FINHAVEN

Compliance for Security Token Issuance and Trading

Abstract

In this paper, we present a framework and approach to implement regulated capital markets using blockchain technology. Our goal is to ensure our system is compliant with current and future regulations and that regulators can readily confirm this. Our approach includes a new language for financial transactions and we demonstrate how it can be used to for security token issuance and exchange. While the idea of a new language may sound overambitious, we take a pragmatic approach by building tools on top of existing infrastructure and tooling.

Overview

We propose an approach to address the issue of demonstrating regulatory compliance with respect to the creation, sale, and future resale of tokenized securities on blockchains. We do this by modelling the participants and processes in a way that makes it straightforward for regulators to confirm compliance. By using models of known entities, regulators can understand the interactions of any proposed market systems. These same models are then transformed into executable code that carries out these transactions as smart contracts to produce a compliant regulatory market system.

In the next section, we present outstanding issues around trust and compliance in applying blockchain to securities and concerns around regulation. In the following section, we introduce a new framework for programming financial transactions on the blockchain, a standard for security tokens, and a strategy to build an exchange that will be able to comply with global regulations.

Background

In the past year, initial coin offerings (ICOs) have raised billions of dollars of investment. This boom quickly drew the attention of securities regulators across the world, who raised concerns that this type of fundraising was likely subject to laws and regulations that have been put in place for good reason. Indeed, many ICO projects have not delivered on promises and some early investors have been left with little or no rights or value. It was not long before there was a crackdown on raising funds in this manner, with tokens regarded as securities being banned from most exchanges and issuers receiving notices from regulators that they must comply with regulations.

Interestingly, although regulators are not opposed to this new form of investment, they are struggling with how to regulate blockchain assets and their default response is to shut down anything which may not meet regulations. This uncertainty places a big risk on anyone participating in this new blockchain economy. What is needed is a clear endorsement from regulators that this mode of business can be compliant with regulations.

Although regulations continue to evolve, securities regulations have been written in terms of the roles of participants in traditional markets. This creates confusion when considering blockchain based securities where traditional market participants like clearing or transfer agents, are no longer present or needed.

From our inception, Finhaven has predicted this would be the outcome and we have been focused on producing a blockchain architecture for securities that can clearly demonstrate that assets issued and traded comply with security regulations.

Smart Contracts

It is worth understanding the broader context of smart contracts as they offer more than the ability to issue a token or an ICO. One of the less obvious innovations when Bitcoin was introduced was the idea of programmable money, or what are known as smart contracts. Bitcoin offered a highly restrictive, security focused scripting language that allows simple contracts to be encoded with transactions. This supports contracts like escrow, multiple signatures, and so on. Ethereum came along and provided a much richer, feature focused scripting language that allowed arbitrarily complex contracts to be written, but along with that, many security issues.

Being able to encode logic and law into transactions is a very useful innovation, as long as we can program it to work as expected. One of our primary motivations is to improve the situation where “you get what you expect” from a smart contract. Establishing trust in code is a requirement if want to proceed with a system that is able to enforce laws and regulation. Although the original vision of smart contracts [Szabo-97] spoke of “specification through clear logic” the current state of smart contract languages is something that is far more obtuse, even to those who have experience with the languages. For example, Bitcoin has a security maximalist approach [Script] and Ethereum has a feature maximalist approach [Solidity] but it would be a stretch to say either is specification through clear logic.

We believe that verifiability is one of the main problems that remains to be solved with blockchain technology and which is necessary if we are to achieve regulatory compliance. Although blockchain technology allows the verifiability of approving transactions, we remain very far from Szabo’s original objective:

> ‘So our second objective is verifiability, the ability of a principal to prove to an adjudicator that a contract has been performed or breached, or the ability of the adjudicator to find this out by other means. The disciplines of auditing and investigation roughly correspond with verification of contract performance.’

Although blockchain technology can verify transactions there is currently little accountability and knowledge of the counterparties in a transaction. For example, being able to tell if an ICO is run by serial scammers or a reputable company, or if an investor is a criminal who is using the blockchain to launder money, fund terrorism, or evade tax, or that the funds you send as an investor have legal rights that the recipient is obliged to respect. These are challenges that we wish to address and confront, rather than avoid.

