Haaga-Helia
University of Applied Sciences

Decentralized Finance – A deep dive into the future of finance and how to build on it.

Corneo Patrick

15 March 2022

Abstract

Haaga-Helia

15.04.2021

Author Corneo Patrick	
Degree program Bachelor's degree in business information technology – Exchange Student	
Report/thesis title Decentralized Finance – A deep dive into the future of finance and how to build on it	**Number of pages and appendix pages** 38 + 4

Decentralized Finance, often called DeFi, is a new financial system built on top of a blockchain. Its main point is that it does not relies on centralized institutions, such as brokers or banks, to deliver financial instruments to its users. By removing the middlemen, cost of transactions is often reduced, and financial trades can be made peer-to-peer.

In January 2020, it was estimated that there were 109'132 unique wallet addresses and just a year later in January 2021, this number rose to 1'306'315. Nowadays, in March 2022, the estimations tell us there is about 4'522'000 unique addresses.

This large but also rapid increase in the number of users, shows us that DeFi is getting more and more adopted by a lot of users and that it could play a significant role in our financial system. But if we compare it to the data of cryptocurrencies users, which was 295 million as of December 2021, we can see that even compared to its own niche, it has a lot of room to grow before becoming mainstream.

The goal of this thesis is to show what problems are or can be solved by DeFi, it's use-cases and the new dangers that might come with it. But also, how to learn programming on it and learn how to build a Decentralized Application on the Ethereum blockchain.

Keywords
Blockchain, cryptocurrency, Ethereum, DApp, DeFi, smart contract

15.04.2022

Haaga-Helia

Abstract

15.04.2022

Table des matières

1. Introduction 0
2. Centralized Finance 2
   - 2.1. Centralized control 2
     - 2.1.1. Rising Bank Concentration 2
     - 2.1.2. Banking reserve requirements 2
     - 2.1.3. Banking fees 3
   - 2.2. Limited Access 3
   - 2.3. Inefficiency 3
     - 2.3.1. Credit card fees 3
     - 2.3.2. Slow transfer of funds 4
     - 2.3.3. Wire transfer fees 4
   - 2.4. Lack of interoperability 4
   - 2.5. Opacity 5
3. The blockchain 6
   - 3.1. Bitcoin blockchain 6
     - 3.1.1. The 2008 subprime crisis 6
     - 3.1.2. The creation of Bitcoin 6
     - 3.1.3. Transactions 7
     - 3.1.4. Timestamps 8
     - 3.1.5. Proof-of-Work 8
     - 3.1.6. Incentive and inflation 9
     - 3.1.7. Privacy 10
     - 3.1.8. Inflation 10
   - 3.2. Altcoins blockchains 11
     - 3.2.1. Altcoin 11
     - 3.2.2. Blockchain trilemma 11
     - 3.2.3. Proof Of Stake 12
   - 3.3. Ethereum 13
     - 3.3.1. Smart Contracts 13
     - 3.3.2. DeFi statistics 13
4. Decentralized Applications 14
   - 4.1. Their aspect 14
   - 4.2. Oracles 14

4.2.1.Definition 14
4.2.2.Chainlink 15
4.3. The storage of data 16
5. Decentralized Finance use cases 17
5.1. Lending 17
5.1.1.Traditional lending 17
5.1.2.DeFi lending 17
5.1.3.The mechanism behind it 18
5.1.4.Advantages 19
5.1.5.Lack of options 19
5.2. Exchanges 20
5.2.1.Centralized exchanges 20
5.2.2.Decentralized Exchanges 20
5.3. Tokenization 22
5.3.1.Definition 22
5.3.2.Examples 22
5.4. Non-Fungible Tokens 23
5.4.1.Definition 23
5.4.2.Purpose 23
5.4.3.Possible future functionalities 25
6. Decentralization Risks 26
6.1. Smart Contracts 26
6.2. Oracles 26
6.3. Custodial 27
6.4. Scaling transactions 28
6.5. Environmental 28
7. How to code a Decentralized Application 29
7.1. Programming languages 29
7.1.1.Solidity 29
7.1.2.JavaScript 29
7.1.3.Typescript 30
7.2. Frameworks 30
7.2.1.React 30
7.2.2.Web3.js 30
7.2.3.Truffle 31
7.3. Development Environment 31
7.3.1.Remix 31
7.3.2.Visual Studio Code 31
7.4. Testnet 32

7.5. Smart Contracts 33
7.5.1. Version 34
7.5.2. Addresses type 34
7.5.3. Msg 34
7.5.4. Revert 34
7.5.5. Built-in triggers 35
7.6. Further programming expertise 36
7.6.1. Distributed storage 36
7.6.2. Security 36
7.6.3. Blockchains types 37
7.6.4. The future of Ethereum 37
8. Conclusion 38
9. References 1

1. Introduction

Finance has always evolved through history of time. At first, economy being inexistent compared to today standards, our ancestors developed a primitive economy style called ‘Bartering’, it consists of exchanging goods for other goods, without having a medium exchange, such as money like we have today. (Wikipedia, 2022b).
The problem with this type of economy is the lack of common grounds, how much does each good cost when compared to another one? Well, it depended on the person that had the good.

Around 500 B.C., our ancestors started using coins resembling our current coins. It first started in Lydia, currently Turkey, but quickly developed to other empires, such as the Romans, that then had their currency used in other regions of the world due to their powerful state. The coins were made of precious metal, such as gold, which therefore had some inherent value.
This allowed people to trade their goods on a common basis, making exchanges easier. (Wikipedia, 2022a).

In today’s world, we still have coins and paper money. But most of our transactions, whether we are buying groceries, paying for vacations abroad or buying fancy furniture’s that draw our attention, are made through credit or debit cards. (Shepherd, 2020).
When we use them, it looks as if our money is transferred from our bank account straight to the retailer’s account. But behind this apparently simple transaction, there is numerous other approval’s that need to be made before we can make our purchase.
So, if a merchant wants to implement credit cards payments to facilitate their customers buying experience, he must have a processing company that allows them to have these approval’s.

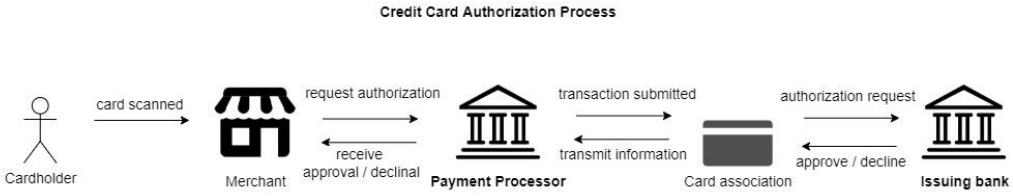

Figure 1: Credit card authorization process.

First, there is an authorization process, where a client scans its card on a device, then a request authorization is sent to the merchant’s payment processor, who submits the transaction to the corresponding card association, that will then request authorization to the client’s issuing bank.
When the bank gets the demand, depending on different points such as if the client has enough money available, allows or denies the transaction.
Then the answer goes through every other intermediary and all the way back to the merchant’s device. (Fis Global, 2022).
This all happens in a couple of seconds but makes the merchant reliant on all the other participants.

Along with the credit card intermediaries’ problems, our financial world has other numerous problems that need to be solved. To name some key problems, before diving deeper into explaining their cause and how they might affect us, there is:

- Centralized control, our funds are managed by the banks which decide if we can access them or not for various reasons.
- there is a limited access to banking systems since 1.7 billion people are unbanked, which makes it very difficult for them to have loans or make purchases over the internet.
- Inefficiency, as shown before with credit cards transaction fees.
- Lack of interoperability, when wiring money from a bank to a non-bank account it might take longer than we need it to.
- Opacity, bank customers don’t know where their money goes when deposited in a bank,

thus they don’t know how their bank would react to a bank run.

These problems have a bigger impact on the regular consumers than the wealthy classes, therefore the people in power of our governments never tried to solve them or even some of them for a start.

This is where the Blockchain, and more specifically Decentralized Finance, comes in play. Decentralized Finance, commonly called DeFi, has for main goal the removal of third parties on any kind of financial transaction or exchange. It gives users more freedom with their assets and allows them to have full control of their actions.
But to understand how DeFi could solve our current financial world problems, we must first identify them.

2. Centralized Finance

To understand why Decentralized Finance is evolving rapidly and why it was created, we first need to understand our current financial world, also called Centralized Finance (CeFi) in the cryptocurrency's world, it's problems.
So, let's look at the biggest problems in CeFi.

2.1. Centralized control

Most of our money is kept in banks, therefore they are the ones that have full control over our funds and what do to with it, meaning that there is a centralized control over our funds.

2.1.1. Rising Bank Concentration

Research made by Federal Reserve bank of Minneapolis, found that from 1984 to 2018 the concentration of funds among the top 4 commercial banks in the US, has risen from 15% to 44%, which is three times higher. (Corbae & D'Erasmo, 2020).
In Europe, the top 4 commercial banks hold 42% of all funds are detained by the top 4 commercial banks. (Godoy, 2019).
Those numbers are high, but if we take a country, Finland for example, 94.17% of all funds are detained by the top 3 commercial banks which is way more astonishing. And there is even a big gap even between the top 3 banks assets. The 1st one with a total of 552.2 billion euros of assets, the 2nd one with 160.2 billion euros which is less than one third of the 1st one and the 3rd one has only 7.6 billion euros, that is 72 times less that the first one. (Norrestad, 2021).
These numbers about bank concentration will probably keep going up, since the number of banks in Europe is decreasing. From 2008 to 2019, there was a diminution of 30% banks. Germany, the European country with most banks, went from 1989 banks in 2008, to 1445 in 2021, a drop of 27%. (Norrestad, 2022).
The banks that closed were mainly banks not making enough profits due to the lack of clients and mostly small banks.
The point is that customers can't freely choose where to store their money, they will probably have to choose between a handful of banks most of the time. And this is even more accurate when they are in small towns with little to no banks close to them.

2.1.2. Banking reserve requirements

Banks are businesses too, so they need to make profit. So, almost all the money deposited is not stored or kept safe. Instead, it is lent to other customers, who will then pay interests on the loan and the bank will profit from it.
The amount of money banks must keep in reserve is regulated by the 'Federal Reserve System' for the USA and by the 'Central European Bank' for Europe.
Let's focus on Europe, the CEB used to require from banks that they held a minimum of 2% of their balance sheet, since January 2021 it has been lowered to 1%. (ECB, 2016).
It means that theoretically, on average, when a customer deposits 100€, their bank only needs to maintain 1€ in reserve and they can lend the other 99€ to someone else to profit from the interests.
What does it mean for their customers? Well, if they want to withdraw above a certain amount of money in cash, they need to go through a certain procedure where they must first ask permission to the bank and then, if they're allowed to, the bank needs to get the money ready which takes a couple of days and finally, the client can have his own money.

2.1.3. Banking fees

To have a bank account, the client needs to pay a fee that corresponds to different parameters, like the cost of having a credit card or SMS alerts. (Crédit Agricole, 2021).
With the interest rates being negative for years, all the banking reserve that is stored in the central bank costs money to the banks, therefore they must find ways to compensate for this loss. To do so, services that can be done online and the client wants to do in person at the bank are now charged a fee. (Crédit Agricole, 2021).
So even though as seen previously we give the banks our money, then they use it to lend it to other clients or banks and earn interests on it, we still must pay more and more to have them store our money. (ECB, 2022). For some people the cost of having a bank account, or using bank services, might be expensive compared to their salaries.
This shows us some problems of the centralized control that there is over our money and to which customers have to agree whether it benefits them or not.

2.2. Limited Access

Even though our financial system has been built over 100 years ago, there is still a lack of access for a lot of the world population. It is estimated that in 2017 there was still around 1.4 billion people that were completely unbanked, which means they must keep their money in cash themselves. (Demirguc-Kunt, Klapper, Singer, Ansar & Hess, 2018).

With a number this high of unbanked people, we can only imagine how much only have a bank account for their daily use and their paycheck, but still are underbanked and can't access financial services such as loans or credit cards and instead they must use money orders, check-cashing service if they can't have a checking account, or payday-loans for their everyday needs. (Rasure, 2022).
Also, entrepreneurs, who most of the time have high needs of financing to get their business started, sometimes need to use their credit cards, and spend more than they have in their bank account, making them must pay credit fees up to around 15%, which might be huge for them and could slow their enterprise growth, maybe even financially ruining it. (Campbell, 2021).

2.3. Inefficiency

The current financial world has a lot of inefficiencies in it, either they are fees or long delays, which could maybe be reduced, or even removed, by having a decentralized finance.

2.3.1. Credit card fees

As mentioned before, enterprises must go through credit cards associations to have a terminal being implemented.
But with doing so, each transaction that is made has up to 4% of credit card processing fees, which are reflected in the price of the good or service being sold. (Williams, 2022). Customers won't notice it since it is included in the price, but this does reduce their purchasing power by inflating the prices of services and goods.
It also reduces the profit margin of the retailer because knowing that customers are ready to pay that price with the 4% included in it, if he had not those fees, he could reduce his prices by 2% and still have up to 2% of more profits than before.
Therefore, removing credit card associations as intermediaries would be beneficial for all parties involved in a transaction.

2.3.2. Slow transfer of funds

Another problem is the delay of wire transfers between banks.
Some countries offer faster transfers if the recipient has the same bank, such as 'instant transfers' that take only a couple of minutes. Or when it is with a different bank but in the same country, it might only take a day. (N26, 2022).
But when it comes to international transfers the delays start getting longer, there might be 1 to 2 working days before your payment is received by your counterparty if it is a SEPA transfer within Europe. (N26, 2022).
When it is a SWIFT transfer, as a European person sending money to the USA, delays can go up to 4 working days, which could lead to almost a week if the weekend is in between. (Smith, s.a.).
At the age of internet, with optical fiber cables able to go up to 1GB/s, having to wait multiple days for a transfer shouldn't be normal, and DeFi solves that.

