Design History

Tyler Danno
CID: 01500381
td1118@ic.ac.uk

Christoph Renschler
CID: 01580976
car3418@ic.ac.uk

Kurt Martin-Brown
CID: xxxxxxxx
xxxxx@ic.ac.uk

asdfadfs asdkfadsf
CID: asdfkadfs
asdfasdf@ic.ac.uk

3rd Year Group Project
Minexx
Department of Electrical Engineering
Imperial College London
16-06-2021

1 Definition

One unique aspect of the group project with Minexx was the amount of freedom we were given, both with regards to the specific problem we want to address, as well as the solution we propose.
While this made the work more interesting, it also made it much more challenging. Before developing any product, we had to figure out what the product should be, and before that even, what problem it should address. As a consequence, we approximately spend the first third of the project with ideation and the second third with conceptualization before we started developing what we then knew to be the final product. An overview of our work throughout the project is shown in the Gantt chart in the project organisation section of our Github repository. Accordingly, this design history is split into two parts: A theoretical design part, which describes our ideation and conceptualization work in developing the project idea, as well as a technical design part, explaining the different decisions we made during the development phase.
Furthermore, working on topics such as web development and security tokens, both of which we knew little about beforehand, forced us to plan our goals with caution: Developing in an alien environment and programming language always takes more time then expected and can cause the team to run into more dead ends than usual.
With these two special requirements in mind, we decided to work according to Agile methodologies, organizing tasks in a Kanban boards throughout ideation, conceptualization and the development process. More specifically, we organized all notes, resources and communication using Notion.so [1].
You can find out our rationales behind the tools and methodologies we used in the methodology report.

2 Research

Given the open-ended project specifications, we used the first few meetings with Marcus Scaramanga, the CEO of Minexx, to fully understand the work Minexx is doing at the moment and the technology

it is using to do so. Our goal was to discuss potential project ideas with Marcus directly, figuring out both what is useful for Minexx and what is possible in the given time. Quickly, two topics stood out: On the one side, we learned about the benefit of blockchain technology in an opaque and unfair value chain. On the other side, artisanal miners, payed in cash, find it difficult to borrow from regular banks without a credit history. [2] Minexx solves this partially by tracking transactions through their MineSmart platform. However, there was no channel through which external companies, governments or investors could get involved in this funding.
Thus, the early project definition decided on with Marcus was to somehow use blockchain technology to allow external players to somehow invest in the artisanal mining market. However, it was unclear, how our project could combine the two.
The first three weeks were thus dominated by research and ideation. We mainly targeted our research at the following three topics:

1. Artisanal and small-scale mining in Africa
2. Current impact investment funds and how they operate
3. Blockchain and security tokens

We uploaded useful sources and papers about these topics in the Sources folder.

3 Ideation

As mentioned before, the first part of the project focused extensively on ideation and conceptualization. Developing agile was great for this as it allowed us to quickly test and alter our ideas. This avoided wasting time fully developing a feature that has little to no use for Minexx or potential investors. A more in-depth explanation of our project development can be found in the methodology report.
Initially, there were two general ideas of how we could use blockchain to involve external investors. The first idea was to provide them with a direct channel to purchase minerals in advance, effectively funding mine sights through prepayments. This could be achieved by offering utility tokens: virtual representations of future mineral resources.
The other idea was to give investors a way to participate in an impact fund that provides artisanal miners with loans, and channelling the interests generated by the loans back to the investor. This in turn could be achieved through a security token offering [3].
The main advantage of using security or utility tokens is that the ownership stake is preserved on the blockchain ledger. The technical rationale behind using either token over regular securities or prepayments is further discussed in the smart contract section of our repository.
To further illustrate the differences between our two initial ideas, the two token types compare as shown in table 1 [4].
Throughout the ideation phase of the project, we came up with multiple different ways in which we could use utility tokens or security tokens to provide artisanal miners with funding. These ideas generally moved from ”prepayment” channel to ”investment” channel for two main reasons.

1. Integration is hard.
2. There is a bigger market of external investors that are not interested in the physical resources.

The following four ideas are ordered accordingly, with decreasing integration to the Minesmart platform as well as increasing separation between resource purchase and investment:

1. Platform for external buyers to directly buy digital representation of minerals that will be mined in the future in the form of utility tokens, allowing the buyer to transparently trace their resource throughout the value chain.

