WORLDQUANT UNIVERSITY

MASTERS OF SCIENCE IN FINANCIAL ENGINEERING

DATA FEEDS AND TECHNOLOGY (C18-S3)

THOMAS DARLINGTON ADISENU

CEPHAS AKORMEDIE-TAY

DAVID KOFI GOGOVIE

DAVID KWASI NYONYO MENSAH-GBEKOR

GROUP WORK PROJECT – THIRD SUBMISSION

KEY CONCEPTS OF SMART CONTRACTS AND THEIR APPLICATION TO FINANCIAL ENGINEERING

GROUP 2-A

2019

ABSTRACT

The advancement of technology has brought the global economy into the hands of individuals making it convenient for business transaction between parties across the world. Blockchain technology is replacing the current traditional systems that have been in existence for over many decades.

This project seeks to take an in depth look into smart contracts used in blockchain technology and how it is replacing the current traditional systems with special use cases within financial markets.

We start by explaining smart contracts and how they function. We then compare them to traditional trading systems by giving their advantages and disadvantages. We then finish of by giving examples of use-cases of smart contracts with special focus on its application in finance.

Keywords: Smart Contracts/Blockchain, Use Cases and Application in Finance.

Smart Contracts:

Smart contracts are software codes that facilitate and enforce the conditions of an underlying agreement between two or more parties allowing decentralized automation in a transparent and conflict-free manner. They are said to make blockchains programmable and for creating decentralized applications (DApps). After the introduction of Bitcoin in 2008 and realizing the power of the underlying blockchain technology, smart contracts were then introduced in a new blockchain technology called Ethereum and have become well known and being used in other new blockchain technologies. Smart contracts are meant to remove the human interference as seen in most traditional systems or contracts. Smart contracts also seek to remove the amount of paper work in traditional systems where paper work is involved.

The name *smart contract* can be misleading, since they are not smart, but rather do as they have been programmed and they may not have anything to do with the law. The smart contract term was introduced by Nick Szabo in 1994 on the idea of *self-executable or digital contracts converted into codes to be run on a decentralized ledger or blockchain* rather than a paper-based contract involving a third-party. Ethereum is the most popular blockchain technology with this concept of smart contracts. Ethereum came up with the idea of turing complete machine known as Ethereum Virtual Machine (EVM) which runs the smart contract codes.

How it functions (Life Cycle):

There are generally four phases in the life of a smart contract namely ***creation, deployment, execution and completion*.** The creation of smart contracts is an iterative process involving multiple parties such as stakeholders, lawyers and software engineers. Sometimes the stakeholders could take on the role of the lawyer or software engineer. After *agreement is reached on discussions and negotiations on the obligations, rights and prohibitions of the contract*, lawyers help the parties draft an initial contractual agreement using *natural language* which are then converted into smart contracts in the form of *computer codes* by software engineers using computer programming languages such as *solidity* in the case of Ethereum. This conversion process normally follows a similar software development process such as *design, implementation and testing*. The high-level solidity code which looks like java-script must be compiled into native low-level bytecode for the EVM to run.

The validated code of the smart contract is then deployed or copied to an address on the main blockchain production platform and can no longer be amended due to the immutability property of the blockchain. *New contracts would have to be deployed if changes are required*. On the blockchain, the smart contract can keep track of the accounts of digital assets of the parties involved since every account has an associated wallet. Execution of the smart contract is done automatically once conditions set forth in the smart contract are met. The execution of the smart contract normally involves updating of account balances of parties involved and other states as well as generating transactions to be mined and committed to the ledger by miners. A typical smart contract code involves declarative statements with logical connections using if-statement programming syntax. To complete the process, these updated states are saved onto the blockchain with the transfer settlement of digital assets of the parties involved. Most of these life cycle activities are carried out with the help of tools such as remix editor for writing the solidity code, MetaMask to provide a web3 browser access to the blockchain which is a decentralized network as well as an account with some amount of Ether (currency for Ethereum) to be used to pay for the gas price when deploying or running the smart contract code.