Clarity

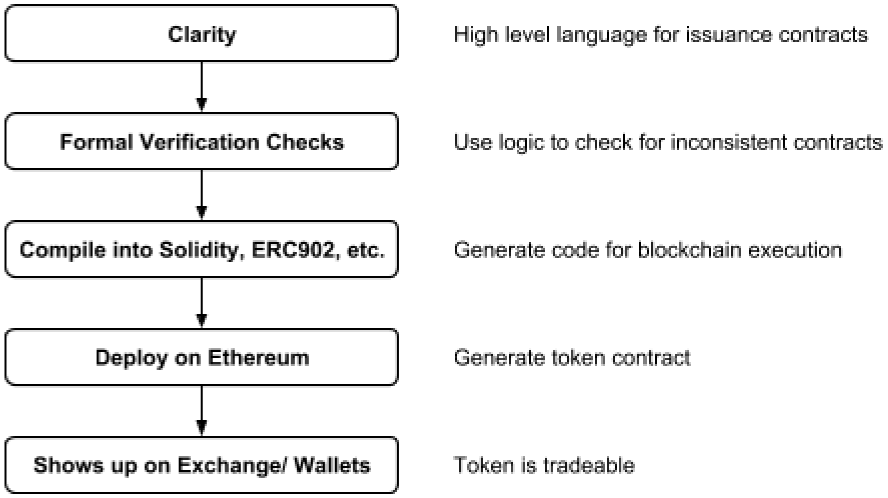
Clarity is a high-level domain-specific and declarative programming language for financial transactions. This means Clarity more closely resembles a specification unlike more general purpose languages like Solidity or other blockchain smart-contract languages. By focusing on a restricted domain of financial transactions, Clarity is designed to be verified, both in terms of correctness as well as intent. The focus on verifiability allows both regulators and investors to confirm compliance and will make smart contracts safer for everyone who uses them.

Principles

- **Security-focused language** - Clarity is a new programming language for blockchain finance that is more flexible than Bitcoin but more restrictive than Ethereum. This language will be translated into existing blockchain infrastructure and operate on multiple blockchains.
- **Formal verification** - Clarity will be designed in a way to make it suitable for formal verification, much in the same manner that other critical software systems are programmed (airplane control, nuclear reactors, etc).
- **Readable by non-experts** - smart contracts need to be able to be understood by multiple actors within financial markets, not just the software developers who write them. For a smart-contract to be trusted, it must be possible to understand what it does. Therefore our language will strive to be clear and understandable.

How It Works

The process in which Clarity is used works as follows. First the entities and rules are modelled using domain-specific language constructs, such as asset, owner, transfer, allowed, and so on. The result is a Clarity model that maps directly to the real world situation it is capturing. The second step is to apply formal verification and other checks to the model which can bring to the surface any inconsistencies or unintended scenarios that might be allowed. Such problems can then be remedied by improving the model. Once the model has been verified as correct it is transformed into a specific execution language, for example smart contract code in Solidity, which is then compiled into EVM bytecode. The transformed model can use specialised mappings from each of our specific language constructs to specialised implementations, such as those being implemented by ERC902, which is described below in the Security Token section.


Clarity
High level language for issuance contracts
Formal Verification Checks
Use logic to check for inconsistent contracts
Compile into Solidity, ERC902, etc.
Generate code for blockchain execution
Deploy on Ethereum
Generate token contract
Shows up on Exchange/ Wallets
Token is tradeable

Extensible

We have only started work on Clarity, but we believe this framework to be very powerful and extensible. By separating the modelling from code execution, we are able to clearly capture and reason with the actual constraints. Not only does this mean that Clarity code is easier for humans to use and understand, it is also much easier to apply automated analysis methods as compared to attempting to apply them at a lower level, e.g. bytecode operating on the EVM. We must still ensure that our translation layer is correct, but to us, this seems more attainable than a general purpose validation tool that works on any Solidity program.