2.3.3. Wire transfer fees

I just mentioned wire transfers having too much delay, but there is also the fact that they can sometimes be charged a fee to either the sender or the recipient.
Within Europe, there is the possibility to use SEPA (Single Euro Payments Area) transfers, which cost little to no money to do and that is a great thing for customers. But just like for credit cards fees, there is a hidden cost which is then translated into the customer's service charges, generally paid monthly. Still, those fees are around a few dozen cents per month, mostly less than 50 cents, so we could consider SEPA transfers as free.
But as soon as the transfer goes outside of Europe, we need to use the SWIFT (Society for Worldwide Interbank Financial Telecommunication) system and the fees we need to pay start getting higher. The costs range from a couple of euros, 5.25€ in "Crédit Agricole" in France for a transfer of less than 200€, and can go up to dozens of euros, 31€ in Crédit Agricole (Crédit Agricole, 2022) for a transfer of over 1500€ and for Nordea bank in Finland the cost could go over 40€ (Nordea Bank, 2022). depending on different fees you may have to pay to have you transfer go through.

Those 3 problems already show us the inefficiency that there is in our current financial world and could, even should, be solver in our modern times with all the newest technologies available to us.

2.4. Lack of interoperability

Our current institutions are heavily siloed, and each network doesn't interact with another network as fast as it does within itself.
This means that whenever we want to send money from a banking institution to another or from a banking institution to a non-bank, like a stock market broker, we can't do it instantly or with a short delay because we must do so by using SEPA or SWIFT transfers.

This is inconvenient if we need the money to be deposited fast to do a certain action.

For example, if I spot a great opportunity on the stock market and I want to buy a stock option, but I don't have the money on my broker right now. I'll first need to send the money from my bank account to the broker, which might take between 1 to 4 days, depending on the type of transfer. By the time the money is deposited, I probably already missed the opportunity.
And this is going to happen every time if I can't let my money on my broker account because I might need it on my bank account.

Some companies started to recognize this problem and therefore tried to take actions to solve it. Like Visa, that tried to acquire Plaid in January 2020, a FinTech that amongst other services and solutions, helps their users to obtain faster transactions and fundings, making services more

interoperable. But the acquisition of Plaid had to be stopped due to some jurisdictional problems concerning data privacy and monopoly reasons. (Harvey, Ramachandran & Santoro 2021, 6). Although, this shows us that Visa knows there is a problem with the length of transfers and wants to solve it.

2.5. Opacity

Different banks might offer different interest rates, but the customer does not know why there is a difference from a bank to another and why one of them has better offers on its services. It might be because the bank has a larger profit margin and can rise its interest rates to attract more customers for a certain time until going back to lower rates, but it might also be because the health of its finances isn't doing great, and the bank is in dire need of new customers to save it. (Harvey, Ramachandran & Santoro 2021, 6-7).

3. The blockchain

3.1. Bitcoin blockchain

Since the concept of blockchain first appeared at the same time as the creation of Bitcoin by Satoshi Nakamoto, we are going to start by introducing Bitcoin.

3.1.1. The 2008 subprime crisis

To explain the context and reasons of why Bitcoin was made, we need to focus on the financial crisis that occurred in 2007-2008, also called the “Subprime crisis”. (Wikipedia, 2022c).
It occurred due to credit lending standards being lowered by investment and commercial banks, which therefore increased subprime lending.
A subprime lending is the action of giving a loan to a lower income population that might not be able to adhere to the repayment schedule and therefore be insolvent.
These loans might seem too risky for banks to give them, but by doing so they asked higher interest rates and gave them worse terms than other loans, thus trying to gain as much as possible from giving them and taking higher risks. (IG, s.a.).
But, due to the FED, the federal reserve system which is the central bank of the United States of America, having high rate hikes from 2005 to 2007, these subprime loans couldn’t repay their debt anymore due to their mortgage payments being too high after interests being taken into account. (Macrotrends, 2022).
This led to a lot of loans not being repaid, therefore their house taken and being sold, collapsing the value of the housing market and the houses being valued less than the loan that was given by the bank.
Due to this, financial markets started following, collapsing one after the other, as we can see when looking at any past charts on any trading website.
The SP500, a stock market index tracking the 500 biggest US companies listed on exchanges, falling by 58% in less than a year. (Kenton, 2022). The CAC40, which tracks the 40 largest French companies, fell by 60 over 2 years. (Hayes, 2022a). And finally, the Bovespa, the Brazilian stock market index tracking its 50 most valued companies, fell by 60% in half a year. (Hayes, 2022b).
I took 3 countries that aren’t on the same world regions, to showcase how big this crisis was and that it was indeed worldwide, even though it started with an American subprime lending crisis.

3.1.2. The creation of Bitcoin

While all these events were occurring, Satoshi Nakamoto, who might be a person, or a group of persons, started creating a new peer-to-peer financial system, Bitcoin.
Nakamoto stated that he worked on the project throughout 3 years, from 2007 to 2009.
Bitcoin is based on the b-money concept, designed by Wei Dai in 1999, and bitgold, described in 2005 by Nick Szabo. This led to Nakamoto wanting to create a peer-to-peer cash electronic system based on cryptography and trustless. (Lars, 2020).
On Bitcoin whitepaper, Nakamoto stated that the fact of having to rely on a third-party to trust a transaction, wasn’t practical due to possible mediating disputes because the transaction can be reverted. (Nakamoto, 2008, 1).
And because of the cost of these possible mediations being considered, the cost of transaction might be higher than expected, which increases the minimum practical transaction size. Meaning that, for example, a transaction of 5€ wouldn’t be possible due to the costs being higher than the amount transferred, so it cuts all possibilities of small transaction between people.
Also, since a transaction can be reversed in our current financial system, retailers must apply the KYC process, Know-Your-Customer, which leads them to asking more information than needed on their customer just to be safe in case of a transaction reversal after delivering the good or service, so they can protect themselves by following up with lawsuits if needed. (Wikipedia, 2022l).

Of course, all these problems are inexistant when dealing with physical currency, but as stated before, most transaction nowadays are made with credit cards or other communication channels.
Therefore, Nakamoto thought we needed a system based on cryptography and trustless, solving these unwanted costs by removing the need of a trusted third party and making transactions irreversible. By doing so, sellers would be protected from fraud, and buyers too, with the implementation of routine escrow mechanisms. (Nakamoto, 2008, 1).

3.1.3. Transactions

According to Nakamoto in Bitcoin's White Paper, for a transaction to be valid, we need two things: the proof that the one sending the currency currently owns it and the proof that the owner didn't double-spend the owned currency by sending two transactions at the same time.
To solve the first problem, Nakamoto proposed to have each coin containing the current owner public key and a hash of the previous transaction made with the same coin, all of this signed by the previous owner who used its private key to do so. Here is a diagram describing how the process would look like.

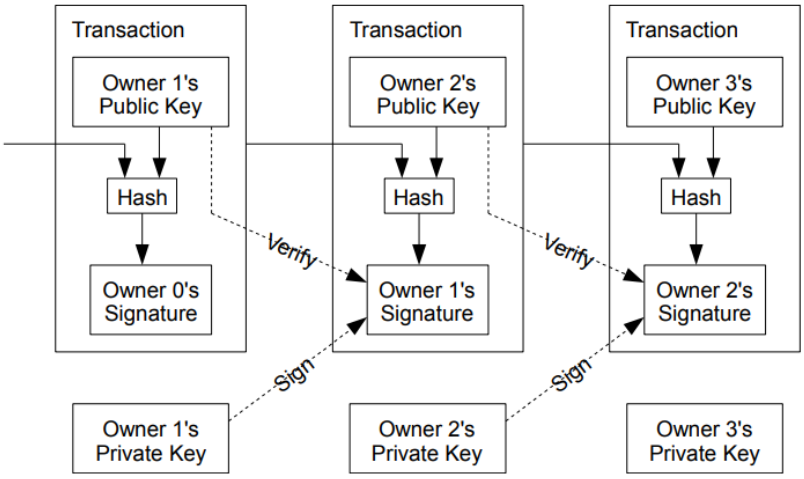

Figure 2: Transactions in the Bitcoin blockchain (Nakamoto, 2008, 2).

But this does not solve the double-spending problem, since the sender might use an older transaction hash and state that it is still his coin even though there is somewhere a more recent transaction stating that he already spent it.
To solve this second problem, Nakamoto stated that the receiver needed a way to find the earliest transaction made by the sender on this coin, so his newer attempts to double-spend it wouldn't be considered. And to do so without a trusted third party, all transaction must be public and there must be only one timeline for all orders. (Nakamoto, 2008, 2).

3.1.4. Timestamps

This is where timestamps come into play. Each new hash includes the previous timestamp and the current one, which then forms a chain, and all additional timestamps reinforce the previous ones, as seen on figure 3. By doing so, this creates a chain of hashes that have a certain chronological order and avoid double-spending. (Nakamoto, 2008, 2).

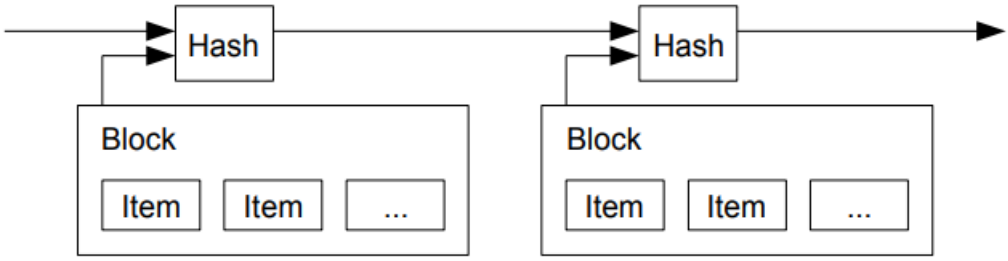

Figure 3: Timestamp Hash included in a block (Nakamoto, 2008, 2).

3.1.5. Proof-of-Work

Nevertheless, to implement this timestamp on a peer-to-peer basis, Nakamoto thought it needed to use a proof-of-work system.
Proof-of-Work, also called PoW, is the fact of having a CPU solving a calculus with a certain complexity. As showed in figure 4, multiple CPUs try to solve it, but the one who gets the correct solution has its block added onto the chain and gets the reward. (Ledger, 2019).
The reward consists of coins, which for the Bitcoin blockchain is Bitcoins, and this is how the 'miner', which is the person or entity running the CPU that solved the calculus, gets paid for his work. (Hong, 2022).

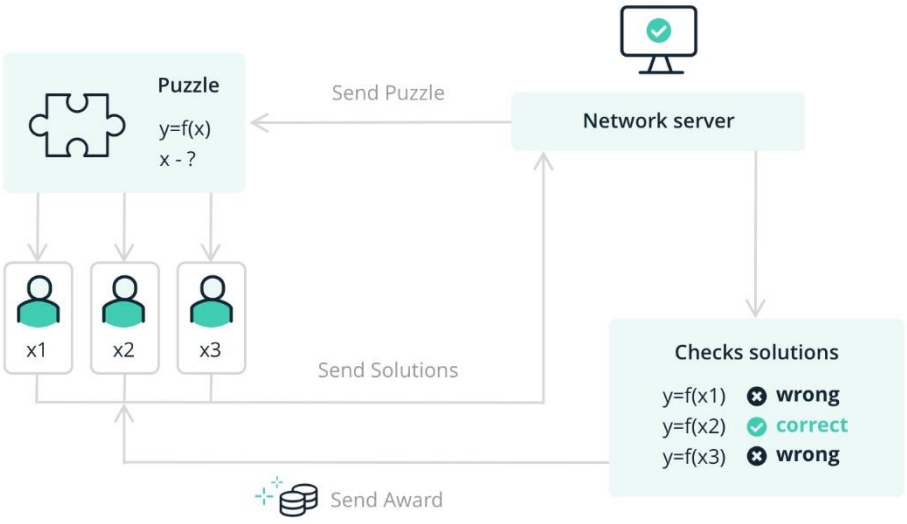

Figure 4: Proof-of-Work mechanism (Ledger, 2019).

And this secures the chain because, there is only one timestamp block added each time, making a double-spending error being impossible to occur.
It also avoids a malicious miner from trying to change previously created blocks because to do so, he would also need to change all the following blocks that were created after that. So, for a malicious miner to achieve this, he would need to have more than 51% of all the computing power of the blockchain, just so he can be faster than all other miners combined to have a chance to

solve the algorithmic puzzle before them and change the previous block. (MIT, s.a.).
Also, the difficulty of creating a new block increases according to an average number of blocks per hour. It there is too many per hour, the difficulty increases, keeping a stable rate even if someone manages to gather a lot of CPU power in a short period of time.
Therefore, it makes it almost impossible for an attacker to falsify the blockchain, and it becomes even harder as there are more and more validators because it would require more CPU power to have 51% of it.

3.1.6. Incentive and inflation

As stated before, to have miners do the work to maintain the security of the blockchain by checking he integrity of transaction and creating new blocks, they receive incentives in the form of new coins that are created, which are bitcoins in this case. (Hong, 2022).
This also adds a way to distribute new coins into circulation, without the need for a central entity, like a central bank as the FED.

But they also, receive incentives from transaction fees that are calculated from the output value being lower than the input one, and miners perceive the remainder. (Origin Stamp, s.a.).

Also, it is logic that miners receive an incentive for their work, since they must use CPU time, which deteriorates over time, and electricity, that depending on the country might have a higher cost. We could compare them to gold miners, that use their time and resources to add gold into circulation and then receive gold as the incentive.

But, just like gold, Bitcoin has a fixed supply, which is 21 million, therefore once this limit is reached, they won't receive new bitcoins as an incentive for their work, but they will continue to receive transaction fees paid by users. (Hayes, 2022c).
This means that at 21 million bitcoins created, it won't be an inflationary currency anymore and it would be beneficial to current users.

