Shinzo: A Technical Architecture for Trustless, Decentralized Blockchain Indexing

Abstract

This paper presents Shinzo, a novel protocol for decentralized blockchain indexing that fundamentally reimagines how blockchain data is extracted, structured, and made accessible to applications. Unlike traditional indexing services that operate as trusted intermediaries, Shinzo embeds indexing directly within blockchain validators, leveraging their existing role in network consensus to create a trustless data layer. Through the integration of DefraDB, a peer-to-peer database with cryptographic verification capabilities, and LensVM, a bi-directional transformation engine, Shinzo creates a unified, verifiable, and permissionless infrastructure for blockchain data access. We detail the technical architecture, cryptographic foundations, and distributed systems principles that enable Shinzo to provide guaranteed data availability, cross-chain interoperability, and developer-friendly access patterns while maintaining the decentralization ethos of blockchain networks.

1. Introduction

Blockchain networks generate vast quantities of data with every block produced, yet accessing this data in meaningful ways remains one of the most significant challenges facing decentralized application development. While blockchains themselves are decentralized and trustless, the infrastructure for querying and analyzing blockchain data paradoxically relies on centralized, trusted services. This architectural contradiction undermines the fundamental value propositions of blockchain technology and creates systemic risks for applications built on these networks.

The current landscape of blockchain indexing is dominated by centralized providers that run full nodes, extract data, and expose it through proprietary APIs. These services, while convenient, introduce multiple points of failure into otherwise decentralized systems. They require users to trust that the data provided is accurate and complete. They create availability risks, as applications become dependent on the continued operation of specific service providers. They limit data access to predefined queries and schemas, constraining

Draft

innovation. Most critically, they transform blockchain data from a public good into a gatekept resource, controllable by single entities.

Shinzo proposes a fundamentally different approach to blockchain indexing that aligns with the core principles of Web3: decentralization, permissionlessness, trustlessness, verifiability, and sovereignty. Rather than treating indexing as an external service, Shinzo makes it an intrinsic function of blockchain validation. Validators, already responsible for processing every transaction and maintaining consensus, become the primary source of indexed data. This data is then stored and distributed through DefraDB, a peer-to-peer database that provides cryptographic verification and decentralized replication. The integration of LensVM enables sophisticated data transformations, allowing raw blockchain events to be shaped into application-specific views without sacrificing verifiability.

This paper provides a comprehensive technical analysis of Shinzo's architecture, implementation, and implications for blockchain infrastructure. We demonstrate how the combination of validator-native indexing, content-addressable storage, and decentralized distribution creates a new paradigm for blockchain data access that is simultaneously more reliable, more flexible, and more aligned with blockchain principles than existing solutions.

2. Background and Motivation

2.1 The Blockchain Data Access Problem

Blockchain networks store data in structures optimized for consensus and verification, not for application queries. Transaction data is embedded in blocks, state changes are recorded as traces, and smart contract events are emitted as logs. While this data is technically public and accessible to anyone running a full node, extracting meaningful information requires significant computational resources and technical expertise.

Consider a simple query like "What is the current balance of token X for address Y?" Answering this question requires processing the entire history of transactions involving that token contract, computing the state transitions, and arriving at the current balance. For more complex queries involving multiple contracts, time ranges, or aggregations, the computational requirements multiply exponentially.

This complexity has led to the emergence of indexing services that pre-process blockchain data and expose it through developer-friendly APIs. However, these services reintroduce many of the problems that blockchains were designed to solve. They create central points of control over data access. They require trust in the accuracy and completeness of

Draft

indexed data. They introduce availability risks as applications become dependent on specific providers. Perhaps most problematically, they transform blockchain data from an open, permissionless resource into a service that can be restricted, rate-limited, or monetized by intermediaries.

2.2 Trust Assumptions in Current Indexing Solutions

Existing blockchain indexers operate on a trust model that is fundamentally at odds with blockchain principles. When an application queries an indexer for blockchain data, it must trust that the indexer has accurately processed all relevant transactions, that it hasn't omitted any data, and that it hasn't been compromised or manipulated. This trust extends beyond just the indexer operator to their entire infrastructure stack, including the nodes they connect to, the databases they use, and the APIs they expose.

The problem is compounded by the lack of cryptographic proofs in traditional indexing architectures. While blockchains provide Merkle proofs for transaction inclusion and state roots for verification, centralized indexers typically provide no such guarantees for their derived data. An application receiving data from an indexer has no way to verify that the data accurately reflects the true state of the blockchain without independently processing the same data.

