S2BCacademy

# Blockchain & Solidity

Whitepaper

## Project:

Team: Group 3

**Members:**

Fabian Rittmeier
Simon Hirner
Vitaliia Savchyn

S2BCconsult
S2BCacademy
S2BCcoach

MORPHEUS LABS SEED

## Project Name: Bazinga!

A Blockchain project to help the creators of immaterial assets, such as memes or other artworks, to sell their copyrights as well as gain some money and offer a wide variety of immaterial assets to potential buyers.

### What is the problem?

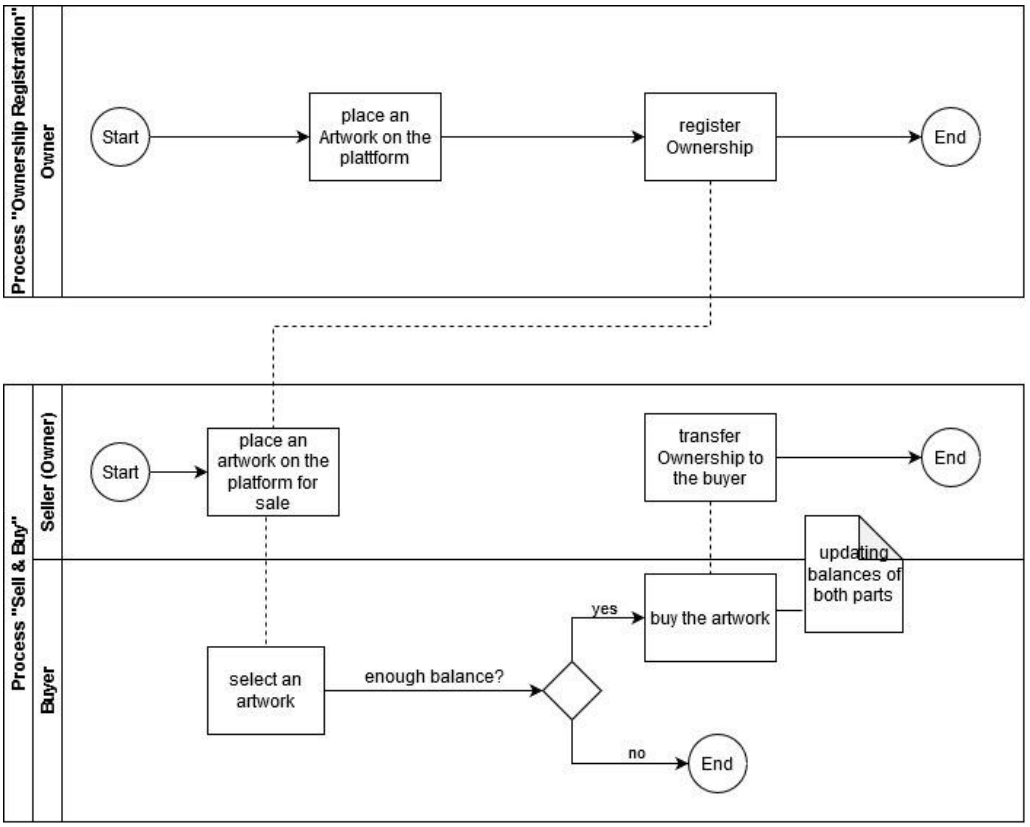
In the Internet you can find various pieces of art which copyrights are not protected in a proper way. Our project is a good start to solve this problem. At the same time there are already many people who follow the copyright laws and our application will help them to make it easier and faster.

### What is the solution?

Our application allows users to register, buy and sell the copyrights of an artwork on a marketplace. The copyrights are represented as a smart contract with which the users can interact.

### How will it work?

A seller places his artwork on the platform and registers his ownership. The platform is charging a registration fee for every registration thus generating profit. A buyer selects an artwork and if his account balance is enough for a purchase, he is allowed to continue the process. The balance of the buyer is being reduced by the artwork price and the balance of the seller is being increased. The seller has to transfer the copyrights to the buyer. If the balance of the buyer does not allow the purchase that is the end point.




