SDG BLOCKCHAIN ACCELERATOR

Prototype (PoC) Report – Template

1. Project Information

- **Project Name:** Plastiks
- **Challenge & UNDP Office:** Armenia
- **Document Version:** v.2

2. Project Overview

Overview of Cardano

The Plastik Smart Contract Ecosystem is deployed on Cardano, leveraging its latest Conway Era advancements for improved scalability and smart contract execution. The Conway Era introduces several innovations beneficial to this ecosystem:

• Plutus V2 Enhancements: Provides more efficient scripting, reducing transaction size and execution costs.

• UTxO and Reference Scripts: Enables optimized contract execution, reducing on chain storage needs and improving performance.

• Stake-Based Certification: Strengthens the verification of plastic recovery projects by integrating stake-based voting mechanisms.

The document ensures a scalable, efficient, and decentralized approach to plastic recovery incentives, NFT-based sustainability credits, and transparency. This document provides a comprehensive technical breakdown of the Plastik Smart Contract Ecosystem on Cardano, including:

• Smart Contract Design: A modular architecture using Plutus V2 and the UTxO model.

• Security Mechanisms: Implementation of multi-signature approvals, formal verification, and commit-reveal schemes.

• Blockchain Integration: Use of reference scripts, inline datums, and CIP-68 metadata.

• Risk Mitigation: Identification and resolution strategies for contract vulnerabilities.

• Deployment & Validation: Simulation of security risks and real-time contract monitoring.

The document ensures a scalable, efficient, and decentralized approach to plastic recovery incentives,NFT-based sustainability credits transparency. This provides a comprehensive overview of the Plastik Smart Contract Ecosystem on Cardano, detailing its integration approaches, technical components, security mechanisms, potential risks, and mitigation strategies to ensure seamless development and deployment.

Integration Approaches

Smart Contract Architecture

• Plutus-based Smart Contracts: All contracts are written in Plutus V2, leveraging UTxO-based transaction validation.

Core Contracts:

• NFTStoreFrontV4: Handles NFT minting, sales, including lazy minting mechanisms.

• PlastikCryptoV2: Ensures cryptographic signature verification structured signing.

• Plastik Burner: Implements automatic token burning, i.e. discarding the tokens to

maintain token economy, same process followed in existing system.

• PlastikNFTV4 : Manages NFT issuance, ownership transfers.

• PlastikPRGV3: Issues environmental impact NFTs based on verified recycling contributions.

• Plastic Recovery Projects: Maintains a whitelisted registry of verified plastic recovery initiatives.

• Verified Accounts: Ensures KYC-verified users interact with core contracts.

• PlastikRoleV2: Implements multi-signature role-based access control (RBAC) to prevent unauthorized contract modifications.

• UtilsV2: Provides cryptographic utilities, structured hashing, and reusable helper Functions.

APIs & Integrations:

● Web3 Wallet Integration: Support for Cardano-compatible wallets including Nami, Eternl, and Yoroi, enabling user authentication and transaction signing.

● Sustainability Data Sources: APIs for verified external sustainability data providers to enhance reporting and validation.

● Cardano Blockchain Explorer APIs: Real-time integration with blockchain explorers for tracking and verifying issued plastic and CO2 credits.

● Payment Gateway Integration: Stripe is integrated into the platform, allowing users to make payments via credit or debit card.

● Reporting & Analytics Tools: Integration with third-party visualization tools for generating detailed sustainability impact reports.

Blockchain Implementation:

Blockchain: Cardano (Native Implementation) for issuing and verifying plastic credits as NFTs.

● Smart Contracts:

○ Plutus scripts for executing on-chain logic related to plastic credit issuance and validation.

○ Monetary Policy Scripts for defining the NFT minting rules, ensuring compliance with sustainability verification standards.

● Tokenization:

○ Plastic credits will be issued as NFTs on Cardano, embedding immutable metadata that includes both plastic recovery data and CO2 savings information.

○ Each NFT will contain structured metadata stored on IPFS, linking it to its respective sustainability data.

3. Repository Structure

Suggested Structure:

```
/prototype
    /offchainCode → Smart contract Offchain code
    /contracts    → Smart contract source code
    /tests        → Unit and integration tests
    /docs         → README.md, deployment guide, screenshots
```

4. Build Instructions

This provides a comprehensive technical breakdown of the Plastik Smart Contract Ecosystem on Cardano, including:
Smart Contract Design: A modular architecture using Plutus V2 and the UTxO model.
Security Mechanisms: Implementation of multi-signature approvals, formal verification, and commit-reveal schemes.
Blockchain Integration: Use of reference scripts, inline datums and CIP-68 metadata.
Risk Mitigation: Identification and resolution strategies for contract vulnerabilities.
Deployment & Validation: Simulation of security risks and real-time contract monitoring.