This trust requirement creates several downstream effects. Applications must carefully vet indexer providers, often relying on reputation rather than cryptographic guarantees. Regulatory and compliance requirements become complex when critical data flows through trusted intermediaries. Innovation is constrained as developers must work within the limitations of existing indexer schemas and query patterns. Most concerning, the entire ecosystem becomes vulnerable to the failure or malfeasance of a small number of critical infrastructure providers.

2.3 Cross-Chain Complexity

The multi-chain future of blockchain technology exacerbates the indexing challenge. Each blockchain has its own data structures, event formats, and state models. Applications that need to access data from multiple chains must integrate with multiple indexers, each with its own API, data model, and trust assumptions. This fragmentation makes cross-chain application development unnecessarily complex and error-prone.

Current solutions to cross-chain data access typically involve aggregation services that themselves query multiple underlying indexers. This adds another layer of trust and potential failure, while doing nothing to address the fundamental problem. The lack of a unified, trustless approach to multi-chain data access remains one of the primary barriers to true blockchain interoperability.

Draft

3. The Shinzo Architecture

3.1 Core Design Principles

Shinzo's architecture is guided by several fundamental principles that distinguish it from traditional indexing approaches. First, indexing should be performed by the most authoritative sources of blockchain data: the validators themselves. Second, indexed data should be cryptographically verifiable, allowing any party to confirm its accuracy without trust. Third, data distribution should be decentralized and censorship-resistant, ensuring that no single party can restrict access. Fourth, the system should be flexible enough to support arbitrary data transformations and query patterns without compromising its security properties.

These principles lead to an architecture that consists of three primary components working in concert. Validator-embedded indexers extract and structure blockchain data at the source. DefraDB provides a peer-to-peer storage and distribution layer with built-in cryptographic verification. LensVM enables flexible data transformations that allow the same underlying data to serve multiple use cases. Together, these components create a complete pipeline from raw blockchain data to application-ready datasets.

3.2 Validator-Embedded Indexing

The foundation of Shinzo's trustless model lies in performing indexing within validator nodes themselves. Validators are already processing every transaction, executing smart contracts, and maintaining the complete state of the blockchain. By embedding indexing logic directly within validator infrastructure, Shinzo eliminates the need for external parties to reconstruct this same information.

The technical implementation involves a lightweight indexer client that runs alongside the validator's primary blockchain node software. This client subscribes to the node's event stream, receiving notifications for each new block, transaction, and state change. Unlike external indexers that must request this data over RPC interfaces, the embedded indexer has direct access to the validator's local state, ensuring both efficiency and accuracy.

The indexer client processes incoming blockchain data according to configurable schemas that define which events to track, how to structure the data, and what aggregations to maintain. For efficiency, validators typically maintain a sliding window of recent blocks in full detail while aggregating older data. This approach balances the need for detailed recent history with practical storage constraints.

Draft

Critically, all indexed data is stored in DefraDB with cryptographic proofs of its derivation from blockchain state. These proofs, constructed using Merkle trees and recursive SNARKs, allow any party to verify that the indexed data accurately reflects the underlying blockchain without needing to re-process the raw data. This creates a trustless bridge between the blockchain's consensus layer and Shinzo's data layer.

3.3 Decentralized Storage and Distribution

Once validators have indexed blockchain data, it must be made available to applications in a decentralized manner. This is where DefraDB's peer-to-peer architecture becomes essential. Rather than validators serving data directly to applications, which would create scalability and availability challenges, the indexed data is published to a distributed network of storage nodes called Hosts.

DefraDB implements a sophisticated content-addressable storage system based on InterPlanetary Linked Data (IPLD) and Merkle DAGs. Every piece of indexed data is identified by a cryptographic hash of its content, ensuring that data integrity can be verified by any recipient. The use of Merkle DAGs allows complex data structures to be built up from simple content-addressed blocks, with each level of the structure providing its own integrity guarantees.

The distribution of data across Hosts leverages DefraDB's publish-subscribe (PubSub) system. When a validator indexes new data, it publishes an announcement to relevant PubSub channels. Hosts that have subscribed to these channels receive notifications and can choose to replicate the data based on their storage policies. This creates a dynamic, market-driven distribution network where popular or valuable datasets are widely replicated while less frequently accessed data may be stored by fewer Hosts.

