

Group 13 Scope Statement

Scope

1. What type of platform (Windows, Macintosh, UNIX, etc.) must the software work with? The software works on all platforms.
 2. Will the software function as a standalone application on a given computer, or will it function over a network connection? HTTPS
 3. What other software, if any, must the software interact with? For example, you might be building a subsystem component that will be integrated into a larger system. In such a case, it's important that you don't duplicate functionality provided by existing subsystems.
- Frontend:
 - Languages
 - Javascript/HTML/CSS
 - Browser
 - Modern browser that allows a crypto wallet
 - Option 1. Install the [Metamask](#) wallet extension.
 - Option 2. Install the privacy-focused [Brave browser](#) which bakes in [IPFS](#) and their [crypto wallet](#)
 - I recommend using the privacy-focused Brave browser founded Brendan Eich, creator of JavaScript and former CEO of Mozilla Corporation.
 - If our professor is not secure these options, we can supply a dummy account. Therefore, skipping these requirements.
 - Storage
 - Frontend application is stored on [IPFS](#)
 - Backend:
 - Languages
 - [Node.js](#)
 - [Solidity](#)
 - Contract
 - The contract is developed with solidity.
 - Storage
 - The contract is stored on the ethereum blockchain.
4. What programming language(s) will be used for the project?
- Javascript
 - HTML
 - CSS

- Solidity

5. Will the software use a graphical interface or a command line interface? Web UI/UX

Synopsis

In short, the project is a NFT [decentralized application \(Dapp\)](#). For an in-depth explanation of a NFT, visit ethereum.org/nft.

The application is split into a frontend and backend.

Frontend	Backend
UI/UX	Contract
Single page web app (SPA)	Ethereum NFT Contract
JavaScript/Typescript/HTML/CSS	Soldity/JavaScript/Node.js
Source code compiled to a bundle	Source code compiled to bytecode
Bundle stored on IPFS	Contract bytecode stored on Ethereum
Gets data from client	Creates a transaction and mints a NFT
Reads user input of a file and wallet address	Reads blockchain/memory
Writes output of the transaction	Writes blockchain/memory

The application has three main stages:

1. The frontend takes user input of a file and thier crypto wallet address.
2. The frontend connects to the backend, which is the ethereum blockchain, and then mints a NFT.
3. On success, the output is a successful transaction. The client now has a NFT.

Once the client has a NFT, they now have digital ownership of their file.

Notes

- UI connects to the NFT contract's address on the ethereum blockchain.
- UI reads input of a **file** and **crypto wallet address**.
- Contract is compiled to bytecode and deployed to the ethereum blockchain.
- On a successful contract deployment, a transaction of the contract address is returned.
- The contract address is how you call the contract's functionality.

Resources

Blockchain

- [Blockchain Wiki](#) - Blockchain wiki.
- [Ethereum Wiki](#) - Ethereum wiki.
- [Blockchain Explorer](#) - Ethereum blockchain explorer to view all transactions on the etereum blockchain.
- [AWS: What is ethereum](#) - Article on ehtereum.

- [Investopedia: Blockchain explained](#) - Article on blockchains.

Crypto

- [Crypto Wallets in Depth](#) - Article on crypto wallets in depth.
- [Crypto Wallets in Action](#) - Article on crypto wallets in action.
- [Public and Private Keys](#) - Article on public and private keys.
- [Asymmetric Cryptography \(Public Key\)](#) - Public-key cryptography wiki.
- [Symmetric Key](#) - Symmetric-key algorithm wiki.

Dev Resources

- [Ethereum Developer Resources](#) - Ethereum dev resources.
- [Docs](#) - Ethereum docs.

Foundation

- [Ethereum Foundation](#) - Ethereum foundation dev documentation.
- [Intro to Ethereum and Blockchain Basics](#) - Ethereum blockchain basics dev documentation.
- [Intro to Ether](#) - Ethereum ether dev documentation.
- [Introduction to Dapps](#) - Ethereum Dapps dev documentation.
- [Web2 vs Web3](#) - Ethereum Web2 vs Web3 dev documentation.
- [Accounts](#) - Ethereum accounts dev documentation.
- [Transactions](#) - Ethereum transactions dev documentation.
- [Blocks](#) - Ethereum blocks dev documentation.
- [Gas](#) - Ethereum gas dev documentation.
- [Nodes and Clients](#) - Ethereum nodes and clients dev documentation.
- [Networks](#) - Ethereum networks dev documentation.
- [Consensus Mechanism](#) - Ethereum consensus mechanism dev documentation.

Stack

- [Ethereum Stack](#) - Ethereum stack dev documentation.
- [Intro to the stack](#) - Ethereum stack intro dev documentation.
- [Smart Contracts](#) - Ethereum smart contracts dev documentation.
- [Intro to Smart Contracts](#) - Ethereum intro to smart contracts dev documentation.
- [Smart Contract Languages](#) - Ethereum smart contract languages dev documentation.
- [Smart Contract Anatomy](#) - Ethereum smart contract anatomy dev documentation.
- [Compiling Smart Contracts](#) - Ethereum compiling smart contracts dev documentation.
- [Deploying Smart Contracts](#) - Ethereum deploying smart contracts dev documentation.
- [Dev Networks](#) - Ethereum dev networks dev documentation.
- [IDE](#) - Ethereum IDE dev documentation.

Advanced

- [Ethereum Advanced](#) - Ethereum advanced dev documentation.
- [Standards](#) - Ethereum standard dev documentation.

- [Oracles](#) - Ethereum oracles dev documentation.

