



Suruhanjaya Sekuriti
Securities Commission
Malaysia



**Capital Market Architecture Blueprint in a
Decentralised World**

November 2018

1. Introduction: Distributed Ledger Technologies – The Infrastructure for More Decentralised Markets?

For most readers of this paper, probably the most distinctive feature of the capital market is the national stock exchange. For Malaysians, this is Bursa Malaysia. However if we cast our memories slightly further back in history, what we know today as a single centralised exchange are often the result of consolidation between multiple smaller markets. Investors were already trading securities with each other, where buyers found sellers via intermediaries such as brokers, or within small private pools before central exchanges came into place.

The capital market rapidly embraced centralisation as a means of greater efficiency – exchanges, clearing houses, custodian banks and trade repositories are prime examples of this phenomenon. Centralisation has brought about greater price discovery and liquidity, certainty of clearing and settlement, more orderly market operations as well as greater pre- and post-trade transparency. Adopting centralised models have delivered significant benefits – Today, markets are more efficient and resilient than ever before, and a decent amount of product innovation still take place in this space.

However as markets continue to evolve together with new innovative products, as well as the increasingly varied demands of investors, we foresee the emergence of multi-tier markets, where there are markets catering to specific needs and niches. This is already happening as we observe how the unlisted and OTC markets have evolved compared to listed markets.

In this case, a one-size-fits-all solution to market structures might not be the most appropriate. For example, centralisation would require a high degree of product homogeneity and standardisation, which would not work well in the OTC markets space which value bespoke and tailored solutions. Unlisted and OTC markets also operate in a bilateral manner, sometimes requiring greater privacy and exclusivity, whereas listed markets emphasise broader fairness and transparency.

The challenge is to consider how we can build decentralised, highly flexible market infrastructures to support the unlisted and OTC markets, while still adopting some of the best practices from centralised markets, i.e. transparency. The invention of blockchain and distributed ledger technologies, with its promise of distributed trusted networks, should precipitate a re-think of how it can be applied to the capital market.

With the unlisted & OTC markets, there are inherent weaknesses which can be improved upon:

1. Unlisted markets are less liquid and are dependent on intermediaries to connect buyers and sellers
2. The lack of transparency creates uneven markets where smaller players are sometimes disadvantaged
3. This also creates inefficiencies in matching and price discovery
4. The instruments traded in these markets are often bespoke and tailored
5. Due to the bilateral nature of such deals, the process of settlement is often complex, having to rely on intermediaries such as brokers, custodian banks, escrows, etc to maintain trust between both parties

At first glance, DLT has the potential to provide solutions to overcome these weaknesses:

1. The features to build a decentralised, yet private network means buyers and sellers can connect directly with each other without the need for intermediaries.

2. All transactions performed on a DLT-based network will be automatically recorded on the ledger with is highly auditable and immutable, while there are methods to still maintain individual transactional privacy.
3. With the right permissions, a participant on the network can view the entire transaction history of a particular instrument.
4. “Smart Contracts” on the DLT can be implemented to codify any tailored or bespoke form of products, and are self-executing based on pre-defined criteria.
5. The nature of the DLT being a decentralised trusted network means complex bilateral settlement processes can be simplified – a participant does not need to know or trust the other participants, but only need to trust the integrity of the ledger itself to ensure that transactions are executed properly.

Which led to the SC embarking on this project to proof point the technical implementation feasibility of DLT at this juncture to serve as the underlying market infrastructure for unlisted & OTC markets. We had hope, through the project, to take stock of the possibilities of the technology, both in its current form today, as well as try to peek at what is potentially possible in the future.

It is important at this juncture to point out that DLT is still in its nascent stage of development. There are several concerns with the technology itself that would limits its feasibility in selected use cases. For example, it still has limitations in scaling to a higher transaction per second capacity means that it is not yet suited in the trading space where speed and throughput is paramount. However, it is not unreasonable to anticipate that the continued development of this space will close these technical gaps eventually, thereby expanding its usefulness.

At the same time, DLT enthusiasts should caution against prescribing the technology as a panacea for all ills – there are benefits to certain levels of centralisation that would be more effective and efficient than attempting to decentralise everything. Rather we foresee both centralised and decentralised market structures co-existing with each other, with the most appropriate technologies implemented to match the needs of the market. This way, we would be able to enable the capital market to cater to a broader and wider set of market needs.

The contents of this report are not meant to be prescriptive in nature. Rather, we have tried our best to present each chapter as a series of design decisions and highlight key considerations when intending to build such a market. As a regulator, the SC holds a view as to the proportionate standards required to operate a regulated market, but such requirements need not apply in a less regulated space. We hope that our industry participants will come along with us on this journey in order to explore the feasibilities of this innovative technology to solving some of the weaknesses in our markets. Lastly, while within the pilot we have used equity crowdfunding (ECF) as the prime use case of an unlisted or OTC market, and this could be unfamiliar to some, we believe that the same capabilities and standards are sufficiently generalised and can be applied to other unlisted or OTC-type assets or products.

The SC would like to thank the Neuroware.io team as our technology partner for this pilot and for their continued support during the early ideation through to the end of the project, and the willingness to challenge and push the boundaries of what could be possible within the scope of the pilot.

2. Key Challenges of Existing Unlisted & OTC Markets

Within the capital markets, listed products are issued and traded in formal, regulated venues. The obvious examples of these are listed public equities, listed derivatives contracts, exchange-traded funds (ETFs), and foreign currency.

The opposite of listed products is of course unlisted and OTC products, which typically do not have established formal venues to facilitate the issuance, trading, clearing and settlement of such products. They are typically issued, bought, traded and settled bilaterally between parties. For example, bonds and sukuk, OTC swaps, and even assets such as property would fall under this category. While these products can benefit from being highly tailored in nature, the lack of established market structures gives rise to a distinct set of challenges.



Unlisted markets are less liquid and are dependent on intermediaries to connect buyers and sellers



A lack of transparency results in an uneven market, which disadvantages smaller players



A lack of shared data creates inefficiencies in price discovery and matching of supply to demand



The instruments traded in these markets are often bespoke and tailored



Complex and onerous processes are required to maintain trust during settlement

Over the years there have been initiatives made to alleviate the issue of transparency within unlisted & OTC markets – most notably the creation of mandated trade repositories where transactions are reported to a single central database. The Bond & Sukuk Information Exchange (BIX) is a prime example, providing a record of all information related to Malaysian bonds and sukuk. However such repositories only obtain records post-event after a period of time has passed, sometimes in summarised form and are dependent on the integrity of its participants.

3. What We Hope to Accomplish: DLT-based Unlisted & OTC Markets

Cognisant of these challenges, it was clear that there could be structural improvements made to unlisted & OTC markets to increase its transparency and efficiency. However we also wanted the distributed, bepoke nature of such markets to be preserved.

This was the primary appeal for building these sorts of market with DLT – as a peer-to-peer network would still allow the market to retain these unique characteristics, avoiding the rigidity of a formal market venue, while at the same time allowing for open innovation in transparent, efficient and hopefully more cost-effective solutions for the future.



Single system of record where all market participants have a full copy of the relevant information, which is constantly kept in sync in near real-time.



At the same time the appropriate privacy controls to preserve the confidentiality and privacy of parties and transactions, where required.



Self-executing transactions utilising “Smart Contracts” to codify and execute conditions upon specific triggering conditions being met.