The peer-to-peer networking layer, built on LibP2P, provides robust connectivity even in the presence of network partitions, firewalls, and NAT traversal challenges. Nodes can discover each other through multiple mechanisms including distributed hash tables, multicast DNS for local networks, and bootstrap nodes for initial network entry. All communications are encrypted using modern protocols, ensuring that data privacy is maintained even as data is distributed across multiple nodes.

3.4 Data Transformation Layer

Raw indexed blockchain data, while accurate and verifiable, is often not in the optimal format for application consumption. Different applications may need different views of the same underlying data. A DeFi analytics platform might need aggregated trading volumes, while a wallet application needs individual token balances. A compliance tool might need

transaction histories organized by counterparty, while a tax application needs them organized by date.

LensVM addresses this challenge by providing a flexible, verifiable transformation layer. Built on WebAssembly for portability and security, LensVM allows developers to define bi-directional transformations between data schemas. These transformations, called lenses, can reshape, aggregate, filter, and enrich indexed data to match specific application needs.

The bi-directional nature of lenses is crucial for maintaining data integrity across transformations. Each lens consists of a forward transformation function and a reverse transformation function, with mathematical guarantees that composing these functions yields the identity. This ensures that no information is lost during transformation and that the original data can always be recovered if needed.

Lenses are themselves content-addressed and distributed through the network, allowing developers to share and reuse transformations. Common patterns, such as ERC-20 token balance calculations or DEX liquidity aggregations, can be defined once and used across many applications. The lens composition system allows complex transformations to be built from simpler components, promoting modularity and reuse.

3.5 Query and Access Patterns

Shinzo provides multiple mechanisms for applications to access indexed blockchain data, recognizing that different use cases have different requirements for latency, bandwidth, and trust models. The primary access pattern involves querying Hosts through their GraphQL APIs, which are automatically generated from the schemas of available datasets.

For applications requiring real-time data with minimal latency, direct subscription to validator PubSub channels is available. This allows applications to receive indexed data as soon as it is produced, without waiting for distribution through the Host network. The cryptographic proofs are included with the data stream, maintaining the same verification guarantees as data accessed through Hosts, though the application must handle storage of the received data.

Applications with specific performance or availability requirements can integrate DefraDB in various configurations. Developers can choose to run DefraDB with storage in their own cloud infrastructure, maintaining full control while benefiting from automatic synchronization with the network. Alternatively, applications can embed DefraDB directly within their client software, enabling user devices to store their own data locally. This edge-first approach ensures users maintain sovereignty over their data while the application continues to function even without network connectivity.

Draft

The flexibility of DefraDB's architecture allows developers to offer users choice in their data management strategy. Users can rely solely on their local device storage for maximum privacy and control, opt for developer-provided backup services for convenience, or leverage the Host network for decentralized backup and availability. This multi-modal approach ensures that applications can adapt to different user preferences and regulatory requirements while maintaining the same cryptographic guarantees and verification properties across all storage configurations.

4. Cryptographic Foundations

4.1 Merkle CRDTs for State Verification

A critical innovation in Shinzo's architecture is the use of Merkle CRDTs (Conflict-free Replicated Data Types) for managing indexed state. Traditional CRDTs provide eventual consistency guarantees for distributed data structures but lack built-in mechanisms for cryptographic verification. Merkle CRDTs combine the conflict-resolution properties of CRDTs with the verifiability of Merkle data structures.

In a Merkle CRDT, each state update is represented as a new node in a directed acyclic graph (DAG), with cryptographic hashes linking to previous states. This structure provides several key properties. Causal ordering is enforced by the DAG structure, as updates can only reference states that preceded them. Cryptographic integrity is guaranteed by the content-addressing scheme, as any tampering would change the hash and break the chain. Efficient verification is enabled by Merkle proofs, allowing specific state transitions to be verified without accessing the full history.

The challenge with naive Merkle CRDT implementations is that verification time grows linearly with the length of the update chain. For frequently updated data, this can lead to impractical verification times. Shinzo addresses this through the use of recursive SNARKs (Succinct Non-Interactive Arguments of Knowledge), which allow the entire history of state transitions to be verified with a constant-size proof.

4.2 Recursive Proof Systems

The integration of recursive SNARKs represents a significant technical advancement in blockchain indexing. These proof systems allow a prover to demonstrate that a sequence of state transitions was performed correctly without requiring the verifier to replay all transitions. In Shinzo's context, this means that the derivation of indexed data from blockchain state can be proven succinctly, regardless of how many blocks or transactions were processed.

