White Paper

BITAGORA

A Decentralized Voting Platform

Ignasi Ribó*

8 July 2018

Introduction

Distributed transaction ledgers (blockchains) maintained by networks of decentralized nodes are transforming the way individuals hold and exchange value and information across the Internet (e.g. Bitcoin, Ethereum, etc.). Thanks to modern peer-to-peer communication protocols and the use of cryptographic algorithms it is possible to make these networks secure, reliable and confidential, without the need to establish trust between participants [https://bitcoin.org/bitcoin.pdf].

Bitagora is an open source software application that uses this same technology to provide a platform for conducting polls and elections with the same guarantees of reliability, security and confidentiality found in most conventional elections, but without the need for a central authority to oversee and validate the voting process or the results.

The Bitagora Platform relies on a fully decentralized network of validator nodes running a purpose-built application developed on top of Intel's Hyperledger Sawtooth blockchain framework. The network of validator nodes maintain the public transaction ledger (blockchain) where polls initiated by pollsters and ballots submitted by individual voters are stored. Polls and ballots are stored as transactions in a chain of validated blocks that is fully distributed, inalterable, secure, and demonstrably identical across all nodes in the network [https://sawtooth.hyperledger.org/docs/core].

Thanks to the consensus mechanisms of the network, voter's ballots are protected from tampering, even when polls are conducted during an extended periods of time. Voter registration has been separated from the voting system by design. In the Bitagora Platform deployment, voter registration is built as event-driven and customizable code running in an opaque and auditable serverless computing platform. It is a fully automatic process that relies on cryptographic algorithms to ensure that a voter can never cast a ballot in the same poll more than once. This is achieved without storing, disclosing or recording the identity or personal information of voters at any time.

* www.ignasiribo.com

The Bitagora Platform also includes client modules that can be distributed as mobile or web applications and allow voters to register and cast their ballots into the system, as well as other modules that allow pollsters and network administrators to initiate and close polls on the network.

Motivation

The initial motivation for the development of the Bitagora Platform came from the events of October 1st 2017 in Catalonia. The Catalan government, with the support of civic organizations and a large part of the population, had promoted a referendum on independence in spite of the opposition of the Spanish government. As the day of the referendum approached, Spanish police and public prosecutors increased their repressive actions against Catalan citizens and institutions, including the seizure of websites, print houses, postal mail, and any other means by which the referendum could be carried out. In spite of the repression, citizens occupied the polling stations days before the referendum in order to prevent the police from shutting them down. During the day of the referendum, in an exemplary show of self-organization and resilience, thousands of people kept the polling stations open, defending with their bodies the ballot boxes that policemen armed with riot gear tried to seize by force. Anonymous citizens also managed to sustain the computer systems used for voter registration against sustained and coordinated DDoS attacks. Although more than 2 million people were able to cast their ballot, police brutality and the repressive actions of the Spanish authorities seriously disrupted the polling and managed to put in question the final results.

Inspired by the events of October 1st 2017, Bitagora was born in order to provide a confidential, secure and reliable platform for the celebration of polls and other elections under the control and supervision of citizens interconnected in a fully decentralized network that is effectively able to withstand attacks from hostile parties and does not need to rely on any central authority to deliver accurate and auditable results from the vote.

The Bitagora Platform aims to support existing efforts to improve the quality and extent of direct democracy, empowering citizens throughout the world to take in their own hands the management of all kinds of elections, polls and referenda. It is a modular, customizable and flexible tool that can be adapted to different contexts and needs. The software is open source and is available for use under an Apache-2 license. It can be deployed, in part or in full, for the celebration of a single poll or to support any number of successive or simultaneous polls.

Overview of the voting process

A basic principle of the Bitagora Platform is the separation of voter registration (certification) from the actual casting of ballots (voting). The network of Validators sustaining the blockchain takes no part in the registration of voters while the Certifier instance has no role in the validation of ballots. This guarantees that ballots are cast in complete confidentiality while ensuring the principle of "one voter, one ballot" for any given election. It also allows pollsters to establish any custom certification requirements without having to alter the voting system or affecting the validation of ballots in any way.

