StartEngine LDGR™

WHITEPAPER

With the launch of Regulation Crowdfunding and amendments to Regulation A, investors who have purchased securities have the right to sell them to other investors under the conditions specified in the purchase agreement, without needing to register the transaction with the Securities Exchange Commission (SEC). Liquidity is a critical issue for ordinary investors who purchase securities in privately held companies. They are not necessarily able to wait years before having the opportunity to sell their securities, and in some cases their financial situation requires them to sell them immediately. Solving this issue is critical to allowing these new regulations to continue to provide corporations the capital they need to grow their businesses.

Peer to Peer trading with the Blockchain (patent pending)

The ability to store documents digitally has become the norm in our new digital economy. However, the securities industry has been slow to adopt new digital technologies that enhance its service and reduce costs to its customers. The ability to store securities digitally with a central database has raised a new set of concerns about security of the securities and--more importantly--personal data. There have been many well-documented hacks of personal information from both government and banking databases. The security risks in the financial industry are higher and the consequences very serious. Consumers are worried about their data being stolen and then sold to other fraudsters, while financial institutions are worried about their reputation and high costs to restitute investor losses.

The advent of the new cryptocurrency and digital ledger technology called the Blockchain offers a very secure and cost effective solution for providing security to the consumer and lower costs for the industry. The securities can be held securely using end-to-end encryption, yet openly authenticated, referenced and documented so that ownership of the securities can be trusted as reliable. In the past, this type of technology would have been extremely expensive to build and manage, but today it it can be implemented and deployed relatively easily at little cost. The winners are investors, who can 1) rely on the security's ownership authentication; 2) reduce their costs for maintaining their accounts, and 3) sell securities when they would like to.

StartEngine is building a system to facilitate the recording of ownership and transfer of securities sold under the new Securities Act Regulation Crowdfunding, Regulation D and Regulation A, and the trading of those securities to other investors without the fear of the loss of data or the high costs that are ordinarily imposed on trading of those securities. StartEngine intends to create a new cryptocurrency for each company that raises capital using these above regulations. Each individual security sold is represented as one coin in the Blockchain. For example if an investor purchases 100 shares in a company then those shares are represented as 100 tokens. If an investor lends a company $1,000 in a loan then the loan is represented as 1,000 tokens. StartEngine hopes this system can reduce the overall cost of trading securities, and by enhancing security and greatly improve the experience for investors. The buyer can use

fiat, bitcoin or ether to purchase the securities and the seller can either keep the currency tendered or convert them into other currency.

Background

Financial Regulation

Since the creation of the Securities Exchange Commission and the enactment of the Securities Act of 1933, companies generally needed to register their offerings and securities with the SEC if they wanted to sell securities to the general public. Without registration, companies could only sell their securities if the transaction complied with an exemption from registration, such as that available under Regulation D, which provides a relatively easy to comply with exemption from registration if securities are only sold to accredited investors: those with incomes of over $200,000 or net worth (excluding homes) of over $1 million.

This changed in April 2012, when president Barack Obama signed the JOBS ACT, a bipartisan new financial law designed to help small businesses raise the capital and hire the people they need to grow and build a successful business. The JOBS ACT changed the system by requiring the SEC to implement or amend its regulations to permit greater ability for companies to sell their securities directly to the general public. Three regulations have emerged from the JOBS ACT that are being used by companies to raise capital online:

1. New Regulation Crowdfunding, which permits companies to raise capital from the general public up to $1,070,000 per year upon filing a Form C with the EDGAR SEC public database. This regulation is gaining strong popularity since its launch in May 2016 with more than XX companies having currently filed and raised close to $YY.

2. Amendments to Regulation A, popularly referred to as “Regulation A+,” which permits companies to also raise capital directly from the general public up to $50M per year and requires the company to file the Form 1A for review and commenting by the SEC. In order to proceed, the offering must be qualified by the SEC. This process is more expensive and it can take several months to get to qualification. Since 2015, hundreds of companies have filed a form 1A with the SEC.

3. New Regulation D 506(c), which permits companies to raise capital directly from accredited investors with no limit. The company files a form D with the SEC and States after the close of the raise. This process is not expensive but it is limited to accredited investors and the securities are “restricted securities,” meaning that for at least a year they cannot be easily traded.

The capital market for small businesses

Small businesses with under 50 employees are the backbone of the US economy, generating more jobs than medium and large businesses. There are more than five million businesses operating in the United States, and many of them seek capital for equipment, purchasing inventory, improving a property, leases and hiring employees. However, because they are small, getting access to capital is very hard. Banks typically do not lend to small businesses because they do not have tangible assets or sufficient track record of profits, Venture Capitalists provide around $70B in capital every year, but they only invest in a few thousands of companies a year, and are biased towards highly educated and experienced management teams in narrow fields of the industry. Finally, angel investors provide somewhere around $40B year in capital to tens of thousands of companies who are able to find these wealthy accredited investors. Unfortunately, only a small portion of the population is accredited and many of those do not invest in high risk companies as they are generally reaching the age of retirement and seeking more income-generating investments such as real estate or municipal bonds.

With the JOBS ACT, the general consumer has access to startups and small businesses who can take advantage of these new regulations. However, one issue those regulations do not address is liquidity. Even successful startups do not typically provide any liquidity until 5 to 7 years after it is founded. Other small businesses have longer business cycles where investors have to be very patient. While accredited investors are used to having their capital locked for a long period of time with the hopes of a strong return, ordinary investors are probably not able to wait this long.

If a company decides to go public they typically hire an investment bank and file a registration statement on Form S-1 with SEC. Once that registration statement is effective, they sell securities to a set of investors during the Initial Public Offering and then list their securities on a national market such as the NASDAQ or the New York Stock Exchange. After listing, any investor can purchase securities from other investors through any registered broker-dealer. The share price fluctuates based on the supply and demand for these securities, and there is an entire industry catered to analyzing and reporting on these listed companies. The cost to trade these securities used to be very expensive until the discount brokers appeared on the market, followed by online trading websites. Today, it costs just a few dollars to execute a trade.