3.1.7. Privacy

Nowadays, as stated before, companies need to KYC their customers if they want to avoid being frauded, or there needs to be a third-party who knows their customers identity and insures them of being protected.
We can see this model on figure 5, and below it, the 'new privacy model', which corresponds to the Bitcoin one.

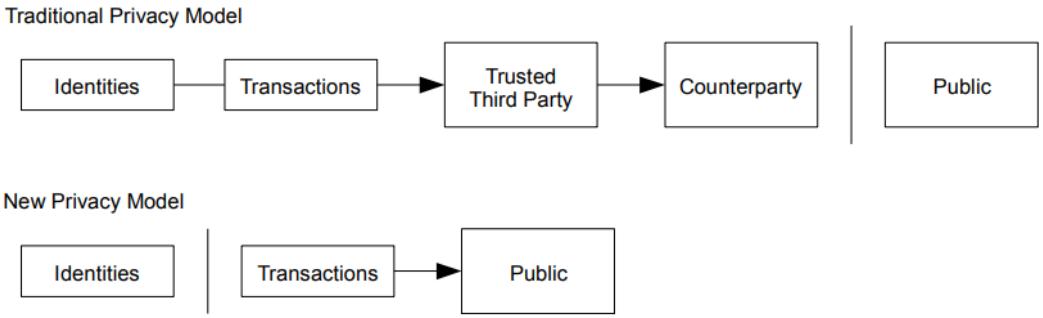

Figure 5: the privacy model of Bitcoin (Nakamoto, 2008, 6).

Since all transaction are publicly available, the older method can't take place anymore. But there is a new layer that allows user to hide their identity and that is the fact that their public key can remain anonymous, there is no need to link it to a person. This makes the Bitcoin blockchain neither anonymous nor public, it makes it pseudonymous, we know there is a public key doing certain actions and sending transactions, but we don't know who that is. (Nakamoto, 2008, 6).

Also, if a user wants to remain as private as possible, each transaction should be done with a new key pair, so multiple transaction can't be linked to each other.

3.1.8. Inflation

As said before, Bitcoin has a fully diluted supply of 21 million coins, therefore once that limit is reached the currency will no longer be inflationary.
On the Bitcoin white paper, Satoshi Nakamoto doesn't directly state why this was made, but nowadays there is a consensus saying that this system was done to fight inflation and detach from the current fiat model, where central banks and government institutions can increase the supply of their currency by printing more money, which in correlation decreases the value of the currency. (Phillips, 2020).
Bitcoin supply can't be increased as liked by a central entity, instead it is increased by the miners' incentives received for their work, but every 4 years there is a halving in the quantity of coins received by miners. Therefore, every 4 years Bitcoin inflation is halved, slowing its inflation, until the limit is reached. (Conway, 2021).
The estimated date for the mining of the last bitcoin is in year 2140, so until then miners will continue to receive coins as incentives. (Buchko, 2022).

3.2. Altcoins blockchains

3.2.1. Altcoin

Altcoins, or ‘alt coins’, meaning ‘alternative coins’, which englobes every cryptocurrency other than Bitcoin itself. (Frankenfield, 2022a).
Altcoins often have common points with Bitcoin but are different on others.
For example, on the Ethereum blockchain, the second biggest cryptocurrency by market capitalization, there are smart contracts which are used to build decentralized applications, a functionality that the bitcoin blockchain has not. Later we will do a deeper review of the Ethereum blockchain and smart contracts.

3.2.2. Blockchain trilemma

When building a blockchain, there are 3 main characteristics to it:

- Scalability: for a blockchain, it is the ability to being able of change the number of transactions per second that can be done. A scalable blockchain would be needed if there is a high number of users using it every day, therefore the creating the need to support a big quantity of transactions.
- Security: a blockchain is secure by definition, but not entirely hacking proof. If someone manages to have over 51% of the network validators, they could alter the blockchain and manipulate transactions, making them able of stealing from it. Therefore, the more nodes/validators it has, the more secure it is. (Frankenfield, 2022b).
- Decentralization: it means that there is no central authority that can take control of it. Even better, it could be entirely controlled by a program previously coded which works on its own. The more decentralized, the better it is.

The blockchain trilemma means trying to find the perfect balance between these 3 aspects. Most of the time having a good combination between 2 of the 3, means the third one isn’t good enough. (CoinMarketCap, 2022).

Blockchains need to be decentralized and secure.
Decentralization is the main point of having a blockchain, so that there is no central authority, and everyone has the same access to it. In traditional finance, we have banks, which act as a third-party between other entities exchanging currencies. But they have authority to freeze your funds for various reasons and you might not be able to get your money back.

A recent case on this topic would be the recent events that occurred in Canada. After a huge protest against a Covid vaccine mandate, Justin Trudeau, the Canadian prime minister, decided to take extreme measures to stop them. So, he started ordering banks to freeze assets of previously identified protesters to dissuade other people of joining the protests and forcing them to stop. (BBC, 2022).

Blockchain wants to avoid this, one shouldn’t see its funds frozen due to its views or beliefs. Unless a criminal case has been conducted and justice found the person to be guilty before doing so.

3.2.3. Proof Of Stake

Previously, when explaining the Bitcoin blockchain, how it was built and how it works, we talked about 'Proof of Work' (PoW).
But even though some of the biggest blockchains, like Bitcoin or Ethereum, still use this type of protocol, the newer ones tend to prefer the Proof of Stake (PoS) mechanism.
As opposed to Proof of Work where miners are needed to run the blockchain and secure it, and by doing so they create new cryptocurrencies, Proof of Stake doesn't need 'miners' that are continuously calculating the hash of the next block. (Coinbase, s.a.).

PoS validates blockchain transactions by using the native token of the blockchain, such as 'Binance Coin' for the 'Binance Chain'.

There are multiple ways to implement PoS, the major ones are:

- **Pure Proof of Stake (PPoS):** in this type, everyone could become a validator if they have the required amount of the blockchain native currency. The required amount is defined withing the creation of the mechanism. (Algorand, s.a.).

Once the user has enough cryptos, they can 'Stake' them for a certain period, which means they are now locked, can't be moved and are able to validate transactions. The higher the amount of cryptos locked, the higher the chances to being chosen by the blockchain algorithm to create a block.
Once the chosen validator creates the new block, the role of other participants is to verify each transaction within the block and make sure they are all correct to avoid the block creator of trying to validate fake transactions.
If they indeed tried to add some fake transactions, they would be denounced by others and then the block would be invalidated and the previously staked funds by the user could be destroyed or given to the denouncer who found out the validator was trying to cheat. This mechanism is called 'slashing'.
If not, the block is validated, transaction executed and the blockchains remains secured. (Algorand, s.a.).

- **Delegated Proof of Stake (DPoS):** as opposed to PPoS where one could stake its funds and become a validator. In a Delegated Proof of Stake mechanism, the staked funds are used to vote for a validator. (Le Journal Du Coin, 2022).

In this case the blockchain has a reduced number of validators, like Tron which has 27, since the requirements of becoming one are quite hard to attain, such as a high computing power and sophisticated material. Its users can stake their funds and delegate them to one of these validators. By doing so, they increase that validator total cryptos staked and therefore their chances of being chosen to create the next block.
This type of Proof of Stake allows the blockchains to be more scalable since the blocks must be shared between a reduced number of validators as opposed to a PPoS where anyone could become a validator.
The trade-off of this consensus mechanism is that it could become more centralized because when a validator has most of the users' funds, they will be the one receiving all the cryptos created by each new block. (Le Journal Du Coin, 2022).

There are other types of Proof of Stake, such as Liquid PoS, Leased PoS or Importance PoS, but we will not review them as they are less frequent.

3.3. Ethereum

Now that we have introduced the concept of blockchain and altcoins, let's have a focus on a particular one of these two: Ethereum.
Ethereum is the second biggest cryptocurrency by total market value. It became so in January 2018, and it hasn't lost its place since then.

3.3.1. Smart Contracts

Ethereum was the first cryptocurrency to add the smart contracts functionality.

A smart contract is a piece of code which is deployed in a blockchain. The code can contain different functionalities with various purposes. The main reason for having smart contracts is to remove the need for two parties to trust each other when exchanging funds, or assets, and therefore removing the need for a trusted third party. (Wikipedia, 2022d).

There is a wide range of possibilities in which smart contracts could be used but to which they aren't yet because of they wouldn't be legally binding nor recognized by governments.
Later, when exploring DeFi use cases we will review some of the possible use cases for smart contracts. For now, we will focus on the fact that Ethereum introduced them and that it was a huge step forward for cryptocurrencies use cases apart from being a store-of-value, mean of payment or inflation hedge.

Also, in the part explaining how to learn to build a DApp, we will go more into details explaining how smart contracts are coded and how to interact with them.

3.3.2. DeFi statistics

DeFi Llama is a known website that has a leaderboard containing most decentralized applications and ranking them by their total value locked (TVL), which corresponds to the total USD value of all the cryptocurrencies that are deposited on a DApp either for staking, lending, or other functionalities that we will see more in depth later. (Defi Llama, 2022a)
When looking at the leaderboard ye can see that 8 out of the top 10 DApps with most TVL available on the Ethereum blockchain.
This shows us that, even though in 2021 there were a lot of promising other blockchains with decentralized applications, as of 2022 Ethereum still remains the biggest blockchain in the Decentralized Finance sector.

Also, if we look more from the outside and focus directly on the TVL by blockchain, according to DeFi Llama, we can see that Ethereum has a TVL of around 55.3 billion dollars, while the second biggest blockchain only has around 6.7 billion dollars. That means the TVL of Ethereum is more than eight times greater than its biggest competitor. (DeFi Llama, 2022b).
Ethereum has 64% of the total decentralized finance TVL, even after a huge inflow of money throughout 2021 and a growing number of competitors, it still managed to retain most of the money in the sector.

Gathering all these statistics which show us the dominance of Ethereum over the DeFi market, that is why I decided to mainly focus on this blockchain for the rest of this thesis.

4. Decentralized Applications

In this chapter we are going to see the structure of a Decentralized application and what is needed to build them.

4.1. Their aspect

As stated before, smart contracts started a new way of using cryptocurrencies thanks to the code inside them and their functions.
But, to directly use a smart contract, every user would be required to know how to manipulate read the code, understand each function withing, write their own code which then would interact with the smart contract and execute its functionality.
This would therefore limit the number of possible users to those who already have a coding background and/or have time and are willing to learn it.

To solve this problem an interface needs to be coded, giving users a better User Experience by making interacting with smart contracts seem as like they are using an everyday website and not throwing them off by having to change their habits.

Therefore, as we will see more in depth later, the frontend part of a decentralized application is like any regular one even though it connects to the blockchain.

4.2. Oracles

4.2.1. Definition

Decentralized applications are fully disconnected from the outside world, and the problem with that is that if they receive information from the outside the blockchain scope, they can't verify it and know whether it is true or not.

DApps can only verify everything that is in the blockchain, they can for example verify which wallet has a certain amount of Ethers in it. But they can't verify what is the outside weather for the day, since that information requires human interaction by having someone input the weather in the blockchain in a smart contract variable or something similar.
Therefore, there needs to be a reliable source giving DApps the information they need.

And that is where 'Oracles', which were named after the Oracles of ancient Greece, who was a person that was able to be an intermediary between the humans and gods, and by being so, they could give answers from the gods to questions that human might ask themselves about the future or other. (Merriam-Webster, 2022).

The oracle in the blockchain acts the same way, connecting the 'blockchain world' to the outside world, our everyday world. If an information is needed within the blockchain, the oracle can give it and make it 'verified' within the blockchain. (Berné, 2018).

4.2.2. Chainlink

Having a single centralized oracle would be contrary to blockchain purpose of being decentralized, creating a point of weakness in this aspect.
If this single oracle spreads wrong data, either due to a bug or maybe malicious intentions, then how can we be sure the data received by the blockchain is trustworthy?

Chainlink, an oracle provider solution for blockchains, solves these problems.

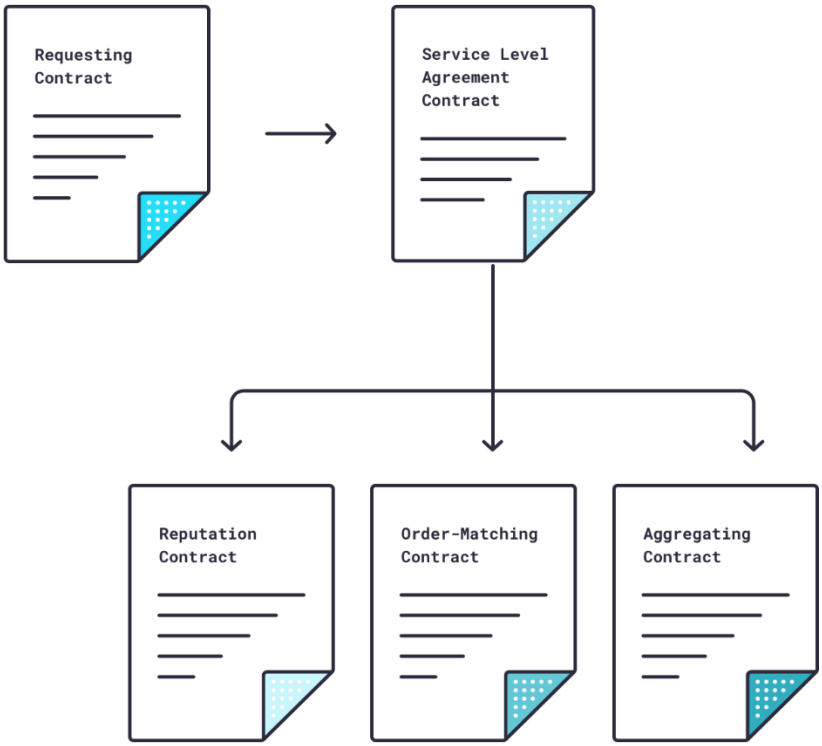

Figure 6: Chainlink Diagram, (Cryptopedia Staff, 2022).

