Data Feeds & Technology Group Project - Submission 3

- Bosco Keown bosco.keown@gmail.com
- Jeremie Sabban jeremie_sabban@hotmail.com
- Raza Akhtar raza.akhtar@me.com
- Aliaksandr Panko panko.aliaksandr@gmail.com

April 2019

Approx 1500 words excluding headers, contents, and references.

Introduction

In this report we initially describe what smart contracts are and how they work, with particular focus on the Ethereum network. We further describe their advantages and disadvantages relative to traditional trading solutions particularly with regard to risk and regulatory concerns. Finally, we identify and discuss a specific appropriate use case for smart contracts.

Smart Contracts

What is a Smart Contract?

A Smart Contract is software program, i.e. a bundle of self-executing functional computer code that incorporates different elements of a legally binding or a non-legally binding (but nevertheless enforceable) contract such as offer, acceptance, and consideration. This code can also execute terms of a contract.[1]

Essentially, a Smart Contract combines “enforceability” with “automation”, ensuring the contract is enforceable terms of ensuring legal rights and obligations are adhered to, or of enforcing the actions of the code. In other words, if we take a paper contract and automate parts of it in code so it can be enforced without being tampered with by either party (or a third-party), we have the elements of a Smart Contract.

“*A smart contract is an automatable and enforceable agreement. Automatable by computer, although some parts may require human input and control. Enforceable either by legal enforcement of rights and obligations or via tamper-proof execution of computer code.*”[2]

[1] "A Primer on Smart Contracts - Commodity Futures Trading Commission." Accessed April 7, 2019. https://www.cftc.gov/sites/default/files/2018-11/LabCFTC_PrimerSmartContracts112718.pdf.
[2] "Smart Contract Templates: foundations, design landscape and ...." Accessed April 7, 2019. https://arxiv.org/abs/1608.00771?context=cs.

How Smart Contracts Work in the Ethereum Network

Ethereum is a distributed blockchain-based open-source application platform, backed by Ether (ETH), a cryptocurrency which is mined via a proof-of-work consensus algorithm. Ethereum allows smart contracts to be written in Turing-complete languages like Solidity, LLL, Mutan, Viper, and Serpent.

Serpent is an older language very similar to Python. Security vulnerabilities have made it unattractive and, since September 2017, Solidity has been the preferred language of smart contract development on Ethereum. Solidity is a contract-oriented, high-level language - syntactically similar to JavaScript - designed to compile to bytecode to execute within the Ethereum Virtual Machine (EVM). With Solidity, developers can create automated rule-based contracts that specify ownership, format of transactions, and transitions between different contract states. Viper has fewer features than Solidity but better auditability and is Pythonic in nature. Mutan and LLL are rarely used nowadays.

Within the Ethereum network, transactions and computations are subject to fees measured in *gas* units (note: this does not refer to fuel but is an arbitrary unit name). Transactions specify a variable called *gasLimit* - the maximum amount of gas to use while executing the transaction. Every transaction has its own *gasLimit* set by the developer. Additionally, every transaction has a *gasPrice* which is paid to miners in ETF per gas unit mined as a mining reward. In order for a transaction to take place and be valid, the account must be sufficiently funded to cover *gasPrice*gasLimit* or an *out-of-gas* situation occurs that can lead to a wide number of contract vulnerabilities.[3]

There is also the concept of *gasUsed* which represents compute cycles and storage utilized by a transaction. Miners may target the verification of transactions with larger fees to increase their profits. The idea of having gas resources in smart contracts has three drivers. The first driver is that up-front payment for gas by the party proposing the transaction prevents wasting miners' resources by ensuring they are rewarded appropriately. The second driver is that gas fees the use of valuable replicated storage resources that are a necessary part of a functional blockchain. The third driver is capping the amount of compute resources per transaction to some upper limit, in order to prevent denial of service or other attacks that might target endlessly long executions.[4]

In summary, a Smart Contract, coded in a Solidity, Serpent, or Vyper program, effectively represents a real paper contract. The program contains a series of triggering

[3] "Smart Contracts for Machine-to-Machine Communication: Possibilities ...." Accessed April 7, 2019. https://arxiv.org/abs/1806.00555.

[4] "GASTAP: A Gas Analyzer for Smart Contracts." Accessed April 7, 2019. https://arxiv.org/pdf/1811.10403.

conditions which may involve transactions that consume or distribute gas units and other assets such as access to a particular resource or set of resources. It is deployed and distributed to each node in the Ethereum network, receiving an address in the process. The contract is subsequently accessed on the network via this address.

