ItemBlocks

The digital passport for your physical items

Fabian Zeiher
Iosif Koen

18/Dec/2023

Agenda

- What is ItemBlocks
- Motivation
- Tech Stack
- Implementation
- Testing
- Demo
- Reflection

What is ItemBlocks

- Connect physical items to the blockchain
- QR Code based identification
- Store and Access Information
- Track Ownership
- Track Changes

Motivation

- Data Transparency
- Data Access
- Data Integrity
- Verifiable Information

➡ Full access to potentially security relevant information.

Possible Use Cases

- Cross-company supply chains
- Food safety information
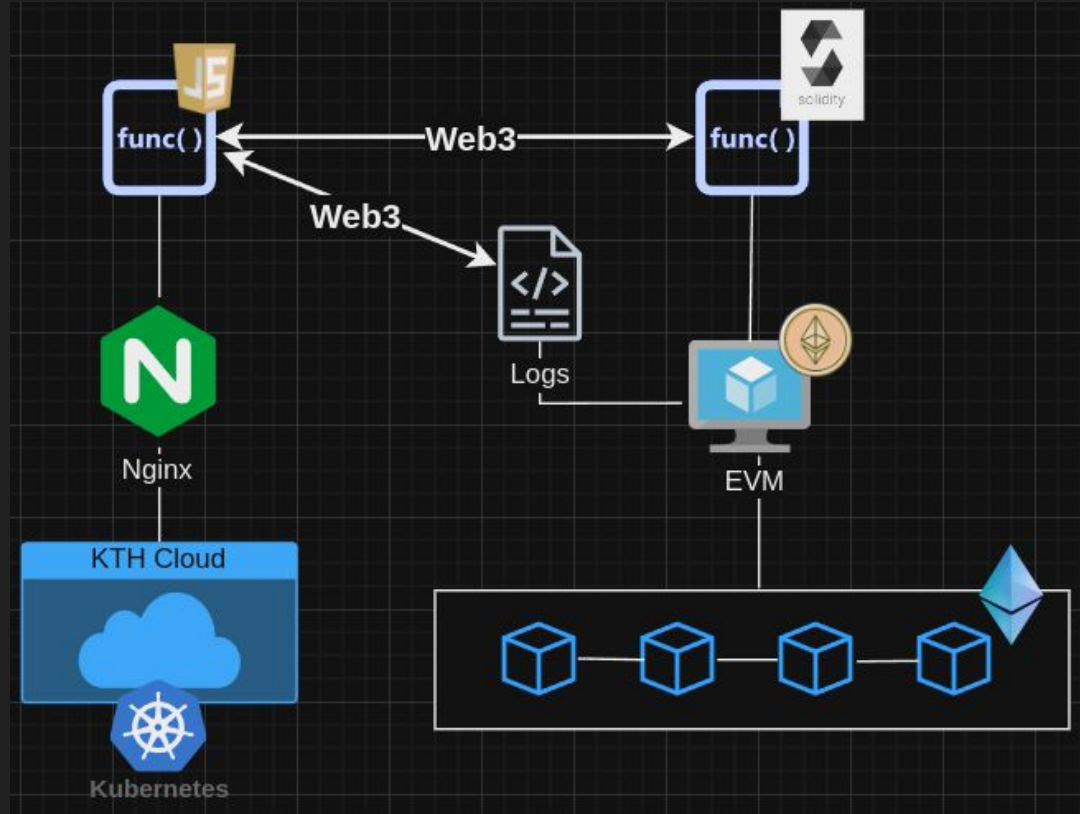- Labelling of Construction Materials
- Origin Tracing

Deployment

Smart Contract

● Manual Deployment w/ Remix

Web Application

● Continuous Deployment
  ○ GitHub Actions
  ○ KTH Cloud (Thank you to Pierre!)