First, it needs to receive a request of data by a smart contract withing the blockchain.
Once it gets it, a 'Requesting Contract', which is a smart contract made by Chainlink to make requests, sends out the demand to a 'Service Level Agreement (SLA) Contract'. (Cryptopedia Staff, 2022).

The SLA creates three other contracts which are:

- Chainlink Reputation Contract
- Chainlink Order Matching Contract
- Chainlink Aggregating Contract

Chainlink data feeds are made of networks of independent, highly available and vastly geographically placed node operators, allowing high reliability, and the request is sent to them by the Requesting Contract.
Then, the Reputation Contract verifies that an oracle provider has a good track record of authenticity and performance, if not, the node is discarded.

Later, the Order Matching one selects the correct number and types of nodes to answer the request of the Requesting Contract.

Finally, the Aggregating Contract takes the received data from the chosen oracles and either approves them or reconciles them to have a correct answer.
It does so by comparing all the data received from each node, and if some of them deliver the

data from a wrong source, it will recognize them as faulty and discard them, by doing so it validates the data from a single source.
Then, if the data received isn't always the same even after these first steps of verification, if it can be done, the Aggregating Contract does an average between the set of data and validates it to the blockchain. (Cryptopedia Staff, 2022).

The data is now in the blockchain and is made publicly available, immutable, and verifiable by multiple third-party interfaces.

By doing so, the blockchain can have exterior data that is coded in it and accessible by everyone so it can be used for other decentralized applications. Thus, the use cases of the blockchain can be expanded to outside needs and isn't only confined within it.

4.3. The storage of data

When building a Web Application, one of the main problems we might encounter is the storage of the data and creation of our database.

What size of database do is needed? How much bandwidth will be used? Are we going to exponentially need to scale or is the amount of data going to have a certain limit? Is It better to keep the data stored on-premises or to have a Cloud provider?

There is no need to ask ourselves all of these questions when building a decentralized application. Our storage is the blockchain, all the data we will interact with, create, and need, will be always available in the blockchain as long as the nodes running the blockchain are available. (Ethereum.org, 2022b).

So, the backend part of a decentralized consists of making smart contracts that interact with the data in the blockchain and connects it to the frontend.

5. Decentralized Finance use cases

Over the course of the years, the use case for Decentralized Finance has grown a lot.
First, we only had bitcoin, which could be used to transfer money to someone all over the world with only a couple of clicks. It was also seen as an inflation hedge due to its monetary politics implemented within its code.
Then, the blockchain started to be seen as a way to follow up the delivery chain, or supply chain, of a product. Some big companies like Carrefour in France started using the technology. (Carrefour, 2019).

And nowadays, mostly since then end of 2020 and throughout 2021, with the use of smart contracts, a lot of use cases have started, creating an almost fully functioning Decentralized Finance aside from our current one.

In this chapter we will talk about the major use cases of DeFi and what is different from the way they might already be implemented without the use of DeFi.

5.1. Lending

5.1.1. Traditional lending

Lending as we always knew, was done by going to a bank or a financial institution.
A borrower could go to a the third-party for different reasons, like starting a business, buying a home, buying a car, or even taking a small loan to go on vacation or shopping.
The bank would then, if the different conditions are met, such as having a good credit score for US households or staying under 35% of debt ratio for French households, give the money to the borrower. (Direction de l'information légale et administrative, 2022).

And by doing so, a debt in the name of the borrower would be created, stating that the borrower needs to repay a certain amount to the bank each month, plus certain fees such as interests.
If the borrower defaults and isn't paying his debts, the bank can take legal actions and the borrower would see his assets being seized to cover as much of his debt as possible.

And that is the security that the banks have, amongst other probabilities of default which are calculated in numerous ways, when giving uncollateralized loans to different borrowers.

5.1.2. DeFi lending

Now, the problem with DeFi is that the person taking a loan isn't going to be identifiable, since it is going to be a wallet address without any information attached to it. So, there is no security of being able to force him to pay back the loan and not run away after taking the money.

That is why, for now, in decentralized finance, the only loans available are overcollateralized loans.
Overcollateralized loans, are loans where the borrower needs to give the lender a certain amount of funds and he can borrow up to a certain percentage of the given funds depending on what type of assets he gave. (Kenton, 2020).

For the purpose of introducing the next part, I am going to explain what a stablecoin is.
A stablecoin is simply a cryptocurrency which price is correlated to a FIAT currency and therefore if one wants, they can redeem an amount of FIAT that correspond to the same amount of stablecoins he has and vice-versa. (Ichbiah, s.a.).
The USDT is a stablecoin indexed to the American dollar. USDT stand for USD Tether, Tether being the company which created the cryptocurrency. (Ichbiah, s.a.).

Getting back to our loans, for example, if a borrower gives a lender 10'000 USDT, he could borrow up to 97% less the interest fees so the lender can take his cut. (Aave, 2022a).

But, if the borrower gives Bitcoin as collateral, the borrower could only have up to 70% of his funds, and that is because bitcoin price is volatile, therefore the lender needs a bigger margin to make sure he can close the debt while still being in profit if Bitcoin's price falls quickly. (Aave, 2022b).

5.1.3. The mechanism behind it

In traditional loans, the loan is done through a bank.
But in DeFi, there is no intermediate. Both the borrower and the lender only interact directly with the smart contract. Smart contracts have the possibility of keeping funds in them since they have their own address and can function as a wallet too.

Let's take Aave, the biggest lending platforms in DeFi. (DeFi Llama, 2022c).
They have built smart contracts that allow you to:

- Deposit your money, becoming a lender and receiving a certain annual percentage yield (APY) that depends on how much demand there is for the deposited currency. On Aave you can only deposit cryptocurrencies, but other DeFi protocols may allow you to directly deposit FIAT if you want to avoid having exposure to cryptocurrencies while having the benefit of using decentralized finance. (Lutz & Benson, 2022).
- Lend money by, first depositing some collateral. Once you have deposited it, you can then borrow up to the specified percentage allowed of the collateral you have. Remember, the less volatile the price of the collateral, the more you are allowed to borrow according to your funds.

These are the 2 main contracts that are available on Aave, allowing users to have a fully decentralized experience, lending and borrowing from each other without the need for a third-party authorizing them to do so. (Lutz & Benson, 2022).

Aave smart contracts also have built-in liquidation methods that activates when certain criteria are met. (Aave, 2021a).
For example, if you have deposited bitcoin as a collateral, and it is stated that the liquidation threshold is 80% and the maximum available to borrow is 75% of the funds you deposited.
Therefore, if you take a loan of 75% of your funds, then if bitcoin's price falls by 20% your position will be liquidated by liquidators that are monitoring every position. Meaning that your loan will be closed, and the collateral seized by the smart contract.

The 5% difference between the two prices above is the 'liquidation penalty', it is used as a bonus for the liquidators that monitor all the loans to prevent any default and keep the balance sheet of Aave positive. (Lutz & Benson, 2022).

5.1.4. Advantages

Decentralized lending has different advantages against centralized one.
As stated before, one of them is the lack of centralized control by a bank that decides whether we can or not borrow some money.
Another advantage is the fact that rates aren't controlled by a central authority, which adds to the decentralization factor but also resolves some inefficiencies. In Aave, the rates are determined algorithmically and optimized to be sustainable and reduce slippage for borrowers to have lower rates but also for lenders too. (Aave, 2022c).
A third advantage would be the fact that one could clearly see the health factor of its open borrows and monitor them since the collateralization ratios are visible in Aave but also on-chain, meaning in the blockchain. (Aave, 2021b).

5.1.5. Lack of options

In conclusion, DeFi lending as for now, is a great tool if someone already has funds and wants to take a loan using it as collateral.
But, for bigger loans as mortgages or buying a car, where most of the time the loan is uncollateralized, there is no real option other than proceeding to do a KYC (Know Your Customer), the identity verification protocol, of the client and acting as a regular bank by doing background checks and estimating the default of payment risks. (Thales, s.a.).

A lot of researchers and developers are currently studying this matter, and probably one day, some sort of decentralized KYC will be available and uncollateralized loans will work as easily as collateralized ones are working today.

5.2. Exchanges

Another use case for decentralized finance is Exchange platforms.
Decentralized platforms have a better purpose for users looking to trade various currencies

5.2.1. Centralized exchanges

Before talking about Exchanges specialized for trading purposes, let's start with a basic form of exchange, the one needed when someone wants to swap between two currencies.

He would either need to go through an exchange office, most of the time a physical place, or sometimes his bank might do it, so he could do it online but with a higher spread which means the bank takes a bigger cut then the exchange office. (Probasco, 2021).

These two ways of exchanging between currencies already show us some inefficiencies in the way it is done.

Now, let's concentrate on financial exchanges aiming at amateur or professional traders.
If one would like to start trading between currencies, he must first create an account on an aggregated exchange, then he must verify his identity so the broker can comply with regulation standards. (Ziyanurov, 2020).
Only then he is allowed to deposit, withdraw, and trade a certain limit of money.

This process takes time and is constraining. If a person wants to trade anonymously without giving his identity, it is impossible to do so on any exchange that wants to fully comply with laws.

5.2.2. Decentralized Exchanges

Decentralized exchanges (DEX) solve most of the problems a user may encounter with centralized ones.
In this section we will mainly focus on Uniswap, the biggest decentralized exchange by daily volume and market share.

Just like Aave, Uniswap works completely with smart contracts and needs no third-party for a user that wants to use it.

First, the liquidity of each cryptocurrency pair, which is needed for users to swap from a cryptocurrency to another, is completely given by users that deposit it directly on a smart contract.
Everyone can become a liquidity provider for a pair A-B of cryptos by simply depositing a certain amount of crypto A and crypto B in the liquidity pool. (Uniswap, s.a.).
Liquidity providers are rewarded a certain percentage which comes from the transaction fees. So, the more volume there is, the better their revenue, allowing everyone to gain interest by providing liquidity. (Uniswap, s.a.).
This type of liquidity providing and return from it is mostly reserved to large institution in traditional exchanges.

Then, as opposed to centralized exchanges, any pair of cryptocurrencies can be added without the need of having the permission from Uniswap or a third-party, so if a project has its own crypto, they can directly implement it in the biggest exchange available if they want to, which gives them a bigger user base that could possibly use their native cryptocurrency. (Uniswap, s.a.).

Also, Uniswap has an integrated AMM (automated market maker), which is an algorithm coded to allow users to trade whenever they want to.
Let's say someone wants to exchange some bitcoin for Ethereum, to keep the liquidity pool stable and not moving the invariant, the balance between the two assets, too much which keeps the price as stable as possible even if there isn't a lot of liquidity. (Uniswap, s.a.).

Another useful feature of Uniswap is the fact that it can easily be implemented as a feature on a website since it is open source. All you need is to get the front-end code and add it to your website or customize it to have it blend better with your style, and then connect to the smart contracts already created by Uniswap. (Uniswap, 2022).
Now you have a decentralized exchange on your website and full access to Uniswap AMM and liquidity pools without the need to do everything by yourself.

Another problem with centralize exchanges is the lack of transparency about their balance and liquidity pools. With a DEX you don't have this problem since everything can be seen on the smart contracts and is on-chain.
This gives users more security about where their funds are and how they are used.

5.3. Tokenization

As we seen before, Oracles give the blockchain the possibility of receiving information from the outside and have in transcribed under its language.
This allows us to have outside elements, stocks, or everything that can have or has a trading market, under tokens that are on the blockchain.

5.3.1. Definition

Tokenization is the procedure of taking assets that can be on and off chain, and then:

- Representing it directly in the blockchain with the possibility of owning fractionalized parts of it.
- Creating a composite of a list of underlying tokens, which allows you to buy them all together without having to buy them one by one. (Blockchain France, 2018).

The tokenization can be done differently depending on the blockchain and the type of token that we want to create.

Our main example is going to be ERC-20 (Ethereum Request for Comment 20), which is the standard way for creating fungible tokens in the Ethereum blockchain.
This interface allows tokens to have a common interface and defines how users can interact with them. (Reiff, 2022a).
Therefore, the standard ERC-20 makes it easier to implement a token on a DApp because the main function within it that are used for sending or receiving should be the same as any other and have the same name.

5.3.2. Examples

Now, let's talk about Mirror protocol and how this decentralized application uses tokenization.
Mirror protocol is a decentralized application that creates synthetic assets of big enterprises stocks traded in financial markets. (Mirror, s.a.).
Even though it recently went through huge problems, due to the blockchain 'Terra' in which they were implemented collapsing, it is still a great example of how tokenization can be used so I chose them as our use case. (Davies, 2022).

It does so by using oracles, which work the same way we've described previously, retrieving the price in real time from trustful sources and once it is received, the price is updated constantly in their application.

In most of centralized brokers, it can be hard for a user to transfer their equities from their broker to another one if they want to change. This problem is caused by the fact that these brokers don't have a common interface or platform for the way the stock actions are created and programmed into code. (The Investopedia Team, 2022).
But with the fact that on a blockchain all tokens follow the underlying ERC-20 interface, decentralized brokers always have a full compatibility between their assets.
This makes it easier for users to transfer funds from a platform to another to get better offers and go to the one with less associated costs.
Also, having to buy a full action and not a fraction of it may stop someone from investing if he has a small capital since some stocks are highly priced and cost thousands of dollars. So always proposing fractionalized shares is better in the interest of new investors that have small amounts of money to use. (Gravier, 2022).

5.4. Non-Fungible Tokens

As of now, we talked about cryptocurrencies and the ERC-20 standard, which are fungible tokens, meaning that they can be fractionalized and have decimal units.
These fungible tokens are used as monetary tokens, but there is another type of token that covers other specificities, and it is non-fungible tokens (NFT).

