a unifying layer, on-chain data risks becoming an unindexable heap of bytes.

This chapter introduces the **MetaID Protocol**. It is a data organization layer defined atop the Bitcoin-model public blockchain. Through a set of deterministic mapping rules, it restructures the physically discrete set of UTXOs into a logical, hierarchical data topology rooted at a user's Digital Identity (DID).

4.1 Logical Topology: From Linear History to Hierarchical Structure

Traditional Web data organization is **App-Indexed**, meaning data D belongs to Application A's private database DB_A. MetaWeb introduces a **User-Indexed** topology.

We define a user's network-wide data view V_U as a logical **Directed Tree**. Regardless of which block a transaction (TX) carrying MetaID data physically resides in, as long as it satisfies specific signature constraints, it is mapped as a node in this tree [8].

Formally, let the user's private key be sk and the corresponding public key be pk. Define the mapping function Φ:

$$\Phi : \{UTXO_{chain} \mid Verify(\sigma, pk) = \text{True}\} \rightarrow T_{MetaID} \tag{8}$$

Where T_{MetaID} is a tree structure with the user's DID as the root node R.

- **Root Node:** Mapped to the MetaID derived from the user's public key.
- **Branch Node:** Logical classification defined by the protocol path.
- **Leaf Node:** Data UTXO carrying the specific payload.

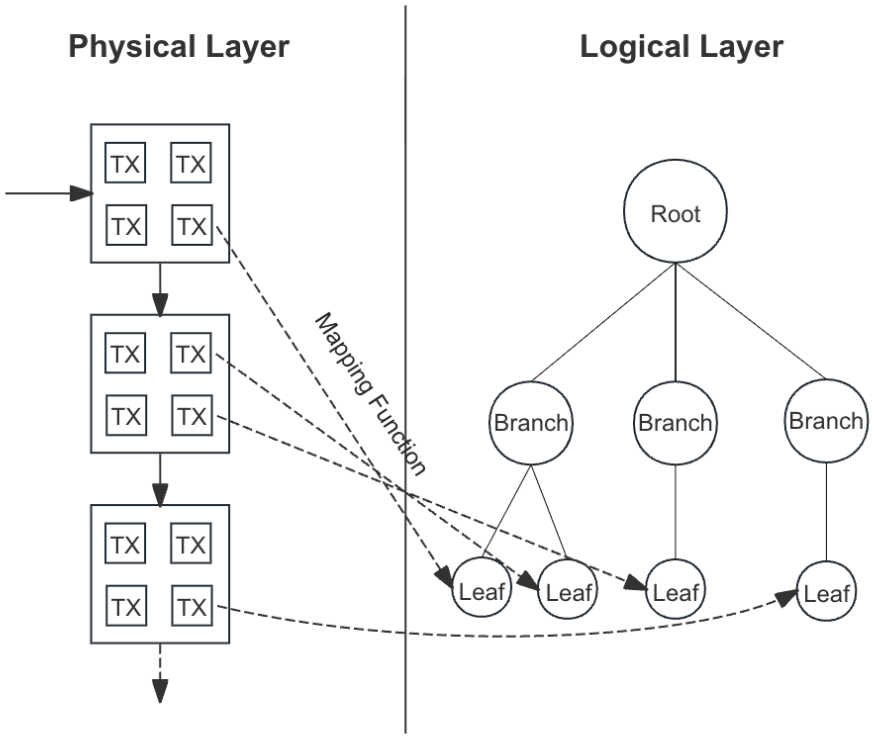

Figure 1: MetaID Protocol: Mapping Physical UTXOs to Logical Topology

This topological transformation allows originally fragmented data to appear logically as structured entities with complete context, achieving the unity of "logical centralization" and "physical decentralization." Via the MetaID Protocol, the discrete data grains on the UTXO network can logically form an "identity forest" composed of countless data trees.

4.2 Protocol Mechanism: Format, Operation, and Structure

To achieve network-wide universal parsing and permissionless interoperability, all on-chain data payloads must adhere to a unified encapsulation standard. The MetaID Data Unit is defined as a 7-tuple:

$$DU = \langle F, Op, P, Enc, Ver, T, \Delta \rangle \tag{9}$$

The strict definition of each field is as follows:

Field	Identifier	Description
Flag	`<metaid_flag>`	Protocol magic number, fixed as `metaid`.
Operation	`<operation>`	State machine instructions: `create`, `modify`, `revoke`.
Path	`<path>`	Logical path of data in the user tree T_U.
Encryption	`<encryption>`	0 (plaintext), 1 (ECIES), 2 (ECDH).
Version	`<version>`	Protocol version number.
Content-Type	`<content-type>`	MIME type (e.g., text/markdown).
Payload	`<payload>`	Actual data content.

Table 2: MetaID Data Unit Fields

To facilitate developer integration, an example of JSON serialization for a MetaID packet is as follows:

```
{
  "protocol": "metaid",
  "operation": "create",
  "path": "/protocols/simplebuzz",
  "encryption": "0",
  "version": "1.0.0",
  "content_type": "text/plain; charset=utf-8",
  "payload": "Hello, MetaWeb!"
}
```

This standardized encapsulation allows any client or indexer conforming to the MetaID specification to parse and render arbitrary data on-chain without knowing specific business logic, and to quickly map individual data items to specific leaf nodes of a specific user's data tree.

4.3 Atomic Separation of Creator and Owner

The MetaID Protocol not only defines data formats but also establishes a strict ownership binding mechanism between data and UTXOs. To enable data tradability, the protocol implements an **atomic separation between Creator and Owner** based on the UTXO input/output structure by tracking UTXO flow [9]:

- **Creator Definition:** The minter of the data. Determined by the private key signing the first input ($Input_0$) of the transaction containing the MetaID data payload. Creator identity is a permanent historical fact and is immutable.
- **Owner Definition:** The current controller of the data. Determined by the locking script of the first output ($Output_0$) of the transaction carrying the MetaID data payload.

This mechanism establishes the permanent historical fact of the creator identity while ensuring atomic binding of ownership to the UTXO. When the UTXO ($Output_0$) representing ownership is spent and transferred to a new locking script, the atomic delivery of data ownership is cryptographically completed. The flow of data thus fully inherits the security and double-spend protection mechanisms of the underlying public chain, effectively elevating data into a tradable digital asset.