   Full-on integration with the MineSmart platform required

2. Extending the above with a marketplace through which utility tokens can easily be traded. This could attract investors that aren't interested in the physical resource but rather buy utility tokens to sell them with profit.

   The marketplace would need to integrate with both the MineSmart platform and crypto exchanges. After more research, we realized this goes far beyond the scope of this group project.

3. Platform to allow investors to directly fund a certain mine buy buying security tokens associated with that mine, being payed via cash. This gives the investor control over where his money is going.

   This would require a sophisticated system in which the smart contract incorporates the risks and output associated with each mine.
   We decided against this as this leaves the buyer with much more risks. Success or failure of a specific mine sight depends on many factors that are unforeseeable for an external investor who knows little about the ASM market and only has surface level information about a certain mine sights available. Directly investing into specific mine sights would thus be more similar to gambling.

4. Using security tokens to allow investors to purchase a share of the ATIF, being payed out the average interest that the invested portion of the ATIF generates. The fund is invested by a funding committee instead of by each individual investor.

   The funding and resource purchasing processes are fully separated.

3.1 Meeting with Finboot Tech

Our final decision of which idea to proceed with was mainly motivated by a meeting with Alvaro Llobet, the general manager of the blockchain technology provider FinbootTech [5]. FinbootTech has worked with Minexx on their MineSmart platform before, integrating their MARCO platform [6] to track minerals throughout the value chain.
Having worked in the blockchain domain for years, Alvaro could give us a better real-world perspective on our project idea. He strongly advised against using a a single token to both track minerals and fund mines, mainly arguing that a single system would end up unnecessarily complex and inflexible.
As a consequent, our final design concept removed as much direct links to the MineSmart platform as possible, allowing us to develop independently with a potential future integration of both platforms in mind. In other words, while MineSmart and the MARCO platform use tokens to track minerals on the miner side, our goal was to develop a separate process on the investor side, selling security tokens as investment assets.

Table 1: Two ideas of yielding capitalization: Utility Token vs Security Token.

Utility Token	**Security Token**
represents a virtual voucher to be redeemed for a good or service in the future. → Generally attracts investors interested in the physical minerals, such as tech or battery companies.	represents a fraction of the ownership of some asset, e.g. a stock or share of real estate. → Generally attracts investors looking for a financial investment rather than physical resources.
(+) No specific legal requirements	(–) Classifies as a security, coming under legal requirements such as identification and asset freezing.
(–) Direct link between investment and resource (integration with the MineSmart platform)	(+) Mineral resource and capitalization can be separated completely.

4 Smart Contract

4.1 Blockchain Platform Decision

One of the first aspects that needed to be decided upon when beginning to do research into security tokens was the blockchain platform that would be used in order to code and deploy the smart contract on. As mentioned in the Blockchain and Security Tokens file on GitHub, there are many different characteristics that differ blockchains from one another. Some are designed to solve specific problems that various blockchain platforms display, and others are designed for more general use cases.

The most important difference, in our eyes, was the fact as to whether the blockchain is public or private - public meaning that all transaction history is available to see by anyone, and private where this information is hidden. Out of the blockchains available that we could select to develop a security token on, the majority of ones that do not use a public ledger required payment to the company managing the blockchain to use it. After much research, our team came to the decision between two blockchains - Ethereum and Polymesh. Ethereum, as explained in the Blockchain and Security Tokens file, is a public blockchain that is the most widely used for developing smart contracts. Polymesh is a private blockchain that is managed by a security token company called Polymath, who offer security token issuing as a service to institutional companies.

Ideally, we would have liked to be able to use Polymesh, due to a number of reasons, especially with Polymesh having been developed for the specific use case of security tokens, rather than the much more general use-case for Ethereum. However, when researching Polymath's fees, we quickly found that they were far higher than the budget we had available to us for this project. This resulted in us settling for Ethereum, but following is a list of reasons as to why we would have preferred to use Polymesh.