The Plastik Smart Contract Ecosystem is deployed on Cardano, leveraging its latest Conway Era advancements for improved scalability, and smart contract execution. The Conway Era introduces several innovations beneficial to this ecosystem:
Plutus V2 Enhancements: Provides more efficient scripting, reducing transaction size and execution costs.
UTxO and Reference Scripts: Enables optimized contract execution, reducing on-chain storage needs and improving performance.
Stake-Based Certification: Strengthens the verification of plastic recovery projects by integrating stake-based voting mechanisms.
The document ensures a scalable, efficient, and decentralized approach to plastic recovery incentives, NFT-based sustainability credits, and transparency. This document provides a comprehensive technical breakdown of the Plastik Smart Contract Ecosystem on Cardano.

Example:

```
# Compile Smart contracts

cabal clean – Removes previous build artifacts to ensure a clean slate and
avoid conflicts.

cabal update – Refreshes the local package repository index so that
dependencies are up-to-date.

cabal build – Compiles the smart contract code into an executable form.

cabal repl – Launches an interactive environment for testing your smart
contract functions.
```

- Generate the CBOR Hex File
- GHC – 8.10.7
- Cabal – 3.8.1.0
- Haskell and Plutus
- Cardano Node (Conway era) (10.4.1)
- Cardano addresses 4.0.0
- Deploy the Smart Contract
    - Blockchain: Cardano Blockchain that supports Plutus.
    - Deployment: Deploy the smart contract to the blockchain.
- Smart Contract Test
    - cabal clean //Removes previous build artifacts to ensure a clean slate and avoid conflicts.
    - cabal update //Refreshes the local package repository index so that dependencies are up-to-date.
    - cabal build //Compiles the smart contract code into an executable form.
    - cabal repl //Launches an interactive environment for testing your smart contract functions.

5. Test Instructions

This details test cases of the Plastiks Smart Contract on Mainnet, covering various functionalities like CryptoV2, Burner, TokenMint, TokenMintOnChain, PRGMint, PRGV3, RecoverProjects, RoleV2, and Verify Accounts, and WhiteList. Each section outlines user stories, conditions, steps, expected and actual outputs, and the test result (mostly "Pass"). It also includes API endpoints, request bodies, and transaction hashes for successful operations, and error messages for failed scenarios like minting with excessively large amounts or unauthorized actions.

Example:

```
Run tests

Unit tests and property tests (Tasty + QuickCheck) are typically run via:

cabal test

Generate `.plutus` files / compile validators

Many Plutus projects expose a `plutus` or `scripts` target to produce
serialized validator files. Check the repo’s build scripts or `Makefile`.
```

- Keep QuickCheck-based property tests (as you have been doing) for critical invariants: signature checks, datum format, duplicate ID checks, reputation calculations, role grants, and registry updates.

- Use Tasty to group unit tests and to run deterministic test cases for both happy and failure paths.
- Add a CI job to compile all modules and run tests on pushes and PRs.

6. Deployment Instructions

This document outlines the deployment plan for smart contracts to the Cardano Mainnet.
It defines milestones, timelines, responsibilities, and validation checkpoints to ensure a secure and smooth rollout.

Steps	Description
S1. Preparation	Review contract logic, complete functional testing on preprod/testnet and validate contract parameters (admin keys, token names, policies).
S2. Pre-Deployment Testing (Preprod)	Deploy contracts to preprod, generate script addresses & policy IDs, run mint/burn/transfer/admin tests, collect logs, obtain sign-off.
S3. Mainnet Pre-Checks	Confirm Plutus hash, validate compiled build, update `.env` with mainnet admin keys, set up wallets & backups, confirm funding and access rights.
S4. Deployment Artifacts & Off-Chain Setup	Generate mainnet script addresses & policy IDs, configure backend service, connect to Blockfrost API & IPFS. Maintain manifest file.
S5. Transaction Preparation	Prepare minting/initialization transactions, dry-run identical preprod transactions, confirm datum & redeemer formats, validate ADA sufficiency.
S6. Initial Mainnet Deployment	Deploy smart contracts to mainnet, run minimal non-critical transactions, validate balances & logs, confirm contract execution.
S7. Enable Core Functionality	Initialize tokens, enable validators
S8. Post-Deployment Validation	Verify contracts behave identically to preprod, run regression suite, confirm token balances & datums, circulate deployment summary. **If there are any issues in the production we will revert back it to the identified last stable version.**

7. Testnet / Emulator Results

This ensures a scalable, efficient, and decentralized approach to plastic recovery incentives, NFT-based sustainability credits transparency. This provides a comprehensive overview of the Plastik Smart Contract Ecosystem on Cardano, detailing its integration approaches, technical components, security mechanisms, potential risks, and mitigation strategies to ensure seamless development and deployment.

Link — Infrastructure Verification Report .pdf

No.	Smart Contract Name	Purpose
1	VerifiedAccounts	UpdateVerification, CheckVerification, and UpdateValidator enable managing verifiers and validators by adding, updating, or validating a pubKeyHash based on its assigned boolean status.
2	PlastikRoleV2	GrantMinter assigns the minter role to a given pubKeyHash, while VerifyMinter checks if a pubKeyHash is in the minters list and returns a boolean result.
3	PlastikRecoveryProjects	UpdateProject lets the admin set a project's pubKeyHash status using a boolean value, while CheckProject retrieves the current status of a given pubKeyHash.
4	PlastikBurner	CurrentBurnRate updates the burnPerKg in the inline datum based on total supply and MaxBurnYear, SetPlasticProduction lets the admin update the plastic production amount, and SetMaxBurnYear allows the admin to adjust the maximum yearly burn limit.
5	PlastikCryptoV2	RedeemVoucher validates that an NFTVoucher redeemer is signed by its creator, while CompleteSale verifies the details of a SellRequest redeemer from the seller.

