OST
Eastern Switzerland
University of Applied Sciences

Inventory Tracker

Blockchain – Testat

Dominic Walther, Marc Kissling

3 December 2023

Department of Computer Science

Agenda

- Concept
- Architecture
- Demo
- Design decisions
- Challenges
- What we learned
- Questions?

Concept

- **Approach:**
  Use a contract to track physical products throughout their lifetimes using Non-Fungible Tokens (NFTs) via the Ethereum blockchain.

- **Key Features:**
  - Minting new NFTs representing physical products.
  - Tracking the history of each product.
  - Transferring ownership of the NFTs.

Architecture

- Components
- Functionalities

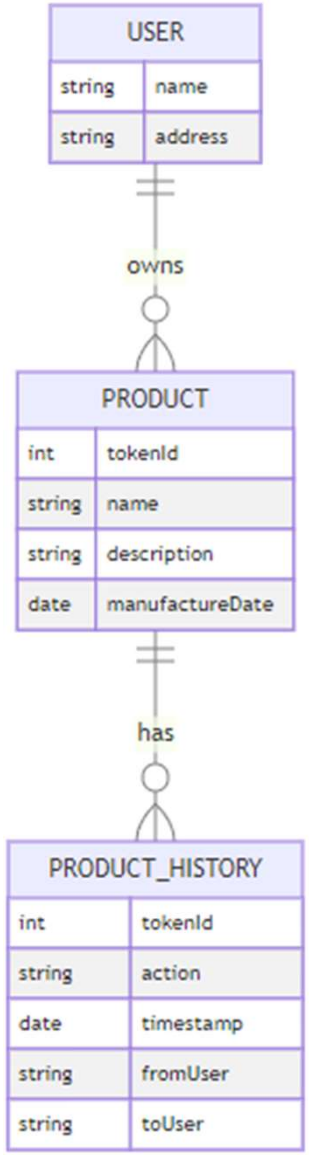
Architecture

Component: Solidity Contract

- **Structs:**
  - Product: Name, description, and manufacture date.
  - ProductHistory: History of actions taken on each product.

- **Key Functions:**
  - mintProduct: Create a new NFT for a product.
  - getProduct: Get product details by token ID.
  - transferFrom: Transfer ownership of an NFT.
  - getProductHistory: Get the history of a product.

- **Events:**
  - ProductEvent: Emitted when minting or transferring a product.



OST

Component: ABI File

- **Key Points:**
  - Describes the interface of the Solidity contract (errors, events, available functions…)
  - Essential for frontend integration with the contract.