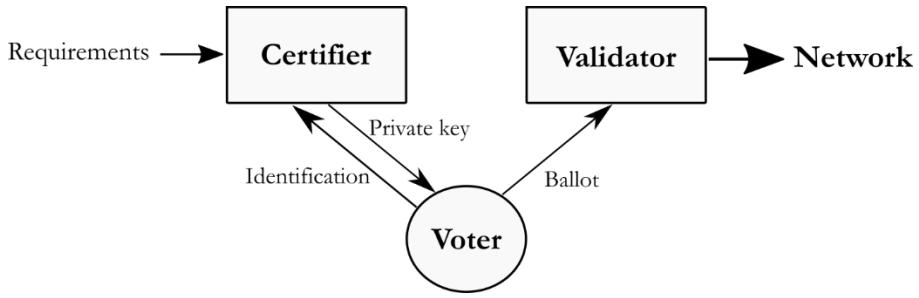

Figure 1 - Overview of voting process (source: author).

In the Bitagora Platform implementation, the Certifier instance is set up as an opaque, automatic and self-managed script running on a serverless AWS lambda. This script is initiated by a network administrator and then acts independently of any human intervention, creating a random poll private key and a random poll secret:

- The poll private key is used to create the poll and to close it at the end of the voting. A human administrator initiates both processes but has no access to the poll private key, which is only published by the Certifier instance when the poll is closed.
- The poll secret is used as a salt in the hashing function when processing the identification information of voters and is destroyed by the Certifier instance when the poll is closed.

The Bitagora Platform implementation ensures that no human can possibly know the poll key or the poll secret, thus being prevented from altering the registration of voters during the duration of the poll. While other implementations of the voter registration process are possible, for example with a Certifier instance hosted on a server, they cannot ensure with the same degree of confidence the secrecy of the poll key and secret.

The requirements for the registration of voters can be customized for each poll. They are included in the Certifier script before launching it and serve to ensure that all voters are presenting a valid form of identification (e.g., a national id or passport number). These requirements can be made as loose or strict as needed, but cannot be modified at any point during the duration of the poll.

When voters submit their identification to the Certifier, the script checks the validity of the information provided according to the set requirements. If the identification is valid, the Certifier produces a 32 bytes voter private key that is deterministically derived from the voter identification through a hashing function that includes the poll secret, making it impossible to recover the identification information of the voter from the private key. The Certifier also produces a token that includes the public key derived from the voter private key using the elliptic curve secp256k1 protocol, a cryptographic cypher that allows the disclosure of the public key without revealing the private key. The Certifier signs this token with the poll private key and sends the signature and the voter private key back to the voter. This completes the registration and the Certifier's participation in the process.

After obtaining a voter private key, the voter can cast the ballot, choosing one of the available options defined in the poll. This ballot is signed by the voter with the voter private key and includes the certifier's signature, as well as the date, the public key derived from the voter private key and the identifier of the poll. The signed ballot is sent to a node in the network for processing. Each node in the network checks that the ballot is valid using the same transaction processors. A valid ballot needs to correspond to an active poll, be cast within the set dates for the poll, choose one of the available options, include a valid certifier signature, and be signed by the owner of the voter private key that corresponds to the voter public key that identifies the ballot.

Validators also check that the ballot identifier (the voter public key) does not match a ballot already included in the public ledger or blockchain. Since this identifier is deterministically derived from the identification provided by the voter at the moment of registration, it is impossible for a voter to cast a ballot more than once in the same poll using the same identification. If the registration requirements set in the Certifier instance are sufficiently rigorous the system will effectively prevent the same person from voting twice in the same poll. One of the advantages of the Bitagora Platform is that this is achieved without storing or revealing the person's identity at any time.