6	PlastikPRGV3	VerifyPlastikStore uses datum to track total supply and PRE NFT transfers, allowing the admin to verify and redeem completed roadmap details.
7	PlastikToken	Pause/Unpause let the admin toggle the transferability of the Plastik token by updating the inline datum, SetLocker allows the admin to assign locker status to a pubKeyHash, Transfer enables a sender to send tokens to a recipient with signature verification, and TransferFrom allows a locker to move tokens from a sender to a recipient with both parties' signatures.
8	WhiteList	UpdateEntries lets an admin update the whitelist by mapping PubKeyHash values to booleans, while Verify checks if a given PubKeyHash is present and marked as whitelisted.
9	PlastikTokenMint	The contract takes the admin's pubKeyHash as a redeemer to produce a Plutus file, which can then be used to generate a script address and policy ID.
10	PCMint	Mint and transfer a plastic credit issuer to plastic credit buyer

8. Dependencies & Environment

- **Blockchain Environment:**
    - Native deployment on **Cardano mainnet**, using **Plutus V2** smart contracts and the **Extended UTxO (EUTxO)** model for deterministic validation, scalability, and low transaction costs.
    - Utilization of **Reference Scripts** and **Inline Datums** to reduce on-chain storage and improve contract efficiency.
    - Implementation of **CIP-68 metadata standard** to structure NFT data and ensure interoperability with sustainability reporting systems.
    - Integration of **Stake-Based Certification** to enable decentralized validation and governance of verified plastic recovery projects.

- **Core Smart Contract Dependencies:**
    - **NFTStoreFrontV4:** NFT minting and sales, including lazy minting mechanisms.
    - **PlastikCryptoV2:** Cryptographic signature verification and structured signing.
    - **PlastikBurner:** Automated token burning to maintain a balanced token economy.
    - **PlastikNFTV4:** NFT issuance, metadata linkage, and ownership transfer management.
    - **PlastikPRGV3:** Environmental NFT generation based on verified plastic recovery metrics.
    - **PlastikRoleV2:** Multi-signature, role-based access control (RBAC) to prevent unauthorized contract modifications.
    - **UtilsV2:** Shared cryptographic utilities, hashing functions, and reusable helpers for smart contracts.
- **External Integrations:**
    - **Web3 Wallets (Nami, Eternl, Yoroi):** Used for user authentication, transaction signing, and wallet connectivity.
    - **Sustainability Data APIs:** Integration with verified external data sources to validate environmental performance and traceability.
    - **Blockchain Explorer APIs:** Real-time verification of issued plastic credits, CO_2 certificates, and contract activity.
    - **Stripe Payment Gateway:** Enables fiat-to-crypto transactions for credit purchases and platform services.
    - **Analytics & Reporting Tools:** Integration with third-party data visualization systems for impact reporting and compliance dashboards.
- **Technical Stack & Development Environment:**
    - **Languages:** Haskell (Plutus logic), Haskell and Plutus (contract verification).
    - **Testing & Simulation:** Plutus Playground and Cardano Testnet for pre-deployment validation.

    - **Infrastructure:** Docker-based containerization for backend services, ensuring portability and version control.
    - **CI/CD Pipelines:** Automated testing, linting, and deployment workflows for smart contract integrity.
- **Security and Verification:**
    - **Multi-signature authorization** implemented in PlastikRoleV2.
    - **Formal verification** of smart contracts to ensure deterministic execution and prevent vulnerabilities.
    - **Commit-reveal mechanisms** to mitigate front-running and manipulation risks.
    - **Real-time monitoring scripts** for on-chain event tracking and anomaly detection.
- **Off-chain and Cloud Environment:**
    - **Hybrid architecture** combining decentralized data storage (IPFS) with cloud orchestration (AWS/Azure compatible).
    - **Node synchronization services** to maintain alignment with Cardano ledger updates.
    - **Backend APIs** designed for scalability and secure data exchange between blockchain and sustainability systems.
- **Dependency Management and Upgrades:**
    - All dependencies version-controlled and aligned with **Cardano's Conway Era** upgrades.
    - Continuous monitoring of protocol updates, CIPs, and Plutus revisions to ensure backward compatibility.
    - Regular audits to maintain operational continuity and long-term ecosystem stability.

9. Demo / Walkthrough

- Transfer of minted credit on cardano from issuer to purchaser

- Screenshots from the blockchain verifying that Plastic and CO2 Credits have been successfully transferred from issuer to buyer.

10. Remaining Issues / Next Steps

(Mention pending improvements, planned features, or optimizations.)

New features:

- **Redesign of the create credit form/screen: OLD vs new here**
    - **Including more metrics and new types of waste**
- **Redesign of the Plastic credit NFT design: see New vs Old here**
-