# AERLEASE: A TECHNICAL WHITE PAPER ON BLOCKCHAIN BASED AIRCRAFT LEASING

A PREPRINT

**CLARA ONUNKWO***
x18122558
Blockchain Technology
MSc Fintech
National College of Ireland

**ALEXANDER HYLAND**
x18166342
Blockchain Technology
MSc Fintech
National College of Ireland

**ROSHAN RAMCHANDRAN**
X18120245
Blockchain Technology
MSc Fintech
National College of Ireland

**KEVAL JOSHI**
X17161487
Blockchain Technology
MSc Fintech
National College of Ireland

April 24, 2019

## 1 INTRODUCTION

The airlines have a huge concern of earning profits considering the high cost of an aircraft itself. There are many costs involved when purchasing an aircraft including huge down payments and high monthly repayments along with the depreciation costs during the operational life of an aircraft. So, rather than purchasing an aircraft, airlines lease these aircraft in order to increase the size of their fleet. The percentage of aircraft's leased by the airlines in last 50 years in shown in Table 1. As seen from the Table 1, this percentage has risen from $\mathbf{0.5}\%$ in 1970 to more than $\mathbf{40}\%$ in 2014. This percentage is estimated to reach around 50 percent as per a study and forecast conducted by Boeing Company. This rapid increase in the percentage indicates how important the aircraft leasing is for the airline companies. Moreover, there will be a huge rise in the number of commercial aircraft in the world because of the worldwide growth in air transportation. According to a prediction made by Boeing Company (2014), the demand of the new aircraft would be so much that almost 5.2 trillion USD will be invested in 36,770 new aircraft would be required by the airlines from 2014 to 2033. These statistics indicate that finances of the airlines would be affected by the decisions made for aircraft purchasing and leasing in the coming years (Chen, Huang and Ardiansyah, 2018).

| Years | 1970 | 1980 | 1990 | 2000 | 2014 | 2020 (Forecasted) |
|---|---|---|---|---|---|---|
| Percentage | 0.5% | 1.7% | 14.7% | 24.7% | 40.7% | 50% |

Figure 1: The share of leased aircraft in airline's fleets across the world (Chen et al., 2018)

The decisions taken by the airlines with respect to their products, network, prices and the number of aircraft in their fleet play a major role in setting their company's strategies. The most challenging decisions are the ones that are made about aircraft as they require huge investment and would last for a longer time, usually more than 25 years. However, airlines can be little flexible while making decisions on their fleet of aircraft as they can opt for operating leases rather than financial leases or ownership of the aircraft which will enable them to operate the leased aircraft for a limited period of time and with a comparatively lower capital requirement. In operating leases, the proprietorship risks and

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

incentives stay with the lessor and the assets are off the lessee's accounting sheet (BouRjade, Huc and Muller-Vibes, 2017).

Ireland has for quite some time been viewed as a leading country for leasing and aviation finance. The aviation industry started to develop in Ireland since the time when Guinness Peat Aviation (GPA) was founded by late Dr. Tony Ryan in Shannon, County Clare back in the 1970s. Despite its fizzled buoyancy in the mid-1990s, GPA assumed a key role in shaping Ireland's notoriety for being a key – and probably the most important – hub for the activities related to aviation finance. Meanwhile other countries, such as Malta, Singapore and the Cayman Islands as of late, have advanced cases to challenge its status, Ireland has kept on improving its offering in an attempt to not only hold on to its current businesses but also to contend successfully in the worldwide commercial centre (Keaveny and Eustace, 2013).

Ireland's achievement as the leader in the aviation finance industry can be seen by the number of companies who have chosen Ireland as a location for operational purposes. These companies include GE Capital Aviation Services (GECAS), Sumitomo Mitsui Banking Corporation (SMBC), International Lease Finance Corporation (ILFC), Ansett Worldwide Aviation Services (AWAS), Avolon, Industrial and Commercial Bank of China Ltd. (ICBC), CIT and Orix. Out of the top 10 aircraft leasing companies in the world, 9 have their bases in Ireland and the number of aircraft managed in Ireland go as high as 3500 or almost 50 percent of the overall leased aircraft in the world. The aviation finance sector pays almost 300 million euros in corporation tax to the Irish government and directly and indirectly employs over 2000 people (Keaveny and Eustace, 2013).

The factors that led to Ireland's phenomenal success in the aviation leasing industry and its advantages over other countries such as Malta, Singapore and the Cayman Islands are:

- A stable and sophisticated legal system and a very favourable taxation regime.
- Membership of the European Union (EU) and the Organization for Economic Co-operation and Development
- A geographic location which can serve as the connecting point between the United States and Europe and Asia.
- It is an English-speaking country and has a workforce with ample amount of the experience in the industry (Keaveny and Eustace, 2013).

## 2 MOTIVATION

Undoubtedly, the aircraft leasing industry is one of the fastest growing industries in the world as a result of various factors such as increase in the frequency air passenger traffic every year, increase in the number of low-cost airlines and the growth of the Asia Pacific (APAC) market. Due to the financial nature of this industry, it is interesting to know that some of the traditional and conventional methods and practices are still in place. For example, aircraft maintenance is the process which uses traditional as well as modern techniques like strong data on the recent and modern databases and on the other hand, the system has a lot of paperwork which can lead to errors and losses (Kehoe and Hallahan, 2018).

With the rapid digitalisation of the most of the industries in the world, we are not far from the time when the technology will be implemented in the aircraft leasing industry as well. This paper explores the advantages of the blockchain technology and proposes the implementation of blockchain smart contracts to the aircraft leasing industry which will have disruptive effects on the stakeholders involved in the process (Kehoe and Hallahan, 2018).

