



Technical description of the Waves Enterprise Voting

Release master

<https://wavesenterprise.com>

May 03, 2023

ABOUT THE SYSTEM

Functioning and development of society are impossible without a mechanism for making collective decisions. These mechanisms are elections and voting, which are considered to be the fairest and most democratic form of determining public consensus today.

With the intense social interactions that characterize modern society, almost everyone faces the need to participate in decisionmaking procedures. However, despite the incredible development of technology, voting tools remain archaic.

Nevertheless, information technology allows a huge number of tasks to be done online, but most voting procedures still require either inperson attendance or the printing, distribution, and processing of paper ballots.

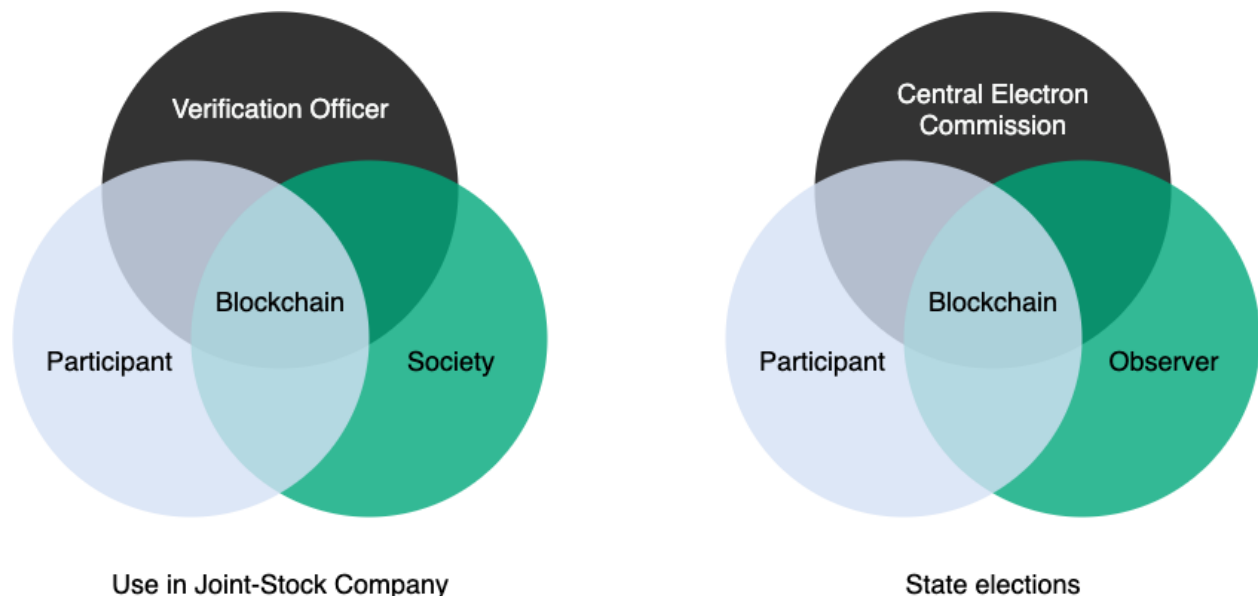
The development of electronic voting systems eliminates the problem of personal presence, but their application in practice is still limited. This slow implementation of technology can be considered partially justified. Besides the general inertia of large public structures, which does not allow them to enjoy all the advantages of advanced technologies, there are concerns about possible falsifications.

It should be admitted that the opportunities for falsification in electronic voting are not more frequent than in the “paper” version. The weak point in both cases is the centralized body that stores and counts votes. Though this body is trusted for voting participants, if there is a possibility to change a paper ballot or to “correct” a record in an electronic database, it is impossible to exclude completely possibilities for manipulations with voting results.

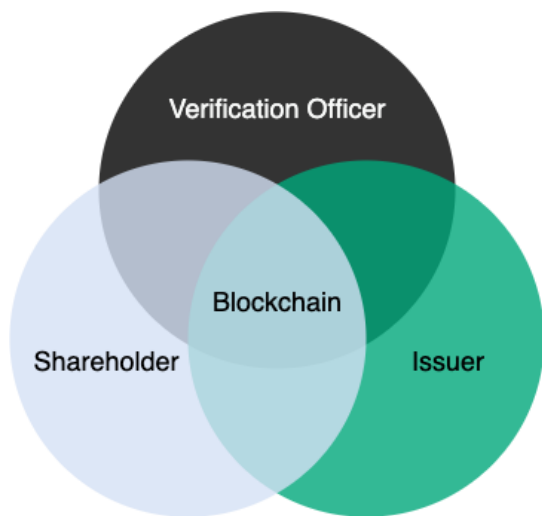
It is in the field of voting that distributed registry technology most clearly demonstrates its advantages mainly , transparency and security.

Using the properties of the blockchain, as well as modern cryptographic algorithms *cryptographic protection of data integrity*, decentralized *consensus*, inability to manipulate information, *homomorphic encryption* and separation of encryption keys Waves Enterprise has developed the Waves Enterprise Voting platform, allowing any organization in which decisions are made collectively to organize and conduct trusted voting.

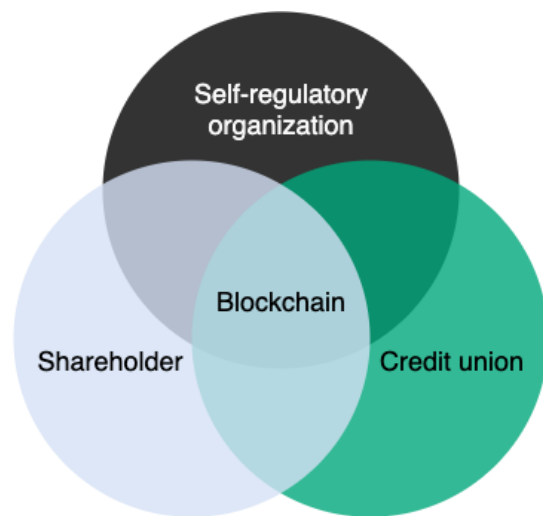
System Users



Basically, any organization in which decisions are made using voting — these can be system users. First of all, these are limited liability companies, credit cooperatives, and jointstock companies. The solution is also suitable for selfregulatory organizations — homeowners’ associations and management companies. The platform can also be used to organize municipal and federal state votes.



Use in Limited Liability Company



Application in credit cooperatives

PROBLEMATICS OF ONLINE VOTING

Although online voting is gaining in popularity, it presents new challenges.

First, most online voting services are centralized solutions that are fully managed by the operator. Such solutions may be comprehensively protected from external threats, but theoretically give their developer access to the votes being organized. Accordingly, the operating company can somehow interfere in the voting process or obtain information about the results. This creates a trust problem between the organizer of the voting, its participants, and the service company.

The problem of trust is solved by creating a decentralized system that meets two basic criteria:

- only the organizer and participants have access to the voting process and the results;
- only the participant has access to his or her own ballot.

One of the technical solutions allowing to implement such a system is **blockchain** – a system of data storage and transfer in the form of a successive chain of interconnected blocks. Each block contains the hash sum of the data of the previous block. This makes it impossible to subsequently change the content of any of the blocks, since it is necessary to change the content of the blocks throughout the chain at all nodes.

The blockchainbased system does not have a single control center; all data is stored simultaneously on all nodes in the network, either open or encrypted. This ensures the security and integrity of transmitted data, minimizing the possibility of forging.

The use of modern encryption algorithms and rules for handling sensitive data, along with the blockchain technology, protects the system as much as possible from possible attacks.

CONCEPT OF THE BLOCKCHAINBASED ONLINE VOTING SERVICE

The use of blockchain technology makes it possible to create a secure voting system which works according to the following rules:

- The organizer fully administers the rules for access to the voting agenda.
- The list of voting participants is created by the organizer of the voting.
- The persons indicated in the voting list shall take part in voting.
- The rules for voting are set by its organizer.
- The integrity of his or her ballot with an agenda is guaranteed for the participant.
- The secrecy of the vote is fully respected: only the participant himself has access to the contents of his ballot.
- Only the organizer and participants have access to the voting results.
- The integrity of the voting results is guaranteed for the organizer and participants.

In such a system, blockchain acts as a universal solution for storing and transmitting information. Data published on the blockchain through transactions cannot be changed without changing the entire chain. In addition, each transaction is signed with the sender's public key. Therefore, blockchain data can be used to verify the integrity of data transmitted to voting participants from the service's backend.

Since all data in the blockchain is public, encryption algorithms are used for the system's sensitive data, also guaranteeing its integrity.

THE ADVANTAGES OF WE.VOTE

The WE.Vote service implements the concept of a secure online voting system to the fullest extent.

The service is based on the Waves Enterprise Mainnet blockchain network, which allows participants to interact with the system:

- receiving, storing, transmitting and reconciling data on voting and votes of the participants;
- delimitation of access rights by means of a blockchain permission model.

Encrypted data is stored simultaneously on all blockchain nodes, guaranteeing its integrity and impossibility of loss.

Each participant's account is provided with a key pair for voice signing and blockchain identification. The key pair, as well as the means to access it, can be stored either in the participant's own possession or in the service's secure cloud storage. This allows the participant to determine the security of their credentials and, if the key pair is stored in cloud storage, to quickly restore access to the account and votes.

Voting data is protected by cryptographic algorithms that encrypt the votes of participants and hash out the voting materials. The El Gamal encryption scheme used allows votes to be counted without decrypting them.

A zeroknowledge proof system is used to ensure that each participant can verify that his or her vote is recorded correctly. This system allows a participant to record the fact and content of ballots without revealing the content to the vote organizer.

To easily organize and conduct online voting, the service is equipped with an intuitive web client.

CREATING AND CONDUCTING YOUR FIRST VOTE

The WE.Vote service gives you the opportunity to learn the basic features of voting. To do this, 20 free ballots are added to each new user's balance.

To create your first ballot using demo ballots, follow these steps:

1. Go to **we.vote** website and click **Log in**. You will be redirected to the login page of the WE.Vote client application.
2. Select the **Create Account** link.
3. Enter your email address and password, then confirm your email address with the link in the email you received.
4. Log in to the client application with your username and password.
5. Enter your name and the name of your organization.
6. Click the **Create** button in the message that appears in the upper right corner of the screen.
7. Choose how you want to store the seed phrase: in the cloud or by yourself.
8. If you choose selfcustody, write down your seed phrase and store it in a safe place.
9. Invite participants: Click on **Participants** in the upper right corner of the screen.
10. Click **Add Participant**.
11. Enter the email addresses of the voters, separated by commas.
12. Click the **Invite** button. Invitation letters with a link to register in the service will be sent to the email addresses of your invited participants.
13. Press **X** in the upper right corner of the screen to return to the main service menu.
14. Click the **Create New** button to create a new vote.
15. Enter the name of your vote.
16. Select the date and time of the vote, its type, and quorum.
17. Enter the items to be placed on the voting agenda. To add a new question, click **Add Question**.
18. Check the boxes next to the participants you selected to vote.
19. If necessary, upload additional materials for participants.
20. Click **Publish**.

You do not need to do anything to start voting: it will start automatically at the defined time.