4.1.1 Hard Forks

In the Blockchain community, you have what is called a "hard fork". As blockchain platforms mature and develop, they tend to have periods where certain parts of the community are not happy with certain technical aspects of the platform (e.g. consensus mechanisms, settlement times, block sizes, etc). If a large enough portion of a certain blockchain community decide that they want to change certain aspects, but there is another large portion that decides they either prefer the way the blockchain currently runs, or think a different change would benefit it better, what can happen is a "hard fork".

Essentially what happens is the blockchain (and all coins/tokens on the blockchain) is duplicated, creating two new blockchains (or one new, one with the old name). These two blockchains share exactly the same history, which is the history of the original blockchain, but from that point forward they will have two separate ones, implementing their technical differences going forward). For normal crypto assets, this is not a big problem, however for security tokens, where each token is meant to be tied to a specific asset in the real world, this does cause one. Suddenly there are two tokens, both supposedly linked to the same asset. This begs the question - which token is the "real" security token?

Hard forks are only a trait of blockchains that are completely public, as their technology and future are managed by a decentralized community of developers. Private blockchains, that are managed by specific companies, do not display this same trait - as all potential changes to the blockchain are managed by the same party. Therefore, deploying the smart contract (and therefore the security token) on Ethereum runs the risk of these hard forks affecting how to value these tokens. Polymesh on the other hand will never go through this process, and therefore the value of each token is kept safe.

4.1.2 Finality of Transactions

Due to the way that Ethereum processes transactions and places them in blocks on the blockchain, there is a small probability that blocks containing transactions are reordered as new blocks are placed on top

of them. Depending on how you interact with the smart contract, and how much gas you are willing to pay for those specific transactions, there is even a probability that your transaction will not be processed, and fail to be added to the ledger. This probability decreases as more blocks are added, however these two facts do not provide enough assurance for finality of transactions for the finance industry.

Polymesh, being specifically designed to solve these types of problems, does not have this trait, and one can be certain that their transaction will be finalized when interacting with smart contracts deployed on it. For this reason, developing the security token on Polymesh would have been advantageous over Ethereum.

4.1.3 Confidentiality

As mentioned earlier, Ethereum's ledger is publicly distributed, meaning anyone is able to see all transactions on it. This means that investors' holdings and trades in the security token linked to the ATIF are shown to any interested party. This could be risky when investors are making large trades - there are definitely parties that would not want information as to exactly where their capital is being invested. Although all that would be shown on the ledger would be the investor's Ethereum wallet, this would definitely not be ideal for many, hence Polymath's Polymesh would be preferable. Polymesh is a private blockchain platform, hence they are the only party that is able to see the complete ledger history.

4.1.4 Identity

Finally, there are problems with Ethereum to do with connected individuals when it comes to the transaction of the security token (and hence the underlying financial security, as well). There are very strict regulations in place that ensure that individuals in certain sanctioned countries are not able to make investments in certain products or financial instruments. Although the whitelist described in the Token Technical Documentation helps to avoid these parties gaining ownership of any security tokens, there are problems to do with the consensus algorithms for Ethereum.

In the case that Ethereum blockchain miners, who provide the computational power required to produce new blocks, are paid transaction fees for processing a transaction that is involved with the transferral of a security token, the issuer may run the risk of this being a big red flag for regulators. There is no way to be certain of, nor control, who is receiving transaction fees for certain transactions before they have been processed on Ethereum. Polymesh, on the other hand, requires validator nodes (involved in the consensus algorithm for Polymesh) [7] to be approved by Polymath before they can participate in the network. This completely removes the risk that using Ethereum has.

4.2 Ethereum Design History/Challenges

4.2.1 Solidity Data Types

The main issue that we found almost immediately after beginning to look into Ethereum smart contracts and programming in Solidity was that Solidity does not natively support any types of decimal numbers, only integers [8]. This would obviously lead to trying to calculate any types of interest payments proving more difficult than in other programming languages, as the standard for expressing interest rates is down to 2 decimal places, or 1 "basis point" (0.01%). We wanted to be able to use this standard when setting the rate for investors.

There are a number of smart contract libraries available online that try to solve this problem by implementing more complicated mathematical functions, allowing you to express floating point numbers via a number of different data types [9] [10]. After importing and trying to use some of these smart contracts, we found that the complexity of the calculations we were trying to complete was too high, running up gas costs per transaction to far higher than would make sense for a retail investor to pay. This was

mainly due to the fact that these libraries available are all in the form of smart contracts themselves, meaning that interacting with them costs even more gas compared to just the cost of interacting with the contract for the security token itself.

