



Integration Report: Cohort 1 Pilot Projects

Prepared by: EMURGO Labs Engineering Team

Version: 3.0

1. Executive Summary

The first cohort of pilot projects under the EMURGO–UNDP accelerator program focused on exploring the application of blockchain technology for sustainable development goals (SDGs) across multiple geographic and sectoral contexts.

This report documents the **technical integration, prototyping, and validation activities** conducted during the program, detailing the EMURGO-developed **reference repositories, code scaffolds, and technical templates** utilized by participating teams. The purpose of this report is threefold:

1. **To provide a structured account of technical resources and frameworks used by Cohort 1 projects.**
2. **To demonstrate the integration and testing process**, including validation of smart contract logic, off-chain backend interactions, and user-facing application prototypes.
3. **To capture lessons learned and provide a foundation** for Cohort 2 and future accelerator cycles.

All integration activities were carried out in alignment with the UNDP accelerator program objectives, ensuring **functional prototypes with measurable SDG impact potential** were produced by each team.

2. Introduction

2.1 Background

Blockchain technology offers a decentralized and transparent approach to data integrity, asset management, and financial inclusion. By leveraging distributed ledger capabilities, pilot projects in Cohort 1 explored innovative solutions in areas such as **tokenized carbon credits, supply chain traceability, crowdfunding for energy access, and incentivized recycling initiatives**.

To facilitate these explorations, EMURGO Labs provided a **comprehensive set of reference repositories, smart contract templates, backend scaffolds, and front-end DApp frameworks**. These resources enabled teams to focus on **problem-specific customization** rather than foundational blockchain infrastructure.

2.2 Objective of the Report

The primary objective of this report is to **document the integration and testing of EMURGO technical resources** across Cohort 1 pilot projects. Specific goals include:

- Cataloging the **reference repositories** provided to each team.
- Mapping each pilot project's **solution focus** to the relevant repositories used for integration and development.
- Describing the **technical integration methodology**, including deployment, customization, and testing procedures.
- Providing a **comprehensive analysis of outcomes, challenges, and lessons learned**, to inform the design of Cohort 2 and subsequent accelerator programs.

3. EMURGO Technical Reference Resources

3.1 Overview of Repositories and Scaffolds

EMURGO Labs provided a curated set of repositories, smart contract templates, and development scaffolds intended to accelerate blockchain prototyping for SDG-focused projects. Each repository served a distinct purpose, addressing **on-chain contract logic, off-chain backend orchestration, or front-end application interaction**.

The following table summarizes the **available repositories and their intended application domains**:

No.	Group	Repository	Description	Integration Purpose
1	Aiken Smart Contracts	hello-world-aiken-smart-contract	Basic Aiken smart contract demonstrating deployment and interaction	Served as the initial onboarding framework for new developers to understand deployment procedures, transaction submission, and contract verification on the Cardano blockchain.
2	Aiken Smart Contracts	guessing-game-aiken-smart-contract	Contract illustrating state transitions and validation logic	Used to explore dynamic contract behavior, state management, and validation rules in prototype applications.
3	Aiken Smart Contracts	crowdfunding-aiken-smart-contract	Implements crowdfunding logic with pledge collection and refund mechanics	Provided a ready-made framework for projects focusing on fundraising, pooled asset management, and automated refund logic, particularly relevant for energy and climate finance projects.
4	Aiken Smart Contracts	aiken-unit-test	Testing suite for validating contract behavior	Enabled teams to perform unit testing and debugging of smart contracts, ensuring logical correctness and resilience of deployed contract logic.
5	Plutus Smart Contracts	nft-ticketing-plutus-smart-contract	NFT minting and lifecycle management for digital assets	Repurposed for tracking physical and digital assets, including carbon credits, supply chain products, and tokenized incentives.

6	Plutus Smart Contracts	<code>credit-scoring-plutus-smart-contract</code>	Implements a tokenized credit scoring mechanism	Provided reference logic for assessing eligibility, verifying contributions, or tracking performance metrics in financial inclusion and incentive-focused applications.
7	Plutus Smart Contracts	<code>lending-plutus-smart-contract</code>	Lending protocol using scoring tokens for eligibility and risk evaluation	Adapted to model peer-to-peer trading, micro-lending, and conditional asset transfers.
8	Backend (NestJS)	<code>template-backend-nestjs</code>	Scalable API boilerplate for dApp backends	Used to orchestrate off-chain operations, manage user sessions, and connect to blockchain endpoints.
9	Backend (NestJS)	<code>wallet-management-backend-nestjs</code>	Service handling wallet management, signing flows, and transaction tracking	Enabled secure wallet interactions, transaction signing, and integration with front-end DApps.
10	Cardano DApp	<code>cardano-dapp-template-nextjs</code>	Front-end scaffold integrating wallet connectors and on-chain interactions	Served as the primary interface for end-users to interact with blockchain applications, facilitating testing and prototype validation.
11	Tokenization APIs	<code>API-Carbon-credit-tokenization</code>	Tokenization APIs initially meant for Carbon credit through the Nexchange project	Enabled tokenization of any assets through an API approach, from minting to buying and selling orders

