Citizen Data:
A Multiprotocol Polychain Data Sync Network with Validation, Verification and Delayed Reveal

Technical White Paper

Jonathan Alexander, Steven Landers
Citizen Data

December 9, 2018
Version 1.10

Introduction

This paper introduces the Citizen Data Network which will be a multiprotocol and polychain layer 2 network for data validation, verification and encryption that can be used to synchronize data and documents to blockchain ledgers and decentralized file storage systems and can be integrated with existing systems through connectors or mobile and web applications through available protocols. The core functions of the Citizen Data Network will be controlled by smart contracts executing on decentralized consensus blockchain platforms. The Citizen Data Network is an evolution of the Netvote Project for blockchain-based elections.

The Citizen Data Network will provide data and document synchronization, validation, verification, and security for a wide variety of data types and document formats. It will integrate with multiple public or private blockchains and decentralized file storage systems. Clients who utilize the Citizen Data Network will be able to synchronize to any of the supported blockchains and decentralized file storage systems with full portability.

This paper presents the core features, including privacy and security features, of the Citizen Data Network as well as future architecture vision.

Data Groups

Registered protocols, whose public keys are listed in the network-controlled protocols smart contract, will be used to create new Data Groups to a smart contract on the Citizen Data Network. Each Data Group will be associated to a unique Data Group ID that will be used in subsequent data transactions. It is expected that each protocol will have many Data Groups, for example a voting protocol might create a separate Data Group for every ballot in each election.

For each Data Group the controlling protocol will also be required to provide the following attributes which will be stored on the Data Groups smart contract (each attribute described more below):

- Permission-Authority Public Key used to verify Permission Tokens used on data transactions
- Administrator Public Key used to verify changes to the Data Group
- File reference address(es) of JSON, XForms or other schema(s) to validate data submissions
- Flag indicating whether clients can issue (logical) delete or update transactions

Optional data elements may also be specified that will be synchronized to the blockchain or decentralized storage system along with the validated data.

For each Data Group that is created, the smart contract will trigger the creation of encryption keys that are stored in a secured software vault that will only be accessible to registered Validator nodes, and a Public Key for the Data Group that will be stored on the smart contract. The keys stored in the vault will subsequently be deleted or revealed based on state changes made on the Data Groups smart contract. These encryption keys are described further in sections below.

Each Data Group will have a series of states for submissions and encryption/reveal. Data Groups will originally be set to the **Locked** submission state and a **Not Revealed** encryption state. The state values will be updated by transactions on the smart contract that must be signed by the private key of the Data Group Administrator, with the following available states:

- Submission states: **Locked, Opened, Paused, Closed**
- Encryption states: **Not Revealed, Partially Revealed, Fully Revealed**

Data Submissions, Updates and Deletes

To submit data to the Citizen Data Network, the submitter will be required to have a signed Permission Token that has been provided by a Permission Authority for the target Data Group. The Permission Authority will be associated to an application and outside of the Citizen Data Network. The Permission Token will be a cryptographically signed payload that must contain the Data Group ID and a Global Unique Identifier (GUID) for each citizen as will be described below. The Citizen Data Network will never reveal the association of a Permission Token or GUID to any submitted data in order to avoid any association of citizens to individual data records.

