# VoteHub
# System Architecture Overview

Version 0.1

Publish Date: Oct 27, 2023

# Table of Contents

# 1.0 Summary

This document lays out the system architecture basics for the Assembly end-to-end verifiable remote accessible voting system (E2EV RAV; referred to simply as the voting system or RAV from here on out). The document will highlight the technical design around individual components and the integration points of those components at an architectural level. This document does not delve into the details of the cryptography involved in securing data but may mention this where applicable. Additionally, this document will not contain user specific instructions or steps involved in executing an election with the voting system except as applicable to descriptions of the architecture.

## 1.1 Purpose

The purpose of this document is to provide detailed descriptions of the implementations of various technologies used within the system such that a reviewer can better understand the design of the product. The document can be used for assessing the readiness of such designs for use in public elections as well as a baseline to compare the built and deployed product against designs for testing and evaluation purposes.

## 1.2 Intended Audience

The intended audience for this document are reviewers evaluating the core architecture of the system with regard to known/published specs or academic descriptions of generic end-to-end verifiable voting systems, system evaluators attempting to assess the readiness/suitability of the product for deployment within a given election jurisdiction, and system auditors interested in a more detailed understanding of the components of the system that created the election data they are auditing for any given election or use of the product.

## 1.3 Scope of Document

Below are descriptions of what can be found within this document and specifically what will not be found but may be found in other supporting documents related to this system.

### 1.3.1 What is covered

Descriptions of all the system components, data elements and interactions/integration points between components can be found in this document. Simplified diagrams illustrating more complex or overarching concepts laid out here are also included. User groups/actors and broad descriptions of their use cases within the system are also included.

### 1.3.2 What isn't covered

Specifics of the protocols, in particular the cryptographic protocol, is not included. Those can be found in the accompanying cryptographic protocol documentation. That document may reference descriptions provided here especially around defining components and actors. Additionally, configurations required for deployment and operation of the system are not included in this document although it is acknowledged that to satisfy many security and operational considerations configuration aspects are critical. As such, an accompanying system configuration document, which contains discussions on potential attack vectors, does address these items. This document exists to describe the system architecture in an assumed clean and attack free deployment, describing mostly the happy path use cases only.

## 1.4 Prerequisite Knowledge/Information Needed

Readers of this document are expected to know the basic definitions/use cases of both election specific terms and some of the technologies provided. A glossary of terms is provided in Appendix A but it is expected that most readers of this document will already be familiar with many if not all of the terms and technologies presented. Where a particular technology is deemed to be not common knowledge references to supporting documentation is provided. Below is a list of technologies that are used within this document.

# 2.0 Basic System Requirements

This section lists a few assumptions/requirements that are inherent to the system.

## 2.1 Functional Requirements

Functional requirements relate to properties of the election system that voters, or users in general (including election officials, candidates, or auditors), can actively choose to perform. These properties are, in some way, measurable. The system has the following functional requirements:

- election types supported:
  - referendum: direct vote on a proposal or issue,
  - candidate election: one vote for a candidate from a predefined list, – multiple choice: a selection of multiple vote options,
  - ranked election: an ordered selection of multiple vote options,
  - write-in vote: a free-form text of a maximum size,
- verification mechanisms for voters to check that the encrypted ballot contains what they expect,
- possibility to confirm selected options after voting by an overview of the complete ballot,
- possibility to correct mistakes before submitting an encrypted ballot,
- ability for overwrite your vote, i.e., a voter can vote multiple times while only the latest submitted vote will be counted in the final tally,
- ability to check the status of your ballot after submission,
- public auditability of the election process throughout the election period.

## 2.2 Non-functional Requirements

Non-functional requirements describe properties of the voting system that impact the user experience.

**Mobility** is the property that enables voters to use any internet-connected device (PC, laptop, tablet, smartphone) to connect to the election system. They do not need to vote from a particular location (e.g., a polling station). Instead, they can participate in the voting process from any place they consider private and with an internet connection.

**Vote & go** entails that voters are only required to be present during the voting phase. Results can be computed without the presence of voters.

**Transparency** implies that election data is available for auditing through a public bulletin board.

**Multiple voting period voting rounds** enable election officials to reuse most of the election configuration for multiple sub-elections where the same set of voters must vote on different ballots. A separate election result is computed for each voting period voting round.

**Ballot Replacement** allows voters to overwrite their votes, enabling them to vote multiple times. However, only the most recent vote submitted by a voter will be taken into account in the final tally.

## 2.3 General Security Requirements

Security requirements describe properties of the election system that contribute to the quality and reliability of an election result. This section briefly describes the properties, while the explanation of how these properties are achieved is presented in other supporting documents.

**Eligibility** property is defined as the fact that only a limited number of predefined voters can cast a valid vote.

**Privacy** property implies that no entity can read a partial result or any votes before the intended time. This is to prevent influencing the subsequent voters throughout the election period. Voters' initial intentions may change if the current results were publicized.

**Anonymity** property implies that no single entity can determine how a particular voter voted.

**Integrity** of voting data is the property that implies detection mechanisms of whether any votes recorded on the bulletin board during the election phase have been modified or deleted.

**End-to-End Verifiability** property describes that all steps of the election protocol are verifiable by following some auditing process.

**Receipt-freeness** property is defined as the fact that voters cannot prove to a third party how they voted after they submitted the encrypted ballot.

**Replay protection** is a security measure that prevents an adversary from capturing and replaying a valid set of actions or data to compromise a system. It ensures that any captured information or actions cannot be reused to gain unauthorized access or perform malicious activities.

# 3.0 System User Types and Descriptions

This section describes the human actors/user groups that have a stake in an election.

## 3.1 Election Officials

Election officials use the election system to configure and run an election. Election officials possess credentials that are used to access different system components to complete different election tasks. Election officials have an interest in the election running correctly, according to the election configuration they define.

## 3.2 Trustee(s)

Trustees are a particular type of election official. In addition to potentially holding an election official role, trustees are responsible for preserving the secrecy of the voting data throughout the election process. Trustees are responsible for some cryptographic key management as well as system configuration steps and post-election ballot processing steps. Trustees do not need to be election officials but in most cases election officials will serve as trustees simply due to the logistics of finding enough interested individuals to participate in the process.

## 3.3 Voters

Voters are the crucial user group in any election. Within this system the voters belong to a predetermined list of approved users who own credentials used to gain authorization to the voting portion of the system. Additionally, voters may choose to perform an important data verification step when they opt to ‘challenge’ the system, described later in this document.

## 3.4 Public Auditor(s)

Any person can be a public auditor of the election process, assuming they have access to the proper auditing tools that perform all the cryptographic operations. Auditors don’t have an active role in the election process, and most of the time, the auditing can occur independently of the rest of the system. Some audit steps can be performed prior to an election becoming live but most auditing of the system occurs after the election is over such that an auditor can verify all of the data and all steps within any given election.

## 3.5 Private Auditor(s)

# 4.0 Overall System Description/Diagrams

This section describes the components of the system and how they interact with one another. A general overview of the system is provided first, followed by detailed component descriptions and finally data flow descriptions as they relate to various stages of each election cycle.

## 4.1 General System Description

The VoteHub product consists of four main categories of components, each with distinct responsibilities. (See *Figure 01: General System Overview Diagram* below). The general groupings are as such:

1. **Air-gapped Enclave components -** The hardware, software and services responsible for the most sensitive operations of the system. These components are kept off of all public networks and exist only on specific hardware devices, operated and stored in secure locations determined by the administering jurisdiction. Specialized personnel (the Trustees from section 3.2 above) are the only authorized users of these components along with an election official who may be in charge of setting up and maintaining this trusted environment.

2. **Administrative components -** The software and services, in most cases cloud deployed, that allow an election official to define, edit, and administer the election. Election settings, voter lists, and other election parameters are defined here. The data created here is used to populate the publicly auditable components of the system described below. The administrative components are maintained and operated by an election official or representative from the administering jurisdiction.

3. **Election Operation components -** The software and services responsible for actually executing the election. These consist of the digital ballot box (the component that validates data on and off of the bulletin board as well as executing some critical election operation steps, to be discussed later), the bulletin board (the digital storage location where all publicly auditable election data is stored including the election configurations, encrypted marked ballots submitted by voters, and other critical election data), and the voter authentication/authorization components (services that validate a voters identity and eligibility to participate in any given election).

