SDG BLOCKCHAIN ACCELERATOR

Debugging and Testing Report

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
- **Report Version:** 1

2. Testing Approach

Manual end-to-end tests were conducted on the Cardano Testnet. We deployed the contracts, executed real transactions against them, and validated that the resulting on-chain data and states were correct. This included confirming that our tokenization strategy was working properly and verifying that all relevant information was accurately recorded on-chain.

3. Error Logs

Error 1:

```
2025-09-02 15:12:01,234 - INFO - Starting batch minting...
2025-09-02 15:12:01,235 - INFO - Fetched 12 untokenized readings from database
2025-09-02 15:12:01,237 - INFO - Selected UTxO 8a7b6c...#0 with 3.00 ADA for minting
2025-09-02 15:12:01,239 - DEBUG - Calculated min_lovelace for 12 assets: 5000000
2025-09-02 15:12:01,240 - ERROR - Insufficient ADA in UTxO: need 5000000, have 3000000. Consider splitting UTxOs or reducing batch size.
2025-09-02 15:12:01,241 - INFO - Released UTxO 8a7b6c...#0 back to pool
2025-09-02 15:12:01,242 - WARNING - Batch mint failed due to insufficient ADA in selected UTxO.
```

4. Resolved Issues

Issue ID	Description	Root Cause	Resolution	Status
001	Inefficient Tokenization – One Transaction per Token	Each reading/token was minted in a separate transaction, resulting in high fees, slow processing, and network congestion.	The tokenization strategy now batches multiple readings into a single transaction. This allows minting many tokens at once, significantly reducing fees and improving throughput.	✅ Fixed
002	Incorrect UTxO selection strategy	Our tokenization strategy would automatically select additional UTxOs as inputs, sometimes including UTxOs with assets or insufficient ADA, causing transaction failures or unexpected results.	Batch minting only uses pure ADA UTxOs and avoids accidental inclusion of unwanted assets.	✅ Fixed
003	Unhandled Not Enough ADA in UTxO for Batch Mint	If the selected UTxO did not have enough ADA to cover the minimum required for the output (especially with many assets), the code would fail with a cryptic error or proceed with an invalid transaction.	The code now checks ADA sufficiency before building the transaction, logs a clear error, releases the UTxO, and aborts the batch gracefully, preventing failed or wasteful transactions.	✅ Fixed
004	Timestamps Should Be Handled in UTC	Currently, reading timestamps are used directly from the database and serialized into metadata without explicitly converting them to UTC.		Pending

5. Optimization Notes

We did not focus on performance optimizations at the system level. However, we introduced a key improvement in our tokenization strategy by implementing batching of minting operations into a single transaction. Instead of minting tokens individually, the system now batches multiple readings into one transaction. This significantly reduces transaction fees and improves throughput, making the overall process more cost-efficient and scalable.

6. Tools and Environments Used

- *Python 3.10 – Core language used for scripts, orchestration, and application logic.*
- *Uvicorn 0.24.0 – ASGI server for running the API.*
- *PyCardano – Library for building, signing, and submitting Cardano transactions.*
- *Blockfrost API (via BlockFrostChainContext) – Interface for blockchain queries and transaction submission.*
- *Cardano-node (Preview Testnet) – Provides network connectivity and transaction propagation.*
- *Logging (Python standard library) – Used for monitoring, debugging, and audit trails.*
- *PostgreSQL – Database for storing readings and tokenization records.*

7. Remaining Issues / Next Steps

Analyze generation of data

The next step in expanding the analysis is to integrate generation data alongside consumption data to create a full view of the energy flow. This requires consolidating data sources from production sites, consumption points, and transmission infrastructure into a unified model. By mapping generation outputs against consumption patterns, we can identify imbalances and trace where losses may be occurring, whether in transmission, distribution, or at the point of use. This approach will also involve developing metrics to quantify inefficiencies and applying analytics or visualization tools to highlight critical loss areas. Once this framework is established, it will enable targeted interventions to reduce waste, optimize operations, and improve overall system reliability.

Expand to other electricity consumers

Expand the prototype beyond its initial scope by incorporating additional categories of electricity consumers, enabling a more representative and realistic picture of demand. In parallel, the minimum viable product (MVP) should be designed to scale up and integrate the full ecosystem of the country’s electricity sector—covering all registered generators, transmission infrastructure, and consumer groups. This broader integration will provide a comprehensive baseline for system-wide analysis, ensuring that both supply and demand dynamics are accurately captured. By doing so, the MVP will not only validate the prototype’s approach but also lay the foundation for more advanced features such as predictive modeling, optimization of energy flows, and policy or investment scenario testing.

Unresolved / Pending Improvements:

- Automated testing is not yet in place; current validation has been performed through manual end-to-end tests.
- UTC timestamp handling needs to be fixed to ensure consistent data recording across all environments.
- Some blockchain non–happy path scenarios (e.g., network failures, chain reorgs, or invalid transactions) still need more robust handling to ensure graceful recovery.