User-friendly experience allowing for participants to audit the entire history of activities and reducing reconciliation efforts.



Use of cryptographic techniques to encrypt and secure the data, including key managements systems to control access to the market.



Regulatory node allows a full view of all market activities in real-time, enabling better supervision and reducing reporting from participants.

With this in mind, an initial scope and conceptual blueprint¹ was crafted to explore the feasibility of such a market. While part of the scope has evolved throughout the course of the project, the end-objective has remained the same – to prove that it was feasible to construct such a market on a DLT network. Of the possible unlisted products, the team chose equity crowdfunding (ECF)², owing to the familiarity of the team with this product³, and the relative “white space” of this new product, which would allow us to innovate and test potential solutions without being overly encumbered by legacy.

However building on DLT also presents its own unique challenges. As much as possible we wanted to push the boundaries of what could be done on a decentralised architecture. This included:

¹ See next chapter for further details.

² For readers who might not be familiar with ECF, a brief explanation of the product is provided within the appendix.

³ Neuroware has prior experience working with one of the registered ECF operators in building out their platform on blockchain.

Challenges with DLT



Access and permissions

- Allowing participants access to the network
- Limiting the functions of participants only to what is intended
- Allowing access to data



Decentralisation of everything

- Decentralised credential management
- Decentralised data storage



Trust the math

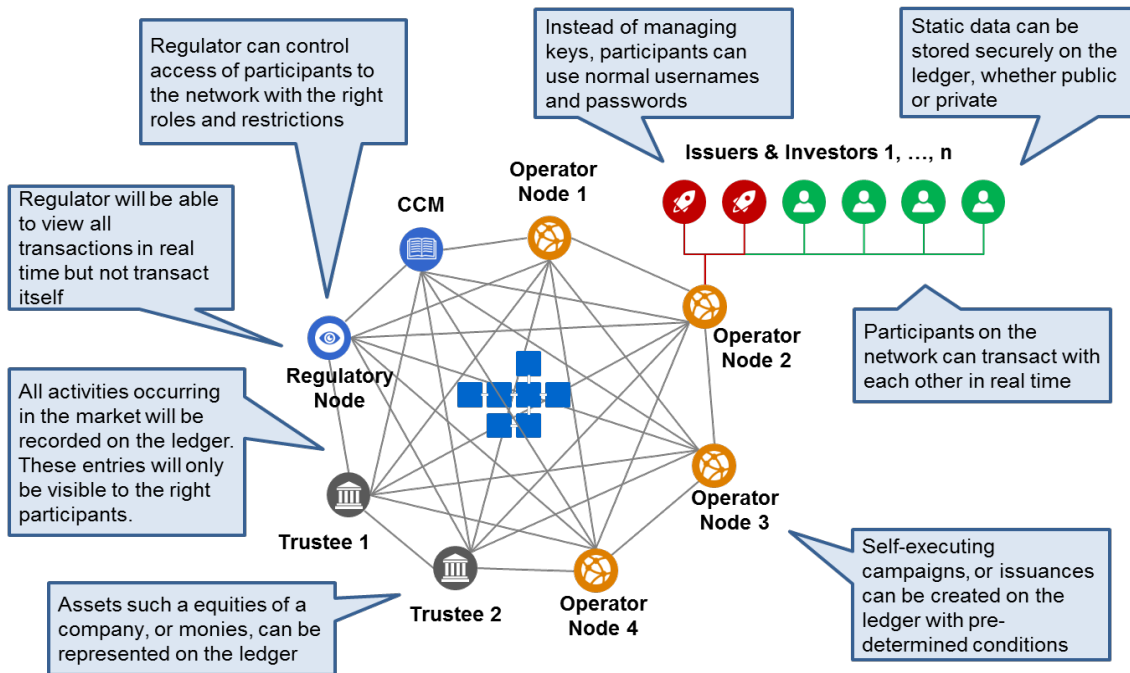
- Records are auditable and tamper-proof
- Self-executing transactions and transaction certainty

While DLTs came from public blockchains which are built to be openly accessible, regulated markets required permissions and gatekeeping. We intended to explore how we could do so in both private networks as well as public blockchains so that the roles and functions of each participant mimics what we currently have in real life.

As we looked to build a decentralised system, we also wanted to minimise the use of centralised databases which needs to be maintained by a central trusted party. Without a central trusted party, it was important to understand how could credentials be managed, and whether we could really store all data on a ledger.

Finally the concepts of trust and security of distributed ledger systems will need to be assessed holistically. While enthusiasts will tell you to “trust the math”, how far this is true needed to be explored. At the same time we also wanted to explore the concept of a distributed ledger being a trust machine, and able to help achieve some market efficiencies such as self-executing transactions and escrow functions.

4. Unpacking the Conceptual Blueprint



From a conceptual perspective, the diagram above illustrates the multiple participants in a market. It was decided early that only the regulator, operators and trustees needed to exist “on-chain”. To mimic what was today happening in the real-world, issuers and investors would participate through operators and would largely not interact directly with the blockchain itself.

The call-outs describe the key functional elements performed by each role in the market, which are distinct from each other. Collectively, these roles and functions form the broad scope of what we hope to demonstrate within the pilot.

5. Laying the Foundational Building Blocks for a DLT-based Market

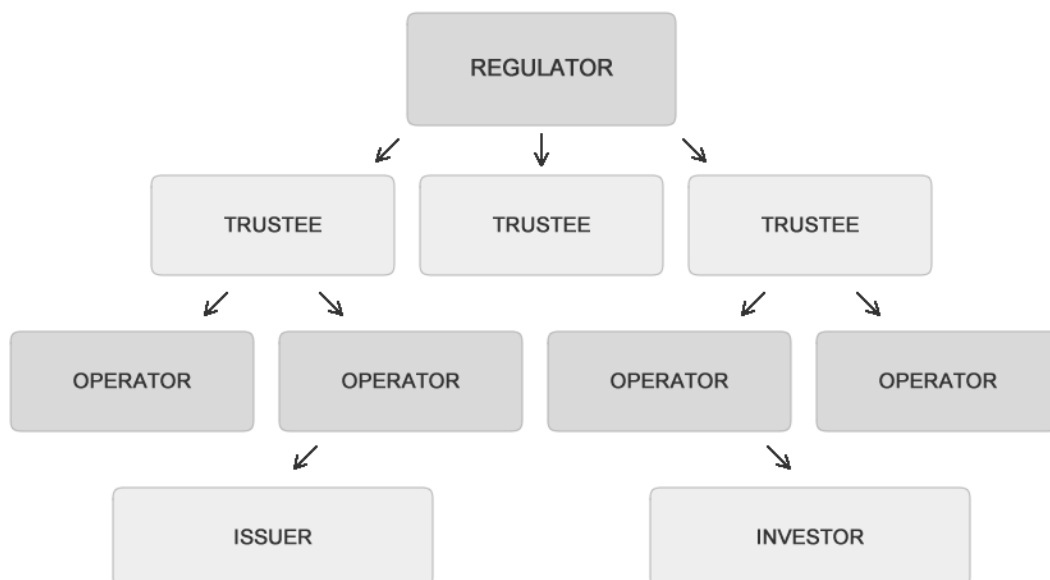
A. Access, Transparency and Privacy on the Market

Before piecing together the market structure itself, there are a few foundational building blocks which had to be explored and considered upfront. The first consideration is the concept of gatekeeping access to the market. Only entities that have been licensed or registered with a regulator should be allowed access to the market, and in turn only persons or entities who have been KYC-ed by these regulated entities should be allowed to transact within the market. In effect, the network needs to be permissioned on multiple levels depending on the delegation of duty.

