KAI Data DAO

Decentralized Data Governance on Sui

Abstract

Current data economies extract value from user-generated information without consent or compensation. KAI establishes a decentralized autonomous organization on Sui Network that enables cryptographic data ownership, community governance, and direct creator compensation. Through Seal encryption, Walrus storage, and on-chain voting mechanisms, KAI transforms data from an exploited resource into a governed digital asset where contributors retain control and capture economic value.

1 Introduction

The global data economy exceeds $200 billion annually, yet data creators receive no compensation. Centralized platforms operate under terms-of-service agreements that grant perpetual, irrevocable rights to user information. This extraction occurs without transparency regarding data usage, processing, or monetization. Regulatory frameworks like GDPR and CCPA acknowledge the problem but provide only reactive remedies rather than structural solutions.

KAI addresses this through three architectural components. First, Seal encryption ensures that raw data remains cryptographically inaccessible without explicit authorization. Second, Walrus provides decentralized storage that prevents single-point control or censorship. Third, Sui's object model enables granular on-chain governance where every data submission, approval, and pricing decision undergoes democratic review.

The system operates through a token-based economy where the KAI governance token allocates voting power, compensates contributors, and facilitates marketplace transactions. Rather than replacing existing data infrastructure, KAI establishes a parallel economy where consent and compensation are embedded in the protocol rather than enforced through regulation.

2 Technical Architecture

2.1 Sui Network Foundation

KAI leverages Sui's object-centric design where each entity exists as an independent on-chain object with typed properties and ownership semantics. This differs fundamentally from account-based blockchains. A data submission becomes a `DataSubmission` object with fields for the Walrus blob identifier, metadata, category reference, approval status, and pricing information. The submitter address establishes provenance, while the object's shared status enables DAO-wide governance.

Sui's parallel execution model processes independent transactions simultaneously. When multiple users vote on different proposals, these operations execute concurrently rather than sequentially. This architecture supports high throughput necessary for a data marketplace with potentially millions of submissions and governance actions.

Move, Sui's programming language, provides resource safety through linear types. KAI tokens cannot be duplicated or destroyed except through explicit burn operations. The treasury balance exists as a `Balance<SUI>` object that can only be modified through defined functions, preventing unauthorized withdrawals or creation.

2.2 Encryption and Storage

Seal performs client-side encryption before data leaves the creator's environment. The encrypted blob uploads to Walrus, receiving a content-addressed identifier derived from the data's cryptographic hash. This identifier becomes part of the on-chain `DataSubmission` object, linking the governance layer to the storage layer without exposing raw information.

Walrus distributes encrypted data across multiple nodes with erasure coding, ensuring availability even if individual nodes fail. The decentralized topology prevents any single entity from controlling access or removing data. Content addressing guarantees integrity since any modification changes the hash, breaking the link to the on-chain reference.

Access control operates through cryptographic key distribution. When a buyer purchases data access through the marketplace, the smart contract records the transaction on-chain. Off-chain infrastructure then provides the decryption key to the authorized address. The original submitter retains the master key, maintaining ultimate ownership while enabling controlled distribution.

2.3 Smart Contract Design

The `DataDAO` object serves as the central coordinator, maintaining the SUI treasury, KAI reserve for purchases, reward pool for contributors, and governance parameters including quorum thresholds, approval percentages, and voting durations. This object exists as a shared resource accessible to all participants.

User accounts manifest as `AccountCap` objects containing KAI balance and DAO membership credentials. These capability objects enable participation in governance and marketplace activities. The `kai_balance` field stores KAI as a `Balance<KAI>` resource, ensuring type safety and preventing unauthorized transfers.

Data categories define submission types and reward structures. A `DataCategory` object specifies the category name, description, reward amount for approved submissions, and active status. Categories emerge through governance proposals rather than administrative fiat, ensuring community control over ecosystem direction.

Proposals implement the governance mechanism. Each `Proposal` object contains the proposal type (category creation, data approval, or pricing), encoded data specific to that type, vote count, voter list to prevent double-voting, expiration timestamp, and execution status. Proposals transition through states: active voting, quorum verification, threshold check, and execution or rejection.

3 Token Economics

KAI has a fixed supply of 1,000,000 tokens with 6 decimal precision, yielding 1,000,000,000,000 base units. The distribution allocates 300,000 tokens (30%) to a purchase reserve and 700,000 tokens (70%) to a reward pool. This 7:3 ratio ensures sustained contributor compensation while enabling governance participation.

The purchase mechanism exchanges SUI for KAI at a rate initially set to 1000:1. A user sending 1 SUI receives 1,000 KAI from the reserve. The SUI enters the treasury, building reserves that back the token economy. The inverse operation allows KAI holders to burn tokens and redeem SUI from the treasury at the same rate, establishing a price floor based on treasury backing.