Network components

The Bitagora network is built on the Hyperledger Sawtooth open source blockchain platform developed by Intel Corporation [https://sawtooth.hyperledger.org/docs/core]. Every validator node in the network relies on the same Transaction Processors, ensuring the coherence and reliability of the whole system across multiple and independent nodes. The main Transaction Processors are **polls-tp**, which handles transactions that create and close polls, and **ballots-tp**, which handles the submission of ballots to existing polls. Besides these two purpose-built Transactions Processors, each node operates a **settings-tp** to handle the network settings and a **poet-validator-register-tp** to handle registration of new nodes using the PoET protocol.

Each validator node also includes a **shell** that allows the user to monitor the node from the local console and a **REST-API** that communicates with external clients through http port 8008 in order to receive new transactions and serve network information when requested. All of these components are installed in Docker containers and connected with the validator using the tcp protocol on port 4004.

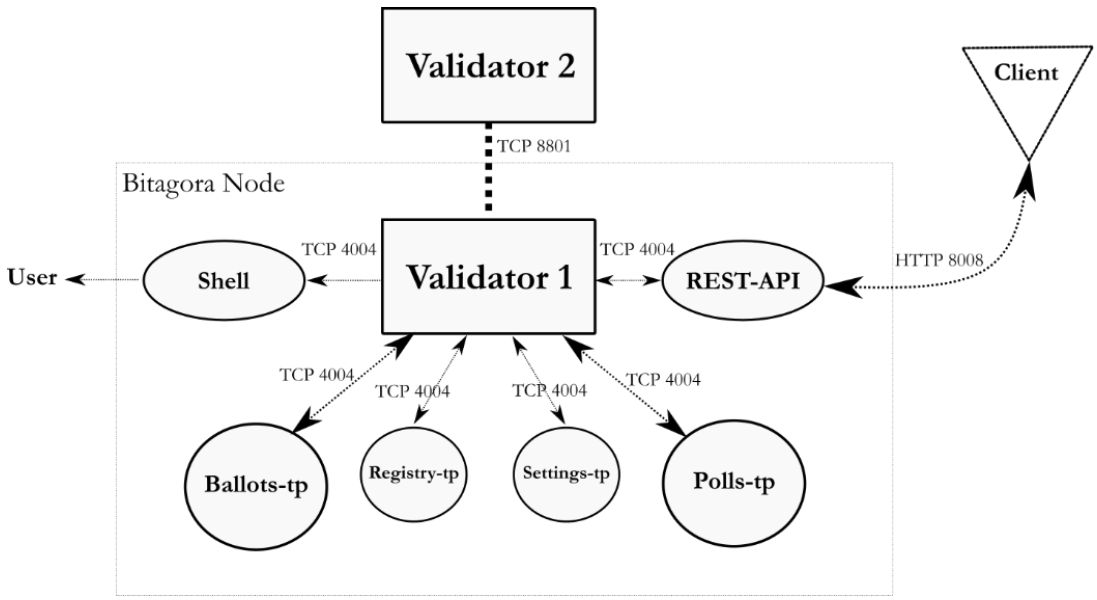

Figure 2 – Bitagora network components (source: author).

The validators on the same network are connected to each other through the Hyperledger Sawtooth network layer, which uses a gossip or epidemic 0MQ protocol in order to establish initial connectivity, peer discovery, and message handling [https://sawtooth.hyperledger.org/docs/core]. Nodes access the network through external port 8801 and establish active bidirectional peering relationships with available nodes at the moment of installation. These relationships will vary as nodes leave or join the network.

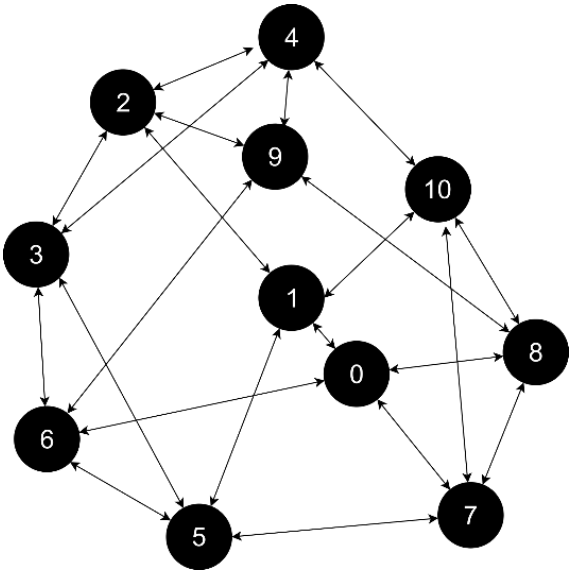

Figure 3 - Bidirectional peering between nodes (source: Hyperledger Sawtooth docs).