There are half the number of listed companies today compared to 20 years ago. To go public in the traditional way, a company needs to be very large and raise hundreds of millions of dollars to interest any of the large investment banks. A small business has no chance to raise capital this way. With the changes implemented by the JOBS ACT companies can raise money from the general public, but there is no way to provide liquidity to the investors by listing shares on a national exchange. The smaller trading forums such as the OTCQX market have large requirements such as a minimum of $2M in assets, and a company must have a financial sponsor or investment bank willing to be a market maker for the company's securities - another barrier for the small businesses.

Investors who purchase securities in a company who is using Regulation Crowdfunding need to wait one year before they can easily sell their securities to other investors. With Regulation A+ there is no waiting time. However, only a few companies have listed their securities with any kind of exchange or trading forum. The investor has no ability to sell their securities unless they find themselves an investor who wants to purchase them. This lack of liquidity is a hurdle for these ordinary investors and until an effective solution exists, it will slow down these investors and small businesses will have harder time raising capital which defeats the purpose of having these new regulations.

Blockchain

Blockchain technology is the backbone of cryptocurrencies such as Bitcoin and Ether. However, as a 'distributed ledger,' blockchain also has a vast number of additional applications. Smart contracts, or code executed on a blockchain, bring significant advantages over existing database-centric applications. Open blockchains such as the one used by BitCoin offer these benefits:

Low-cost
There is no middleman to impose fees. Transfers require only small transaction fees paid to the blockchain miners.

Immutability
The ledger is policed by every member in the network and its integrity is checked and agreed by the network as a whole on an ongoing basis. Any changes that a minority party attempts to make to the blockchain are recognized and rejected by the majority.

Transparency
Everything that takes place on the ledger is visible to everyone. It is possible to see everything that has been recorded since the ledger's beginning on the blockchain.

Irreversibility
Because the ledger is immutable, a transfer that has been accepted into the blockchain cannot be reversed.

Security
Because the blockchain is maintained by a large network of participants, no hacker can get enough influence to submit a fraudulent transaction. The more valuable the tokenized security or cryptocurrency, the larger the network and therefore the more resources it would take for a hacker to be able to enter into fraudulent behaviors.

The first application of the blockchain was Bitcoin, first introduced in January 2009 as an innovative and secure currency. Without the blockchain, owners of Bitcoin could not be guaranteed that someone else could not copy their bitcoin and 'double spend.' Normally that occurs via an intermediary such as a bank, and that is where the costs and delays occur. The intermediary may prove to be untrustworthy and can reverse a transaction without the user's permission. What's more, there is a single point of failure and the bank can be hacked or the transaction intercepted by hackers. Bitcoin needed a solution to reducing the costs of transferring Bitcoin to other parties, that eliminated security risk and virtually all cost. The Bitcoin system that emerged is a peer-to-peer shared ledger and therefore is secure without the need for an intermediary.

The initial innovation of the blockchain is that it is very difficult to add a Bitcoin transaction to the blockchain but extremely easy for anyone to check if a transaction is valid. Fraudulent transactions are quickly identified and prevented from entering the blockchain.

The second innovation of the blockchain is smart contracts. The transfer of funds does not involve moving Bitcoin from one database to another; rather it updates the ledger to reflect how much Bitcoin is owned by each address. Using the same technology of transparency, security and immutably by consensus of the entire network, this can be extended to the trading of securities issued under Regulation Crowdfunding and Regulation A. Smart contracts are code executed on the blockchain that enable the additional complexity required for managing tokenized securities (versus managing a currency).. With smart contracts, the securities industry can create decentralized apps--known as “dapps”--for a number of applications which require the security and low cost that the blockchain can offer.

Product Description

An Alternative Trading System (ATS) is a broker dealer who uses the blockchain to efficiently and securely manage the trading of securities for both Regulation Crowdfunding, Regulation D and Regulation A. A Registered Transfer Agent (RTA) manages the ownership and records the transactions of the investors who purchased securities under Regulation Crowdfunding, Regulation D and Regulation A. The RTA is a trusted source for the ATS. The ATS is to use the blockchain to efficiently and securely manage the trading of securities for both Regulation Crowdfunding, Regulation D and Regulation A. The trading of securities is subject to SEC and State blue sky laws.

An investor who has purchased securities under these three rules can choose to sell these securities on an ATS by posting their proposed offer using a set of possible pricing models. Buyers can set a buying limit order setting a ceiling to the price of a purchase of securities. Sellers can set a selling limit order setting a minimum price to sell the securities.

Once posted, the ATS will verify with the RTA if the sellers owns free and clear the securities and they can be sold according to any applicable transfer restrictions. The ATS places a marker to block any further transactions for these securities. This function requires the RTA to permit a portion or all of the securities to be restricted and not sold by other trading platforms or broker dealers for a determined amount of time. If securities are locked by the ATS, they can only be unlocked at anytime by the same ATS; or if 7 days passes they are automatically released. If an investor tries to sell any securities which are locked then the RTA will not accept the transaction until they are released by the ATS who requested a lock of those securities.

The buyer can now purchase the securities using fiat, bitcoin or ether. This is done peer-to-peer without intermediaries using a smart contract. Once completed the smart contract updates the RTA with the identity of the new owner and the number of securities purchased. The RTA is the trusted source for executing the smart contract.

This peer to peer model is innovative and secure because it uses the blockchain and the bitcoin cryptocurrency for payments.

An investor can sell securities and buy securities through the ATS while keeping its money in bitcoin and incurring low transaction fees. The RTA charges the seller a small fee for each transaction. With this innovative system, an investor can make purchases and sales nearly instantly with low fees. Others, if they want, can convert their money back into dollars or fiat.

The identity of the owners of securities is known to the ATS but it is not visible on the blockchain which encrypts this information.

StartEngine LDGR™

StartEngine LDGR™ is a method that allows companies (“Issuer” or “Issuers”) to list in the blockchain each Regulation Crowdfunding, Regulation A+, Regulation D securities or Regulation S sold in an offering. StartEngine LDGR is owned by StartEngine Crowdfunding Inc, a Delaware Corporation (“StartEngine”). StartEngine LDGR requires the Issuer to select and hire the following vendors:

- Registered Transfer Agent (“RTA”) which records the identity of each investor and amount of securities sold in the offering or transferred. The RTA has a contract with one or more ATS to approve the transfers.
- Alternative Trading System (“ATS”) which permits the secondary trading of securities subject to SEC and State blue sky rules.