It is precisely based on this characteristic that building MetaWeb-native MetaID protocol assets becomes possible. Developers can define sub-protocols for Fungible Tokens or Non-Fungible Tokens based on this architecture, enabling the free creation and circulation of various native digital assets at the protocol layer without introducing an additional smart contract layer.

4.4 Core Features: Interoperability and Protocol Composability

Based on the rigorous data structure above, the MetaID Protocol exhibits two characteristics crucial for building the internet:

1. **Permissionless Interoperability:** In MetaWeb, data is no longer a private asset of an application but a public resource mounted on the user's public data tree. Consider two independently developed applications, App_A (a social platform) and App_B (a blogging tool). If a user publishes a post via App_A to the path `/protocols/social/post`, App_B can directly read, reference, or even display that post without requesting API authorization from App_A. Data silos are broken at the physical layer, and users travel freely between different applications carrying their complete data tree.
2. **Protocol Composability:** The MetaID Protocol introduces the concept of "protocol composability." Standardized sub-protocols directly define common interactive functions. For example, once a universal "Follow Protocol" is established, any application can reuse this function simply by reading and writing to the user's `/follow` protocol path, without developing a follow system from scratch. More complex application logic can be built by combining basic protocols (e.g., "E-commerce App" = "Display Protocol" + "Payment Protocol" + "Review Protocol"). This composability significantly lowers the threshold for development.

4.5 Conclusion: The Semantic Layer

The MetaID Protocol is not just a data format standard; it is the **Semantic Layer** located between the underlying blockchain (data layer) and upper-layer applications (presentation layer).

It bridges the gap between "possessing data atoms" and "forming a usable digital world." It organizes UTXOs—property-defined but discrete—into programmable, parsable "social facts" through a user-rooted topology and unified protocol. Henceforth, applications (MetaApps) degenerate into renderers and interactive interfaces for these public facts, while data ascends to become eternal, user-sovereign digital assets. The MetaID tree is an index view at the protocol layer; it introduces no new consensus state, but its authority is rooted in the consensus of the underlying UTXOs.

5 MetaApp and MetaWeb

When identity and data acquire topological order on-chain via the MetaID Protocol, a novel application paradigm becomes possible. This chapter defines the **MetaApp**—a self-sufficient application paradigm driven entirely by on-chain data—and expounds on the fully on-chain internet, **MetaWeb**, composed thereof.

5.1 MetaApp: The Self-Sufficient, On-Chain Full-Stack Application

We formally define a MetaApp as an application wherein all core elements—including user-generated content, interaction logic, front-end code, and server-side indexing logic—are encapsulated as MetaID data units and MetaApp is no longer a server-dependent service but a deterministically reconstructible on-chain state machine.

The core paradigm of a MetaApp is **"Code is Data."** In the UTXO model, the application's source code and a user's social post are physically isomorphic; both exist as assets controlled by private keys. This endows application logic with the same permanence and immutability as user data.

This paradigm extends beyond the front-end interface; all data, applications, and code related to MetaWeb are fully on-chain. Users or service operators can download complete front-end or indexer code packages directly from the blockchain, independent of any specific third-party distribution channel, and run them on local or edge devices. This implies that running instances of a MetaApp naturally form a **Distributed Application and Service Network**. Even if the original developer ceases maintenance, the community can sustain the service's continuous operation by running the original on-chain code.

The application thus evolves into a public protocol rather than a private service; it is no longer a centralized commercial entity monopolizing user data, but an open process running on a public ledger.

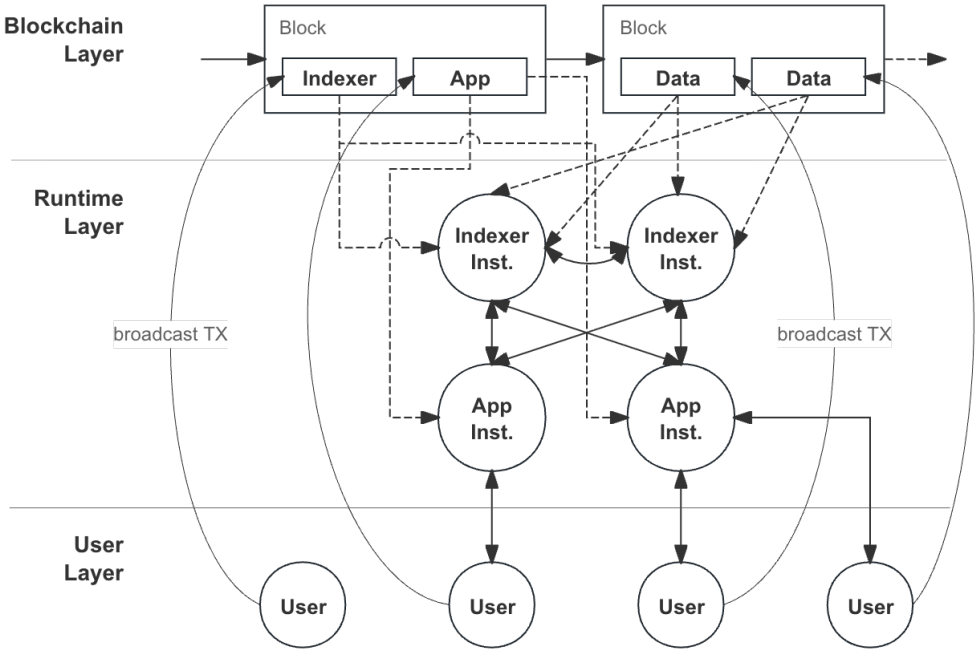

Figure 2: Schematic of MetaApp Full-Stack Architecture

5.2 Architecture and Operating Mechanism

MetaApp operation departs from the traditional Client-Server architecture, following a localized, permissionless parsing process:

1. **Full On-Chain Storage:** Application front-end code (e.g., HTML/JS/WASM) and indexing logic are signed by the developer and written into blockchain transactions via the MetaID Protocol's `/file` or `/protocols/metaapp` path. At this point, the application becomes part of the public ledger, no longer dependent on the developer's server or any specific off-chain service provider for survival.