Now, taking a typical example, supposing you want to sell a property of yours, say a house and apart from complexities in communication with multiple parties and the paper work involved, there is the risk of fraud, as you would require the services of an intermediary like say a real estate agent to deal with the paperwork, markets and negotiations and since it is hard to trust these agents, the agents provide escrow service to transfer funds from one party to the other and when the deal is finalized, you pay both the agent and the escrow service some percentage which leads to extra cost.

By using smart contracts, which work on condition-based principles represented in code logic as simple if-else statements, ownership issues would be resolved as smart contracts are self-executable and would automatically transfer the property to the buyer only when certain conditions set forth in the contract are met like receipt of money by seller. Both the money and the property can be stored in the smart contract in the form of tokens to serve as an escrow service and this is available for view by all parties and since the money transfer transactions can be seen by all network participants, the chances of fraud is reduced and more so there is no need to put your trust in an intermediary since all the functions of the intermediary have been coded up into the smart contract.

Advantages:

Smart contracts possess a lot of features which serve as an advantage over traditional systems such as *transparency, no miscommunication, autonomous, efficiency, cost-effective, reliability, persistent, trustless, convenience* etc.

Transparency is a basic characteristic of blockchain in general and since smart contracts run on a blockchain, its terms and conditions or rules are visible to all parties involved. They are frequently checked to avoid disputes and ***miscommunication*** issues later. Also, smart contracts are ***autonomous***, meaning they do not rely on middle men to carry out execution of the contract. Also, the trouble of having to go through the middle man to process documentations is drastically reduced in a smart contract when compared to traditional systems as smart contracts are just pieces of code to be executed by the computer which helps improve the ***efficiency*** of business processes. Smart contracts are ***automated*** and so are not only ***faster*** and ***cheaper*** but also ***avoid the errors*** that come with manually completing multiple forms making them more ***reliable***. They are also, normally run through multiple test before deploying to the main network and are continually tracked by all participants on the network. The decentralized nature of blockchain and the use of data encryption makes it more ***secured*** as compared to traditional systems. In other words, the blockchain helps prevent denial of service attacks (DoS) which traditional systems are incapable of providing as traditional systems are centralized and mostly run on a single node. Developers of smart contracts can restrict ***access permission*** of users to functions within the smart contract so that only authorized users can call those functions.

The ***immutability*** of blockchain makes smart contracts ***persistent***, ***traceable*** *and* ***auditable***. In other words, they cannot be changed once deployed to the network. Smart contracts are ***cost-effective*** since there are no middle men and so there is a cut down on *administrative and service costs* that could otherwise be used to pay the middle men. Another property of blockchain technology that gives it another advantage over traditional systems is that blockchains are ***trustless*** in the sense that they do not require a central authority to manage contract executions. Smart contracts running on the blockchain are ***automatically triggered and rely on the trust of the whole system*** with the help of consensus algorithms. This raises the confidence level of people in smart contracts as they are decentralized, and the rules are binding on the parties involved with a shared agreement which automatically executes when conditions set forth are met reducing the requirement of litigation and courts. This *autonomous* nature of smart contracts makes them more ***convenient*** to use as they do not require physical presence of the parties.

Disadvantages:

Although smart contracts have lots of potentials over traditional systems, there are some challenges or disadvantages such as *unconvincing, prone to bugs or errors, implementation cost, inflexible, lack of contractual secrecy, uncertainty in regulation etc.*