Each class of security offering has its own symbol represented by a unique four letter/number combination followed by a period and followed by a disbursement number in sequential order (eg: 1, 2, 3, etc.). Each Issuer has one or more unique StartEngine Security Name which cannot be modified because it is stored in the blockchain. The disbursement number is unique and cannot be out of sequence.

The RTA will create for each investor a StartEngine LDGR blockchain address representing their securities.

A RTA designated by the Issuer should offer the following service:

- Be a US based corporation
- Be a SEC Registered Transfer Agent
- Offer a secure application programming interface (“API”)
- Have the ability to lock the securities records during a transaction
- Use the email address as the unique identifier for each investor
- Allow recovery of ownership in case the email address is not longer accessible using a sworn affidavit signed by a Notary
- Have agreements with broker-dealers and ATS

An ATS should offer the following service:

- Be a registered Broker-Dealer with the SEC and a member of FINRA
- Apply to be an Alternative Trading System
- Verify the identity of the investor against a set of federal databases (AML/KYC)
- Verify with the transfer agent that the securities are free and clear to sell and bear no limitations on transfer
- Verify any restrictions in the securities agreement signed with the issuer
- Offer a secure application programming interface (“API”) with the RTA
- Follow the SEC and State blue sky laws for transfer eligibility

The currency used by issuers to use the StartEngine LDGR application is the StartEngine Coin. The Issuer pays a one time fee in StartEngine Coins and provides an Ethereum address and the selected RTA to create the *StartEngine Issuer Name*. StartEngine pays the miners who host the blockchain ledgers in gas to execute the *StartEngine Issuer Name* smart contract. This Ethereum address can be used by the issuer to change the RTA.

StartEngine, the RTA and the Issuer pay gas to the third party mining companies to hold securities in the blockchain by creating a block in the blockchain for each step in the process.

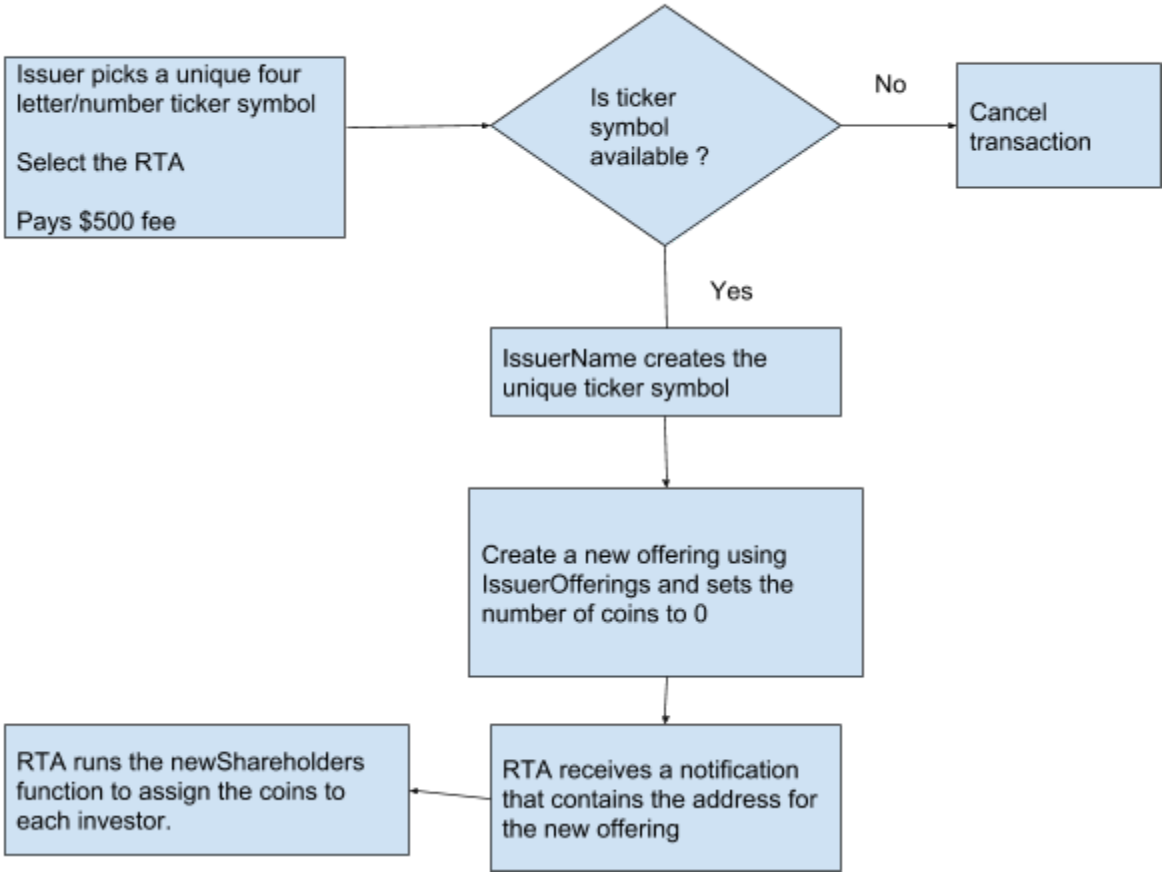
Each block contains the number of securities owned by the investor.

The RTA provides the gas which is used to execute the smart contract to create each offering.

Every time the issuer makes a disbursement for an offering, the RTA who is now the Trusted Source for the StartEngine Issuer Name, call the smart contract to create a new offering.

Example: Best Company Ever Inc. Issuer picks Best Transfer Agent Inc. to be the RTA and conducts an offering of 100,000 shares in a Regulation Crowdfunding offering to 500 investors. The Issuer requests StartEngine LDGR to create a new code BCEV and pays in StartEngine Coins. The RTA uses StartEngine LDGR to create individual investor blockchain addresses each containing the number of issued securities totalling 100,000 shares. The offering is named BCEV.1 until the trading restrictions are lifted and investors can use an ATS to sell BCEV shares. The ATS will execute the trade by calling a trading function. This function will call the RTA to check if the shares are available and, if true/available, the function will notify the Transfer Agent to register the new owner for the number of securities purchased.