4. **Voter facing components -** The software that voters interact with on their devices or a provided device (if deployed in a poll station) that allows them to conduct operations such as proving their identity, marking their authorized and digitally delivered ballot, submitting their vote, and challenging the system (otherwise known as verifying their vote before final casting of the marked ballot.) These consist of two main components, the voting application and the verification application. While the voting application is made in conjunction with the rest of this system the verifying application is unique in that anyone with prerequisite knowledge and skill can create and distribute a valid verifier application. This fact

enhances the transparency of the system and increases the trustworthiness of the system especially in the eyes of voters.

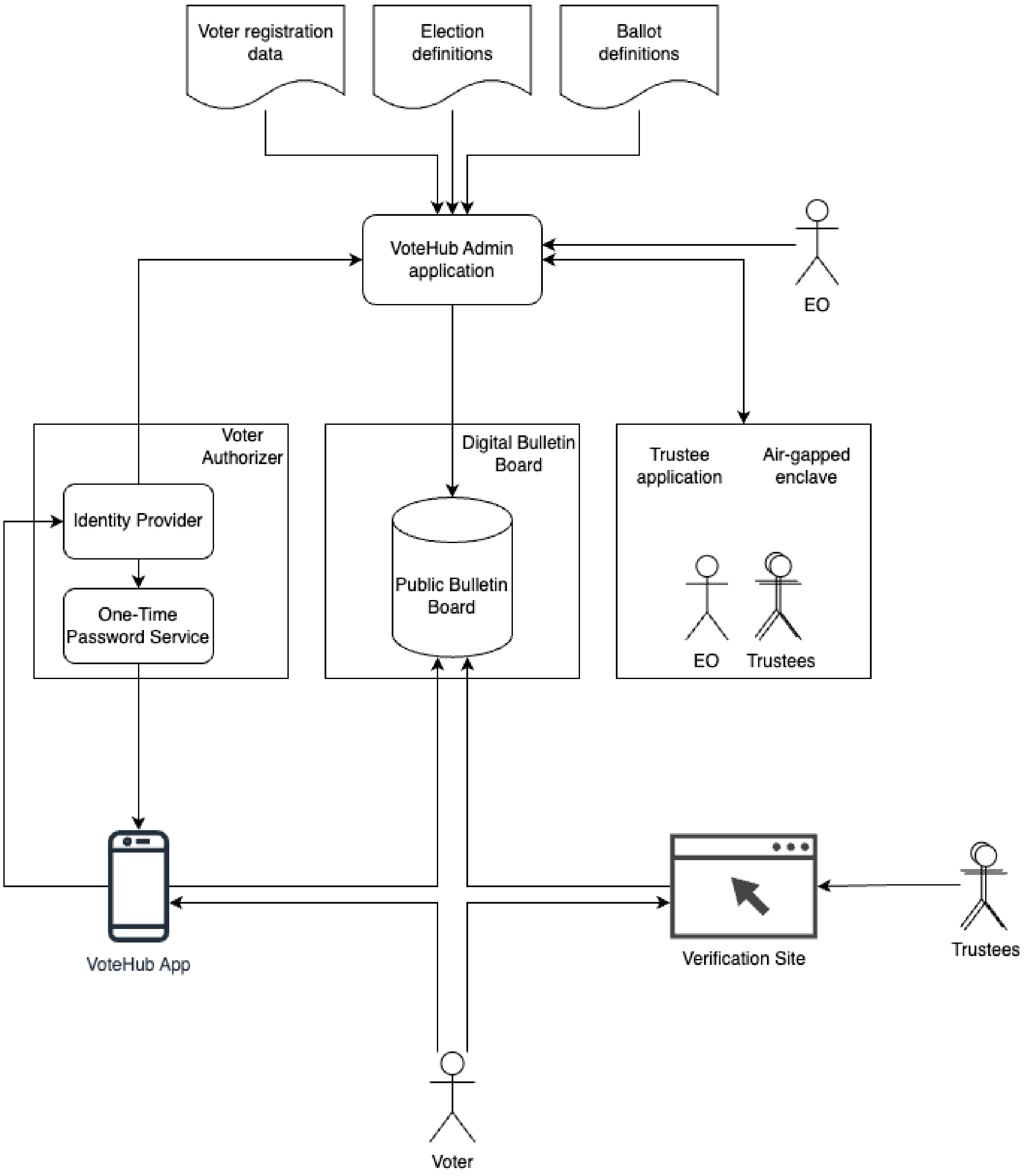
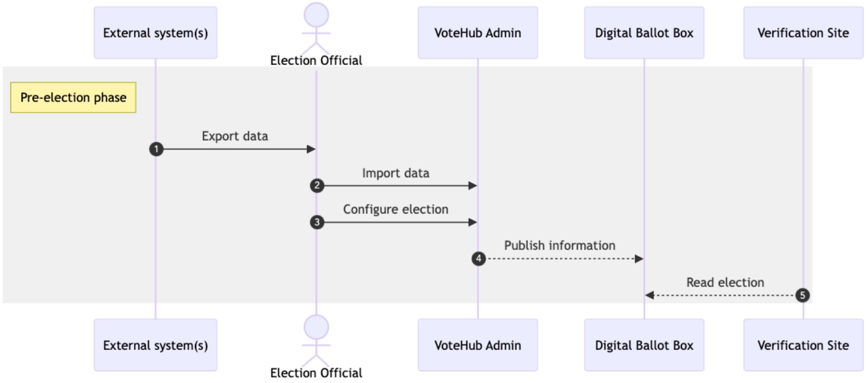
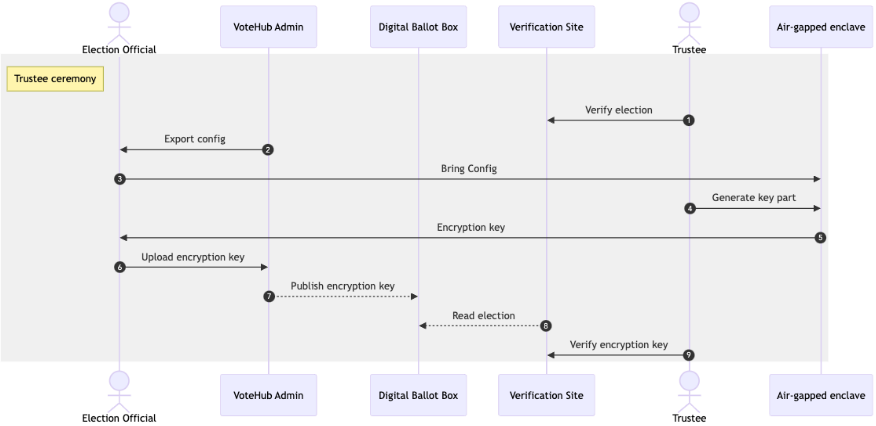


Figure 01: General System Overview Diagram

As shown in the diagram activities can be broken down into three general categories. While somewhat distinct in timing, some of the actions within each category can occur in multiple categories. For example, while election definitions, voter lists, and ballot data creation/definition inputting is considered a pre-election activity, those items can be edited and updated during live voting. Additionally, while encrypted ballot extraction and ballot duplication can be categorized as a post-voting activity, it can be done in certain circumstances while live-voting is still underway. Conceptually the election timing categories are as follows:

1. **Pre-election period** - Activities related to configuring an election including but not limited to setting election open and close times, defining other election parameters such as name and administering jurisdiction, inputting ballot definitions and voters lists, configuring the vote authorizer process, creating the threshold encryption keys (more on this in Section 4.3.2) etc…





2. **Live-voting period** - Activities related to the distribution of ballots to voters, the marking of ballots by voters, the submission of votes by voters, the verification/challenging of submission by voters. As mentioned, some election updates can be done during this period of time around election definition items but critical items such as the threshold encryption key creation cannot be updated once live-voting has begun.

3. **Post-voting period** - Activities such as encrypted marked ballot retrieval, voter affidavit verification, ballot mixing and decryption (more on this in Section 4.3.4), and ballot duplication. Again, some of these activities may occur while live voting is still underway and will only impact already collected marked ballots. Once all live voting is completed, all post-voting activities can be completed for all ballots in full if not done earlier.

## 4.2 Detailed System Component Descriptions

This section describes in greater detail all parties that are involved in the voting system. Each party represents a computer or an application that follows a particular use case or expected action. These parties can be categorized into the following nine types:

1. **VoteHub Admin Service -**
   One administrator service exists that is responsible for setting up and configuring the election. All authorized election officials use this to define an election, configure the rest of the system and input updates as needed. It is an online

application hosted by the administering jurisdiction or representative. Credentialed access controls are required to use the application.