Closely linked to the concept of access to the market is the concept of access to data, or data privacy. While a regulator would want to have full view of all activities and transactions occurring within the market, this privilege is not extended to all market participants. In most cases, participants in unlisted & OTC market transactions prefer a certain level of privacy. Thus data privacy is a key feature which should be taken into consideration when designing the architecture of the network.

A particularly elegant solution to both problems on the DLT is the concept of hierarchies. Put simply if there was a parent key or master contract used to generate a series of child keys and derivative contracts on a ledger, the parents could access all activities conducted by all of its derivatives, but each child could still retain privacy between each other, especially when multi-signature processes are also used. Moreover, through the use of interconnected smart contracts, parents could limit the scope of actions each child can conduct; delineating the roles between different participants in the market.

In our case, the following basic hierarchical structure for the market is easily apparent:



Readers would notice that the hierarchy closely mimics the governance hierarchy which is apparent in our markets today:

1. The regulator permissions the access of market participants such as trustees and operators to the market
2. From a governance perspective, the trustees are higher up the hierarchy from the operators as they are responsible to safeguard the assets and monies of investors and issuers
3. The operators conduct the necessary due diligence on issuers and investors, including processes such as KYC, AML and CTF checks⁴ before allowing them in to the market

From a privacy perspective due to the hierarchical structure:

1. The regulator would be able to view everything
2. Trustees are able to view all activities conducted by the operators which sign up for their services, as well as the activities conducted by the issuers and investors signed up with these specific operators
3. The individual operators will be able to view the activities of their issuers and investors, but not of their competitors

Creating such hierarchies maintains the original balance of privilege and responsibilities of each participant. Parents undertake supervisory and surveillance activities on its children, and can suspend or revoke child capabilities⁵. While some would argue that this leaves the child at the mercy of its parent, it does allow the parent to enforce rules within their “hierarchy”. In theory such rules would ideally be encoded into the network itself and self-enforcing, but exceptions do occur, which is where we rely on responsible parties to take appropriate action.

⁴ Know-your-customer (KYC), anti-money laundering (AML) and counter terrorism financing (CTF)

⁵ Do note that this does not allow the parent to override or overwrite the actions of the child, unless such feature is explicitly build into the network.

B. Generating and Managing Keys

We have talked about generating keys, which represent accounts on a ledger. However, for non-technical users – generating and managing keys can be a nerve-wrecking experience. Below is an example of a key, which appears to be a set of random alphanumeric strings:

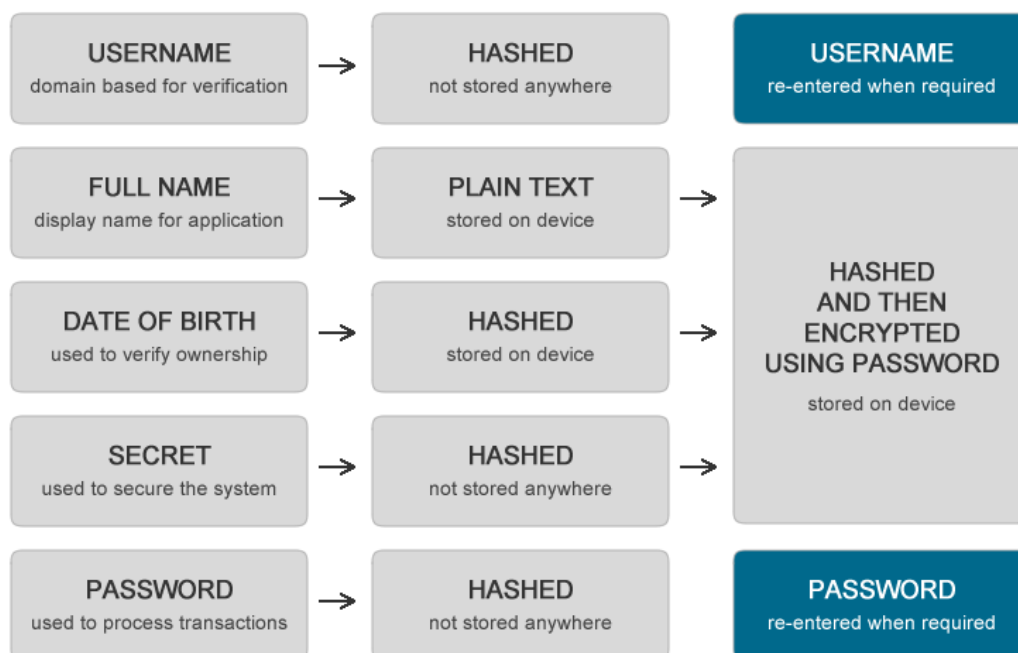
```
// An example of an address - which is derived from a public key:
1GzBZ7eK6wzNjplWt6AxHo73kJL2tzoErq

// An example of a private key:
LlwinVkoRmxMdHKBwssx33Z9ZEuXeJleP9CVYvnNn4TdYA32GsWY

// An example of an extended HD master private key:
xprv9s21ZrQH143K2Ywhg9bhZ5nd31t3EbXsg8v28gkKjSm9PA3PiZ89dWW6YKxWZa2pgTuErQ6
5K46KGVfulxCRBCK3Ppd465QGtH7TmxAEiLv
```

Not only are they almost impossible to remember, but more importantly, since they represent accounts, there is no way to recover access to the accounts should the private key be lost. While this is excellent from a security perspective, it is less practical for non-technical users as these keys can be easily lost or stolen.

To help overcome these issues, we introduced an optional process for generating deterministic keys. Unlike most traditional keys, deterministic keys are not generated at random, but instead use certain data points within the process. For example, when generating a key for an issuer, we can use the name of the company, its company registration number and its phone number to generate the key. A random element, known as a “salt”, is added to the process to provide a certain degree of randomness and security. With the deterministic process, the issuer can always generate keys as long as it has the required details (the company name, registration number and phone number).



This process allows systems to be built in such a way that they do not need to store those keys anywhere, which also means that they cannot be stolen. This is made possible by only ever being generated at the point of being used. This is done by filling-in the missing information needed to recreate those keys. This configuration is attractive from a security perspective –public keys can be safely stored, shared or integrated with DNS records, with private keys used to sign transactions only generated on demand when required, without needing to store or transmit those keys.

By introducing usernames and passwords to the process of generating deterministic keys, we are able to provide a more traditional user experience. Such traditional username and password implementations in the past would have required a central database storing such credentials, which becomes a prime target for hackers who wish to gain access to the network. The technical team proposed an alternative solution that ties everything together using DNS based identities.

The DN-Key Protocol⁶ provides a way to utilize DNS TXT records to broadcast public keys - whilst also enabling a more robust method for generating trusted hierarchical deterministic keys between multiple stakeholders. Imagine a user with various wallet addresses for multiple currencies. Rather than needing to remember each of the individual addresses (or public keys) and which is used for which network, a user can merely share their DN-Key instead.

