📘 Implementation Guide for SDG-Focused Blockchain Projects

EMURGO x UNDP Blockchain Accelerator
Prepared by:
Version: March 2025

🎯 Purpose

This guide provides SDG teams with a comprehensive, step-by-step process to design and deploy blockchain-powered Proofs-of-Concept (PoCs) using the Cardano ecosystem. It draws from real challenges submitted by UNDP country offices and maps each challenge to sample verticals: **Climate Action**, **Digital Identity**, **Financial Inclusion** and **Supply Chain & Product Traceability.**

🔧 Core Design Considerations

Layer	Component	Purpose	Tools on Cardano
Identity	DID / VC Issuance	Attest roles, eligibility, impact	anoncreds-cardano, Veridian SDKs
Wallet	Custody & Signing	Secure access to funds & credentials	Lace, Nami, Eternl, Cardano Identity wallet
Smart Contracts	Rules and automation	Loan logic, payout conditions, traceability	Plutus, Marlowe, Aiken
Data & Metadata	Track assets and flows	Represent impact, VCs, token properties	CIP-68, IPFS, Koios, Blockfrost, GraphQL
UI & APIs	Human interaction layer	Access, dashboards, verification	Lucid SDK, Mesh, React, GraphQL APIs

🌱 Vertical 1: Climate Action

🧠 Use Case:

“Design a transparent blockchain-based Payment for Ecosystem Services (PES) program.”

1. Functional Flow

1. Farmer performs ecosystem work (e.g., reforestation).
2. Verifier validates activity (human or sensor-based).
3. VC issued to farmer for validated activity.
4. CIP-68 NFT minted with impact metadata (tree type, GPS, timestamp).
5. Smart contract automates fund disbursement in ADA or USDA.

2. Architecture

```
[Farmer] → [Verifier App] → [VC Issued]
            ↓
     [Metadata NFT Minted (CIP-68)]
            ↓
 [Smart Contract → Disburses Funds]
            ↓
   [Farmer Wallet (Custodial or Self)]
```

3. Tools & Libraries

Layer	Tool	Description
Identity	Veridian	Issue Verifiable Credentials (location, activity type)
Metadata	CIP-68	Encode GPS, timestamp, ecosystem service in NFT
Contracts	AIKEN	Trigger fund release based on verified NFTs or VC
UI	Mesh or Lucid SDK	Build field officer dashboard or NGO portal
Storage	IPFS or Arweave	Store photos, audio, sensor logs off-chain

4. Dev Tips

- Use **Mesh** for composable dApp elements like wallet connect, mint, and upload.
- Use **GraphQL** from Koios to monitor transaction status.
- Start with manual VC issuance; move to sensor/IoT later

5. Output Expectations

- GitHub repo with Aiken smart contract and metadata NFT minting logic
- Sample Verifiable Credential (VC) JSON issued to a farmer
- Screenshots or mock UI for verifier dashboard and disbursement trigger
- CIP-68 NFT with encoded GPS, timestamp, and activity hash
- Demo flow: manual VC issuance → NFT mint → fund release

🧾 Vertical 2: Digital Identity

🧠 Use Case:

“Develop a decentralized ID system to support equitable access to services.”

1. Functional Flow

1. Citizen applies via mobile app or kiosk.
2. DID and Verifiable Credential (VC) issued by government or trusted verifier.
3. Credential stored in user wallet (self-custodied or managed).
4. Employer or service provider requests and verifies credentials.
5. Citizen gets access to service (employment, health, subsidy).

2. Architecture

```
[Citizen Onboards] → [VC Issuer]
```

```
    ↓                   ↓
[Wallet (DID+VC)]  ← [Veridian/ anoncreds]
    ↓
[Verifier Requests Proof]
    ↓
[Proof Presented + Validated]
```

3. Tools & Libraries

Layer	Tool	Description
DID	Veridian / anoncreds	DID issuance or anoncreds spec
VC	Veridian VC Format	JSON-LD-based schemas supported by Cardano tools
Wallet	Lace	Store credentials with optional backend key control
VC Verifier	Veridian Verifier Tooling	Confirm authenticity of VC + revocation status
Schema Registry	JSON Schema + DID:Cardano	Standardize identity claims (e.g., work status, ID verified)

4. Dev Tips

- For mobile onboarding, abstract wallet interaction via session signing.
- Use low-data DID method to reduce on-chain overhead.
- Start with employment credentials (low risk), expand to health/public identity.