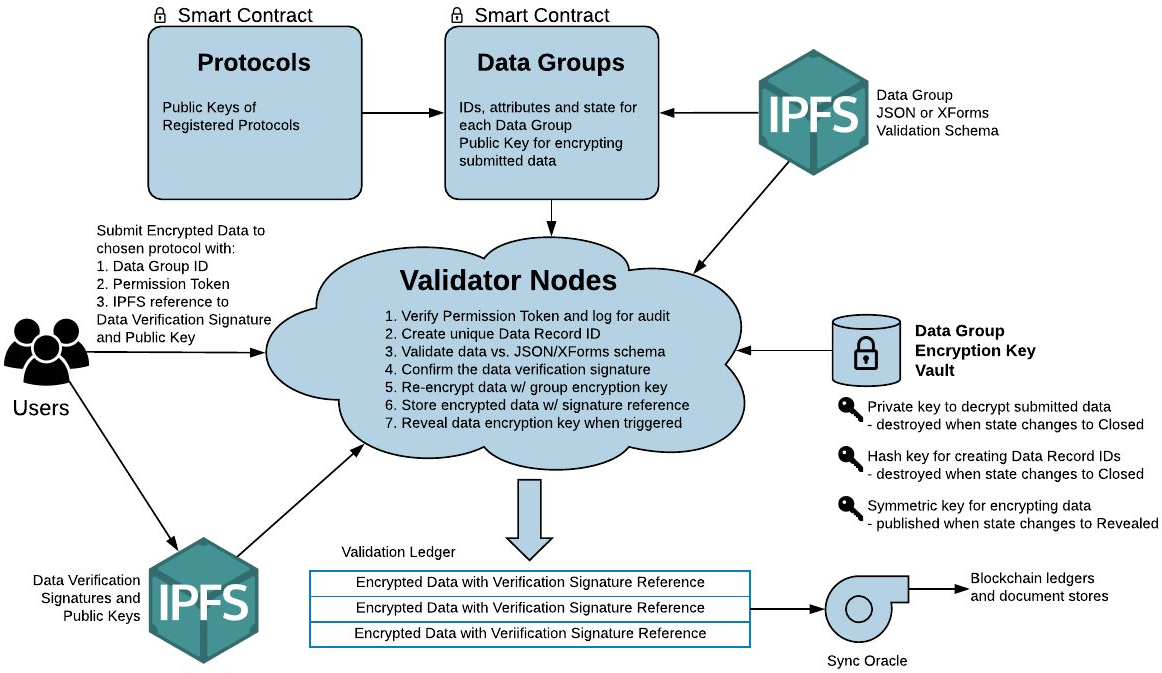
Transactions submitted to the Citizen Data Network must be encrypted with the Public Key of the Data Group and sent to the Validator node, and will be required to include the following elements:

- Permission Token signed with the Permission-Authority Private Key
- Data Buffer that must conform to the validation criteria set for the Data Group
- Reference to a stored Verification Signature and associated Public Key (more details below)

For Data Groups that allow logical updates or deletes as indicated on the Data Group smart contract, the update and deleted transactions will be required to submit a Permission Token with the same GUID as provided in the original (prior) data submission. For Data Groups that do not allow updates, the Citizen Data Network will ensure that each GUID will be used for only one data submission. Updates to the Citizen Data Network will result in the storage of multiple data records preserved on the blockchain, but only the final (last) data record will be included when retrieving active data records for the associated Data Group. Deletes will result in the storage of a data record preserved on the blockchain indicating that a prior data record was logically deleted so that the associated prior data record will not be included when retrieving active data records for the associated Data Group.

Data Verification and Validation

The core verification and validation services of the Citizen Data Network will be supplied by the Validator nodes. Validator nodes, which could be implemented as smart contracts run by independent participants (Validators) who have bonded a Proof of Stake, will execute the verification and validation logic for every data submission.



The Citizen Data Network will protect privacy and provide verifiability and coercion resistance for submitted data with:

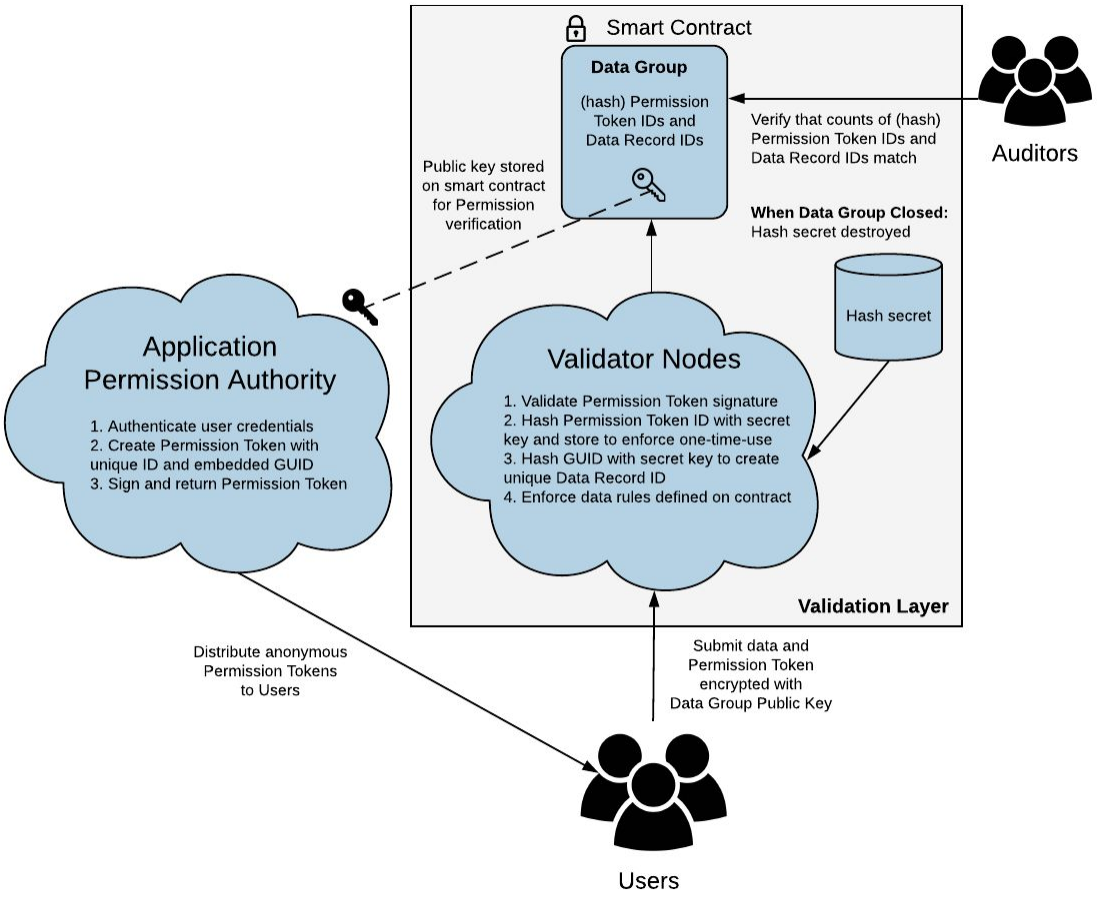
- Controlled access to Data Groups with identity protection via signed permission tokens
- A receipt-free data commitment and data storage process
- Verification Signatures which will be checked to ensure that is submitted and stored as intended
- Recording of all data transactions on the blockchain for tamper-proof auditability