## Elements of the App

The app has a BackEnd consisting of contracts written in Solidity and a React FrontEnd mostly written in JavaScript.

**BackEnd:**

The BackEnd is made up of four different smart contracts. The first being the Marketplace which is representing the platform on which the users can register, sell and buy their immaterial assets. Secondly the Artwork is representing the immaterial asset which can be bought and sold. The contract called Ownable is a module which provides a basic access control mechanism. It is inherited by the Marketplace and the Artwork so that some functions can only be called by the owner of the respective contract. Lastly the MarketplaceCreator is just a factory for deploying the Marketplace.

**Smart Contract: Marketplace (inherits Ownable)**

**Variables:**

- **artworks (address array):** Contains addresses for all registered artworks.
- **registrationFee (constant uint):** Sets value of the fee which is charged for registering a new artwork

**Function:**

- **registerArtwork (payable):** Registers a new artwork by charging the registration fee, sending the fee to the owner of the marketplace and adding the new artwork to the artworks array. The registration fee has to be sent with the function call.
- **getArtworks:** Returns the artwork array with all addresses of registered artworks in it.

**Smart Contract: Artwork (inherits Ownable)**

**Variables:**

- **artworkName (string):** Contains the name of the artwork
- **artworkUrl (string):** Contains a URL to the artwork
- **artworkHash (byte32):** Contains a unique hash of the artwork thus connecting the artwork unequivocally with this contract.
- **artworkPrice (uint):** Contains the purchase price of the artwork.
- **isArtworkForSale (bool):** Contains if the artwork is currently for sale or not.

**Function:**

- **sellArtwork (modifier onlyOwner):** Offers the current artwork for sale by setting the passed price and changing the isArtworkForSale variable.
- **cancelSellArtwork (modifier onlyOwner):** Cancels the offer by changing the isArtworkForSale variable.
- **buyArtwork (payable):** Buys the artwork by sending the price to the owner and subsequently transferring the ownership. The price of the artwork has to be sent with the function call. Furthermore it is checked if the artwork is currently for sale and if the exact price was passed.

**Smart Contract: MarketplaceCreators**

**Variables:**

- **marketplace (address):** Contains the address of the deployed marketplace.

**Function:**

- **createMarketplace (private):** Creates a new marketplace and stores the address in the marketplace variable.

**Smart Contract: Ownable**

**Variables:**

- **owner (address):** Contains the address of the owner of the contract.

**Modifier:**

- **onlyOwner:** Checks if the caller is the owner of the contract.