Each node will hold copies of the transaction history and the contract history. When all the pre-conditions for each function in the Smart Contract are met for a transaction to occur and result in completed changes of state - making extensive use of cryptographic primitives including hashes, digital signatures and zero-knowledge proofs, plus “oracles” which are services that feed data or other information into the Smart Contract from the internet - the overall contract executes and settles; miners validate and include the information in the next block of the blockchain according to the consensus algorithm.

Advantages

Consider the lifecycle of a typical economic transaction. Such a transaction has three phases: formation, execution, and settlement.

A Smart Contract can add a lot of value to this process in terms of standardization, security and transparency (such that the outcome of the contract is fully deterministic and predictable), speed and cost benefits, certainty, business innovation and regulatory innovation. It does this in various ways. For example: standardizing contract code and reusing it, plus automation of transactions and elimination of costly manual intervention, speeds up time to settlement.

The fact that a distributed ledger (blockchain) is used improves security as transactions are both encrypted and immutable. Automation of business process flow via Smart Contracts can lead to new and innovative products and improved business models. Smart Contracts may automate regulatory reporting at appropriate times which is a major benefit.[5]

Contracts have addresses in the network and can be directly addressed. This allows them to be referenced, mapped and listed. They can hold tokens or coins and be duplicated or versioned easily.[6]

Disadvantages

Smart Contract technology faces limitations and challenges. By using them, parties commit themselves to be bound by the rules of the underlying code. While a standard paper contract outlines its terms - typically through legal means - a Smart Contract enforces its terms

[5] "A Primer on Smart Contracts - Commodity Futures Trading Commission." Accessed April 7, 2019. https://www.cftc.gov/sites/default/files/2018-11/LabCFTC_PrimerSmartContracts112718.pdf.
[6] "Implementing a financial derivative as smart contract." Accessed April 7, 2019. https://arxiv.org/abs/1903.00067.

via a relationship with cryptographic code. It follows that it may be exposed to bugs and malicious code.

Writing Smart Contracts is hard as it requires an “economic thinking” perspective which many technical programmers don’t possess. Money leaks may occur where there is a failure to use proper cryptographic primitives. The possibility exists for mistakes and bugs relating directly to the Ethereum platform itself, or inadequacies in the DSL (domain-specific language) used to describe the contract or translate it.[7]

Allowing lawyers and developers to work together, producing bug-free code for smart contracts, finding ways to update Smart Contracts taking into account legal changes, and finding solutions for information hiding in certain Smart Contracts that require privacy are currently some of the biggest challenges.

Use Case

Despite their disadvantages, Smart Contracts have potential to be a game-changer in certain areas. One use case is in financial derivatives which have potential for misuse and threaten to the economy when they go wrong or are misunderstood, e.g. the Collateralized Debt Obligations (CDOs) that led to the 2008 financial crisis.