Through this method, there is no longer a need for a central database storing usernames and passwords, thereby eliminating a potential point of failure and target for hackers to obtain credentials to access the network. However this method is not without its downsides and limitations. This pilot project utilizes DN-Keys in the following unique ways:

- Broadcasting multi-signature redeem scripts
- Improving security in the creation of deterministic keys
- Enabling the enforcement of pre-approved or multi-signature accounts

When utilizing native multi-signature accounts such as those offered by Bitcoin, it is necessary for each participant with an active key to also have access to the redeem script that is generated once all the keys are combined. It can sometimes be difficult to communicate and store this script with multiple parties, which also provides several opportunities for DN-Keys.

```
// An example of a Bitcoin 2-of-3 Multisig Redeem Script:
524104a882d414e478039cd5b52a92ffb13dd5e6bd4515497439dff691a0f12af9575fa349
b5694ed3155b136f09e63975a1700c9f4d4df849323dac06cf3bd6458cd41046ce31db9bdd5
43e72fe3039a1f1c047dab87037c36a669ff90e28da1848f640de68c2fe913d363a51154a0c
62d7adea1b822d05035077418267b1a1379790187410411ffd36c70776538d079fbae117dc3
8effafb33304af83ce4894589747aee1ef992f63280567f52f5ba870678b4ab4ff6c8ea600b
d217870a8b4f1f09f3a8e8353ae
```

In order to pre-approve accounts whilst also providing account recovery, DN-Keys can also be incorporated into the key generation process - but doing so also enforces the use of specific crypto-wallets that adhere to these processes. However, please note that for the sake of more simplified

⁶ <http://dnkey.org/>

demonstrations⁷, not only have the DNS settings been simulated via configuration files, but the multi-signature account recovery options have also been disabled. This means that within the open demonstrations, a failure to remember your username, secret or password will result in a total loss of account control, which is why we have opted to use a private network.

Notwithstanding the method proposed by the technical team, developers should design the security access of their respective networks and applications in their own, respective context. Fundamentally how security is designed will depend on multiple considerations, including but not limited to the type of use case being deployed, the sensitivity of the data being stored on the network, the functionality limitations for each user – all of which is constantly at odds against a seamless experience for users.

⁷ Please refer to the Project Castor micro-site: <https://castor.my>

C. Storing data in the Blockchains

The final technical building block we were looking to solve in the early part of the project was the concept of storing data on the ledger. Part of the benefits commonly cited for the technology is the ability for data stored on the distributed ledger to be constantly kept in sync, tamper proof and easily auditable. The fact that the data is replicated across multiple nodes also means that personal infrastructure can be removed – instead relying on the network of nodes to host the data for you. These are attractive characteristics which should encourage implementations to store as much data as possible “on-chain”.

However there are limitations to this method. While technically any type of data can be stored on the distributed ledger, each ledger has a limit to how much data can be stored in each block, commonly known as the blocksize limit. It would for example be impractical⁸ to try to store media files on a ledger given the limit and burden to the overall network. Approaching this problem would require some thinking around what forms of data are being generated by the system, and which of the data was needed to be constantly updated and broadcasted, as opposed to data which was more static in nature, and hence less important to be constantly broadcasted through out the network.

From an unlisted and OTC markets perspective, there are clear types of data which belonged into each respective category. Data of activities whereupon a new product is issued, and then subsequently traded, are obviously key data which was meant to be constantly updated and broadcasted when it occurs. Moreover these activities were meant to occur “on-chain”, either through smart contracts or simple peer-to-peer transactions, meaning that this data would be captured in its entirety on the ledger by default.

There are however other types of data which exist on the market. For example the details of the stakeholders, such as the corporate information of issuers⁹, or the registration details of investors¹⁰, which are static most of the time but are typically quite sizable. These types of data probably warrant a different approach. For example, the actual data can be hosted “off-chain” in a normal database, but a hash of the data can be stored “on-chain” and validated so that participants are confident that the data have not been tampered with. The address to the database can be encrypted and then stored “on-chain”. Through these methods, only permissioned users know which block to look in to locate the encrypted address, and have the necessary keys to decrypt the address and access the information. Obviously hosting data “off-chain” in a normal database means that the “off-chain” environment is exposed to the same security risks as any other centralised database, but this is a trade-off which each implementation should assess when designing their network.

We would also make mention here that at this point we have only explored storing data strings “on-chain”, and not hypermedia or rich content, such as images, audio and video. There are current projects looking at storing hypermedia effectively within distributed and decentralised systems, the most well-known being the Interplanetary File System (IPFS) protocol. These types of solutions would warrant exploration by developers looking to take advantage of hypermedia in their implementations.

6. Equity Crowdfunding (ECF) on DLT

⁸ And also expensive if it was a public blockchain implementation

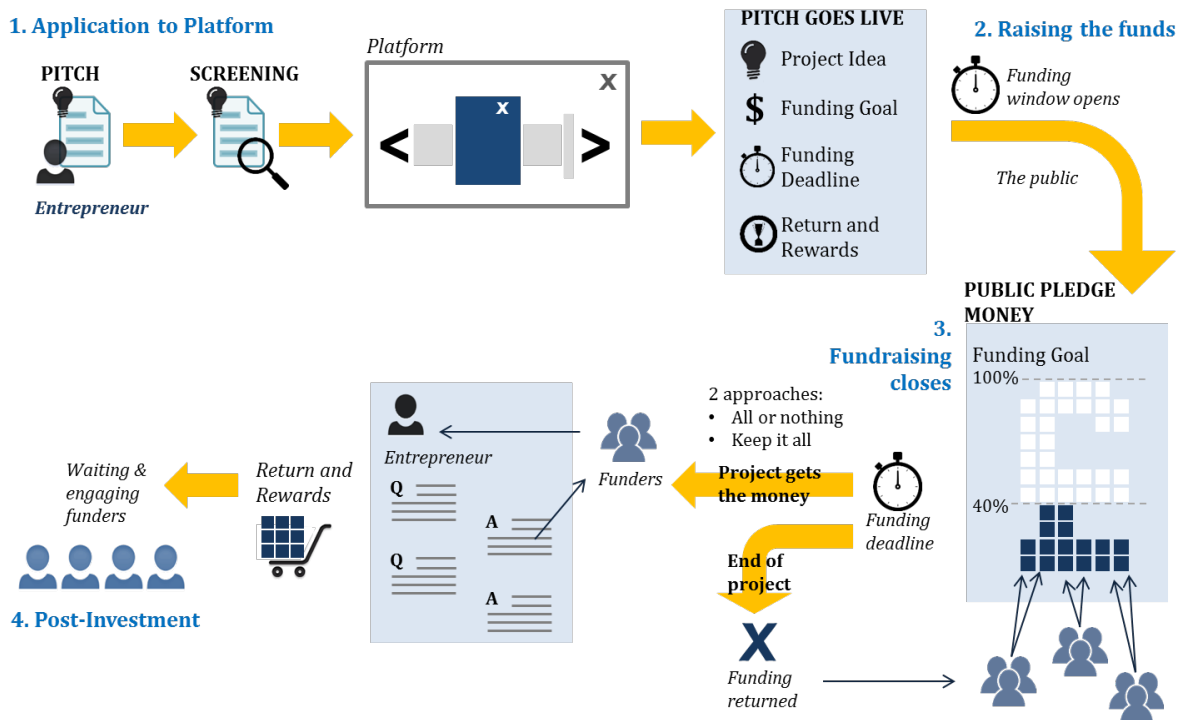
⁹ Company name, registration number, list of shareholders, etc

¹⁰ Name, IC numbers, emails, addresses, etc

A. Quick Introduction to Equity Crowdfunding (ECF)

As mentioned in the introduction, the team chose Equity Crowdfunding (ECF) as the use case for the pilot project, mainly due to the familiarity of the team with this asset class. For those unfamiliar, ECF is an offering of shares in a private company to the general public in order to raise funds. The initial fundraising exercise, also known as an ECF campaign, would have a minimum and maximum limit of funds which the issuer intends to raise in return for a certain share of equity. There is an “all or nothing” approach to ECF where a campaign is only deemed successful, and funds disbursed if the campaign successfully reaches its minimum fundraising target before the end of the campaign.