In order to overcome this problem of lack of support for decimal numbers, we decided that the best way to represent the values in our smart contract, allowing us to complete calculations that would cause integer rounding errors otherwise, was just to scale all the numbers we were using upwards. We found out that it is relatively common practice to do this when programming in Solidity [11], where the values that your contract uses are scaled upwards so that calculations are performed correctly, but then any scaling back downwards is accomplished by the end point UI (which would not be programmed in Solidity, and therefore would not display the same rounding errors).

As we wanted to be able to set the interest rate to 2 decimal places, we realised that the minimum scaling factor we needed to multiply by would be 104. We arrived at this number by looking at what the number of decimal points after our percentage values would be when represented as the proportion of 1. For example, if we wanted to set the interest rate to 3.75%, we would convert this number to 0.0375, and then multiply it by the security token value in someone's account in order to calculate the interest owed to them. In order to remove the need for decimals, we multiply the 0.0375 by 104 to get 375. This means that when Minexx is entering the new monthly interest rate, the rate passed to the smart contract is the scaled up value. We used the same scaling for the number of tokens, in order to have as little rounding error as possible.

The demonstration video uses a function called displayTestBalances() which does actually scale these numbers back downwards, however this was just in order to show more easily in the video how the smart contract works. For this reason, simplistic token and interest rate values were used during the demonstration.

4.2.2 Interest Transferal Gas Costs

The next issue that arose when coding the smart contract was in relation to the transferal of tokens, and specifically the transferal of interest that has been accrued on certain tokens. With automatic interest transferals (between investors) when sending tokens to one another being one of the main benefits that using a smart contract would provide, it was important that this was implemented properly.

Originally, we had tried to implement a structure called "token", which would not only encapsulate which address currently owned it, but also the date on which the interest tied to that token would be claimable. When transferring tokens between two parties, the list of tokens tied to the sender's address would be iterated through, and the address to which the token is linked to would be changed to the recipient's address. In this way, the transfer of the token automatically transfers the interest accrued on that token, as all of this data is linked directly to the token. Similarly, the date on which each token's interest could be claimed was transferred along with the tokens.

What we found when testing this type of transfer, was that the gas costs associated with these transactions ended up being far higher than expected. The reason for this was due to the sheer number of token instances that were required to represent the size of the fund (in the scale of millions) was too high. The smart contract was required to keep track of a vast number of objects, and in the case of any larger transfers, it was sometimes required to modify the states of vast numbers of these objects within one function call. This obviously increased the complexity of these transactions, running up gas costs that would have been equivalent to $70-80. With the focus of the token to be to increase the ease of investing in the fund for retail investors, while keeping minimum entry costs low, these prices meant that it would absolutely not be worth a small retail investor's capital to buy the security token, as certain transfers would be using up large portions of that investor's account just to send the transaction to the blockchain network.

There was a potential to use private storage on each node in order to keep track of the data required for the transfers described above, however it was decided against, due to the big security risk posed by individuals being able to change past or current data on their computer far more easily than on a blockchain.

We decided instead to link an interest-claimable date to each account/address, rather than to each individual token. The only issue that arose from this, was that we now did not have separate dates that the tokens interest could be claimed upon. We wanted to ensure that no-one was able to claim interest before it was due. For example, in the case that an individual owns tokens that they are able to claim interest from on the next day, then they buy more tokens on a secondary market (and hence is transferred tokens and the interest accrued on those tokens), we did not want the individual to be able to claim the total interest (including the new interest accrued), on the earlier date.

To implement this in the smart contract, we decided that within the transfer() function, we would need to include code that would essentially force a recipient of tokens in a transfer to inherit the later interest-claimable date, out of theirs and the sender's. Although this might delay the time in which they are able to claim the initial interest, it enforced the rule that they are not able to claim any interest earlier than they should be able to.