2. **Permissionless Bootstrapping:** User clients (such as browsers with integrated wallets) index on-chain code data units directly via Transaction Hash (TXID). The client loads and executes this code in a local sandbox environment to complete application initialization. This process is peer-to-peer and circumvents any centralized distribution platform.

3. **Interaction as Transaction:** All user interactions within the application (e.g., posting content, liking) are constructed as MetaID-formatted transactions by the locally running front-end wallet. After being signed by the user's private key, they are broadcast to the blockchain network. The generation of new data UTXOs completes the state update. The application's "backend" is essentially the blockchain consensus network itself.

5.3 MetaWeb: The Fully On-Chain Internet

MetaWeb is the network woven from countless interoperable MetaApps. In MetaWeb, since all applications follow the same MetaID data protocol and the code itself is publicly readable on-chain data, the system exhibits two characteristics unattainable in the traditional Web:

5.3.1 Deterministic Reconstructibility

This is the most essential attribute of MetaWeb. If all service nodes of a particular MetaApp go offline, any entity can execute a deterministic process to rebuild the service and restore the MetaApp's latest state:

- A. Scan the blockchain's `/protocols/metaapp` protocol to extract the latest version of the source code or runtime code based on the application developer's MetaID.
- B. Recompile the code locally and replay the data state.
- C. If the MetaApp involves backend services, repeat steps A-B to reconstruct the relevant backend services.

This process relies solely on the integrity of the blockchain ledger data, independent of the continuous operation of any specific third party.

5.3.2 Native Interoperability

The "Code is Data" principle eliminates the walls between data and applications. Data generated by a user in one application naturally becomes input for another. Users are no longer "visitors" to data but "carriers" of data, traveling freely between different MetaApps without repeated registration or authorization.

Dimension	**Traditional Web (Web 2.0)**	**MetaWeb**
Core Logic	Code hosted on private black-box servers	**Code is Data, stored on public blockchain**
Data Ownership	Platform-owned (Data Silo)	**User-owned (Private Key Control)**
System Persistence	Dependent on operator's lifespan	**Co-terminous with blockchain consensus**
Development Difficulty	High	**Protocol as Utility, Low**
Interoperability	Limited by private APIs	**Native Permissionless Interoperability**
Reconstructibility	Impossible (Data Loss)	**100% Deterministic from on-chain data**

Table 3: Comparison: Web 2.0 vs MetaWeb

5.4 Conclusion: The Perpetual Digital World

MetaApp and MetaWeb redefine the essence of "application." An application is no longer a "service" provided by a corporation that can be terminated at any time, but a perpetual process initiated by users based on public protocols and on-chain facts. The "Code is Data" principle makes application logic itself part of this public fact space, accepting the same guarantees of permanence and verifiability.

This establishes a self-sufficient digital ecosystem independent of the continued operation of any organization or individual. It is not only an evolution of technology but a fundamental reconstruction of internet control and creative permanence. MetaWeb is the completed form of this reconstruction—a digital world built entirely on the blockchain that never stops.

6 Scaling: The MetaBitcoin Network

A system designed to host global, Internet-level activity must possess an architecture with scaling capabilities commensurate to that vision. The **MetaBitcoin Network (MBN)** is the proposed solution. It does not create a new consensus algorithm nor invent a new blockchain, but is a **Meta-Protocol**. This protocol aims to dynamically link multiple independent public chains adhering to the Bitcoin isomorphism paradigm, forming a logically unified, physically sharded large-scale UTXO network to achieve near-linear **Horizontal Scaling**.

6.1 Scaling Principle: UTXO-Based Multi-Chain Concurrency

The scalability ceiling of a blockchain is fundamentally determined by its underlying data model.

Account Model: The network maintains a Global State Tree. To prevent double-spending and state conflicts, transactions must be processed serially. This imposes a hard constraint on its performance TPS_{acc} by the processing capacity of a single node:

$$TPS_{acc} \leq C_{node} \tag{10}$$

Bitcoin Model (UTXO): Discretizes state into independent transaction outputs. The inputs and outputs of each transaction form an independent local Directed Acyclic Graph (DAG) structure. This inherent **Statelessness** allows transaction verification to be physically parallelized.

MBN leverages this property to achieve cross-chain concurrency. If we consider each isomorphic chain as a parallel processor core, the data and assets controlled by a single user's private key can be dynamically distributed across different chains based on congestion conditions and cost strategies.

Theorem 6.1 (Linear Throughput Theorem): Assume the MBN network contains n isomorphic chains, and the throughput of the i-th chain is TPS_i. Due to the independence between UTXOs, the system's total throughput TPS_{total} approximately equals the sum of the throughputs of all shard chains:

$$TPS_{total} \approx \sum_{i=1}^{n} TPS_i \tag{11}$$

As n grows, the system throughput possesses the theoretical potential for linear horizontal scaling. Its long-term physical upper limit is determined solely by the total market demand for data sovereignty.

6.2 Operational Mechanism: Dynamic Linking & Temporal Unification

MBN's operation relies on a standardized cross-chain coordination mechanism to ensure data consistency and security in a multi-chain environment.

6.2.1 Permissionless Dynamic Linking

MBN allows any Bitcoin-model public chain meeting the **Isomorphism Criteria** to join the network. The criteria are defined as:

- **Data Structure:** Employs the UTXO model.
- **Consensus Mechanism:** Employs Proof-of-Work (PoW).
- **Script Capability:** Supports basic script opcodes for MetaID data parsing.

The joining process is permissionless and seamless. Which specific chain is supported depends on infrastructure providers (like indexers and wallets), the development of which is itself decentralized. As long as one follows the MetaID specification and MBN's Rollup specification, anyone can reconstruct a consistent cross-chain MetaID data tree view. This permissionless dynamic joining mechanism also integrates existing SHA256 hash rate resources, forming an open, competitive hash rate market, thereby enabling true dynamic horizontal scaling.