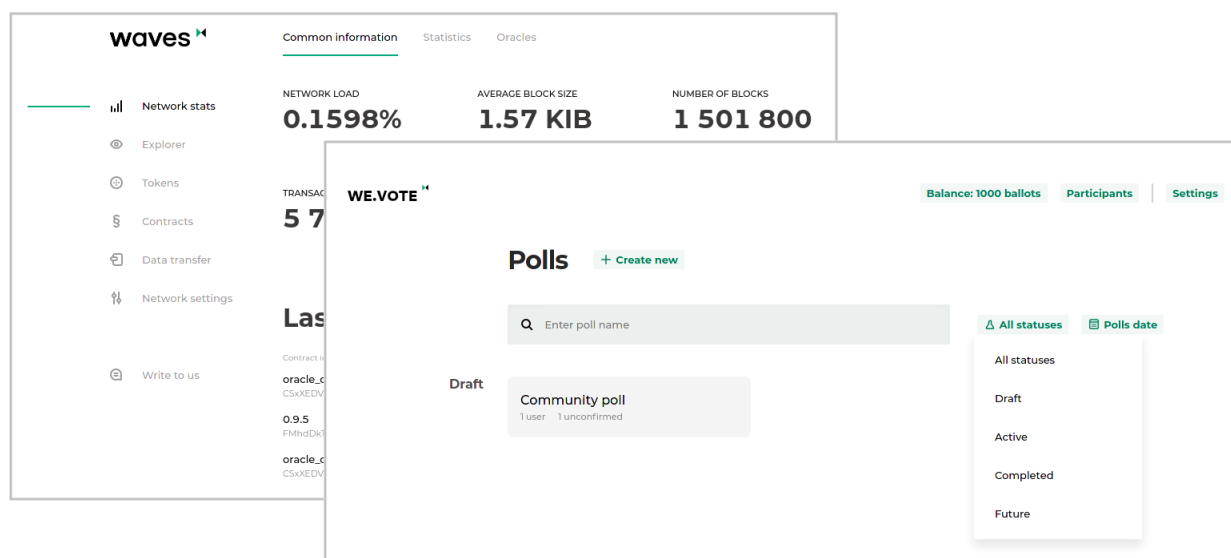
To take part in voting:

1. Choose the card of your voting in the service main menu.

2. Answer the agenda questions by clicking radio buttons or checkboxes next to preferred answers.
3. Click **Vote**.

Voting results will be published automatically upon its finish. All participants will receive notifications per email after the end of the voting and will be able to look through its results.

ACCOUNT REGISTRATION



Before using WE.Vote, create an account in the [system client](#). You can do this either by yourself or by invitation from another member. To work on WE.Vote, we use a common account for the whole ecosystem of Waves Enterprise services. If you have already registered on our resources, simply enter your username and password on the login page.

5.1 Selfregistration

To register in the client, click **Create Account**. Enter your email address, make up a strong password and click **Register**. Confirm your email address by following the link in the email. After this, you will be able to enter the service and proceed your work.

At the first entrance we will ask you:

- enter your name so that your colleagues can recognize you in the system;
- enter the name of the organization on whose behalf you plan to vote.

We will send invitation letters to your colleagues on behalf of the specified organization. This will allow them to pay attention to the email and not send it to “Spam”.



Sign in

Enter login and password

Sign in

[Create an account](#)

Specify you name

Your name will help your colleagues to recognise you.

Continue

Specify you company's name

You'll run polls on your company's behalf. Your company's name will be specified in invitation e-mail title.

[Back](#)

Continue

5.2 Registration upon invitation letter

If you have been invited to sign up for the WE.Vote to participate in a voting, you will receive a welcome email with a link. Follow this link to set your password for your account and log in to your WE.Vote account.

When you log in, we will ask you to join the organization that invited you. If you agree, you will have access to votes and polls of this organization.

5.3 Password change and reset

In case you forget your account password, we can reset it. After this, set a new password and restore access to your personal account. Keep in mind that access to WE.Vote is determined not only by knowing your account password, but also by access to your secret key.

This key is used to electronically sign your vote. The procedure for changing/resetting your password will be different for users storing their keys in cloud storage or on their own. See the next section for more information.

5.4 Key generation and storage

WE.Vote is built using the blockchain technology, which allows it to provide an unprecedented level of information security unattainable by conventional online services. This is achieved due to the fact that any interaction of participants of the voting process with the system is a transaction signed with a unique secret key.

Each user has two keys, a public key and a private key. The public key serves as the user's identifier in the blockchain, and the private key is used to sign their vote. If you are a participant in a particular vote, your key will be registered in a smart contract. The service will only accept and count votes that were sent and signed by one of the registered keys.

Such strict rules for handling data impose an increased responsibility for the security of the key if you lose it, you will not be able to participate in already published votes.

To make the keys easier to work with, on WE.Vote they are transformed from a long random set of letters and numbers into a mnemonic form a set of 15 words called a **seedphrase**. By saving the seed phrase, you retain access to your private key and therefore the ability to vote.

WE.Vote provides two options for storing secret keys: independent storage on the user's device (selfcustody), or in the service's secure cloud storage.

Selfcustody is more preferable for users who are ready to determine the desired level of protection of their key and secure it by their own means: write it down on a piece of paper and put it in a safe, save it in a password manager, save it in a text file with notes, make a screenshot, etc. In this case, the security of the key is entirely in the hands of the user: your key is not transferred and is not stored at WE.Vote, we will not be able to restore it in case of loss.

When working with the service, the private key is stored locally, on the user's device in the browser's secure storage. The key is encrypted with the user's password. Once you enter your login and password and log in to WE.Vote, the key is decrypted and can be used to sign your vote. To be able to vote on a different device or browser, you need to transfer your private key to the new device yourself. This is what the seed phrase is for, which is entered when logging into the user's account on another device and generates a private key.

Changing and resetting the password when storing the seed phrase by itself also has its own peculiarities. During a regular password change, when we ask you to enter the old and new password, the app will locally



Greetings,

You've been invited to participate in e-voting at [WE.VOTE](#).

User group: [REDACTED]

Group administrator: [REDACTED]

WE have built WE.VOTE using modern cryptography and blockchain technology in a way that gives our users an opportunity to vote, with no one (us included) able to forge your vote or corrupt the decision being made.

[Sign-up](#) or [sign-in](#) using your Waves Enterprise account credentials to participate in e-voting.

Please note, that beside account registration it is required to create and store secret seed-phrase. At poll start your unique key will be saved at poll's smart-contract. If you'd require seed-phrase reset, your key will be updated as well and your vote, signed with this new key will be rejected.

Best regards,
team Waves Enterprise.

Follow the news on social networks:





Restore access

Enter an email that you have used for account registration. We will send you account recovery instructions.

Sent

[Go back](#)

js

Blockchain

lema



Secret phrase required



It is impossible to participate in voting without it

Create

Skip

Valid Participant



- ✓ Create Personal Key
- ✓ Register Personal Key
- ✓ Vote with Registered Key



WE.VOTE Smart-Contract



Malicious Participant

- ✓ Create Personal Key
- ⊗ Register Personal Key
- Vote with Unregistered Key



Seed storage

Secret seed-phrase is an automatically generated mnemonic password. You couldn't vote without it.

Choose storage method

Comfortably

In the cloud

We do take care of your seed-phrase

- ✓ Your seed phrase is encrypted and stored safely
- ✓ You could access your seed from anywhere
- ✓ No need to worry about losing your seed

Self-custody

You store your seed-phrase yourself as you find appropriate

- ✓ Only you have access to your seed
- ✓ You require to export seed to any device you want to vote with
- ✓ You could reset your seed in case of a loss

Continue

reencrypt the key with the new password and you will be able to continue your work. At that, on other devices where you have already transferred your key, the reencryption will not happen: you will need to transfer the key to that device again using the seed phrase.

If you've forgotten your password and you need to reset it, we'll help you do that and you'll regain access to your personal account. But the key on your device will be encrypted with the old lost password, and you won't be able to use it: WE.Vote will ask you to enter the secret phrase to retrieve your key from it, encrypt it with the new password and save it.

Since selfcustody has many complicating factors, WE.Vote provides an alternative, no less safe and reliable way **storing your user key in the cloud storage of the voting service**. If you choose this option, your key is created by a dedicated key storage service.

The seedphrase you create is tied to your account and stored in the cloud service in encrypted form. When you send a voice, the key will be encrypted locally on your device and then signed by the key storage service. This way, your key will not be transmitted over the network without encryption, and therefore cannot be stolen during transmission.

Creation of the key is performed instantly after selecting the method of key storage in the cloud storage. You won't have any problems with storing the key or transferring it to other devices: it will be accessible from any browser and from any device. It cannot be forgotten or lost. Resetting or changing your password also does not affect the availability of the key in any way: WE.Vote takes care of all that.

After creating your account and secret key, you can start using the service.

PERMISSIONS

WE.Vote users can have two roles: **Administrator** or **Participant**.

Administrators have the following powers:

- edit the group name and description;
- add and delete participants;
- create and run polls;
- appoint other participants as Administrators;
- refill the balance of the group (see “Payment and balance” for details).

The administrator can view all votes of his organization, even if he or she does not participate in them.

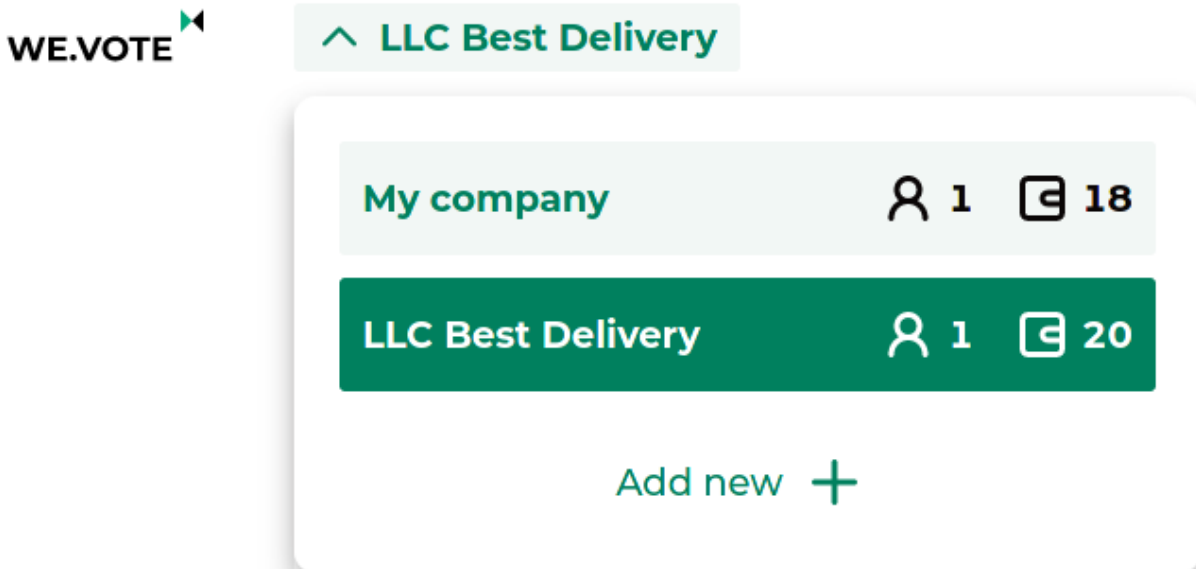
The first user who creates an account in WE.Vote and adds an organization is automatically designated as an Administrator in the organization workspace. He or she can then designate other users as Administrators in the **Participants** window for better management of voting.