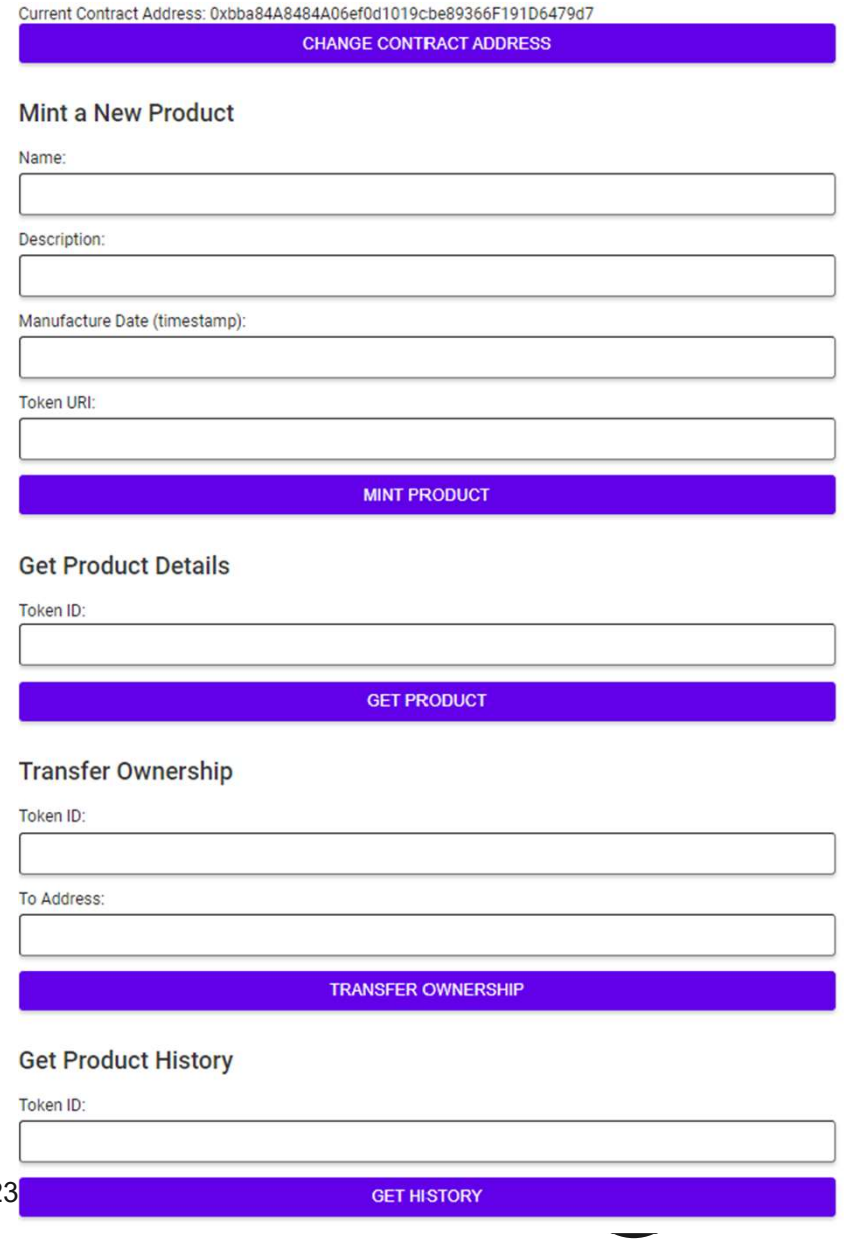
Architecture

Component: HTML

- **Key Functions:**
  - Load Web3.js dependency and main script
  - Provide forms and buttons
  - Apply CSS

Component: CSS

- **Key Features:**
  - Responsive design for various screen sizes.
  - Styling HTML components