We would also like to note that declarative models can be composed together. For example, this can allow regulations encoded by the laws of country A to be combined with the laws of country B. In such a case, we would associate a country to each participant, enforce the appropriate laws, and the system would be able to enforce laws that span international boundaries. We believe that composability is one of the most powerful aspects of our approach.

A final benefit of modelling at this layer is that we will be able to support different execution environments and new blockchain technology without having to rebuild Clarity models. Instead, we focus on building and testing our translation layers which will bring that benefit to all models without having to change anything. Aside from saving engineering effort, this would mean that a regulator who has certified a Clarity model as complying to regulations would be assured that the model is still correct on new blockchain platforms. Indeed, unless there is a need to execute the Clarity model on a blockchain, we can also translate it to traditional execution environments like the JVM or Node.

Security Token

A security token is a ‘digital share’ which offers the holder certain rights. We are exploring how to create a new interoperable standard to represent security tokens and how to ensure and prove that these tokens satisfy regulations. In particular, KYC/AML, transfer of ownership, and other regulations require extending existing token infrastructure.

Compliance

The tokenization of assets has wide application, not least of which is financial instruments such as securities. Most jurisdictions have placed legal constraints on what may be traded, and who can hold such tokens which are regarded as securities. Broadly this includes KYC and AML validation status, restricted transfers depending on the type of investor, but may also include time-based spend limits, total volume of transactions, and so on.

Finhaven proposes demonstrating regulatory compliance with respect to the creation, sale, and future resale of tokenized securities. We do this by modelling the participants and process in a way that makes it easy for regulators to assess compliance. Using digital analogues of familiar entities will allow regulators to understand the interactions of new market systems.

As an example, consider the situation where someone is selling an equity share, or ‘stock’. We represent the seller as one agent and the buyer as another agent. In order to complete the sale, two transactions need to occur: the buyer needs to transfer money to the seller and in return, the seller transfers the share to the buyer. In order to transfer ownership of the share, the seller might just hand the stock certificate to the buyer, providing example of an unregulated exchange of value. However, in most countries, you must use a registry to ensure that only the rightful owner can sell the stock. This requires a third party, where another agent checks the corporate registry and the seller’s proof of identification, the buyer’s identification, and then records the buyer as the new owner (the registrar likely collecting some tax/fees at this time).

If we model this as three agents and each transaction as messages between the agents, it becomes much easier to write rules that ensure compliance with regulations.

For example, there may be a rule

```
IF (SELLER IS OWNER OF SHARE)
    THEN ALLOW TO TRANSFER OWNERSHIP OF SHARE
```

and then to execute the transfer

```
IF (OWNER AGREES AND TRANSFER FEES PAID)
    THEN BUYER BECOMES THE NEW OWNER OF SHARE
```

The current mechanism for enforcing such transactions in blockchain markets is equivalent to handing over the stock certificate. Some elements are missing or may not have a clear mapping to how it would have happened before.

We use a distributed agent model to represent market participants. Each agent communicates with other agents, by sending and receiving messages, and each agent has rules that must be followed. A software agent is a program that acts on behalf of a user. It can be thought of as a software version of a robot or drone, and may be autonomous or intelligent, but may also be controlled by a user. Often the direction and objectives are set by a human, but much of the behaviour and interaction is defined by software. For example, a trading bot may be applied to a particular market with a trading strategy that has been authored by a human.

Regulators and sanctioned third-party compliance agencies need some way to link off-chain compliance information such as identity and residency to an on-chain service. The application of this design is broader than legal regulation, encompassing all manner of business logic permissions for the creation, management, and trading of tokens.