Creating StartEngine Issuer Name for Issuer

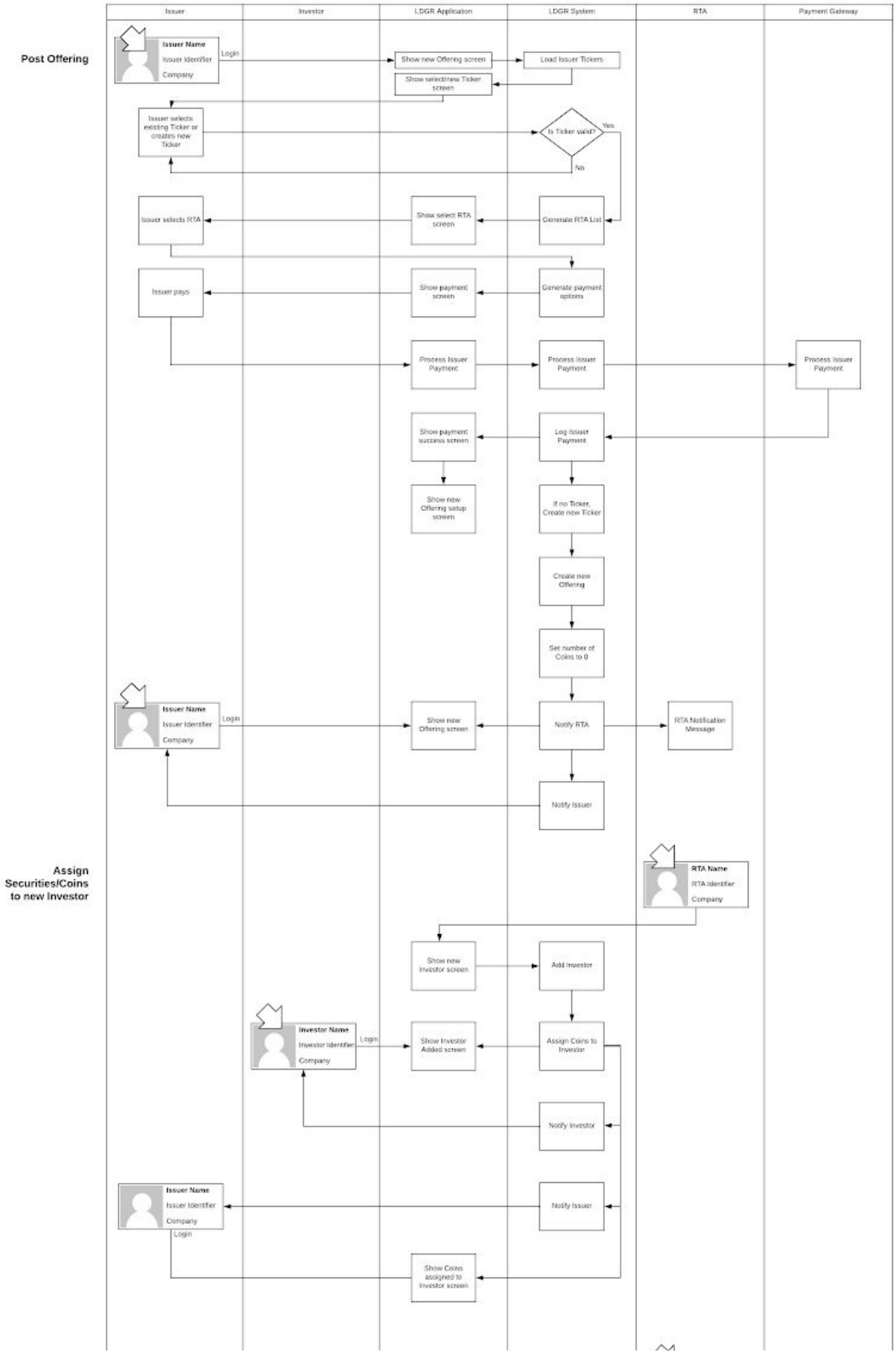


LDGR Technical Diagrams

Create Offering

StartEngine

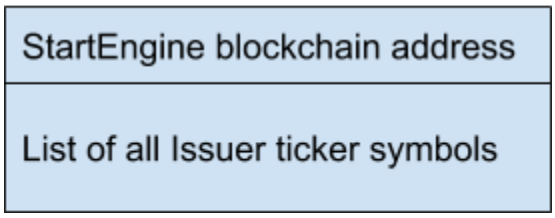
Technical Specification



IssuerName

IssuerName is a Dapp owned by StartEngine. This creates new ticker symbol names. It contains the list of ticker symbols which are unique four letter Issuer names. This information goes into the blockchain and is visible to the public. The Issuer pays in StartEngine Coins for registering a new issuer name (this amount can change).

IssuerName data table




```
/*
  base contract that sets deployer of contract as owner, initializes a transfer agent and provides some modifiers and ability to change transfer agent
*/
contract ownedByStartEngine {
  address owner;
  address issuer;
  address transfer_agent; // transfer agent acting as oracle
  mapping (string => address) offerings;

  function ownedByStartEngine() {
    owner = msg.sender;
  }

  event NewOffering(string company_name, string symbol, address newOffering);

  /* Only performable by StartEngine */
  modifier onlyOwner {
    require(msg.sender == owner);
    _;
  }

  /* Only performable by Transfer Agent */
```

```
  modifier onlyTransferAgent {
    require(msg.sender == transfer_agent);
    _;
  }

  function createOffering(string _company_name, string _symbol, address _issuer, address
_transfer_agent) onlyOwner {
    require(!offerings[_symbol]);
    address newOffering = new Offering(_company_name, _symbol, _issuer, _transfer_agent);
    NewOffering(_company_name, _symbol, newOffering);
    offerings[_symbol] = newOffering;
  }

  /* Allows StartEngine to set a new transfer agent */
  function updateTransferAgent(address newTransferAgent) onlyOwner {
    transfer_agent = newTransferAgent;
  }
}
```

IssuerOfferings

IssuerOfferings is a Dapp to create new offerings. The Issuer pays for each offering in StartEngine Coins (this amount can change). This information goes into the blockchain. Visible to the public. It lists the Issuer's blockchain address, unique ticker symbol, the corporate legal company, State file number, State of incorporation, operations business address (address, state and zip code), Transfer Agent blockchain address and list of offerings which are sequential numbers. This information goes into the blockchain and is visible to the public.