Blockchain technology, also known as the distributed ledger technology, is the decentralised underlying technology on which the Bitcoin cryptocurrency was introduced by Satoshi Nakamoto back in 2009. After its introduction, the application of blockchain technology started to expand in various fields such as Finance, Supply Chain, Real Estate, Crowdfunding, Healthcare, etc. The most widely used technology within the blockchain is the use of Smart Contracts. Smart Contract is a digital document defined by the lines of code which can executed or enforced to perform or not to perform a certain task according the terms and conditions agreed by the participating parties. Smart Contracts have the ability to maintain autonomy, transparency and self-sufficiency on a decentralised blockchain network (Swan, 2015).

Moreover, our group had attended a Blockchain meetup "BlockW Series 5: Blockchain meets Aircraft Leasing" at the ConsenSys office in Ringsend, Dublin where the panel discussed about the possible adoption of blockchain technology in the Aircraft Leasing industry. As Ireland is the global leader in the aircraft leasing industry, this would be of a huge advantage as it would reduce the paperwork, eliminate the middlemen and automate the process of agreement or contract between the two parties (Meetup, 2019).

# 3 RELATED WORKS

## 3.1 Blockchain based Online Voting

Shukla et al. (2018) proposed a blockchain based electronic voting system so that the voters can vote online instead of physically going to the voting booths. Using the blockchain technology and smart contracts, a peer-to-peer network is created where every single vote is considered as a transaction on this distributed ledger network and encryption and hashing algorithms are used to secure all the votes. The government will identify the voter with their unique identification number known as Aadhar number. With the help of the Aadhar number, the number of votes assigned per vote will just be limited to one ensuring that no one could vote twice.

The smart contract is deployed on the private blockchain network with the help of Ethereum network. They have used Solidity (a combination of C++ and JavaScript) to write the smart contract and deployed it on the private blockchain network supported by the peers. Whenever a new peer is added to the network, they have to execute the functions as per the contract just like other peers. The proposed system addresses most of the security factors such as verification, transparency of counting votes, integrity of votes, etc.; however, it does address the authenticity of a voter which can achieved using the biometric features (Shukla et al., 2018).

## 3.2 Blockchain based decentralization of Bucharest Stock Exchange

Pop et al. (2018) proposed a model for decentralizing the Bucharest Stock Exchange (BVB) using the Ethereum Blockchain network. The centralized stock exchanges have some disadvantages such as high transaction fees by the brokers or traders, lack of transparency about the algorithms and market actions and these systems were even vulnerable to attacks due to its centralized nature. The proposed model uses smart contracts that eliminates the need of central authority by correctly executing and settling the orders and enforcing the validation of owner's right. The smart contracts were deployed on the blockchain network and the prototype was implemented on the Ethereum network for validation and testing purposes. The results show that the proposed decentralized solution gives offers low transaction fees as compared to the centralized systems thus eliminating the need of paying commission to brokers.

## 3.3 Eximchain: Blockchain based solution for Supply Chain Finance

Huertas, Liu and Robinson (2018) have build a smart contract based blockchain network for small and medium enterprise (SME) that will help them to gain access to affordable financial sources to grow their business. It will also give the investors an idea about the investment they are making and the cash flow process in their supply chain. Keeping the data privacy in mind, smart contract can be implemented on the permissioned Ethereum network.

Moreover, Eximchain allows the suppliers and buyers to customize the smart contracts and provide solutions according to their specific supply chain needs. The consensus algorithm used by Eximchain is QuorumChain, a vote-based algorithm, instead of proof-of-work. They added this to add some governance rules.

## 3.4 Decentralized Smart Home System using Ethereum

Xu et al. (2018) used Blockchain technology to develop an Ethereum based prototype for controlling smart home applications using IoT. Along with private Ethereum blockchain network, Raspberry Pi computer, Blynk App and DHT11 temperature and humidity sensors, a proposed prototype has been built that can control the real-time temperature and humidity in the house and will be able to react automatically in the event of any uncontrolled mishap.

The temperature and humidity of the house can be monitored by the two DHT11 sensors which are connected to two different Raspberry Pi computers placed in two separate rooms. Smart contracts will be used to store these values on private Ethereum blockchain network. In the prototype, if the temperature or humidity of the room goes beyond the threshold set by the user, the LEDs will light up as warnings to the user. However, in real-time scenario, instead of LEDs, air-conditioner or dehumidifier will be used. When the room temperature or humidity exceeds the threshold value, the air-conditioner or dehumidifier will turn on automatically.

## 3.5 FabRec – Blockchain in Manufacturing

A blockchain based solution for handling manufacturing information has been proposed by Angrish et al. (2018). Their system proposes the use of decentralized network of manufacturing machines and computer nodes to enable the transparency between the organizations and their capabilities. It can also be used to verify and conduct a background check of these organizations and the use of Ethereum smart contracts to drive paperless contracts between the participating organizations.

# 4 ROAD MAP

For a road map to be established on would be obliged to follow a supporting methodology to ensure the proper step are taken to attaining one's goals. Due to the nature of the Ethereum network being immutable to allow for the decentralising of information and power any data stored in the blockchain becomes unalterable, this also applies to the smart contracts that run the decentraliZed apps (DApps). With the immutability now understood one must also acknowledge that due to the general nature of the application running on the Ethereum network consisting of exchanging financial data or currency. This would require a clear method of preparation to ensure no bugs or security flaws exist. While the application of methodologies to blockchain has not been well covered academically, many blockchain or technology websites have weighed in on the situation. One would find that the general understanding is to treat smart contracts within blockchain technologies as a form of software development. Thus, aligning it with the software development lifecycle (SDLC). The most widely known methodology of SDLC is the waterfall or also known as the traditional method. This method was widely adopted in the early software years; however, developers began to find that it was too structured to effectively deliver on the customer's needs. One online publication (Radic, 2018) proposed that all development efforts should roughly follow the structure of:

- Gather information
- Define specifications
- Develop code
- Test
- Audit
- Deploy

The author admits that this is of similar nature to the waterfall method, however, they also point out that this can be applied in iteration to each section of the code to develop functions incrementally. They also note that in doing so you would be closer to an agile methodology. As seen in figure two below is a visual representation of the waterfall methods negative aspects.