The following sections provide more details on these functions.

Step One: Permission Token Verification and Audit Logging

Permission on data submission transactions to a Data Group will be enforced via the use of Permission Tokens which will be required to have a cryptographic signature that can be verified via a Public Key published on the associated Data Group smart contract.

Global unique identifiers (GUIDs) must be generated by a Permission Authority for each citizen and each Data Group and will not require the use of any citizen personal identifier. The Citizen Data Network itself will not create GUIDs or Permission Tokens, but it will verify Permission Token signatures and ensure that each Permission Token is used only once.



The Permission Token payload must contain the following data elements:

- Global unique identifier (GUID) for the citizen (must be unique within Data Group)
- Data Group ID for the Data Group to which the data is associated
- Time that the token was created and at which the token will expire
- A unique identifier for the token itself (must be unique within Data Group)

In its first step the Validator node will validate the signature of each Permission Token and will ensure that the token has not been expired or already used for the identified Data Group.

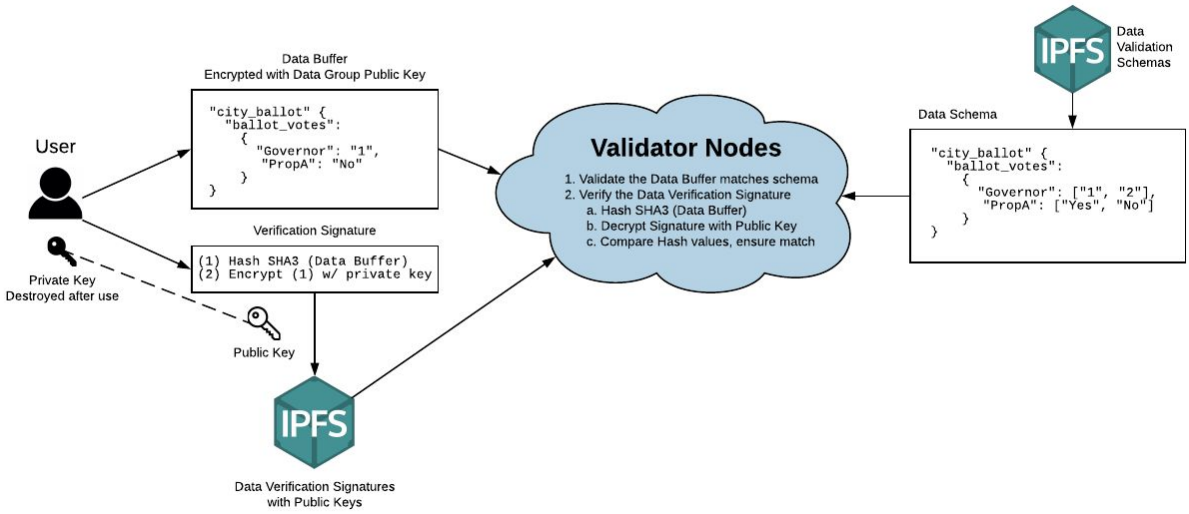
For audit purposes and to control reuse, the Validator node will use a secret and temporary hash key (more below) to hash the Permission Token ID and will store that hashed value on the ledger or on another storage location. The Validator node will then use that data to ensure that each Permission Token will be used only once.