The election administrator application owns a key pair used for signing the election configuration data, and is responsible for privately storing its private key.

2. **Trustee Application -**
   Each election contains a set of trustees, individuals entrusted with highly secure and in some instances offline, air-gapped operations. Trustees can change between elections or can remain the same but will be given access on a per-election basis. Each trustee uses the trustee application to perform all cryptographic processes. The operations run on the trustee application involve but are not limited to the threshold vote encryption key generation, the mixing and decrypting of extracted marked ballots, and the production of the final election outputs. The VoteHub initial product offering will include ballot duplication files as the final output, but in future iterations this could be final tally results instead or in addition to ballot duplication files. Trustees are responsible for preserving the privacy and the fairness of the election by working together to build the threshold election encryption key while securely storing their shares of the decryption key.

   The trustee application will compute and deliver to its trustee, a key pair and a share of the election decryption key. The trustee is responsible for privately storing the keys until a result is computed. Trustees are responsible for destroying the keys after the election event and subsequent data retention period has ended.

3. **Voting Application -**
   There is a list of predefined eligible voters provided by the election jurisdiction that later statements will simply refer to as ‘voters’. The voting application is the software that voters use to perform ballot retrieval, ballot marking and all the cryptographic operations involved in the voter’s part of the voting process. The voting application runs locally, on the voter’s device, and is a native mobile application. For certain operations of the voting application an internet connection is required as well as access to the phone’s camera and file storage.

   During the voting process, the voting application generates a key pair that represents the cryptographic identity of that voter. Apart from the private key, the voting application learns all secrets that its voter inputs such as the voter’s credentials and the plain-text vote. It does not send any telemetry about the voter’s actions outside of the required ones to complete the voter flow.

4. **Identity Provider -**
   There is a set of Identity Providers where an Identity Provider is a third-party application responsible for authenticating a voter during the election phase.

Identity Providers must follow the OIDC protocol. Identity Providers will take inputs from the voting application to confirm the identity of the voter and work in conjunction with a one-time-password service to complete the process of authenticating a voter on a specific device and instance of the voting application.

5. **Voter Authorizer -**
   The Voter Authorizer is a service responsible for authorizing a Voter after being authenticated. The voter authentication can be performed by authenticating with all Identity Providers (described in Section 5.3).

   The Voter Authorizer is responsible for preserving the authorized voting list for each given election. This data is derived originally from the voter registration data inputs provided during the pre-election period but can be updated during live voting to add or remove voters as needed. The Voter Authorizer owns a key pair used for signing voter authorization and it is responsible for privately storing its private key.

6. **Digital Ballot Box -**
   The Digital Ballot Box is the central component to elections operation, such that all other parties push/pull data to/from it. It is a single service, publicly accessible via the internet. The DBB is responsible for validating that data being submitted or requests for data being made by other components are legitimate and coming from trusted sources.

   Additionally, the Digital Ballot Box contains a bulletin board which stores all the public information about an election. This bulletin board has both publicly available data storage and a hidden track where particularly sensitive information is kept. The data which is published on the bulletin board is thoroughly described in Section 5.2.2, but it can be summarized into the following categories:

   - *configuration data*; This is set up during the pre-election phase (described in Section 4.3.2), mostly generated by the Election Administrator application.
   - *voting data*; This is populated during the election phase (described in Section 4.3.3) and is generated by the voting application in collaboration with the Digital Ballot Box.
   - *result data*; This is collected during the post-election phase (described in Section 4.3.4) and includes mixing and decryption files generated by Trustees and which enables the election outcome to be verifiable.

   The Digital Ballot Box owns a key pair used for signing data on the bulletin board, and it is responsible for privately storing its private key.

7. **External/Independent Verifier -**

The External Verifier is an auditing tool that voters use if they choose to perform the process of challenging a vote cryptogram (Section 4.3.3.5). The External Verifier allows voters to decrypt their submitted vote before casting as a way to challenge the system to encrypt properly. Because the system does not know how many times the voter may challenge the system in this way, it must behave honestly every time. This step is critical in completing the 'proofs' necessary in an end-to-end verifiable voting process (described in Section 4.3.1) During this process, the External Verifier will generate a new key pair and it has to protect its private key.

External Verifiers, as the name suggests, can be created from available information by any entity involved in the election process including all external entities solely concerned with election integrity. A voter may choose a valid verifier from any source they trust and do not have to rely solely on the election administrators for this tool.

8. **Signature Verification Service -**
   As part of the process of voting digitally, a voter provides a digitalized affidavit through the voting app to the voting system. This affidavit is then verified by special election officials against the voter registration data to confirm the voter's eligibility to vote. While the process may take some back and forth communication between the voter and the election official, it will ultimately indicate if the ballot is eligible for extraction or not.

9. **Auditing Tools -**
   Two auditing tools are used by different actors to perform the auditing process. The election officials use the administrative auditing tool to complete all the cryptographic operations involved in the administrative auditing process. This confirms to the election officials and election monitors that the election system behaved according to their configuration.

   A public auditor can use a public auditing tool to perform the public audit process. This verifies the integrity of all public data from the bulletin board. More details on auditing in general can be found in the auditing section at the end of this document.

## 4.3 E2EVIV description/method demonstration running an election

### 4.3.1 What does it mean to be E2EVIV?

While the definition of E2EVIV is varied and changing as the technology evolves, some common 'proofs' or concepts hold true. These are as follows:

1. **Marked As Intended** - A voter can verify that the digital ballot has collected the data consistent with their intent. Example, if a voter chooses candidate A within contest N the displayed ballot will represent the intended choice as will the collected data used to make the ballot display.
2. **Cast as Marked/Stored as Cast** - The digital ballot data that is stored within the digital ballot box represents the same data the Voter cast. If a voter can verify that the cast data has been transferred to the digital ballot box unprejudiced this proof is deemed completed/valid.
3. **Counted/Tallied as Cast/Stored** - The aggregated vote data from all stored digital ballots can be proved to represent the accurate count of votes cast for any given contest within an election. Proofs exist that confirm the data a voter cast was accurately and completely counted or tallied with all other votes cast. This proof is a mathematical proof rather than a side by side comparison of ballots in order to maintain the anonymity of any given voter's ballot while at the same time satisfying the requirement to show no unintended votes have been injected or removed from the final tally.

### 4.3.2 Pre-election phase description/steps

During the pre-election phase, human election officials use the Election Administrator service to configure and set up a new election. This consists of the following steps:

- initiate a new bulletin board as described in section 4.3.2.1
- define the election level configuration, including election title, contest titles, candidates, and other services required in the election, as described in section 4.3.2.2,
- define eligible voters and configure the voter authorization mode as described in section 4.3.2.3,
- facilitate the threshold ceremony as described in section 4.3.2.5,
- configure the election phase by setting up voting periods as described in Section 4.3.2.6.

#### 4.3.2.1 Digital Ballot Box initialization

An election official creates the configurations necessary to initialize the digital ballot box including parameters required for the generation of cryptographic key pairs. As part of the ballot box initialization a new bulletin board is created using the election metadata defined by the election official. Upon successful creation a genesis item is placed on the bulletin board and all the corresponding keys are placed in their respective locations based on expected use. More details on this process can be found in the cryptographic protocol paper.

#### 4.3.2.2 Election Configuration

Once a bulletin board is created, an election administrator can publish election definition data created within the administration application to the bulletin board. These definition items are written to the board one by one based on rules defined in the protocol document. All data is signed by the election administrator signing key and reference the address of any previous configuration item as a parent. In such a way, the bulletin board strings together the election configuration events such that they may only come from an approved administrator service with knowledge of previous configuration items. The initial configuration items contain:

- Election definition data
- Ballot data including contest and candidate content and marking/tabulation rules
- Actor configuration items for all other services that will interact with the digital ballot box such as the voter authorizer service.

For each potential actor reference directly above, key pairs are generated with the public verification key being shared with all relevant other components. These public keys are published to the bulletin board to allow data transfer and acceptance between the bulletin board and any authorized service. For more details on this process see the cryptographic protocol document.

#### 4.3.2.3 Voter authorization configuration

An election official interacts with the Voter Authorizer service to configure the voter authentication mode.

The election official selects a list of third-party identity providers used for authenticating voters during the election phase. One of these ‘third party providers’ can be an internally provided service acting as the identity provider or an integration with outside identity providers assuming they implement using the properly defined data structures and protocols. The election official then provides the list of eligible voters, each defined by a unique identifier and a list of identities supported by all of the identity providers.