Participants are allowed to participate in polls, as long as the account is verified (see “Unverified Participants” for details). Participants cannot view or refill balance or vote in polls they are not participating in.

ADMINISTRATOR GUIDE

7.1 Preparing to vote

7.1.1 Organizations



We will create a dedicated workspace for an organization on whose behalf you plan to vote on WE.Vote . There, you will be able to manage your colleagues' accounts, configure participant groups, and run votes.

If you're organizing votes in multiple companies at the same time, you can add a new organization to the service. We will also create a fully isolated space for it with its own users, votes and balance.

To use the WE.Vote service, replenish your ballot balance. Ballots are spent during voting. For example, to start voting for 10 participants, your balance must be at least 10 ballots. When you start voting, the required amount of ballots is frozen in your balance. It will be spent if the voting ends without any glitches from the service. Every new user has the opportunity to try out the functionality of the service. For this, we charge 20 free ballots.

To refill your balance, click on the **Add balance** link in the **Balance** menu section. In the opened window, select the appropriate rate and payment method. Each package of ballots has a validity period after purchase. The ballots you launch must fit completely within the validity period of the purchased package. In addition to fixed packages, you can negotiate individual terms of service for your organization. To do this, check

the **Enterprise** box: when you click on the **Send request** button, you will be redirected to the Waves Enterprise feedback form.

Only **Administrators** of the organization can refill the balance. After selecting the tariff and available payment method, you will be redirected to the payment page, where you will need to select your preferred payment source on the left side of the window and enter the necessary data. Votes will be credited to your account after the payment is made. History of your payments and their status, as well as facts of writing off votes from the balance are available in the **Balance / View entire history** menu.

7.1.2 Invitation of participants

To invite your chosen voters, upload their email to the system. We'll send them invitations to sign up for WE.Vote. Your colleagues will need to create and confirm an account, choose a method of storing their secret key, and create a key.

As the voting Administrator, you can see which of the invited participants have successfully registered and can take part in the voting. If any of the participants have problems with registration, you can send them an invitation again by pressing the **Invite again** button on the **Participants** tab. You can also resend invitations to participants when you create a vote by clicking the **Resend invites** button.

You can add users by specifying their email addresses separated by commas in the **Add User** form. But a more convenient way is to upload a list of users as a Microsoft Excel spreadsheet or a CSV file. A template for creating the file is available in the **Add User** form.

After your member list is uploaded, you will see a message about the results of the upload. If there are incorrect entries in your list, you will see a warning message with a reason. If this is the case, correct the incorrect entries and upload again.

On the cards of the added users, you can assign additional Administrators to help organize the voting, or configure the parameters of the participants the decisive vote right and vote weight. These parameters will be used in the corresponding types of voting.

If one of your colleagues didn't notice the invitation and didn't register on the service, it will be visible in the list of organization members: the card of an unconfirmed user is shaded. In this case you can send the user a second invitation.

7.1.3 Unverified participants

Not everyone who is invited will register immediately after the invitation. Someone may mistake the invitation for spam and not pay attention to it, so we recommend to inform your colleagues about the planned voting on WE.Vote. As long as the vote is in a draft status and not published on the blockchain, all invited users have time to register in the service. Their status will automatically change to **Activated** and they will be able to participate when the vote is published. However, those who don't register before that point won't be able to send their vote because their public keys won't be registered in the voting service's smart contract.

WE.Vote users who store their keys in the cloud storage cannot lose them. When storing on his own, the user may lose access to their key for one reason or another. In this case, even a registered user would have to use the passphrase reset function and would not be able to participate in already running or published votes. After updating the secret key, the user will be able to participate in future polls again, but the current active polls will not be available to him.

Buy more ballots

1 Choose plan

<div><input checked="" type="radio"/></div> <div>50 50 Ballots 0.1 EUR Valid for 10 minutes</div>	<div><input type="radio"/></div> <div>Most popular 100 100 Ballots 3 EUR Valid for 15 minutes</div>	<div><input type="radio"/></div> <div>Best price 300 300 Ballots 5 EUR Valid for 3 hours</div>
<div><input type="radio"/></div> <div>3000 3000 Ballots 7 EUR Valid for 3 hours</div>		
<div><input type="radio"/></div> <div>Enterprise If you need something special</div>		

2 Select currency


Payment method

Bank card EUR

▼





3 Confirm request

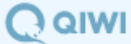



Buy 0.1 EUR

 **ROBOKASSA**
SERVICED BY PAYSEND

Ru

Payment method:
Bank Card



 Others:   

Order


Payment


Enter card details and e-mail

Card number

MM/YY

CVC/CVV





Email for receipt

john.doe@main.com

By clicking the "Pay" button, you agree
with [The Service's Terms Of Use](#) and [The Public Offer](#).

Type Email

Email (one or few separate by commas)

Enter group name

Invite

CSV: [Example](#)  [Upload](#) 

7.1.4 Permission management

If you plan to organize voting among a small number of people, it's enough to invite participants to the service (see “Invitation of Participants” for details). However, when working with a large number of users, you can manage the voting process more efficiently by dividing them into groups. This will allow you to structure the list of participants depending on your needs: distribute participants by departments or divisions of the organization, their location, etc. Later on, you will be able to create different votes for different groups of users more quickly and manage the composition of the groups.

A group is created when you invite participants. Enter a commaseparated list of email members for the group to be created, or upload a CSV/Excel file with the list of them, then type the group name in the appropriate field. To add members to an existing group, select the desired group from the dropdown list.

To add members to an existing group, select the desired group from the dropdown list.

When you create a poll, you can select the groups of participants for which it is intended.

7.1.5 Permission management

As an administrator, you can designate any participant from your organization to be an Administrator to help you organize voting. To do this, go to the Participants section, click on the participant's card and select **Administrator** in the **Role** dropdown menu. Immediately after that, the participant will receive administrator rights.

To return an administrator to the participant's role, click on his or her card and select **Participant** from the **Role** dropdown menu.

7.2 Voting creation

To create a poll, click the **Create New** button in the main client window. Please note that creating a vote is not supported when using the client on a device with a small screen diagonal. In this case, the button will be extinguished.

+ Add user

Type Email


john.doe@gmail.com,
albert.lenz@gmail.com,
ivan.ivanov@gmail.com

Enter group name

Financial department



Group not found

CSV: Example  Upload 

Information

Email: a@a.ru

Last name:

First name:

Role: Administrator (dropdown menu open showing Administrator and Participant)

Weight: 1

Groups

Add group +

Remove user (trash icon)

7.2.1 Basic settings

1. In the opened window, enter your preferred voting name.
2. Then set the start and end dates and time for voting.

Important: Note that the voting start time must be longer than the current time, since the service needs to create a smart contract for your ballot and then do distributed key generation to encrypt the ballots (*Cryptographic algorithms*).

3. Select the type of voting from the dropdown list:

Common settings

Poll dates

Poll start 27 April 2021	13:19	Poll end 27 April 2021	14:19
-----------------------------	-------	---------------------------	-------

Poll type

Select poll type

Basic poll

Basic poll

Weighted poll

Decisive Vote

Common (majority) voting

Basic voting option. Each participant has one vote and may cast a vote for one of the proposed options for each of the issues on the voting agenda. The results are determined by counting the sum of the votes cast for each option; the one with the majority of votes wins.

Weighted voting

A voting option that is appropriate for decisionmaking in an organization where the strength of every vote is determined by the share of participation or ownership for example, a limited liability company or an association of home owners. The strength of the vote (its weight) is set before a voting begins. The option for which the members with the highest combined weight voted wins.

Voting with a decisive vote

Voting with a decisive vote is necessary for decisions made by a small number of participants when there is a high chance of equal division of votes between the options, which leads to blocking the decision. In this case, one of the participants in the voting, such as the chairman of the company's board of directors, obtains a decisive vote. This right is exercised only in the case of a tie the option for which the chairman voted wins. In other cases, his vote has the same effect as that of the other board members.

Multiple choice

With this voting option, each participant can choose more than one answer for each question.

4. Set up the voting privacy. To do this, click on the **Open Voting** toggle switch if you plan to vote with information about the votes of all participants. In this case, participants' votes are not encrypted, and at the end of voting WE.Vote provides a detailed report of each participant's choice.

Note: Note that in this case results of your voting will be available to every participant.

5. Set up a voting quorum:

- **No required** quorum is disabled, voting is considered valid with any number of voters.
- **More than half** voting is considered valid when $> 50\%$ of participants are present.
- **More than twothirds** Voting is considered valid when $> 2/3$ of the participants are present.
- **Unanimously** voting is considered valid only if all registered participants are present.

The screenshot shows a configuration window for a 'Basic poll'. At the top, there's a dropdown menu labeled 'Basic poll'. Below it, a toggle switch is labeled 'Poll is open (votes of each participant are visible)'. Underneath, there's a section titled 'Poll quorum' with a dropdown menu labeled 'Select poll quorum'. The dropdown menu is open, showing four options: 'Any', 'More than half of the votes', 'More than 2/3 of the votes', and 'Unanimously'.

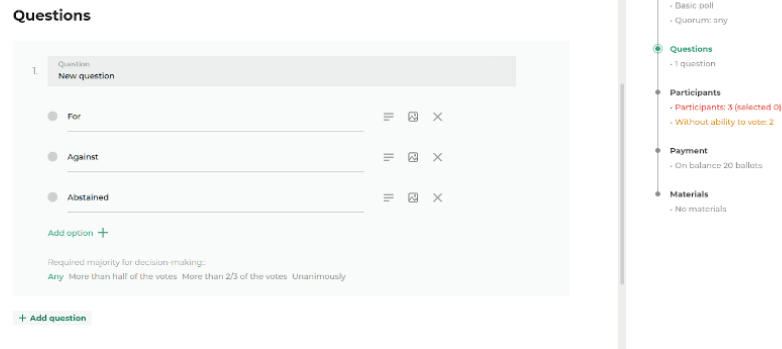
6. Configure the ability to stop voting before it ends. To do this, press the **Voting can be stopped before the end date** toggle switch. In this case, the administrator can stop voting at any time after it starts, the results are summed up in accordance with the distribution of votes before voting stops. If this option is not selected, voting continues until the specified moment of its end.

7.2.2 Adding questions

Scroll down the page to go to the add question card. To create a new card, click the **Add question** button.

In the question card:

1. Enter the question text in the **Question** field;
2. Add answer choices.
3. In case you have selected the **Multiple Choice** option, adjust the number of possible answers to the question.
4. Set the required majority to decide on the issue. The options in the card are identical to the options for setting up a voting quorum.



