SafePick

Tommy, Cami, Michelle, Akshith

Background

Context & Motivation:

- 8M tons of plastic waste enter oceans annually (UNESCO)
- Current deposit systems = centralized + inefficient
- Companies fear losses from unreturned containers

Related work:

- RecycleToCoin: Blockchain rewards for recycling (First Recycling Initiative Blockchain)
- Germany Shopping Carts: Simple, refundable deposits drive behavior

Overview

How the current system works:

1. Customer receives container with a QR code handled by the system (linked to your reusepass).
2. Once the customer is done, deposits the container in a bin.
3. Bins are cleared everyday, QR codes scanned manually by an administrator and updated in the system **(Issue!).**
4. If bins aren't received within a specific timeframe, system charges your card with a set amount **(Issue!).**

Overview

Traditional deposit-return systems face tracking and trust challenges.

Problems:

1. Person/administrator that scans the QR code is untrusted.
2. Customers can always remove card before it gets charged/challenge transaction.

Overview

Solution: A blockchain-based system for tracking reusable containers using QR codes and smart contracts.

Blockchain Relevance: Provides transparent, zero trust design that keeps record of containers usage while automating deposits and refunds without a centralized party. All is managed by immutable smart contract code rather than administrators.

Objectives

What our project needs to achieve:

Decentralized, Trustless System: Enable secure, transparent tracking of reusable containers without relying on centralized authorities.

Automated Deposit Collection & Refunds: Implement a system that handles deposits and refunds via smart contracts.

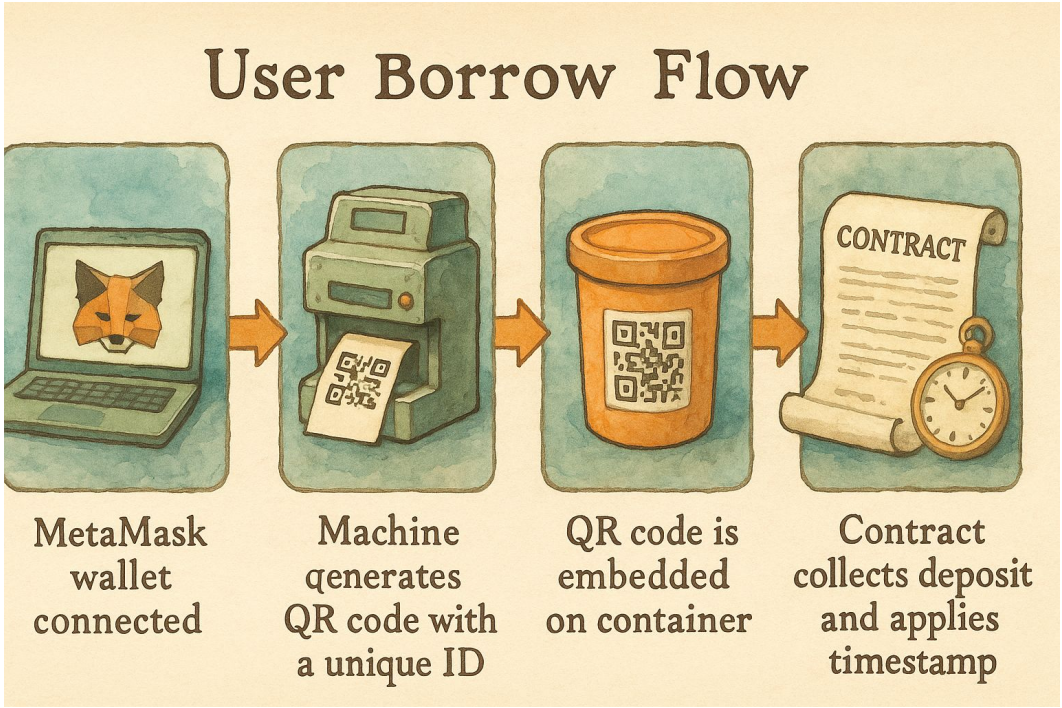
User-Friendly Borrowing/Returning: Make it simple for users to borrow and return containers (with QR codes).

Reduce Owner Risk: Ensure owners do not incur losses from user failures.