Finally, the voter authorizer writes the voter authorization configuration item on the bulletin board. This item is signed by the voter authorizer signing key containing the voter authentication mode and the list of identity providers, each defined by their public key. More details on this process can be found in the cryptographic protocol document.

#### 4.3.2.4 Threshold ceremony

An election official interacts with the election administrator service to define the lists of trustees. The election administrator coordinates the threshold ceremony during which all trustees participate to generate the election encryption key and each trustee’s share of the decryption key. The election official sets the threshold value of how many trustees should be present to perform post-election ballot decryption.

At the end of the ceremony, the election administrator writes the threshold configuration item on the bulletin board. This item contains:

- election encryption key,
- threshold setup parameters,
- other cryptographic parameters used for auditing/verification

Trustees are then entrusted to store their portion of the private decryption key securely until post-voting activities require these to be aggregated for the purposes of mixing the votes and decrypting the ballots. Trustees, therefore, are critical actors in both the pre- and post election phases.

#### 4.3.2.5 Voting periods

An election official interacts with the election administrator service to define all voting periods, namely by setting a start and end date. The election administrator writes a voting period item on the bulletin board and the corresponding enabled contests.

### 4.3.3 Active Voting phase description/steps

The active voting phase lasts from the start date until the end date of a voting period. During this time, any voter assigned to that voting period can cast a valid digital ballot by performing the following steps:

- obtain a list with all configuration items off the bulletin board from the digital ballot box,
- authenticate and become authorized to cast a digital ballot on the bulletin board
- select and encode vote choices
- encrypt the ballot following the process
- optionally, perform an challenge/verification on the encrypted ballot and
- Finally, cast the encrypted ballot and obtain a vote confirmation receipt

#### 4.3.3.1 Voter authorization procedure

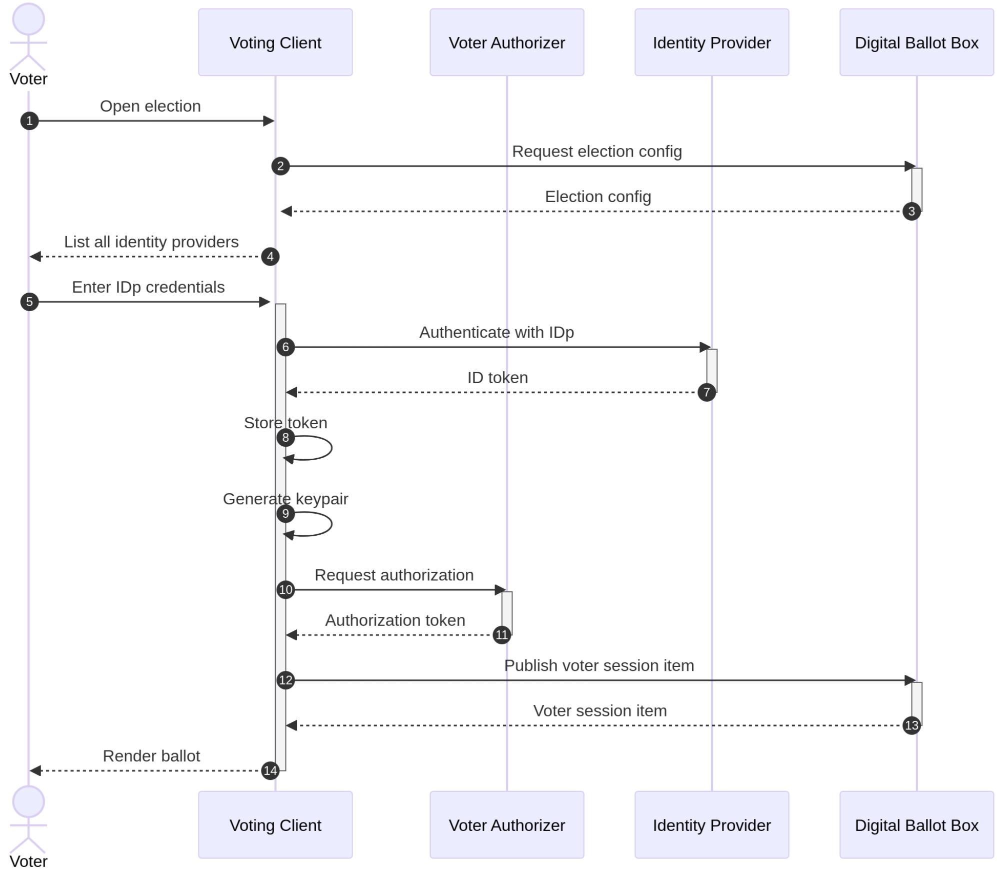
A voter is considered authorized to cast a digital ballot when he/she owns a secret signing key which corresponds to an eligible signature verification key from the bulletin board.

**When using identity-based voter authentication mode**
Each voter must follow the protocol described in the protocol document to obtain authorization to cast a digital ballot on the bulletin board. Specifically, the voter must authenticate and receive identity tokens from all of the identity providers which the voter authorizer configured in the pre-election phase. This can be from a single provider service or multiple depending on the pre-election configuration.

The voting application then generates a key pair and forwards all identity tokens plus the public key to the voter authorizer service proving the identity of the voter.

Once the voter authorizer service can validate all identity tokens the voter is eligible and the voter is then permitted to interact with the digital ballot box. Once authorized the voting application will write an item to the bulletin board indicating the start of a new voting session according to the rules specified in the protocol document.

The voter authorizer service stores a link between the voter identity and all related identity tokens for the administrative auditing process in the post-voting phase. This link is stored privately by the voter authorizer service since the identity tokens likely contain personal information that must not be disclosed on the public bulletin board.



### 4.3.3.2 Ballot Retrieval and Ballot Marking on the Voting Application

Once a voter has been authorized as described above the voting application must retrieve the approved pre-defined ballot as laid out during the pre-election phase for that voter. The voting application, having obtained the election configuration information and ballot data from the bulletin board, constructs the proper ballot style (specific set of contests and candidates for which an individual voter is allowed to vote on) for display to the voter within the voting application.

After the ballot is successfully constructed with all associated artifacts such as supported alternate languages the voter can begin making selections on individual contests according to the marking rules for each contest defined during the pre-election phase. The voter has the ability to implement various accessibility features such as screen contrast changes, text size adjustment, and the enabling or disabling of an audio ballot to accompany the displayed ballot using the features available within the voting application on their device. A voter will proceed through each contest making choices as desired before a final review page of those choices. Although retrieving the ballot does require an internet connection, once the ballot is successfully displayed within the voting application, the marking process does not require connection. It will require the internet again for submission, however.

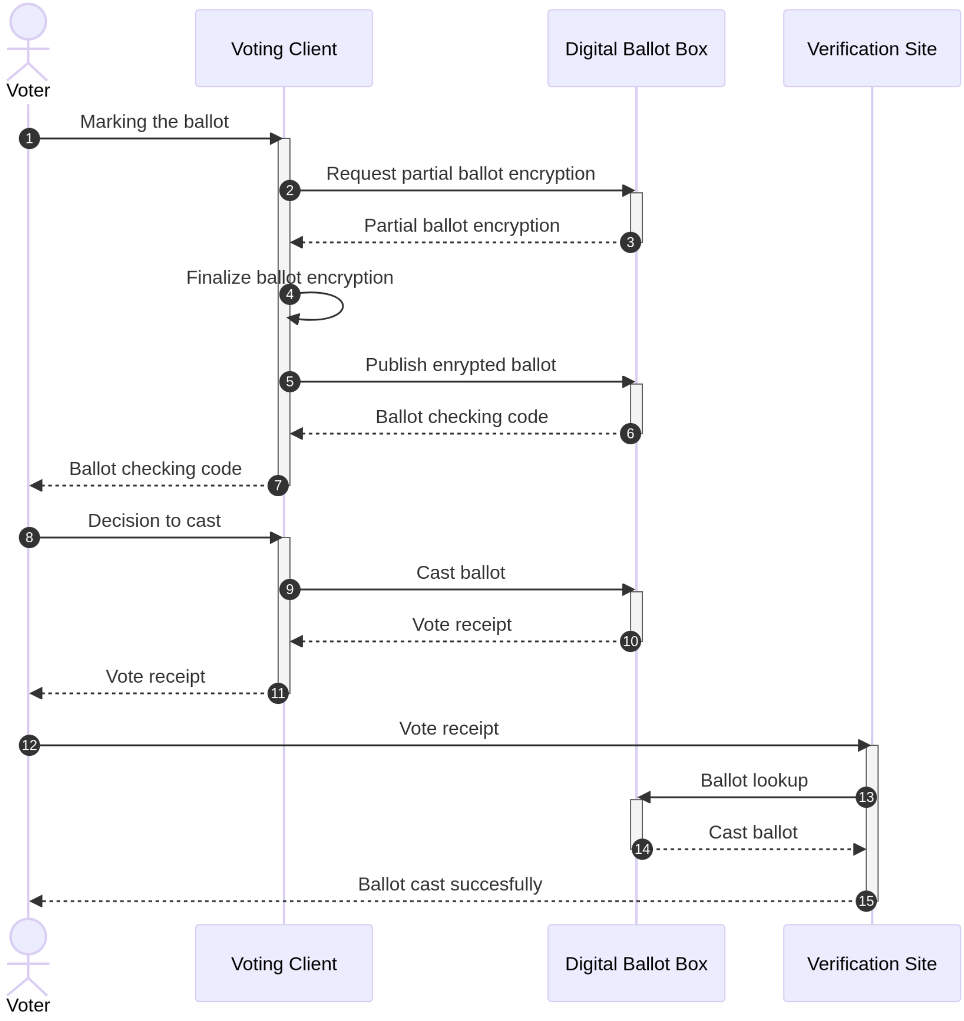
Once the voter is satisfied with the marking progress within the voting application they will ‘submit’ their ballot. Details on the specifics of the voting application use can be found in the accompanying Voting Application Use document.

