EXHIBIT 3N

BSTX Security Token Protocol

BSTX-listed security tokens ("Security Tokens") are equity securities that are able to use the functionality of distributed ledger or "blockchain" technology to maintain ancillary records of ownership on the Ethereum blockchain.[1] An existing state of Security Token ownership on the books and records of market participants is replicated on the Ethereum blockchain through the automatic operation of a "smart contract."[2] The BSTX Security Token Protocol (the "Protocol") is the set of rules and permissible operations of the smart contract that facilitate changes in Security Token ownership to be recorded to the Ethereum blockchain as an ancillary recordkeeping mechanism. Each Security Token would have its own smart contract operating pursuant to the Protocol to facilitate this ancillary recordkeeping process.

The discussion below first provides a general background on smart contracts, tokens, and protocols followed by a specific discussion of the Protocol as it will govern each Security Token (*i.e.*, the smart contract governing the equity security of a BSTX-listed issuer for purposes of facilitating the ancillary recordkeeping mechanism using the Ethereum blockchain).

Smart Contracts

The term "smart contract" is commonly used to describe computer-coded functions in connection with the Ethereum blockchain. An Ethereum smart contract is neither "smart" nor a legal contract in the traditional sense. Smart contracts in this context refer to immutable[3] computer programs that run deterministically[4] in the context of the Ethereum Virtual Machine.[5] Smart contracts operate within a very limited execution context. They can access their own state, the context of the transaction that called them, and some information about the most recent blocks (*i.e.*, the most recent recording of transactions and other events recorded to the Ethereum blockchain).

[1] The Ethereum blockchain is an open-source, public distributed ledger (*i.e.*, a blockchain) that records certain information relating to the operation of various smart contracts built on Ethereum.

[2] The Ethereum blockchain is used only as an ancillary recordkeeping mechanism to reflect the number of securities owned by a market participants at a fixed point in time. It will not be used to effect the trading, clearance or settlement of transactions in Security Tokens. Pursuant to the rules of BSTX, Security Tokens are uncertificated equity securities that have been made eligible for services by The Depository Trust Company ("DTC"). DTC would serve as the securities depository for Security Tokens, and confirmed trades on BSTX will be transmitted to National Securities Clearing Corporation ("NSCC") for clearing such that NSCC would clear the trades through its systems to produce settlement obligations that would be due for settlement between participants at DTC.

[3] Smart contracts are immutable in that, once deployed, the code of a smart contract cannot change. Unlike with traditional software, the only way to modify a smart contract is to deploy a new instance.

[4] Deterministic in this context means that the outcome of the execution of a smart contract is the same for everyone who runs it, given the context of the transaction that initiated its execution.

[5] The Ethereum Virtual Machine can be understood as a global computer on which smart contracts run.

EXHIBIT 3N

In the context of Security Tokens, smart contracts generally may have three components: (i) functions, (ii) configurations; (iii) and events.[6] Functions describe the basic operations of a smart contract, such as the ability to query a particular address to determine how many tokens belong to that address.[7] Configurations are attributes of a smart contract that are typically set at the launch of a smart contract, such as designating the name of the smart contract (*e.g.,* as XYZ Security Token). Events describe the functions of a smart contract that, when executed, result in a log or record being recorded to the Ethereum blockchain, such as the transfer of tokens from one address to another. Not all functions of a smart contract result in a log or record being recorded to the Ethereum blockchain. Smart contracts only run if they are called by a transaction.[8]

Smart contracts can call another smart contract, which can call another contract, and so on. Smart contracts never run "on their own" or "in the background," but rather lie dormant until a transaction triggers them to carry out a specified operation pursuant to the protocol on which they operate. All transactions execute in their entirety or not at all, regardless of how many smart contracts they call or what those smart contracts do. Only if a transaction successfully executes in its entirety is there an "event" representing a change to the state of the blockchain with respect that transaction. If an execution of a smart contract's operation fails due to an error, all of its effects (*e.g.*, events) are rolled back as if the transaction never ran.[9]

Tokens

Tokens historically referred to privately issued, special-purpose coin-like items (*e.g.*, laundry tokens or arcade game tokens). In the context of blockchain technology, tokens mean blockchain-based abstractions that can be owned and that represent assets, currency, or access rights. A security token on the blockchain used for ancillary recordkeeping of ownership can be thought of as a digital representation of shareholder equity in a legal entity organized under the authority of state or federal law and that meet BSTX's listing standards. Having a Security Token attributed to a particular address, however, does not convey ownership of shareholder equity in the issuer because the official records of ownership are maintained by participants at DTC.[10]

[6] However, a smart contract need not necessarily have each of these components. Some smart contracts may simply be used to support the functioning of other smart contracts and may not itself result in events being recorded to the Ethereum blockchain.