5.4.1. Definition

A non-fungible token is a token that is unique from any other one and has its own numeric identity.

This means that each token is itself a unit and doesn't correspond to a numeric value.
It can't be fractionalized, you either own the full token or not, but not a part of it.
As we've seen before, on Ethereum, fungible tokens follow the ERC-20 standard. NFTs follow the ERC-721 standard, creating an interface that makes all NFTs compatible with every DApp that implements them. (JournalDuNet, 2021).

5.4.2. Purpose

Fungible tokens are for numerical or monetary value, NFTs are for numeric identity of something or someone.

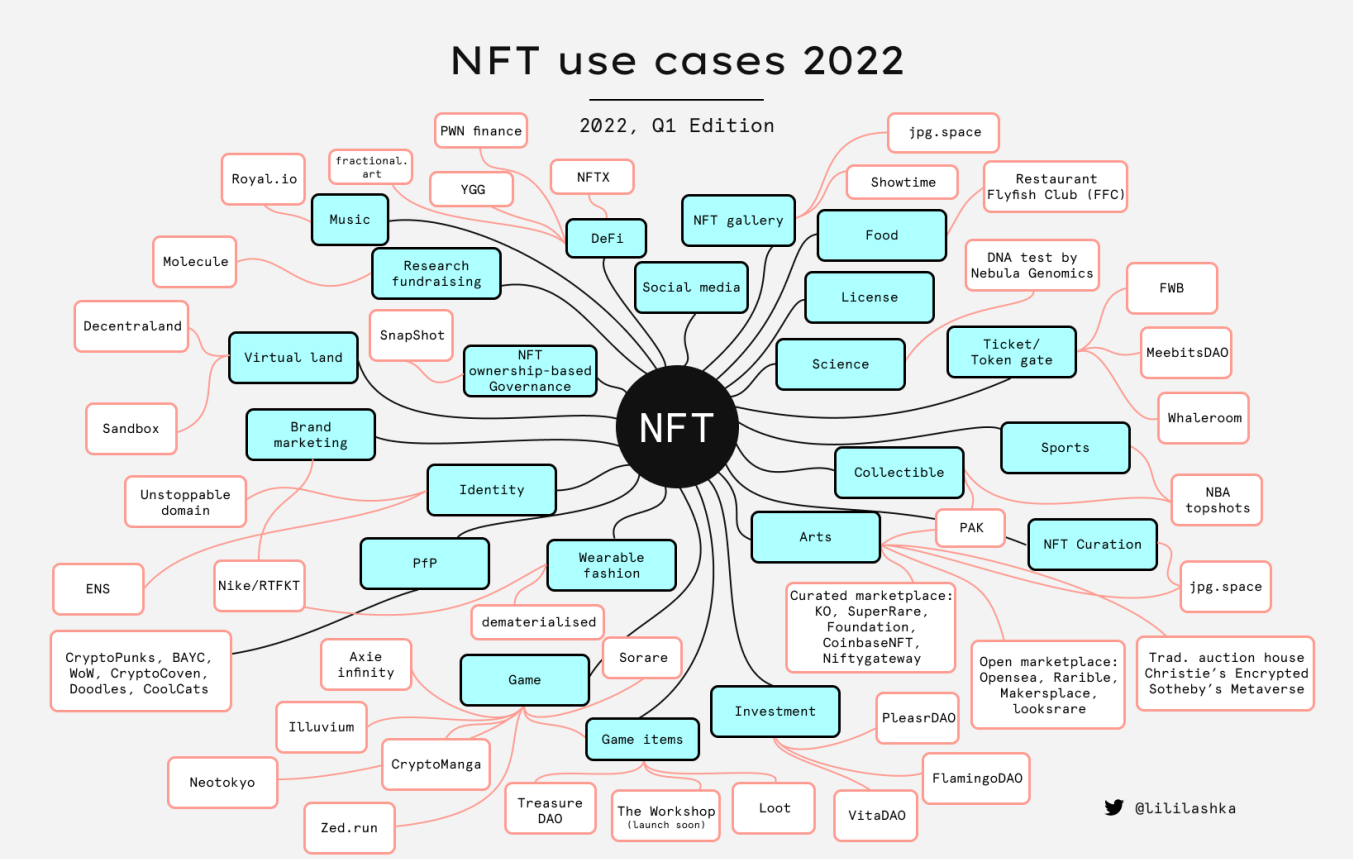

Figure 7: NFT use cases 2022 (Tweet @lililashka, 2022)

As we can see above, already today the use cases of NFT are wide and varied across multiple things. From using them as profile pictures on social media to feel like you belong to a group, using them to own virtual assets, to collecting sport cards online and creating your own team.

We are now going to see some of the biggest examples of NFTs:

- **Identity**: you can use an NFT containing an ENS domain, which works like a web domain but instead of linking a website, it can be linked to a wallet. That way your wallet would have an easier name instead of a cryptographic series of random hexadecimals. This would allow you to share your wallet way easier when needed, like if someone needs to send you money on it. (ENS Domains, 2022).

- **Games**: nowadays there are multiple games with a financial world built within them and with their own currency, players can buy unlockable content with the currency which then improves their gaming experience and the can enjoy the game. (Glover, 2022).

  So, each game has its own currency and their own contents, they can't be used from one game to another. The objects bought by players aren't really owned by them, they are to the game and in their account as long as the game is still playable, but if the company decides to stop the game, the content disappear.

  With NFTs, every object could become compatible from a game to another, players could easily use the object bought for a game on another one without having to spend money again. Also, if the company who created the game decides to stop it, the NFT is still owned by the player so he will continue to have it and it could even be used as a tribute to the game that existed once, and he played. (Glover, 2022).

- **VIP passes**: for investors, sometimes there are online private groups with either professionals in it teaching their subscribers or maybe traders giving financial advice, and to access these groups you need to pay, either a monthly subscription or a one-time payment that is large enough to attract only serious inquiries. (KaizenDAO, 2022).

  Before you had to do the payment with your credit card and find a way to prove that you indeed, did the payment. Also, a subscription was linked to the one that asked for it and himself only.

  By introducing NFTs, the process is simplified. To subscribe you only need to buy the NFT created by the private group, once you own it you can link your account to a software that can verify that you indeed have it in your wallet and give you the access to the group.

  The NFTs are made in a fixed supply by the group owners, allowing only those who managed to buy one in time to access it.
  But, if someone doesn't want the subscription anymore for whatever reason, he can sell the NFT on a marketplace. Therefore, his subscription is then automatically ended, and he is kicked out of the group by the verifying software.
  On the other hand, the one who bought goes through the verification process and gains access to it. This allows users to swap and end their subscriptions more easily. (KaizenDAO, 2022).

5.4.3. Possible future functionalities

In this part, I am going to talk about use cases and functionalities of NFTs that haven't yet been applied because of legal constraints that would make them void, or financial ones that keep companies from trying to implement them, or other reasons that could potentially be solved in future, making NFTs being even more relevant in every day's life.

- **Tickets**: another use case, would be having a ticket as an NFT. Let's take the example of a football match. Supporters go on the website, buy the ticket, and receive a NFT on their wallet. (Binance, 2022).

  On the game day, they go to the stadium with their wallet connected on their phone, upon arrival they show their NFT. The person at the entrance then sends a transaction to that wallet asking them to sign a message with it, to prove that they indeed own the wallet.
  Once the message is signed, the supporter can enter.
  To prevent supporters from using twice the same NFT, it can have a simple Boolean attribute named "used" and once it was used to enter, at the same time the supporter signs the transaction proving he has he wallet, the value of the attribute goes from "false" to "true". And that NFT now can't be reused by someone else to enter. (Sacristán, 2022).

- **Proof Of Ownership**: nowadays, to prove that you detain something you either need a receipt if it's an inexpensive object, or a certificate if it's something rare, or maybe even to go through a notary if you want to buy a house so he can attest that the house is now indeed to your name. (Sonenreich, 2022).

  If we were to introduce NFTs to these proof of ownership cases, for most of these, we could eliminate the fact of having to go through a third-party, such as the notary. And make the process administratively easier and with less paperwork since it would be digitalized.

  A token could be used for the ownership of a house, so the one who owns the token owns the house.
  Therefore, if someone needs to sell a house, he does so by simply exchanging the NFT against the requested amount of money for buying the house. (Sonenreich, 2022)

  This transaction could be done without third-party, by creating a smart contract that asks the owner to put the NFT in it and the buyer to put the money in it.
  Once both parties made their part and the right amount of money is put in, the smart contract automatically sends the NFT to the new owner/buyer and the money to the seller.

  The home now has a new owner. And everything was done without an outsider and within 4 transactions in the blockchain.

  This is one of the use cases for proof of ownership but there could be many others, such as:
  - NFT serving as a certificate of authenticity for a unique product with a QR code on it linked to it
  - An NFT that serves as the warranty for a product by having a QR code on the product and on the NFT the date of buy, the time under warranty, and maybe something stating what is covered by it.

  For the proof of ownership of a house, there already is a company that allows you to do so, named RealT. It does so by selling houses dividing in a certain number of shares, like a company would do, and creating an NFT that corresponds to each of these parts.
  Investors can then buy the amount of NFTs they want which has a fixed price, making them partially owners of the house. (RealT, 2021).
  RealT then maintains and rents the house, investors don't need to do anything, and they gain a share of the rent each month that corresponds to their part of the total shares.
  It is way easier to do it this way than the common way of having to go through a notary or

even having to buy an entire house if you don't have the funds. (RealT, 2021).

These possibilities are just the top of the iceberg to what NFTs could serve as, but they already solve a lot of inefficiencies that lie within our currently used methods.

6. Decentralization Risks

Until now, I stated multiple benefits of having a Decentralized Finance with application running by themselves with algorithms and smart contracts.
All these benefits can indeed improve the processes we use and make things easier to execute.

But as all new things, all of this comes with new risks, risks that might not have existed up until now and to which we never searched a solution.
This means that along trying to make DeFi become better, developers will also have to make it become secure, or people might continue losing their funds to these vulnerabilities.

6.1. Smart Contracts

Smart Contracts can themselves be a risk for the users.
If one does not know how to code them or even at least read them, he could interact with a harmful contract.
The user wouldn't be able to understand what the contract does when a certain function is executed and how it will affect their wallet and funds in it.

Nowadays phishing scams are recurrent all-over social media, indicating that there is a new project coming out and that it is a great catch investing in it and linking to a website. (Kumar, 2022).
Of course, they make it look like they're from established teams in the industry by having fake accounts with a lot of followers and even sometimes verified by the social media itself.
Once the user clicks on this link, he is then asked to connect his wallet to interact with the website and invest in the project.
But, if the user signs the transaction to connect his wallet, the hacker gets access to all his funds and starts draining it by sending everything to another wallet that he has. (Kumar, 2022).

And since the blockchain is immutable, there is no way to revert the transaction to return the funds to the person who got scammed.
Therefore, a crucial part of DeFi is to warn new users about the dangers of signing transactions of unknown contracts and always verify that the website they are in is the correct one and not a reproduction.

6.2. Oracles

We have seen before that Oracles allow us to import data from off-chain, the outside world, to on-chain, which is everything that lies within the blockchain.
But since a lot of decentralized application rely on them for pricing various things, this can create a lot of problems if at any moment the oracle has a problem and displays wrong data.

And that is what happened once to Mirror Protocol. The oracle had a problem with the price of the cryptocurrency LUNC, and it said it was valued at 10$.
This price was indeed wrong and corresponded to another crypto named LUNA. The real value of LUNC was 0.0001$, which is a huge difference from 10$. (Certik, 2022).

The oracle pricing error allowed attackers to take loans by giving LUNC as a collateral and with their collateral being overvalued.
One of the attackers took a loan valued at 300'000$. He used a collateral of 100'000 LUNC

tokens, valued at 10$ each by the oracle, so the total collateral value was of 1’000’000$. But since the real price of a LUNC token was 0.0001$, the attacker only needed 10$ to buy his LUNC tokens.
In the end, due to this oracle error, Mirror Protocol had an exploit totaling 2 million dollars in losses. (Certik, 2022).

This shows us the vulnerability of relying on oracles and the importance to mitigate risks by comparing multiple oracles or using other solutions to reduce risk.

6.3. Custodial

One of the main reasons behind decentralized finance and the blockchain is to have the property over our own funds.
“Not your key, not your coins” as a decentralization fervent would say. (Ledger, 2020).

The reason behind this way of thinking is that if the institution where you decided to put your money goes bankrupt or if there is a bank run and everyone tries to cash out at the same time, then you might not get your money back, or only up to a certain amount of it that is covered by insurances. (CFI Team, 2021).

Therefore, what better way to make sure your funds will always be available than keeping them yourself?
Well, that is indeed a great idea since you could have your funds with you no matter where you go, and they would always be completely available if you need them.

But this comes with multiple risks.

Self-custody means taking full responsibility of your funds. If they get stolen, lost, hacked or even if you sent them to a wrong address by mistake, it is your own fault, and you have no one else to blame for it. Your funds will most likely never be retrieved, and you will have to do a complaint with the police and hope they find who stole them or to whom the address to which you mistakenly sent them belongs to. (Keller & Hofstetter, s.a.).

Also, when creating a wallet you have a recovery phrase of multiple words, around 13, and if you ever lose access to your wallet you need to use this phrase to regain it.
Therefore, another thing to consider when keeping your own funds is storing this recovery phrase.
If you store it on your computer and you get hacked, then the hacker has a complete control over your wallet. (Trust Wallet, s.a.).
When storing it written somewhere, someone might see it and take advantage of it.
Therefore, storing the recovery phrase is already a risk.

These are all risks you need to consider if you want to have a complete self-custody of your funds.