The following sections are the steps necessary for completing the voting process and turning the voter’s selections into a valid vote cryptogram for storage within the system.

#### 4.3.3.3 Vote cryptogram generation process

During the vote cryptogram generation process, the voting application collaborates with the digital ballot box for generating cryptograms that represent the encryption of each vote. Additional details can be found in the cryptographic protocol document.

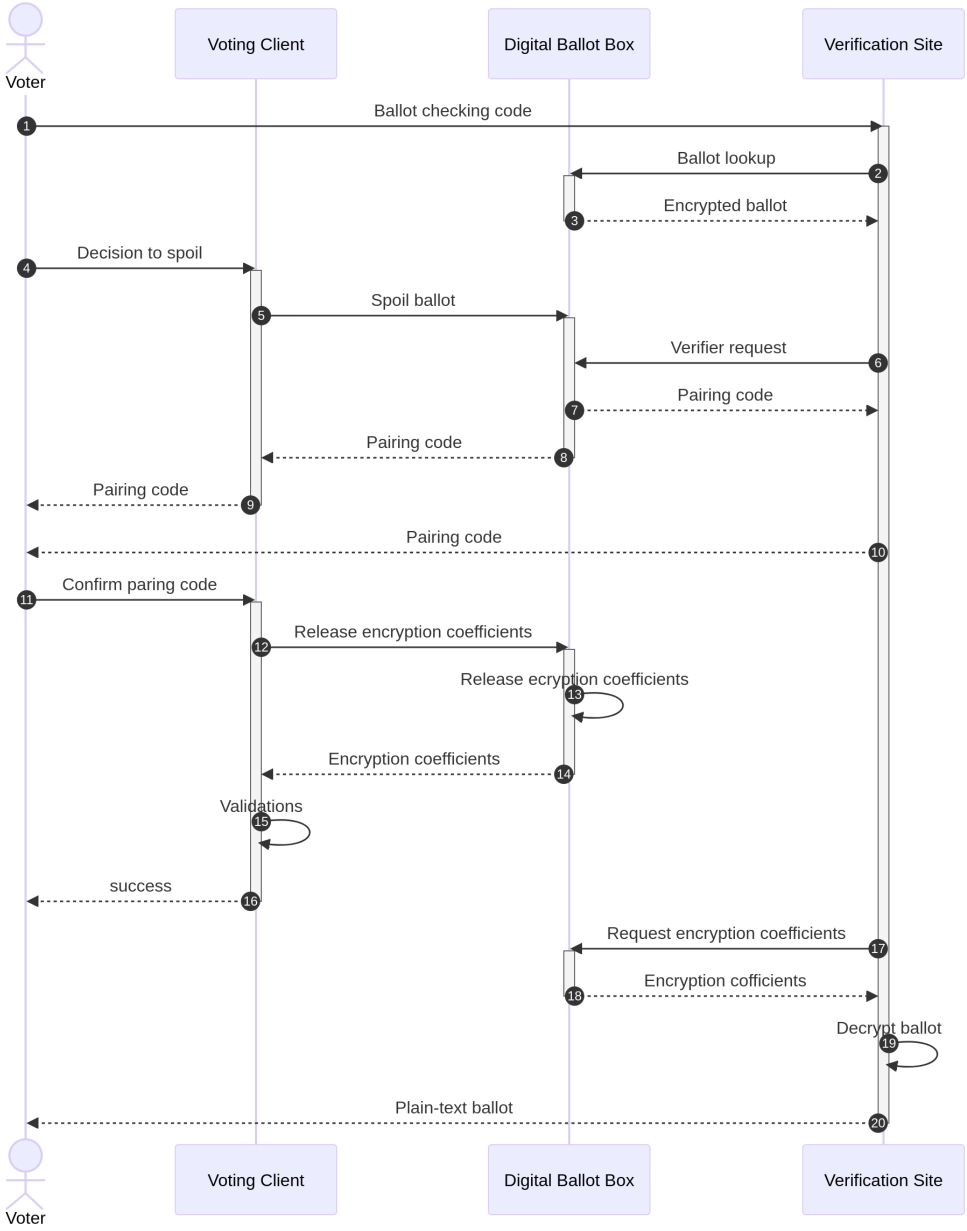
After publishing the ballot cryptograms item on the bulletin board, the digital ballot box immediately self-writes a verification track start item on the hidden track of the bulletin board. The hidden track exists so that later the external verifier can complete its work of decrypting the vote for the voter to verify/challenge the system without compromising the privacy of the voter’s selections. More details on this process can be found in the cryptographic protocol document.



#### 4.3.3.4 Challenging a vote cryptogram

After encrypting a ballot, the voter can choose whether to test/challenge or cast it. To perform the challenge process of an encrypted ballot, the voter needs to interact with the external verifier that will perform all the testing operations, according to the data published on the bulletin board. At the end of the challenge process, the voter will be presented with the vote choices encoded in the encrypted ballot. The encrypted ballot being challenged gets spoiled/discarded by the nature of the process. Therefore, the voter needs to redo the vote cryptogram generation process from section above to get a new encrypted ballot after they are satisfied with the previous challenge. The voter can repeat this process as many times as they want until they feel the system is behaving honestly with regard to encrypting their selections properly. The protocol is inspired by [Simple Verifiable Elections" by Josh Benaloh, 2006].

The external verifier (ideally running on a separate device from the voter’s voting device) retrieves the encrypted ballot, decrypts it and displays the values/selections to the voter. If these match what the voter expects they return to their voting application device and chooses to either change marks and resubmit or just resubmit. In this way the voter's selection choices remain private because the verifier does not know if the voter changed selections or not on the voting application between submissions. Once a voter is fully satisfied with the challenge process they choose to cast their submitted encrypted ballot. After this point no more challenges can be submitted. The detailed description of the entire process can be found in the cryptographic protocol document.



#### 4.3.3.5 Vote confirmation receipt

Upon casting of the ballot the voter receives a receipt from the digital ballot box confirming that the ballot was registered as cast on the bulletin board. The request for casting is logged as an item on the bulletin board. This receipt can be used by the voting to return to the bulletin board at any time to verify the cast ballot is still stored properly and valid. If the receipt and the bulletin board do not agree at any point with regard to the state or location of the cast ballot this is an indicator of a loss of integrity on either end and can be evidence of a tampered election. Voters are provided with instructions on how to report such suspected incidents if evidence of tampering is suspected.

### 4.3.4 Post-voting description/steps

After the active voting phase has finished, or during active voting but acting upon completed votes, the election proceeds to the last step, which will extract encrypted votes for either ballot duplication or final result tallying. If the final election period is over the ballot box will not accept any more valid votes but up until this time encrypted ballots can be extracted given certain conditions exist to maintain voter anonymity. This includes but is not limited to only extracting ballots when 2 or more of the same ballot style are available for extraction. The parameters for extraction, however, can be set by the Election Administrator.