**Function:**

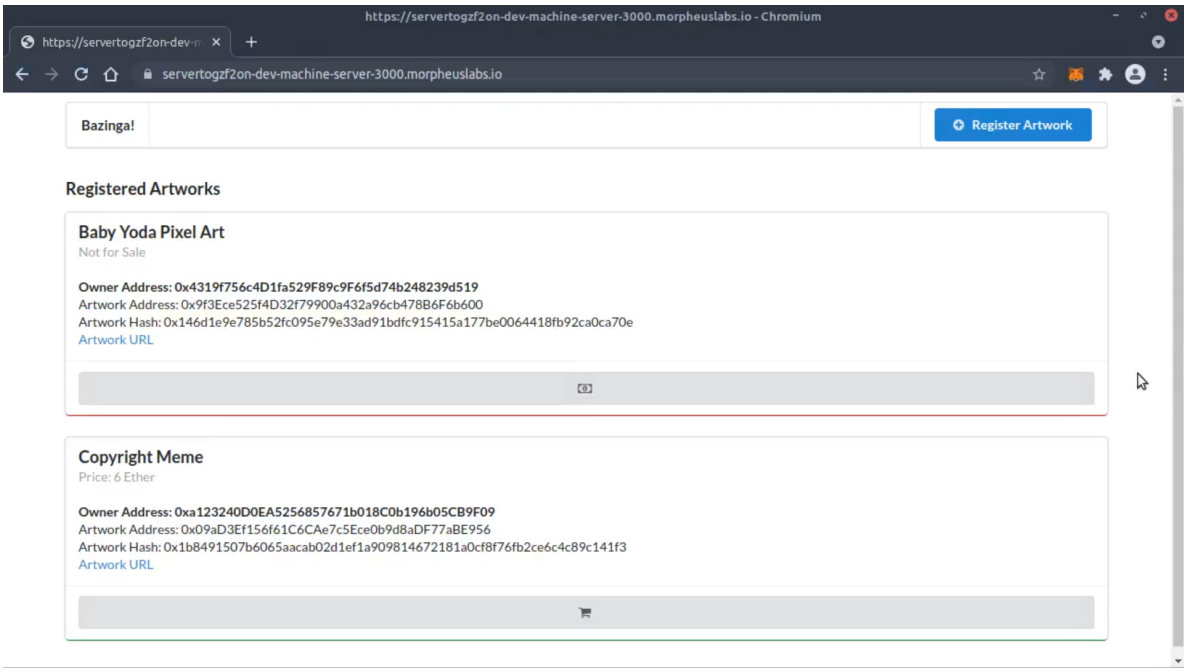
- **transferOwnership (onlyOwner modifier):** Transfers the ownership of the contract.
- **transferOwnershipInternal (internal):** Transfers the ownership of the contract.

**FrontEnd:**

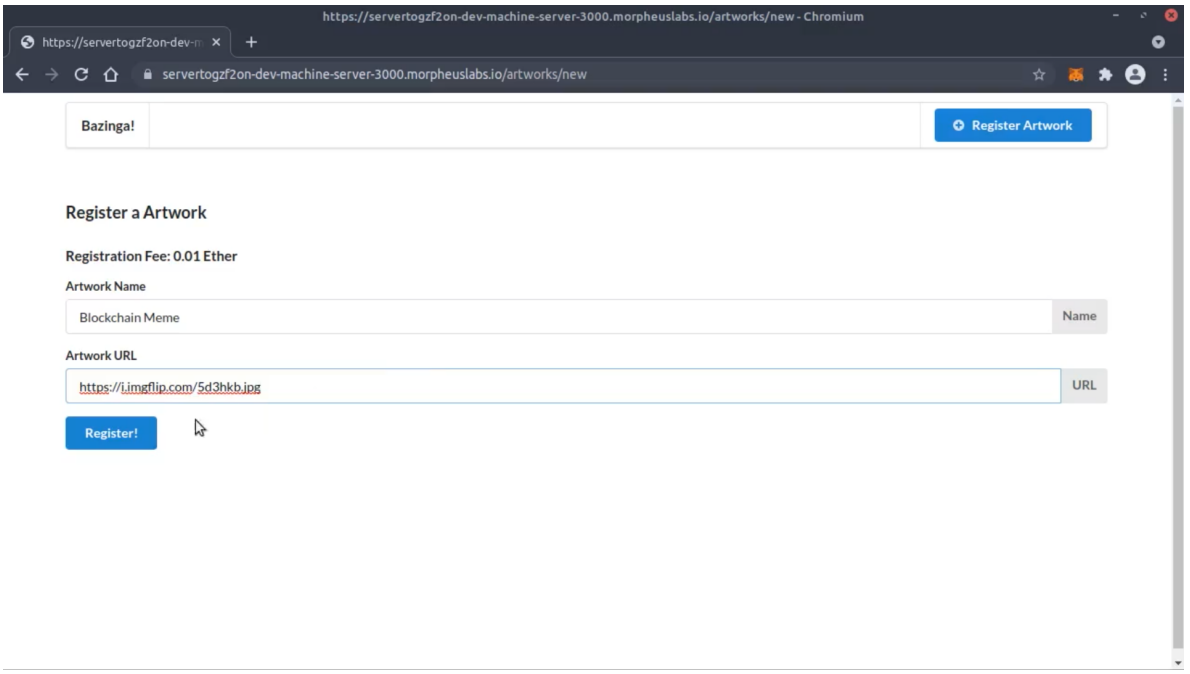
The FrontEnd consists of four different pages. The app opens MetaMask to confirm and send transactions to the smart contracts.