5. Output Expectations

- GitHub repo with DID issuance and VC storage integration

- Sample identity VC (e.g., "ID Verified", "Employed")
- Working prototype of VC request & proof presentation
- Test credentials issued via Veridian or anoncreds
- Demo: user onboarding → VC issuance → proof verification by employer

💸 Vertical 3: Financial Inclusion & Payments

🧠 Use Case::

“Build a blockchain-based microloan platform for underserved small businesses.”

1. Functional Flow

1. MSME registers and completes VC-based identity checks.
2. Credit score assigned manually or via oracle.
3. Smart contract deployed to manage loan.
4. Funds disbursed in stablecoin (or ADA).
5. Repayment tracked on-chain; default conditions enforced.

2. Architecture

```
[MSME Registration]
     ↓
 [VC Issued (Credibility, Income)]
     ↓
 [Loan Request via dApp]
     ↓
 [ Smart Contract Deployed]
     ↓
 [Funds Disbursed + Vault Created]
     ↓
 [Repayment Events Tracked]
```

3. Tools & Libraries

Layer	Tool	Description
Contracts	Aiken	Use Aiken to develop collateralized loan lifecycle with credit scoring
Identity	Community-issued VC	Attests income level, business license, etc.
Frontend	Lucid SDK + React	Borrower-facing interface
Dashboard	GraphQL + Charting	NGO/lender view on repayments, defaults
Oracles	Off-chain webhook or API adapter	For credit scores or fiat rates

4. Dev Tips

- Use Lace wallet with testnet funds for MVP demo.
- Store VC reference off-chain and hash on-chain for proof.
- Customize contracts to reflect NGO policies (e.g., gender-based incentives).

5. Output Expectations

- Aiken or Plutus smart contract repo for loan creation, repayment tracking
- Sample borrower VC (e.g., income level, license)
- Loan lifecycle demo (register → fund → repay) with testnet transactions
- Borrower UI mockup or working front-end in React + Lucid
- Basic dashboard view showing repayment events (mock or real)

📦 Vertical 4: Supply Chain & Product Traceability

🧠 Use Case:

“Build a blockchain-based traceability system for agricultural exports (e.g., coffee or cocoa), ensuring product origin, sustainability claims, and fair trade certifications are verifiable.”

1. Functional Flow

- Farmer or cooperative registers and receives a VC attesting their certification (e.g., organic, fair trade).
- Each batch of product is minted as a native NFT with metadata (origin, harvest date, cert ID).
- As product moves through the supply chain (processor → exporter → retailer), each transfer is logged on-chain.
- End buyers can scan a QR code to verify origin, journey, and certifications.

2. Architecture

```
[Farmer Registration]
      ↓
[VC Issued: Organic / Fair Trade]
      ↓
[Batch NFT Minted (Coffee, Cocoa)]
      ↓
[Transfers Signed by Each Handler]
      ↓
[Retailer Display + QR for Proof of Origin]
```

3. Tools & Libraries

Layer	Tool	Description

Identity	Community-issued VC	Attests to certifications (organic, sustainability, etc.)
Asset	Cardano Native Tokens	NFTs minted to represent individual product batches
Contracts	Aiken or Lucid Script	Optional logic for verified handoffs or condition-based transfers
Frontend	Lucid SDK + QR Tool	Interface for scanning, tracking, and displaying product provenance
Metadata	CIP-68 or IPFS Hash	Store product info, cert links, and chain-of-custody metadata

4. Dev Tips

- Use **NFTs** as dynamic traceability records (not for collectibles but as supply chain passports).
- Ensure **each handoff** (e.g., from processor to exporter) includes a signed transfer + timestamp.
- Use **Verifiable Credentials** to prove upstream certifications, issued by recognized certifiers.
- For demo: simulate scan experience by generating a QR linking to an explorer or dApp page.

5. Output Expectations

- NFT minting script using Cardano native assets + metadata (CIP-68 or IPFS)
- Sample VC for certification (e.g., "Organic Certified: Yes")

- Mock or live scan demo (QR → dApp explorer of asset metadata)
- Chain of custody simulation: 3 signed transfer events
- GitHub repo with asset schema, traceability logic, and metadata structure