7.2.3 Adding participants

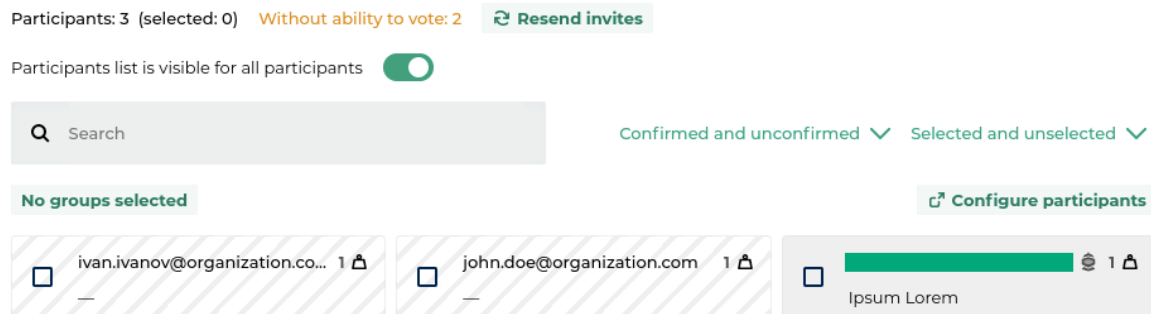
Go to the **Participants** section. To add individual participants to your vote, check the boxes on their cards. You can search and filter the attendee list using the search bar and section filters.

To add a previously configured group of participants, click on the **Groups not selected** button and check the box next to the group. The members of the group will be highlighted in the general list.

If a participant you added to the organization has not been registered in the WE.Vote system, his card will be shaded in gray. Participants who have not verified their accounts in the system before the start of the voting will not be able to participate in it (see section: *Unconfirmed participants*).

To send invitation letters to unconfirmed participants, click the **Resend invites** button.

Participants



7.2.4 Payment

Payment

On balance: 20 ballots

Price of the poll: 0 ballots

Add ballots

In the **Payment** section, you will see the current balance of your WE.Vote account, as well as the cost of voting the number of ballots required to conduct it. If there are not enough ballots in your balance to vote, click the **Add ballots** button and purchase an additional package of ballots.

7.2.5 Uploading of additional materials

In the **Materials** section, you can attach explanatory files or additional documents for your vote. Participants will be able to view them when the vote is published. The materials are available for download before the start of the voting, as well as during it.

Materials

If necessary, add materials to the poll. Members will see them when the poll is published



Click and select files to upload or drag and drop them to this area.

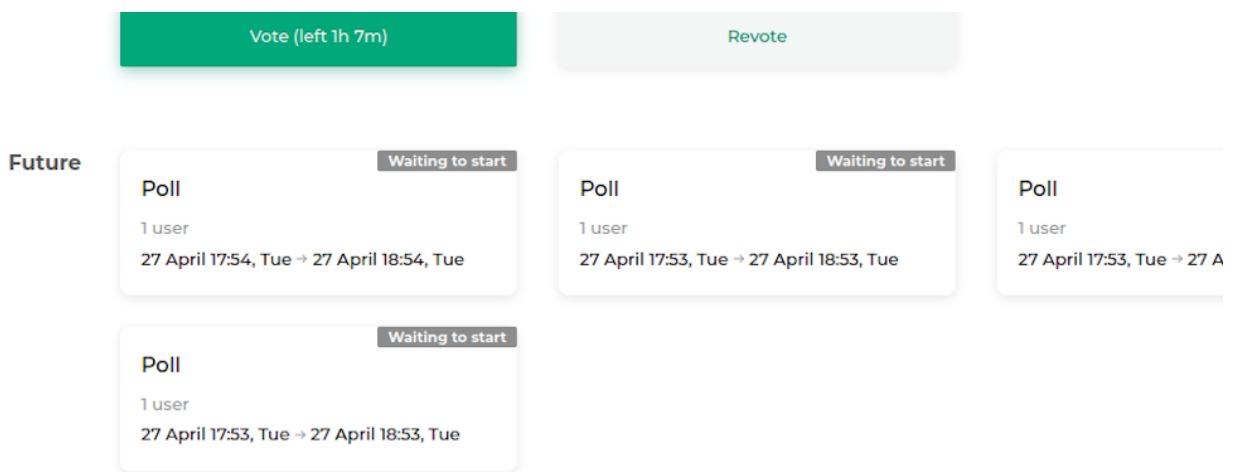
7.2.6 Publishing voting

To publish your vote, click the **Publish** button on the right side of the screen. To save the voting preferences you set without publishing it, click **Save Draft**.

Attention: Publishing is an irreversible action. After publishing, it will be impossible to change voting parameters.

From the dropdown menu, you can **Duplicate** the vote to create a copy of its draft, as well as **Remove** it.

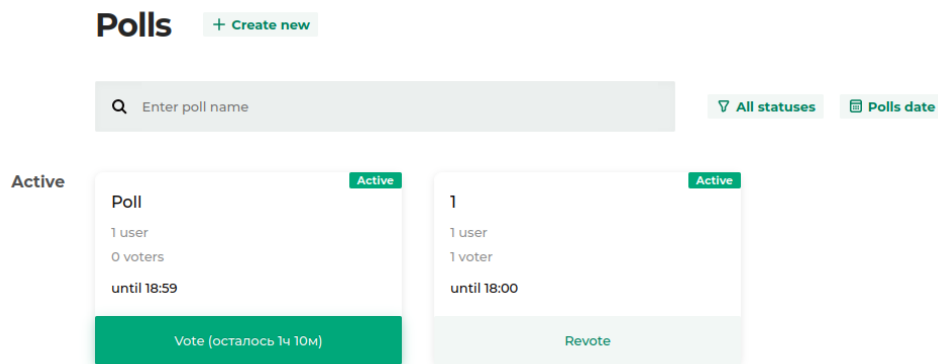
7.2.7 Launch of a voting and work with its results



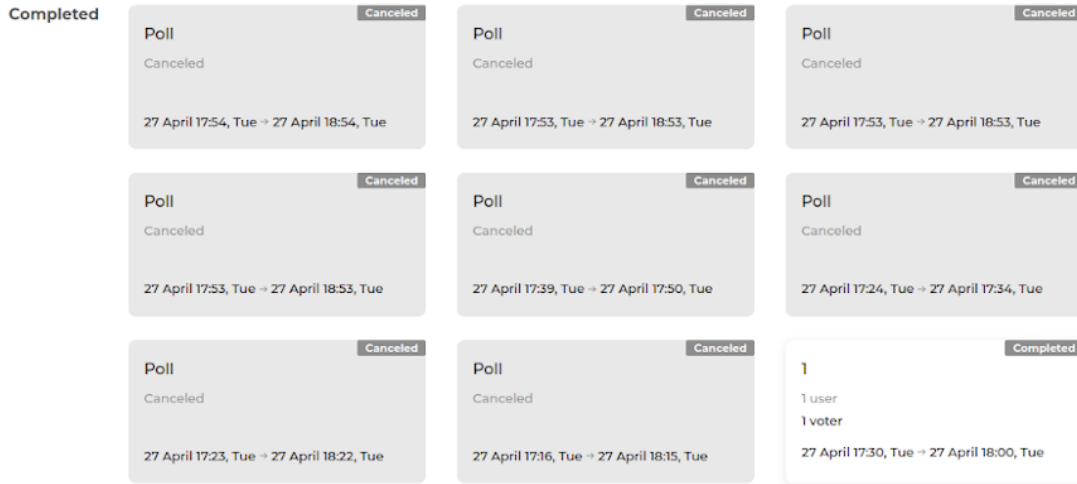
Publish a voting you have created. It will not be possible to change its parameters after publishing.

Until the start time, the published voting will be in the **Future** section on the main service page. Participants will be able to read the questions and additional materials, but will not be able to vote before it starts.

Voting will start automatically at the time you set. Once it starts, it will move to **Active** on the main service page.



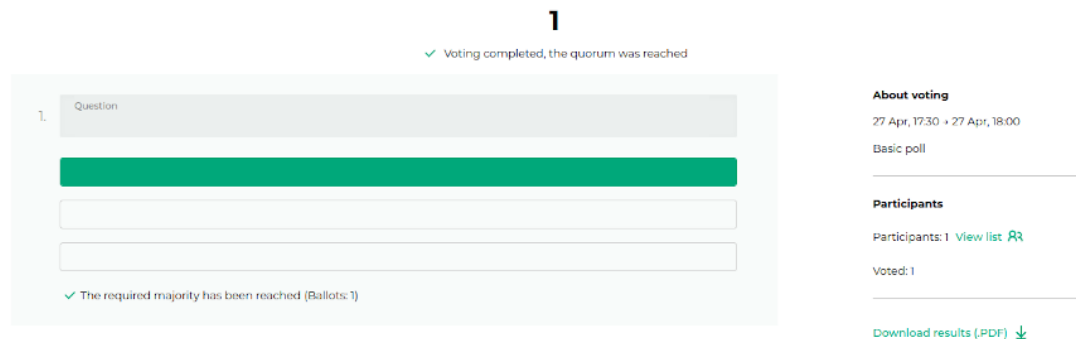
When the vote is complete, it goes to **Completed** on the main service page.



If voting for any reason will not take place (lack of quorum, absence of participants or service failure), the vote is canceled and also goes to **Completed** with the reason for cancellation.

By clicking on the completed voting card, you can view your answer choices and the parameters of the vote and download a PDF report of the vote. If open voting is selected, you can also view the answers of all voters.

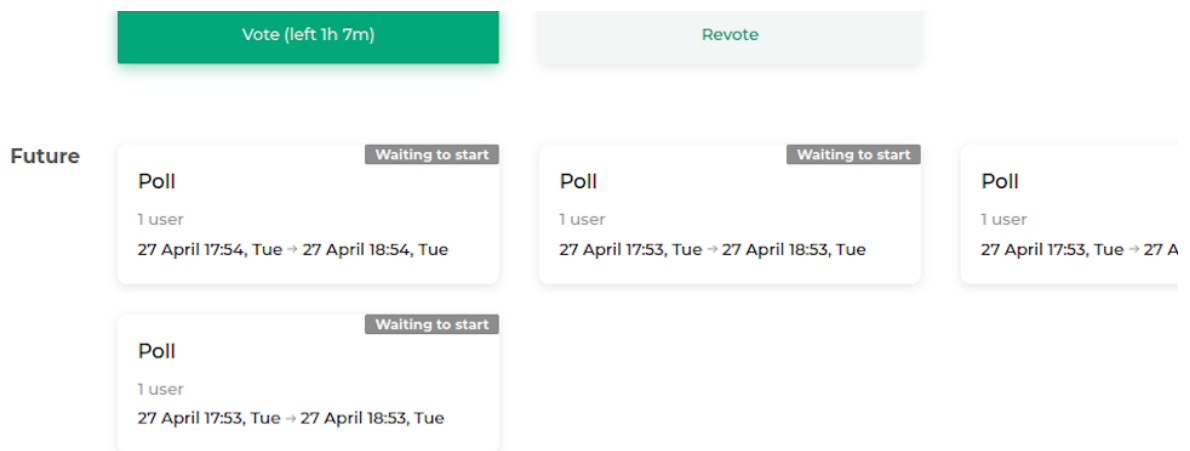
As an Administrator, you can duplicate a vote to restart it.