6.4. Scaling transactions

A major problem that stops cryptocurrencies and decentralized finance from going mainstream and attracting more users is their relatively low number of transactions per second (TPS).

Ethereum has around 15 TPS, which is way too low for a main way of transacting money. (ETHTPS, 2022).
Bitcoin has a up to 7 TPS, it is even worse than Ethereum, proving it can't be a way of payment. But recently, developers are working on a 'Lightning Network' feature that is already allowing up to 500-1'000 TPS which is already way better. (BitPanda, 2022).

But when comparing it to a method of payment like VISA that claims to have up to 24'000 TPS theoretically and a proven average TPS of 1'700. (Sedgwick, 2018). The average TPS is still higher than those of the main blockchains.

Therefore, if major blockchains cannot manage to improve their number of transactions per second they will never be usable by everyone daily because there would be a long processing time for each transaction.

6.5. Environmental

Some blockchains may have environmental problems in the way they are run.
This is mostly true in Proof-of-Work blockchains, because they need large computational powers to solve their hash to find the next block and validate transactions. (Reiff, 2022b).

Therefore, to solve the hash, miners use a lot of powerful GPUs (Graphics Processing Unit) that overheat a lot when doing so and have a reduced lifespan compared to a normal use. (Blockbase, s.a.).

For major blockchain such as Bitcoin, this problem has been solved by using renewable energies to run the GPU's and everything associated with them such as cooling methods. (Newar, 2022).
But smaller blockchain may not have this option since their miners might not be as established and big as Bitcoin's ones.

7. How to code a Decentralized Application

In this chapter we are going to see how Ethereum DApps are built, from smart contracts being used as backend, to frontends built in JavaScript while being connected to the blockchain.
Then, I will try to cover how someone could learn to program decentralized applications and what would be their path, from beginner to expert, from doing simple smart contracts to developing a renowned DApp that could benefit others and the developer himself.

7.1. Programming languages

First, we will go through the programming languages needed to build a complete Decentralized Application with a working User Interface (UI) and connected to the blockchain.

7.1.1. Solidity

Solidity is a programming language which is object oriented and was made to code smart contracts.

The main blockchain that uses it is of course Ethereum.
But it is also used on other blockchains, such as the Binance Smart Chain (BSC), Ethereum Classic (ETC), or even Avalanche C-Chain, which are also some of the biggest blockchains available after Ethereum. (Wikipedia, 2022e).

It was designed in 2014 by Gavin Wood, making it a relatively new language. Then the Ethereum team took over and developed it into the fully functional programming language that we know today. (Wikipedia, 2022e).

Even though it is quite new, Solidity comes with new flaws because smart contracts are immutable once deployed. Therefore, the developer must be very careful in the testing phase and needs to pen test his own application as much as possible.

7.1.2. JavaScript

JavaScript (JS) is a script programming language created in 1995, but it was only in 1996 that it was fully functional with a working JavaScript Engine. (Wikipedia, 2022h). It is used for web programming, mostly combined with HTML and CSS.

To have a common ground between developers using the language, ECMAScript was created. It is done by Ecma International, based in Geneva Switzerland.
It introduced a standard to have as much interoperability as possible between web pages and different web browsers. (Wikipedia, 2022g).

It is the most used language for websites and almost any web browser has a built-in JavaScript Engine to execute it. (Wikipedia, 2022f).

The functionalities stated before are all front-end ones, but JS can also be used as a backend language for web servers. This can be done with the use of NodeJS.

7.1.3. Typescript

As stated on their website, TypeScript is JavaScript with syntax for types. (TypeScript, 2022). It was created in 2012 in 2012 by Microsoft with 2 years of development. (Wikipedia, 2022h).

Since JavaScript is a typeless programming language, which some would argue that it is a bad thing since it introduces more error possibilities for developers, and it is more difficult to find them even after executing their code.
Therefore, TypeScript if a strongly typed programming language which is built on top of JS and follows its standards. (TypeScript, 2022).

7.2. Frameworks

To make coding easier and faster, developers never code without the use of frameworks.
Frameworks are made of components which are already coded and can be reused in a new project. This makes it easier for developers to focus on specificities of their own work and their constraints. (Wikipedia, 2022i).

We are going to see some of the frameworks that are mainly used when developing Decentralized Applications.

7.2.1. React

React is a JavaScript framework developed by Facebook since 2013. It is used to create interactive User Interfaces easier and faster, by using components that are already coded and can be modified to have a better integration to any application someone is doing. (Wikipedia, 2022j).

Its main purpose it for single pages applications but by making it look like there are multiple ones. It does so by rendering certain parts of the code and therefore the user might think he is on a new page when in fact the previous code has been hidden and the new one is shown.
This allows for faster rendering web applications. (Wikipedia, 2022j).

7.2.2. Web3.js

Web3.js is a collection of libraries that allow web application to connect to local or remote Ethereum nodes. This is done through HTTP, IPC or WebSocket. (web3.js, 2016).

Therefore, Web3.js is used when a web application, which is outside of a blockchain, needs to interact with a smart contract, that lies within a blockchain.
To do so, web3.js uses libraries that are coded in JavaScript but allows us to interact with Solidity written contract by translating our frontend requests. (web3.js, 2016).

When using Web3.js a provider is needed, this is what will connect you to a certain local or remote node.
There are already built-in providers for multiple blockchains, such as the ‘Eth.givenProvider’ for the Ethereum blockchain. This allows you to interact with smart contracts within the Ethereum blockchain only. (web3.js, 2016).

7.2.3. Truffle

Truffle is a suite to help developers by making the creation and deployment of a smart contract easier. The suite has three parts, Truffle, Ganache and Drizzle. (Truffle Suite, s.a. (a)).

Truffle is a development framework for Ethereum.
It is used for smart contract lifecycle management, automated contract testing, scriptable deployment and migrations, a simpler network management, interactive consoles, or even external script runners. (Truffle Suite, s.a. (b)).
The most important one would be the automated contract testing because all smart contracts are immutable. So, having a complete contract testing tool, which tests your contracts with known blockchain exploits, and your frontend directly through JavaScript pen tests, is one of the most needed tools.

Ganache is a personal blockchain that can be used for Ethereum development. It can be used to deploy contracts, develop DApps, and run tests (Moralis, 2021).
Since deploying and developing directly on the Ethereum blockchain would cost you money, having a personal blockchain is a must-have for development purposes.

Drizzle is like Web3.js, it is also a collection of libraries that help you with your DApps frontend. The main difference is that it is based on a JavaScript Redux store. (Truffle Suite, s.a. (c)).

7.3. Development Environment

Over the years, Solidity has seen some new IDEs arrive. From, Remix to now even VS Code integrating it.
We'll see the main two that are used for Solidity development and their implementations.

7.3.1. Remix

Remix IDE is a development environment both available online and as a desktop application. Remix allows users to verify contracts, compile them, deploy the contract on different testing blockchains, generate testing files, and even send transactions to test the functions within your contracts to see if they do what we want them to and if they are secured. (Remix, 2022)

Remix is easy to use and user friendly, even for someone starting it should be a good learning experience.

7.3.2. Visual Studio Code

Being the most used IDE, Visual Studio Code (VS Code) is a must have for any experienced developer. Therefore, if someone has already used it, he would possibly prefer to continue using it. (Carbonnelle, 2022).

To do so, there is the possibility to add the Solidity language to VS Code, allowing programmers to stay in a known IDE.
It allows you to have built-in snippets, compile contracts, have code completion, a default project structure, code generation, linting, and other useful functionalities.
Therefore, once someone has some experience, VS Code would be the best IDE to use and the more complete one. (Blanco, 2022).

VS Code also has an 'Ethereum Remix' extension, allowing users to combine both Remix and VS Code. This is a good transition for users that have learned with Remix and are looking to move to VS Code. The transition would then be smoother. (Remix Project, 2021).

7.4. Testnet

Another useful tool for developers is a Testnet.
Because if you want to develop and deploy something directly on the Ethereum blockchain, you will need to pay ‘gas’ fees, which are paid in Ether, the native currency of the blockchain. Those fees are paid in each transaction in the blockchain. (Coen, 2020).
So, you would need to pay them when deploying or even sending a test transaction. And as stated before, smart contracts being immutable, you would need to pay each time you want to release a new version of your contract when you are still in the testing phase, and you make changes to it. (Coen, 2020).
This could cost you hundreds or even thousands of dollars before even finishing your application. (Lastname, 2021).

To avoid all these costs, developers created Testnets, which are blockchains where you can get Ethers for free and use them to pay gas fees. So, you can deploy as much contracts as needed and test your code for free. (Stakepool, 2020).

There are various options, the main used ones are Ganache, as we have seen before, or Ropsten, which was created by the Ethereum Foundation themselves in 2017.
Even though Ropsten was one of the most used testnets, it seems that it is now deprecated and will no longer receive updates from the Ethereum foundation. They themselves have advised developers to migrate their applications from Ropsten to newer testnets, such as Sepolia or Goerli. (Ethereum.org, 2022a).

7.5. Smart Contracts

First, we are going to see the structure of a smart contract and define the way it is built and the differences from usual coding, such as JavaScript or Python.

The smart contract below is a simple smart contract that I coded myself which creates a cryptocurrency with the possibility of creating new tokens if needed and sending tokens between wallets.
I managed to code it by learning Solidity on Udemy and recreating it later. (North, 2022).

All the following explanations of this code in section 7.5 are from what I have learnt throughout the course and what was explained in it,. Therefore, refer to the Udemy course source if more details are needed in case something seems unclear. (North, 2022).

```
pragma solidity ^0.8.4;

contract Coin {
    address public minter;
    mapping (address => uint) public balances;

    event Sent(address from, address to, uint amount);

    constructor() {
        minter = msg.sender;
    }

    function mint(address receiver, uint amount) public {
        require(msg.sender == minter);
        balances[receiver] += amount;
    }

    error InsufficientBalance(uint requested, uint available);

    function send(address receiver, uint amount) public {
        if (amount > balances[msg.sender])
            revert InsufficientBalance({
                requested: amount,
                available: balances[msg.sender]
            });

        balances[msg.sender] -= amount;
        balances[receiver] += amount;
        emit Sent(msg.sender, receiver, amount);
    }
}
```

We are now going to review the main differences that there is in Solidity programming from non-smart-contract oriented languages.

7.5.1. Version

```
pragma solidity ^0.8.4;
```

This defines the version of the file and which compilers can compile it.
The pragma activates certain functionalities or verifications from the compiler. It applies only to the current file and not any other file in the project. (Solidity, 2021).

As for the '^0.8.4', it states that this file can be compiled by compilers with the same version and only this version, nor any anterior nor any posterior versions.

7.5.2. Addresses type

```
address public minter;
```

Address is a type mostly proper to Solidity programming which does not exist in major languages such as Java or C#.

This type of data corresponds to a 20-byte value which links to a wallet on the blockchain. It is your identity on the blockchain and the value that tells who initiated a transaction or to whom it is sending something. (Amanwachi, 2022).

In the example above we can see that the address corresponds to the minter, which is the wallet that can create new tokens as he wants to.

7.5.3. Msg

```
        minter = msg.sender;
```

The 'msg' variable is a global variable. It has properties within it that can help you interact with the blockchain. (Crescenzi, 2018).
In the example above, we can see that the 'msg.sender' property is used, the sender always corresponds to the one that called a function, more specifically the address of the wallet that did so.

This is useful when a transaction is used to send money, because the next step would be to verify that the wallet has indeed enough tokens to send and isn't overspending.

7.5.4. Revert

Another useful function available in Solidity and often used is the 'Revert'.

```
        if (amount > balances[msg.sender])
            revert InsufficientBalance({
                requested: amount,
                available: balances[msg.sender]
            });
```

The code above corresponds to the verification that the address trying to send tokens has indeed enough tokens in the wallet.
We can see that if the amount that he wants to send exceeds that amount of tokens in his balance, then the 'Revert' function is called. (McKie, 2017).

When the ‘Revert’ function is called, two parameters are set within it:

- the requested, which is the amount the sender tried to send.
- the available, which is the actual balance that was withing the sender wallet.

The revert function is also useful because once the gas fees to execute the transaction have been paid, the rest of the funds that were sent are returned to the sender.

7.5.5. Built-in triggers

Solidity has some inherited member that act like triggers for the contract.

One of them is the ‘event’ member. (TutorialsPoint, s.a.).

```
    event Sent(address from, address to, uint amount);
```

The line above corresponds to the creation of a ‘Sent’ event that takes as parameters:

- An address indicating who sent it.
- An address stating to who it was sent.
- A uint indicating the amount that was sent.

```
        emit Sent(msg.sender, receiver, amount);
```

After an event is created, to use it you need to ‘emit’ it. The line above emits the ‘Sent’ event and gives it the needed parameters that were coded before.

The ‘Sent’ event will create a log that will be stored in the blockchain and will be always available by using the address of the contract. So, this is what allows each transaction to be stored in the contract.

7.6. Further programming expertise

Up until now, we have seen how a basic smart contract works and the basics to Solidity programming that differ from conventional programming languages.

In this part we will introduce some aspects that could be later be learned more in deep and that are needed to become an expert in this domain. (OffcierCia, 2022).

7.6.1. Distributed storage

In IT distributed data storages are file systems where the files are stored in more than one node, mostly replicating the files multiple times. (Wikipedia, 2022k).

A protocol used by decentralized applications to try and make their web pages decentralized is Arweave.
Arweave permits users to store data permanently and sustainably. (Arweave, 2022).
It does so by connecting users that have spare data space in their hard disks to people or enterprises that are looking to store data permanently. This is done withing a blockchain, like Bitcoin, and backed by a sustainable environment with the help of a single upfront fee paid by those looking for storage. (Arweave, 2022).

Therefore, if one really wants to build a fully decentralized application, frontend included and not only the blockchain backend, learning to implement distributed storage systems and using them should be one of the priorities.