The nodes on the Bitagora Platform use a simplified version of Proof of Elapsed Time (PoET) as the consensus mechanism that ensures that the data stored in the ledger and recorded in state is consistent across all nodes in the network. This mechanism offers a solution to the Byzantine Generals Problem in a fully distributed application setting. The complete PoET protocol uses Intel SGX processors to establish a trusted execution environment. However, this imposes a limitation on the number of machines where nodes can be installed and raises some concerns about the privacy of nodes. Moreover, PoET SGX is a proprietary solution requiring a license and registration with a private corporation, while the philosophy underlying Bitagora is to provide an open source solution that can be implemented without any proprietary bindings.

For these reasons, Bitagora implements the PoET consensus without relying on the trusted execution environment guaranteed by SGX. This version of PoET has been tested by Hyperledger Sawwtooth (PoET-SIM consensus) and provides similar levels of fairness, consistency and transparency as the PoET-SGX consensus. However, this implementation forgoes to some extent the Byzantine fault tolerance available in the complete PoET version. In a fairly straightforward and simple network as the one deployed in the Bitagora Platform, however, this reduction in Byzantine fault tolerance seems less critical for the network's security than relying on a limited number of validators.

Unlike Proof-of-Work and other consensus mechanisms common in distributed ledgers, PoET relies on a lottery algorithm and a trusted function that allocates the ability to create blocks amongst connected validators based on randomness and the time elapsed since they were allowed to publish a block. This mechanism ensures at a negligible cost that all validators will claim approximately the same number of blocks and provides protection against validators that try to overtake the network by publishing an excessive amount of blocks.

Client components

There are four basic types of users of the Bitagora Platform: voters, monitors, pollsters and administrators. Each one will interact with the Platform using different Clients or client-side components.

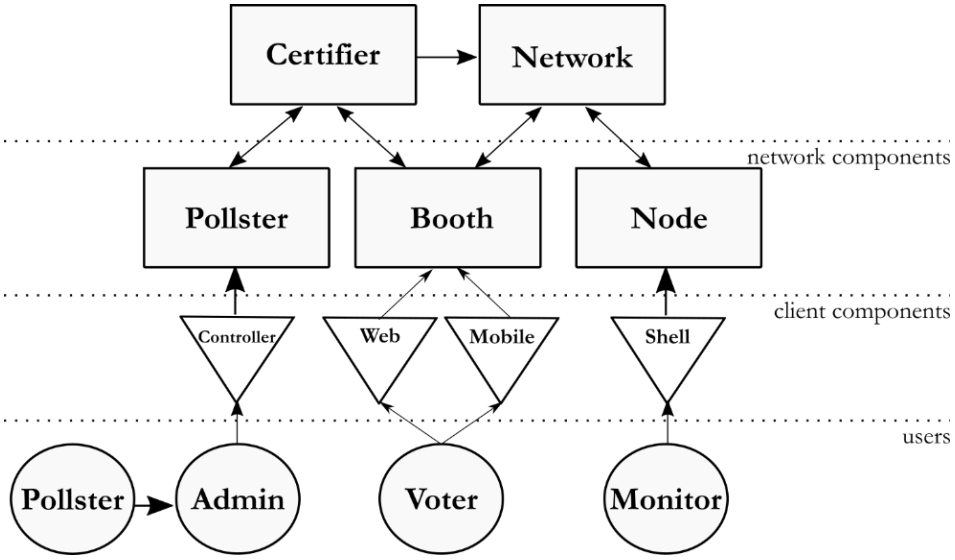

Figure 4 - Client components and connections with network layers (source: author).