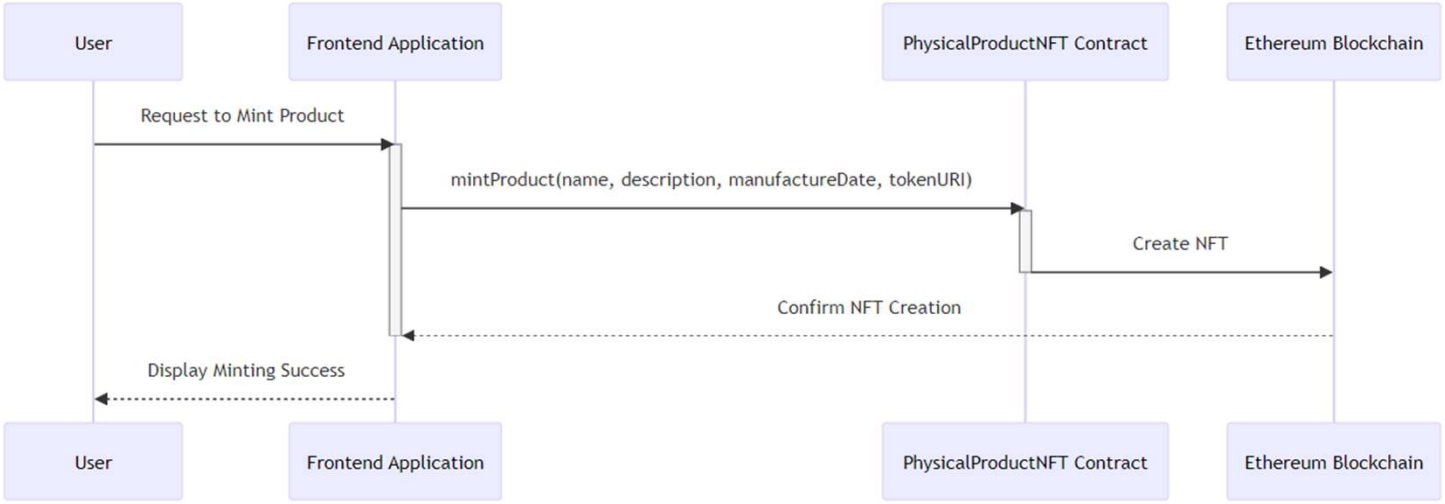


Component: JavaScript

- **Key Functions:**
  - Initializing Web3 for blockchain interaction.
  - Connecting to MetaMask.
  - Use ABI to interact with contract
    - Minting
    - NFT transfer
    - Get product details
    - Get product history

Functionality: Product-Creation

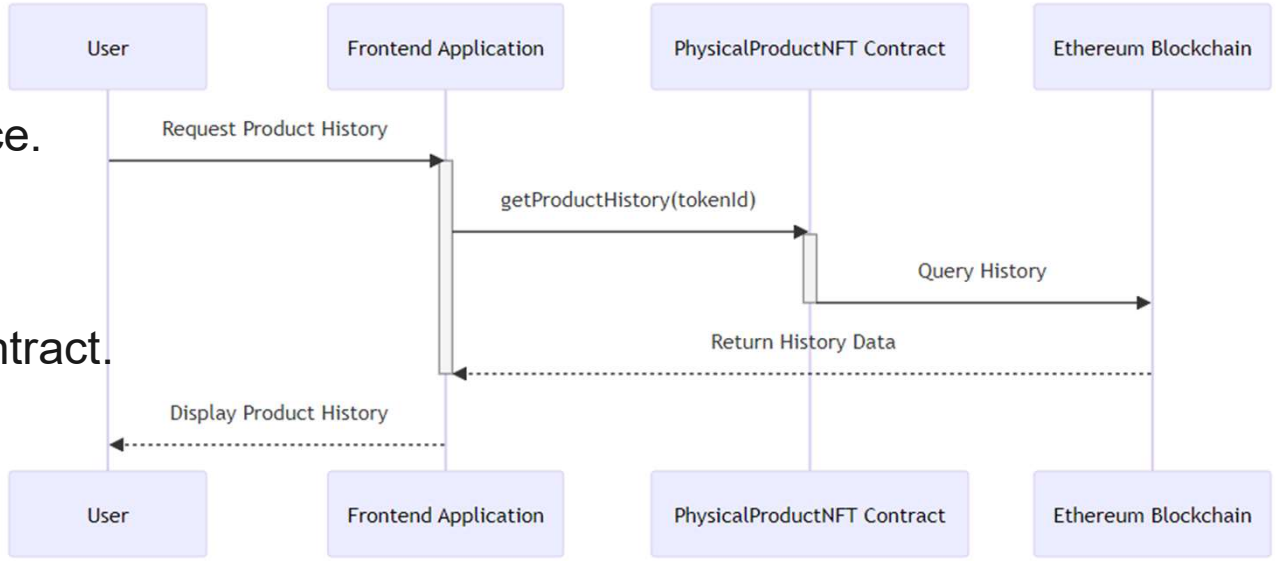
- ***Input Product Details:***
  User provides product details such as name, description, manufacture date, and vendor-specific URI.
- ***Contract Call:***
  Upon the user's request, the frontend calls the "mintProduct" function of the contract with the provided details.



- ***NFT Creation:***
  The contract mints a new NFT with a unique ID representing the product on the Ethereum blockchain.
- ***Confirmation and Logging:***
  The blockchain confirms the minting transaction, and the event is logged in the contract.
- ***Success Notification:***
  The frontend notifies the user of the successful minting and displays the new token ID.