Voting power scales linearly with KAI holdings. A participant with 10,000 KAI contributes 10,000 votes when supporting a proposal. This proportional system aligns influence with economic commitment. Unlike quadratic voting schemes, the linear relationship maintains simplicity while preventing Sybil attacks through economic cost.

Rewards distribute when data receives DAO approval. If the Medical Data category specifies 10,000 KAI per approved submission, the contributor receives exactly that amount from the reward pool upon successful governance approval. This direct compensation mechanism eliminates intermediaries and ensures creators capture value immediately.

The treasury sustains long-term operations through multiple revenue streams. KAI purchases contribute SUI to reserves. Marketplace transactions where buyers pay SUI for data access flow directly to the treasury. As the ecosystem matures, the treasury accumulates sufficient reserves to support ongoing operations without requiring external funding.

3.1 Governance Mechanisms

Quorum requirements prevent low-turnout decisions. Proposals must attract votes representing at least 30% of circulating KAI supply, calculated as total supply minus locked reserves. If 300,000 KAI circulates and a proposal receives 100,000 votes, it reaches the 33.3% participation threshold and proceeds to approval evaluation.

Approval thresholds determine passage. A proposal needs 70% of votes cast to execute. With 100,000 total votes, 70,000 must support the proposal for approval. This supermajority requirement ensures broad consensus for significant decisions while remaining achievable for well-supported proposals.

Time bounds establish voting periods. Each proposal remains active for 604,800,000 milliseconds (7 days), providing sufficient time for deliberation while maintaining decision velocity. Proposals cannot execute during the active period, and votes cast after expiration have no effect. This deterministic timeline enables predictable governance cycles.

Three proposal types implement different governance actions. Category proposals (type 1) encode the category name, description, and reward amount. Upon approval, the system creates a `DataCategory` object with these parameters. Data approval proposals (type 2) automatically generate when users submit data, encoding the submission identifier, Walrus blob reference, metadata, and category identifier. Pricing proposals (type 3) encode a submission identifier and proposed SUI price, enabling marketplace listing upon approval.

The voting process prevents manipulation through multiple safeguards. Each address can vote once per proposal, with the voter list stored on-chain. Vote weight derives from KAI balance at vote time, preventing flash-loan style attacks where tokens briefly borrowed would inflate voting power. Time locks prevent premature execution, while the immutable blockchain record ensures vote transparency.

4 Data Lifecycle

Data enters the system through a structured submission process. The creator encrypts data locally using Seal, then uploads the encrypted blob to Walrus. The upload returns a blob identifier that serves as the permanent reference. The creator then invokes the `submit_data` function with this identifier, metadata describing the submission, and the target category reference.

The submission function validates that the specified category exists and remains active. It creates a `DataSubmission` object with the provided parameters, setting the submitter field to the transaction sender and