- **Voters** are individuals registering and casting a vote in any active poll. They interact with the platform through mobile or web applications running Bitagora Booth, a set of client-side functions and utilities that allow voters to cast and check their ballots with the private key provided by the poll Certifier.
- **Monitors** are individuals or institutions operating a validator node in their devices. To become a monitor they only need to download a bash script that installs the Bitagora Node software on their machine. Monitors can access the information processed by their node through the shell component, either directly or through the installation script. Since nodes are interconnected, monitors have access to live information of the network and serve this same information through the REST-API of their nodes to external clients.
- **Pollsters** are individuals or institutions who organize and launch a poll using the Bitagora Platform. They might also be administrators or monitors (and, of course, voters). Pollsters need to be administrators to interact directly with the Bitagora Platform. Only administrators can create or close polls in the platform. If pollsters are not administrators of the network, they need to contact an administrator and get the administrator's approval before the poll can be launched. This protects the system from being flooded by rogue polls.
- **Administrators** are individuals or institutions who hold an administrator key which allows them to make proposals and vote on changes to the network settings. They are also able to create and launch new polls, either through a Certifier instance or directly by using a custom-built Controller to access the functions and utilities of the Bitagora Pollster component.

Distribution

The Bitagora Platform core code is distributed as an open source modular application freely available on the Bitagora github repo: https://github.com/bitagora/bitagora-core. The repo includes the following components:

- **Bitagora Node**: Includes a script to install a validator node, the configuration files for the Bitagora Platform network and the source code of the Docker images that are installed when deploying a node in this network, in particular for the shell, ballots-tp and polls-tp. The other transaction processors (settings-tp and poet-validator-registry-tp) are installed directly from the Hyperledger Sawtooth Docker images. The REST-API is installed from a forked version of the Hyperledger Sawtooth REST-API available in the Bitagora repo.
- **Bitagora Booth**: Includes a set of functions and utilities that can be used by web and mobile clients to cast, check and recount ballots for an existing poll. This module can be installed in any mobile or web application as part of a user interface in a fully-working voting system.
- **Bitagora Pollster**: Includes a set of functions and utilities that can be used by administrators to approve, create and close polls in the network. This module can be installed in any mobile or web application as part of a user interface in a fully-working voting system. It can also be implemented as part of a command-line interface or bash script.
- **Bitagora Certifier**: Includes the code of the Certifier instance to be deployed in an AWS lambda to provide serverless registration in a fully-working voting system. This code can also be adapted to create a different registration architecture, for example using a web server as Certifier instance.
- **Bitagora Library**: A set of functions and constants shared by the different components of the Bitagora Platform.

Practical applications

The Bitagora Platform has been designed and developed so that it can be used in any situation where a citizen, a group of citizens, or a citizen's organization wishes to organize a poll or an election without relying on any central authority for the monitoring and validation of ballots. This is achieved by

- separating the registration of voters (Certifier) from the handling of ballots (network of Validators);
- incorporating custom rules for the registration of voters, which can be adapted for any use case;
- providing a peer-to-peer network that can handle an unlimited number of polls at the same time;
- incorporating in the design the possibility of initiating open polls (results can be viewed in real time) as well as encrypted polls (results are encrypted with the poll private key and can only be viewed after the poll has been closed);
- providing client-side modules that can be incorporated in different kinds of applications, to be deployed in the web, mobile or both.