Functionality: Product-History

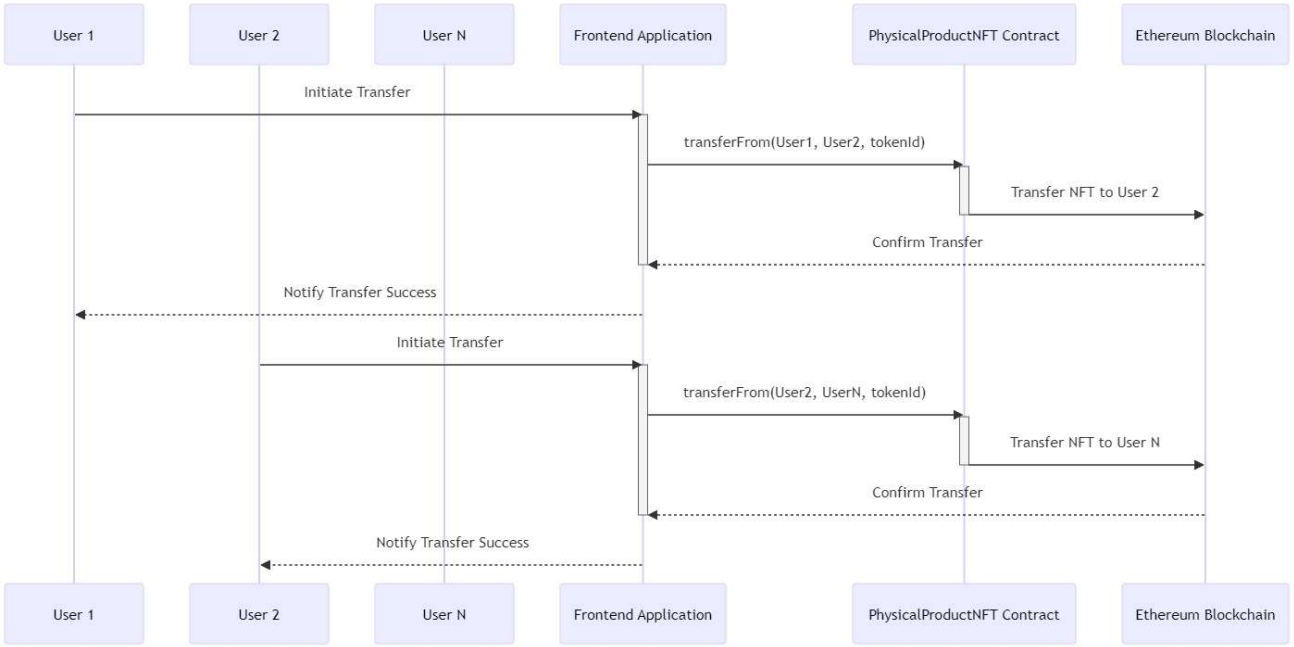
- ***Request History:***
  User queries the history of a product NFT via the frontend interface.
- ***Fetch Data:***
  Frontend application calls the getProductHistory function of the contract.
- ***Contract Processing:***
  The contract retrieves the history data from storage via the blockchain.



- ***Display History:***
  The frontend displays all past transactions related to the product's NFT.

Functionality: Product-Transfer

- ***Initiate Transfer:***
  User requests the transfer of a product NFT through the frontend application.
- ***Contract Interaction:***
  The frontend calls the transferFrom function in the PhysicalProductNFT contract.
- ***Ownership Change:***
  The contract executes the transfer, changing the ownership of the NFT on the blockchain.



- ***Notification:***
  The frontend application notifies the old owner upon successful transfer.
- ***History Update:***
  The product's transfer history is updated to reflect the new ownership.