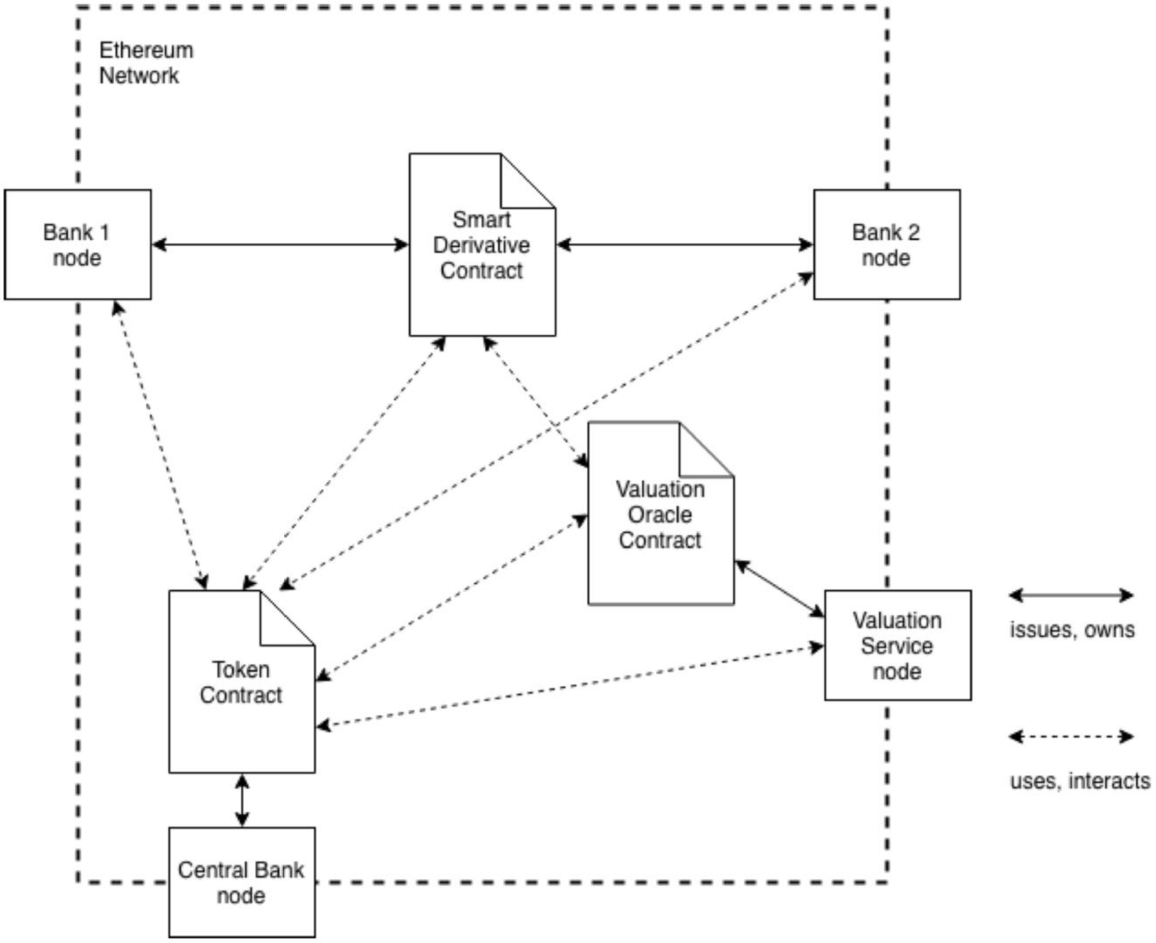
We could represent a Smart Derivative Contract between two banks in an Ethereum network as being driven by an application that interfaces with nodes in the Ethereum network.

There will be a Valuation Oracle; a service that provides derivative valuations to the network from applications off the blockchain (since most derivative valuation software is necessarily complex and expensive to run on the blockchain network). The Valuation Oracle will access a blockchain node so that it can address the Valuation Oracle Contract that represents its work - this will have functions to request and act on valuations from the Valuation Oracle. There will also be a Token Contract that manages and keeps track of token balances of either party in the contract, connecting to a Central Bank node in the network (presumably for regulatory and management purposes).

Both the Token and Valuation Contracts may be Smart Contracts also, but the important contract will be the Smart Derivative Contract representing the derivative and its lifecycle. The illustration below shows interactions between the various entities, services and contracts mentioned.[8]