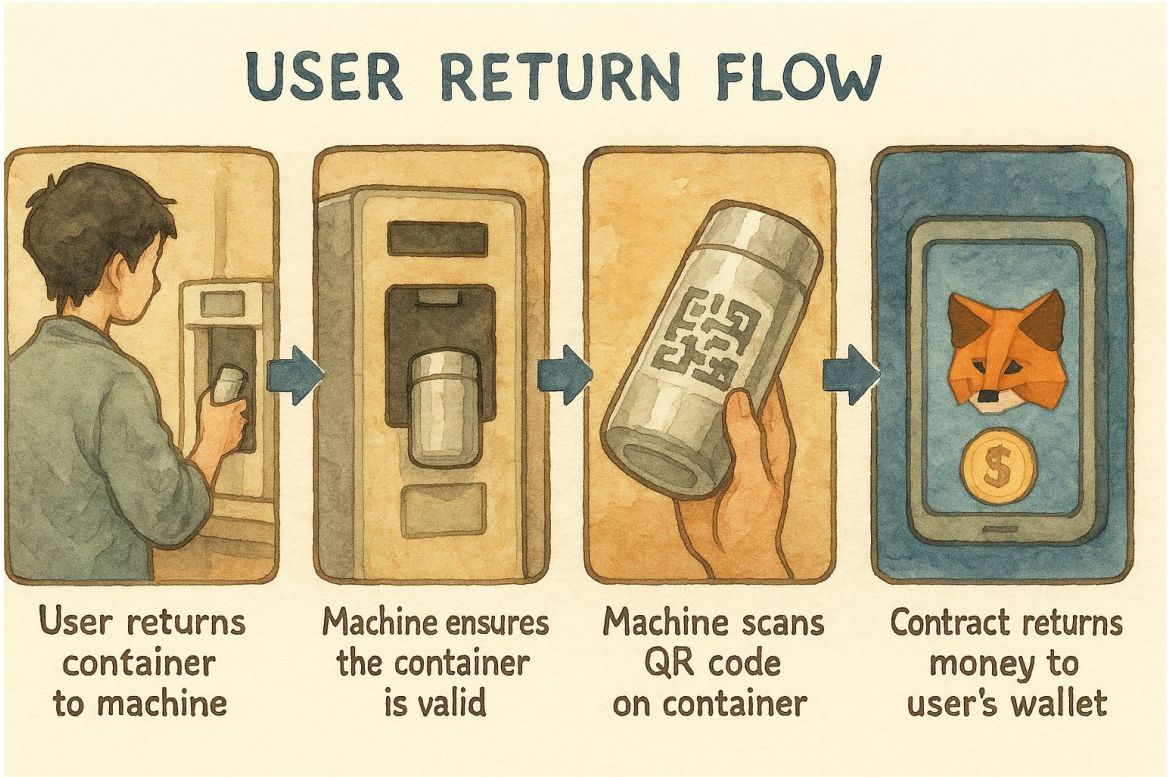
Assumptions

- The Physical Machine (like a vending machine) ensures it is a valid container shape and size.
- Container Storage Design/QR generation
- Multi-chain implementation