Draft

The recursive nature of these proofs is particularly powerful. Each proof can incorporate previous proofs, creating a chain of verifiable computation. When a validator indexes a new block, it generates a proof that incorporates the proof from the previous block, demonstrating that both the historical indexing and the new indexing were performed correctly. This recursive structure allows months or years of indexing history to be verified with a single, small proof.

The practical implementation uses a combination of techniques to achieve efficiency. The underlying SNARK system is optimized for the specific computations involved in indexing, such as Merkle tree operations and hash functions. Proofs are aggregated at regular intervals to bound proof generation time. Caching strategies ensure that common sub-proofs can be reused across multiple indexing operations.

4.3 Access Control and Privacy

While blockchain data is inherently public, Shinzo recognizes that access control and privacy are still important considerations for many applications. The system implements a capability-based access control system using Decentralized Identifiers (DIDs) and cryptographic credentials.

Each participant in the Shinzo network, whether a validator, Host, or application, is identified by a DID derived from their cryptographic keypair. Access policies can be defined that grant specific DIDs permission to access certain datasets or transformation views. These policies are themselves stored as content-addressed objects in DefraDB, ensuring they are tamper-proof and auditable.

For privacy-sensitive applications, Shinzo supports selective disclosure through zero-knowledge proofs. Rather than revealing entire datasets, applications can generate proofs about specific properties of the data. For example, a proof could demonstrate that a particular address holds more than a certain token balance without revealing the exact balance. This capability is particularly valuable for compliance and regulatory applications that need to verify properties without accessing raw data.

5. Implementation Considerations

5.1 Performance Optimization

The performance characteristics of Shinzo reflect careful optimization across multiple dimensions. Validator-side indexing is designed to have minimal impact on block production and validation. The indexer client operates asynchronously, processing blocks after they have been validated and committed. Configurable schemas allow validators to

Draft

index only the data relevant to their users, avoiding unnecessary computation and storage.

The distributed storage layer employs several optimization techniques. Data sharding ensures that no single Host needs to store entire datasets, with intelligent placement algorithms ensuring that related data is co-located when possible. Bloom filters and other probabilistic data structures enable efficient queries across distributed data. Caching layers at multiple levels reduce repeated computations and network requests.

Query performance is optimized through a combination of indexing strategies and query planning. Secondary indices are automatically generated for frequently queried fields. Query plans are optimized to minimize network hops and data transfer. Parallel query execution allows complex queries to be distributed across multiple Hosts and aggregated efficiently.

5.2 Network Resilience

Shinzo's architecture is designed to remain operational even in adverse network conditions. The peer-to-peer nature of DefraDB means that there is no single point of failure. If some Hosts become unavailable, others can continue serving data. The use of content addressing means that data can be served by any node that has it, providing natural load balancing and redundancy.

The system includes mechanisms for handling network partitions gracefully. Validators continue indexing locally even if they cannot immediately publish to the network. When connectivity is restored, the Merkle CRDT properties ensure that all updates are eventually synchronized and conflicts are resolved deterministically. Applications can continue operating with cached data during network disruptions, with guarantees about the staleness of data.

Economic incentives align with resilience goals. Hosts are rewarded for storing and serving data, with higher rewards for storing less-replicated datasets. This creates natural incentives for maintaining good data availability across the network. Validators have intrinsic motivation to ensure their indexed data is available, as it increases the value of the blockchain network they are securing.

5.3 Scalability Analysis

Shinzo's scalability characteristics differ fundamentally from centralized indexing services. Rather than scaling vertically by adding more powerful servers, Shinzo scales horizontally by adding more validators and Hosts. This approach aligns with the scaling properties of the underlying blockchains and avoids creating new centralization points.

Draft

Storage scalability is achieved through intelligent data distribution. Not every Host needs to store all data. Instead, datasets are sharded based on access patterns, with popular data widely replicated and specialized data stored by fewer nodes. The content-addressable nature of the storage allows for efficient deduplication, as identical data across different indexes is stored only once.

Query scalability leverages the distributed nature of the network. Simple queries can be served by any Host with the relevant data. Complex queries can be parallelized across multiple Hosts. The transformation layer allows expensive computations to be performed once and cached, rather than repeated for each query.