4.2.3 Interest Payment Method

The project required us to think about how the investors would be paid their interest when it is due. We originally thought that the simplest way to do this would be to add the value of the interest payment directly into the balance of the account in the form of more tokens. The individual would then be able to sell the tokens when they wanted, which would essentially be redeeming them for cash. However, as we thought more about this, we realised that this solution would not work - mainly due to the fact that we wanted the total supply of tokens to be representative of the size of the ATIF. The reason this would be a problem is because in order to pay the investors interest in security tokens, this means that more tokens would need to be minted in order to pay the investors, signifying that the size of the fund is growing, when in reality it is not changing.

The one way in which this method would work, would be if Minexx could guarantee that all investors would choose to reinvest the interest they receive. This would mean that the Investment Committee would know in advance the exact amount of capital that the fund would increase by when receiving interest payments from their debtors, and scope out new projects to begin with that extra capital once those payments have been received. This is possible as they would know that none of their investors would be requesting interest payments to be redeemed for cash.

However, this cannot be the case. Some investors will want to reinvest their money allowing it to grow faster due to compounded interest, however some will be investing in the fund for a form of cash flow, by redeeming the interest payments for cash whenever they are paid. This means that the fund cannot be exactly sure how much of the extra capital that is received in the form of interest payment will be retained by investors, compared to redeemed by them for cash.

For this reason, we decided to implement interest payments using a separate balance, solely for storing interest that has been claimed, rather than incorporating it into the token balance they already have. Interest that will be owed to the investor can be calculated at any time over the interest holding period, however it can only be claimed into the individual's interestBalance once the holding period has ended. At this point, the individual can decide to either reinvest that interest balance during a future funding round, or redeem their interestBalance for cash (or a stablecoin alternative).

4.3 Final Design Overview

In order to describe the designed security token, and how it fits into the functions of Minexx and the ATIF, the following diagram was constructed. There are 5 main stages depicted by the diagram (please view the presentation slides for highlighted stages), and these can be broken up into:

1. Loan operations
2. Investor document verification
3. Token purchasing
4. Interest payments
5. Regulatory intervention

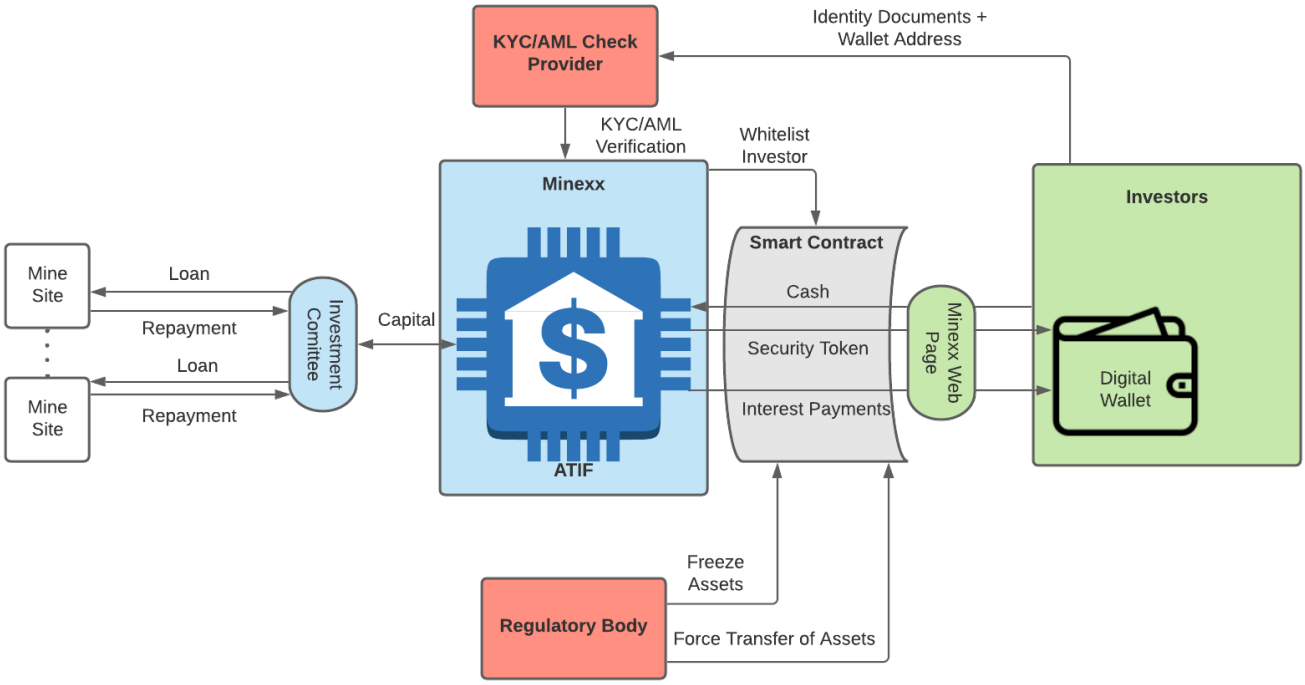

Fig. 1: Overview diagram for the ATIF and smart contract.

The 5 stages are described in the following sections. The actual implementation of the final design is described in the Token Technical Documentation on GitHub.

4.3.1 Loan operations

On the left hand side of the diagram, you are able to see how the fund decides to allocate their capital. As explained earlier, this process is controlled completely by an internal Minexx Investment Committee. This committee is in charge of scoping out new potential projects, and managing the loans that are paid out to the funded mine sites. They are also responsible for collecting the repayments made by the mine sites on the loans, as well as updating the website with any information regarding progress of mine funding and any loan defaults.

4.3.2 Investment Document Verification

As mentioned earlier, before an investor is able to purchase the security token from Minexx, in line with most countries' Know Your Customer and Anti-Money Laundering regulations, the individual must pass

identity checks, to prove they are who they say they are. In order for this to take place, potential investors will register their interest to purchase the tokens with Minexx, and at the same time be asked to submit required personal documents (e.g. passport, proof of address, etc.). Along with these documents, the investor will be required to send their bank details and their respective blockchain wallet address (for our code, this would be their Ethereum wallet address).

These documents will be sent to an external KYC/AML verification provider, of which there are many companies or digital solutions available. Once the provider has verified the investor and their documents, Minexx will interact with the smart contract, adding their wallet address to the investor "whitelist". This will inform the smart contract that this wallet will now have permissions to hold tokens (whether this is through a purchase or a transfer).

4.3.3 Token Purchasing

Once an investor has been added to the whitelist, they are able to interact with the smart contract, allowing them to send capital to the contract (which in turn transfers said capital to the Minexx account). Doing this will result in the smart contract accrediting their wallet balance (which is linked to their wallet address in the smart contract) with the desired amount of the security token. This is in essence how the purchasing of the security token takes place.

Whitelisting of an investor's address also means that they can now receive security tokens via transfer from another investor's wallet, for example in the case of buying security tokens on a secondary market.If the receiver address is not present on the whitelist, then the transaction will revert, meaning no tokens are sent.

4.3.4 Interest Payments

Once tokens are purchased, the owner is entitled to regular interest payments, with the interest rate set on a monthly basis by the Minexx Investment Committee. The monthly interest rate is calculated based on a number of factors.

1. The interest rate should be an aggregation of the interest rates receivable by the fund on the loans it provides to fund mine sites (as different loans will have different interest rates, depending on the risk profile of the loan application, determined by MineSmart and the Investment Committee).
2. The credit default rate on the loans should be factored into the interest rate. For example, if a higher percentage of loans are defaulted on in a particular month, the Investment Committee should decrease the interest rate for the next month, ensuring that the total value of the interest that they are paying out to investors is not greater than the interest payments they are receiving from the loans.
3. The portion of the value of the fund that is currently wound up in loans (compared to held in cash) must be taken into account as well. Ideally, in order to maintain as high of an interest rate offered to investors as possible, the fund should aim to have 100% of its funds loaned out at any one time - but of course, this cannot be guaranteed. The following calculation shows how the interest rate offered to customers should change based on the portion of the fund that has been loaned out:

Case	1	2
Fund Size	10M	
Avg. Interest rate (to debtors)	10%	
No. of Investors	1,000	
Total portion of fund loaned out	**100%**	**75%**
Interest received	1M	0.75M
Interest per investor	1000	750
Realised return for investor	10%	7.50%

Fig. 2: Calculation showing effect of fund portion loaned out on interest rate paid to investors.