At any time regardless of election state the bulletin board remains publicly available for voters to check that their encrypted ballot is included (using their confirmation receipt described above) and for auditors to check that the list item addresses are consistent (the integrity of the board has been maintained).

The process of computing a result consists of the following:

- the election administrator requests an extraction event to be completed
- the digital ballot box identifies all the valid ballots to be included in the extraction
- a subset of all trustees collaborate in the mixing process to anonymize the encrypted ballots. The number of trustees required to complete this action is determined by the setting applied during the pre-election phase as described previously.
- the same subset of trustees collaborate in the decryption process of the anonymized votes,
- Finally, the election administrator either creates ballot artifacts for ballot duplication to occur outside this system and into the system of record or publishes the tally results of this process.

This process can occur multiple times throughout the election cycle and all previously extracted ballots will be excluded from future extractions to ensure the proper number of ballots are duplicated each round. At least one final round of ballot extraction must occur after the last voting period has ended to make sure all remaining ballots have been successfully duplicated or counted as the case may be.

#### 4.3.4.1 Extraction procedure

Triggered by the extraction intent item being published, the digital ballot box bundles a matrix of cryptograms that represent only the valid votes from the ballot cryptograms items on the bulletin board. To be considered valid, the cryptograms must be extracted from the latest ballot cryptograms item for each voter, followed by a cast request item. All other ballots are considered overwritten and, therefore, discarded. This is, essentially, the votes that will be decrypted and duplicated or tallied. Details on this process can be found in the cryptographic protocol document.

The cleansing/extraction procedure is publicly auditable as both the list of vote cryptograms and the initial mixed board are publicly available.

#### 4.3.4.2 Mixing Phase

Mixing is a step conducted by the Trustees in the air-gapped environment for the sole purpose of anonymizing the votes. An extraction file is taken to the trustee application where the cryptograms are shuffled in an indistinguishable way described in further detail in the protocol document. Each trustee is responsible for mixing at least once and each mix is conducted against a given ballot style. If only one ballot of any given ballot style exists the mixing will not anonymize the ballot, thus the condition that when possible this step should only be done with multiple ballots per ballot style are available.

As part of the process proofs of proper mixing are created by each trustee application as well as the result of mixing. These proofs are validated and if the validation passes the mixing step continues to the next trustee who mixes the newly mixed data in turn. This proceeds until all trustees assigned the mixing task have mixed the data at least once.

Although each trustee and subsequent trustee application knows its own shuffling coefficients and could link the votes to the previous mixed extraction the anonymity comes from the fact that trustees should not be sharing this information even amongst one another. There is a necessary level of trust required that each trustee is acting honestly, or at least that enough of them are doing so as to not expose all the coefficients and allow a tie from individual ballots back to individual voters. In this regard, the system relies on a human factors component where the selection of trustees is critical to creating trust in the entire system.

One recommendation to build this trust is to select trustees that either have no stake in the outcome of the election or who have equal but competing stakes in the outcome. In this regard each trustee will not trust one another equally and trust is therefore built on a platform of mutual distrust where the only possible action is to act honestly by all parties. Of course the more trustees involved the harder it would be to convince all to collude against an honest election result and so numbers also come into play. It is recognized that in small jurisdiction elections it may be difficult to find enough qualified candidates to serve as trustees, and in such cases the jurisdiction will have to build the public’s trust in the election operation in the same manner as current election processes already operate.

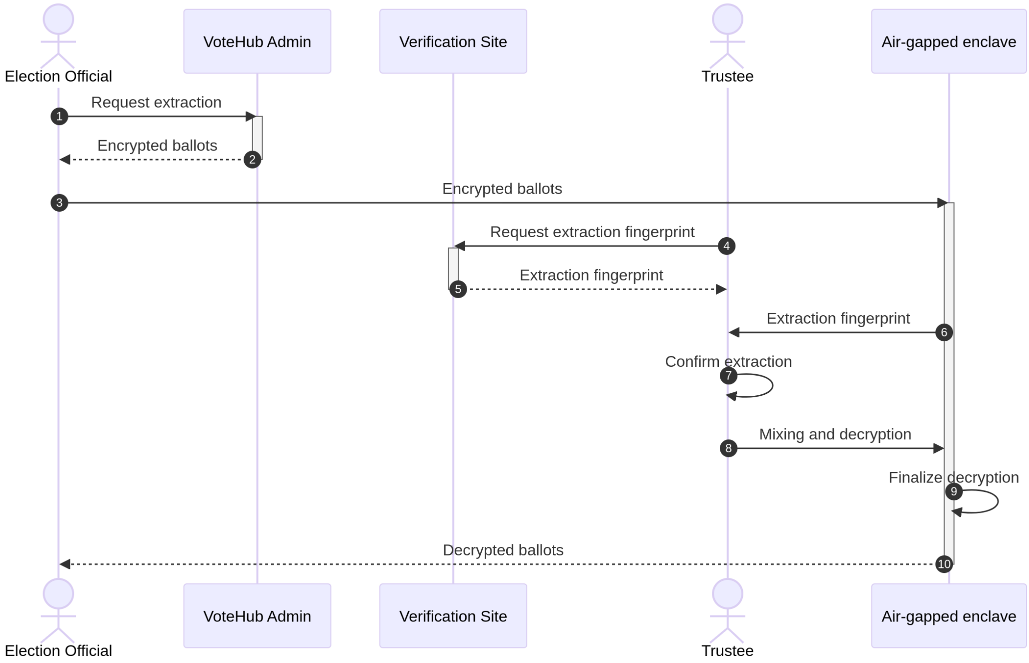
Once the mixing is completed the process moves to a decryption phase, described below, and then either ballot duplication or tallying depending on the setup requirements of the administering jurisdiction.

#### 4.3.4.3 Decryption Phase

Because the link between a vote cryptogram and its voter has been broken during the mixing phase, it is safe to decrypt all the cryptograms from the final mixed extraction as it does not violate the secrecy of any given voter. Furthermore, decrypting this list of cryptograms will lead to accurate and correct results as it contains the exact same votes as the initial mixed extraction, a fact proven by the mixing proofs.

During decryption each trustee must provide their share of the decryption key created during the pre-election phase. It is possible that not all trustees need to be present for this process as a ‘threshold’ number of trustees was determined during the initial setup of the election. For example, an election may assign 5 trustees but only require 3 partial keys present to complete any decryption event. This n of m with regard to trustees present is a failsafe for unforeseen circumstances such as one or more trustees have become compromised, incapacitated, or unable to provide their decryption artifacts to the air-gapped room for any reason. By not needing all the pieces of the key the election decryption can occur even when such circumstances interfere with the election.

Each trustee completes a partial decryption on their trustee application before turning over the results and subsequent proof of work to the trustee administration application. Upon reception of enough partial decryptions the trustee administration application can aggregate the work and complete decryption exposing all the ballots in full. These plain-text results are made available for either ballot duplication or tally actions. More details on the decryption process can be found in the cryptographic protocol documentation.



#### 4.3.4.4 Ballot Duplication or Results Publication

In many early use cases of this system VoteHub will not be the system of record for any election and will in fact serve as a supplemental ballot delivery and electronic marking system. In short this means jurisdictions will not want to produce election results from VoteHub but rather collect marked digital ballots and reproduce or duplicate them into the larger system of record being used to run the rest of the election. In most cases this means transferring the digital ballot markings to a paper system for scanning and counting in the more traditional voting system. To support this process VoteHub has a ballot duplication process that takes decrypted plain-text votes to produce an artifact that an election administrator can use to quickly, accurately and easily recreate a system of record paper ballot for scanning.

This process creates one ballot per voter with at least the following information:

1. A plain-text human readable record of selections per contest
2. A plain-text human readable record of the election details such as name, jurisdiction, election date, etc…
3. A plain-text human readable record of the ballot style representing each individually represented marked ballot
4. An encoded method (such as a QR code) with the same information as above for the purposes of integrating with a ballot-on-demand system that can take marked selection data and print a properly marked ballot for scanning that accurately represents the data of record described above.
5. Some indicator within the system that each marked ballot has been sent out of the VoteHub system for duplication such that the same record cannot be reproduced repeatedly

The election administrator will take the artifact described above and conduct a ballot duplication process outside of this system. It is recommended that some record is kept of duplication such that evidence can be provided for audit purposes that the artifacts produced here were successfully duplicated to the system of record in the proper manner.

# 5.0 Component Integration Points

Previous portions of this document have described the individual components and their responsibilities during different phases of an election. However, the mechanisms by which the various components exchange, validate and share information have been explicitly left out until now. Below is a description of the various communications and data types, some of which have been mentioned previously but not described.