A summarised introduction of how a campaign works is described here:



Subsequently after the crowdfunding period, shares will be disbursed accordingly to the investors. After a specified period, investors will be able to transact these shares on a secondary market. When trading shares (as well as other types of securities), there are typically certain best practices to adhere to. One of the most common is Delivery vs Payment (DvP), where the securities and monies are exchanged between buyer and seller at exactly the same time to minimise settlement risk. Such transactions would typically require the use of a counterparty or escrow, and will also be explored as part of the pilot.

B. Creating Market Participants and their Specific Roles in the Market

Within the ECF Market, each of the market participants have specific roles:

- **Regulator:** Able to view all transactions and approve access to Trustees and Operators
- **Trustees:** Mint and destroy equity and Monies tokens (*discussed further below*)
- **Operators:** Create Issuers and Investors, as well as approving Campaigns
- **Issuers:** Can create campaigns. Have wallets for equity tokens and Monies
- **Investors:** Have wallets for equity tokens and Monies. Can pledge Monies to campaigns, as well as transact with each other between equity tokens and Monies

For each of these market participants, the project set up specific smart contracts with custom functions to represent their specific roles. The source codes can be referred to here¹¹.

Setting up these roles as smart contracts rather than merely distributed data has several key benefits:

- Other applications and services can connect and interact with the system
- Individual users and entities can more easily be revoked by those authorised to do so
- Assets can be more easily integrated with the various stakeholder roles and compliance

Please note that while the technical team has chosen to build to these roles using Smart Contracts, it is entirely possible for this hierarchy to be achieved using different methods. At this point in the project it seemed more efficient to perform all functionality through smart contracts, but for each developer and implementation the requirements may vary.

¹¹ Please refer to the Project Castor micro-site: <https://castor.my>

C. Representing Equity and Monies on the Blockchain

After confirming the roles, the next step was to represent assets on the distributed ledger. The ECF market essentially involves two different types of assets. The equity or shares of the issuers and monies. However both assets have different characteristics, which would require that we make use of different ways to represent them on the ledger.

The first of the two is monies. In traditional ECF campaigns, all monies are held by Trustees which safeguard the funds, and ensure that there is no co-mingling. This means that the Trustees are the ones with the final say when it comes to individual money balances of issuers and investors. The same structure exists on the ledger and as in real life, all money is fungible, which we have also sought to replicate on the ledger. What does this entail from a technical perspective? It means that:

1. There needs to be a way for Trustees to “mint” money tokens on the ledger to represent the actual real world trust account balances.
2. For it to be completely fungible, tokens need to be backed by real funds.
3. Each trustee maintains their own supply of fungible tokens.
4. Trustees may only issue tokens direct to investors upon their deposit of funds.
5. Money tokens are destroyed when an investor or issuer withdraws their actual money balances from the Trustee in real life.

To facilitate this, the team chose the most common fungible token standard smart contract, which is familiar to most. The Ethereum ERC20 token standard is used to represent monies on the distributed ledger. A good description of how the ERC20 standard looks like can be found on The Ethereum Wiki¹².

The second of the assets to be represented on the ledger is equity belonging to private companies, which is a different beast compared to monies. To start with, equities of different companies are not fungible between each other. In fact if a company has a tiered share structure, different shareholders might hold different shares types, which even in themselves are not fungible with each other. An issuer can also undertake multiple ECF campaigns, where the equity offered by each in various campaigns could be significantly different from each other. Hence we need:

1. The Trustee must be able to “mint” non-fungible assets at the onset of an ECF campaign to represent the actual units of equity of the company
2. The equity assets should be disbursed by the ECF campaign smart contract (*discussed in further detail in the next sub-chapter*). The campaign smart contract will specify, amongst other criteria, the “exchange rate” of equity vs monies to be disbursed to investors when the campaign is successful
3. If an ECF campaign is successful, the smart contract should facilitate the exchange of equity assets vs. monies tokens and automatically send the equity assets to investors upon completion of the campaign
4. If the ECF campaign is not successful, equity assets should not be issued to investors and monies tokens should be automatically returned to investors upon closing the campaign. If need be the equity assets are destroyed.

For non-fungible tokens, the technical team turned to yet another Ethereum standard, choosing to use the ERC721 smart contract standard, which can be found here¹³.

Finally, we need a way to permission smart contracts in the market network so that only the right parties can access and use the contracts. Permissioned smart contracts can be referred to here¹⁴.

¹² https://theethereum.wiki/w/index.php/ERC20_Token_Standard

¹³ <http://erc721.org/>

¹⁴ Please refer to the Project Castor micro-site: <https://castor.my>

D. Quick aside: Why are Smart Contract Standards Important

Astute readers would have noticed that the pilot project has used two open Ethereum Smart Contract Standards, namely ERC20 and ERC721 to represent tokens and assets instead of building our own proprietary standards. We wanted to elaborate a bit on why we chose to do so.

Interoperability has long been a key goal of DLT development. As different chains and distributed ledger technologies continue to be developed, each with different programming languages, the need to be able to integrate between each other, whether to share data or trigger specific services, become increasingly important to extend the usefulness of DLT. In this regard, development standards are important because they provide a possible solution to interoperability. We have seen the benefits of this in the API world as most of the industry has adopted REST as a development standard. It helps developers talk with applications and systems using the same language – Standards help us describe a certain construct or schema, which is easily understandable to other developers, regardless of whether they have used Ethereum or not.

Currently, DLT or blockchains as a whole lack any formal development standards put in place by any recognised standards-setting body. In fact, some DLTs have invented their own programming language¹⁵. While efforts to set global standards are underway¹⁶, one could argue that in some way, being open source and decentralised means DLT should be self-governed and the development community would rally around its own set of standards.

One of the most popular communities for now, especially within the Smart Contract space, are the standards being adopted within the Ethereum development community. One could argue that the only reason this is so is because of the boom with Initial Coin Offerings (ICOs) that have taken place on the Ethereum platform using the ERC20 standard, or the CryptoKitties craze which is built upon ERC721. Regardless of how they achieved adoption, they are now widely used by many different use cases, products and services, which in-turn helps with interoperability. Most major cryptocurrency wallets that accept Ether now support or are in the midst of implementing support for ERC20 tokens. The Hyperledger Project, one of the large enterprise blockchain initiatives, have also announced support for ERC20 and ERC721 as token models for their Fabric and Sawtooth protocols.

Since smart contract logic code cannot be changed once it has been deployed to a network, and standards are continually improving, and in-turn evolving, we must consider what this means. In order to future proof our systems, we have to consider the different ways we can account for change. It is likely that any improvements or enhancements made to the fungible and non-fungible token space will be built upon the foundations of ERC20 and ERC721. In fact this is already happening as we speak, for example on top of the existing ERC20 standard, there is already proposed enhancements ERC223 which prevents lost Ether, and ERC777 which uses contract registrars.