Network scalability benefits from the hierarchical nature of data distribution. Validators publish to a subset of well-connected Hosts, which then distribute data to edge nodes. This creates a natural content delivery network structure without requiring centralized coordination.

6. Use Cases and Applications

6.1 Cross-Chain DeFi Infrastructure

The fragmentation of DeFi across multiple blockchains creates significant challenges for applications that need unified views of liquidity, pricing, and user positions. Current solutions rely on aggregating data from multiple centralized APIs, each with their own formats, rate limits, and reliability issues. Shinzo provides a unified substrate for cross-chain DeFi data that dramatically simplifies application development.

Consider a cross-chain DEX aggregator that needs to find optimal trading routes across multiple blockchains. With Shinzo, the aggregator can subscribe to standardized liquidity data from validators across all supported chains. The data arrives in a consistent format, with cryptographic proofs of accuracy. LensVM transformations can normalize different DEX protocols to a common interface, allowing the aggregator to compare liquidity across Uniswap, SushiSwap, PancakeSwap, and other protocols without custom integration code for each.

The benefits extend beyond simple data access. Because Shinzo provides historical data with the same guarantees as real-time data, applications can perform sophisticated analyses like calculating time-weighted average prices, detecting wash trading, or identifying liquidity trends. The decentralized nature ensures that this critical trading infrastructure cannot be manipulated or shut down by any single party.

6.2 Institutional Data Feeds

Draft

Financial institutions entering the blockchain space require data infrastructure that meets their standards for reliability, auditability, and compliance. Traditional blockchain APIs, with their trust requirements and lack of SLAs, are insufficient for institutional needs. Shinzo provides a foundation for institutional-grade blockchain data access.

An investment firm can deploy its own Host nodes that replicate exactly the datasets it needs for its trading strategies. These nodes receive data directly from validators with cryptographic proofs, creating an audit trail that satisfies regulatory requirements. The firm maintains complete control over its data access, with no dependency on external API providers that could change terms, experience downtime, or be subject to regulatory action.

The transformation capabilities of LensVM allow institutions to shape blockchain data to match their internal systems. Trade data can be transformed to FIX protocol format. Token transfers can be enriched with KYC information from permissioned sources. Risk metrics can be calculated and embedded directly in the data stream. All of these transformations maintain cryptographic links to the underlying blockchain data, preserving the audit trail.

6.3 Decentralized Analytics and Research

Blockchain analytics currently requires either significant technical expertise to process raw blockchain data or reliance on centralized analytics providers. Shinzo democratizes blockchain analytics by making structured, verified data publicly accessible. This enables a new ecosystem of decentralized analytics and research.

Independent researchers can access the same high-quality data as large analytics firms, without needing to run their own indexing infrastructure. They can define custom transformations to extract novel insights from blockchain data, then publish these transformations for others to use. The economic model allows researchers to monetize valuable data views, creating incentives for innovation in blockchain analytics.

The verifiable nature of Shinzo data is particularly valuable for research. Academic papers can reference specific datasets with cryptographic proofs, ensuring reproducibility. Investment theses can be backed by verifiable on-chain data. Regulatory reports can include tamper-proof evidence of blockchain activity. This creates a new standard for transparency and verifiability in blockchain research.

6.4 Real-Time Applications

Applications requiring real-time blockchain data face significant challenges with current infrastructure. Centralized APIs introduce latency and reliability concerns. Running

Draft

dedicated infrastructure is complex and expensive. Shinzo provides multiple solutions for real-time data access that match different application requirements.

A high-frequency trading system can subscribe directly to validator data feeds, receiving indexed updates within milliseconds of block production. The cryptographic proofs included with each update allow the trading system to verify data accuracy without trusting the validator. Multiple validator feeds can be aggregated to ensure resilience against any single validator failure.

Gaming applications built on blockchains can use Shinzo to maintain real-time game state. Player actions recorded on-chain are immediately indexed and distributed to all participants. The CRDT properties ensure that all players eventually see a consistent game state, even in the presence of network delays. Custom transformations can convert raw blockchain events into game-specific data structures.

Social applications benefit from Shinzo's ability to aggregate activity across multiple blockchains. A decentralized social network can display user activity from various chains in a unified feed. The peer-to-peer nature ensures that social data remains accessible even if specific infrastructure providers go offline. Users maintain sovereignty over their social graph and content without sacrificing the convenience of aggregated feeds.