PARTICIPANT GUIDE

As a participant, you can register in the WE.Vote service at the invitation of the administrator. For more information on the registration process, see the **Account registration** section.

8.1 Participation in a voting



In the main window of the service, you will see all polls available for your participation:

- **Future** polls planned and created by the administrator.
- **Active** polls taking place currently.
- **Finished** polls that have been finished upon expiry of voting time, as well as cancelled polls.

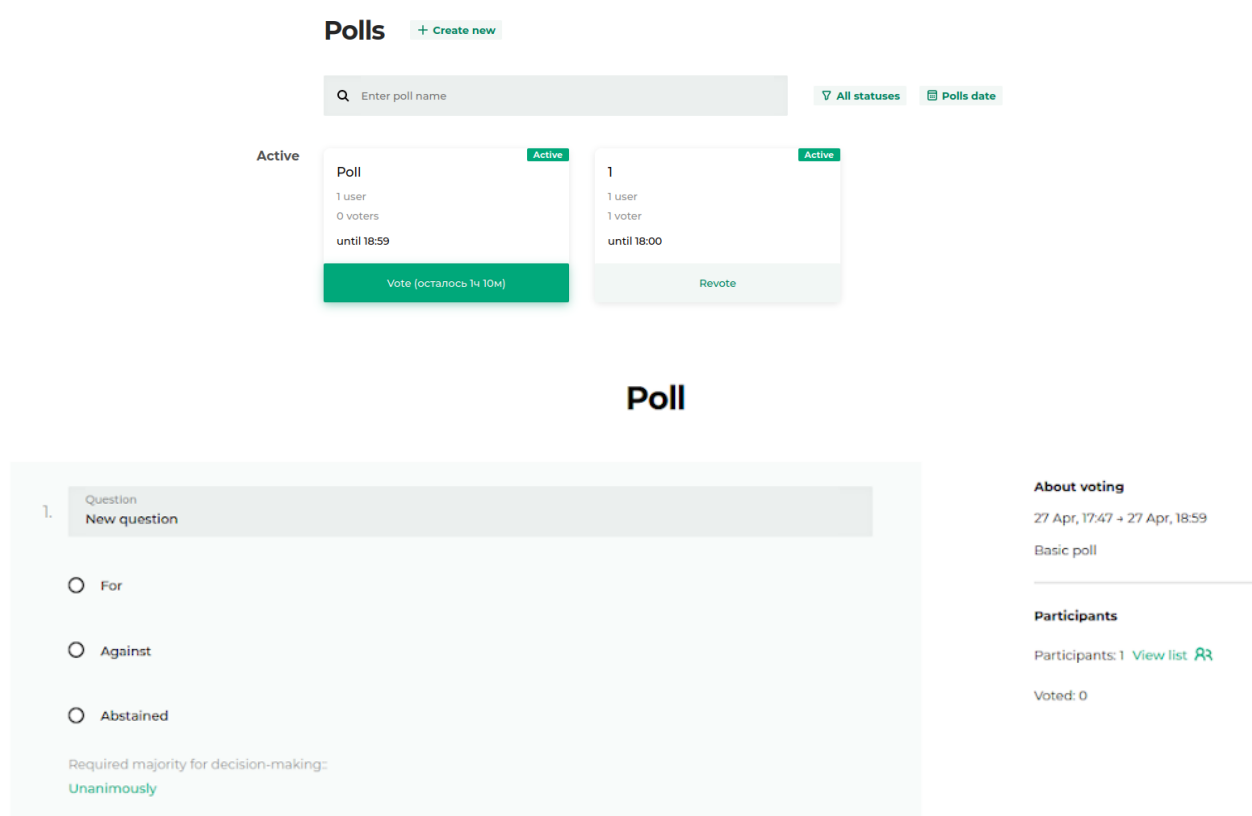
Up to the starting time, the published voting will be in the **Future** section. You can look through its agenda and additional materials, but will not be able to vote before its start.

Voting will start automatically at the time you set. Once it starts, it will move to **Active** on the main service page.

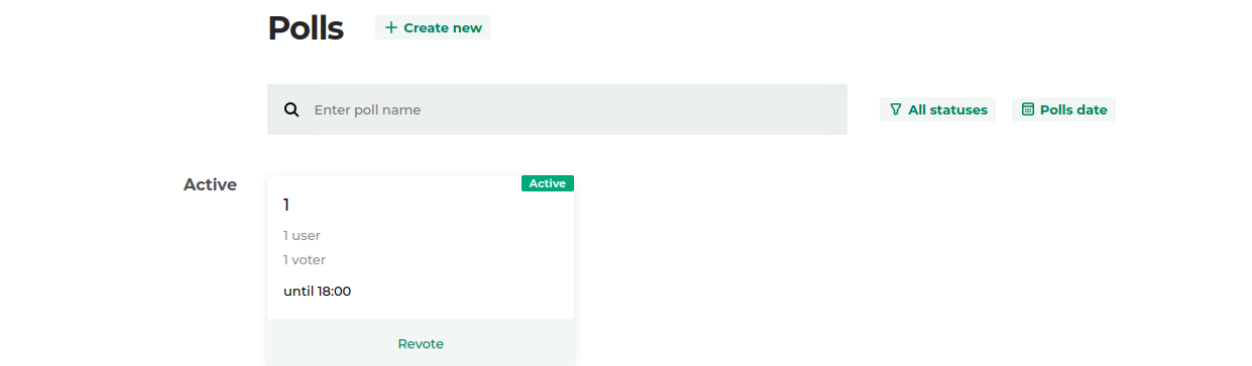
Participants can vote in an active ballot until it is complete. The voting card displays the remaining voting time.

To cast your vote, click on the active voting card and go to the list of questions. Select your answer choices for each of them and click **Vote**. If the service is inactive for a long time, accessing this button may require you to enter a password from your account in order to prevent any unauthorized access to the voting process.

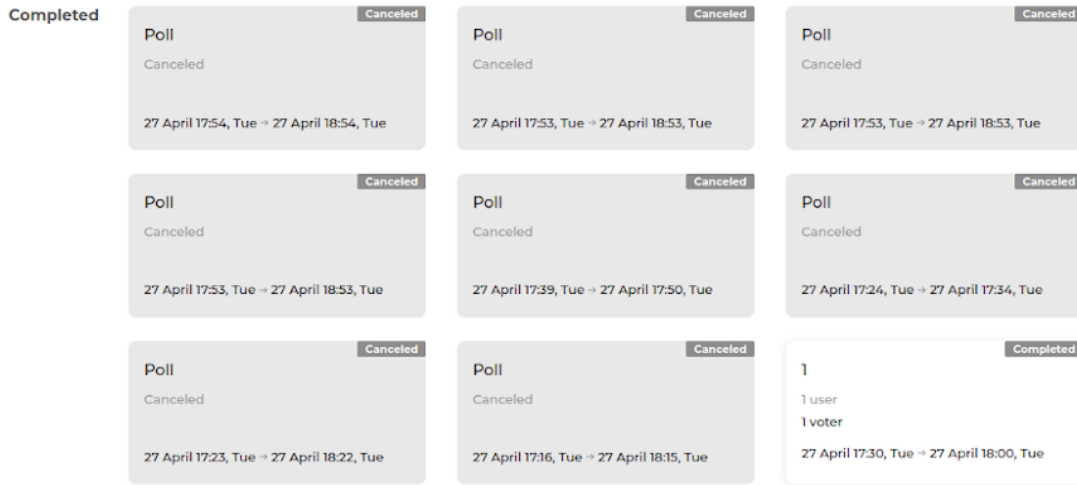
Your answer choices will then be encrypted and sent to the database.



While voting is active, you can change your answer choices. To do this, click **Revote** on the voting card, change your answer choices, and vote again.

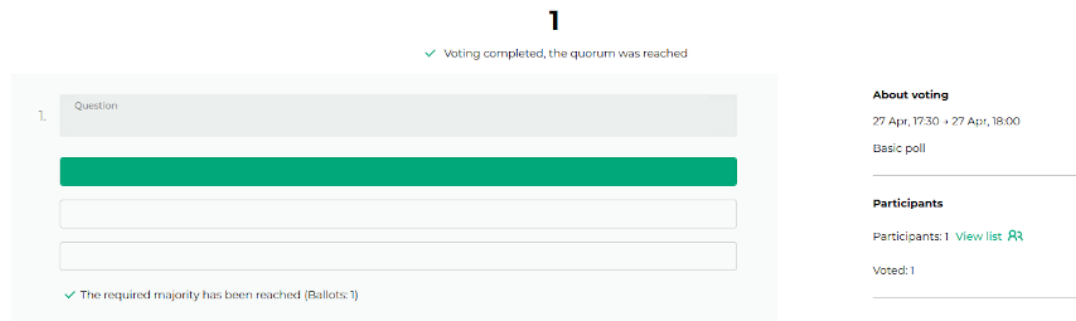


8.1.1 Work with results



Upon completion, the vote goes to the **Completed** section on the main page of the service. If the vote does not take place for any reason, the vote is cancelled and also goes into the **Completed** section with an indication of the reason for the cancellation.

By clicking on the completed voting card, you can view your answer choices and the parameters of the vote and download a PDF report of the vote. If open voting is selected, you can also view the answers of all voters.



FIXING ISSUES

9.1 I cannot publish my voting

If any inconsistency is detected in the voting parameters you have set, you will not be able to publish your voting.

- Check the date and time of voting start. Start time should be later than the current time, because the service needs a couple of minutes to create a voting smart contract.
- Check the number of ballots in the account. If the number of participants is greater than the number of available ballots, you will not be able to publish the ballot.

9.2 I cannot vote in an active voting

If you lose your seed phrase and regain access to your account, you will not be able to participate in polls that were active at the time your account was regained.

9.3 Voting ended with an error

Below is a list of situations that can cause voting to end in error. The specific reason for the error is indicated on the completed voting card.

- The specified quorum was not reached. Revise the voting parameters and change the quorum, or inform participants of the required quorum. Then recreate and rerun the vote by duplicating it.
- No registered participants have voted. Recreate the vote by duplicating it, and make sure that all participants in your vote have been informed and have registered.
- No invited participants have registered to vote. Recreate the vote by duplicating it and make sure that all participants in your vote have been informed about it and have registered. The registration status of participants is available in the **Participants** section of the client.
- Technical problems with the WE.Vote service. Contact Waves Enterprise Technical Support.

ARCHITECTURE

The WE.Vote system is based on the Waves Enterprise blockchain platform and consists of several servers deployed in a blockchain network.

The system can be deployed in two variants:

- in the form of several servers on the Waves Enterprise Mainnet;
- in the form of a private blockchain network consisting of the organization's servers.