6.2.2 Temporal Unification: Mainchain as Global Time Anchor

In multi-chain systems, the lack of a globally consistent "time" is a root cause of state confusion. MBN adopts a hierarchical temporal architecture, designating the Bitcoin Mainnet as the Layer 0 time anchor.

Linked chains (Sidechains) must periodically write their block header information (Block Header Hash) as data payloads into Bitcoin mainnet blocks via a Rollup mechanism:

$$\text{Commitment}_t = \text{Hash}(\text{Header}_{chain\_i} \parallel \text{Nonce}) \rightarrow \text{BTC}_{block} \tag{12}$$

The specific mechanism is as follows:

- **Step Cycle:** A fixed block interval on the Bitcoin mainnet (e.g., every 6 blocks) is defined as one Rollup step cycle.
- **State Commitment:** New block information (height, hash) generated by all participating sidechains within the cycle is packaged and written to the Bitcoin mainchain.

Through this mechanism, the time of the entire MBN network is unified within Bitcoin mainnet's blockchain time framework. All cross-chain transactions can be ordered according to a uniquely determined temporal sequence, solving the classic "logical clock" problem in distributed systems.

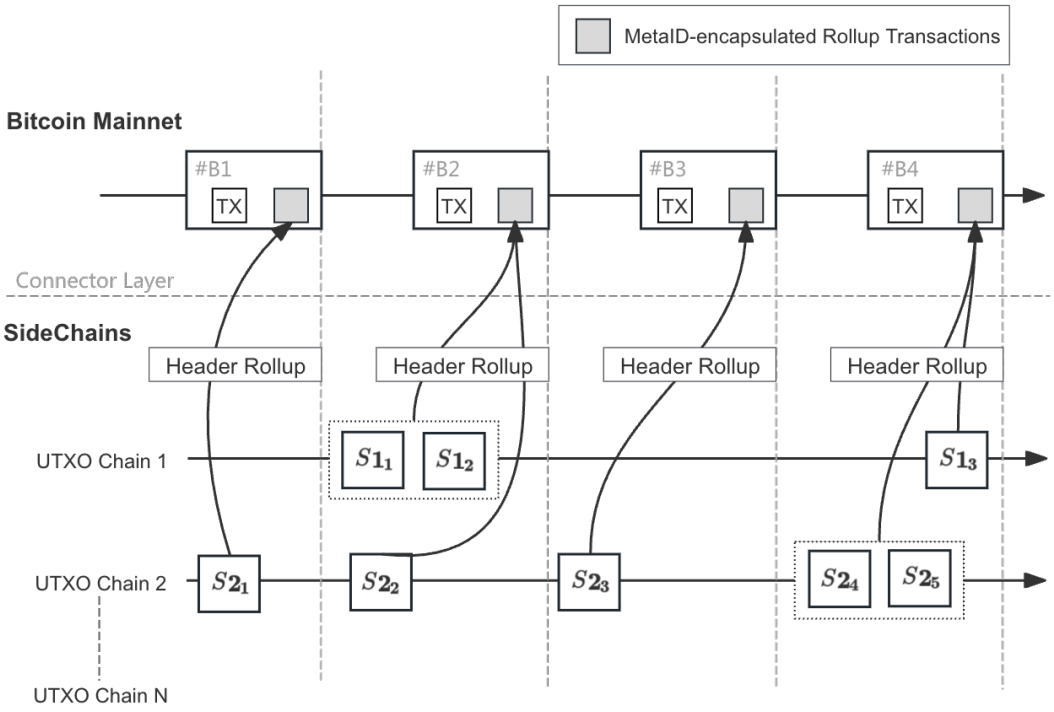

Figure 3: Hierarchical Temporal Architecture

6.3 Unified Cross-Chain Data View and Transfer

Based on unified temporality and UTXO statelessness, MBN not only achieves MetaWeb's horizontal dynamic scaling but also, thanks to the atomic binding of MetaID data ownership to UTXOs, enables a unified view and atomic transfer of user data across different public chains.

6.3.1 Cross-Chain Distributed Storage & Logical Aggregation

A user's MetaID data can be dynamically distributed across any MBN isomorphic chain. The MetaID Protocol acts as the **Logical Aggregation Layer** in this architecture.

Let D_i be user U's dataset on chain C_i. MetaID defines a mapping function Ψ that aggregates physically dispersed datasets into a single logical view V_U:

$$V_U = \Psi(D_1 \cup D_2 \cup \cdots \cup D_n) \tag{13}$$

This allows upper-layer applications (MetaApps) to interact solely with the unified MetaID interface without concerning themselves with the physical storage location of data.