8. Comparison with Existing Solutions

8.1 Centralized Indexers

Traditional centralized indexers like The Graph's hosted service, Covalent, or proprietary exchange APIs represent the current state of the art in blockchain data access. These services have demonstrated the demand for indexed blockchain data but suffer from fundamental architectural limitations that Shinzo addresses.

Centralized indexers require trust in the service provider for data accuracy and availability. If the service experiences downtime, misconfigures their indexing, or deliberately manipulates data, applications have no recourse. Shinzo's validator-based indexing with cryptographic proofs eliminates these trust requirements.

The API surface of centralized indexers is limited to what the provider chooses to expose. Adding new queries or data views requires convincing the provider to implement them. Shinzo's transformation layer allows any developer to create new data views without permission, fostering innovation and customization.

Centralized indexers create vendor lock-in through proprietary APIs and data formats. Migrating between providers requires significant application changes. Shinzo's

Draft

standardized, open-source approach ensures that applications can switch between data providers or run their own infrastructure without code changes.

8.2 The Graph Protocol

The Graph represents the most significant attempt to decentralize blockchain indexing before Shinzo. While The Graph shares some goals with Shinzo, the architectural approaches differ significantly. Understanding these differences illuminates Shinzo's design decisions.

The Graph relies on independent indexers who stake tokens and can be slashed for providing incorrect data. This creates economic incentives for accuracy but still requires trust that the slashing mechanism works correctly. Shinzo's use of cryptographic proofs provides mathematical certainty rather than economic probability.

The Graph's indexing is performed by third parties who must independently sync blockchain data and run indexing infrastructure. This duplicates work already being done by validators and creates additional points of failure. Shinzo's validator-embedded approach eliminates this redundancy and ensures indexing happens at the most authoritative source.

The Graph requires developers to write custom subgraphs in AssemblyScript for each indexing use case. While flexible, this approach requires significant technical expertise and creates opportunities for bugs. Shinzo's combination of configurable schemas and LensVM transformations provides flexibility while maintaining verifiability.

8.3 Direct Node Access

Some applications attempt to avoid indexer dependencies by directly querying blockchain nodes through RPC interfaces. While this approach eliminates intermediaries, it faces severe practical limitations that make it unsuitable for most applications.

Direct node access requires processing large amounts of raw data to answer simple queries. Calculating a token balance might require processing thousands of transactions. This is computationally expensive and slow, making it impractical for user-facing applications.

Running dedicated nodes for each blockchain an application needs to access is operationally complex and expensive. Nodes require significant storage, bandwidth, and maintenance. For applications needing data from multiple chains, the infrastructure requirements multiply accordingly.

Draft

Node RPC interfaces are designed for basic blockchain operations, not application queries. They lack mechanisms for efficient filtering, aggregation, or historical queries. Building application-specific queries on top of raw RPC interfaces essentially means building a custom indexer, defeating the purpose of direct access.

10. Conclusion

Shinzo represents a fundamental reimagining of blockchain indexing that aligns with the core principles of decentralization, trustlessness, and permissionless innovation. By embedding indexing within validators, distributing data through peer-to-peer networks, and enabling flexible transformations, Shinzo creates infrastructure that is both more reliable and more capable than centralized alternatives.

The technical architecture we have presented demonstrates that decentralized indexing is not only possible but practical. The combination of cryptographic proofs, distributed systems principles, and economic incentives creates a system that can serve the needs of applications ranging from simple wallets to complex trading systems. The unified approach to multi-chain data promises to simplify cross-chain development and enable new categories of applications.

Perhaps most importantly, Shinzo returns blockchain data to its status as a public good. No longer gatekept by centralized providers, blockchain data becomes freely accessible to any developer or researcher who needs it. The permissionless nature of the transformation layer ensures that innovation in data access can happen at the edges, driven by user needs rather than provider decisions.

As blockchain technology continues to mature and find real-world applications, the importance of reliable, trustless data infrastructure will only grow. Shinzo provides a foundation for this future, ensuring that the data layer of blockchain applications can match the decentralization of the underlying networks. We invite the community to build upon this foundation, extending and improving the protocol to meet the evolving needs of the decentralized web.

The transition from centralized to decentralized indexing will not happen overnight. It requires validators to adopt new responsibilities, developers to embrace new patterns, and users to understand new trust models. However, the benefits – in terms of reliability, flexibility, and alignment with blockchain principles – justify this transition. Shinzo is not just a technical solution but a step toward a more open, accessible, and decentralized future for blockchain data.

Draft