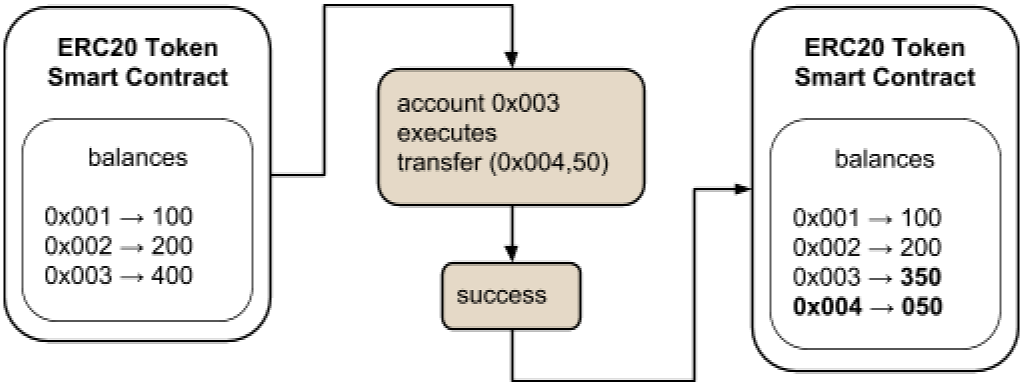
Tokens are encoded as smart contracts: code that runs on a public Ethereum network node. The ERC20 token protocol has become the most widely used smart contract interface for Ethereum tokens, many of which are likely a form of securities. To ensure compliance with applicable securities laws, security tokens (and tokens broadly) need a method to ensure that regulations and other policies are upheld. The control over some of these policies and data may be spread over many individuals, companies, consortia, and regulatory bodies, and need a unified way of interacting autonomously. Currently any such controls on tokens are being applied in a very ad-hoc way, with no common mechanisms or standards.

Transfer Standard - ERC902

We have submitted ERC902 (Token Validation) which proposes a standard protocol for validating if an action is permitted by a user on a token. This has clear application for regulatory compliance as it can be used to enforce things like restricted transfer. *This is not a new kind of token*, but rather a way to confirm that actions such as token transfers between particular users are allowed based on some criteria. These validations may be very simple checks, or aggregate to form graphs of permissions across multiple parties.

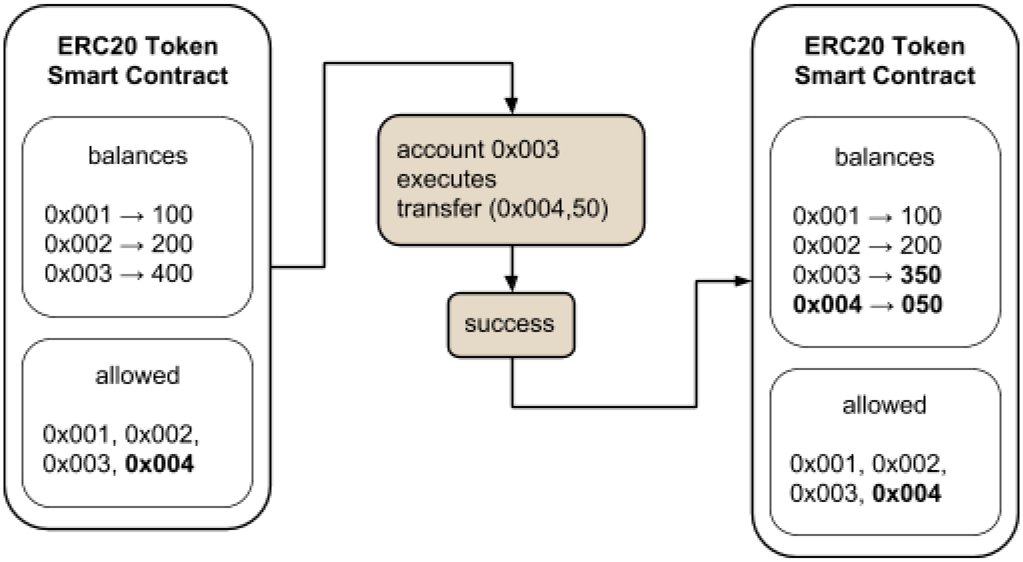
This proposal is compatible with any token, but was initially designed with securities in mind. It should be noted that this is not intended a general roles/permission system, but is specifically aimed at tokens. In any system, the more that you know about your use case, the more assumptions a design can make, and the more powerful your function can be for those cases. By restricting ourselves to token transfers (ex. ERC20 or ERC777), we are able to optimize validators for these use cases, and can make the API small, useful, and extensible. Further, Solidity lacks abstraction at the language-level (e.g., generics, polymorphic returns, &c), and so this design must be especially concrete. The proposal's two "check" functions are modelled on token actions, but may be used in other interesting and innovative ways, both financial and nonfinancial.