After the Permission Token has been verified, the Validator node will use the secret hash key to hash the citizen GUID and produce a unique Data Record ID which will be stored on the blockchain and to which the submitted data will be associated. The Data Record ID will be used by the Validator node to enforce the data submission rules defined on the Data Group smart contract, either limiting each GUID to one submission or allowing logical update or delete for the data record. The secret hash key will be destroyed by the Data Group smart contract when the smart contract submission state is changed to Closed, so that it will no longer be possible to generate or recreate Data Record IDs for the Data Group.

Step Two: Data Validation

After the Permission Token and its payload has been verified, and the Data Record ID has been produced, the Validator node will verify that the format of the data submitted matches the requirements for the Data Group.

The Citizen Data Network will support JSON and XForms (see References at the end of this document) and potentially other data formats. The Data Group administrator may store in IPFS a reference JSON document or XForms schema, and then register that IPFS document address on the Data Group smart contract. When data is submitted to the Validator node, it will verify that the submitted data format matches the registered schema format. Note that for multi-part data involving external files (such as media files stored on IPFS or similar) the file references will be expected in the data, not the actual files.



Step Three: Data Verification

In addition to verifying that the data submitted is valid, the Validator node will also check and confirm the Data Verification Signature and the Public Key stored for the data record. This will ensure that the data will be received and stored as submitted, and that every data record submitted will have an associated Verification Signature.

The Data Verification Signature will be produced by the citizen application as follows:

- Generate a one-time use Private/Public RSA key pair
- SHA3 hash the data record
- Encrypt the hash using the Private key
- Destroy the Private key
- Store the encrypted string (the Data Verification Signature) and the Public Key on IPFS or similar storage system
- Submit storage address of the Data Verification Signature and Public Key when submitting the data to the Citizen Data Network

To verify the data submitted is as intended, the Validator node will itself hash the submitted data record, decrypt the Data Verification Signature using the Public Key, and compare the hash results to ensure that they match. That will confirm that the data record is as the citizen intended proven by the Data Verification Signature.

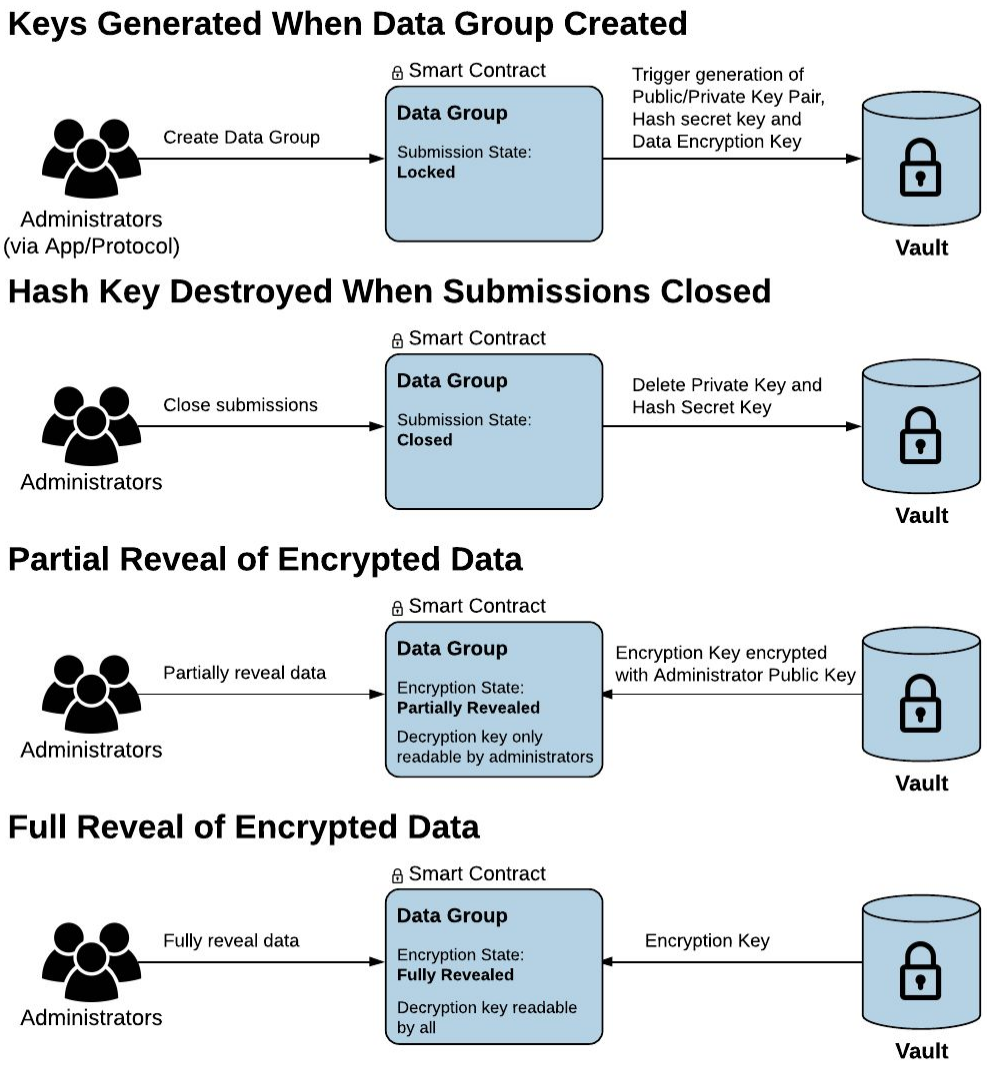
When storing the data, the Validator node will also store the reference to the Data Verification Signature and its Public Key, so that public auditors will also be able to confirm that data was recorded as submitted and intended, and that every record has a Data Verification Signature and that every signature has an associated data record.