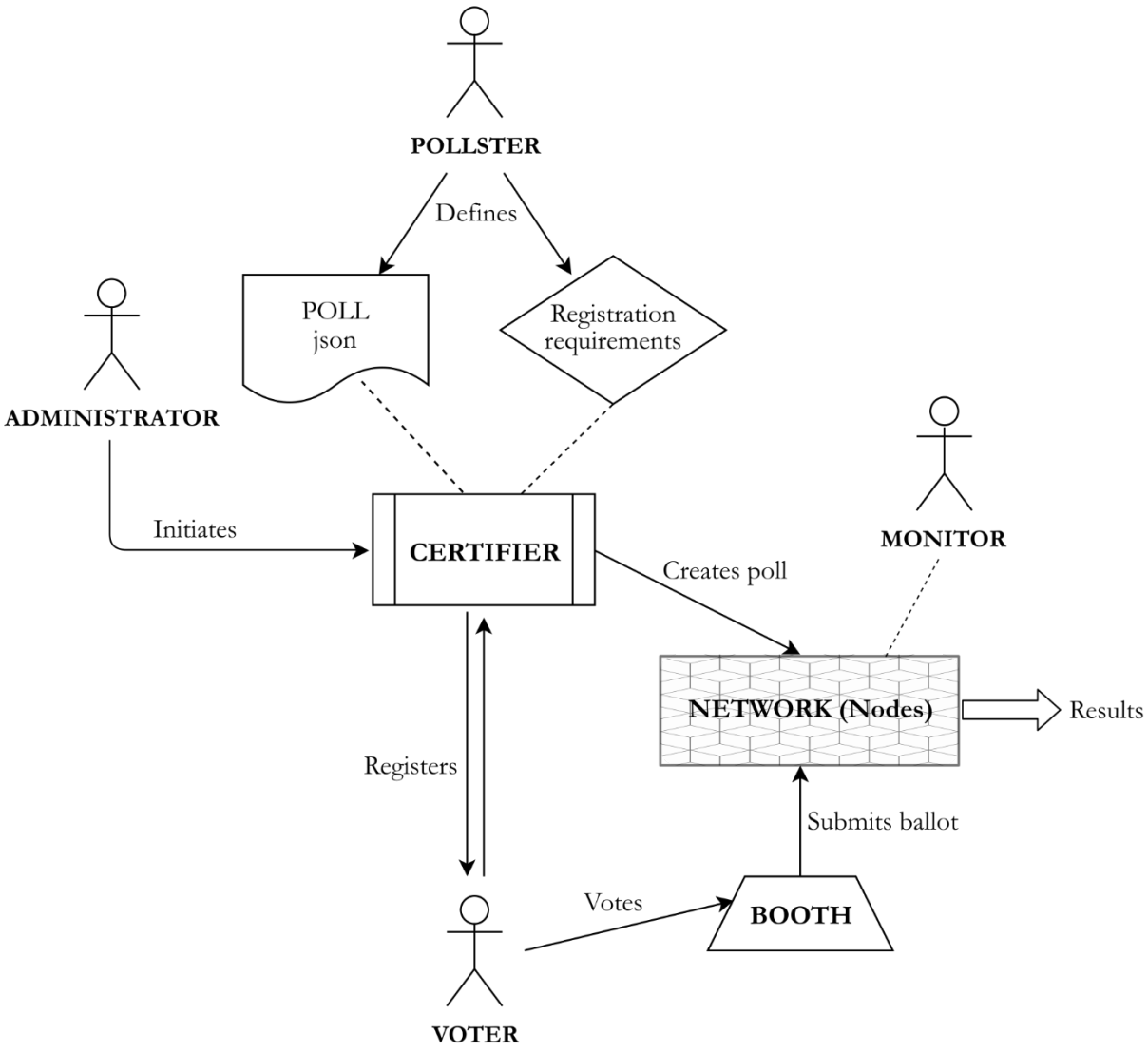

Figure 5 - Workflow of the polling process with the Bitagora Platform (source: author).

In principle the separation of registration and voting allows for the implementation of any system of voter registration, including centralized systems. However, the Bitagora Platform recommends the use of an independent, automatic and opaque Certifier instance, such as the one implemented in the Bitagora Certifier component. This guarantees the protection of confidential poll parameters (poll private key and poll secret) and can easily be audited by external parties to ensure there has been no interference in the registration system.

Even when users decide to deploy a centralized voter registration mechanism, for example by setting up a server to register voters, the Bitagora voting system ensures that there is no need to keep the personal information provided by voters in any database. Thanks to the cryptographic scheme used to identify ballots, the identity and personal information of voters can be kept at all times inaccessible to humans, including pollsters and network monitors or administrators.