[7] An "address" in this context refers to a number that is associated with a particular market participant within the smart contract that can be updated to reflect changes in ownership of tokens.

[8] The term "transaction" in this context does not refer to an actual securities execution or transaction occurring on BSTX or in the marketplace, but rather to an operation triggering a smart contract to carry out its specified function, which must ultimately originate from a human source.

[9] A failed transaction (*i.e.*, an attempted operation of a smart contract) is still recorded as having been attempted, but it otherwise has no effect (*e.g.*, as a change in registered ownership on the blockchain as an ancillary record).

[10] Rather, a digital representation of a Security Token associated with a particular address reflects an ancillary record of Security Token ownership based on data provided to BSTX by market participants. The records

EXHIBIT 3N

To create a new token on Ethereum, including for purposes of facilitating ancillary recordkeeping of Security Token ownership, one must create a new smart contract. The smart contract would be configured to detail, among other things, the name of the issuer and the total supply of the tokens. Smart contracts can be designed to carry out any event that one wants, but using a set standard or protocol allows for participants transacting in those smart contracts to have uniform expectations and functionality with respect to the tokens.

Protocols

A protocol (also sometimes referred to as a "standard" or "protocol standard") defines the functions, events, configurations, and other features of a given smart contract. The most common protocol used with Ethereum is the ERC-20 protocol, which describes the minimum functions that are necessary to be considered an ERC-20 token. The ERC-20 protocol offers basic functionalities to transfer tokens, obtain account balances, and query the total supply of tokens, among other features. The BSTX Security Token Protocol is compliant with the ERC-20 protocol, but adds additional requirements and functionality, as described below.

Ether is the digital currency used to pay fees associated with operating smart contracts (known as "gas") on the Ethereum network. Payment of gas is required to operate smart contracts because there are costs involved in performing the computations necessary to execute a smart contract and to record any state transitions onto the Ethereum blockchain.

There is an important conceptual distinction between ERC-20 tokens, including Security Tokens, and ether itself. Where ether is transferred by a transaction that has a recipient address as its destination, token transfers occur within the specific token contract state and have the token smart contract as their destination, not the recipient's address. The token smart contract tracks balances and issues events to the Ethereum blockchain. In a token transfer,[11] no transaction is actually sent to the recipient of the token. Instead, the recipient's address is added to a map within the token smart contract itself. In contrast, a transaction sending ether to an address changes the state of an address. A transaction transferring a token to an address only changes the state of the token contract, not the state of the recipient address. Thus, an address is not really full of tokens; rather it is the token smart contract that has the addresses and balances associated with each address in it.

BSTX Security Token Protocol

BSTX Rule 26138 requires that a BSTX listed company's Security Tokens must comply with the Protocol to trade on BSTX. The purpose of this requirement is to ensure that all Security Tokens are governed by the same set of specifications and controls that allow for

reflected on the Ethereum blockchain regarding Security Tokens may not be current to reflect the most recent transactions in the marketplace and may not reflect ownership by all market participants.

[11] A "transfer" in the context of the BSTX Security Token Protocol regarding a Security Token refers to a reallocation of the digital representation of a Security Token on the Ethereum blockchain as an ancillary recordkeeping mechanism to reflect corresponding changes in ownership of the Security Token.

EXHIBIT 3N

ownership of Security Tokens to be recorded to the Ethereum blockchain as an ancillary recordkeeping mechanism.

The Protocol involves three smart contracts. The Asset Smart Contract is the primary smart contract that contains the balances of tokens associated with each address and carries out the functions necessary to reflect changes in ownership for ancillary recordkeeping purposes. There are two accompanying smart contracts that are called by the Asset Smart Contract in executing operations to facilitate updates to the Ethereum blockchain. The first of these is the Registry Smart Contract ("Registry"), which contains the list of permissioned (or "whitelisted") addresses, and the second is the Compliance Smart Contract, which includes a variable list of additional compliance related rules that the Asset Smart Contract must comply with in executing operations to facilitate updates to the Ethereum blockchain. Each of these three smart contracts are described in greater detail below:

(1) Asset Smart Contract – The Asset Smart Contract defines and creates the Security Tokens (*e.g.*, the maximum number of Security Tokens available for a particular issuance) for purposes of the Ethereum blockchain ancillary recordkeeping function and records a list of each BSTX Participant or non-BSTX Participant broker-dealer addresses and the Security Tokens held at each address.