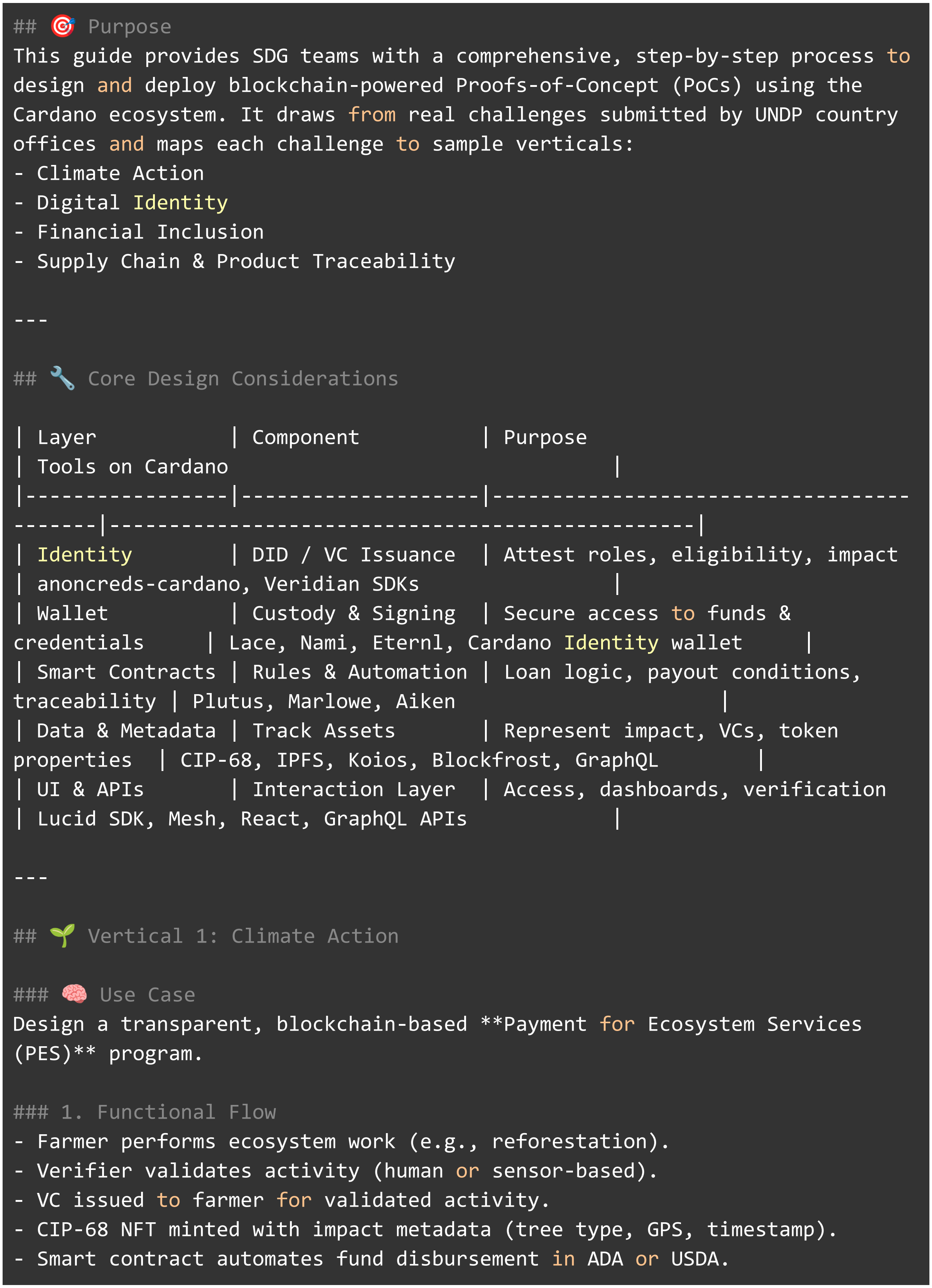
✅ General Implementation Checklist

Task	Tools / Outcome
Define actors, flow, and logic	Use PoC design canvas
Select wallet strategy	Lace, Eternl, or backend module
Choose VC tool	Veridian / anoncreds / JSON-LD
Fork from open source or build from scratch	Marlowe, Mesh
Mint tokens / NFTs if needed	CIP-68 + Mesh
Store off-chain data	IPFS, Arweave
Build front-end	Lucid SDK, React, Mesh
Document everything	GitHub with README + architecture.md