Step Four: Data Encryption with Delayed and Targeted Reveal

After validating and verifying the data, the Validator node will re-encrypt the data before storage via the use of a secret encryption key specific to the associated Data Group. The encryption will provide a delayed reveal capability so that the data records will only be revealed when and to whom intended.

The creation of the encryption key will be done via an oracle triggered by the Data Group smart contract when a Data Group is created (the other encryption keys discussed in other sections will also be created at the same time). The encryption key and other Data Group secret keys will be stored in a secure software key vault which will provide:

- Strict access control rules so that only the verified Validator node from registered Validator nodes can read from the key vault
- Full ability for the public to audit the key vault security rules and all access records



To reveal the data records, the Data Group administrator will change the encryption state of the Data Group via a transaction to the smart contract which must be signed using the Administrator's Private Key. Setting the state to Partially Revealed will cause the smart contract to publish the encryption key on the blockchain encrypted with the administrator's public key so that it can only be read by the Data Group administrator. Setting the state to Fully Revealed will cause the smart contract to publish the encryption key on the blockchain so that it can be read by anyone.

Example use cases for the delayed and targeted reveal capability will include:

- Forms and documents can be revealed to individuals with approved access
- Votes in an election can be revealed to everyone for tally when voting is closed
- The signatures on a petition can be revealed only to the target audience
- Election audits can be revealed to audit authorities when voting closes and to the public later
- Survey results can be revealed to researchers and later published to the public

Audit Trail

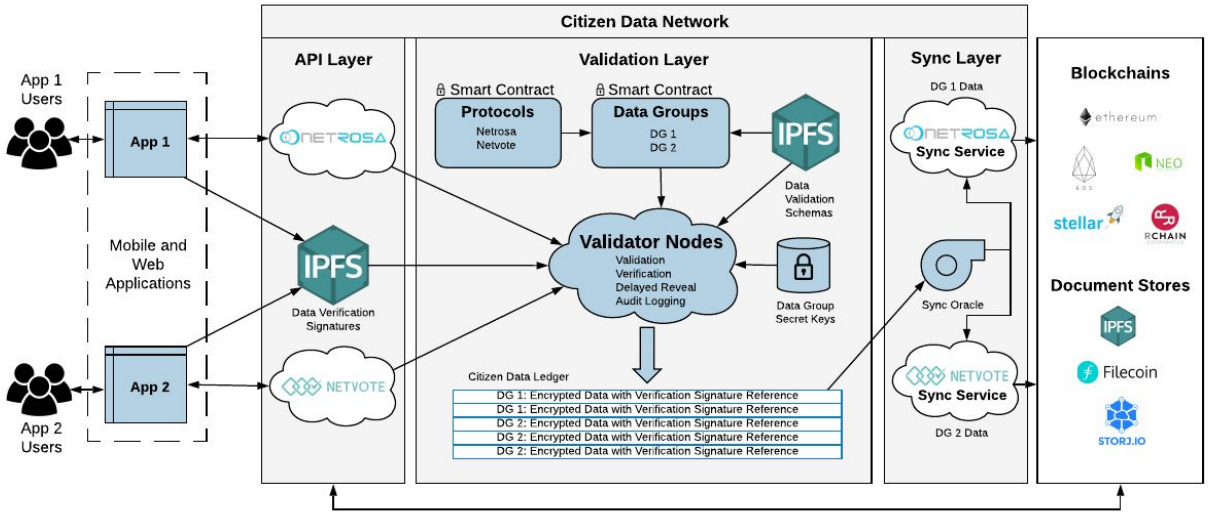
The Citizen Data Network will record data transaction information on tamper-proof data stores including multiple blockchain ledgers and decentralized file storage systems so that auditors will be able to perform the following checks for every Data Group:

- The count of Permission Tokens used matches the count of data records
- Every Data Verification Signature has an associated data record
- Every data record matches its Data Verification Signature
- Encryption keys were created, read, revealed and destroyed as intended and only accessed by registered Validator nodes

Application Connectors

The Citizen Data Network will support multiple application connectors and protocols that will be designed for compatibility with any web or mobile application. Data records processed by the layer 2 Citizen Data Network will be synchronized to the layer 1 blockchain ledgers and file storage systems supported by the network.

Clients will be able to synchronize data and documents via the available connectors, and applications will be able to access the APIs and thereby embed the supported functions which will in turn store verified and validated data on a chosen blockchain and decentralized file storage system.



Web and mobile applications will be able to integrate with the Citizen Data Network via the application programming interfaces (APIs) supported by the available registered protocols. The web or mobile application will communicate with the protocol APIs which will in turn send data transactions to the Citizen Data Network's Validator nodes. Connectors and integrated applications will create and store the Data Verification Signature before sending data through the protocol APIs to ensure that the data will not be tampered with or inadvertently changed. Integrated web and mobile applications will not need to create or maintain wallets or obtain cryptographic tokens to utilize the Citizen Data Network protocols.

The Citizen Data Network will also provide asynchronous synchronization and message queueing capabilities to ensure scalability and performance of data synchronization.

Applications that utilize the Citizen Data Network will be able to read the verified and validated data from the target blockchain and any associated decentralized file storage system. Auditors will also be able to access the verification and validation data directly.

In addition to connectors and APIs, open source application software development kits (SDKs), developer tools, and example programs will be published to facilitate the use of the Citizen Data Network and integration with existing applications and systems.

Disclaimer

This document may contain forward-looking statements including, but not limited to, statements as to future plans that involve risks and uncertainties. The use of words such as “expects”, “anticipates”, “believes”, “estimates”, “will”, “plans”, the negative of these terms and similar expressions identify forward-looking statements. Such forward-looking statements involve known and unknown risks, uncertainties and other factors which may cause the actual results, performance or achievements to differ materially from any future results, performance or achievements expressed or implied by those projected in the forward-looking statements for any reason.

This document does not constitute a prospectus or offer document of any sort and is not intended to constitute an offer of securities or a solicitation for investments in securities in any jurisdiction. No person is bound to enter into any contract or binding legal commitment, and no form of payment is to be accepted based on this presentation.

No regulatory authority has examined or approved of any of the information set out in this document. No such action has been or will be taken under the laws, regulatory requirements or rules of any jurisdiction. The publication, distribution or dissemination of this document does not imply that any such applicable laws, regulatory requirements or rules have been complied with.

There are material risks and uncertainties associated with cryptographic tokens, the authors do not make or purport to make, and hereby disclaim, any representation, warranty or undertaking in any form whatsoever to any entity or person, including any representation, warranty or undertaking as to the accuracy and completeness of any of the information set out in this document.

References

1. Open Data Kit
   https://opendatakit.org/

2. XForms W3C Standard
   https://www.w3.org/standards/techs/xforms#w3c_all

3. JSON Schema
   http://json-schema.org/

4. Cosmos Network
   https://cosmos.network

5. Civitas: Toward a Secure Voting System (Clarkson/Chong/Myers, Cornell Univ.)
   http://www.cs.cornell.edu/projects/civitas/papers/clarkson_civitas.pdf