OST
Eastern Switzerland
University of Applied Sciences

Demo

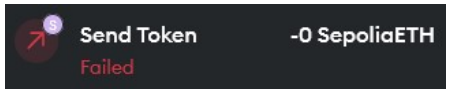
Dominic Walther, Marc Kissling

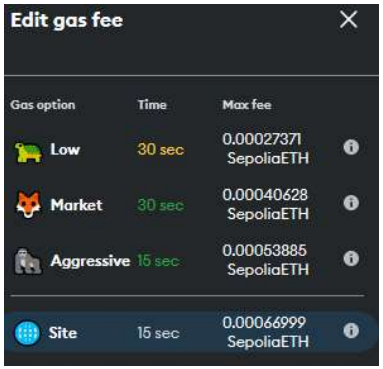
Design Decisions

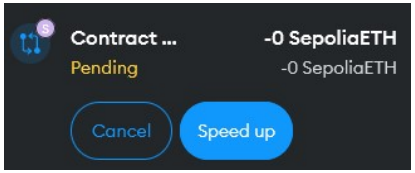
- Why NFTs?
- GitHub pages
- Docker

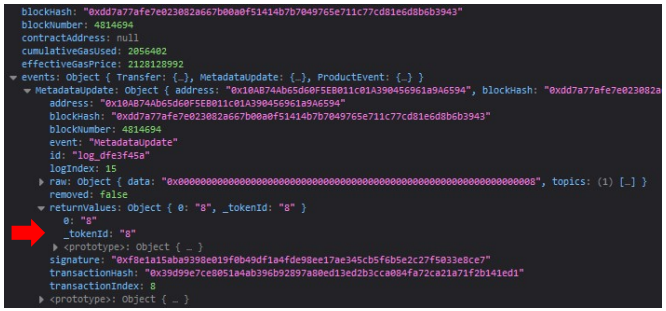
Challenges

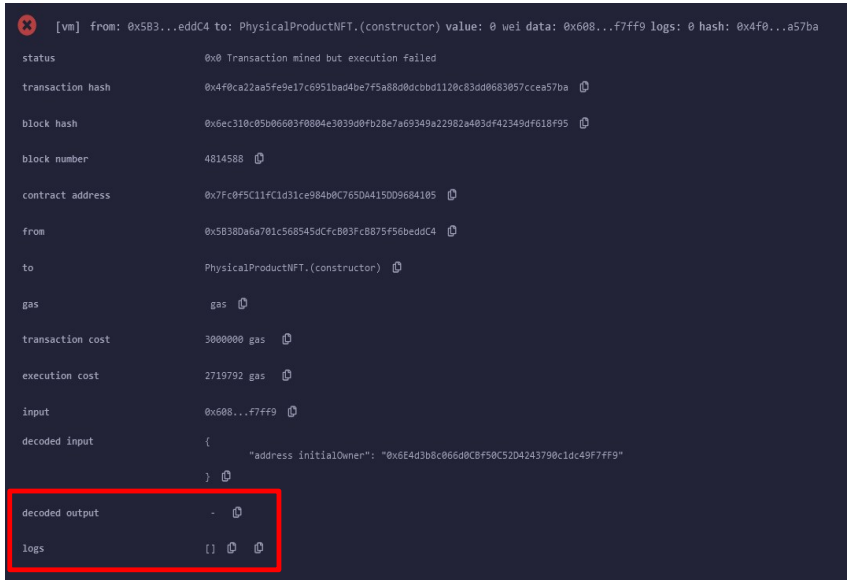
- Debugging
- Speed
- “Speed up”
- NFT ID

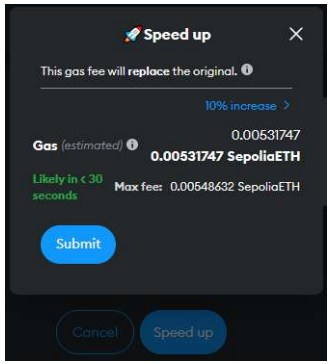












OST

What we learned

- NFT Advantages:
  - No infrastructure
  - No up-front cost
  - No maintenance fee

- NFT Disadvantages:
  - Developer experience
  - High costs (>0.10 CHF/transaction)
  - Cost variability (>50% over 24 hours)
  - Rapidly evolving = outdated information

Questions?