Whether a pollster decides to use the Bitagora Certifier instance or set up a custom registration mechanism, the rules used to validate the identification provided by voters have to be defined in advance for the particular poll

that will be conducted. There is no limitation to the kind of registration requirements that can be implemented, but they should be susceptible of being deployed programmatically.

Pollsters also need to define the poll parameters, which are encoded in a JSON file that includes the question being asked, the possible answers that voters can give, the values given to each answer, the start and end dates of the poll, as well as other information (views, locales, etc.) needed by clients to interpret the poll data and provide appropriate information when requested by users.

Once the poll parameters and voter registration mechanism have been defined, pollsters would need to decide if they want to use the existing Bitagora network. This network is sustained by individuals and organizations who have downloaded and installed the Bitagora Node software and are interconnected through a particular peer-to-peer protocol. Alternatively, pollsters can also choose to set up their own network, to be used for a single poll or for any number of polls. In that case, they would need to ensure that there is enough nodes installed to guarantee the security and reliability of the network.

Any network deployed using the Bitagora (Hyperledger Sawtooth) framework will have a number of administrators. Each administrator has a private key that provides the ability to make proposals and vote on proposals made by other administrators. All the settings in the network are adjusted through a democratic voting system where each administrator has one vote. The Bitagora network will be administered by a number of individuals and organizations committed to the promotion of direct democracy throughout the world. If you are interested in joining the Bitagora network and become an administrator, please get in touch with us.

Whether pollsters use the Bitagora network or any other custom network, polls need to be approved by an administrator of the network. Only administrators have the ability to create polls in the system. First of all, administrators need to sign a poll token (approval). This approval signature is included in the poll before it is submitted to the network. When the poll is set up using a Certifier instance, the administrator will have to initiate the poll by sending a request to the Certifier instance with his administrator's private key. If this key is correct, the Certifier instance will automatically complete the poll data, approve it and submit it to the network for validation. Similarly, the poll can only be closed by the same administrator that initiated it. The administrator needs to send a request to the Certifier instance who then submits a closing transaction to the network.

The Bitagora Platform allows the organization of two types of polls: open and encrypted. This is defined by the pollster before requesting the approval of the poll by an administrator. In open polls, ballots are inscribed in the ledger unencrypted and can be read and recounted by any user through the client modules. This allows for the results to be updated in real time during the duration of the poll. In encrypted polls, ballots are registered in the public ledger using an ECDH encryption scheme. Ballots are encrypted using the public key of the poll and can only be decrypted with the private key, which only the Certifier instance holds. When the poll is closed by the administrator, the Certifier instance submits this private key to the validator network, making it publicly available. At that point, validators will not accept the submission of any more ballots. Ballots that have already been cast can then be decrypted by any client module and results can be published in whatever format.

The Bitagora Platform distribution does not provide specific UI applications. It only includes client components (Bitagora Booth and Bitagora Pollster). These components can be incorporated into any custom-made UI for mobile, web or both. If you are considering launching a poll using the Bitagora network, please get in touch with us to discuss how to adapt for your use case the mobile and web UI applications that have already been developed by Bitagora.

Acknowledgements

Thanks to the engineers and programmers working at Hyperledger Sawtooth and other members of the community for their readiness to answer questions and solve problems regarding the Sawtooth protocol and architecture. Thanks also to Rubèn-Dario Castañé for suggesting the use of an opaque AWS lambda for the Certifier and for his contribution, together with Álex Bueno and other members of the Catalan Pirate Party, to testing the application.

Follow up

Bitagora is a work in progress. Active research is under way and new versions of this paper and the distributed application will appear at https://bitagora.cc.

Being an open-source, non-for-profit project which aims to support direct democracy throughout the world, any contributions from individuals and organizations will be greatly appreciated and can help to extend and further develop the platform.

If you wish to make a donation to support the project, you can do so by sending Bitcoin to 1BiTaGNwLVNRZrweD1bXkvuDr43xAafrAy.

For comments, suggestions or to cooperate with the project, please send us an email to mail@bitagora.cc.

"Our vote is our weapon" – Martin Luther King