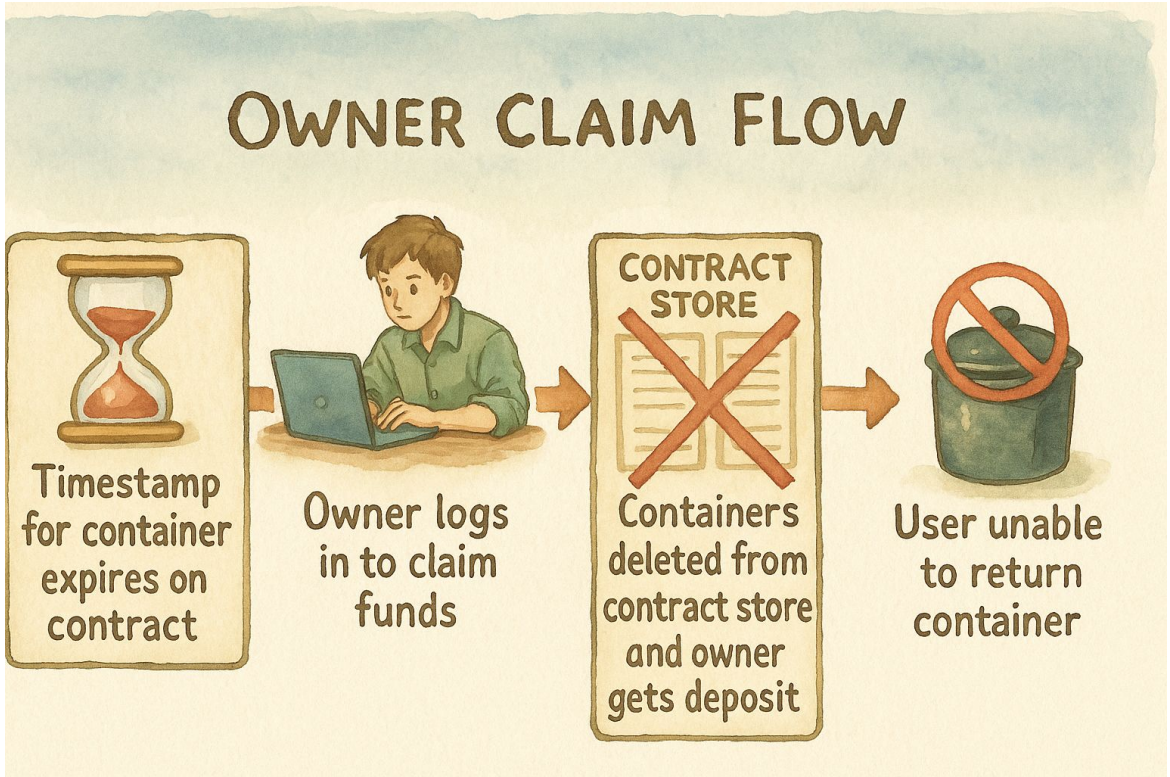
Architecture



Architecture



Architecture



Implementation - Part 1

Smart Contract Core Functions:

borrowContainer(): Records container as borrowed, collects deposit
returnContainer(): Validates return, refunds deposit
forfeitExpiredDeposits(): Owner-only function to claim deposits from containers that have exceeded the return expiration date.
isBorrowed(containerID): Checks container status

Key Design Decisions:

Container ID: Identified from a string ID (from QR) rather than a token
Deposit: Simple fixed deposit amount (0.001 ETH)
Automatic Refund: Automatic refund processing on return
Off-Chain Tracking: Events emitted for off-chain tracking

Implementation - Part 2

Technologies Used:

- Tailwind CSS for rapid styling
- ethers.js for blockchain interaction
- html5-qrcode for QR scanning
- Third party QR code generator website
- Remix for deployment

Key Features:

- Wallet connection via MetaMask
- QR scanning for borrow/return actions
- Adaptive UI showing transaction status

Evaluation

MetaMask Integration (Decentralized)

Integrates with MetaMask wallets enabling secure, trustless interaction with the Ethereum blockchain making our system decentralized.

Seamless checkout/deposit containers (Automated Deposit/Refund)

Implemented contract that allows users to deposit money in exchange for reusable container, and automatic refund when container is returned.

QR Code -> Smart Contract (User-Friendly)

Each container contains a QR code that is scannable, allowing the front-end to use the ID to interact with the smart contract.

Ensure owners can recuperate losses (Reduce Owner Risk)

Owners can claim the deposit money from "stolen/un-returned" containers once the reclaim timer expires.