Github Markdown

```
# 📘 Implementation Guide for SDG-Focused Blockchain Projects
**EMURGO x UNDP Blockchain Accelerator**
_Prepared by:_
_Version: March 2025_

---
```



```
## 🎯 Purpose
This guide provides SDG teams with a comprehensive, step-by-step process to design and deploy blockchain-powered Proofs-of-Concept (PoCs) using the Cardano ecosystem. It draws from real challenges submitted by UNDP country offices and maps each challenge to sample verticals:
- Climate Action
- Digital Identity
- Financial Inclusion
- Supply Chain & Product Traceability

---

## 🔧 Core Design Considerations

| Layer           | Component          | Purpose                                   | Tools on Cardano                                |
|-----------------|--------------------|-------------------------------------------|-------------------------------------------------|
| Identity        | DID / VC Issuance  | Attest roles, eligibility, impact         | anoncreds-cardano, Veridian SDKs                |
| Wallet          | Custody & Signing  | Secure access to funds & credentials      | Lace, Nami, Eternl, Cardano Identity wallet     |
| Smart Contracts | Rules & Automation | Loan logic, payout conditions, traceability | Plutus, Marlowe, Aiken                          |
| Data & Metadata | Track Assets       | Represent impact, VCs, token properties   | CIP-68, IPFS, Koios, Blockfrost, GraphQL        |
| UI & APIs       | Interaction Layer  | Access, dashboards, verification          | Lucid SDK, Mesh, React, GraphQL APIs            |

---

## 🌱 Vertical 1: Climate Action

### 🧠 Use Case
Design a transparent, blockchain-based **Payment for Ecosystem Services (PES)** program.

### 1. Functional Flow
- Farmer performs ecosystem work (e.g., reforestation).
- Verifier validates activity (human or sensor-based).
- VC issued to farmer for validated activity.
- CIP-68 NFT minted with impact metadata (tree type, GPS, timestamp).
- Smart contract automates fund disbursement in ADA or USDA.
```

~~~
### 2. Architecture
```
[Farmer] → [Verifier App] → [VC Issued]
           ↓
     [Metadata NFT Minted (CIP-68)]
           ↓
 [Smart Contract → Disburses Funds]
           ↓
   [Farmer Wallet (Custodial or Self)]
```

### 3. Tools & Libraries

| Layer    | Tool              | Description
|
|----------|------------------|--------------------------------------------
---------|
| Identity | Veridian          | Issue Verifiable Credentials
|
| Metadata | CIP-68            | Encode GPS, timestamp, ecosystem data
|
| Contracts| Aiken             | Trigger fund release based on verified NFTs
or VC   |
| UI       | Mesh / Lucid SDK | Build field officer dashboard / NGO portal
|
| Storage  | IPFS / Arweave    | Store photos, audio, sensor logs
|

### 4. Dev Tips
- Use Mesh for wallet connect, mint, and upload.
- Monitor transactions via GraphQL (Koios).
- Start with manual VC issuance, move to sensor-based validation later.

### 5. Output Expectations
- GitHub repo with Aiken smart contract + metadata NFT logic
- Sample Verifiable Credential (VC) JSON issued to a farmer
- Screenshots or mock UI for verifier dashboard and disbursement trigger
- CIP-68 NFT with encoded GPS, timestamp, and activity hash
- Demo flow: manual VC issuance → NFT mint → fund release

---
~~~

~~~~
## 🧾 Vertical 2: Digital Identity

### 🧠 Use Case
Develop a decentralized ID system to support equitable access to services.

### 1. Functional Flow
- Citizen applies via mobile app or kiosk
- DID and Verifiable Credential (VC) issued by government or trusted
verifier
- Credential stored in user wallet (self-custodied or managed)
- Employer or service provider requests and verifies credentials
- Citizen gains access to service (employment, health, subsidy)

### 2. Architecture
```
[Citizen Onboards] → [VC Issuer]
     ↓                   ↓
 [Wallet (DID+VC)]  ← [Veridian / anoncreds]
     ↓
 [Verifier Requests Proof]
     ↓
 [Proof Presented + Validated]
```

### 3. Tools & Libraries

| Layer   | Tool                 | Description
|
|---------|---------------------|------------------------------------------
---------------|
| DID     | Veridian / anoncreds| DID issuance or anoncreds spec
|
| VC      | Veridian VC Format  | JSON-LD-based schemas supported by
Cardano tools         |
| Wallet  | Lace                | Store credentials with optional backend
key control      |
| Verifier| Veridian Verifier   | Confirm authenticity of VC + revocation
status           |
| Schema  | JSON Schema / DID   | Standardize claims (e.g., ID verified,
employment)       |

### 4. Dev Tips
- For mobile onboarding, abstract wallet interaction via session signing
~~~~

~~~~
- Use low-data DID method to reduce on-chain overhead
- Start with employment credentials, expand to healthcare or subsidies

### 5. Output Expectations
- GitHub repo with DID issuance and VC storage integration
- Sample identity VC (e.g., "ID Verified", "Employed")
- Working prototype of VC request & proof presentation
- Test credentials issued via Veridian or anoncreds
- Demo: user onboarding → VC issuance → proof verification by employer

---

## 💸 Vertical 3: Financial Inclusion & Payments

### 🧠 Use Case
Build a blockchain-based microloan platform for underserved small
businesses.

### 1. Functional Flow
- MSME registers and completes VC-based identity checks
- Credit score assigned manually or via oracle
- Smart contract deployed to manage loan
- Funds disbursed in stablecoin (or ADA)
- Repayment tracked on-chain; default conditions enforced

### 2. Architecture
```
[MSME Registration]
     ↓
 [VC Issued (Credibility, Income)]
     ↓
 [Loan Request via dApp]
     ↓
 [Smart Contract Deployed]
     ↓
 [Funds Disbursed + Vault Created]
     ↓
 [Repayment Events Tracked]
```

### 3. Tools & Libraries

| Layer     | Tool                     | Description
~~~~

```
|
|-----------|-----------------------|--------------------------------------
--------------------|
| Contracts | Aiken                 | Develop loan lifecycle contracts
|
| Identity  | Community-issued VC   | Income level, business license, etc.
|
| Frontend  | Lucid SDK + React     | Borrower-facing interface
|
| Dashboard | GraphQL + Charting    | Lender/NGO visibility into repayments
|
| Oracles   | Webhook or API        | For credit scoring or exchange rate
feeds                     |

### 4. Dev Tips
- Use Lace wallet with testnet funds for MVP demo
- Store VC reference off-chain and hash on-chain for proof
- Customize contracts for NGO policies (e.g., gender-based incentives)

### 5. Output Expectations
- Aiken/Plutus smart contract repo for loan lifecycle
- Sample borrower VC
- Testnet demo of full loan flow
- Borrower UI mockup or MVP
- Dashboard showing repayment status

---

## 📦 Vertical 4: Supply Chain & Product Traceability

### 🧠 Use Case
Build a blockchain-based traceability system for agricultural exports
(e.g., coffee or cocoa), ensuring product origin, sustainability claims,
and fair trade certifications are verifiable.

### 1. Functional Flow
- Farmer or coop receives VC attesting certification (e.g., fair trade)
- Product batch minted as NFT with origin metadata
- Supply chain transfers logged on-chain (processor → exporter → retailer)
- Buyers verify provenance via QR scan

### 2. Architecture
```
```
```

~~~
[Farmer Registration]
     ↓
[VC Issued: Organic / Fair Trade]
     ↓
[Batch NFT Minted (Coffee, Cocoa)]
     ↓
[Transfers Signed by Each Handler]
     ↓
[Retailer Display + QR for Proof of Origin]
```