3.2 Rationale for Repository Selection

Each repository and scaffold was carefully selected to **maximize the applicability to SDG-focused projects**:

1. **Aiken Smart Contracts**: Ideal for onboarding, prototyping simple contracts, and testing UTxO-based patterns.
2. **Plutus Smart Contracts**: Provided higher-level financial logic, NFT capabilities, and risk scoring mechanisms for complex use cases.
3. **Backend Scaffolds**: Ensured consistent, scalable, and secure integration of off-chain data flows, critical for real-world deployments.
4. **Front-end DApps**: Allowed rapid interface development for end-users, including wallet interactions, transaction submission, and data visualization.

By providing modular and reusable frameworks, EMURGO enabled teams to **focus on domain-specific adaptation** without needing to build low-level blockchain infrastructure from scratch.

4. Cohort 1 Pilot Projects Integration

Each Cohort 1 project leveraged one or more of the reference repositories based on its **use case, operational requirements, and SDG focus**. The following table provides a detailed mapping:

No.	Project	UNDP Challenge Owner	Focus Area	Reference Repositories Applied	Integration Summary
1	Thallo	UNDP Tanzania	Inclusive climate finance and tokenized carbon credits	Nexchange Tokenization APIs , template-backend-nestjs , cardano-dapp-template-nextjs	Adapted the Nexchange APIs logic to tokenize climate projects and manage user contributions; backend orchestrated carbon registry workflows; DApp template provided user-facing

					interface for token management.
2	Socious Fund	UNDP Mauritius & Seychelles	Transparent crowdfunding for solar energy access	N/A	The team is Cardano native and have working platform. Scious prepared a whitelabel solution for UNDP Mauritius
3	Atlas Ledger	UNDP Burkina Faso	Donor-backed reforestation systems	lending-plutus-smart-contract , template-backend-nestjs , cardano-dapp-template-nextjs	Lending logic modeled donor funding cycles; backend orchestrated data aggregation; DApp frontend enabled visualization of reforestation progress.
4	Genius Tags (ClimateAid)	UNDP Malawi	Predictive humanitarian aid delivery	hello-world-aiken-smart-contract , template-backend-nestjs	Lightweight Aiken contract used to model event triggers for anticipatory aid; backend provided integration with external data sources.
5	Grinplus	UNDP Tanzania	Peer-to-peer solar energy trading	lending-plutus-smart-contract , wallet-management-backend-nestjs , cardano-dapp	Lending contract logic repurposed for energy credit exchange; wallet backend managed token transfers between peers; DApp

				<code>-template-nextjs</code>	facilitated user interaction.
6	Karbon Ledger	UNDP India	Tokenizing textile wastewater reductions	N/A	Team is Cardano native with a working product. Karbon Ledger worked with UNDP india to add their textile products to their marketplace
7	Afrikabal	UNDP Malaysia	Supply chain traceability for climate-friendly farming	<code>nft-ticketing-plutus-smart-contract</code> , <code>wallet-management-backend-nestjs</code>	NFTs represented produce batches; backend ensured traceable verification for supply chain participants.
8	Cladfy	UNDP Bangladesh	Transparent fund disbursement for local climate adaptation	<code>credit-scoring-plutus-smart-contract</code> , <code>template-backend-nestjs</code> , <code>cardano-dapp-template-nextjs</code>	Credit scoring logic determined disbursement eligibility; backend handled fund allocation; DApp displayed transparency dashboards.
9	ZenGate Global (Palmyra Pro)	UNDP Bangladesh	Agricultural and procurement traceability	N/A	Team is Cardano native. Zengate exposed APIs to UNDP bangladesh allowing them to trace Agri products and anchor events on-chain

10	Creative Operations	UNDP Georgia	Incentivized e-waste collection	<code>credit-scoring-plutus-smart-contract, cardano-dapp-template-nextjs, wallet-management-backend-nestjs,</code>	Credit scoring logic adapted for reward distribution; DApp enabled collection tracking and user engagement. Wallet management in the backend to enhance user experience
11	Plastiks	UNDP Armenia	Plastic recovery traceability and tokenized recycling	N/A	Team is Cardano Native. The plastics team had already worked on a working PoC on Cardano testnet, and will be running the first pilot on Cardano with UNDP Armenia
12	IotaOrigin UG / BlackFrog	UNDP Istanbul Regional Hub	Mineral traceability and peacebuilding	<code>nft-ticketing-plutus-smart-contract, template-backend-nestjs</code>	NFT contracts tracked mineral batches; backend verified provenance and compliance.
13	Unicorn.eth	UNDP Nordic Office	Crowdfunded microgrids and verifiable energy access	<code><u>crowdfunding-aiken-smart-contract</u>, wallet-management-backend-nestjs, cardano-dapp-template-nextjs</code>	Crowdfunding logic used for microgrid financing; wallet backend managed contributors' tokens; DApp provided interface for monitoring energy deployment.