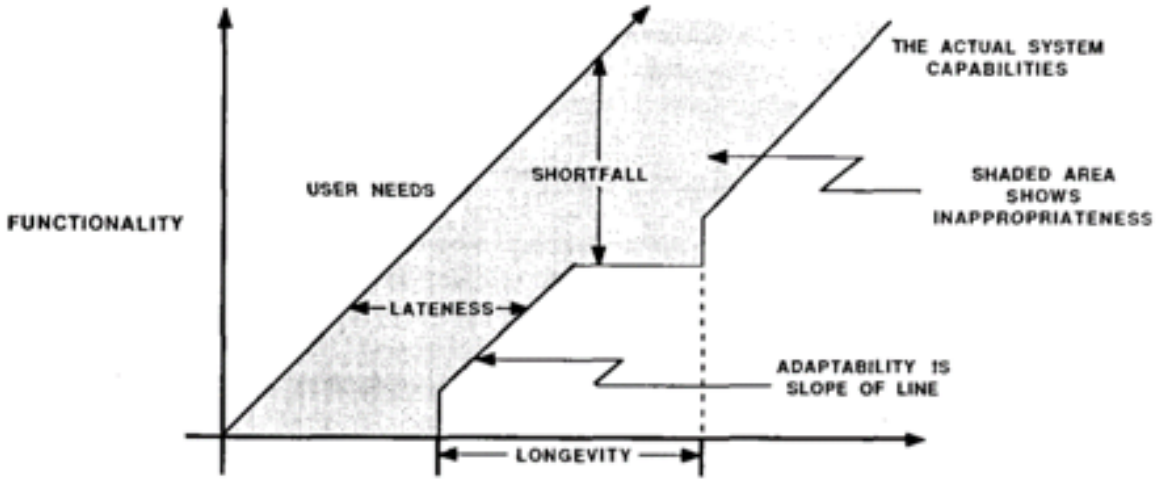


Figure 2: The negative aspect associated with waterfall process (Davis et al. 1988)

With this in mind, an online article (education, 2018) proposed a method that called upon the core elements of the agile methodology which are people over tools, working code over extensive documentation, customer engagement over contract engagement and reacting to changes over following a plan ("Manifesto for Agile Software Development," n.d.). The online work extends from this as it specifically addresses the different challenges met by developing smart contracts. The online article elaborates that the work to do me done is broken down into 3 sections; the work to be done, the time-line and the budget. This allows for incremental goals to be set and communicated to customers. One journal could be seen to support this information as it concludes that in cases of smaller projects and in shorter time periods that the agile methodology is most suited also boasting the ability to allow for greater ability to adapt to user changes which

could be frequent in building a smart contract (Balaji, 2012). This need for the correct choice of methodology is evident on one study as in a survey conducted by the paper $81\%$ of participants cited that incorrect choice of a project's lifecycle could provide significant problems (Khan and Beg, 2013). Again the choice for an agile methodology is supported in another journal, citing that agile is best suited for projects that need to be frequently changed (Mahalakshmi and Sundararajan, N.D.).

According to the first article referenced in addition to adopting an appropriate methodology one should also include a UML and data work-flow chart (Radic, 2018). according to ("The Importance of Using UML for Modeling," 2014), UML diagrams provide a far better understanding between the business teams and the IT department. This understanding will enable communications between the two to be more understandable as it can be visually represented. It allows for a focused effort by all parties to achieve the specifications needed. The article makes a comparison to a house being built without any plans being drawn up. Equally important to a project's objective is a flow chart. Flowcharts allow for better understanding of the logic of operations within an application or system. They are said to be an "imperative part of developing an information system" ("Importance of flowcharts when developing information systems," 2013). One journal specifically referenced flowcharts to communicate the complex process to different stakeholders (Figl and Strembeck, 2014). This view can also be transferred to UML diagrams. This can also be supported by another journal's research comparing pseudocode to flowcharts (Scanlan, 1989). The journal states that apart from very basic workflows that flowcharts can be comprehend up to two and a half times fast when compared to pseudocode, while also citing that pseudocode resulted in more errors. The difference between both UML and flowcharts are subtle in observation but quite different in application. UML is aligned with the logical flow of processes mainly consist of questions generally leading to a yes or no route. While a flowchart would hold the specific code actions to take place when a condition is met within a stage of a UML diagram, the code support can be either pseudo or actual code snippets depending on the stakeholders that will see it ("Comparing flow charts and activity diagrams," 2014). With the importance of both, one would be inclined to develop them in relation to this application. Examples of how both techniques may look for this application is shown in figure 3 below while figure 4 shows the flowchart:

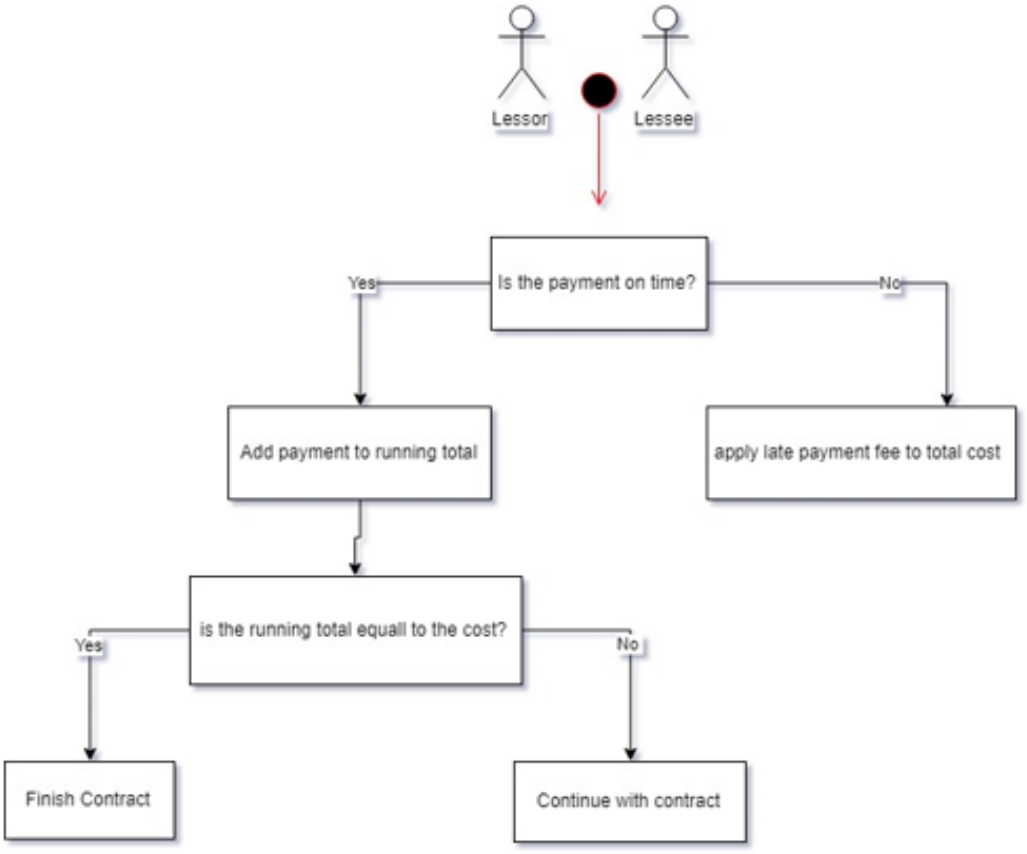


Figure 3: The UML diagram

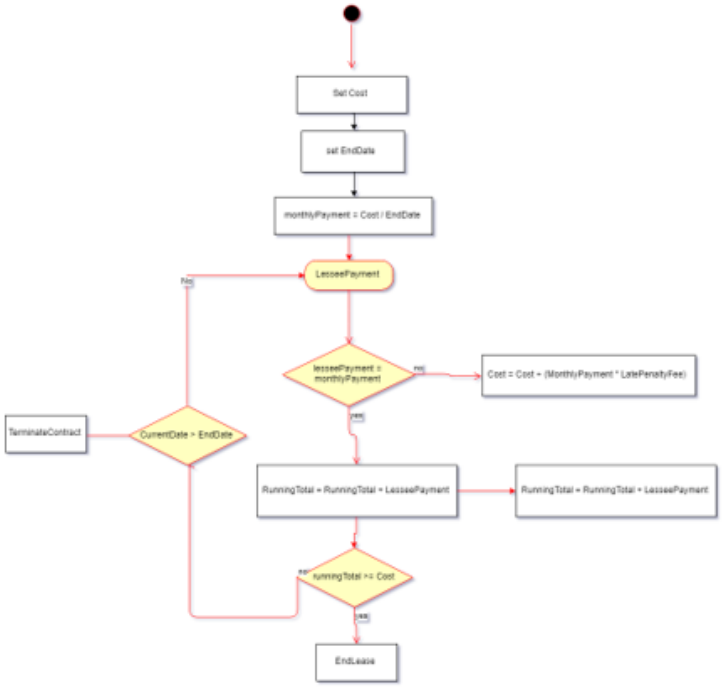


Figure 4: The flowchart

## 5 Implementation of Smart Contracts on Ethereum

At the beginning of the Ethereum whitepaper, Vitalik Buterin talks about bitcoin introducing two "Radical" concepts to the world. There first being a total unbacked currency, meaning it is aligned to no country or currency that currently exists. This currency and how it was spent was governed by the second radical concept of using peer to peer public agreement to determine the time-line and legitimacy of payments. This allows for the currency to maintain its integrity allowing for greater use. Building on Bitcoins accumulation of knowledge to build the first blockchain network, Buterin proposed a new network that would give the users the ability to right stipulations to transaction in the form of "Smart contracts". Smart contracts were first coined by Nick Szabo in his early paper (Szabo, 1997), making reference to the ability to own property through programmable contracts. It is widely considered that Nick Szabo is, in fact, the original author of the bitcoin whitepaper and is either a solely or jointly attributed to the identity of Satoshi Nakamoto the unknown "Founder" of Bitcoin. This is evident in one article (Buzz, 2018). Ethereum white paper is a technically challenging read and provides a deep understanding of security, payments and network consensus which can be found at (Buterin, n.d.). There is no need to address this information again as this application is built upon this technology not building another one. An overview if the whitepaper would be that it was originally built as an upgrade of a cryptocurrency network but as was investigated further it was discovered that the applications can provide far more than monetary transfer, it can also facilitate the transfer of information or sharing of physical assets from cars to servers. The network has been built to enable further development and ingenuity through open source codes and community-built application.