## 5.1 Data Communications

The election protocol uses three types of communication channels to transfer data between two parties, i.e., a sender and a receiver. They are categorized as private, authentic, or public channels. Two relevant criteria differentiate the channel types, namely secrecy, and authenticity.

A secret communication channel implies that any outside observer cannot read the data being transferred. The communication channel provides a way to obfuscate the data. An authentic communication channel involves some mechanism that grants the receiver a confirmation that the data has been genuinely constructed by the sender.

The following subsections describe what criteria are provided by each of the communication channels. The type of channel being used during the election protocol depends on the cryptographic environment available at that step in the process and on the data being transferred.

### 5.1.1 Private Channels

A private channel provides both secrecy and authenticity to the data being communicated. This type of channel is used when the data in transfer is confidential to the two actors communicating but also sensitive (i.e., any tampering with the data causes the protocol to break). A private channel prevents any outsider from reading any part of the data or modifying it. Usually, private channels are used where a cryptographic infrastructure has not been established yet.

The requirement of private channels is seen as a weakness as it introduces external security dependencies to achieve specific properties. In general, the election protocol was designed with the least need for private communication channels.

### 5.1.2 Authentic Channels

An authentic channel provides only the authenticity property to the data that is being communicated. This channel type is used when the data in transfer is not secret but also cannot be tampered with. Therefore, the data must contain proof that it genuinely comes from the sender. An authentic channel protects against a man-in-the-middle attack but allows that man in the middle to read all the traffic.

An example is when exchanging public keys. They are, as the name suggests, public, while they have to represent their owner authentically.

### 5.1.3 Public Channels

A public channel does not provide secrecy or authenticity by itself. Instead, the data in transfer must have built-in mechanisms that ensure secrecy and authenticity. Examples of such mechanisms are encryption and digital signatures.

Obviously, we recommend taking measures to secure all communication channels in an election deployment. Though, theoretically speaking, not all of them need to be secure for the protocol to work. The details of how to mitigate various attack vectors to communications channels through deployment specifics is discussed in the Configuration Documentation associated with this document.

## 5.2 Public Bulletin Board Details

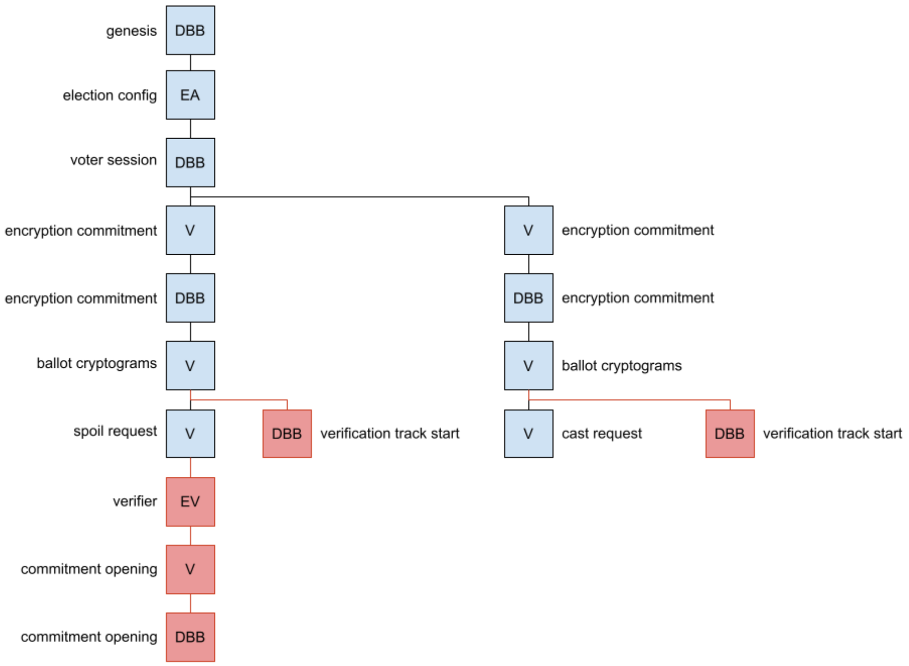
All events happening during an election are published by the digital ballot box as items on a publicly available bulletin board. Each item from the board is owned (or written) by a relevant actor. Each item posted on the bulletin board describes a specific event, and it is uniquely identifiable by its hash value or address. The address of the last item on the board represents the board hash value at that specific point in time. All events are stored as an append only list, meaning no event can be removed or replaced, and each new event is appended at the end of the list. The structure of the bulletin board was inspired by “The append-only web bulletin board" by James Heather and David Lundin, 2008.

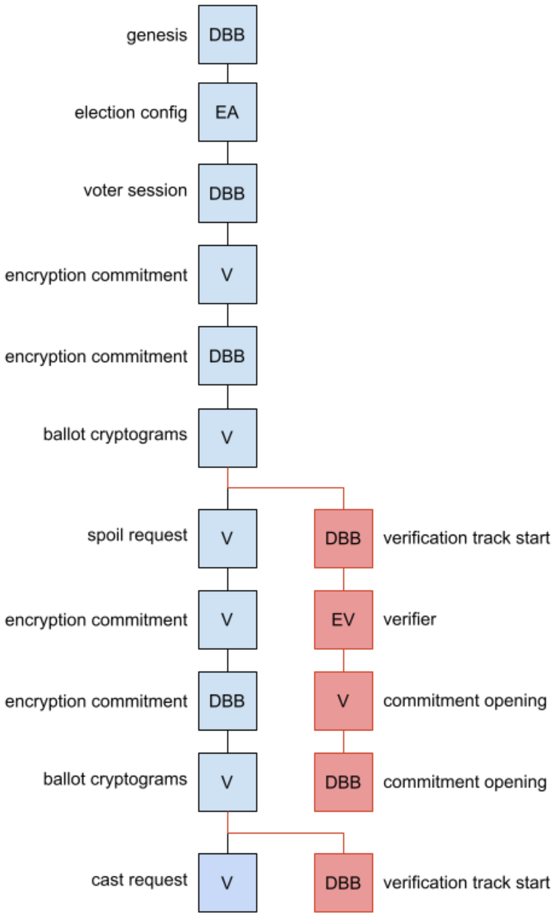
The way we deviate from “The append-only bulletin board”  is that to append a new item on the board, the writer needs to include the address of any existing item from the board as part of the new item, instead of referencing exactly the previous item. We call this reference the parent item. Finally, the address of the new item is computed by the digital ballot box by hashing the content of the item (including the reference to the parent item) concatenated with the current board hash value (i.e., the address of the previous item) and a registration timestamp. The address of the new items is signed and delivered it to the writer as proof of acceptance of that item on the board. Note that it is the digital ballot box which validates/ensures the link of the new item to the previous item on the board.

As of this modification, each bulletin board item references two other items:

- an existing item that the writer chooses as the parent item
- and the previous item on the board.

This modification to the bulletin board structure implies that the digital ballot box protects the history property described in “The append-only web bulletin board" by James Heather and David Lundin, 2008. Furthermore, a new property is introduced to the bulletin board called ancestry, which is defined by all items being related to each other meaningfully. As a result, when traversed on the ancestry line, the structure of the bulletin board looks like a tree. However, when traversed on the history line, the structure looks linear.


genesis
DBB
election config
EA
voter session
DBB
encryption commitment
V
encryption commitment
DBB
ballot cryptograms
V
spoil request
V
DBB
verification track start
verifier
EV
commitment opening
V
commitment opening
DBB
V
encryption commitment
DBB
encryption commitment
V
ballot cryptograms
V
cast request
DBB
verification track start


genesis
DBB
election config
EA
voter session
DBB
encryption commitment
V
encryption commitment
DBB
ballot cryptograms
V
spoil request
V
DBB
verification track start
encryption commitment
V
EV
verifier
encryption commitment
DBB
V
commitment opening
ballot cryptograms
V
DBB
commitment opening
cast request
V
DBB
verification track start

Another new concept is introduced to the bulletin board structure called a hidden verification track, used to perform the ballot checking process described in Section 5.2. It is called:

- hidden because it is not publicly available as part of the bulletin board. Instead, it is available only upon specific request based on the address of a unique item.
- verification because it is used only for the ballot checking/challenging process.
- track because it spawns an extra history of events that is injected under a specific item from the main history.