Deployed Stack

On the shoulders of giants
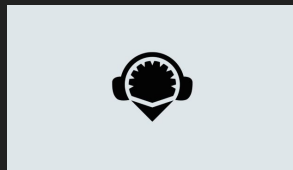
App

 UI / UX

 Framework

 Bundler

Contract

 Static Analysis

 Deploy

 Testing

Implementation

Extension of ERC721

TokenID ➡ Owner

TokenID ➡ Passport

TokenID ➡ [All owners]

Creator ➡ [Token IDs]

```solidity
import "openzeppelin-contracts/contracts/token/ERC721/ERC721.sol";
import "openzeppelin-contracts/contracts/access/Ownable.sol";

contract ItemBlocks is ERC721, Ownable {
    struct Passport {
        string name;
        string desc;
        string family;
        string url;
        string img;
    }

    mapping (uint256 ⇒ Passport) public itemPassports;
    mapping (uint256 ⇒ address[]) public allItemOwners;
    mapping (address ⇒ uint256[]) public createdItems;

    enum Role {Owner, Creator}
```

Blockchain Scanning

Utilising the power of ethereum logs we can track:

- Item Passport Creation (Who, When)
- Ownership Changes
- Data Changes
  - Who made the change?
  - What changed?
  - When did it change?
  - In which role was the change made?

Core Functions

createPassport()

updatePassport()

updateOwnership()

```solidity
function updatePassport(uint tokenId, string memory name, string memory desc, s
    require( isEligible(tokenId, msg.sender), "Must be the owner of the item or

    itemPassports[tokenId] = Passport ({
        name: name,
        desc: desc,
        family: family,
        url: url,
        img: img
    });

    emit PassportUpdate(
        tokenId,
        msg.sender,
        msg.sender == ownerOf(tokenId) ? Role.Owner: Role.Creator,
        name,
        desc,
        family,
        url,
        img);

    return (tokenId);

}
```

Unit Testing

Fully tested implementation.

```
❯ forge coverage
[⊞] Compiling...
[⊞] Compiling 37 files with 0.8.23
[⊞] Solc 0.8.23 finished in 3.49s
Compiler run successful!
Analysing contracts...
Running tests...
| File                      | % Lines         | % Statements    | % Branches      | % Funcs         |
|---------------------------|-----------------|-----------------|-----------------|-----------------|
| src/ItemBlocks.sol        | 100.00% (21/21) | 100.00% (22/22) | 100.00% (12/12) | 100.00% (9/9)   |
| test/ItemBlocksTest.t.sol | 100.00% (1/1)   | 100.00% (2/2)   | 100.00% (0/0)   | 100.00% (1/1)   |
| Total                     | 100.00% (22/22) | 100.00% (24/24) | 100.00% (12/12) | 100.00% (10/10) |
```

Static Analysis

No significant findings.

THIS CHECKLIST IS NOT COMPLETE. Use `--show-ignored-findings` to show all the results. Summary

- incorrect-exp (1 results) (High)
- divide-before-multiply (8 results) (Medium)
- shadowing-local (2 results) (Low)
- reentrancy-benign (2 results) (Low)
- reentrancy-events (1 results) (Low)
- assembly (3 results) (Informational)
- pragma (1 results) (Informational)
- dead-code (35 results) (Informational)
- solc-version (13 results) (Informational)

incorrect-exp

Impact: High Confidence: Medium

- ☐ ID-0 Math.mulDiv(uint256,uint256,uint256) has bitwise-xor operator ^ instead of the exponentiation operator **:
    - inverse = (3 * denominator) ^ 2

lib/openzeppelin-contracts/contracts/utils/math/Math.sol#L123-L202

Property Testing (Fuzzing)

Demo Time

Join the fun!

Future Work

- DAO to manage "Editors"
- Certifications:
  - A verifiable claim attesting
    to the correctness of information
- Native Mobile Application:
  - access to camera

Reflection

- Good coding experience
- We learned a lot
- Powerful tech stack (for Solidity)
- Blockchain has a lot of potential to be useful for the community

Take Home Message

Build a dApp for your community.

THANK YOU!