



SDG BLOCKCHAIN ACCELERATOR

Technical Architecture Document

1. Project Information

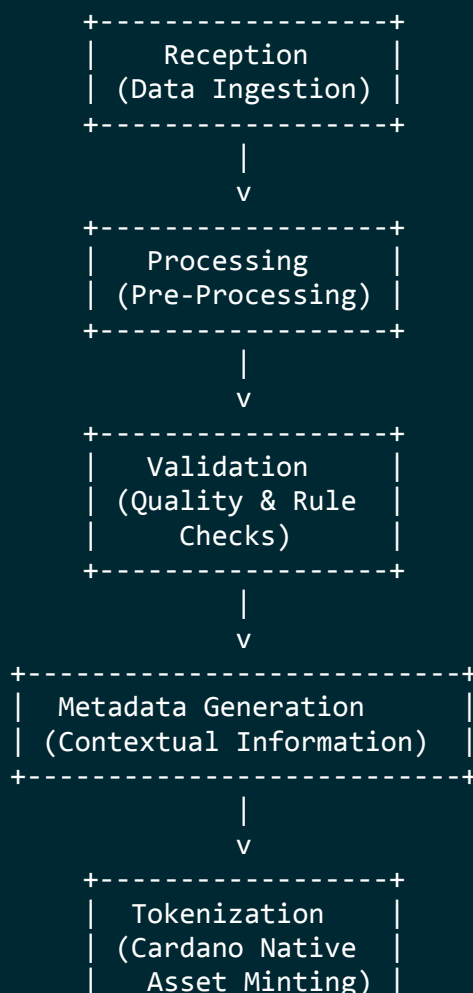
- **Project Name:** AegisGrid: using blockchain to address Tanzania’s electricity losses
- **Challenge & UNDP Office:** UNDP Tanzania – Frequent energy theft, poor billing transparency, and limited integration of renewable energy into national utility systems
- **Document Version:** 1

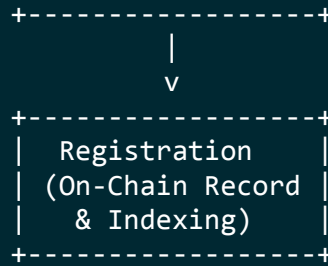
2. Overview

Background: TANESCO, Tanzania’s national electricity utility, faces over 14% losses from meter tampering, billing errors, low consumer trust, and lack of real-time monitoring. Its centralized billing system is vulnerable to manipulation, affecting revenue and planning.

The **proposed solution** is to use blockchain with smart meters or mobile billing to securely record consumption data, reduce fraud, and boost trust. This could also enable future innovations like peer-to-peer energy trading and integration with decentralized renewables

3. System Architecture Diagram





Stage Descriptions

1. Reception (Data Ingestion)

- Energy consumption data is received from meters, IoT devices, or external data providers.
- Ensures secure transmission, typically via APIs, message queues, or secure file transfer.

2. Processing (Pre-Processing)

- Raw data is cleaned, normalized, and formatted into a consistent structure.
- Handles error correction, missing values, and aggregation if needed.

3. Validation (Quality & Rule Checks)

- Ensures data accuracy and integrity.
- Applies business rules such as valid ranges, timestamp consistency, and anomaly detection.

4. Metadata Generation (Contextual Information)

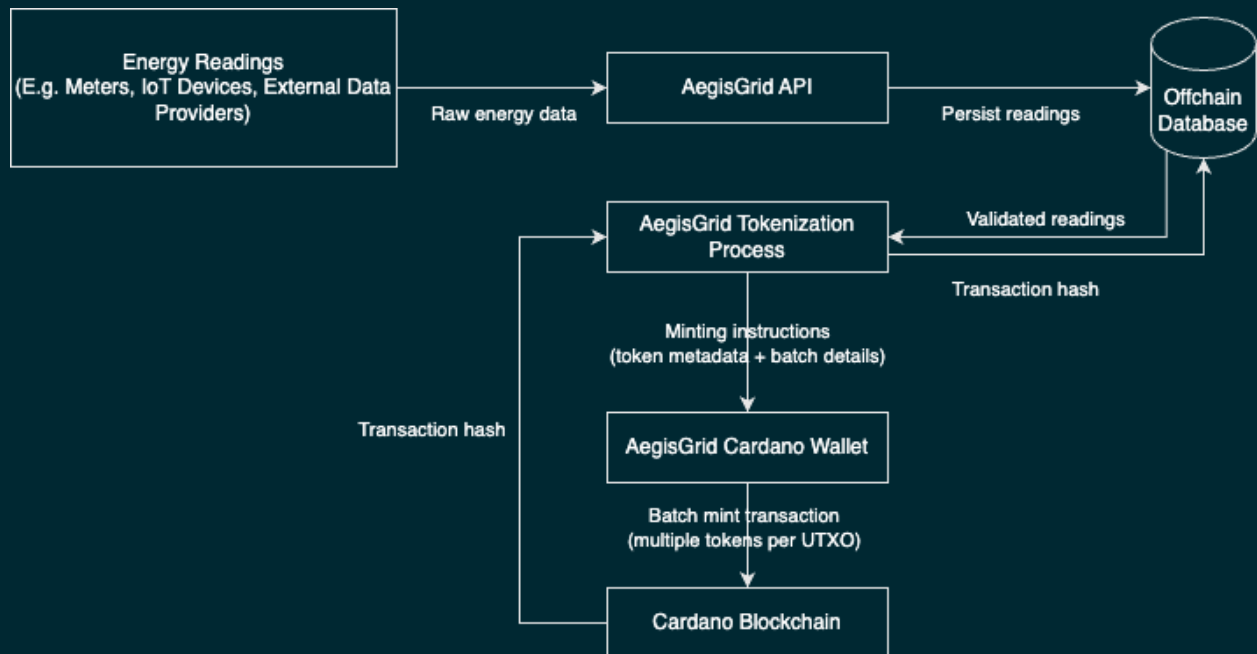
- Enriches the dataset with additional information like device ID, location, time zone, and ownership.
- Creates descriptive labels and provenance data for traceability.