IssuerOfferings data table

Issuer blockchain address
Unique ticker symbol
Corporate legal name
State file number
State of incorporation
Physical Address of operation
Transfer Agent blockchain address
List of offerings

createName() is a function to created the unique cryptocurrency with the opening balance number of securities issued in the offering set to zero. Sets the trusted sources for the Transfer Agent using a unique blockchain address.

```
/*
  Offering that represents an Issuer, has a mapping of addresses (investors) to an int balance (shares held by address), is a factory to create Trades
*/
contract Offering is ownedByStartEngine {
  string public company_name; // name of company that issued shares
  string public symbol; // nice asset symbol for display purposes
  address[] disbursements;

  /* Constructor, owner = StartEngine, initial supply is 0 until company disburses */
  function Offering(
    string _company_name,
    string _symbol,
    address _issuer,
    address _transfer_agent,
  ) {
    company_name = _company_name;
    symbol = _symbol;
```

```
    issuer = _issuer;
    transfer_agent = _transfer_agent;
  }

  function createDisbursement() onlyOwner {
    string token_symbol = _symbol + (disbursements.length + 1)
    address newDisbursement = new Disbursement(token_symbol, issuer, transfer_agent);
    disbursements.push(newDisbursement);
  }
}
```

Offering

Offering is a Dapp that contains the Issuer blockchain address, the unique ticker symbol, the unique sequential offering number, total number of securities, list of investor blockchain addresses each with the number of securities. It also contains the total of list of initiated trades, whether completed or not. This information goes into the blockchain and is visible to the public.

Offering data table

Issuer blockchain address
Ticker symbol
Unique sequential offering number
Total number of securities
Investor blockchain address with number of securities
Transfer Agent
List of trades

Transfer Agent gets notified to see the new offering. Calls the newShareholder() function to create blockchain addresses with the values for each shareholder.

Other functions:

updateTransferAgent(): This functions allows to switch the Transfer Agent.

```
contract Disbursement is ownedByStartEngine {
  mapping (address => uint256) balanceOf;
  string public token_symbol;
  uint256 totalSupply;
  address[] trades;
```

```
  function Disbursement(_token_symbol, _issuer, _transfer_agent) {
    token_symbol = _token_symbol;
    issuer = _issuer;
    transfer_agent = _transfer_agent;
    totalSupply = 0;
  }

  event TradeRequested(address indexed newTrade); // Transfer agent can listen to
TradeRequested events, and approve / cancel
  event TradeSettled(address indexed trade); // Event signaling Trade has been settled, buyer
receives shares
  event TradeCancelled(address indexed trade); // Event signaling Trade has been cancelled,
seller is refunded shares


  /* Adds new shares that have just been issued to investor */
  function newShareHolder(address investor, uint256 sharesSold) onlyTransferAgent {
    balanceOf[investor] += sharesSold;
    totalSupply += sharesSold;
  }

  /* Create a new Trade contract, store it in a list, and also send out an event with the address of
the new contract */
  function requestTrade(
    address buyer,
    uint256 numShares
  ) public returns(address newTrade) {
    address seller = msg.sender;
    require(balanceOf[seller] >= numShares); // require the seller to have enough shares
    /* remove shares from seller, "hold" them in a pending Trade and send out event */
    balanceOf[seller] -= numShares;
    newTrade = new Trade(seller, buyer, numShares, transfer_agent);
    trades.push(newTrade);
    TradeRequested(newTrade);
    return newTrade;
  }

  /* Send shares balance to buyer address, mark Trade as settled */
  function approveTrade(address trade) onlyTransferAgent {
    require(trade.pending);
    balanceOf[trade.buyer] += trade.numShares;
    trade.settle();
```

```
    TradeSettled(trade);
  }

  /* Refund share balance to seller address, mark Trade as cancelled */
  function cancelTrade(address trade) onlyTransferAgent {
    require(trade.pending);
    balanceOf[trade.seller] += trade.numShares;
    trade.cancel();
    TradeCancelled(trade);
  }
}
```

SecuritiesTrade

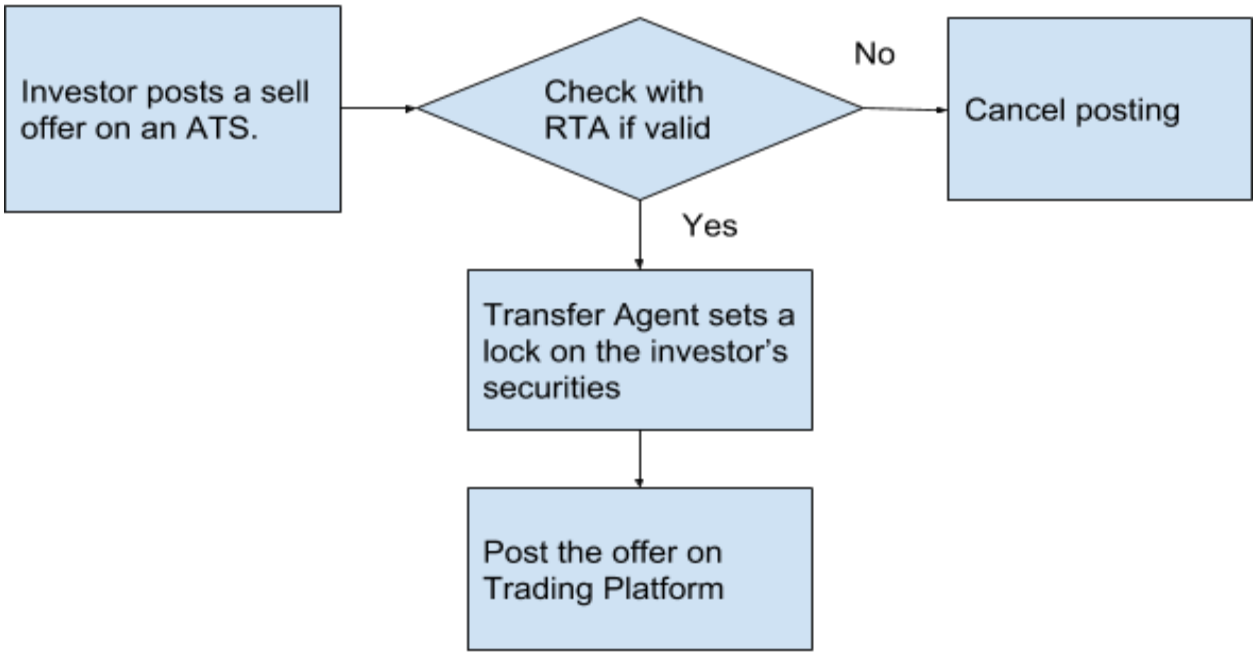
SecuritiesTrade is a Dapp that contains the seller's blockchain address, the buyer's blockchain address, number of securities in the transaction, completed or not status, date when the listing was posted and date when the trade was completed (optional). The broker-dealer pays per trade in StartEngine Coins for each trade initiation.