Having a record of token ownership on the public blockchain is valuable for many use cases. While most people think of tokens or coins as "in" an owners account, they are actually recorded as a balance in a token smart contract. This is more like a spreadsheet per token than a bank account with multiple currencies. The token contract lists each owner's addresses and the balance of tokens that that address controls.



Standard ERC20 Transfer

A transfer of tokens between two addresses calls the `transfer` function of a token contract. A sender address requests a transfer of some amount from their account to the recipient account. In the most common case, if the sender has a high enough balance to satisfy the transfer amount, the tokens are debited from the sender and credited to the receiver.


ERC20 Token Smart Contract
balances
0x001 → 100
0x002 → 200
0x003 → 400
allowed
0x001, 0x002, 0x003, 0x004
account 0x003 executes transfer (0x004,50)
success
ERC20 Token Smart Contract
balances
0x001 → 100
0x002 → 200
0x003 → 350
0x004 → 050
allowed
0x001, 0x002, 0x003, 0x004

Security Token Smart Contract with “Allowed“ Whitelist

Our core proposal is that security tokens check with an on-chain validator before performing transfers or issuances. This is generally done inside the transfer function itself, and prevents the transfer from executing if it fails this validation.

An example of a simple validator is a whitelist directory. This smart contract contains a list of vetted users, and may only be updated by a single controlling account (the contract's "owner"). Any contract may request a check that a user has been vetted by this whitelist. For an ERC20 transfer request, the contract will ask this whitelist contract to check that the user is allowed to transfer funds. If they are, then the transfer if free to continue. Otherwise the transfer fails and no balances are changed. Creators of validator smart contracts may use a variety of on- and off-chain processes to determine whitelist membership. This system can additionally be used to make tokens completely non-transferable, time-lock tokens, allow only certain amounts to be transferred, check if a user has reached some transfer limit, and so on.

Trading Security Tokens

Blockchain assets have been traded on exchanges for many years, but the boom of "ICOs" attracted the attention of securities regulators across the world. Regulators correctly noted that many of the "utility" tokens being offered are actually speculative securities, so should have applicable regulatory controls on issuance and trading. As most traditional exchanges were unable to satisfy these regulations, this led to either these security tokens being removed from the exchange or broad restrictions on the countries where users may trade. Another approach involves sidestepping these rules altogether by using a decentralised exchange (DEX) to trade, however, avoiding exchange regulations does not address the underlying issue of satisfying the regulators; it only pushes the risk onto the issuer and investor.

This uncertainty has spooked many investors and issuers and made them hesitant to use blockchain assets for fear of either losing their investment or an expensive fine or lawsuit from a regulator. The enforcement of these rules was no surprise to experienced investors, who are eager to see a compliant solution to issuance and trading. Furthermore, investors are keen to break out of the expensive and slow mechanisms of traditional exchanges, so there is an appetite for a new market system.

What is needed is a marketplace that fully supports security tokens. There are several requirements to achieve this. The most obvious is the ability for buyers and sellers to be able to trade security tokens, with visibility on any restrictions and rights that involve each token. A secondary but critical objective is to be able to demonstrate and enforce securities regulations so that the exchange is allowed to do business.

We believe the solution to this cannot be piecemeal. We need to build a securities exchange that supports regulation through the entire process from issuer, investor, trading, and payments. Our approach is to assemble all of these components and build the necessary interoperational protocols to integrate with both the current regulated trading systems and participants, and also provide a system that is easy to evolve and adapt to new regulatory environments.

Our Approach

Rather than trying to disrupt the fundamental process that current securities markets follow, we are going to take a path of meeting regulations where they are today. This means securing appropriate licences, modelling market participants (backed with licenced participants in the real-world) and enforcing regulations showing a direct mapping to our model and how it complies to those regulations. Clearly this is no small undertaking.

We will use Clarity, our contract specification language, to create software models of each entity and regulation. As we proceed to represent many different kinds of markets, participants and jurisdictions, our models will become richer. Unlike the restrictions of traditional markets to include global investors, our models can support international trade and provide a bridge between regulatory environments in different countries. For example, we envision a system where an investor selling an asset in one country can be assured that regulatory compliance is met through our system when trading with another investor in a different country with foreign rules.