### 5.2.1 Writing on the board

To write a new item on the bulletin board, a writer must follow the protocol described in the cryptographic protocol documentation. The following actors are allowed to write on the bulletin board:

- **Election Administrator application** - the actor which writes all of the configuration events of an election.
- **Voter Authorizer application** - the actor which authorizes voters to interact with the digital ballot box based on successful authentication.
- **Voting application** - the actors which write all of the vote-related events of an election.
- **Digital Ballot Box** - the actor which ultimately accepts all of the events published on the bulletin board. In addition, the ballot box also writes all of the auxiliary events supporting the voting process on the board. See Section 4.3.3.5
- **External Verifier application** - the actor which writes events related to the ballot checking process. These events are written on the hidden verification track of the bulletin board.

The protocol for event writing to the board can be found in detail within the cryptographic protocol paper. The basics are as follows. A writing application chooses an event type and the content to be appended to the bulletin board. Event types can be found in Section 5.2.2. Each event type has a predefined data structure set by the rules described in the protocol document. The writing application then chooses a pre-existing item on the bulletin board as the parent, which is referenced by its address, of their suggested new event. There are some rules regarding the choosing of parent items described in the protocol document as well.

The writing application sends the new event to the digit ballot box signed with their own private key as a request to append the new item on the board. The ballot box verifies the item is of valid structure based on the rules defined as well as checking for a valid signature. If all validations succeed the ballot box computes the address of the new item and stores it on the bulletin board as an item. The ballot box provides details to the writing application as evidence the item was appended successfully on the board.

Details on the validations and this computational process can be found in the protocol document.

Finally, upon receipt from the ballot box the writing application verifies that the address of the new item is computed correctly and that the response has a valid signature. Once verified, the writing application closes the process and the new item on the bulletin board is final.

### 5.2.2 Bulletin board event types

The bulletin board has been designed as a self-documenting event log. To support its function as such, it must contain many kinds of items that document different events throughout the election. These include events related to the pre-election phase for configuring the election, events related to the voting process, or events associated with the post-voting phase for publishing items such as a results event. Each event is documented as an item on the bulletin board.

All bulletin board items are structured as described in the protocol document but each item type has its own rules when it comes to:

- what data it contains,
- who the author is,
- what parent it can have.

The comprehensive list of item types and rules can be studied in the protocol document. The list below briefly describes all item types which can be grouped into the following categories:

**Configuration items**

1. The *genesis* is the initial item of the bulletin board which describes some metadata of the election. This is the only item that doesn't have a parent reference, as it is the very first item on the board.

2. The *election configuration* specifies the configuration on the election level (e.g., election title, enabled languages). Follow-up election configuration items act like configuration updates. Generally, all configuration items reference the previous configuration item as a parent.

3. The *contest configuration* is an item defining the configuration of a contest. It contains a unique identifier of the contest, its marking rules, question type, result rules, and the list of candidates with their distinct labels. Follow-up contest configuration items with the same contest identifier act like updates to that contest configuration.

4. The *threshold configuration* is the item that defines the ballot encryption key and the number of trustees necessary for decryption to occur. It also specifies all of the trustee data: the set of trustees, their public keys. The threshold configuration cannot be updated during the election phase.

5. The *actor configuration* is an item that introduces a new actor on the bulletin board. This new actor is defined by a role and a public key. The role that actors can have is one of the following: the Voter Authorizer. New roles might be included in the following versions of the system.

6. The *voter authorization configuration* is the item defining voter authentication details. The item defines the voter authorization mode and the configuration of the Identity Provider.

7. The *voting round* item describes what contests can be voted on at any given time. Voting round is synonymous with the voting period. The item also defines how long the election phase lasts (i.e., the start and end dates of the period). At least one voting round must exist per election but multiple voting rounds can be enabled simultaneously or follow each other sequentially.

**Voting items**

8. The *voter session* is the item indicating a voter has been successfully authorized to vote. The item contains the voter identifier, the voter's public key, and an authentication fingerprint used for auditing. When a voter tries to vote again (therefore overwriting the previous vote where jurisdictionally allowed), a new voter session item is generated for the same voter identifier.

9. The *voter encryption commitment* is the item which settles the encryption parameters chosen by the voter during the vote cryptogram generation process. Note that this item is written by the voter, while the public key is defined in the voter session item. The item is used in the verifications that happen during the process of challenging vote cryptograms (section 4.3.3.5).

10. The *server encryption commitment* is the item which settles the encryption parameters chosen by the Digital Ballot Box during the vote cryptogram generation process. The item consists of the commitment to the randomizer values of the Digital Ballot Box. This item is generated in response to the voter encryption commitment item being published. The item is used in the verifications that happen during the process of challenging vote cryptograms (section 4.3.3.5).

11. The *ballot cryptogram* is the item which contains the encrypted digital vote.

12. The *cast request* is the item which documents the action of casting a previously submitted vote.

13. The *challenge request* is the item which documents the decision to challenge a previously submitted vote cryptogram.

### Adjudication items

14. The ballot accepted is the item that marks a ballot accepted to be included in a tally, after it has been verified through the adjudication process.

15. The ballot rejected is the item that marks a ballot rejected from being included in a tally, after it has been verified through the adjudication process.

### Hidden items

16. The *verification track start* is the initial item of the hidden verification track, essentially spawning a verification track for each ballot cryptogram item. The item is automatically written by the digital ballot box after a ballot cryptograms item has been posted.

17. The *verifier* item defines the external verifier and its public key. The external verifier is involved in the ballot checking processes when a voter chooses to challenge their ballot submission and check the contents of their ballot cryptogram. Its public key is used for encrypted communication, such that the privacy of the ballot checking process is achieved.

18. The *voter commitment opening* is the item containing the voter's encryption parameters which are necessary for unpacking the challenged encrypted ballot. This data is encrypted, so only the external verifier can read it.

19. The *server commitment opening* is the item containing the encryption parameters of the digital ballot box, which are necessary for unpacking the challenged encrypted ballot. This data is encrypted, so only the external verifier can read it. This item is generated as a response to the voter commitment opening item being published.

### Result items

20. The *extraction intent* is the item which documents the request for a result to be computed. The request is made by the election administrator.

21. The *extraction data* is the item which lists all of the ballot cryptograms making up the initial mixed board. These cryptograms are the only ones that will be counted as part of the election result. Cryptograms that had been challenged by the voter or cryptograms which were replaced through a traditional spoil method are not included in the extraction data.

22. The *extraction confirmation* is the item which documents that the result has been computed. It contains fingerprints of the files containing mixing and decryption data leading to the final result. All of this data is signed by the trustees, proving that the rightful actors have computed the result.

## 5.3 Voter Authentication Details

During the pre-election phase, the voter authorizer service is loaded with a list of eligible voters. To be authorized to cast a vote on the bulletin board, a user has to authenticate to the voter authorizer as a valid voter. Once authenticated and authorized, a voter can interact directly with the digital ballot box.

The voting system supports two mutually exclusive voter authentication modes: **credential-based** and **identity-based**. One involves proving possession of credentials that have been pre-established before the election starts, while the other consists in proving ownership of some identity provided by a third party. Some actors mentioned in the authentication modes presented below are exclusive to that mode only (i.e., a credentialing authority for credential-based mode and identity provider for identity-based mode).

### 5.3.1 Identity-based mode

In this mode, the voter authorizer service lists a set of third-party identity providers. Voters have to authenticate with all identity providers and receive identity tokens from each of them. Then a voter must submit all identity tokens to the voter authorizer, which checks whether they relate to an eligible voter identity from..

Because the voting system has to integrate into third-party identity providers, all voters must be defined with distinct identities supported by all approved identity providers.

For auditing purposes, the voter authorizer stores all identity tokens received for each successful authorization performed. This must be audited and validated during the administration auditing process, as described in Section 6.0.

# 6.0 Auditing

Auditing is a unique activity because some steps MAY be done during election setup or live voting but most will be completed after an election is completed. Auditing is also unique in that it is the only aspect of the voting system that is expected or CAN BE completed entirely independently of all other aspects of the system. The only dependency audit activities have is the availability of access to the system and tools associated with various audit tasks/goals. For these reasons auditing is described independently of the voting system as audit activities/capabilities supported by the voting system rather than a description of how to audit the system. Expanded details regarding audit capabilities may be found in supporting documentation as well.

# Revision History

| Version | Date | Author/Publishing Agent | Description of Change(s) | Version Reviewer |
|---|---|---|---|---|
| 0.1 | 09/12/2023 | David Wallick | Initial Version | *In Draft* |
| | | | | |