(2) Registry Smart Contract – The Registry Smart Contract (or "Registry") defines the permissions available to different types of market participants to perform certain functions. Under the Protocol, there are five different types of market participants connected with the Registry, each with different abilities and permissions (as detailed below):[12] (1) Contract Owner, (2) Custodian, (3) Broker Dealer, (4) Custodial-Account, and (5) Investor. The Registry also contains the list of whitelisted addresses to which security tokens may be sent and additional information associated with each address (*e.g.,* whether an address has been suspended).

(3) Compliance Smart Contract – The Compliance Smart Contract is the set of rules held in a separate smart contract that a Security Token can be configured to abide by to ensure compliance with applicable laws and regulations (*e.g.*, by restricting a movement of Security Tokens to an address that has not been added to the Registry for purposes of the Ethereum blockchain ancillary recordkeeping mechanism). The Compliance Smart Contract can be modified to add or remove applicable rules in light of changes to applicable regulatory requirements.

Each of these three smart contracts work together to facilitate the ancillary recordkeeping mechanism for Security Tokens using the Ethereum blockchain. The discussion below describes the specific functions and configurations of each.

[12] There are additional roles that are not technically part of the Registry and are instead specific to certain smart contracts. For example, an "Issuer" is an Asset Smart Contract-specific role, as discussed in further detail below. Also, an "Administrator" is a Compliance Smart Contract-specific role that allows such a user to, for example, freeze the transfer of tokens for purposes of the ancillary recordkeeping function under certain circumstances and modify or add compliance rules to govern a Security Token.

EXHIBIT 3N

Asset Smart Contract

The Asset Smart Contract component of the Protocol sets forth 16 functions, 10 configurations, and six events. The 16 functions of the Asset Smart Contract are as follows:

1. "TotalSupply" – a query function that returns the value of the TotalSupplyTokens configuration, which provides the number of Security Tokens created with respect to any issuance of a particular Security Token.
2. "BalanceOf" – a query function that provides the number of tokens held by a given address.
3. "Allowance" – provides a limit to the amount of tokens allowed to be transferred from a given address to another given address when using the TransferFrom function.[13]
4. "HolderAT" – a query function that returns the address of the holders of a Security Token at the time of the query. This function enables an individual to generate a list of every address holding a given Security Token at the time of the query.
5. "IsHolder" – a query function that returns a "yes" or "no" answer to whether a given address holds a Security Token.
6. "IsSuperseded" – a query function that returns a "yes" or "no" answer to whether a given address has been frozen (*i.e.*, canceled, subject to the CancelAndReissue function).
7. "GetSuperseded" – a query function that returns the superseding address (if any) when an address is entered. This function is useful in the context of a lost or forgotten address. For example, if an address has been canceled, this function will return the address that took the place of such canceled address. Entry of a currently active address returns a null set.
8. "SetCompliance" – requires the Asset Smart Contract in executing a transaction to call the Compliance Smart Contract and operate in accordance with and specified requirements set forth in the Compliance Smart Contract (*e.g.*, to not transfer Security Tokens to an address in the Registry that has been frozen for ancillary recordkeeping purposes).
9. "SetRegistry" – requires the Asset Smart Contract to call the Registry, which houses whitelisted addresses and the status (*e.g.*, Custodian) associated with certain address to perform certain functions.

[13] To ensure that Security Tokens remain compatible with the ERC-20 protocol for ancillary recordkeeping purposes, which is the baseline upon which the BSTX Security Token Protocol is built, the Protocol will feature "Allowance," "TransferFrom," and "Approve" functions. However, during the initial stages of Security Token trading these functions will be disabled through a rule in the Compliance Smart Contract, as use of these functions may enable transfer to non-whitelisted wallet addresses. However, such functions may be enabled in the future, if appropriate. BSTX will provide notice to BSTX Participants via regulatory circular in advance of such functionality becoming operational as part of the BSTX Security Token Protocol.

EXHIBIT 3N

10. "Transfer" – allows for the transfer of tokens for ancillary recordkeeping purposes to other specified, whitelisted addresses, and requires two parameters: the receiver address and the amount of tokens being sent. One use for this function is to allow an address with Investor status to transfer tokens to a Custodian.

11. "TransferFrom" – allows a Contract Owner (described below) to delegate to a separate address the ability to spend tokens on its behalf.[14]

12. "Approve" – allows the owner of an address to approve execution of the TransferFrom function.[15]

13. "SetIssuer" – allows a Contract Owner to designate the address of an Issuer of a Security Token.

14. "IssueTokens" – allows an Issuer to issue new Security Tokens.