### 3. Tools & Libraries

| Layer     | Tool                   | Description
|
|-----------|-----------------------|--------------------------------------
----------------------|
| Identity  | Community-issued VC   | Certification (e.g., organic,
sustainability)              |
| Asset     | Cardano Native Tokens | NFTs for batches
|
| Contracts | Aiken / Lucid Script  | Logic for authenticated transfers
|
| Frontend  | Lucid SDK + QR Tool   | dApp for scan, display, and chain
history                   |
| Metadata  | CIP-68 / IPFS         | Store origin, timestamp,
certification hash                  |

### 4. Dev Tips
- Use NFTs as digital passports for batches
- Sign every transfer event for full chain-of-custody
- Link certifier-issued VCs to batch NFTs for verifiable claims
- Simulate scans using QR + transaction explorer

### 5. Output Expectations
- NFT minting script with traceability logic
- Sample VC for product certification
- Mock scan → explorer flow
- Chain-of-custody simulation with 3 transfers
- GitHub repo with asset metadata + transfer verification

---
~~~

```
## ✅ General Implementation Checklist

| Task                           | Tools / Outcome
|
|-------------------------------|------------------------------------------
------|
| Define actors & logic          | Use PoC design canvas
|
| Wallet strategy                | Lace, Eternl, backend wallet module
|
| VC tool selection              | Veridian / anoncreds / JSON-LD schemas
|
| Smart contract base            | Aiken, Marlowe, or from-scratch
|
| Token/NFT minting              | CIP-68 + Mesh
|
| Store off-chain data           | IPFS, Arweave
|
| Front-end build                | Lucid SDK, React, Mesh
|
| Documentation                  | GitHub with README + architecture.md
|
```