4.3.5 Regulatory Intervention

In order to pass further regulatory requirements compared to just the KYC/AML laws [12], it is necessary for regulatory bodies to have overriding access to the tokens in someone's digital wallet. This is due to the fact that in order for the security token to legally qualify as a security, it must be able to be frozen by bodies such as HMRC, in the instance where an individual who owns some tokens is being investigated for tax evasion or fraud.

Normally, these bodies would be able to contact whatever institution this individual banks or invests with, and will send them a "hold notice", which essentially freezes all assets in their account until the tax/fine has been paid or the investigation has come to an end. It is worth noting that this is a last resort that they use. For this reason, there are functions within the smart contract that only allow regulatory bodies (pre-approved by Minexx) to freeze or force transfer assets out of an individual's account.

Note that although wallet addresses on Ethereum are public (as they are stored in transactions on the public distributed ledger), the investors' identities are not shown. These identities are only held by Minexx and the KYC/AML provider (both in private storage). Due to this, in order for a regulatory body to identify which address they need to take action with, they must first contact Minexx to retrieve this information.

4.4 Smart Contract: Next Steps

As the smart contract continues to be developed, we would like to implement the following changes to the token:

1. We would like to migrate the smart contract from the Ethereum blockchain to the specialized Polymesh blockchain, due to the number of advantageous features it has for the security token.
2. Adapting the smart contract so that it pulls the interest rate directly from a calculation on the MineSmart platform, rather than an individual in the Minexx Investment Committee manually inputting the monthly interest rate each month.

5 Webpage

The MNEX security token represents an investment in the ATIF, allowing external investors to participate in and profit from the funding of artisanal mine sites. However, the user experience of learning about and investing in the security token is vital in order to attract investors. This is especially true

as the fund operates in an unusual market and as few people know and trust security tokens yet [13]. Thus, we thought of the webpage as an essential part of the project. The webpage can be visited at minexx.codes.

5.1 User Experience Rationale

We designed this webpage with two things in mind: On the one hand, we want to advertise our token to potential investors and allow them to learn everything about Minexx, the ATIF and the MNEX token. On the other hand, we want to give current investors a direct and transparent channel to learn about where their money is going.
Accordingly, our webpage has four main pages.

5.1.1 Home Page

The landing page has been designed to catch the investors attention and give him an overview of the MNEX token. We give some background information on the ASM industry in Africa, introduce Minexx and explain how the MNEX token allows investors to get involved.

5.1.2 Projects Page

We decided to have a dedicated Projects page giving an overview of the ATIF's current and past investments. The rationale behind this page is to give investors the best possible transparency about were their money is going. We found this especially important as the individual investor does not have any control over which mines are being funded or not. Instead, this decision is being made by a dedicated funding committee.
The main section of the Projects page is a grid of all current and past mine sites. The goal in designing this grid was to give users a clear overview of all projects without loosing the possible depth of information. We do this through three levels of detail:

1. The normal grid shows only the most important information: An illustration of the mine site, its resource and location, and the funding progress. Furthermore, we added a traffic light system to give a simple indication of the quality of the investment. In the future, the color assigned would be decided via a number of variables defined by the ATIF, such as turnover, debt repayment and forecast events.
2. By hovering over any listed mine site, the user can find out the most relevant information about it through a compiled paragraph.
3. Clicking on the site brings up a separate page dedicated to each mine site with more in-depth details.

Currently, both the traffic light system and the funding progress have to be edited manually in the doc folder on the GitHub repository. As the development of the investment fund continues, we would automate this so that this information is sourced from a separate file or external database.

5.1.3 Invest Page

The main goal of the Invest page is to give interested users a straight forward place to invest in the MNEX security token. The page advertises the ATIF and the MNEX token itself.

5.1.4 Contact Page

Despite the information provided throughout the webpage, there are many unknowns that a potential investor would be faced with. We expect most investors to know little about the context of this fund, including the artisanal mining economy or the political and economic situation in Rwanda and the Democratic Republic of Congo (DRC). More so, the idea of Impact Investment Funds is rather new [14], and the idea of a security tokens even newer [13]. We thus thought created a Contact page to allow

investors to directly contact Minexx about any questions. We plan to extend this with page with a Frequently-asked-questions (FAQ) section over time.