¹⁵ See Ethereum with Solidity

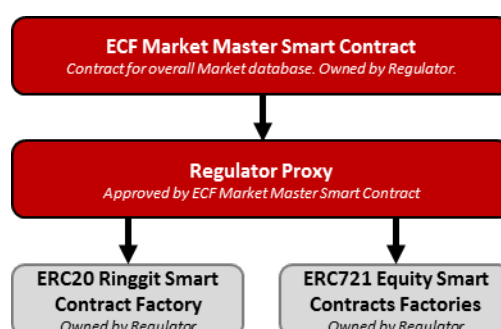
¹⁶ SC is part of the ISO Standards Committee for Blockchains and Distributed Ledger Technology

E. Upgradable Smart Contracts

One of the main benefits of smart contracts is that once they have been published to a public blockchain, they cannot be altered or removed. Although this is a powerful feature – once you discover an error in the contract or need to update your supply from one standard to another; making changes becomes a serious problem too. This is especially important in consideration of the fact that over 25% of all smart contracts¹⁷ currently contain unfixable critical errors.

Upgradable smart contracts represent an alternative approach to design where data and logic are split into different contracts rather than storing everything in the same place as traditional contracts do. In order to do this, additional layers of code abstraction are required, which is often achieved through what is known as a key-value store and proxy approach.

We use the following approach:



This set up allows the Regulator proxy to make changes to the ERC20 and ERC721 Smart Contracts without impacting the downstream. For example, there could be a new Trustee coming into the market, which will need to be permissioned to use the ERC20 and ERC721 factories. We may even want to upgrade the ERC20 monies tokens to a new standard in the future. Having certain contracts upgradable means that we can get around these problems without having to migrate account balances from one contract to another as the balances are stored in master data contracts.

While purists will cringe at the idea of breaking immutability, we felt that this method makes the market more sustainable in the long run. Do note that this does not change the immutability of the transactions themselves – which is what we believe should be truly immutable. Upgradable contracts do mean that an operator could in theory make changes to the terms of future campaigns, but to do so the relevant parties must approve it for interconnectivity, where specific rules are put in place to ensure that investors are notified of such changes if and when they occur.

Another benefit of upgradable contracts is the ability to introduce contract factories. Contract factories provide a way for a single master ERC20 or ERC721 contracts to support an unlimited number of different ERC20 or ERC721 supplies. In order for standard ERC20 and ERC721 applications and services to support these factories, new interface contracts need to be generated.

¹⁷ <https://news.bitcoin.com/25-of-all-smart-contracts-contain-critical-bugs/>

F. Self-Executing ECF Campaigns and Secondary Market Transactions

Once we have both the monies tokens and equity assets on the blockchain, we want to have self-executing ECF campaign smart contracts, or in general terms a “primary market offering” where we use smart contract to codify the rules of the offering and distribute the appropriate tokens and assets once the offering is closed. These smart contracts are custom built, and how they are coded will largely depend on the particular rules and criteria of the campaign, issuer or operator. For the version developed during the pilot project, please refer to our microsite here¹⁸.

Our primary objective is to facilitate secondary trading without requiring human intermediaries, where investors can exchange equity assets and Monies tokens with each other. While this can be done on a peer-to-peer basis already, both parties would need to trust each other to send through their respective tokens and assets. In the real world, a central counter party or an escrow agent would be required to facilitate delivery vs payment (DVP). In the ledger world such a function can be easily replaced by an escrow smart contract, an example we created here¹⁹. Again, the smart contract can be coded to be as simple or complex as required, depending on the conditions of the transaction.

¹⁸ Please refer to the Project Castor micro-site: <https://castor.my>

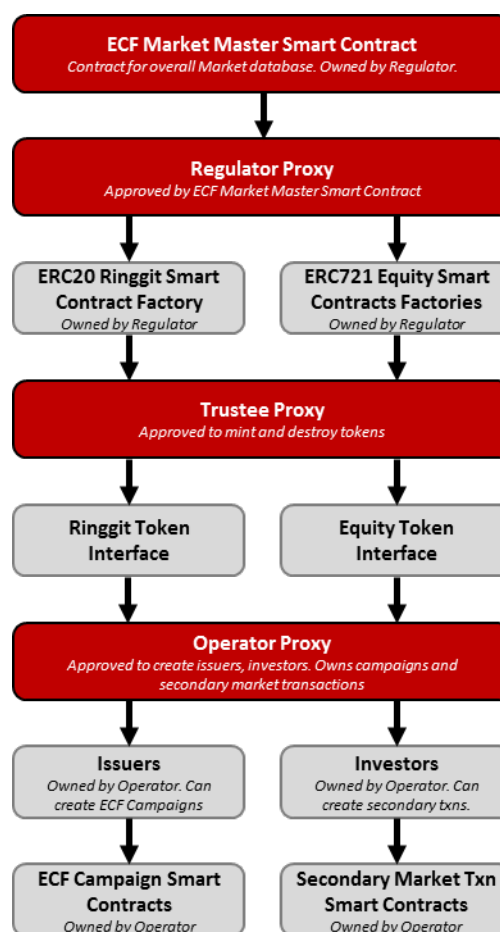
¹⁹ Please refer to the Project Castor micro-site: <https://castor.my>

G. Combining All the Parts to Form a Market

We now have all of the ingredients to build a functioning market. Just to recap on the pertinent points:

1. We used a hierarchical structure to gatekeep access to the market, whilst maintaining transactional privacy between operators and trustees whilst defining each individual role.
2. We used standardised ERC20 and ERC721 standards to represent equity and Monies
3. We used custom Smart Contracts to have self-executing ECF campaigns, as well as automated escrow contracts to fasciltate peer-to-peer secondary market transactions

The following diagram represents the overall structure of the entire market:



As discussed two sub-chapters up, the proxies are used as anchors that provide a permanent address that the key-value store can then use to index data shared across multiple applications or contract components. This allows the final contract layers (in grey) to be updated without altering the underlying data

7. Thinking Aloud about Security: Private vs Public Networks

The traditional approach to securing systems can be characterised by how someone would defend a castle – you dug a moat and built strong walls, with fixed entry and exit points which were heavily guarded and patrolled. This approach followed the logic that you have a number of systems to secure, and it all remained within your environment. Securing networks followed the same method – you just add highly secure connections or tunnels between multiple castles. While these methods worked, there was no doubt that it was also expensive to build up and maintain.

Distributed systems however, take a less conventional approach to security. As the same data is replicated across multiple nodes, the security of individual nodes becomes less important. Instead, the security of distributed systems rely on having as many independently running nodes as possible, making it almost impossible for a hacker to take control of more than 50% of the nodes in order to take control of the network.

Hence when building a distributed ledger based system, the choice on whether to build a private or public network presents some interesting security and economic trade-offs. To secure your system you would want as many independently run and secured nodes as possible in order to prevent 51% attacks. However a private network would rely on the traditional security approach described in the first paragraph, which means that it is unlikely that such an implementation would have too many independently secure nodes – it would be much too expensive to maintain.