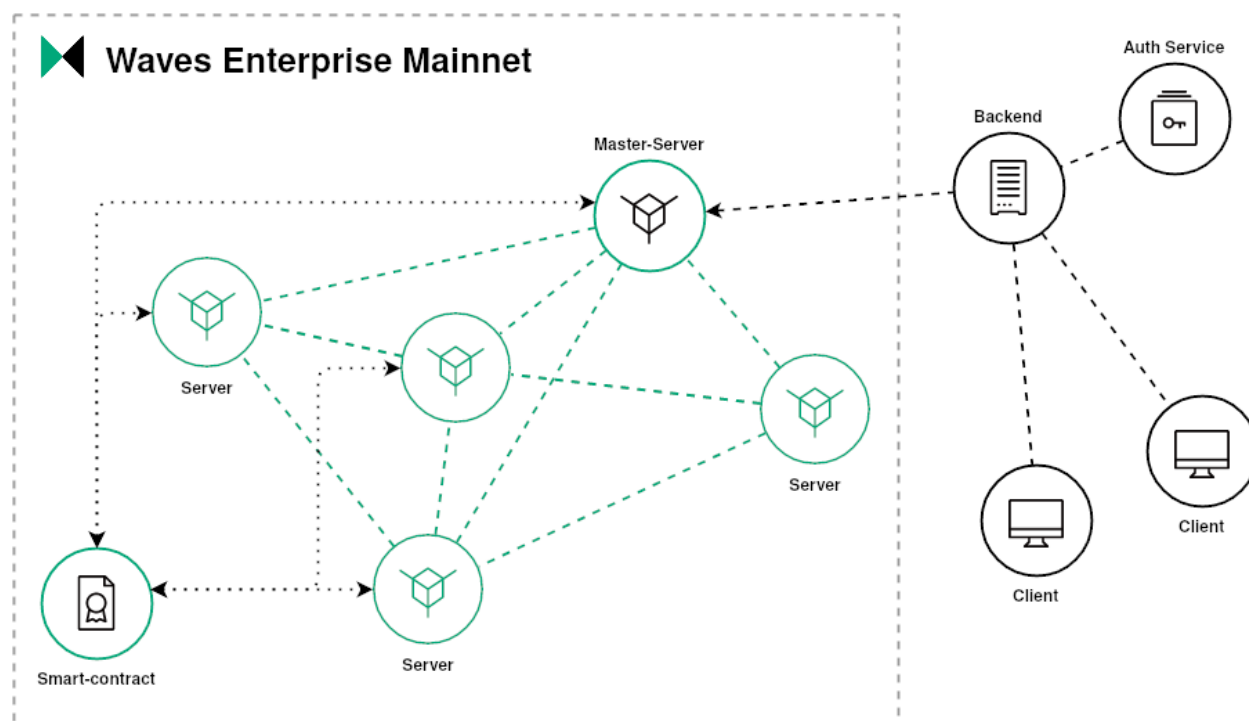


Fig. 1: WE.Vote architecture

Basic components of the system:

1. **Server** a system node consisting of the following elements:
 - **Node** a blockchain network node that processes transactions, forms blocks and implements the consensus algorithm.
 - **Cryptographic service** a service that participates in the process of distributed main key generation and produces partial decryption of voting results.

2. **Master server** the main node of the system, which, in addition to the functions of the server, carries out the functioning of the system as a whole:
 - creation of new polls;
 - access monitoring of cryptographic services;
 - formation of the voting main public key;
 - publishing voting results.
3. **Online voting smart contract** is a blockchain application that performs the following functions:
 - storage of voting rules and lists of participants;
 - registration of public data obtained during distributed key generation;
 - verification and storage of sent votes and voting results.
4. **Backend** the server side of the system, which:
 - handles client requests;
 - interacts with the master server;
 - keeps confidential data related to voting.
5. **Client** the client part of the system, consisting of the following components:
 - **Client application** a web application that provides user interaction with the service.
 - **Encryption service** a service that encrypts the filledin ballot on the public part of the master key.

CRYPTOGRAPHIC ALGORITHMS

To ensure the confidentiality of transmitted and processed data, the service uses a set of modern cryptographic algorithms.

The general view of the operation of the cryptographic algorithms of the service is shown in the diagram:

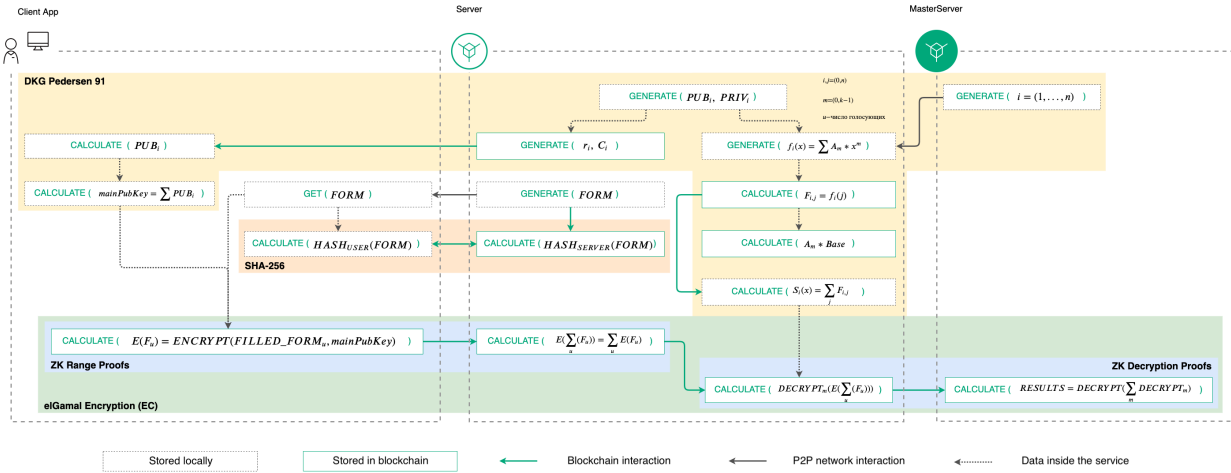


Fig. 1: Operation procedures of cryptographic algorithms

11.1 Generation of keys

The interaction of the system components is based on the **MPC (MultiParty Computation)** cryptographic protocol, which is used to generate key pairs.

It allows several participants to independently perform cryptographic calculations based on their own secret data, without having information about each other's secret data. During the computation process, the participants do not exchange secret data, independently generating a set of private keys for the common public public key. This method eliminates a single point of failure: the assembled key pair does not exist on any of the participating servers.

The MPC protocol utilizes the 'K out of N' principle corresponding with the Shamir's Secret Sharing scheme:

- decryption does not require all **N** parties involved in the encryption process: decryption can be done using a smaller threshold number of **K** parties;
- at that, **K - 1** and less parties have no way to decipher the data.

This principle allows the service to work even if several servers in the system fail, while providing a high degree of data protection. Each MPC member server generates its own public and private keys, as well as a **general public key (MainPublicKey)**, exchanging unclassified data about its computations with other members via blockchain transactions.

To generate a public key, the system uses the **algorithm of Distributed Key Generation (DKG)** made by [Torben Pedersen](#) (**DKG Pedersen 91**), transferred to elliptic curves **secp256k1** and **P256**. The generation process involves cryptographic operation services of the system servers, communicating with each other via transactions to their own nodes. For each new vote, the process of generating a new shared public key is started.

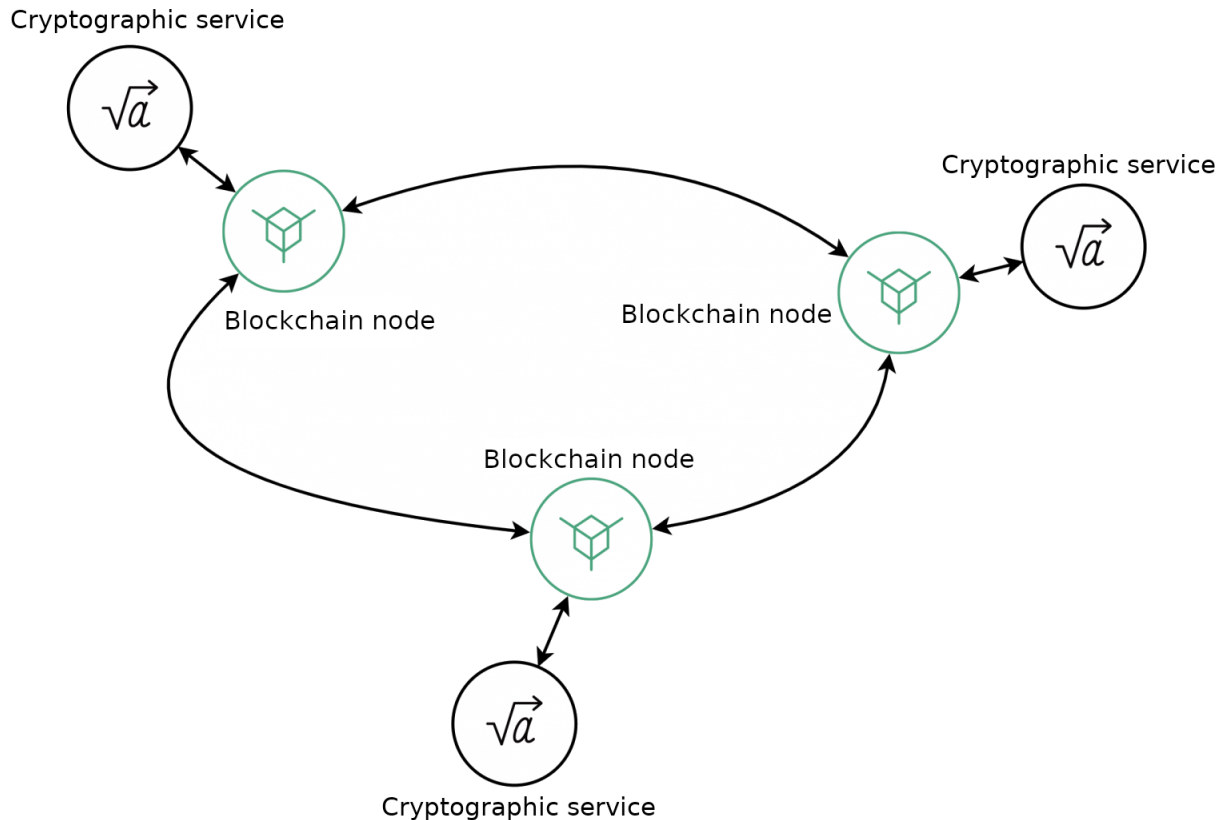


Fig. 2: DKG algorithm scheme

The process of **DKG Pedersen 91** key generation is divided into following steps:

1. After publishing a new vote, the master server polls available cryptographic services of participating servers. Based on the received data, it forms a list of **N** services and assigns each of them an ordinal number from **1** to **N**.
2. Each cryptographic service generates public and private keys for voting.
3. Each cryptographic service publishes a **Pedersen commit** and a corresponding scalar (C_i and r_i) to the blockchain. The commitment and the scalar are published for voting participants.
4. After receiving C_i and r_i , each voter (client encryption service) computes the public keys of each cryptographic service.
5. Based on the public keys, the voting members calculate a common voting key.

To realize the **K out of N** Shamir's Secret Sharing scheme, cryptographic services perform following algorithm:

1. Each cryptographic service randomly generates a $f_i(x)^{K-1}$ polynomial.
2. Each cryptographic service calculates polynomial values for N other servers according to their ordinal numbers: so called "shadows" $f_i(j)$.
3. Cryptographic services publish the calculated "shadows" and polynomial coefficient values for all other servers to the blockchain.
4. Cryptographic services check the correctness of published "shadows" and calculate the private key needed for decryption.