7.6.2. Security

As said multiple times, smart contracts are immutable.
The security of a smart contract before deploying it should be a main point in the way it is developed.

Therefore, one should learn the best practices that should be applied when coding a smart contract and use tools that helps testing them.

One of these tools is OpenZeppelin, it has built-in smart contracts implementing the most used ERC standards and which have already been tested multiple times and deployed by other applications. They are available for Ethereum but also other blockchains. (OpenZeppelin, 2022).

Another aspect of security in the blockchain is simply being able to analyze transactions. Some can read a transaction and understand what happened and what was the purpose of it, but it is rare for someone to be able to track a wallet and make sure that it has no illegal uses or was not involved in a previous scam, exploit or even a hack. (CIA, 2022).

So, to avoid interacting with malicious smart contracts, a developer should be able to deeply do an analysis of existing transactions and wallets.
GraphSense is a tool that allows you to do so, by showing possible links between different wallets that have interacted with each other in a graph, this allows you to go back and see the wallet has no suspicious connections to it. (GraphSense, s.a.).

There are other security aspects that should be learned to be an expert in smart contracts security, such as the scope of an audit or knowing the most known attacks and how they happened.
So, to become a fully decentralized application expert one should dig deeper into the security field by starting with these points.

7.6.3. Blockchains types

Before, renowned decentralized applications were almost only developed on Ethereum, but with the expansion of other blockchains, some useful DApps are starting to emerge on other blockchains.

For that reason, if a developer wants to be versatile, after mastering the Ethereum blockchain since it is still the biggest, he should understand some others, at least to know the key differences and how they work.
And with some time, maybe even learn programming on them to expand its options.

Some starting points could be Sidechains, which take some of the load off a main blockchain if there is too many transactions coming in to later reinject them into the main blockchain. (Roth, 2022).

7.6.4. The future of Ethereum

To be up to date with the Ethereum blockchain and its possible future updates, just like being up to date with a programming language, a developer should be aware of possible future changes that are being developed by the Ethereum foundation.

One of these changes that is currently being developed is EIP1559, with the main goal being to change the current way transactions fees are calculated and done. (Beck & Asher, 2021).

Or Ethereum 2.0, which will be one of the major updates done to Ethereum and mainly consists of transitioning from a Proof-of-Work blockchain to a Proof-of-Stake, by doing so the number of transactions per second will be upgraded and the fees paid for each transaction decreased. (Millman, Graves & Kelly, 2022).

These are the main major upcoming changes on which a developer should monitor and that could affect its work and the specificities he would need to learn.

8. Conclusion

The goal of this thesis was to introduce Decentralized Finance and going into details about how it works.

The thesis stated current problems with the way of doing things in Centralized Finance, showing what needs to be improved for a person to have more benefits and a better user experience.

We have seen multiple use cases of DeFi, ranging from simply having a Decentralized Exchange for traders to have more freedom and less fees, to having Non-Fungible Tokens replacing paper certificates which need to be signed and approved by a legally allowed third-party.

But as we have seen, all these new ways of doing things, such as the introduction of smart contracts, come with newer risks that did not exist before and to which there are no verified solutions yet.

Therefore, the goal of the thesis in introducing DeFi to newcomers has been achieved. Even people that already know some of it but want to learn more could use this thesis as a starting point and they could then expand their research after.

As for the second part of the thesis, where a detailed programming part was written, introducing new concepts created by blockchains and their languages, the thesis is detailed enough for someone new to understand the basics.

This part also guides the reader on to outer scopes that he could learn to upgrade his programming skills in this field, which have not been included in the thesis due to the high difficulty level of learning them.

For that reason, this technical part of the thesis fulfills its main role.

9. References

Harvey, C. R., Ramachandran, A. & Santoro, J. 2021. DeFi and the future of finance. John Wiley & Sons P&T. USA.

Wikipedia, 2022a. “Roman currency”. URL: https://en.wikipedia.org/wiki/Roman_currency. Accessed 16 March 2022.

Wikipedia, 2022b. “Barter”. URL: https://en.wikipedia.org/wiki/Barter. Accessed 18 March 2022.

Shepherd, M. 2020. “Cash vs Credit Card Spending Statistics (2021)”. URL: https://www.fundera.com/resources/cash-vs-credit-card-spending-statistics. Accessed 2 April 2022.

Fis global, 2022. “How credit card processing works”. URL: https://www.fisglobal.com/en/insights/merchant-solutions-worldpay/article/how-credit-card-processing-works. Accessed 5 April 2022.

Corbae, D. & D'Erasmo, P. 2020. “Rising Bank Concentration”. URL: https://www.minneapolisfed.org/research/staff-reports/rising-bank-concentration. Accessed 30 April 2022.

ECB, 2016. “What are minimum reserve requirements”. URL: https://www.ecb.europa.eu/ecb/educational/explainers/tellme/html/minimum_reserve_req.en.html. Accessed 10 May 2022.

Godoy, P.H. 2022. “Classement des principales banques européennes en 2021, selon le total des actifs possédés”. URL: https://fr.statista.com/statistiques/882866/classement-banques-total-actifs-europe/. Accessed 5 April 2022.

Norrestad, F. 2021. “Largest banks in Finland in 2020, by total assets”. URL: https://www.statista.com/statistics/1057228/leading-banks-in-finland-by-total-assets/ Accessed 11 May 2022.

Norrestad, F. 2022. Number of banks in Germany from December 2008 to December 2021. URL: https://www.statista.com/statistics/350502/eurozone-germany-number-mfi-credit-institutions/. Accessed 13 March 2022.

Demirguc-Kunt, A., Klapper, L., Singer, D, Ansar, A. & Hess, J. 2018. The Unbanked. The global Findex Database. The World Bank Group, number 29510.

Rasure, E. 2022. Underbanked. URL: https://www.investopedia.com/terms/u/underbanked.asp. Accessed 15 March 2022.

Campbell, H. 2021. A brief overview of CeFi Problems. URL: https://www.youtube.com/watch?v=etlqmQTMXxA. Accessed 16 March 2022.

Williams, R. 2022. Credit Card Processing Fees and Costs. URL: https://www.valuepenguin.com/what-credit-card-processing-fees-costs. Accessed 18 March 2022.

Crédit Agricole, 2022. “Vos Tarifs au quotidien”. URL: https://www.credit-agricole.fr/content/dam/assetsca/cr881/npc/documents/tarifs/220802-TARIFS-2022-PARTICULIERS.pdf. Accessed 18 March 2022.

Macrotrends, 2022. “Federal Funds Rate – 62 Year Historical Chart”. URL: https://www.macrotrends.net/2015/fed-funds-rate-historical-chart. Accessed 10 March 2022.

IG s.a. “Crise des subprimes définition”. URL: https://www.ig.com/fr/glossaire-trading/crise-des-subprimes-definition#information-banner-dismiss. Accessed 25 March 2022.

Wikipedia, 2022c. “Financial crisis of 2007-2008”. URL: https://en.wikipedia.org/wiki/Financial_crisis_of_2007%E2%80%932008#Subprime_lending. Accessed 30 March 2022.

Lars, L. 2020. “La B-Money de Wei Dai : une préfiguration conceptuelle de Bitcoin ». URL: https://journalducoin.com/analyses/b-money-wei-dai-prefiguration-conceptuelle-de-bitcoin. Accessed 30 March 2022.

Nakamoto, S. 2008. “Bitcoin: A Peer-to-Peer Electronic Cash System”. URL: https://bitcoin.org/bitcoin.pdf. Accessed 30 March 2022.

Nakamoto, S. 2008. “Transactions in the Bitcoin blockchain”. URL: https://bitcoin.org/bitcoin.pdf. Accessed 30 March 2022.

Nakamoto, S. 2008. “Timestamp Hash included in a block”. URL: https://bitcoin.org/bitcoin.pdf. Accessed 30 March 2022.

Ledger, 2019. “Proof-of-Work mechanism”. URL: https://www.ledger.com/academy/blockchain/what-is-proof-of-work. Accessed 30 March 2022.

Nakamoto, S. 2008. “The privacy model of Bitcoin”. URL: https://bitcoin.org/bitcoin.pdf. Accessed 30 March 2022.

Coinbase, s.a. “What is ‘proof of work’ or ‘proof of stake’?”. URL: https://www.coinbase.com/fr/learn/crypto-basics/what-is-proof-of-work-or-proof-of-stake. Accessed 26 March 2022.

Algorand, s.a. “Algorand’s pure proof-of-stake approach”. URL: https://www.algorand.com/technology/pure-proof-of-stake. Accessed 20 May 2022.

Le Journal Du Coin, 2022. “Delegated Proof of Stake (DPoS) – Définition et explication”. URL: https://journalducoin.com/lexique/delegated-proof-of-stake-dpos/ . Accessed 20 May 2022.

Wikipedia, 2022d. “Smart Contract”. URL: https://en.wikipedia.org/wiki/Smart_contract. Accessed 20 May 2022.

DeFi Llama, 2022a. “DeFi Dashboard”. URL: https://defillama.com/. Accessed 20 May 2022.

Defi Llama, 2022b. “Total Value Locked All Chains”. URL: https://defillama.com/chains. Accessed 20 May 2022.

Cryptopedia Staff, 2022. “What Is Chainlink? Oracles, Nodes and LINK Tokens”. URL: https://www.gemini.com/cryptopedia/what-is-chainlink-and-how-does-it-work. Accessed 10 June 2022.

Certik, 2022. “Mirror Protocol exploited due to incorrect oracle price”. URL: https://www.certik.com/resources/blog/3IrakqB3V6L9x56trNVY80-mirror-protocol-exploited-due-to-incorrect-oracle-price. Accessed 13 June 2022.

Lililashka 13 January 2022. “NFT use cases 2022”. Tweet @lililashka. URL: https://twitter.com/lililashka/status/1481639448425074694. Accessed 18 June 2022.

OffcierCia, 2022. “DeFi-Developer-Road-Map”. URL: https://github.com/OffcierCia/DeFi-Developer-Road-Map. Accessed 19 June 2022.

Wikipedia, 2022e. “Solidity”. URL: https://fr.wikipedia.org/wiki/Solidity. Accessed 20 June 2022.

Wikipedia, 2022f. “JavaScript”. URL: https://fr.wikipedia.org/wiki/JavaScript. Accessed 20 June 2022.

Wikipedia, 2022g. “ECMAScript”. URL: https://en.wikipedia.org/wiki/ECMAScript. Accessed 20 June 2022.

TypeScript, s.a. TypeScript. URL: https://www.typescriptlang.org/. Accessed 21 June 2022.

Wikipedia, 2022h. TypeScript. URL: https://fr.wikipedia.org/wiki/TypeScript. Accessed 21 June 2022.

Wikipedia, 2022i. Framework. URL: https://fr.wikipedia.org/wiki/Framework. Accessed 21 June 2022.

Wikipedia, 2022j. React. URL: https://fr.wikipedia.org/wiki/React. Accessed 21 June 2022.

Web3.js, 2016. “Ethereum JavaScript API”. URL: https://web3js.readthedocs.io/en/v1.7.4/. Accessed 21 June 2022.

Truffle Suite, s.a (a). “Truffle Suite”. URL: https://trufflesuite.com/. Accessed 22 June 2022.

Truffle Suite, s.a (b). Truffle. URL: https://trufflesuite.com/truffle/. Accessed 22 June 2022.

Moralis, 2021. ’What is the Truffle Suite?’. URL: https://moralis.io/truffle-explained-what-is-the-truffle-suite/. Accessed 22 June 2022.

Truffle Suite, s.a (c). Drizzle. URL: https://trufflesuite.com/drizzle/. Accessed 22 June 2022.

Remix, 2022. Documentation. URL: https://remix-ide.readthedocs.io/en/latest/. Accessed 23 June 2022.

Remix Project, 2021. “Ethereum Remix”. URL: https://marketplace.visualstudio.com/items?itemName=RemixProject.ethereum-remix. Accessed 23 June 2022.

Blanco, J. 2022. Solidity. URL: https://marketplace.visualstudio.com/items?itemName=JuanBlanco.solidity. Accessed 23 June 2022.

Ethereum.org, 2022a. Networks. URL: https://ethereum.org/en/developers/docs/networks/. Accessed 23 June 2022.

Solidity, 2021. “Layout of a Solidity source file”. URL: https://docs.soliditylang.org/en/latest/layout-of-source-files.html. Accessed 23 June 2022.

TutorialsPoint, s.a. “Solidity – Events”. URL: https://www.tutorialspoint.com/solidity/solidity_events.htm. Accessed 24 June 2022.

OpenZeppelin, s.a. OpenZeppelin. URL: https://www.openzeppelin.com/contracts. Accessed 24 June 2022.

Wikipedia, 2022k. “Distributed data store”. URL: https://en.wikipedia.org/wiki/Distributed_data_store. Accessed 25 June 2022.

Arweave, s.a. “Arwiki”. URL: https://arwiki.wiki/#/en/main. Accessed 25 June 2022.

GraphSense, s.a. “GraphSense”. URL: https://graphsense.info/. Accessed 25 June 2022.

CIA, 2022. “TX Analysis Tools”. URL: https://graph.org/TX-Analysis-tools-04-19. Accessed 25 June 2022.

Kenton, W. 2022. The S&P 500 Index: Standard & Poor’s 500 Index. URL: https://www.investopedia.com/terms/s/sp500.asp. Accessed 30 July 2022.

Hayes, A. 2022a. CAC 40. URL: https://www.investopedia.com/terms/c/cac40.asp. Accessed 30 July 2022.

Hayes, A. 2022b. São Paulo Stock Exchange. URL: https://www.investopedia.com/terms/s/sao-paolo-stock-exchange-sao-.sa.asp. Accessed 30 July 2022.

Wikipedia, 2022l. “Know your customer”. URL: https://en.wikipedia.org/wiki/Know_your_customer. Accessed 30 July 2022.

