ScarCoin Deployment to Polygon Amoy

Smart Contract Deployment & Validation Report

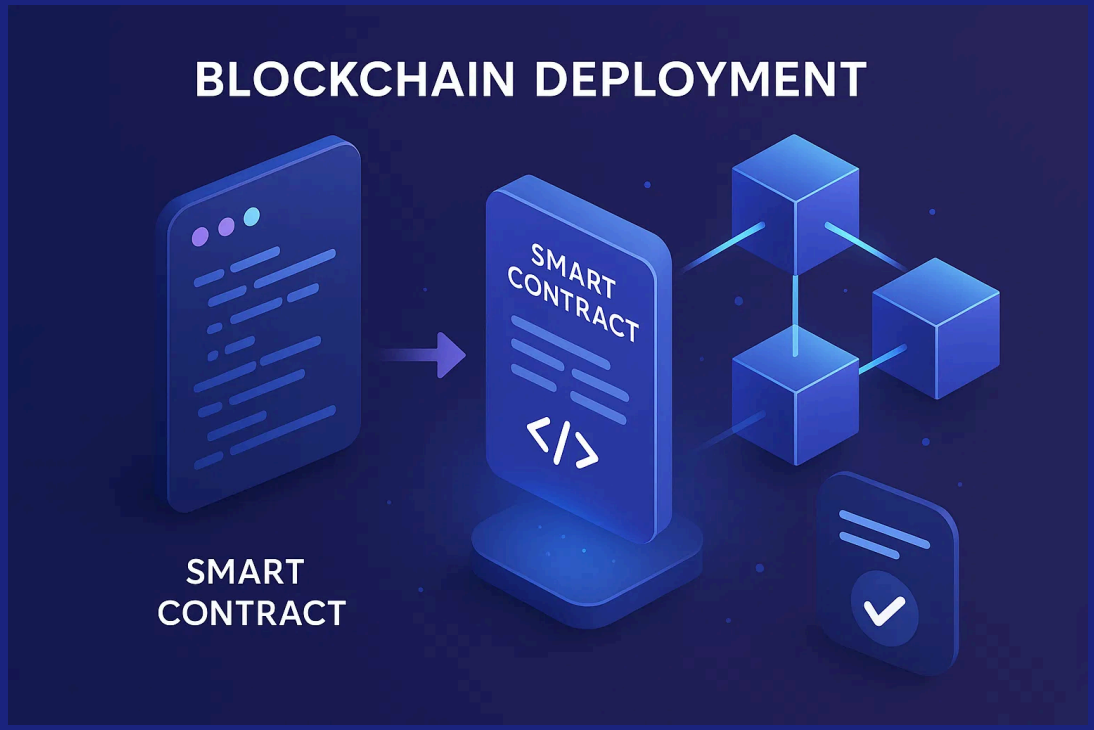
August 30, 2025

Project Overview

ScarCoin (SCAR) is an ERC20 token with a conditional minting mechanism based on an external oracle's value.

- Implements **acheExpiry** (24h limit) and **acheCooldown** (30m per address) protections
- Uses **ScarIndexOracle** to provide the index value that determines minting eligibility
- Minting only allowed when oracle index ≥ threshold (default: 0.5)
- Deployed to **Polygon Amoy** testnet (Chain ID: 80002)

This deployment serves as a validation of the contract functionality before potential mainnet launch.



Contract Architecture

ScarIndexOracle

Stores and provides the index value used by ScarCoin for minting decisions.

- **getIndex()** - Returns current index value (6 decimals)
- **updateIndex()** - Owner-only function to update index

ScarCoin (ERC20)

Main token contract with conditional minting based on oracle index.

- **mint(amount, acheTimestamp)** - Mints tokens if conditions met
- **burn(amount)** - Burns tokens from caller's balance
- **setOracle(address)** - Updates oracle address (owner only)
- **setMintThreshold(value)** - Updates threshold (owner only)

Deployment Process

1. **Repository Setup**

   Cloned repository and checked out `feat/scarcoin-initial-scaffold` branch containing contract code.

2. **Environment Configuration**

   Created `.env` file with deployment secrets: private key, RPC URL, owner address, and initial oracle index.

3. **Project Scaffolding**

   Updated `package.json` with deployment scripts and configured `hardhat.config.ts` for Polygon Amoy.

4. **Contract Compilation**

   Compiled contracts with `npm run compile` to generate artifacts and ABIs.