15. "FinishIssuing" – when this function is executed by an Issuer, there may be no additional issuances of a given Security Token.

16. "CancelAndReissue" – in the context of a lost, forgotten, or inaccessible address, this function allows a Contract Owner, Issuer, or Custodian to transfer Security Tokens to an address that previously did not hold a given Security Token and disallows the inaccessible address from holding any amount of such Security Token for ancillary recordkeeping purposes.

The BSTX Protocol allows for the ten configurations discussed below.

1. "Cancelations" – stores information about addresses that enables linking of a particular address to a successor address. When a Contract Owner executes the "CancelAndReissue" function, the Cancellations configuration stores the results.

2. "Balances" – stores records of addresses and their associated balances of a given Security Token and is updated upon a transfer of Security Tokens.

3. "Allowed" – stores records of execution of the "Approved" function.[16]

4. "Issuer" – when a Contract Owner confers Issuer status upon a given address using the SetIssuer function, this configuration stores information related to execution of such function. For example, Company ABC may be designated as the Issuer of a Security Token and this configuration stores the appropriate address of Company ABC.

[14] *See id.*

[15] *See id.*

[16] Note that the Allowed configuration will be dormant during the initial stages of Security Token Trading because the "Allowance" function is disabled. *See supra* note 13.

EXHIBIT 3N

5. "IssuingFinished" – when a Contract Owner or Issuer executes a FinishedIssuing function for a given Security Token, this configuration is made active and no new issuances for such Security Token may occur.

6. "Compliance" – when a Contract Owner employs the SetCompliance function, the Compliance configuration specifies the particular Compliance Smart Contract(s) to which the Security Token (*i.e.*, Asset Smart Contract) must call upon in executing transactions.

7. "Name" – allows a Contract Owner or Issuer to input and store the name of a Security Token. For example, a Security token may be named "Company ABC Preferred Series A1."

8. "Symbol" – allows a Contract Owner Issuer to input and store the ticker symbol of a Security Token.

9. "Decimals" – if enabled, this configuration would allow for fractional interests in Security Tokens. However, the BSTX Exchange and certain potentially applicable laws, including laws of the State of Delaware, do not allow for fractional share interest in Security Tokens.

10. "TotalSupply" – sums the amount of tokens issued through execution of the "IssueTokens" function for a given Security Token.

The Asset Smart Contract specifies six events. A description of each of these events is below. As noted, the occurrence of an event generates a record on the Ethereum blockchain that is publicly viewable.

1. "Transfer" – this event records the details of the movement of the digital representation of Security Tokens from one address to another, as recorded in the ledger of the Asset Smart Contract for purposes of ancillary recordkeeping.

2. "Approval" – this event records successful execution of the Approve function, which is used in conjunction with TransferFrom, which, as noted, will be disabled initially for Security Tokens.

3. "VerifiedAddressSuperseded" – this type of event occurs when a user successfully executes the CancelAndReissue function; a log of the superseded and replacement addresses is created.

4. "IssuerSet" – this type of event occurs upon successful designation of an Issuer of a Security Token; a log of the Issuer's address is created.[17]

5. "Issue" – this type of event occurs whenever an Issuer successfully executes the "IssueTokens" function; a log of the Issuer's address and amount of Security Tokens issued pursuant to the most recent execution is created.

[17] Note that once an Issuer has been designated such Issuer may perform the tasks discussed herein.

EXHIBIT 3N

6. "IssuerFinished" – this event occurs when an Issuer successfully executes the FinishedIssuing function; a log is created noting that new issuance of a given Security Token can no longer occur.

Registry

To facilitate recording an existing state of ownership of Security Tokens on the Ethereum blockchain as an ancillary recordkeeping function, the Protocol requires that the Asset Smart Contract call the Registry Smart Contract that grants certain permissions to different types of market participants, and the Registry stores the list of whitelisted addresses that may transact in Security Tokens. For example, in the event that an address is lost or otherwise compromised, the "CancelAndReissue" functionality would be employed to facilitate the replacement of the lost digital representation of the Security Tokens on the Ethereum blockchain as an ancillary record of ownership, but only certain permissioned market participants may exercise this function.