Conclusion

Our approach is designed from the core to be multi-chain and able to adapt and evolve, but with an emphasis on being able to continuously demonstrate regulatory compliance. Instead of asking regulators to define rules on blockchain token models, we take the opposite approach; we provide tools that allow blockchain market innovators to fit existing regulations into their own model.

The Finhaven platform is a proposed model for a regulated securities exchange, with verifiable compliance and security tokens to represent secure digital assets. The regulated securities exchange is a blockchain enabled bourse for issuance and trade of tokenized securities/security tokens.

While blockchain has the potential to remake global capital markets, significant technical challenges remain to transform capital markets. We plan to address these challenges through a global technology platform that embraces trust, safety, and compliance.

Selected Bibliography

- Szabo-97: the original version of smart contracts http://journals.uic.edu/ojs/index.php/fm/article/view/548/469
- Bitcoin Script https://en.bitcoin.it/wiki/Script#Script_examples
- Ethereum Solidity http://solidity.readthedocs.io/en/develop/solidity-by-example.html
- Domain-specific language https://en.wikipedia.org/wiki/Domain-specific_language
- Declarative programming https://en.wikipedia.org/wiki/Declarative_programming
- Constraint programming https://en.wikipedia.org/wiki/Constraint_programming
- General purpose validation tool https://www.ideals.illinois.edu/handle/2142/97207
- How to write a financial contract https://www.microsoft.com/en-us/research/wp-content/uploads/2000/09/pj-eber.pdf
- Software agent https://en.wikipedia.org/wiki/Software_agent
- ERC20 token protocol https://github.com/ethereum/EIPs/issues/20
- ERC902 token validation https://eips.ethereum.org/EIPS/eip-902
- ERC777 token standard https://github.com/jacquesd/ERC777/blob/master/README.md
- Investor Bulletin: Initial Coin Offerings, July 25 2017 https://www.sec.gov/oiea/investor-alerts-and-bulletins/ib_coinofferings
- Removing tokens from exchanges upon recognition as unregulated securities https://info.shapeshift.io/blog/2017/08/17/shapeshift-and-tokens-securities
- Decentralized Exchange (DEX) https://blog.0xproject.com/a-beginners-guide-to-0x-81d30298a5e0

- Regulatory enforcement for ICOs https://www.technologyreview.com/s/610513/the-next-generation-of-icos-will-actually-have-to-follow-the-rules/
- The LMAX Architecture https://martinfowler.com/articles/lmax.html
- Nex white paper https://neonexchange.org/pdfs/whitepaper_v1.1.pdf
- The Open Agent Architecture: A Framework for Building Distributed Software Systems http://www.ai.sri.com/~cheyer/papers/oaa.pdf
- Multi-agent systems https://en.wikipedia.org/wiki/Multi-agent_system
- Simulation of a trading multi-agent system http://ieeexplore.ieee.org/document/972041/?part=1
- Legal entity identifiers https://en.m.wikipedia.org/wiki/Legal_Entity_Identifier
- The impact and potential of blockchain on securities transaction lifecycle https://dspace.lboro.ac.uk/dspace-jspui/bitstream/2134/24271/1/The%20Impact%20and%20Potential%20of%20Blockchain%20on%20the%20Securities%20Transaction%20Lifecycle%20-%20Mainelli%20and%20Milne%202016.04.11%20v2.1.pdf
- Custodian https://www.investopedia.com/terms/c/custodian.asp#ixzz59WKjZQUU
- The Bitcoin Lightning Network https://lightning.network/lightning-network-paper.pdf

Forward-looking Statements

Please note that in this presentation, references are made to the Finhaven exchange, including references to a "regulatory compliant" exchange or language similar in nature. The concept of a regulatory-compliant exchange, and the language used herein to describe it, represents the vision and ultimate goal of Finhaven. However at this time, the Finhaven exchange remains in development, no securities regulator has reviewed or approved the same, and it is has not been determined to be compliant with the securities laws or regulations of any jurisdiction.