Demo

Choose to Reuse ⚡

1· Connect Wallet

Connect MetaMask

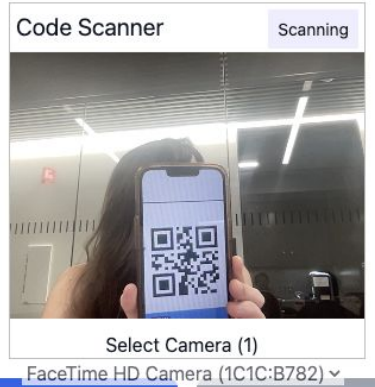
Connected: –

2· Generate New Container QR

Generate QR

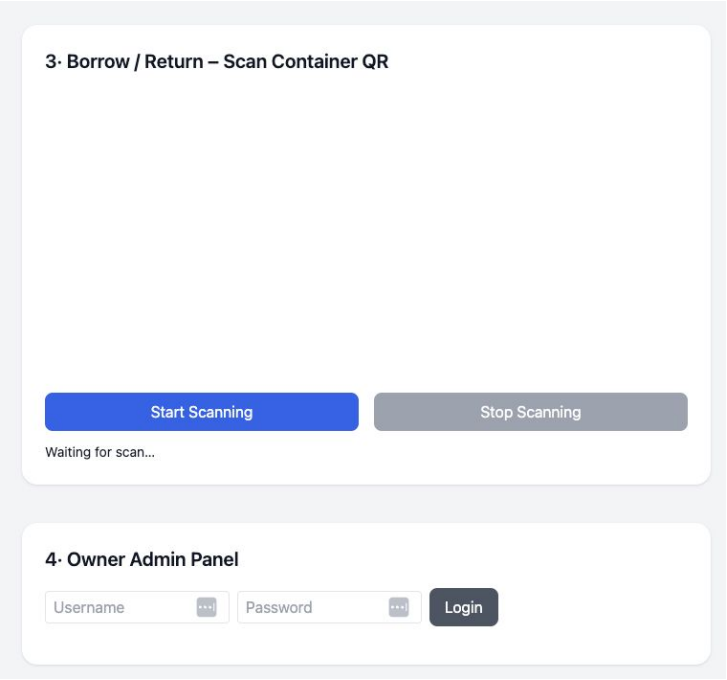
ID: –

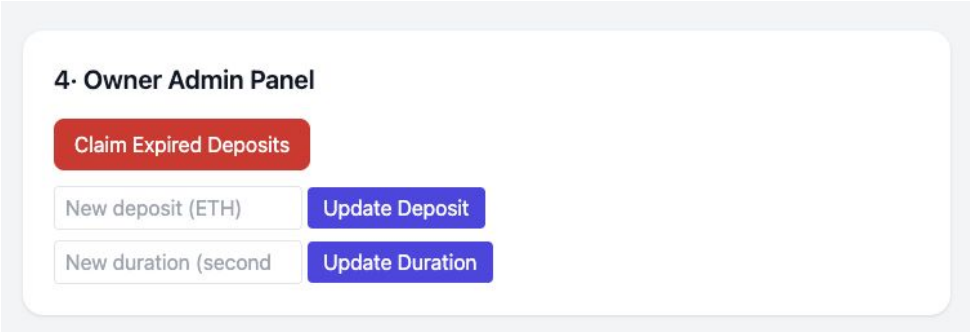
3· Borrow / Return – Scan Container QR

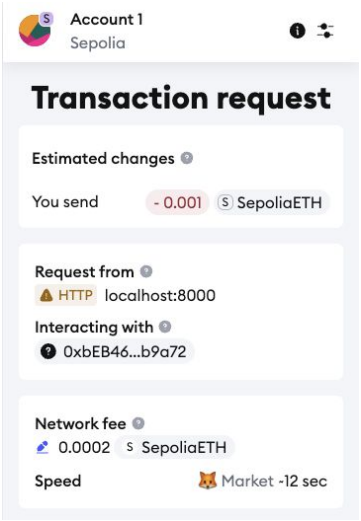


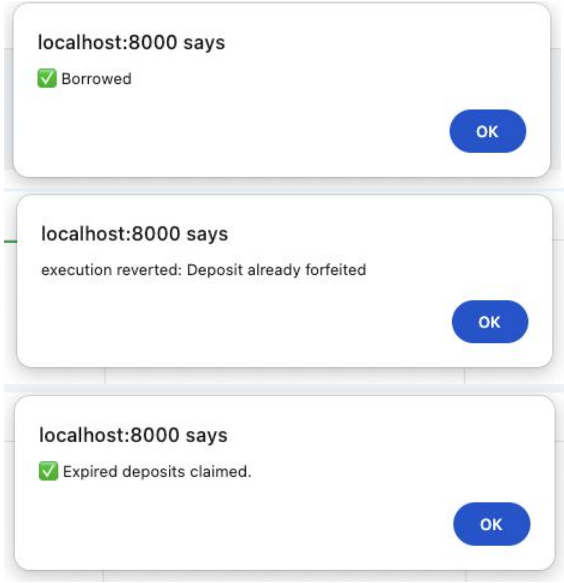
Start Scanning | Stop Scanning | Stop Scanning

Scanned ID: 7QPEP3Q5BMBC


3· Borrow / Return – Scan Container QR
Start Scanning
Stop Scanning
Waiting for scan...
4· Owner Admin Panel
Username
Password
Login


4· Owner Admin Panel
Claim Expired Deposits
New deposit (ETH)
Update Deposit
New duration (second
Update Duration


Account 1
Sepolia
Transaction request
Estimated changes
You send
- 0.001
SepoliaETH
Request from
HTTP
localhost:8000
Interacting with
0xbEB46...b9a72
Network fee
0.0002
SepoliaETH
Speed
Market -12 sec


localhost:8000 says
Borrowed
OK
localhost:8000 says
execution reverted: Deposit already forfeited
OK
localhost:8000 says
Expired deposits claimed.
OK

Challenges and Lessons

Challenges Faced:

- Handling MetaMask connection states and errors
- Ensuring proper deposit validation in the contract
- Managing QR scanner activation/deactivation
- Error handling for failed transactions

Insights:

- Blockchain provides trustless verification without centralized authority
- QR codes offer simple identification without specialized hardware

Next Steps

Extend or Improve:

- Implement progressive deposit amounts based on container value
- Add user statistics/history tracking
- Integrate with existing recycling programs

Roadmap toward Deployment:

- Mobile app development
- Partnership with cafes/restaurants
- Pilot program in university settings, connecting with BU's Recycling Initiative

Thank you!