5. Tokenization (Cardano Native Asset Minting)

- Converts validated energy consumption data into a **tokenized digital asset** on the Cardano blockchain.
- Each token represents a unique piece of consumption data with attached metadata.

6. Registration (On-Chain Record & Indexing)

- Final tokens and metadata are registered permanently on the Cardano blockchain.
- Provides immutability, transparency, and accessibility for auditing or marketplace transactions.



4. Blockchain Design

- **Smart Contracts:**

- Minting policy: native script that requires the project owner's signature to mint or burn energy tokens; can include optional timelock or supply limits.
- Datum structure
 - project_id: string
 - reading_id: string (unique identifier)
 - kwh: integer (scaled unit to avoid floats)
 - owner: address
 - minted_at: ISO8601 timestamp
 - nonce: integer (for replay protection)
- Redeemer (actions)
 - Action enum: { mint, burn, transfer, claim }

- Mint: { action: "mint", signer_pubkey_hash, reading_id, nonce }
- Burn: { action: "burn", owner, reason }
- Transfer: { action: "transfer", from, to, nonce }
- Claim: { action: "claim", claimer, proof }

- **UTxO Model Usage:**

- Tokens are minted in batches by selecting available UTxOs, reserving each for a transaction, and submitting one mint per energy reading.
- UTxOs are tracked to avoid double-spends, with retries and refreshes on conflict.
- Ownership and uniqueness are enforced off-chain, and only authorized signatures can mint.

- **Token Management:**

- Only transactions signed by the policy key may mint or burn.
- Off-chain: DB-enforced unique reading_id + idempotent minting process.
- On-chain: use a registry UTxO (consumed/updated atomically) to record minted reading_ids.
- Represent kWh as scaled integers to preserve fractional values.

5. Data Flow & Transaction Lifecycle

1. *AegisGrid receives energy readings through its API.*
 - a. *Readings are stored offchain in the DB.*
2. *Minting service reads non-minted readings from the DB.*
 - a. *Minting service prepares and signs a mint transaction with the required keys.*
 - b. *Transaction is submitted to the blockchain network.*
 - c. *Blockchain validator checks signatures and minting policy.*
 - d. *If valid, tokens are minted and a new UTxO is created on the ledger.*
 - e. *Minting service detects confirmation, updates its database, and links the token to the original data.*

6. Off-chain Components

- Database: Stores energy readings, tokenization records, and user information.

- API server: Handles user requests, aggregates data, and exposes endpoints for dashboards and external systems.
- Tokenization service: Prepares, signs, and submits blockchain transactions for minting tokens.

7. Sandbox/Testnet Results

All Cardano Preview transactions can be found in wallet
[addr_test1vz3dfpcaxs384dtgvm6r7rgww3ypq9mu6sat363mhuha4qguqf5vu](#)

9. Remaining Considerations / Next Steps

- Strengthen on-chain uniqueness by adding a registry or Plutus validator..
- Improve UTxO reservation for multi-process safety (e.g., use a shared DB or cache).
- Add more automated tests for date grouping, token amount conversion, and DB uniqueness.
- Expand monitoring for transaction confirmation and error handling.
- Document operational procedures for wallet backup, recovery, and system upgrades.
- Plan for scaling: optimize batch sizes, UTxO splitting, and concurrency as data volume grows.