This process allows to restore the encrypted results of the vote, even if some of the servers are unavailable.

11.2 Encryption

Ballots are not transferred in an open format within the system. To encrypt them on the client side, the **El Gamal cryptosystem** based on elliptic curves is used. This cryptosystem implements the **homomorphic encryption method with respect to addition**: as a result of the addition operation on the ciphertext, the encryption service generates an encrypted sum of the original values.

$$ENCRYPTED(1) + ENCRYPTED(1) = ENCRYPTED(2)$$

To implement this encryption method, each ballot is represented as a matrix, where each row is a separate question, and the row cells are the answer choices to the question. Each of the cells is initially represented as zero, and the answer choices chosen by the participant change the value of the corresponding cell to one. Additionally, each cell of the completed ballot is encrypted and then published to the blockchain.

Not filled					Filled					Encrypted				
Participant 1					Participant 1					Participant 1				
	Variant 1	Variant 2	Variant 3	Variant 4		Variant 1	Variant 2	Variant 3	Variant 4		Variant 1	Variant 2	Variant 3	Variant 4
Q1	0	0	0	0	Q1	0	0	1	0	Q1	e(0)	e(0)	e(1)	e(0)
Q2	0	0			Q2	1	0			Q2	e(1)	e(0)		
Q3	0	0	0		Q3	0	1	0		Q3	e(0)	e(1)	e(0)	
Q4	0	0			Q4	0	1			Q4	e(0)	e(1)		

Fig. 3: Ballot encryption procedure

Then, using homomorphic encryption, the encrypted cells corresponding to each answer choice are summed separately by each server in the system:

As a result, each of the servers in the system independently receives the voting results without revealing the voting results of participants.

This process solves the following problems faced by online voting:

- A participant's voice cannot be faked: it is not transmitted openly and is not even deciphered.
- A participant cannot be forced to vote for one or another option: in addition to complete anonymization of the voting results, the participant has the ability to change their choice during the vote. When counting, the system will only count the last ballot sent on behalf of the participant's public key.

Q1	Variant 1	Variant 2	Variant 3	Variant 4	Q2	Variant 1	Variant 2
Participant 1	e(0)	e(0)	e(1)	e(0)	Participant 1	e(0)	e(1)
Participant 2	e(1)	e(0)	e(0)	e(0)	Participant 2	e(1)	e(0)
Participant 3	e(0)	e(1)	e(0)	e(0)	Participant 3	e(0)	e(1)
Participant 4	e(0)	e(1)	e(0)	e(0)	Participant 4	e(0)	e(1)
Total	e(1)	e(2)	e(1)	e(0)	Total	e(1)	e(3)

Fig. 4: Addition of encrypted answers

11.3 Zero Knowledge Proofs (ZKP)

The El Gamal cryptosystem avoids compromising of voting results by the organizer or an external intruder. However, it does not protect the ballot from being compromised by the voter himself: since an algorithm for adding up the encrypted ballot results is applied, the voter can change the data on the side of his client application. By making changes to the answer choice and sending the “valid” ballot to the system for further encryption and addition, he can achieve the desired number of votes for a planned variant.

Therefore, in addition to encrypted ballots and closed vote counts, the **Zero Knowledge Proofs (ZKP)** technique is used to prove the integrity of the ballots. This technique allows you to prove possession of information without disclosing it including the correctness of an encrypted value without disclosing it.

In general, the ZKP principle can be demonstrated by the ‘Ali Baba Cave’ illustration:

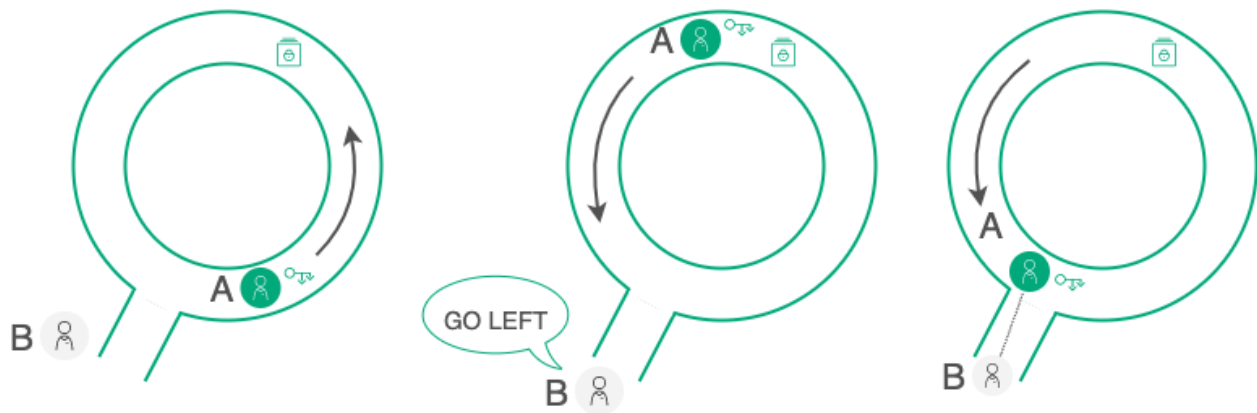


Fig. 5: ZKP demonstration

An **A** participant has a key to open a door in the labyrinth and wants to prove this for a **B** participant without revealing the key itself. For **B** to be able to verify the correctness of the **A** statement, they organize a series of tests:

1. **A** goes into the labyrinth while **B** turns his back. **B** doesn't know which way **A** went.
2. **B** gives **A** an instruction to come out on either side, for example on the left.
3. If **A** really has a key, it can appear on either side and execute the instructions of **B**.

The chance that **A** just got lucky and, not having the key, initially went left is 50/50. So they repeat the test

several times until the probability of “luck” is negligible: as a result, **B** will have sufficient evidence that **A** does possess the key. In doing so, **B** will not see the key itself, nor will he get any of the information that **A** possesses (the direction that **A** chooses in each trial) but, through a series of trials, he will get probabilistic proof, with any accuracy necessary.

As applied to the WE.Vote voting process, NonInteractive ZeroKnowledge Proofs (NIZK)** are used for interaction between participants.

The proof process itself is divided into two algorithms:

1. ZeroKnowledge Range Proofs

This proof is used when the encrypted ballot is published, before the votes are counted.

Majority						Multiple-choice						Weighted (weight of one vote equals to 15)					
Question	Variant 1	Variant 2	Variant 3	Variant 4	Question sum	Вопрос	Variant 1	Variant 2	Variant 3	Variant 4	Question sum	Вопрос	Variant 1	Variant 2	Variant 3	Variant 4	Question sum
Answers	e(0)	e(0)	e(1)	e(0)		Ответы	e(0)	e(1)	e(1)	e(0)		Ответы	e(0)	e(0)	e(15)	e(0)	
ZKP	[0,1]	[0,1]	[0,1]	[0,1]	[1]	ZKP	[0,1,2,3]	[0,1,2,3]	[0,1,2,3]	[0,1,2,3]	[1,2,3]	ZKP	[0,15]	[0,15]	[0,15]	[0,15]	[15]

Fig. 6: ZeroKnowledge Range Proof scheme

For **majority voting**, the proof process is following:

1. A NIZK is added to each cell, proving that the cell contains one of two possible values in an encrypted format: **0** or **1**. At that, the value itself is not revealed.
2. Additionally, each cell contains a proof that the sum of all encrypted cells related to the question is **1**.

This will prove, that a participant can set the **1** value in any cell in the row (choose any answer).

For **multiplechoice voting**, where a participant can choose multiple cells of the row, the process is more complex:

1. A NIZK for the $[0, 1 \dots N]$ array is added for each filled and encrypted cell: a participant can choose all **N** answers, as well as some of them.
2. Proof of the sum of the cells for a single question can be applied to the range $[1 \dots N]$ (the participant can choose from **1** to **N** options, but cannot leave the question unanswered), or to the value **N** (the participant distributes the available votes between the answer choices).

For **weighted voting**, where each participant casts a number of votes equal to their weight in the system (e.g., proportional to their ownership interest in the company), the proof process is as follows:

1. Each of the filled and encrypted cells contains a NIZK, proving that the cell contains one of the values in an encrypted form: **0** or **N**, where **N** is weight of the participant.
2. As proof of the sum of the cells, the **N** value is attached, since in weighted voting the participant can put the **N** value in one cell in the row, just like in majority voting.

The system only accepts ballots whose ZKPs have passed the authenticity checks in full. That is, an attacker can distort the data in his ballot by taking advantage of the fact that the system does not disclose its contents however, in this case, the ballot will still fail validation.

2. ZeroKnowledge Decryption Proofs

This proof is used when each service publishes cryptographic operations of the results of the preliminary decryption of the voting results. Since the services operate independently of each other, at this stage there is a danger of data spoofing by one or more individual services. Decryption proofs make it possible to verify that each service decrypted and published exactly the results of performed voting.

To do this, each of the services appends a proof to its decryption result, using the ZKP ChaumPedersen algorithm. This algorithm proves the knownness of the number X for two ratios:

- $A = X * B$;
- $C = X * D$.

Here, A , B , C and D are points of one curve.

Due to the attached proof, each qualified spectator can independently perform following operations:

- perform homomorphic addition of valid ballots and get the totals of the voting results for comparison;
- check the proof of decryption totals from each individual cryptographic service;
- conduct a final transcript of the voting results using the public data placed on the blockchain during voting.

Decryption proof eliminates the possibility of data spoofing if one or more of the system's servers are compromised.

HOW WE.VOTE WORKS

The voting process in WE.Vote is divided into four main stages. This section describes how the system elements work for each stage. User scenarios are given in the **How to use the service** section.

12.1 1. Creation of a voting and inviting prticipants

Administrator creates voting in the WE.Vote client. The voting contains following data:

- voting decryption;
- voting date and time;
- list of questions and answers;
- list of voting participants;
- additional documents for voting participants.

Voting date and time are transferred into the blockchain in an open format.

Voting description, list of questions and answers, as well as additional documents are hashed up, their hash sums are also published in the blockchain. This process allows to prevent data forging:

- with data change, its hash sum is also changed;
- data forging is detected on the participant side by comparison of data with its hash sum published in the blockchain.

The public keys of invited participants are also sent to the blockchain when voting is created. At the same time, the personal and contact details of the accounts are not transmitted to the network: during voting, the transaction is signed with the participant's private key, and the public key becomes his or her only public identifier.

The raw data published in the blockchain in the form of hash sums are also not transmitted over the network for privacy reasons: they are stored in the local server database and are only available to voting participants upon successful authentication. This makes it possible to reliably limit access not only to the results of the vote, but also to its agenda.