5. **Oracle Deployment**

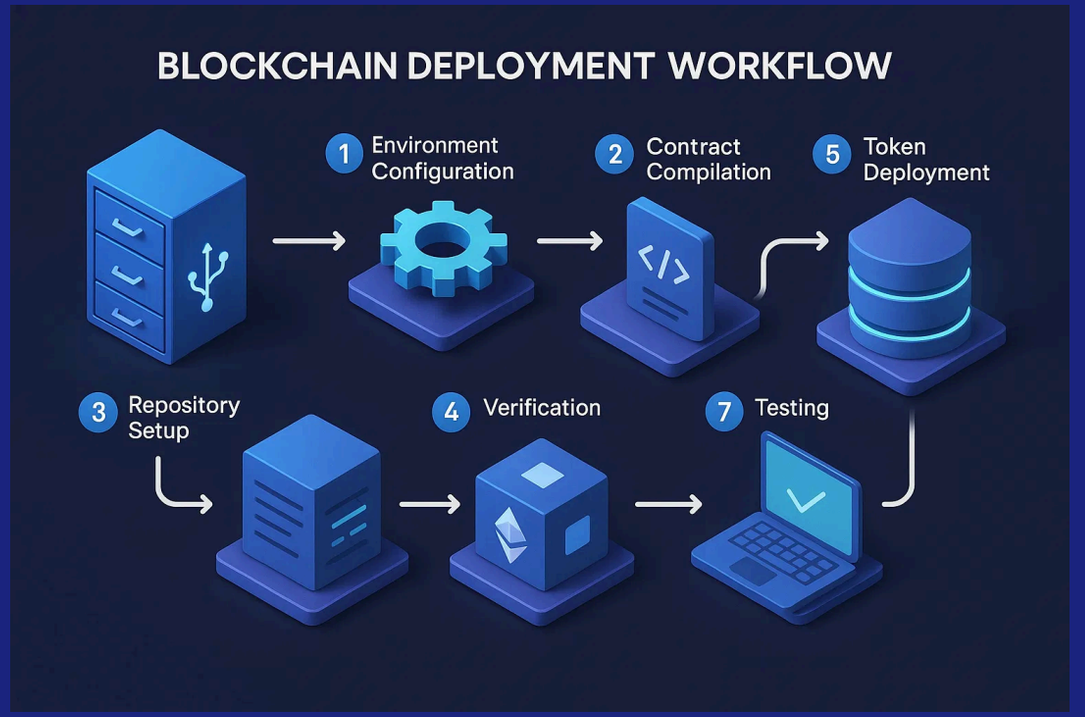
   Deployed ScarIndexOracle with `npm run deploy:amoy:oracle` using initial index value of 600000 (0.6).

6. **ScarCoin Deployment**

   Deployed ScarCoin with `npm run deploy:amoy:scar` using the deployed oracle address.

7. **Verification & Testing**

   Exported ABIs and ran health checks to validate contract functionality.



Deployed Contracts

ScarIndexOracle

Address: 0x9e7aa8fa9Da10972e9b6F8B694E0dd45EF18b6f8

Transaction: 0xbba3f6cf2f930bcc3dd6899fa7a7bb0da782520f067d0650d47c075481223882

View on Explorer

ScarCoin (SCAR)

Address: 0xF220789583BfBFB091A4d097529a3832CCbFBF0a

Transaction: 0x3c8f97a46e91dc00d9199a6b225dffb660dbb17b08c25da480e088ca6718f48a

View on Explorer

Deployed on Polygon Amoy Testnet (Chain ID: 80002)

Health Checks & Validation

Post-deployment health checks were performed to validate the functionality of the deployed contracts.

Oracle getIndex()

Returned **600000** as expected (0.600000)

ScarCoin mintThreshold()

Returned **500000** as expected (0.500000)

High Index Mint

Successfully minted 1 SCAR token with index > threshold

Cooldown Mechanism

Correctly rejected second immediate mint attempt

Low Index Rejection

Correctly rejected mint when index set to 400000 (below threshold)

All tests passed successfully, confirming that the contracts are functioning as expected on Polygon Amoy testnet.

Next Steps & Recommendations

Security Audit

Conduct a comprehensive security audit before mainnet deployment. Focus on the oracle update mechanism and minting conditions.

Mainnet Deployment

Deploy to Polygon mainnet using the same orchestration process validated on Amoy. Consider a phased rollout with limited initial minting.

Additional Features

Consider implementing governance mechanisms, multi-signature ownership, and enhanced oracle data sources for index calculation.

User Interface Integration

Develop a frontend application for users to interact with ScarCoin, monitor oracle index, and track minting eligibility.