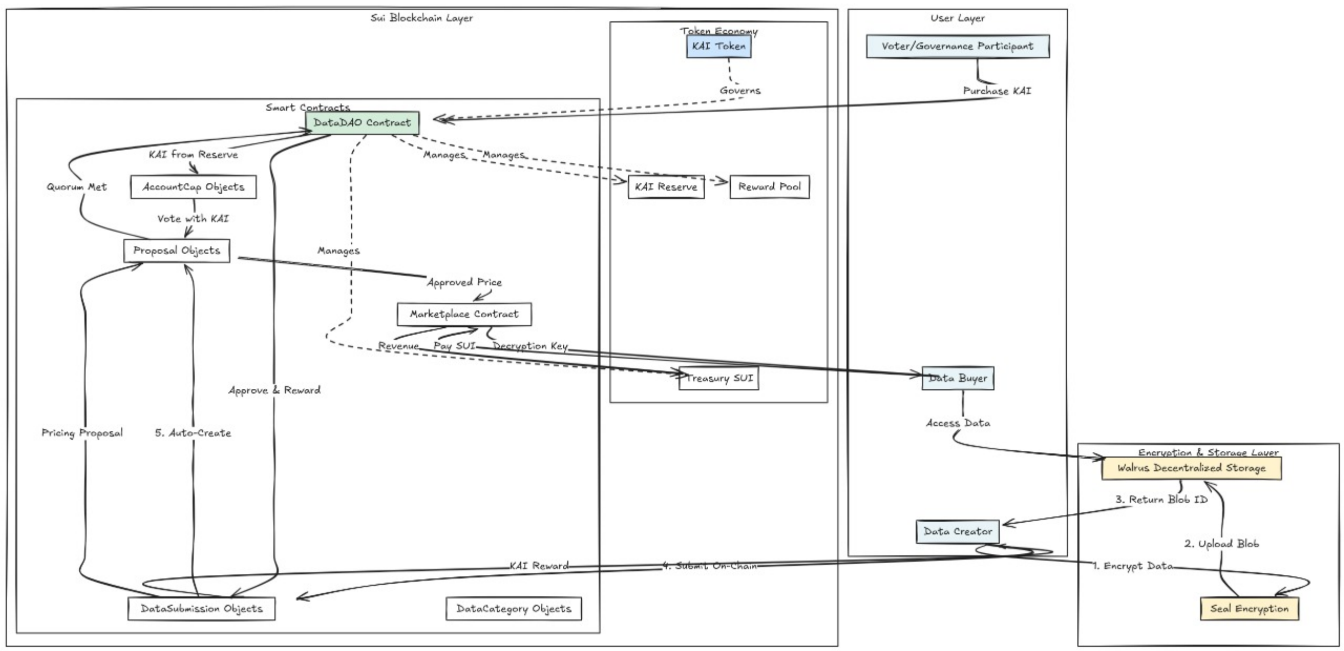

Figure 1: Architecture Diagram

initializing approval and listing flags to false. The function then automatically creates a type 2 proposal, encoding the submission details for DAO review.

Governance participants examine submissions through off-chain interfaces that display metadata and allow sampling of data quality. Voters with sufficient KAI holdings call the `vote` function, referencing both their account capability and the proposal object. The system verifies membership, checks voting deadlines, prevents double-voting, and increments the vote count by the participant's KAI balance.

After the voting period expires, anyone can trigger execution by calling the appropriate execution function. For data approval proposals, the `execute_data_proposal` function calculates participation percentage and approval rate. If both quorum and threshold requirements satisfy, the function updates the submission's approval status and transfers the category-specified reward from the pool to the submitter.

Approved data becomes eligible for marketplace listing through pricing proposals. A participant creates a type 3 proposal suggesting a SUI price for specific approved data. The DAO votes on this valuation, and upon approval, the system sets the submission's price field and adds an entry to the marketplace listings table. The data becomes purchasable at the community-determined price.

Purchase transactions involve buyers sending SUI to the `purchase_data` function along with the submission reference. The system verifies the submission has marketplace listing status and that the payment meets the required price. The SUI enters the treasury, and off-chain infrastructure provides the buyer with decryption keys to access the purchased data.

5 Security Properties

The architecture provides several security guarantees through cryptographic and economic mechanisms. Data confidentiality derives from Seal encryption where raw information never exists in unencrypted form on Walrus or on-chain. Even DAO participants cannot access data without authorization, since decryption keys remain off-chain and distribute only to legitimate purchasers.

Governance integrity relies on on-chain vote recording and execution. Vote manipulation becomes prohibitively expensive since acquiring sufficient KAI to dominate decisions requires substantial economic commitment. The 30% quorum requirement means attackers must control at least that percentage of circulating supply to force decisions, while the 70% threshold demands even greater

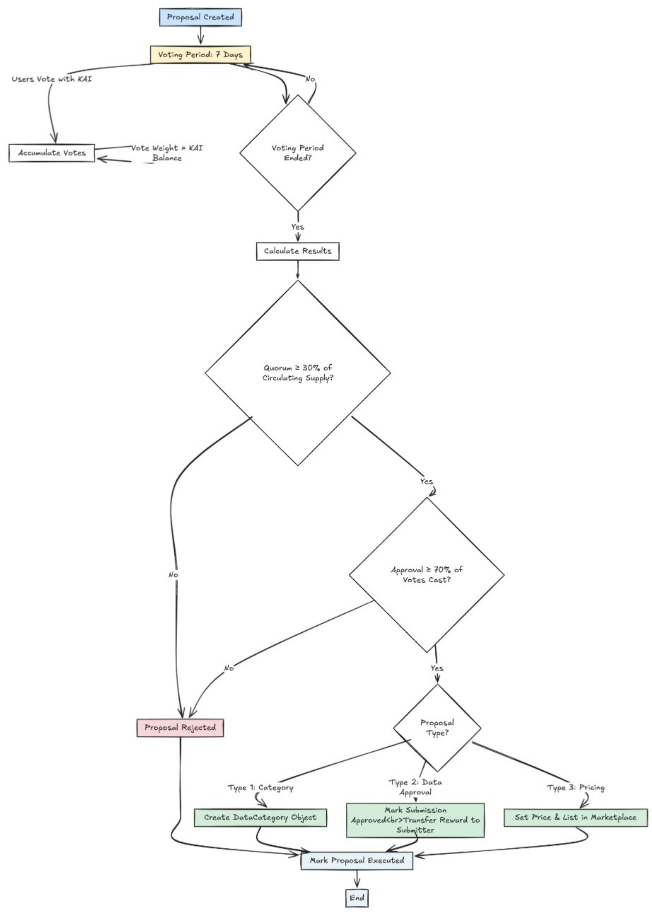

Figure 2: Governance Flow Diagram

control for approval.