Then why not public networks? One would argue that any major public blockchain today would be sufficiently decentralised and distributed²⁰ to not be susceptible to 51% attacks and hence tamper proof. But there are other factors to consider. Firstly building your system on a public blockchain means that you will be paying “gas” for everything – from recording data, publishing smart contracts to effecting transactions, which also puts the cost of upkeeping your system at the mercy of fluctuating “gas” prices. Also, using a public blockchain means having to deal with general lower throughputs²¹, and possible occurrences of network congestion.

Ultimately the decision on whether to build a private network, or run the system on a public blockchain will closely hinge on how the developer envisions their system to function. If high performance and throughput beyond the scope of existing public blockchains are absolute priorities, then a private network could be a better choice. For all other cases, a public blockchain should be strongly considered, possibly as the primary default option. However the economics of having to stockpile gas will need to be taken into account as to whether it would be viable for the long term.

²⁰ Developers should take care to assess each public blockchain for different degrees of decentralization. Not all blockchains are equally decentralized.

²¹ Private networks tend to be configured for much higher levels of performance

8. The Risk of Disintermediation: Do We Still Need Operators and Trustees

Astute readers would have noticed that in the second half of the pilot project, the team had managed to replicate the main functions of our traditional intermediaries, namely the operators and trustees, on the distributed ledger as self-executing Smart Contracts. While within the project we have enforced a hierarchy in order to retain the roles of the operators and trustees, in theory these Smart Contracts could be deployed by any other party. The governance of these functions can be left to the ledger itself.

Which begs the obvious question – if a few simple lines of code deployed on to a distributed ledger could replace such functions, is there still a need for these traditional intermediaries to exist? In the distributed system which we have built, they exist solely as a gatekeeper to issuers and investors, as well as a delegated authority to effect transactions.

It is important to note however that our pilot system is incomplete – i.e. there are still functions that we currently exists but are not part of the scope of the pilot project. For example the know-your customer (KYC) process of conducting KYC, AML and CTF checks on issuers and investors, the issuer curation process where operators assess the value proposition of issuers, and the payments process where Monies is transacted between bank accounts²². To a certain extent these processes are much more complex and nuanced, requiring human judgement and are not easily automated. That could well be the saving grace for traditional intermediaries.

However that is not to say that blockchain and DLT enthusiasts are not attempting to automate away these roles. Self-sovereign digital identities is one of the biggest field of study within the DLT space where multiple projects are underway, as is the field of payments, of which the earliest blockchains were designed to replace. In a perfectly decentralised world, such applications could be integrated with each other to completely eliminate the role of intermediaries.

Some would say that such a scenario would remain the pipe dream of purists, but there is no doubt that the role of traditional intermediaries in maintaining trust and governance are being diminished in distributed systems. To a certain extent this shift is already occurring without DLT – most new markets are moving towards a platform model with non-intermediated access. If intermediaries are to continue to exist, it needs to rediscover its role or risk being replaced by several lines of code.

²² While we have represented the monies on the ledger, in the pilot project it is merely meant to be a reflection of the actual transactions and balances in real world bank accounts.

9. Putting it All Together

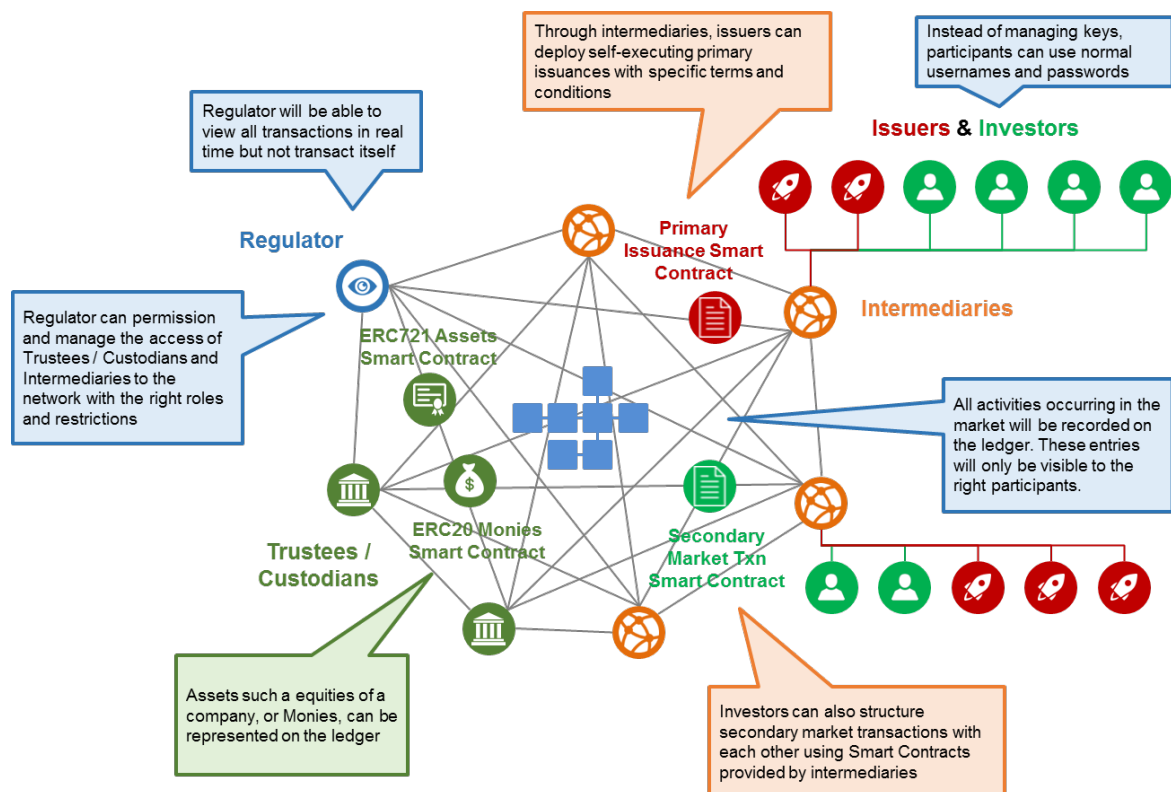
We have demonstrated how a DLT-based infrastructure can be implemented for the ECF market. Equally, the same blueprint can be applied to other unlisted or OTC markets as well.

When designing the infrastructure of a market, the specific participants of the market, and their respective roles and functions will first need to be clearly identified. It is also crucial to understand the relationships between the participants, as well as their interactions with each other, both in the real world, and how such “interactions” can be modelled on the ledger. This would include permissioning new participants on to the system, minting and sending new tokens or deploying certain Smart Contract between multiple participants.

In a typical regulated market, at a minimum we would have the following participants and their roles respective roles and functions:

Participant	Roles and Functions
Regulator	<ul style="list-style-type: none"> Supervise and survey the market through having full read-only view of all transactions and activities Permission and manage the access of the intermediaries as well as trustee / custodians to the market Ensure clear segregation of duties between trustee and custodians by controlling access to the appropriate smart contract factories
Trustees / Custodians	<ul style="list-style-type: none"> In the real world, trustees / custodians ensure that the assets of investors and issuers are clearly segregated, protected and no-comingling In the DLT world, they would be responsible to reflect or represent the actual asset holdings of investors and issuers on the ledger. This would mean ensuring that the market has the right amount of supply of assets, and the assets are attributed to the right parties In codifying asset tokens, they should work together with intermediaries to capture specific characteristics of the assets within the tokens itself
Intermediaries	<ul style="list-style-type: none"> Intermediaries gatekeep access to the market for investors and issuers In some use cases, the intermediaries themselves would also function as issuers They also facilitate the primary issuance process of an asset as well as any subsequent secondary trading This would mean deploying specific Smart Contracts which codifies the terms and conditions of a primary issuance (as agreed with the issuers) as well as making available secondary trading Smart Contracts where necessary
Investors and Issuers	<ul style="list-style-type: none"> The end-users of the market. They own addresses which reflect their asset balances. Participate in primary issuance and secondary trading activities by sending tokens to the specific Smart Contracts. As in the real world, they do not participate directly in the market itself, but rather through the intermediary, and similarly in our current design that means they do not access the ledger directly. However this can change depending on the design of the market.