The technology looks ***unconvincing*** to the public at large partly due to lack of adequate insights and especially during these early years of its inception raising lots of skepticism about the technology and its workings looking at the broad market base and complexities. People see it as a high-risk adventure and so are used to their traditional way of document writing.
Smart contract codes are written by humans and so are prone to ***bugs or errors*** if not programmed well to suite the intentions of the parties. Without checks and balances, there can be vulnerabilities in the smart contract code creating loop holes for attackers to steal digital currencies such as reentrancy where an interrupted function code is recalled again. This example was depicted in the decentralized autonomous organization (DAO). Lack of code optimization also result in huge execution cost due to the concept of gas in Ethereum where you pay more for using more computing resources to run your smart contract code and so since smart contracts require good programming, it is essential to have a good coder on staff to make fail-safe smart contracts and adopt the company's internal structure for the blockchain technology and this adds up to the ***implementation cost***. More so, though immutability is a good property of blockchains on one side, on the other side, it raises flexibility concerns of the system making smart contracts and other data stored on the blockchain ***inflexible*** and cannot be altered by the parties in case of change of mind. It turns out that, third parties who are meant to be replaced by smart contracts ***do not disappear entirely*** as they take on new roles such as advisory roles. Blockchains like Ethereum are public and smart contracts are run on all nodes in the network posing some level of ***lack of contractual secrecy*** or ***lack of privacy*** that may be required by the parties involved in the smart contract although information processed by the smart contract must be decrypted. ***Uncertainty in regulation*** is another challenge posed by smart contracts as it raises concerns as to how smart contracts will respond to the law and how government taxes can be imposed on smart contracts.

Other issues like ***scalability*** and ***latency*** have received attention in recent times and there are layer 1 solutions, for instance sharding and layer 2 solutions such as off-chain transactions being implemented to improve the blockchain system. The consensus algorithm is also being upgraded from proof-of-work to proof-of-stake for the Ethereum blockchain.

Use Cases:

Smart contracts can be used wherever contracts exist between parties or where relationship needs to be managed and therefore have a lot of variety of use cases in many different industries and fields such as finance, supply chain, e-commerce, insurance, real estate, vendors, asset management, securing copyright content, employment agreement, voting, internet of things (IoT) etc. but however, we take a deeper look into its application in finance.

To be more useful in the real world, for instance in trading or *finance* in general, smart contracts need a way of accessing ***off-chain external information*** and since blockchain cannot directly fetch data, there are real time data feeds for blockchain known as ***oracles*** which can be used as middleware between data and the smart contract. For example, ***Chainlink*** which is a decentralized oracle can be used to provide highly secure and reliable oracle data to smart contracts by performing computations on multiple data sources such as websites or sensors (IoT) before writing into the smart contract code.

Using smart contracts in financial markets, counter-party risk in international trade can be eliminated. For instance, with increasing globalization, ***remittances*** in finance have become very necessary and remittances using traditional systems is still expensive and slow. By building a network of banks for international settlement in the blockchain technology like *Ripple*, efficiency in remittances can be improved and through Chainlink oracle, reliable currency conversion data can be provided to the smart contract. By aggregating data from multiple data sources or exchanges with different ***market price data*** for assets, Chainlink oracle can feed smart contracts with most up-to-date and trustworthy prices in real time in a decentralized way for ***algorithmic trading*** in finance. Chainlink also has external ***adapters*** for cryptocurrency market data from CoinMarketCap, CryptoCompare, Binance, Brave New Coin and Kaiko though this can also be done in traditional markets using tools like Bloomberg and Thomson Reuters. Again, with smart contracts, assets such as gold, oil, real estate and stocks can be ***tokenized*** to maintain certain price based on market data fed into smart contracts through Chainlink oracle to enable *trading*. ***Maker DAO***, one of the leaders in ***decentralized finance (DeFi)*** already uses about 14 oracles as reference prices for the Maker system. *Smart contracts can also be used, as is already the case, to trigger payments in Bitcoins, Ethereum, XRP, stable coins and other digital currencies as well as off-chain transactions on lightning networks that makes transactions fast.*

The blockchain technology is still in its infancy and quite volatile as many find it difficult to trade them with their fiat currency and so to meet current increasing demand for payments, smart contracts should be given access to many types of payment options and banks can be urged to share the APIs with smart contracts to enable automatic account to account payment once certain conditions are met and this could decrease the cost of online payment systems such as VISA and MasterCard. Smart contracts can also take advantage of international payment messaging standards like SWIFT.