The Registry designates five different types of market participants under the Protocol, each of which is discussed below along with their respective permissions. The permissions associated with each market participant are organized as a hierarchy from the most permissive (*i.e.*, can perform the most functions) to the least permissive (*i.e.,* can perform the fewest number of functions). Each type of market participant described below is able to confer statuses below their permissioned level to additional addresses, but never above their status (*e.g.*, a Custodian may confer Broker Dealer status to additional addresses, but may not confer Contract Owner status to an address). The different classes of market participants specified by the Registry are as follows:[18]

1. "Contract Owner" – only one individual address can be designated for this role for a given Security Token. This role is specified at the time of Security Token deployment. Among other functions, a Contract Owner confers "Issuer"[19] status to an address and adds/removes whitelisted addresses with "Custodian" status.
2. "Custodian" – multiple addresses may be designated as Custodians of a Security Token. Custodians may add/remove whitelisted addresses of "Broker Dealer" parties (*e.g.,* introducing brokers) and "Investors." A Custodian may also temporarily suspend Broker Dealers (and one or more of their respective customers) and Investors from transferring the digital representation of Security Tokens for purposes of updates to the Ethereum blockchain as an ancillary recordkeeping mechanism. Custodian status would generally be assigned to BSTX Participants that act as a carrying broker-dealer on behalf of other broker-dealers.
3. "Broker Dealer" – represents an address of an introducing broker who can add/remove whitelisted addresses of "Investors" and direct Custodians to transfer digital

[18] For a given Security Token, multiple wallet addresses may function as Broker Dealer, including wallet addresses carrying out other roles in the status hierarchy (*e.g.*, a Custodian may also be a Broker Dealer).

[19] As discussed above, while Issuer status confers privileges onto the owner of a designated wallet address, Issuer status is technically not part of the Registry and is instead a Security Token-specific designation.

EXHIBIT 3N

representations of Security Tokens to certain whitelisted parties for purposes of the ancillary recordkeeping mechanism. A Broker Dealer may suspend Investors from transferring Security Tokens for purposes of the ancillary recordkeeping mechanism.[20]

4. "Custodial-Account" – this is a wallet address established by a Custodian to maintain custody of the digital representation of Security Tokens on behalf of the Broker-Dealer for purposes of the ancillary recordkeeping mechanism. Custodial-Accounts may only transfer and receive the digital representations of Security Tokens for purposes of the ancillary recordkeeping mechanism.[21]

5. "Investor" – this status generally applies to individual retail investors who may want their own address. This status confers the least amount of rights to an address. An Investor may only transfer digital representations of Security Tokens to his/her address (*e.g.*, by requesting that a Broker Dealer or Custodian perform this function) and must transfer the digital representation of the Security Tokens back to a Custodian in order to trade the Security Tokens in a manner that would allow for updates to the Ethereum blockchain as an ancillary recordkeeping mechanism. For a given Security Token, multiple addresses may function as an Investor.

Compliance Smart Contract

The Compliance Smart Contract is a set of rules that are called upon by an Asset Smart Contract in executing a transaction for purposes of the ancillary recordkeeping function depending on how a Security Token is configured. For example, when the digital representation of a Security Token (*i.e.*, Asset Smart Contract) is executing a transaction (*e.g.*, an instruction to move a digital representation of Security Tokens from one address to another to reflect a change in record ownership as an ancillary record), the SetCompliance function requires that the digital representation of the Security Token look to the Compliance Smart Contract that was configured for the digital representation of the Security Token and comply with any rules or requirements therein. One such rule is to require that the digital representation of the Security Token look to the Registry to determine if the transaction is with another whitelisted address for purposes of updates to the Ethereum blockchain as an ancillary recordkeeping mechanism. The rules configured in the Compliance Smart Contract can be tailored to each particular Security Token.

The Compliance Smart Contract might also specify additional Compliance Smart Contracts with which the Security Token must comply. As a result, to the extent a digital representation of a Security Token needs to comply with a new rule of some kind, that new rule can be programmed into an additional Compliance Smart Contract to which the digital

[20] A Broker Dealer address will generally not have digital representations of Security Tokens associated with such address for purposes of the Ethereum blockchain ancillary recordkeeping mechanism. However, such an address may be linked to a Custodial-Account address and actions taken by a Broker Dealer (including certain transfers to and from its Custodial-Account address) may be audited and traced back to the individual owner of the Broker dealer address.

[21] Whereas a Broker Dealer represents a wallet address for an introducing broker, a Custodial-Account represents the account carried by a Custodian on a Broker Dealer's behalf.

EXHIBIT 3N

representation of the Security Token must comply when executing. The BSTX Security Token Protocol uses the Compliance Smart Contract functionality to allow for changes to the rules and requirements applicable to a Security Token when carrying out functions (such as transfers) with respect to the Ethereum blockchain ancillary recordkeeping mechanism without redeployment of the entire Security Token. In the absence of this structure, adding an additional rule could require a recall and reissuance of the digital representation of the Security Token to accommodate new rules.