4.1 Complementary Tools and Integrations (Cross-Cohort Overview)

Tool / Integration	Function	Used By	Purpose / Outcome
Supabase	Backend database & authentication	Thallo, Karbon Ledger, Reloop	Store off-chain project data, user info, and metadata
IPFS	Decentralized storage	Thallo, Karbon Ledger, ZenGate, Plastiks	Store metadata, carbon credit proofs, NFT metadata
Next.js / React frontend	Dashboard / user interface	Thallo, Atlas Ledger, Karbon Ledger, Reloop, Cladfy	To build dashboards for users and admins to interact with the platform
Chart.js / Grafana	Data visualization	Thallo, Reloop	Visualize KPIs, carbon credit metrics, or platform usage
Lucid / PyCardano	Blockchain interactions & off-chain scripts	Atlas Ledger, Afrikabal, Grinplus	To read/write UTxOs, build transactions, and interact with Cardano nodes
Blockfrost API	Cardano blockchain node access	Grinplus, Karbon Ledger, Cladfy, ZenGate, Reloop	Query blockchain data, submit transactions, fetch UTxOs
NestJS / Flask / Node.js backend	Off-chain server logic	Atlas Ledger, Cladfy, IotaOrigin UG	Handle API requests, off-chain computation, and integration with smart contracts

Redis / Postgres	Data caching & persistence	ZenGate	Store transaction queues, pre-process data, and maintain backend state
IoT sensors / Remote sensing	Environmental data capture	Karbon Ledger, Genius Tags	Feed verified environmental metrics to carbon credit systems
Stripe / Mobile money integration	Payments / off-ramping	Cladfy, Plastiks	Collect payments for credits or NFTs, enable fiat on/off-ramping
Fireblocks	Custody & secure transaction signing	Reloop	Safely store keys and sign blockchain transactions
Ogmios / Kupo	Node interaction & indexer	Reloop	Access UTxOs, blockchain state, and index transactions for dashboards
Preprod / Testnet nodes	Blockchain testing	Most projects	Test smart contracts, validators, and integrations before mainnet deployment
@meshsdk/react & @meshsdk/core	Wallet and transaction management	Unicorn.eth	Simplify frontend wallet connections, transactions, and token handling

5. Technical Integration Methodology

5.1 Smart Contract Deployment

All projects followed a structured process for smart contract deployment:

1. **Repository Access:** Teams have been granted access to the relevant EMURGO repositories to local development environments.
2. **Environment Configuration:** Preprod and testnet parameters were configured for Cardano blockchain deployment.
3. **Compilation and Validation:** Contracts were compiled and subjected to unit tests (using `aiken-unit-test` where applicable) to ensure logical correctness.
4. **Deployment:** Smart contracts were deployed to Cardano testnets; functional verification was conducted with simulated transactions.

5.2 Backend Integration

Backends built on NestJS templates served as the off-chain orchestration layer:

- Managed user authentication and session handling.
- Coordinated transaction signing via wallet management services.
- Aggregated data for reporting and visualization dashboards.

5.3 Front-End DApp Integration

Front-end DApps built with the Next.js template facilitated:

- Wallet connection and transaction submission.
- Visualization of on-chain events and user-specific data.
- Real-time feedback during prototype testing.

5.4 Testing and Validation

Testing included:

- **Unit testing:** Verifying contract logic correctness.
- **Integration testing:** Ensuring end-to-end functionality between front-end, backend, and blockchain.
- **Sandbox testing:** Deploying functional PoCs in isolated environments for risk-free evaluation.

6. Lessons Learned and Recommendations

- **Repository scaffolds significantly reduced setup time**, enabling teams to focus on domain-specific customization.
- **Modular architecture enabled diverse use cases**, from crowdfunding and microgrid financing to tokenized traceability of carbon credits and agricultural produce.
- **Technical mentorship was critical** for guiding teams on contract optimization, backend integration, and front-end alignment.
- **Future recommendations:**
 - Expand test data templates and prebuilt use-case modules.
 - Develop additional tutorials and example workflows for emerging SDG-aligned use cases.
 - Integrate more robust monitoring tools for real-time impact measurement.

7. Conclusion

Cohort 1 pilot projects successfully **integrated EMURGO's reference repositories, scaffolds, and DApp templates** to deliver functional PoCs aligned with SDG objectives.

These technical resources acted as the **functional blueprints**, enabling rapid development, reliable testing, and user-centric DApp interfaces. Insights from Cohort 1 will guide **future improvements, repository expansion, and the onboarding of Cohort 2 projects**, enhancing both scalability and sustainability of blockchain solutions for global development challenges.