The technical knowledge needed to build an application of this size is substantial. Initially, the requirement to obtain a strong knowledge of solidity is required. Solidity holds similarities to both Java and C programming languages. While Solidity is not the only language supported on the Ethereum Virtual Machine (EVM) it is the most commonly used (Dannen, 2017). There is scope for another language to be introduced that trumps solidity, however, that is not the case now. As the Ethereum network is Turing complete, essentially meaning it can complete loops, the language used within it must also allow for this. Loops can be built like Java. As most programs are written is solidity involves the exchange or settlement of monetary value it is imperative that an application is tested prior to deployment. In the Ethereum network, there are a few ways of doing so, an example is the Ropsten network, which is identical to the actual network only the ether within it holds no value. Another test network was built known as Rinkeby. Unlike Ropsten, this network run on a proof of authority which allows for a centralized server to do most of the work. As the ropsten network was an exact replica of the main network it implemented a proof of work protocol which meant that miners

would complete the computations related to adding a block in exchange for ether. As the ether in ropsten holds no value there was no incentive to use the network. The easiest way to compile solidity is to do it in an online compiler, an example of such is https://remix.ethereum.org/ in the remix compiler one can connect to any network address, this can allow for testing to be done locally with having to connect to a wider network. this can be done through an Ethereum affiliate known as Ganache. Ganache is a subsection of the truffle suit, which a tri-section Ethereum development suit. It provided programmes to build, test and analysis prototype application. The sole problem with this suit in relation to this application is that it currently has a steep learning curve to use its building and analysing tools. The Ganache tool provides the user with 10 test addresses and a local network to deploy contracts and evaluate how the code is working ("Truffle Suite | Ganache," n.d.). This these address and local network can be linked to the Remix IDE platform through changing the environment dropdown to web provider in the run tab of Remix. The user then inserts the local address from the ganache application to remix and the two are connected. A list holding the 10 address on the local network will now be available in Remix. The steps are shown below with the help of diagrams as in figure 5, 6, 7 and 8:

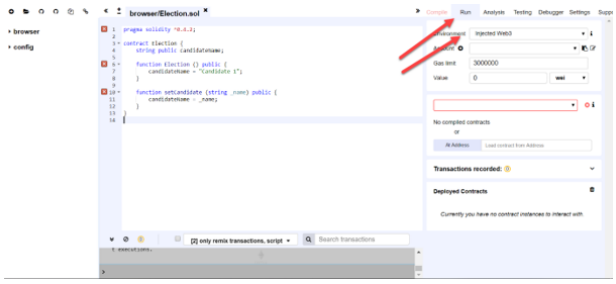

Figure 5: The introduction of the remix I.D.E

Select the run tab, then the drop-down environment menu. From the list select the web3 provider option and press ok on the pop-up. Next, open Ganache and copy the localhost address.

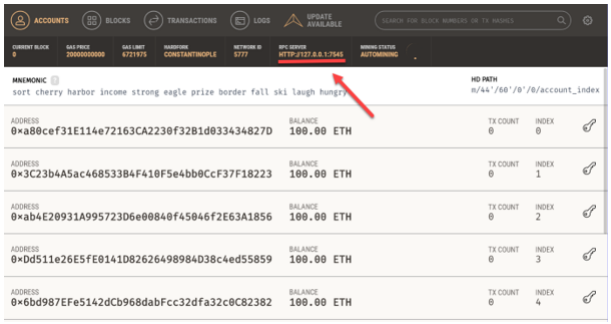


Figure 6: Ganache

Paste the address from your Ganache to Remix.

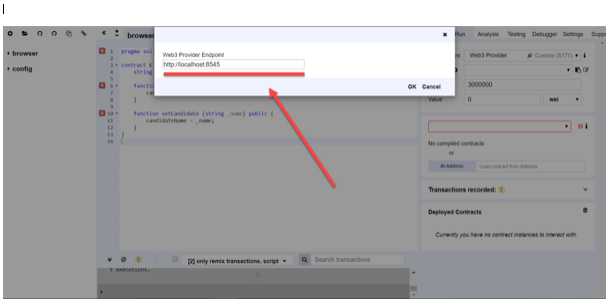

Figure 7: The port address of Ganache connecting with remix

Your Ganache address is now available in the account drop-down menu as below.

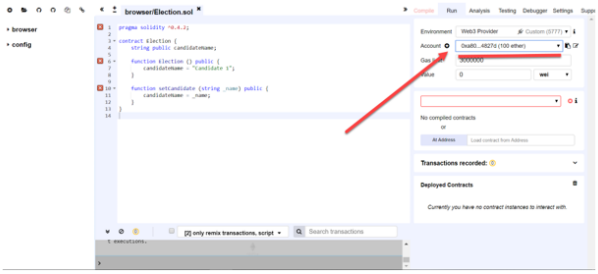

Figure 8: The figure shows that remix is now linked to ganache

Based on the figures as shown above,it shows that from this point, the user can follow the methodologies as discussed before to build and test the application before consumer use.