requestTrade() is a function that allows an investor to transfer one or more securities to another investor. First, the Broker-Dealer checks if the transaction is permitted per SEC and State blue sky laws, verifies the standard purchase agreement terms, gets the seller from the Transfer Agent API call using the blockchain address as the key and the buyer from the Trading Platform which onboarded the buyer. Then the Transfer Agent needs to approve this transfer and creates the new investor and add the investor to the list of investors [Gets the buyer identification from the Trading platform API call using the blockchain address as the key]

newshareholders() is a function to create a new investor with a new address and zero balance. This is initiated by the Transfer Agent [or should it be by the Trading Platform so a buyer can get an address prior to purchasing the securities ?]

```
/*
  Represents a Trade Request to be settled by transfer agent, has no real internal balance but instead is more like a struct that gathers info necessary to model a real world trade
*/
contract Trade is ownedByStartEngine {
  address seller;
  address buyer;
  uint256 numShares;
  bool settled;
  bool cancelled;
  bool public pending;

  function Trade(
    address _seller,
    address _buy,
    uint256 _numShares,
    address _transfer_agent,
  ) {
    seller = _seller;
    buyer = _buyer;
    numShares = _numShares;
    transfer_agent = _transfer_agent;
    pending = true;
  }

  modifier onlyPending {
    require(pending);
    _;
  }

  function settle() public onlyTransferAgent, onlyPending {
    pending = false;
    settled = true;
  }

  function cancel() public onlyTransferAgent, onlyPending {
    pending = false;
    cancelled = true;
  }
}
```