Conway, L. 2021. “Bitcoin Halving”. URL: https://www.investopedia.com/bitcoin-halving-4843769. Accessed 30 July 2022.

Buchko, S. 2022. “How many Bitcoins are left?”. URL: https://coincentral.com/how-many-bitcoins-are-left/. Accessed 30 July 2022.

Frankenfield, J. 2022a. “What is Altcoin?”. URL: https://www.investopedia.com/terms/a/altcoin.asp. Accessed 30 July 2022.

Frankenfield, J. 2022b. “51% Attack”. URL: https://www.investopedia.com/terms/1/51-attack.asp. Accessed 30 July 2022.

CoinMarketCap, 2022. “Blockchain Trilemma”. URL: https://coinmarketcap.com/alexandria/glossary/blockchain-trilemma. Accessed 30 July 2022.

BBC, 2022. “Trudeau vows to freeze anti-mandate protesters’ bank account”. URL: https://www.bbc.com/news/world-us-canada-60383385. Accessed 30 July 2022.

Merriam-Webster, 2022. Oracle. URL: https://www.merriam-webster.com/dictionary/oracle. Accessed 30 July 2022.

Carrefour, 2022. “La Blockchain alimentaire”. URL: https://www.carrefour.com/fr/groupe/la-transition-alimentaire/la-blockchain-alimentaire. Accessed 30 July 2022.

Direction de l'information légale et administrative, 2022. “Crédit immobilier : durée limitée à 25 ans et taux d’endettement plafonné à 35 %”. URL: https://www.service-public.fr/particuliers/actualites/A15426. Accessed 30 July 2022.

Kenton, W. 2020. “Over-Collateralization”. URL: https://www.investopedia.com/terms/o/overcollateralization.asp. Accessed 30 July 2022.

Ichbiah, D. s.a. “Stablecoin : qu’est-ce que c’est?”. URL: https://www.futura-sciences.com/tech/definitions/cryptomonnaies-stablecoin-19671/. Accessed 30 July 2022.

Uniswap, s.a. “How Uniswap works”. URL: https://docs.uniswap.org/protocol/V2/concepts/protocol-overview/how-uniswap-works. Accessed 30 July 2022.

Reiff, N. 2022a. “ERC-20”. URL: https://www.investopedia.com/news/what-erc20-and-what-does-it-mean-ethereum/. Accessed 30 July 2022.

Mirror, s.a. Home. URL: https://docs.mirror.finance/. Accessed 30 July 2022.

JournalDuNet, 2021. “ERC-721 : définition et traduction”. URL: https://www.journaldunet.fr/patrimoine/guide-des-finances-personnelles/1500111-erc-721-definition-et-traduction/. Accessed 30 July 2022.

KaizenDAO, 2022. KaizenDAO. URL: https://fr.kaizendao.com/. Accessed 30 July 2022.

ENS Domains, 2022. ENS Domains. URL: https://ens.domains/fr/. Accessed 30 July 2022.

Glover, E. 2022. “Play-to-Earn games let users… Play to earn”. URL: https://builtin.com/blockchain/play-to-earn. Accessed 30 July 2022.

Binance, 2022. “What is NFT ticketing and how does it work?”. URL: https://www.binance.com/en/blog/nft/what-is-nft-ticketing-and-how-does-it-work-421499824684904022. Accessed 30 July 2022.

RealT, 2021. “What is RealT?“. URL: https://wiki.realt.co/far/what-is-realt. Accessed 30 July 2022.

Sonenreich, A. 2022. “NFTs and the future of commercial real estate“. URL: https://www.forbes.com/sites/forbesbusinesscouncil/2022/02/16/nfts-and-the-future-of-commercial-real-estate/?sh=6b3e620a9bac. Accessed 30 July 2022.

Kumar, R. 2022. “How scammers steal NFTs and cryptos from Twitter users“. URL: https://www.financialexpress.com/digital-currency/how-do-scammers-steal-nfts-and-cryptos-from-twitter-users/2540842/. Accessed 30 July 2022.

Keller, A. & Hofstetter, R. s.a. “Crypto custody: risks and controls from an auditor’s perspective“. URL: https://www.pwc.ch/en/insights/digital/crypto-custody-risks-and-controls-from-an-auditors-perspective.html. Accessed 30 July 2022.

Reiff, N. 2022b. “What’s the environmental impact of cryptocurrency?“. URL: https://www.investopedia.com/tech/whats-environmental-impact-cryptocurrency/. Accessed 30 July 2022.

ETHTPS, 2022. Ethereum. URL: https://ethtps.info/Network/Ethereum. Accessed 30 July 2022.

BitPanda, 2022. “Le problème de scalabilité du réseau Bitcoin“. URL: https://www.bitpanda.com/academy/fr/lecons/le-probleme-de-scalabilite-du-reseau-bitcoin/. Accessed 30 July 2022.

Sedgwick, K. 2022. “No, Visa doesn’t handle 24,000 TPS and neither does your pet blockchain“. URL: https://news.bitcoin.com/no-visa-doesnt-handle-24000-tps-and-neither-does-your-pet-blockchain/. Accessed 30 July 2022.

Stakepool, 2020. “Définition: Testnet“. URL: https://stakepool.fr/definition/testnet. Accessed 30 July 2022.

Beck, J. & Asher, M. 2021. “What is EIP-1559? How will it change Ethereum?“. URL: https://consensys.net/blog/quorum/what-is-eip-1559-how-will-it-change-ethereum/. Accessed 30 July 2022.

Millman, R., Graves, S. & Kelly, L.J. 2022. “What is Ethereum 2.0? Ethereum’s consensus Layer and Merge explained“. URL: https://decrypt.co/resources/what-is-ethereum-2-0. Accessed 30 July 2022.

Berné, R. 2018. “Qu’est-ce qu’un Oracle ? Blockchain et monde réel“. URL: https://cryptoast.fr/oracle-blockchain/. Accessed 30 July 2022.

Aave, 2022a. Tether. URL: https://app.aave.com/reserve-overview/?underlyingAsset=0x9702230a8ea53601f5cd2dc00fdbc13d4df4a8c7&marketName=proto_avalanche_v3. Accessed 30 July 2022.

Aave, 2022b. Wrapped BTC. URL: https://app.aave.com/reserve-overview/?underlyingAsset=0x50b7545627a5162f82a992c33b87adc75187b218&marketName=proto_avalanche_v3. Accessed 30 July 2022.

Defi Llama, 2022c. “Lending TVL rankings”. URL: https://defillama.com/protocols/lending. Accessed 31 July 2022.

Lutz, S. & Benson, J. 2022. “What is Aave? Inside the DeFi lending protocol”. URL: https://decrypt.co/resources/what-is-aave-inside-the-defi-lending-protocol. Accessed 31 July 2022.

Aave, 2021a. Liquidations. URL: https://docs.aave.com/faq/liquidations. Accessed 31 July 2022.

Aave, 2022c. Borrow Interest Rates. URL: https://docs.aave.com/risk/liquidity-risk/borrow-interest-rate. Accessed 31 July 2022.

Aave, 2021b. “What is health factor?”. URL: https://docs.aave.com/faq/borrowing#what-is-the-health-factor. Accessed 31 July 2022.

Thales, s.a. “Know Your Customer in banking”. URL: https://www.thalesgroup.com/en/markets/digital-identity-and-security/banking-payment/issuance/id-verification/know-your-customer. Accessed 31 July 2022.

Probasco, J. 2021. “Currency conversion fees”. URL: https://www.investopedia.com/currency-conversion-fee-definition-4768870. Accessed 31 July 2022.

Ziyanurov, I. 2020. “Why KYC is important for Brokerage Companies”. URL: https://b2broker.com/fr/news/why-kyc-is-important-for-brokerage-companies/. Accessed 31 July 2022.

Uniswap, 2022. Iframe integration. URL: https://docs.uniswap.org/protocol/V2/guides/interface-integration/iframe-integration. Accessed 31 July 2022.

Blockchain France, 2018. “Comprendre la tokenisation”. URL: https://blockchainfrance.net/2018/05/22/comprendre-la-tokenisation/. Accessed 31 July 2022.

Davies, P. 2022. “Terra Lune stablecoin collapse explained: Is this the 2008 financial crash moment of cryptocurrency?”. URL: https://blockchainfrance.net/2018/05/22/comprendre-la-tokenisation/. Accessed 31 July 2022.

Sacristán, M. 2022. “NFT tickets, the new fan engagement”. URL: https://barcainnovationhub.com/nft-tickets-the-new-fan-engagement/. Accessed 31 July 2022.

Ledger, 2020. “Not your keys, not your coins. It’s that simple”. URL: https://www.ledger.com/academy/not-your-keys-not-your-coins-why-it-matters. Accessed 31 July 2022.

CFI Team, 2021. Bank Run. URL: https://corporatefinanceinstitute.com/resources/knowledge/other/bank-run/. Accessed 31 July 2022.

Trust Wallet, s.a. “Best practices for storing your recovery phrase”. URL:

https://corporatefinanceinstitute.com/resources/knowledge/other/bank-run/. Accessed 1 August 2022.

Blockbase, s.a. Crypto Mining. URL: https://blockbasemining.com/when-replace-mining-hardware/. Accessed 1 August 2022.

Newar, B. 2022. “Earth day analysts say Bitcoin mining is naturally gravitating to green energy”. URL: https://cointelegraph.com/news/this-earth-day-analysts-say-bitcoin-mining-is-naturally-gravitating-to-green-energy. Accessed 1 August 2022.

Carbonnelle, P. 2022. “Top IDE index”. URL: https://pypl.github.io/IDE.html. Accessed 1 August 2022.

Coen, E. 2020. “Understanding Ethereum transaction fees: What is Ethereum gas?”. URL: https://cryptotesters.com/blog/ethereum-gas. Accessed 1 August 2022.

Lastname, E. 2021. “How much does it cost to deploy a smart contract on Ethereum?”. URL: https://medium.com/the-capital/how-much-does-it-cost-to-deploy-a-smart-contract-on-ethereum-11bcd64da1. Accessed 1 August 2022.

North, C. 2022. “The complete solidity course – Blockchain – Zero to Expert”. URL: https://www.udemy.com/course/the-complete-solidity-course-blockchain-zero-to-expert/. Accessed 1 August 2022.

Roth, S. 2022. “An introduction to sidechains”. URL: https://www.coindesk.com/learn/an-introduction-to-sidechains/. Accessed 1 August 2022.

Crédit Agricole, 2021. “Document d’information tarifaire”. URL: https://www.credit-agricole.fr/content/dam/assetsca/cr882/npc/documents/tarifs-2021/Document-Information-Tarifaire-012021.pdf. Accessed 1 August 2022.

ECB, 2022. “We have raised interest rates. What does that mean for you?”. URL: https://www.ecb.europa.eu/ecb/educational/explainers/tell-me-more/html/interest_rates.en.html. Accessed 1 August 2022.

N26, 2022. “How long does a transfer take?”. URL: https://n26.com/en-eu/blog/how-long-does-a-bank-transfer-take. Accessed 1 August 2022.

Smith, M. s.a. “SWIFT transfers explained (how they work how long they take & what they cost)”. URL: https://www.keycurrency.co.uk/swift-transfer/#:~:text=A%20SWIFT%20payment%20generally%20takes,laundering%20checks%2C%20which%20takes%20time.. Accessed 1 August 2022.

MIT, s.a. “51% Attacks” URL: https://dci.mit.edu/51-attacks#:~:text=Proof%2Dof%2DWork%20is%20intended,hashrate%20on%20the%20target%20cryptocurrency.. Accessed 1 August 2022.

Hong, E. 2022. “How does Bitcoin mining work?” URL: https://www.investopedia.com/tech/how-does-bitcoin-mining-work/. Accessed 1 August 2022.

Origin Stamp, s.a. “Block Rewards vs. Transaction fees – Why we need both” URL: https://originstamp.com/blog/block-rewards-vs-transaction-fees-why-we-need-both/. Accessed 1 August 2022.

Hayes, A. 2022c. “What happens to Bitcoin after all 21 million are mined?” URL: https://www.investopedia.com/tech/what-happens-bitcoin-after-21-million-mined/. Accessed 1 August 2022.

Phillips, D. 2020. “Why is Bitcoin supply limit set to 21 million?” URL: https://decrypt.co/34876/why-is-bitcoins-supply-limit-set-to-21-million. Accessed 1 August 2022.

Ethereum.org, 2022b. “Définition d’une DApp” URL: https://ethereum.org/fr/developers/docs/dapps/#definition-of-a-dapp. Accessed 1 August 2022.

The Investopedia Team, 2022. “How do you transfer common stock from one broker to another?” URL: https://www.investopedia.com/ask/answers/021015/how-do-you-transfer-common-stock-one-broker-another.asp. Accessed 1 August 2022.

Gravier, E. 2022. “Fractional shares allow you to own part of a big-name stock without the large price tag” URL: https://www.cnbc.com/select/fractional-shares/. Accessed 1 August 2022.

Amanwachi, T. 2022. “The ultimate guide to data types in Solidty” URL: https://blog.logrocket.com/ultimate-guide-data-types-solidity/#:~:text=An%20address%20value%20type%20is,can%20send%20and%20transfer%20Ether.. Accessed 1 August 2022.

McKie, S. 2017. “Solidty Learning: Revert(), Assert(), and Require() in Solidity, and the new revert Opcode in the EVM” URL: https://medium.com/blockchannel/the-use-of-revert-assert-and-require-in-solidity-and-the-new-revert-opcode-in-the-evm-1a3a7990e06e. Accessed 1 August 2022.

Crescenzi, D. 2018. “What you need to know about ‘msg’ global variables in Solidity” URL: https://medium.com/upstate-interactive/what-you-need-to-know-about-msg-global-variables-in-solidity-566f1e83cc69. Accessed 1 August 2022.