

Challenge Task - NFT Creator

Samuel Lemmenmeier, Rolf Oberhäsli, Tim Wisotzki

OST Blockchain

Overview

- Idea
- Architecture
- Components
- Design decisions
- Demo
- Question and Discussion

Idea

Create NFT's based on Text.

Architecture

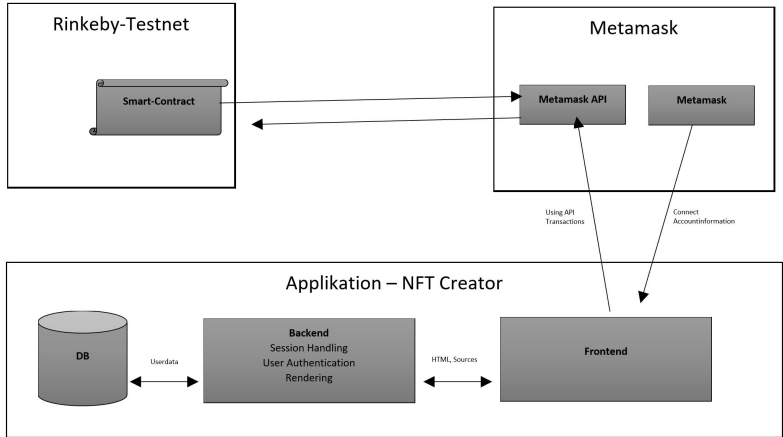


Figure 1: Alt-Text

Components - Overview

- Frontend
- Backend
- DB
- Blockchain
- Contract

Components - Frontend

- Javascript, CSS, HTML
- Userinteraction
- Eventlistening
- View is rendered HTML

Components - Backend

- Nodejs, Express
- Sessionhandling via Cookie
- User Authentication: JWT-Token
- routing
- returns rendered HTML

Components - DB

- NeDB
- Stores Userinformation
 - username
 - Pwd Hash (SHA-256)

Components - Blockchain

- Ethereum testnet Rinkeby
- Metamask as the middleman between application and blockchain
- wallet

Components - Contract

- Solidity SmartContract
- OpenZeppelin
- NFT_Citation is Ownable, ERC165, IERC721, IERC721Metadata, IERC721Enumerable

Design decisions

- simple Design & Architecture, using known technology
- NFT Frontend
 - everything regarding NFT is handled in the frontend
 - one simple script and no extra library, only functionality of Metamask
 - JSON RPC API
 - Contract Application Binary Interface (ABI)
- NFT Contract
 - ERC-721 NFT
 - stores citation directly on Rinkeby (max 32 Bytes)
 - id in 0 - 9999
 - each citation is unique

Functions on the contract

- `mintCitation(string),`
`safeTransferFrom(address,address,uint256)`
- `ownerOf(tokenId)` -> owner address
- `tokenInfo(tokenId)` -> citation
- `addressInfo(address)` -> [(tokenId, citation string)]
- many more functions

Demo

- Mint an NFT on our Contract
- Transfer NFT from one account to another

Questions

Any questions?