**Pages:**

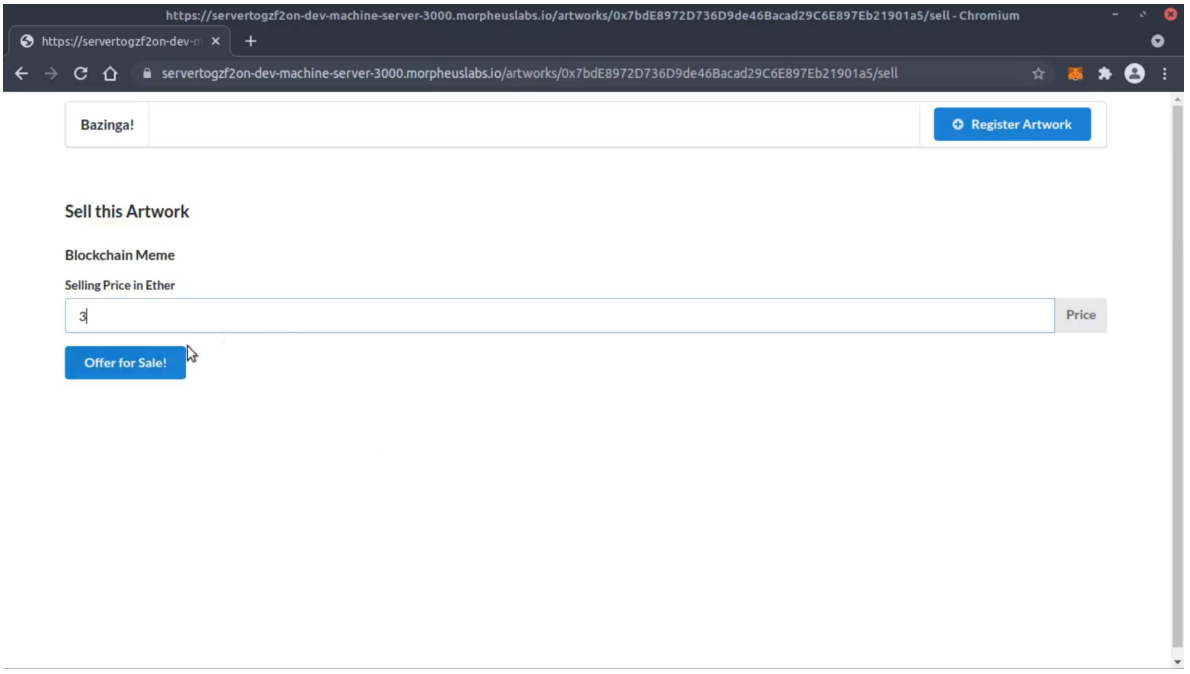
- **Index (/):** Shows a list of all registered artworks and contains buttons for registering a new artwork, selling your own artwork or buying a new artwork. This is the start page of the application.