Treasury security follows from Move's resource model where balance objects cannot be duplicated or created outside minting operations. The `DataDAO` object controls all balance manipulation through explicitly defined functions. Unauthorized withdrawals become impossible without private key access to administrator addresses, and even administrators cannot violate the contract's economic rules.

Censorship resistance emerges from the combination of Walrus storage and Sui's permissionless network. No entity can remove data from Walrus without cooperation from multiple storage providers. On-chain governance decisions remain permanent and verifiable. While the DAO might reject submissions through voting, it cannot retroactively delete approved data or alter historical records.

Double-voting prevention operates through the voter list maintained in each proposal object. Before recording a vote, the system checks whether the sender's address appears in this list. The check occurs on-chain within the Move contract, making circumvention impossible without blockchain consensus compromise. This mechanism costs minimal storage since only addresses vote, not full vote details.

6 Economic Sustainability

The token distribution directly addresses long-term sustainability. Allocating 70% of supply to rewards ensures continuous contributor compensation even with substantial data submission volume. At 10,000 KAI per approved submission, the reward pool supports 70,000 approvals before depletion, providing runway for ecosystem establishment.

Revenue from marketplace transactions replenishes the treasury with SUI that backs future KAI redemptions. As data sales accumulate, the treasury balance grows, increasing the economic backing per KAI token. This mechanism creates a positive feedback loop where successful data commerce strengthens the token economy.

The burn mechanism provides liquidity by allowing KAI holders to exit their position by redeeming SUI from the treasury. This redemption occurs at the established price ratio, creating a price floor. If KAI trades below its SUI backing value, arbitrageurs can purchase discounted tokens, burn them, and claim SUI at the higher rate, naturally supporting token value.

Dynamic pricing could emerge through future governance proposals that adjust the KAI-SUI exchange rate based on treasury reserves and token circulation. Initial fixed pricing provides stability during ecosystem bootstrapping, while eventual flexibility enables market-responsive economics.

7 Comparative Analysis

Traditional data marketplaces operate through centralized intermediaries that set terms, control access, and capture economic rent. Ocean Protocol and similar blockchain projects attempt decentralization but typically focus on data exchange infrastructure rather than governance. KAI integrates governance directly into the data lifecycle, making every submission, approval, and pricing decision subject to democratic review.

Existing data cooperatives lack blockchain infrastructure, relying instead on legal structures and traditional databases. This approach faces jurisdictional challenges and requires trust in cooperative administrators. KAI embeds governance rules in smart contracts, making manipulation detectable and enforcement automatic.

Privacy-preserving computation projects like iExec enable data processing without exposure but do not address ownership or compensation. KAI combines privacy through encryption with economic rights through token distribution, treating data as an asset rather than merely a computational input.

8 Future Development

Current implementation establishes core functionality with fixed KAI pricing, basic proposal types, and manual submission processes. Near-term development will introduce dynamic pricing through DAO-adjustable rates, automated quality assessment using data schemas and validation rules, and delegation mechanisms allowing KAI holders to assign voting power to representatives.

Medium-term expansion includes cross-chain bridges enabling data purchase with non-SUI tokens, zero-knowledge proof integration for privacy-preserving verification, and API standardization for enterprise data consumption. Mobile applications will simplify submission for non-technical users, while enhanced metadata standards enable better discovery and categorization.

Long-term evolution transitions toward full protocol autonomy where governance proposals can modify core contract parameters, fee structures, and reward schedules without administrator intervention. Integration with decentralized identity systems will enable reputation-based submission weighting. Interoperability protocols will allow KAI data to flow between multiple blockchain ecosystems while maintaining unified governance.

9 Conclusion

KAI demonstrates that data sovereignty requires technical infrastructure rather than regulatory mandate. By combining Seal encryption, Walrus storage, and Sui governance, the protocol establishes a functional data economy where creators retain ownership, participate in governance, and receive compensation. The 70% reward allocation ensures sustained contributor incentives, while marketplace revenue provides economic sustainability.

The system's effectiveness depends on adoption that generates network effects. Initial categories and submissions establish proof of concept, while growing participation increases data diversity and marketplace activity. Success requires not merely technical functionality but community development that values data rights and collective governance over convenience and centralization.

KAI's architecture proves technically feasible. Whether it proves economically viable and culturally adopted remains dependent on execution, community building, and continued development. The foundation exists for a data economy that respects creators while enabling productive data use. Implementation determines whether this foundation becomes a transformative alternative to extraction-based data capitalism.