## 6 TECHNICAL PLAN

Description of each state used in Smart Contracts of Ethereum Network:

| State name | Description |
|---|---|
| Created | The Lessor has created the contract and initiated it. |
| Plane-Assessed | The State reached when the condition of the plane assessed by the plane manager. |
| Paid | The state of paying the security deposits, rents, late rent with or without the additional fine. |
| Awaiting-Pay | Every month, this state is monthly due when it reaches every month. |
| Conflict | There is a conflict, when the lessee fails to pay the lease rent within the starting period, this state reaches. |
| Paid-Extended | This is the state when the Lessee should give a one-month notice period without losing his security deposit after a 12-month period passed. This will be conditional possible if the Awaiting-Pay-Extended state attained. |
| Awaiting-Pay-Extended | This is the state when the Lessee should give a one-month notice period without losing his deposit in this state. |
| Conflict-Extended | This state resembles the Conflict state, with the addition that a 12-month period has passed. |
| Final-To-Be-Paid | This state symbols that there will be one last payment to do and should be process by Lessee or by Lessor. |
| Lease-End | When final payment paid by the lessee, this state reached. |
| Lease-Closed | If the Lessee fails to pay his last payment within the designated time, this state can be reached by paying the late rent plus the fine. |
| Finished | This is the final state after reaching through all the possible state which lead to finish state and this is the final state of leasing contract. |

Figure 9: The figure shows the description of each state used in Smart Contracts of Ethereum Network

| Transition Name | Description statement it is based on |
|---|---|
| Create | This is the transition when the Lessor starts a contract. It saves the contract to the blockchain. At the start of the contract, the Lessor must launch the contract. |
| Assess | In this transition, at the beginning of the contract, the aircraft manager inspects the condition of the aircraft. At the beginning of the contract, the plane manager must assess the contract status. |
| Pay Deposit | The Lessee is paying the deposit of security. At the beginning of the contract, the lessor must pay the security deposit. |
| Rent Due | This is an automatic system transition. The system will automatically change the status from Paid to AwaitingPay after 30 days. Every month, the system must shift to accept the payments. |
| Pay Rent | For this transition, the Lessee is the trigger in which lessee transfers currency to pay for the rent, transitioning the state to pay. Every month, Lessee must pay the rent or else the rent is late. |
| Rent Late | If the rent is not paid by the lessee within five days, the system will automatically switch to the following state: Conflict. The condition for this transition is that five days have passed since reaching the state of AwaitingPay. System must register the rent to be late if five days have passed since the status of acceptance of payment has been reached. |
| Pay Late | The Lessee, which pays the rental plus an amount of money for being late with the rent, is initiating this transition. This transition leads to Lease Closed if the contract is in the extended period and notice has been given. If the rent is late or the contract is terminated, a lender must pay the rent and a fine. |
| End Lease | If the contract runs for 12 months, the lease may be terminated by the Lessee or the Lessor. If the final payment has been made within the 12-month period, the system moves into the finished state automatically. When the original leasing period ends, or the lease can be extended, the lease is terminated by the renter. If the original lease period ends or the lease contract is extended, the lessor may terminate the lease contract. If a final payment is made within 12 months, the system must cease the lease. |
| Early Terminate | Before the 12-month contract period, the Lessor may notify the Lessee, but lessee will lose the security deposit. This transition is conditional on the lease period being less than 11 months. The lessee can notification about the security deposit before the lease period ends. |
| Extend | The Lessor may extend the lease by the transition once the original 12-month lease period has agreed. The Lessor may extend the lease contract if the lease period is terminated or the contract is terminated. |
| Give Notice | The Lessee may ask for an extension when the lease is increased, and the rent is due. This is done through such a transition to pay the rent. If the original lease period has expired, Lessee may give a one-month notice to quit through alternative payment. |
| Terminate Contract | The contract is concluded, and the lessee is not refunded the deposit. If the rent is ten days late, the lessor can terminate the contract. If the rent is ten days late, the lessor may keep the security deposit. If notice has been given and final rent is late ten days, the Lessor may terminate the contract. |
| Return Deposit | The Plane Manager inspects the state of the airplane again, after the final payment has been made by the Lessee. If the aircraft is in a similar state as in the first appraisal, the Lessor receives his security deposit. The aircraft manager must evaluate the state of the aircraft system if the assessment of the aircraft is like the first assessment. |

Figure 10: Description of each transition from one state to another:

The figure 9 above shows the description of each variable used in the smart contract of the etherum contract. This consist of of twelve state name along with its description. This is a part of the technical plan used for the achievement of this project. Figure 10 shows the description of each transition from one state to another consisting of 13 transition name with its description statement attached by it.This figures shows the full technical plan used for this project.In the figure 10, the lessor starts a contract which allows the aircraft manager to inspect the condition of the aircraft as will be shown in the full description of how the transition does takes place from one state to another. Just after the assessment,

the lessee pays the deposit of the security then various activities are then followed till the contract is terminated and the deposit returned