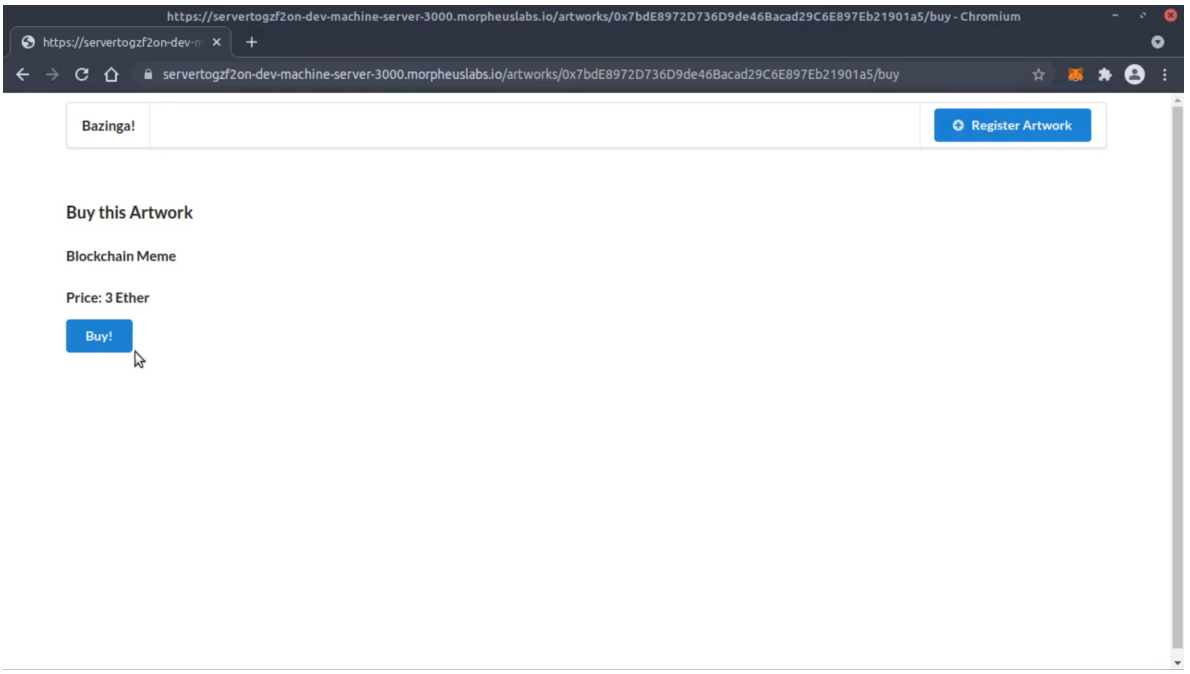

- **Register (/artworks/{address}/new):** Shows an input form for registering a new artwork and a button for confirmation.




- **Sell (/artworks/{address}/sell):** Shows an input form for the selling price and a button for confirmation.



- **Buy (/artworks/{address}/buy):** Shows the purchase price and a button for confirmation.



## Opportunity & Value Proposition

<table>
<tr>
<td rowspan="2">Key partners<br>Ethereum,<br><br>Marketing:<br>- Instagram<br>- Google<br><br>MorpheusLabs SEED</td>
<td>Key Activities<br>Copyrights administration and transfer for immaterial assets with the help of Smart Contracts</td>
<td rowspan="2">Value proposition<br>The Problem of the immaterial assets is that they are being spread in the internet very fast and without a proper copyright procedure<br><br>On the platform is it possible to exchange immaterial assets according the law and not to the detriment of their authors</td>
<td>Customer Relationships<br>Networking<br><br>Support Platform - to help with solving the problems</td>
<td rowspan="2">Customer Segments<br>Authors of various immaterial assets = sellers, especially of memes and small humorous videos<br><br>People who want to use immaterial assets = buyers</td>
</tr>
<tr>
<td>Key Resources<br>Plattform Bazinga!<br><br>Team</td>
<td>Channels<br>Website<br><br>Mobile application<br><br>Internet ads</td>
</tr>
<tr>
<td colspan="2">Cost Structure<br>Development Environment<br><br>Deployment</td>
<td colspan="2">Revenue Streams<br>Revenue is planned to be achieved with a transaction fee for each concluded Smart Contract, the fee depends on the asset type and size</td>
</tr>
</table>

## The Team

Fabian Rittmeier, Business Informatics student - Developer

Simon Hirner, Business Informatics student - Developer

Vitaliia Savchyn, Business Informatics student - Business Analyst

## Annex

Have a look into our GitHub Repository including a video with show cases of the application:

https://github.com/HM2021-BC/BC-Group-3