5.2 Technical Choices

Our initial plan was to build the front-end of our webpage using a more sophisticated framework built on react.js [15], such as Next.js. Some of the advantages of using react.js over bare HTML, CSS and javascript are that code is guaranteed to be stable, easier to maintain, renders faster and generally brings more structure into the code. Next.js provides even more functionality. For example, it achieves much faster load times through code splitting, only loading the code required for a given page instead of the entire code base [16].
We ended up deciding against this for two simple reasons: Time and Necessity. Learning a new complex framework such as Next.js from scratch would cost a lot of time, which we do not believe would have been a good investment. As our webpage mainly serves as a channel to provide potential and current investors with information, regular HTML, CSS and javascript are enough to create a satisfying user experience. Developing in HTML and CSS also allowed us to better understand web development from the bottom up as other frameworks such as react.js simply build on top of them. We prefered this over developing in react.js straightaway without understanding what is happening on a lower level.
Furthermore, we currently use Github pages to host the minexx.codes webpage. This has been a great option so far as it allowed us to directly build and the webpage from our git repository and update the webpage simply by pushing changes to Github. However, Github pages has a few drawbacks for production-ready deployment. Among others, it takes longer for pages to load and does not allow for developer features like site previews [17]. Ideally, we thus would host the webpage on a faster, more feature rich deployment platform instead, such as Vercel [18] or Netlify instead.

5.3 Webpage: Next steps

As the project would develop closer to reality, we plan to make some adjustments and upgrades to the webpage:

1. First, we would like to give investors even more transparency by dynamically pulling relevant statistics from the MineSmart platform, such as the number of past and current investors and the average interests generated over the past months. We think this should be displayed on the Invest page as it fits with information such as the current funding round.
2. Once we switched to the Polymath blockchain, we would also like to directly integrate a crypto exchange into the Invest page webpage, for example using the Polymath API. This way, investors would only need to visit `minexx.codes` to buy, sell and trade the MNEX token.
3. As mentioned above, we also want to extend the Contact page with comprehensive answers to frequently asked questions.

References

[1] Notion.so, "All-in-one project planning platform," http://notion.so.

[2] U. Nations, "The problem of small loans in africa," https://www.un.org/africarenewal/magazine/january-2009/small-loans-widen-horizons-poor.

[3] Deloitte, "Security token offerings," https://www2.deloitte.com/content/dam/Deloitte/cn/Documents/audit/deloitte-cn-audit-security-token-offering-en-201009.pdf.

[4] J. W. Michael Jünemann, "Ico: Legal classification of tokens: Utility token," https://www.twobirds.com/en/news/articles/2019/global/ico-legal-classification-of-tokens-utility-token.

[5] Homepage, "Finboottech," https://www.finboot.com.

[6] "Marco track & trace platform," https://www.finboot.com/track-trace.

[7] Polymath, "Polymesh whitepaper," https://polymath.network/polymesh-whitepaper.

[8] M. Alberto Cuesta Cañada, "Fixed point math in solidity," https://medium.com/cementdao/fixed-point-math-in-solidity-616f4508c6e8.

[9] CementDAO, "Fixed point math in solidity," https://www.hebergementwebs.com/news/fixed-point-math-in-solidity.

[10] A. Consulting, "Abdkmathquad," https://github.com/abdk-consulting/abdk-libraries-solidity/blob/master/ABDKMathQuad.md.

[11] R. Hitchens, "How to store a float or decimal in a contract?" https://ethereum.stackexchange.com/questions/35150/how-to-store-a-float-or-decimal-in-a-contract.

[12] Blockpass, "Kyc for token offerings," https://www.blockpass.org/kyc-for-token-offerings/.

[13] R. Foundation, "The coining of sto in 2007," https://www.rockefellerfoundation.org/blog/bringing-scale-impact-investing-industry/.

[14] S. Market, "First sto legislation passed in 2017," https://blog.stomarket.com/defining-security-tokens-list-of-all-countries-with-legal-definitions-of-digital-securities-6b19eab6c330.

[15] https://reactjs.org.

[16] "Advanced features: Dynamic import," https://nextjs.org/docs/advanced-features/dynamic-import.

[17] Bejamas, "Vercel vs github pages," https://bejamas.io/compare/vercel-vs-github-pages/.

[18] "Vercel deployment platform," https://vercel.com.