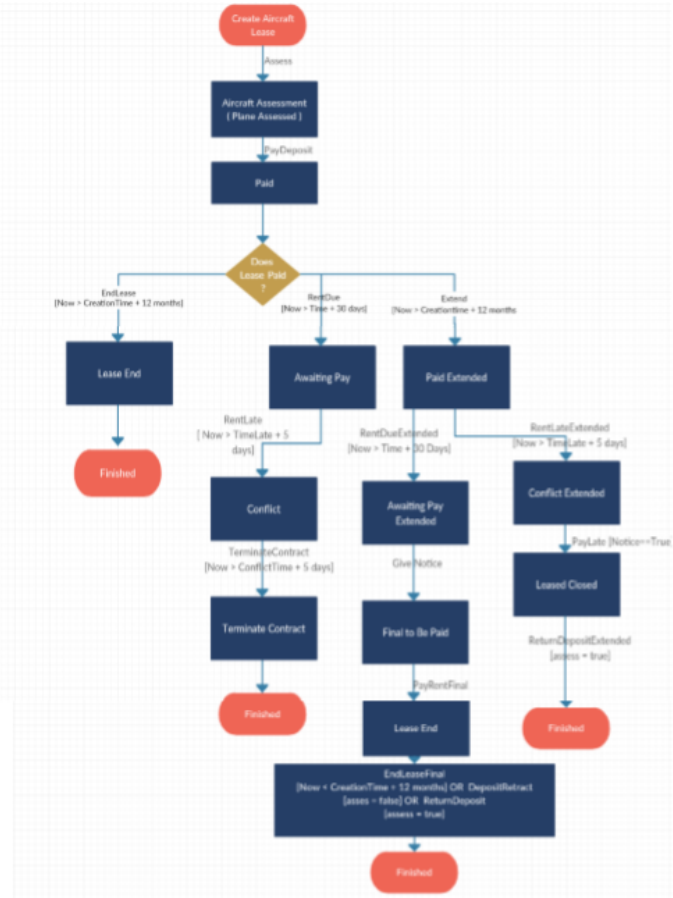


Figure 11: Description of the Unified Modelling Language(UML) diagram

### 6.1 Security implications and solutions

As this application contains a transfer of a financial asset in the form of the ether, one would be required to look at the level of security required to meet the sensitivity of the application. Ethereum has already implemented many security features, from block validation with Merkle trees known as Patricia trees to modified Ghost, which was cover in its own white paper. The main security treat comes in the form a code poor coding as opposed to network errors as the network is self-governed and discourages bad actors. According to one book the vulnerabilities of smart contracts can be broken into 3 sections: 1) solidity 2) Ethereum virtual machine 3) Blockchain (Atzei et al., 2017). Some of the solidity vulnerabilities noted include calling other contracts that can execute malicious code. A re-entrance can incur a loop of sending ether without running out of gas. This can be done by one account sending a payment to a contract, that contracts create a loop calling and forwarding the money to another contract, each forward will reset the transaction loop the gas will keep defaulting to 2300 units of gas, this is how the DOA attack happened resulting in nearly 50 million worth of ether being stolen. An exception disorder will allow a contract to send money to others with confirming that is has happened. This ability for an exception to be skipped through call functions can lead to potential vulnerabilities in the future. The final solidity vulnerability discussed involves keeping secrets. Where information on moves or actions is stored separately to the overall contract. For example, if a game is being played for money and the user makes a move. This move will be added to a block and before the games is over. As anyone can see transactions the opponent may be able to work out the function that was called within the game. For the most part, the security was concerned with the use of the call function within solidity smart contracts. The call function handed over the control of the code to another contract.

The fact that all information is stored on a decentralized network provides positive and negative aspects. The positive is aligned to the consensus mechanisms that the majority in this case $\mathbf{51}\%$ is right and that if the majority are not bad

actors the integrity of information will remain intact. However, the fact that contracts and transactions or accessible by all users can certainly lead to them being manipulated and breached. This could be fixed if the contracts were stored in a permissioned environment where only specific users have access to them. As the scope of this application is only to provide a prototype with the security vulnerabilities discussed above have not been addressed. However, having knowledge of their existence would appropriate security measure would surely be implemented if the application were to move forward.

### 6.2 Performance Implications and Solutions

The consensus methodologies - Proof-of-Work (PoW) and Byzantine Fault Tolerant (BFT) are inversely proportional to each other with respect to the scalability spectrum. In other words, the performance of PoW-based blockchain is poor for large number of nodes and BFT-based blockchain performs better with small number of nodes (Vukolić, 2015). In our application, smart contract is deployed and as the application is built on Ethereum network, the only option we had, was to choose the PoW consensus algorithm for validation. Hence, the performance will depend on the number of nodes in the blockchain network.

### 6.3 Identity implications and solutions

Identity is defined as "The fact of being who or what a person or thing is" ("identity | Definition of identity in English by Oxford Dictionaries," n.d.). According to the bitcoin white paper, it is important for users identity to be protected to ensure transactions and financial status is not transparent to all users on the network as this would be invasive to their privacy (Nakamoto, n.d.). This was achieved by creating anonymous private keys. When Ethereum was created they needed to create a cryptocurrency to ensure that miners had an incentive to do their job. As without miners, there would be no decentralised network. Now that a cryptocurrency was made and that transactions need to be public in order to confirm their validity, they also needed to be anonymous like any monetary account is. In relation to our application, there will always be the question mark over identity within the Ethereum network as it is an integral part of its foundation. A solution to this would be to call upon specific off chain contracts to check a list of accepted addresses to transact with. This should reduce the likelihood of the contract being used by the wrong consumer. Another solution would be to host the application on a permissioned network providing more dept to access controls and consensus mechanisms.

**References:**


Angrish, A., Craver, B., Hasan, M. and Starly, B., 2018. A case study for blockchain in manufacturing: “FabRec”: A Prototype for Peer-to-Peer Network of Manufacturing Nodes. *Procedia Manufacturing*, *26*, pp.1180-1192.

Atzei, N., Bartoletti, M., Cimoli, T., 2017. A Survey of Attacks on Ethereum Smart Contracts (SoK), in: Maffei, M., Ryan, M. (Eds.), Principles of Security and Trust, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 164–186.

Balaji, S., 2012. WATEERFALLVs V-MODEL Vs AGILE: A COMPARATIVE STUDY ON SDLC. Vol. 5.

Boeing Company. (2014). *Boeing Current Market Outlook 2014-2033*. [online] Available at: http://787updates.newairplane.com/Boeing787Updates/media/Boeing787Updates/Commercial-forecast-presentation.pdf [Accessed 24 Apr. 2019].

Bourjade, S., Huc, R. and Muller-Vibes, C., 2017. Leasing and profitability: Empirical evidence from the airline industry. *Transportation Research Part A: Policy and Practice*, *97*, pp.30-46.