Selling securities for sale on a ATS

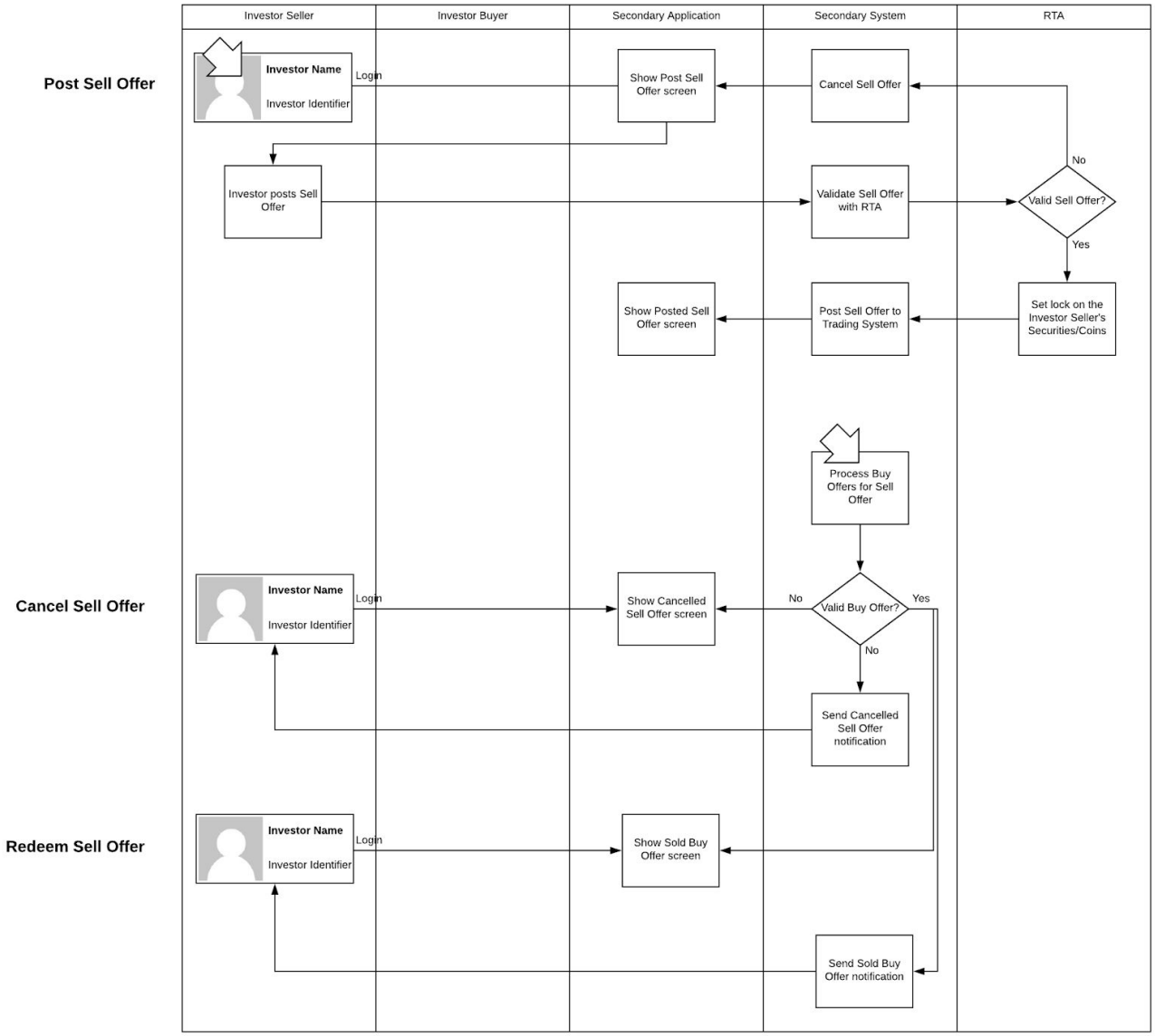


LDGR Technical Diagrams

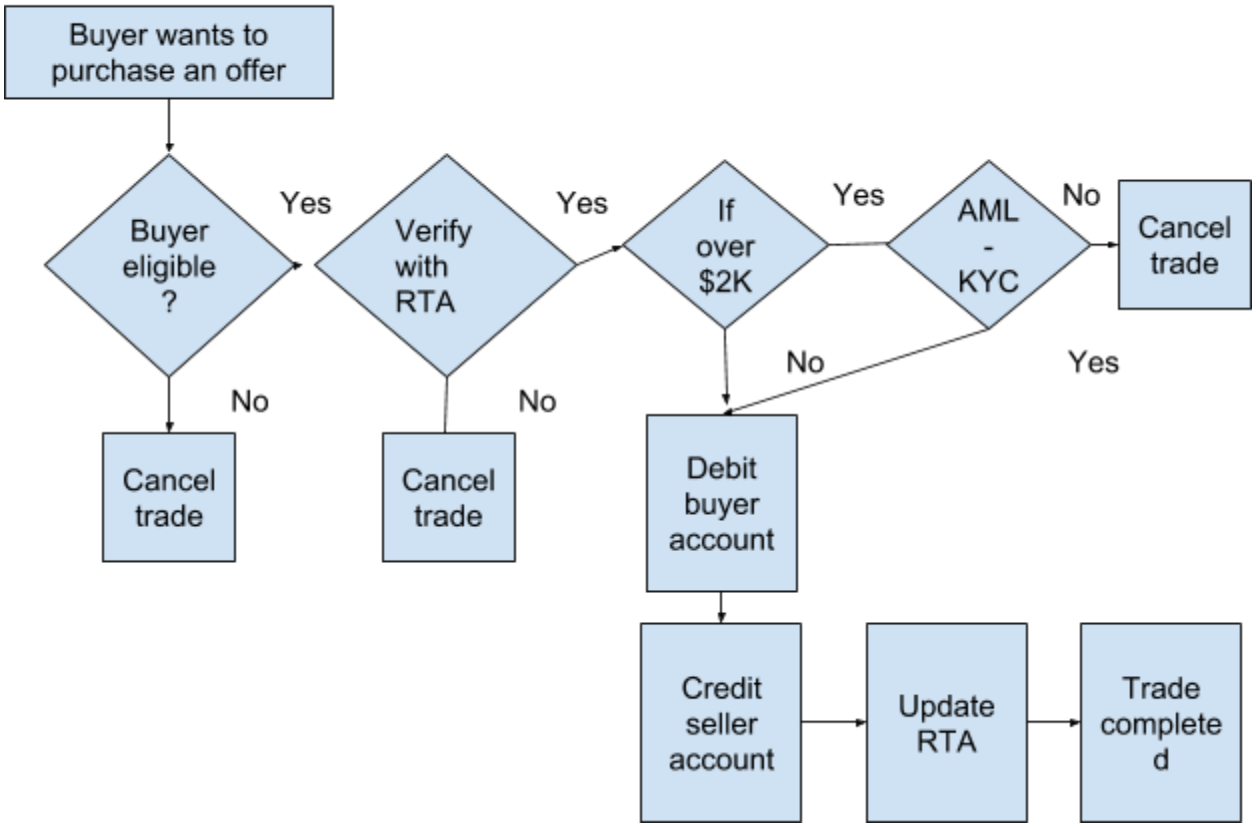
StartEngine

Sell Offer

Technical Specification



Buying Securities on an ATS

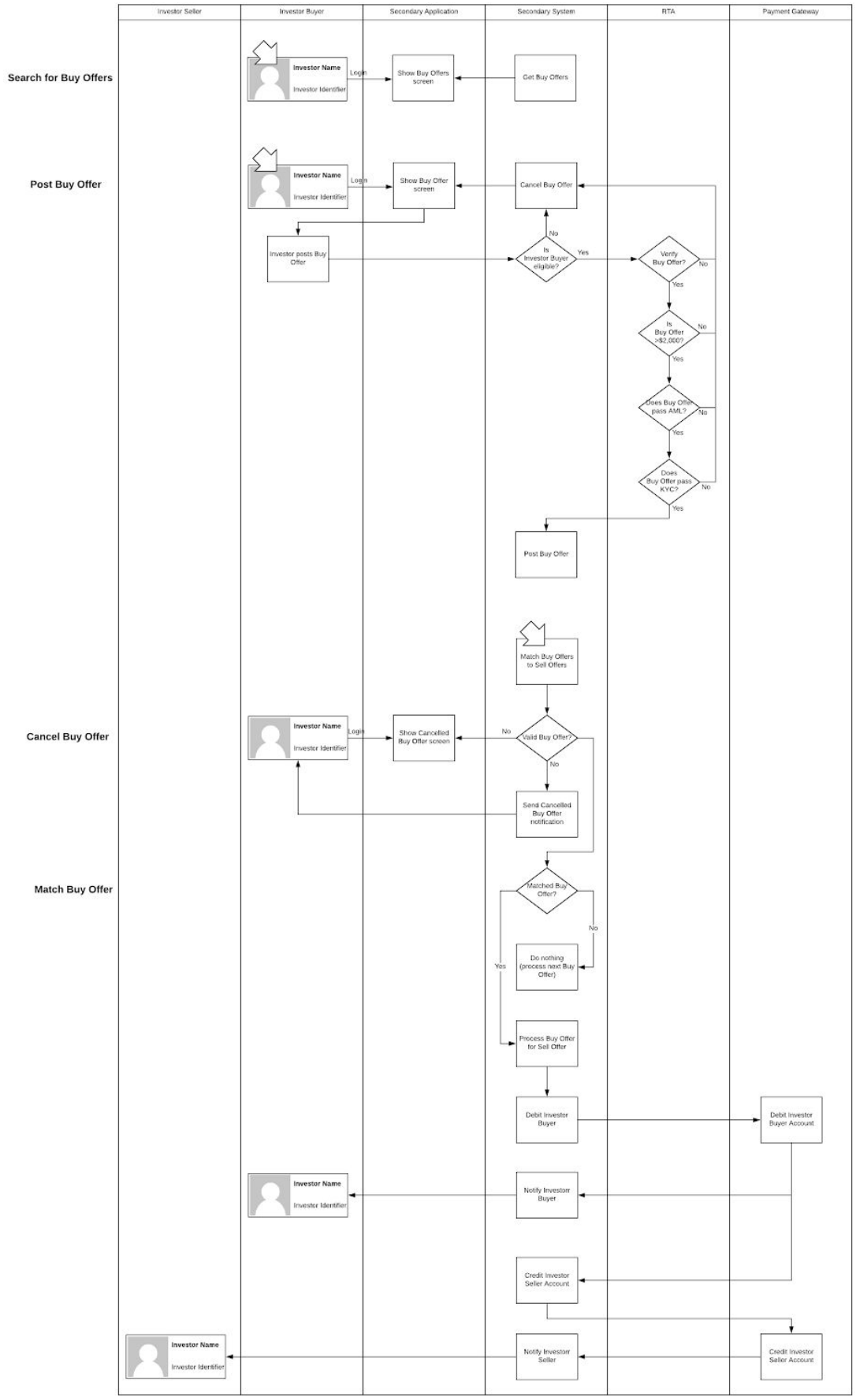


LDGR Technical Diagrams

Buy Offer

StartEngine

Technical Specification



```
pragma solidity ^0.4.8;

/*
  base contract that sets deployer of contract as owner, initializes a transfer agent and provides some modifiers and ability to change transfer agent
*/
contract ownedByStartEngine {
  address owner;
  address issuer;
  address transfer_agent; // transfer agent acting as oracle
  mapping (string => address) offerings;

  function ownedByStartEngine() {
    owner = msg.sender;
  }

  event NewOffering(string company_name, string symbol, address newOffering);

  /* Only performable by StartEngine */
  modifier onlyOwner {
    require(msg.sender == owner);
    _;
  }

  /* Only performable by Transfer Agent */
  modifier onlyTransferAgent {
    require(msg.sender == transfer_agent);
    _;
  }

  function createOffering(string _company_name, string _symbol, address _issuer, address _transfer_agent) onlyOwner {
    require(!offerings[_symbol]);
    address newOffering = new Offering(_company_name, _symbol, _issuer, _transfer_agent);
    NewOffering(_company_name, _symbol, newOffering);
    offerings[_symbol] = newOffering;
  }

  /* Allows StartEngine to set a new transfer agent */
  function updateTransferAgent(address newTransferAgent) onlyOwner {
    transfer_agent = newTransferAgent;
  }
}
```

```
/*
  Offering that represents a company, has a mapping of addresses (investors) to an int balance
(shares held by address), is a factory to create Trades
*/
contract Offering is ownedByStartEngine {
  string public company_name; // name of company that issued shares
  string public symbol; // nice asset symbol for display purposes
  address[] disbursements;

  /* Constructor, owner = StartEngine, initial supply is 0 until company disburses */
  function Offering(
    string _company_name,
    string _symbol,
    address _issuer,
    address _transfer_agent
  ) {
    company_name = _company_name;
    symbol = _symbol;
    issuer = _issuer;
    transfer_agent = _transfer_agent;
  }

  function createDisbursement() onlyOwner {
    string token_symbol = _symbol + (disbursements.length + 1)
    address newDisbursement = new Disbursement(token_symbol, issuer, transfer_agent);
    disbursements.push(newDisbursement);
  }
}

contract Disbursement is ownedByStartEngine {
  mapping (address => uint256) balanceOf;
  string public token_symbol;
  uint256 totalSupply;
  address[] trades;

  function Disbursement(_token_symbol, _issuer, _transfer_agent) {