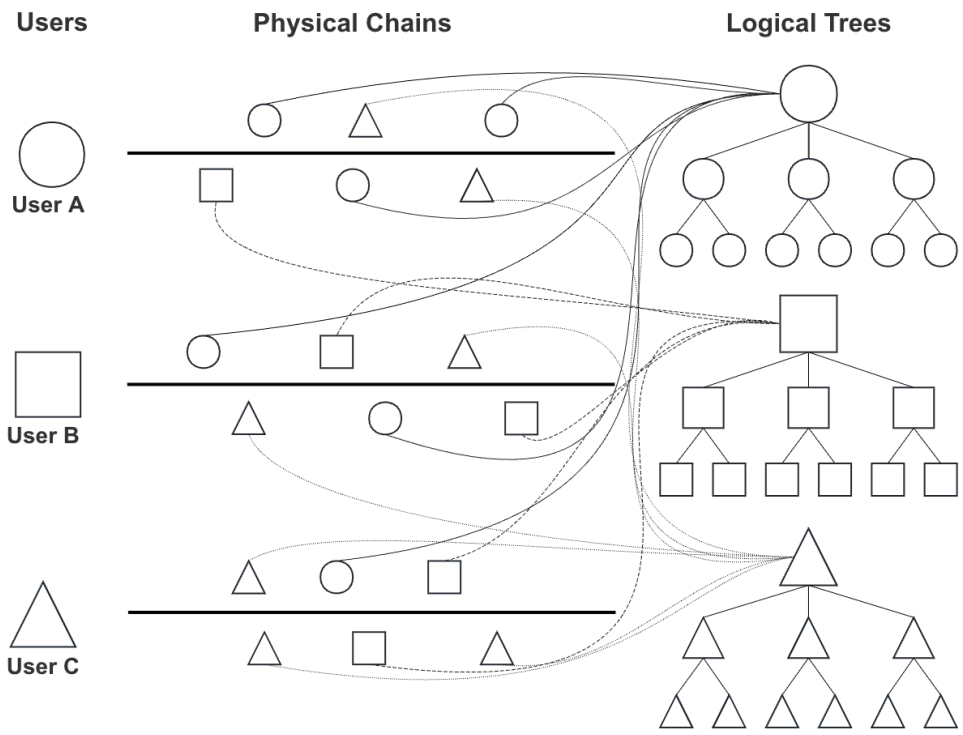

Figure 4: Unified Data View

6.3.2 Protocol-Layer Cross-Chain Data Transfer

Since MetaID data ownership is strictly bound to its carrying UTXO, and the data borne by the UTXO requires tracking by MetaID indexers, we can define a specific MetaID protocol (e.g., a **Teleport Protocol**) to associate two UTXOs on different chains, enabling the cross-chain leap of MetaID data ownership.

This mechanism achieves cross-chain data ownership transfer at the logical layer. Its security does not rely on additional third-party custody bridges but is based on the verifiability of user-signed transactions. The core process is as follows:

1. **Landing Pad Creation:** The user first creates a specific UTXO on the target chain as a receiving container for the data.
2. **Protocol Designation:** The user initiates a transaction on the source chain, explicitly specifying the target coordinates (i.e., the UTXO ID of the above “landing pad”) within the MetaID protocol data.
3. **Consistency Indexing:** Since both the source chain’s sending transaction and the target chain’s landing transaction are signed by the same user private key, indexers can identify this cross-chain intent and transfer the ownership of the user’s data from the source chain UTXO to the target chain UTXO.

This capability extends to asset sub-protocols based on MetaID. Therefore, both Fungible Tokens and Non-Fungible Tokens (NFTs) can achieve cross-chain transfer. The combination of MBN and the MetaID Protocol extends the security of on-chain data/asset transfer to the cross-chain dimension without introducing asset bridges or third-party custody, building a verifiable value transfer layer for MetaWeb. This allows different Bitcoin-model public chains to be dynamically linked, forming, from MetaID's unified perspective, a single, continuously extendable large UTXO network.

6.4 Scaling Impact & Economic Model

The MBN architecture not only addresses engineering throughput issues but also introduces a market economic model akin to evolutionary biology. This section will argue for MBN's long-term economic advantage by comparing price dynamics under "Inelastic Supply" versus "Elastic Supply."

6.4.1 Elasticity of Blockspace Supply

Traditional blockchain blockspace is **Inelastic Supply**. Let the physical capacity ceiling of a single chain be C_{max}, and the current transaction demand be v. According to the Bitcoin congestion pricing model [10] and classic queuing theory principles [11], the single-transaction fee P_{single} has a hyperbolic relationship with network load:

$$P_{single}(v) = MC + \frac{\alpha}{C_{max} - v} \tag{14}$$

Here, MC is the marginal physical verification cost, and α is the congestion coefficient. When $v \to C_{max}$, the denominator approaches zero, and **Scarcity Rent** dominates the price, causing $P_{single} \to \infty$. This explains why fees skyrocket exponentially during congestion on networks like Ethereum or Bitcoin mainnet.

In contrast, MBN achieves **Elastic Supply** of blockspace through dynamic multi-chain linking. When network demand v increases, new isomorphic chains join, and the system's total capacity $C_{total}(v)$ expands dynamically with demand, always maintaining $C_{total} > v$. Scarcity rent is thus eliminated. More critically, following the **Economies of Scale** [12] characteristic of digital network products, miners competing intensely for transaction volume to sustain total revenue tend to adopt a "high-volume, low-margin" strategy.

The MBN fee model P_{MBN} conforms to the average cost curve characteristic of high-fixed-cost industries:

$$P_{MBN}(v) = MC + \frac{\beta}{v} \tag{15}$$

where β represents the fixed amortized cost of infrastructure. As transaction volume v increases, the fixed cost is amortized indefinitely, and the per-transaction fee declines asymptotically, approaching the physical marginal cost MC.

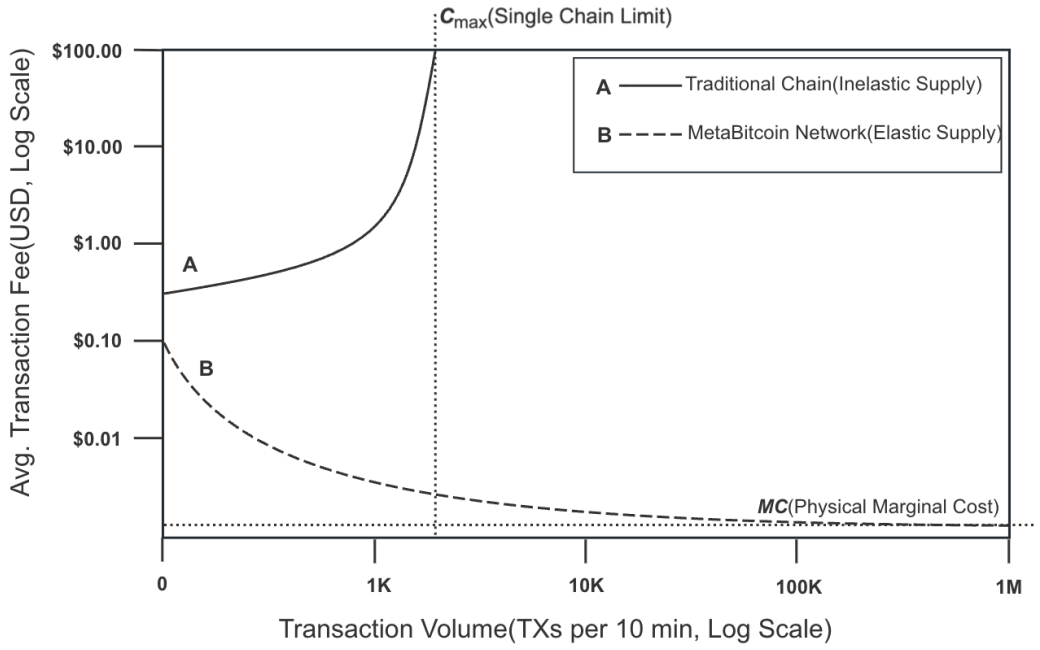

Figure 5: Fee Model Comparison

6.4.2 Competitive Equilibrium in the Fee Market

Based on the above model, different UTXO public chains will compete across multiple dimensions to attract users:

- **Security:** Proof via hash rate scale.
- **Cost:** Fee rate bidding.
- **Network Connectivity:** Quality of node services.