Local Dev Environment

- [Dev Environment](#) - Setup a local solidity dev environment.

NFT

- [NFT](#) - Ethereum NFT.
- [NFT Wiki](#) - NFT wiki.
- [Crypto Punks: The Solution to Digital Art](#) - Article on CryptoPunks.
- [In-Game Asset Trading on the Blockchain](#) - Article on game NFTs.
- [OpenSea](#) - OpenSea NFT marketplace.
- [Foundation](#) - Foundation NFT marketplace.
- [An Evolution in Gaming: Blockchain Gaming & The Metaverse](#) - Article on gaming and metaverses.
- [Sandbox](#) - Sandbox NFT metaverse.

Solidity

- [Solidity](#)
- [Github](#)
- [Blog](#)
- [Docs](#)
- [Cheat Sheet](#) - Solidity cheatsheet.
- [Intro](#) - Solidity intro.
- [Contract](#) - Solidity contract.
- [Structure](#) - Solidity structure.
- [Examples](#) - Solidity examples.
- [Common Patterns](#) - Solidity common patterns.
- [bkrem/awesome-solidity](#) - A curated list of awesome Solidity resources, libraries, tools and more.
- [Best practices](#) - Dev handbook for best practices.

Code

- [Tutorials](#) - Ethereum tutorials.
- [Learn by Coding](#) - Ethereum learn by coding.
- [androlo/solidity-workshop](#) - Comprehensive series of tutorials covering contract-oriented programming and advanced language concepts.
- [CryptoZombies](#) - Interactive code school that teaches you to write smart contracts through building your own crypto-collectibles game.
- [cryptodevhub.io](#) - Community-driven effort to unite like-minded people interested in Blockchain- and Crypto Technologies.
- [Discover Ethereum & Solidity \(ludu.co\)](#) - Complete course that takes you through the process of creating a decentralized Twitter clone using best practices.
- [ExtropyIO/defi-bot](#) - Tutorial for building DeFi arbitrage bots.
- [Syntax cheat sheet](#) - Quick syntax overview.

- [Solidity and Vyper cheat sheet](#) - Review syntax of both languages side-by-side.
- [willitscale/learning-solidity](#) - Complete guide on getting started, creating your own crypto, ICOs and deployment.
- [LearnXInY](#) - Learn Solidity in 15 mins (for experienced devs).
- [Crypto Zombies: Learn to Code Dapps](#) - Crypto zombies dapp.
- [Step by Step guide to creating your own NFT — CryptoPunks or Pudgy Penguins](#) - Code a simple random image NFT.
- [Code A Minimalistic NFT Smart Contract in Solidity On Ethereum: A How-To Guide](#) - Code a minimalistic NFT.
- [Mint an NFT with IPFS](#) - Mint a NFT with IPFS doc.
- [github.com/lavalabs/cryptopunks](#) - Github CryptoPunks are 10,000 unique collectible characters with proof of ownership stored on the Ethereum blockchain.
- [gitlab.com/droptbaskets/random-image-generation-nft](#) - Randomly generate images from traits and create NFTs
- [tomhirst/solidity-nextjs-starter](#) - Github Fullstack dapp starter with nextjs.
- [Solidity Examples](#) - Solidity by example.

Storage

- [What is IPFS?](#) - IPFS doc.
- [IPFS in Action](#) - Curated ipfs list.
- [Uncensorable Wikipedia on IPFS](#) - IPFS blog on thiery uncensorable wikipedia project.
- [ipfs/distributed-wikipedia-mirror](#) - Distributed Wikipedia Mirror Project.
- [ipfs/roadmap](#) - IPFS roadmap doc.

Academia

- [IPFS Academic Papers](#) - IPFS academic papers.
- [Ethereum White Paper](#) - Ethereum white paper.
- [Bitcoin White Paper](#) - Bitcoin white paper.

Definitions

- Dapp: A decentralized application that uses a decentralized network (e.g., Ethereum, Bitcoin, ...).
- Blockchain: An immutable, ledger database of transactions distributed across a network.
- Crypto wallet: A tool that manages a collection of accounts, public keys, private keys, and interacts with a blockchain network.
- Crypto Keys
 - Asymmetric cryptography involves pairs of keys.
 - Generation of keys depends on the cryptographic algorithm.
 - The algorithm generates a private and public key.
 - Depending on your use case, one key signs and encrypts messages, the other key verifies and decrypts messages.
 - For example,
 - Private key: Sign and encrypt messages.
 - Public key: Derived from the private key that verifies and decrypts messages.

- Public address: A hash of the account's public key. An Ethereum public address is digested from a keccak256 hash function where the public key is the input.
- Ethereum: A blockchain platform that facilitates cryptocurrency and smart contracts.
- Ether: The cryptocurrency on the the ethereum blockchain.
- Gas: The fee required to conduct a transaction on the blockchain.
- Smart contract: Self-executed code on a blockchain.
- NFT (Non-fungible token): A smart contract that has a standard, transfers ownership, and mints unique tokens of a digital asset. According to wikipedia.org/wiki/Non-fungible_token, "NFTs are tracked on blockchains to provide the owner with a proof of ownership that is separate from copyright."

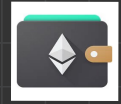
User



Browser



File

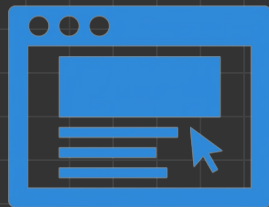


Crypto
wallet

IPFS
storage



UI



Smart Contract