Financial derivative products are also areas worth mentioning. Derivatives are contracts between parties whose value depends on the changes in price of an underlying asset. By feeding the underlying price changes into the automated smart contract code on the blockchain through Chainlink oracles for execution, these derivative products in the form of smart contracts can be verified on the blockchain and enabling settlement of payments without the need for an intermediary who could bias the system for their own gain.

Conclusion:

Indeed, there are numerous applications of smart contracts, and though decentralization is good property of blockchain technology, complete decentralization may be a difficult task to achieve considering scalability and latency. Smart contracts and its underlying blockchain technology is still in its infancy and despite its advantages over traditional systems, it faces challenges that require frequent updates. With these challenges addressed, smart contracts will gain full global adoption and would stand out as good innovation to replace centralized traditional systems.

References

Pdfs.semanticscholar.org. (2020). [online] Available at: https://pdfs.semanticscholar.org/a743/a04db7e973d15da0d389a8b477dcdeeac092.pdf [Accessed 10 Feb. 2020].

Arxiv.org. (2020). [online] Available at: https://arxiv.org/pdf/1912.10370.pdf [Accessed 10 Feb. 2020].

Investopedia. (2020). *Are Smart Contracts the Best of Blockchain?*. [online] Available at: https://www.investopedia.com/news/are-smart-contracts-best-blockchain/ [Accessed 10 Feb. 2020].

Blockgeeks. (2020). *Smart Contracts: The Blockchain Technology That Will Replace Lawyers*. [online] Available at: https://blockgeeks.com/guides/smart-contracts/ [Accessed 10 Feb. 2020].

Hackernoon.com. (2020). *Everything You Need to Know About Smart Contracts: A Beginner's Guide*. [online] Available at: https://hackernoon.com/everything-you-need-to-know-about-smart-contracts-a-beginners-guide-c13cc138378a [Accessed 10 Feb. 2020].

EngineerBabu Blog & Success Stories. (2020). *How is Blockchain Revolutionizing Banking and Financial Markets?*. [online] Available at: https://engineerbabu.com/blog/blockchain-revolutionizing-banking-financial-markets/ [Accessed 10 Feb. 2020].

QuantInsti. (2020). *Automated Trading Systems: Architecture, Protocols, Types of Latency*. [online] Available at: https://blog.quantinsti.com/automated-trading-system/ [Accessed 10 Feb. 2020].

Medium. (2020). *Chainlink: How Smart Contracts can be used in Finance*. [online] Available at: https://medium.com/@jeroen.hesp/chainlink-how-smart-contracts-can-be-used-in-finance-44a8cbdec66e [Accessed 10 Feb. 2020].

Medium. (2020). *How Smart Contracts for Finance Will Make Stock Markets Faster, Cheaper and Less Error-Prone*. [online] Available at: https://medium.com/@adamdavidlong/smart-contracts-for-finance-clearing-and-settling-securities-trades-6a774b28106f [Accessed 10 Feb. 2020].

Rua.ua.es. (2020). [online] Available at: https://rua.ua.es/dspace/bitstream/10045/78007/1/Smart_Contracts_from_a_Legal_Perspective_Utamchandani_Tulsidas_Tanash.pdf [Accessed 10 Feb. 2020].

Appendix:

Comparing Smart Contracts to Traditional Contracts.

	Smart Contracts	**Traditional Contracts**
	Transparent	Not transparent
	No miscommunication	May have miscommunication
	Autonomous	Requires central authority
	Efficient (minutes)	Not efficient (1-3days)
	Cost-effective	Expensive
	Reliable	Not reliable
	Persistent	May not be persistent (can be modified)
	Trustless	Trust
	Convenient (Virtual Presence)	Not Convenient (May require physical presence)
	Escrow may not be necessary	Escrow necessary
	Difficult to regulate	Easy to regulate
	May not be bound by law	Bound by law
	Limits human interference	Needs human intervention
	Works on a distributed ledger	Works on client/server technology
	Still in infancy	Matured