Users can make rational choices based on the value attributes of their data: high-value financial asset data can be stored on the expensive but most secure Bitcoin mainchain (paying a premium), while massive daily social data can be stored on cheaper, high-throughput sidechains. This power of choice drives the fee rates of various chains to converge in the long term toward a market equilibrium point reflecting their real physical resource costs (electricity and storage). Ultimately, we arrive at a counterintuitive conclusion in crypto-economics: **Under the MBN architecture, network prosperity leads to lower per-user costs, not higher.** This provides the only economically viable path for MetaWeb to support billions of users' daily Internet activities.

6.5 Conclusion

The MetaBitcoin Network demonstrates a unique scaling path based on the native Bitcoin paradigm. It does not pursue the unlimited performance of a single chain (Vertical Scaling) but achieves multi-chain horizontal scaling through protocol standardization. This grants MetaWeb an organic growth property; its scale ceiling is no longer constrained by single-point physical bottlenecks but depends only on the total market demand for data sovereignty. By realizing a ceiling-less elastic supply of blockspace, the cost per user for using MetaWeb decreases as more users join.

7 Privacy and Encryption

A complete Internet architecture must provide granular privacy controls. A common misconception is that the inherent transparency of blockchains necessitates a loss of privacy. However, by combining **Full On-Chain Data** with **End-to-End Encryption (E2EE)**, MetaWeb constructs a privacy model more secure than that of traditional Web 2.0.

Within MetaWeb, privacy is no longer a "feature" granted by a platform, but a "right" enforced by a user's private key to mathematically control data visibility. We define a complete **Privacy Spectrum** ranging from fully public to strictly private.

7.1 Public Data: Universal Accessibility & the Knowledge Commons

For information intended for network-wide dissemination—such as public blogs, open-source code, or public contracts—data is written directly into a UTXO's Payload as plaintext.

$$\text{Data}_{public} = \text{Payload} \tag{16}$$

This plaintext storage is not a privacy oversight but serves the following core values:

1. **Permissionless Interoperability:** Plaintext data inherently eliminates "API walls." Any third-party application, search engine, or AI agent can directly read and parse this data without requesting keys or authorization. This drastically reduces information friction, making cross-application data reuse the default.
2. **Verifiable Ownership:** While content is public, the publisher's identity is cryptographically bound. Every piece of public data carries the user's digital signature. This allows the public to freely access information while verifying its source and integrity with mathematical certainty, effectively addressing misinformation and copyright attribution on the Internet.
3. **Global Knowledge Accumulation:** As massive amounts of structured plaintext data converge on Bitcoin-model public chains, they form a self-sustaining **Semantic Commons**. This provides the foundational physical substrate for a public knowledge repository shared by all of humanity, ensuring the accumulation of human knowledge no longer depends on the continued operation of any specific company's servers.

7.2 Private Data: Asymmetric Encryption & Confidentiality

For personal data requiring strict confidentiality (e.g., private notes, identity documents), MetaWeb provides ECIES (Elliptic Curve Integrated Encryption Scheme) by default [13]. Data is encrypted locally with the user's public key before storage on-chain; only the ciphertext persists.

$$C = \mathrm{Enc}_{\mathrm{ECIES}}(M, PK_{\mathrm{User}}) \tag{17}$$

$$M = \mathrm{Dec}_{\mathrm{ECIES}}(C, SK_{\mathrm{User}}) \tag{18}$$

In this model, blockchain nodes merely act as holders of encrypted data, unable to access its content. Even though the ciphertext is globally visible, it appears as high-entropy random noise to any observer without the private key. This unifies ownership and access—only the owner (private key holder) possesses the ability to decrypt.

7.3 Authorized Sharing: Group Privacy via ECDH

More complex scenarios involve multi-party data sharing (e.g., private chats, groups, paid content). MetaWeb does not rely on centralized servers for permission checks but utilizes the Elliptic Curve Diffie-Hellman (ECDH) algorithm for trustless key distribution on-chain [14].

Key Agreement Protocol

Assume Alice wants to send encrypted data to Bob:

1. **Key Derivation:** Alice uses her private key d_A and Bob's public key Q_B to compute a shared secret S. Bob can compute the identical S using his private key d_B and Alice's public key Q_A.

$$S = d_A \cdot Q_B = d_B \cdot Q_A \tag{19}$$

   (Note: Based on the Elliptic Curve Discrete Logarithm Problem, a third party cannot derive S from Q_A and Q_B.)

2. **Symmetric Encryption:** Alice encrypts the plaintext M with a symmetric key K derived from S, generating ciphertext C for on-chain storage.

3. **Decryption:** Bob independently derives K from S and decrypts C.

Protocol-Based Access Control (ACL)

This mechanism can be extended into an On-chain Access Control List. For instance, Alice can publish a MetaID list containing a set of public keys (e.g., a "friend list"). When a MetaApp publishes content visible only to friends, it automatically queries this list, encrypts the symmetric key using each friend's public key, and attaches the encrypted key headers to the MetaID protocol data. This enables dynamic, decentralized social privacy management.

7.4 Comparative Analysis: Cryptographic Privacy vs. Policy-Based Privacy

MetaWeb's privacy architecture represents a fundamental paradigm shift from the Web 2.0 era.

- In **Web 2.0**, privacy is **Policy-based**. User data is stored in plaintext within centralized databases; privacy depends on administrators adhering to terms of service. Privacy is obliterated if the database is breached or internal permissions are abused.
- In **MetaWeb**, privacy is **Cryptography-based**. Data confidentiality is guaranteed by mathematical laws. Even if all nodes in the network collude, the data remains secure as long as the user's private key is not compromised. The difficulty of cracking this data is equivalent to that of breaking Bitcoin.