Buterin, V., n.d. A NEXT GENERATION SMART CONTRACT & DECENTRALIZED APPLICATION PLATFORM 36

Buzz, A., 2018. Why I Think Nick Szabo Is Satoshi Nakamoto, Even Though He Denies It. Medium. URL https://medium.com/@altcoinbuzz/why-i-think-nick-szabo-is-satoshi-nakamoto-even-though-he-denies-it-8c999841fbbb (accessed 4.21.19).

Chen, W.T., Huang, K. and Ardiansyah, M.N., 2018. A mathematical programming model for aircraft leasing decisions. *Journal of Air Transport Management*, *69*, pp.15-25.

Comparing flow charts and activity diagrams [WWW Document], 2014. URL undefined (accessed 4.20.19).

Dannen, C., 2017. Solidity Programming, in: Dannen, C. (Ed.), Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners. Apress, Berkeley, CA, pp. 69–88. https://doi.org/10.1007/978-1-4842-2535-6_4

Davis, A.M., Bersoff, E.H., Comer, E.R., 1988. A strategy for comparing alternative software development life cycle models. IEEE Transactions on Software Engineering 14, 1453–1461. https://doi.org/10.1109/32.6190

education, U.B., work, 2018. The Agile Methodology and Smart Contracts. UMKA Business, work, education. URL https://medium.com/@umka_/the-agile-methodology-and-smart-contracts-363248bdb268 (accessed 4.19.19).

Figl, K., Strembeck, M., 2014. On the importance of flow direction in business process models, in: 2014 9th International Conference on Software Engineering and Applications (ICSOFT-EA). Presented at the 2014 9th International Conference on Software Engineering and Applications (ICSOFT-EA), pp. 132–136.

Huertas, J., Liu, H. and Robinson, S., 2018. Eximchain: Supply Chain Finance solutions on a secured public, permissioned blockchain hybrid. *Eximchain White Paper*, *13*.

identity | Definition of identity in English by Oxford Dictionaries [WWW Document], n.d. Oxford Dictionaries | English. URL https://en.oxforddictionaries.com/definition/identity (accessed 4.24.19).

Importance of flowcharts when developing information systems, 2013. Eternal Sunshine of the IS Mind. URL https://eternalsunshineoftheismind.wordpress.com/2013/02/25/importance-of-flowcharts-when-developing-information-systems/ (accessed 4.20.19).

Keaveny, C. and Eustace, D., 2013. Aviation finance and leasing. *Offshore Investment*, *239*, pp.12-14.

Kehoe, L. and Hallahan, J. (2018). *Blockchain – a game changer in aircraft leasing?*. [online] Deloitte. Available at: https://www2.deloitte.com/content/dam/Deloitte/ie/Documents/Tax/ie-blockchain-a-game-changer-in-aircraft-leasing.pdf [Accessed 24 Apr. 2019].

Khan, P.M., Beg, M.M.S.S., 2013. Extended Decision Support Matrix for Selection of SDLC-Models on Traditional and Agile Software Development Projects, in: 2013 Third International Conference on Advanced Computing and Communication Technologies (ACCT). Presented at the 2013 Third International Conference on Advanced Computing and Communication Technologies (ACCT), pp. 8–15. https://doi.org/10.1109/ACCT.2013.12

Mahalakshmi, M., Sundararajan, D.M., n.d. Traditional SDLC Vs Scrum Methodology – A Comparative Study.

Manifesto for Agile Software Development [WWW Document], n.d. URL https://agilemanifesto.org/ (accessed 4.19.19)

Meetup. (2019). *BlockW Series 5: Blockchain meets Aircraft Leasing*. [online] Available at: https://www.meetup.com/BlockW-Ireland/events/btwldqyzdbzb/ [Accessed 24 Apr. 2019].

Nakamoto, S., n.d. Bitcoin: A Peer-to-Peer Electronic Cash System 9.

Pop, C., Pop, C., Marcel, A., Vesa, A., Petrican, T., Cioara, T., Anghel, I. and Salomie, I., 2018, September. Decentralizing the stock exchange using blockchain an ethereum-based implementation of the Bucharest Stock Exchange. In *2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP)* (pp. 459-466). IEEE.

Radic, D., 2018. Ethereum smart contract planning & specification. Dejan Radic. URL https://medium.com/@dejanradic.me/ethereum-smart-contract-planning-specification-339739adac54 (accessed 4.19.19).

Scanlan, D.A., 1989. Structured flowcharts outperform pseudocode: an experimental comparison. IEEE Software 6, 28–36. https://doi.org/10.1109/52.35587

Shukla, S., Thasmiya, A.N., Shashank, D.O. and Mamatha, H.R., 2018, September. Online Voting Application Using Ethereum Blockchain. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 873-880). IEEE.

Swan, M., 2015. *Blockchain: Blueprint for a new economy*. " O'Reilly Media, Inc.".

S zabo, N., 1997. Formalizing and Securing Relationships on Public Networks. First Monday 2. https://doi.org/10.5210/fm.v2i9.548

The Importance of Using UML for Modeling, 2014. URL https://www.w3computing.com/systemsanalysis/importance-using-uml-modeling/ (accessed 4.20.19).

Truffle Suite | Ganache [WWW Document], n.d. Truffle Suite. URL https://trufflesuite.com/ganache (accessed 4.21.19).

Vukolić, M., 2015, October. The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. In *International workshop on open problems in network security* (pp. 112-125). Springer, Cham.

Xu, Q., He, Z., Li, Z. and Xiao, M., 2018, December. Building an Ethereum-Based Decentralized Smart Home System. In *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)* (pp. 1004-1009). IEEE.