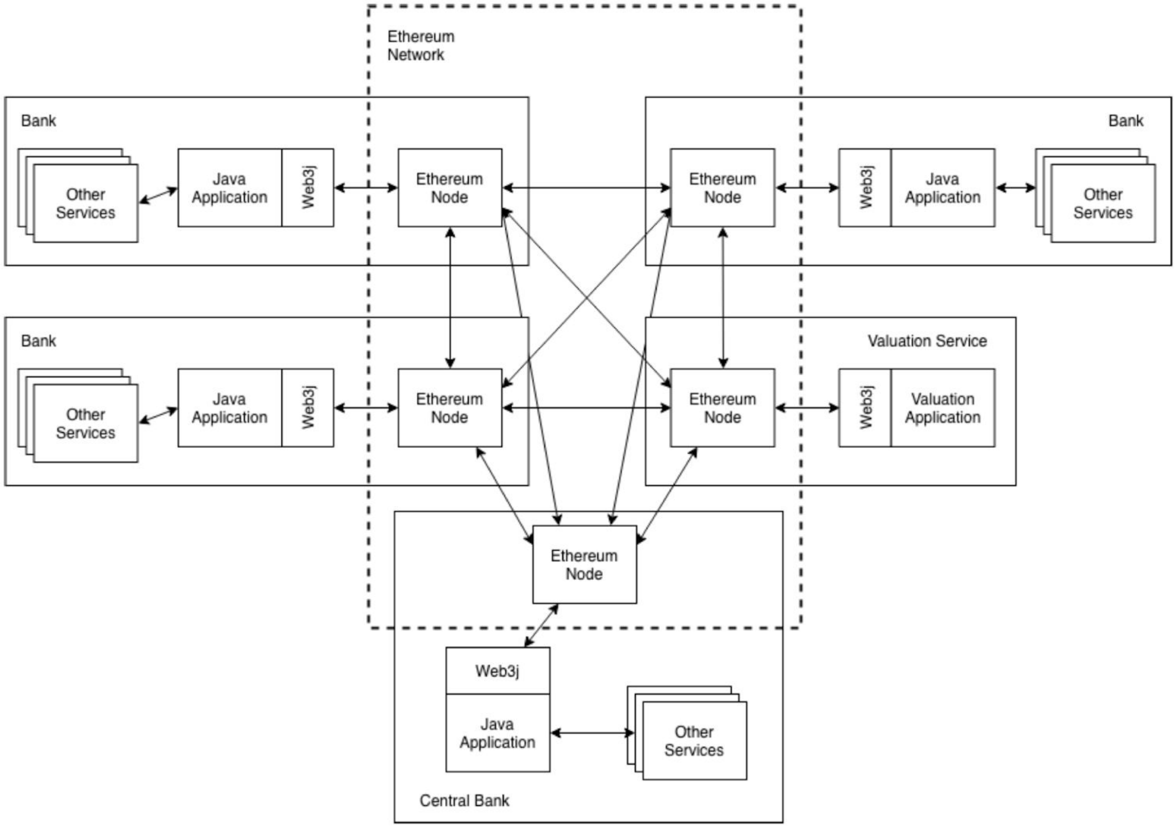
[7] "Trust in Smart Contracts is a Process, As Well - Financial ...." Accessed April 7, 2019. https://fc17.ifca.ai/wtsc/Trust%20in%20Smart%20Contracts%20is%20A%20Process%20as%20Well%20-%20Slides.pdf.

[8] "Implementing a financial derivative as smart contract." Accessed April 7, 2019. https://arxiv.org/abs/1903.00067.



Tokens are represented by the token contract and are issued, owned and provided by the Central Bank node. Token contracts contain all token functions and keep track of balances of participants using the tokens. The following diagram shows a potential architecture for implementing such a solution.[9]

[9] "Implementing a financial derivative as smart contract." Accessed April 7, 2019. https://arxiv.org/abs/1903.00067.



Conclusion

We have explained what Smart Contracts are, their purpose, and how they function and operate on the Ethereum Network in particular. We have also discussed advantages and disadvantages of Smart Contracts over traditional trading solutions, describing how they might be used for implementing Smart Derivative Contracts.

There is clear potential for the use of Smart Contracts as an adjunct to many existing financial and trading operations even where there is adherence to existing tools. Smart Contracts can eliminate counterparty risk, while automating and enforcing previously manual interactions between parties to a financial transaction.

There are many obstacles to be overcome, but the momentum is there to surmount them. We believe that Smart Contracts will play a major part in the financial industry and contribute greatly to market efficiency in the future.

References

1. "A Primer on Smart Contracts - Commodity Futures Trading Commission." Accessed April 7, 2019. https://www.cftc.gov/sites/default/files/2018-11/LabCFTC_PrimerSmartContracts112718.pdf.
2. "Smart Contract Templates: foundations, design landscape and ...." Accessed April 7, 2019. https://arxiv.org/abs/1608.00771?context=cs.
3. "Smart Contracts for Machine-to-Machine Communication: Possibilities ...." Accessed April 7, 2019. https://arxiv.org/abs/1806.00555.
4. "GASTAP: A Gas Analyzer for Smart Contracts." Accessed April 7, 2019. https://arxiv.org/pdf/1811.10403.
5. "A Primer on Smart Contracts - Commodity Futures Trading Commission." Accessed April 7, 2019. https://www.cftc.gov/sites/default/files/2018-11/LabCFTC_PrimerSmartContracts112718.pdf.
6. "Implementing a financial derivative as smart contract." Accessed April 7, 2019. https://arxiv.org/abs/1903.00067.
7. "Trust in Smart Contracts is a Process, As Well - Financial ...." Accessed April 7, 2019. https://fc17.ifca.ai/wtsc/Trust%20in%20Smart%20Contracts%20is%20A%20Process%20as%20Well%20-%20Slides.pdf.
8. "Implementing a financial derivative as smart contract." Accessed April 7, 2019. https://arxiv.org/abs/1903.00067.
9. "Implementing a financial derivative as smart contract." Accessed April 7, 2019. https://arxiv.org/abs/1903.00067.