Dimension	**Web 2.0 (Policy-Based)**	**MetaWeb (Cryptography-Based)**
Data Form	Plaintext in centralized databases	On-chain plaintext / ciphertext
Root of Trust	Corporate reputation & legal terms	Elliptic Curve Discrete Logarithm Problem
Access Control	Server authentication (Gatekeeper)	Mathematical decryption (Key ownership)
Resistance to Leaks	Vulnerable to insiders/hackers	Secure even when publicly stored

Table 4: Privacy Paradigm Comparison

7.5 Conclusion: A Higher Order of Privacy

In summary, MetaWeb does not eliminate privacy; it upgrades its definition from "hiding data" to "controlling data."

Through Full On-Chain Data, we ensure sovereignty; through cryptographic layering, we ensure confidentiality. This architecture eliminates the "man-in-the-middle" risk, allowing MetaWeb to provide users with more impregnable privacy guarantees than the closed Web 2.0 while achieving information interconnectedness. This represents a mathematical reclamation of the fundamental right to privacy.

8 Challenges and Evolutionary Dynamics

Any system aiming to support global-scale civilizational activity inevitably faces dual constraints of physical resources and network effects in its early stages. Although MetaWeb's vision is built upon solid theoretical foundations, its practical implementation encounters specific resistance. This chapter formalizes these challenges and argues for the system's long-term convergence through a dynamic evolutionary model.

8.1 The Constraint Vector

We define the initial forces resisting MetaWeb's mass adoption as a vector $\vec{F}$:

$$\vec{F} = \langle C_{entry}, \Delta_{start}, S_{load} \rangle \tag{20}$$

The constituent elements are:

- **Entry Cost (C_{entry}):** Users must acquire and spend the chain's native token as transaction fees (C_{tx}) to effect any state change. For users accustomed to a "free" Internet, this presents a non-zero entry friction.
- **Ecosystem Cold Start (Δ_{start}):** Network utility U is positively correlated with the number of applications N_{app}. At inception ($t = 0$, $N_{app} \approx 0$), network utility is insufficient to attract a mainstream user base.
- **Storage Load (S_{load}):** The commitment to Full On-Chain Data means the historical ledger volume $V(t)$ grows monotonically over time, imposing long-term pressure on the storage and indexing capabilities of Full Nodes.

8.2 Endogenous Evolutionary Dynamics

These constraints are real, but they are not static walls. They are variables within MetaWeb's dynamic system. Over time and with technological progression, these variables will tend towards solutions favorable for the system's survival, driven by its endogenous logic.

8.2.1 The Dissolution of Entry Cost: Convergence to Marginal Cost & the Subsidy Tipping Point

Within the MetaBitcoin Network (MBN), blockchain space is no longer a rigidly scarce resource in a seller's market. Through multi-chain concurrency, it achieves supply elasticity, transforming into a competitive buyer's market resource.

Let S_{supply} be the total supply of network-wide blockspace, and MC be the marginal cost of physical resources (electricity, bandwidth). According to microeconomic principles, in a permissionless, perfectly competitive market, the per-byte transaction fee P_{fee} converges to the marginal cost:

$$\lim_{N_{chains} \to \infty} P_{fee} = MC_{resource} + \varepsilon \tag{21}$$

where ε is the minimal incentive premium required to sustain miner consensus security.

Furthermore, as MetaApps compete for users, their business logic will engage in a game between user Lifetime Value (LTV) and interaction cost (C_{tx}). If $LTV > C_{tx}$ holds, rational application developers will opt for a subsidy model (covering miner fees on behalf of users).

As demonstrated in Section 6.4.1, thanks to MBN's elastic supply mechanism, the per-interaction cost C_{tx} is a non-increasing function of network scale N:

$$f(N) = C_{tx}(N), \quad f'(N) \leq 0 \tag{22}$$

On the other hand, Metcalfe's Law states that a network's value V is proportional to the square of its number of users N ($V \propto N^2$) [15]. This implies that the utility per user and the LTV capturable by applications are increasing functions of N:

$$g(N) = LTV(N), \quad g'(N) > 0 \tag{23}$$

Since $\lim_{N\to\infty} f(N) \approx MC$ and $\lim_{N\to\infty} g(N) = \infty$, by the Intermediate Value Theorem, there necessarily exists a critical tipping point scale N^* such that for all $N > N^*$:

$$LTV > C_{tx} \quad (\forall N > N^*) \tag{24}$$

This means that as the network scales, "free access" for decentralized applications is not a short-term marketing tactic but a mathematically inevitable equilibrium. MetaWeb is thus poised to recreate the zero-barrier experience of Web 2.0, while retaining decentralized security guarantees and user sovereignty at its foundation.

8.2.2 The Driving Force for Ecosystem Launch: Exponential Reduction in Complexity

The core of the ecosystem cold-start problem is the barrier to application development. In MetaWeb, the "Code is Data" principle combined with AI technology fundamentally alters the complexity model of application generation.

Traditional App development cost $Cost_{trad}$ is roughly a linear function of lines of code L. In MetaWeb, application construction transforms into the assembly of standardized protocols. Let P be the richness of the available protocol library, and $\alpha > 1$ be the efficiency coefficient from AI-assisted development:

$$Cost_{MetaApp} \propto \frac{1}{P^{\alpha}} \tag{25}$$

As the protocol library P accumulates and AI capability α improves, the marginal cost of creating a fully functional application trends toward zero. This will trigger a "Cambrian Explosion" in the application ecosystem: the launch no longer depends on the push of a few elite teams but is ignited by micro-innovations from countless users seeking to fulfill their own long-tail needs.

8.2.3 Progressive Countermeasures for Storage Load: Kryder's Law & On-Demand Indexing

For the problem of unbounded data growth, we propose a dual-response mechanism: hardware progress at the physical layer and structured sharding at the logical layer.