    token_symbol = _token_symbol;
    issuer = _issuer;
    transfer_agent = _transfer_agent;
    totalSupply = 0;
  }
```

```
  event TradeRequested(address indexed newTrade); // Transfer agent can listen to TradeRequested events, and approve / cancel
  event TradeSettled(address indexed trade); // Event signaling Trade has been settled, buyer receives shares
  event TradeCancelled(address indexed trade); // Event signaling Trade has been cancelled, seller is refunded shares


  /* Adds new shares that have just been issued to investor */
  function newShareHolder(address investor, uint256 sharesSold) onlyTransferAgent {
    balanceOf[investor] += sharesSold;
    totalSupply += sharesSold;
  }

  /* Create a new Trade contract, store it in a list, and also send out an event with the address of the new contract */
  function requestTrade(
    address buyer,
    uint256 numShares
  ) public returns(address newTrade) {
    address seller = msg.sender;
    require(balanceOf[seller] >= numShares); // require the seller to have enough shares
    /* remove shares from seller, "hold" them in a pending Trade and send out event */
    balanceOf[seller] -= numShares;
    newTrade = new Trade(seller, buyer, numShares, transfer_agent);
    trades.push(newTrade);
    TradeRequested(newTrade);
    return newTrade;
  }

  /* Send shares balance to buyer address, mark Trade as settled */
  function approveTrade(address trade) onlyTransferAgent {
    require(trade.pending);
    balanceOf[trade.buyer] += trade.numShares;
    trade.settle();
    TradeSettled(trade);
  }

  /* Refund share balance to seller address, mark Trade as cancelled */
  function cancelTrade(address trade) onlyTransferAgent {
    require(trade.pending);
    balanceOf[trade.seller] += trade.numShares;
    trade.cancel();
```

```
    TradeCancelled(trade);
  }
}

/*
  Represents a Trade Request to be settled by transfer agent, has no real internal balance but instead is more like a struct that gathers info necessary to model a real world trade
*/
contract Trade is ownedByStartEngine {
  address seller;
  address buyer;
  uint256 numShares;
  bool settled;
  bool cancelled;
  bool public pending;

  function Trade(
    address _seller,
    address _buy,
    uint256 _numShares,
    address _transfer_agent
  ) {
    seller = _seller;
    buyer = _buyer;
    numShares = _numShares;
    transfer_agent = _transfer_agent;
    pending = true;
  }

  modifier onlyPending {
    require(pending);
    _;
  }
  function settle() public onlyTransferAgent, onlyPending {
    pending = false;
    settled = true;
  }

  function cancel() public onlyTransferAgent, onlyPending {
    pending = false;
    cancelled = true;
  }
}
```