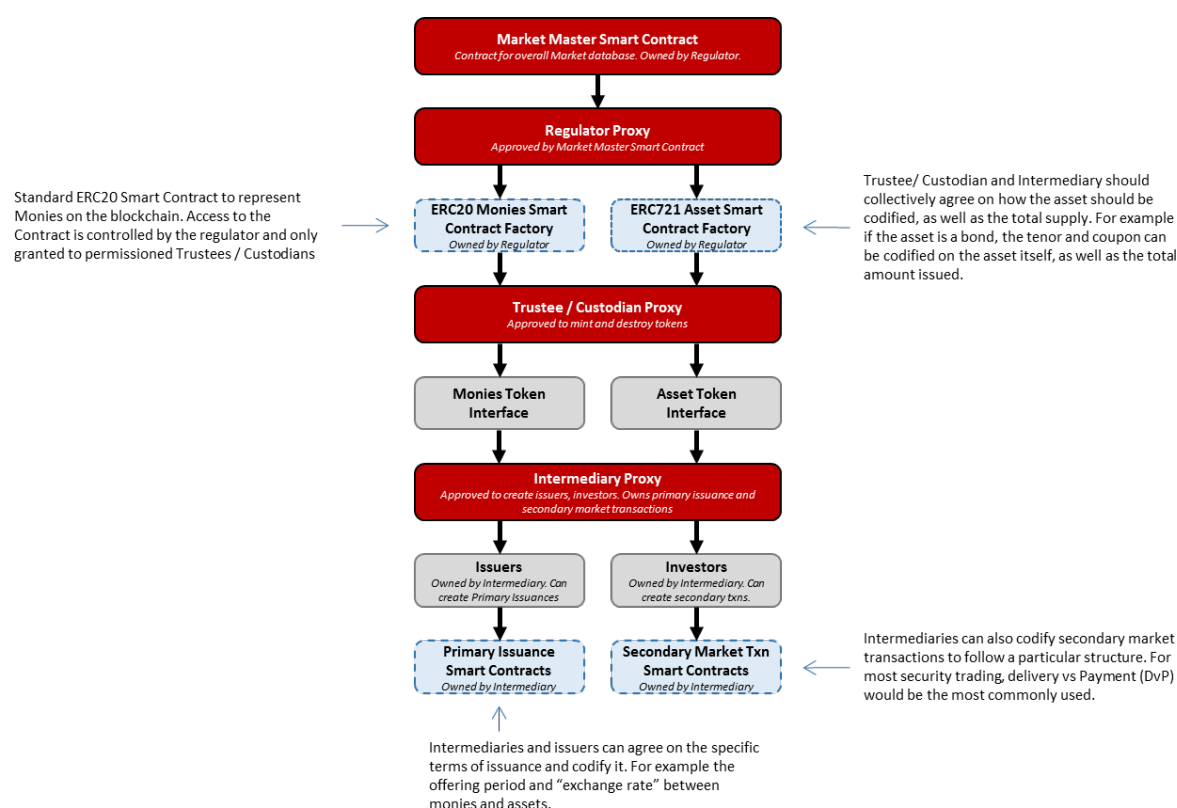
Conceptually, you would have the following structure for the market.



From this “flat” view it would seem as if all participants are equal participants on the market, but as discussed earlier each participant has a specific role and function to perform, and these are coded into the Smart Contracts themselves.

To enable all of these, there is also typically a specific governance hierarchy within the market where participants higher on the hierarchy can permission other participants to perform certain actions. For example the regulator admits trustees / custodians into the network and allows them to access the monies and assets Smart Contract to mint or destroy tokens. As much as possible these mimic the same hierarchy in real life.

Hence from a hierarchical view we arrive at the structure below:



To facilitate a certain degree of interoperability and future-proofing, we strongly recommend that the Monies and Assets Smart Contracts should be coded by leveraging on existing blockchain development standards. As part of the pilot project we have suggested using the ERC20 and ERC721 token standards, though there are others which might be equally viable.

The only custom Smart Contracts to be developed would be those related to market transactions, which are the Primary Issuance and Secondary Market Transaction Smart Contracts. Note that these are specific to each market and can be tailored to suit each specific offering within the market as well. For example, an OTC Swap offering can vary very differently between one to the other, and a developer might need to custom develop specific Smart Contracts to reflect specific conditions within the Swap agreement.

Finally, this blueprint which we have defined can cater for both whether the market is deployed on a private or public network. This decision is dependent on the respective participants' needs/requirements, risk appetite and technology capability when embarking on building such a distributed market.

Similarly for the use of Smart Contracts – while we have had almost everything done “on-chain”, including using Smart Contracts for transactions, the same designs can also be applied by developers who want to implement layer two off-chain or side-chain solutions as well. However doing everything “on-chain” has strong benefits of transparency, which should be carefully weighed and considered.

10. Last Words

To conclude, we wish to reiterate that for the capital markets, “decentralise everything” is not the final solution. Rather, we envision a multi-tier market environment in the future where centralised and decentralised markets co-exist. For the unlisted and OTC markets, we have demonstrated that DLT could be a suitable technology to support the market.

This blueprint lays down what the SC considers the foundational elements when building a decentralised market structure. While we have provided general guidance and expectations on how certain elements are to be approached and thought about, it should serve as a jumping-off point for further, more in-depth discussions on specific implementation details. Ultimately we believe the final solution should aim to be pragmatic, and to re-emphasise the point – dependent on the respective participants’ requirements, risk appetite and technology capability when embarking on building such a distributed market.

For those who wish to learn more about Project Castor, we encourage interested readers to further explore the additional information available at our micro-site (www.castor.my), as well as the numerous online repositories, forums, Telegram groups on blockchain and DLT development. Further conversations and engagements with the team can be requested via our aFINity channel (<https://www.sc.com.my/digital/afinitysc-industry-call-for-participation/>) and afinity@seccom.com.my.

The team would like to thank all who have contributed to the delivery of the blueprint, micro-site and pilot project. We look forward to the discussions this paper will provoke in the future.

Background information:

The Securities Commission Malaysia (SC), a statutory body reporting to the Minister of Finance, was established under the Securities Commission Act 1993. It is the sole regulatory agency for the regulation and development of capital markets. The SC has direct responsibility for supervising and monitoring the activities of market institutions, including the exchanges and clearing houses, and regulating all persons licensed under the Capital Markets and Services Act 2007. More information about the SC is available on its website at www.sc.com.my. Follow the SC on twitter at @SecComMy for more updates.

Authors:

Chin Wei Min

Executive Director

Innovation, Digital & Strategy

Securities Commission Malaysia

Chan Zhong Yang

Assistant General Manager

Innovation, Digital & Strategy

Securities Commission Malaysia