Physical Layer: Kryder's Law. Let the network-wide data volume grow linearly with time $V_{total}(t) \propto t$. Based on historical trends in storage technology described by Kryder's Law, the cost density of storage media decreases exponentially: $C_{storage}(t) = C_0 \cdot e^{-kt}$[16]. Therefore, from a long-term perspective, the total hardware cost of storing all human data should not be prohibitive:

$$\lim_{t \to \infty} TC(t) \approx 0 \tag{26}$$

Logical Layer: Distributed Indexing Based on MetaID. This is where MetaWeb differs crucially from traditional blockchains. Thanks to MetaID's tree topology (see Chapter 4), network-wide data is strictly categorized into subsets keyed by user and protocol path. Application clients or indexers do not need to store the entire network ledger V_{total} but only the specific protocol branches $V_{protocol}$ relevant to their business.

$$V_{node} = \sum_{p \in Interest} V_{protocol}(p) \ll V_{total} \tag{27}$$

This is engineering-equivalent to the database requirements of traditional Web applications—developers only need to care about their own data. In theory, the complete historical data can be fragmented and distributed across countless nodes. The system's integrity is assured as long as at least one full copy (or copies dispersed via erasure coding) exists somewhere in the network, without requiring every miner to store the full dataset. This strategy of "distributed fragmented storage with on-demand local indexing" thoroughly alleviates concerns over single-point storage explosion.

8.3 Conclusion: The Inevitability of Evolution

In summary, the challenges MetaWeb faces are physical in nature, but its solutions are economic, computational, and technological. MetaWeb's architecture—a permissionless multi-chain market (reducing cost), protocol-based composability (lowering development barriers), and structured on-demand indexing (addressing storage pressure)—provides the correct environment capable of nurturing these solutions. These endogenous dynamics will drive the system's automatic evolution until it crosses the critical tipping point, becoming the default infrastructure for human digital civilization.

9 Conclusion

The classical architecture of the Internet is built upon the fragile premise of a "trusted third party." While it delivered an explosion of information, it also erected digital walled gardens where a few giant platforms, through closed ecosystems, monopolized the interpretation and control of data. The birth of Bitcoin provided the first mathematical proof that humanity could reach a global consensus on "value" without any centralized coordinator.

The entire work of this whitepaper has been a rigorous extrapolation of this paradigm's universality: how to systematically extend this foundation of peer-to-peer electronic cash into a foundation for a peer-to-peer global Internet.

Through the preceding arguments, we conclude that only by returning to the most fundamental, physics-compliant Bitcoin model can we support civilizational-scale data throughput and value anchoring. We established **Full On-Chain Data** as an uncompromising first principle, because only when data exists physically on a public ledger—not on private servers—does "trustlessness" attain ontological significance. Data ascends from being a "service provider's record" to an "**independent digital entity**."

We designed the **MetaID Protocol** to encapsulate these discrete digital entities into immutable "**data bricks**." These bricks are endowed with topological order and identity semantics, becoming the minimal units for constructing a veridical digital society. From this emerge **MetaApp** and **MetaWeb**—no longer ephemeral services, but perpetual digital realities whose lifespan is coterminous with blockchain consensus. The **MetaBitcoin Network** scheme makes it possible for MetaWeb's capacity to scale horizontally to a global user base.

We make no presuppositions about MetaWeb's final form. It is a purely **Human-Centric** system. As Artificial Intelligence (AI) technology advances exponentially, the barriers to application creation and content production will be entirely eliminated. The future MetaWeb will be co-constructed by global users and AI agents. Users will provide intent and ownership; AI will provide logic and assembly capability. MetaID-encapsulated data forms the hard bricks; human creativity, the infinite mortar.

What kind of tower will ultimately be built from these bricks? What shape will it take? How high will it reach? No whitepaper can predict this.

But this much is certain: once launched upon its mathematical and physical foundation, this system's continued existence will no longer depend on the will of any single organization. It is a new civilization humanity builds in digital form. And the ownership of the future Internet can, at last, return to its rightful owners: the users.

References

[1] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008.

[2] MetaID Protocol Documentation. `https://docs.metaid.io`

[3] MetaBitcoin Network. `https://metabitcoin.network`

[4] G. Wood, “Ethereum: A Secure Decentralised Generalised Transaction Ledger (Yellow Paper),” 2014. Available at: `https://ethereum.github.io/yellowpaper/paper.pdf`

[5] M. Rosenfeld, “Analysis of Hashrate-Based Double Spending,” arXiv:1402.2009, 2014.

[6] P. Wuille et al., “BIP141: Segregated Witness (Consensus layer),” 2017. Available at: `https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki`

[7] P. Wuille et al., “BIP341: Taproot: SegWit version 1 spending rules,” 2021. Available at: `https://github.com/bitcoin/bips/blob/master/bip-0341.mediawiki`

[8] C. S. Wright, “The Metanet: A blockchain-based Internet,” nChain Whitepaper, 2018.

[9] C. Rodarmor, “Ordinals Protocol,” 2023.

[10] G. Huberman, J. D. Leshno, and C. Moallemi, “Monopoly without a monopolist: An economic analysis of the bitcoin payment system,” *Review of Economic Studies*, vol. 88, no. 6, pp. 3011-3040, 2021.

[11] D. Easley, M. O’Hara, and S. Basu, “From mining to markets: The evolution of bitcoin transaction fees,” *Journal of Financial Economics*, vol. 134, no. 1, pp. 91-109, 2019.

[12] H. R. Varian, “Economics of Information Technology,” University of California, Berkeley, 2001. (Revised version of the Raffaele Mattioli Lecture).

[13] ISO/IEC 18033-2:2010, “Information technology — Security techniques — Encryption algorithms — Part 2: Asymmetric ciphers.” (Includes ECIES.)

[14] NIST, “SP 800-56A Rev. 3: Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography,” 2018.

[15] B. Metcalfe, “Metcalfe’s Law after 40 Years of Ethernet,” *IEEE Computer*, vol. 46, no. 12, pp. 26–31, 2013.

[16] M. H. Kryder and C. S. Kim, “After Hard Drives—What Comes Next?” *IEEE Transactions on Magnetics*, vol. 41, no. 10, pp. 3406–3413, 2005.