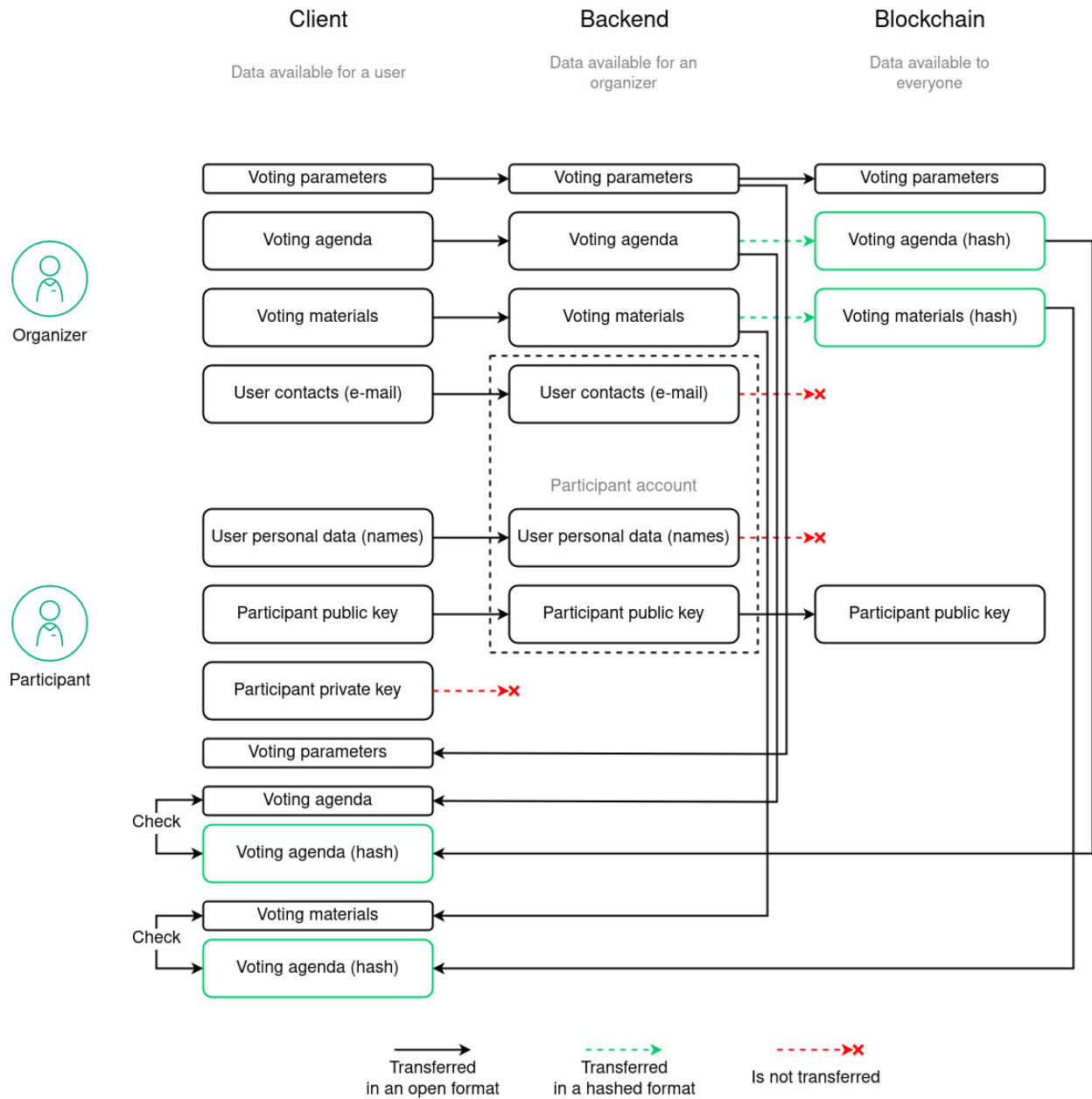


Fig. 1: Voting process

12.2 2. Generating and publishing a voting MainPublicKey

The master server receives data from the blockchain about the preparation of a new vote, and then determines the list of available cryptographic services on the servers. To each of them, the master server sends a request to create a key pair for voting. In response to the request, each cryptographic service returns the public part of the created key (see section *Key generation*).

The master server generates a MainPublicKey from the received public keys and publishes it in the blockchain to the voting address. The shared public key is then used by participant encryption services to encrypt completed ballots.

After the formation and publication of a MainPublicKey, the system receives all the necessary data to initiate voting:

- the original voting data are stored locally in the database of each server;
- hash sums of the data are published in the blockchain and serve to control the integrity of voting materials;
- the MainPublicKey needed to encrypt the votes of participants is also published in the blockchain.

In this case, none of the participants has any common private key, with which it would be possible to decrypt the ballots. Accordingly, neither the organizer of voting, nor the participants themselves have a possibility to tamper with the data in the ballots, and the votes are counted by homomorphic addition of the encrypted results with preliminary and subsequent data integrity control (see **Encryption** and **Zero Knowledge Proofs**).

12.3 3. Conduct of voting

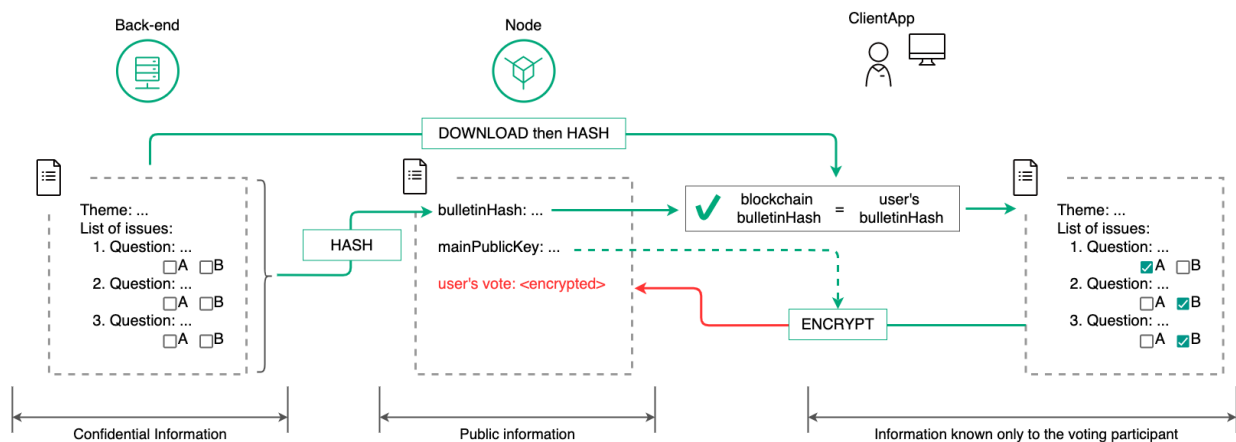


Fig. 2: Voting process

After the public key is generated and published on the blockchain, the system waits for the preset voting start time. Each of the voting participants invited by the administrator receives an email notification of the vote.

After registering and gaining access to their account (see section *Account registration*), the participant can read the questions and voting materials. In doing so, the participant's client application requests the server to save the voting data. The request is signed with the corresponding key given to the participant during registration.

Upon receiving the voting materials from the server, the participant's client application calculates their hash sums and compares them to the sums stored in the blockchain during the voting creation phase. This ensures the participant's data integrity.

When voting begins, the participant is given the opportunity to respond to the questions on the agenda:

1. the participant fills out a ballot;
2. the client application encrypts the completed bulletin using a shared public key;
3. the client application encrypts the completed ballot using the participant's public key;

The data published on the blockchain is publicly available. Therefore, the transaction of publishing an encrypted ballot is available to all voters. However, none of them can view the published responses because they do not have a private key equal to the MainPublicKey of the voting.

When publishing an encrypted bulletin, the cryptographic service checks that the set ranges are correct, without decrypting the ballot data (see section *ZeroKnowledge Range Proofs*). If data spoofing is detected, the transaction is rejected.

During the voting process, each participant has the ability to change their answer choices. At the same time, the client application encrypts and publishes the participant's new ballot on the blockchain using a repeated transaction. Only the last encrypted ballot published by a participant is accepted for counting.

After the voting period, which was defined by the administrator when it was created, any transactions from voting participants are rejected by the system.

12.4 4. Summing up the voting results

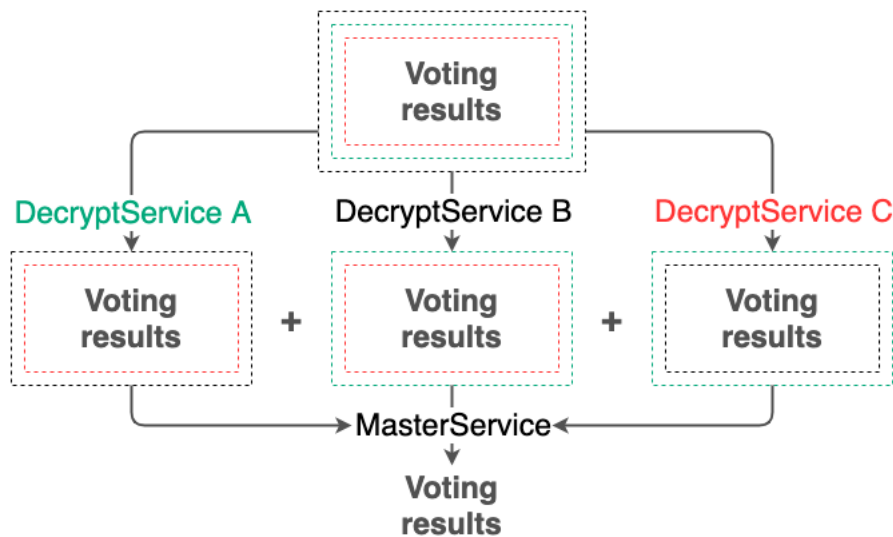


Fig. 3: Final counting of the results

After voting is completed, all servers in the system independently count the votes. The votes are counted without decrypting the voting results, since no cryptographic service has a common private key for decryption.

The voting results for each of the questions are counted by homomorphic addition of the encrypted values (see *Encryption*). As a result, each cryptographic service independently obtains the encrypted value of the sum of participants' votes for each of the questions.

Then, each cryptographic service proceeds to the preliminary decryption of the voting results. For that, the service uses its own private key, which was used at step **2** to generate the common public key. As a result of the decryption, each service gets an unreadable preliminary voting result.

At this stage, the integrity of the decryption results is checked using the **ZKP ChaumPedersen** algorithm (see section *ZeroKnowledge Decryption Proofs*). This ensures that the voting data has not been compromised by any of the cryptographic services.

Predecrypted voting results from each service are published to the blockchain via transactions to the corresponding nodes.

The master server collects the preliminary decryption results published on the blockchain. The cryptographic service of the master server then summarizes the obtained preliminary voting results and finally decrypts them. The final voting results are published to the blockchain.

At the same time, in case one or several servers fail, the master server is able to collect and decrypt the final results of the voting, thanks to the Shamir secret sharing scheme. If **N** servers participated in the formation of the common public key, it is enough to collect **K** < **N** predecrypted voting results to decrypt the results.

FREQUETLY ASKED QUESTIONS

13.1 What is WE.Vote?

It is a blockchainbased platform that allows to organize and conduct proxy voting for any organization with a collective decisionmaking mechanism.

13.2 What differences exist between a blockchainbased voting and traditional voting methods?

- **Decentralization of the voting procedure.** This means that all voting actions are stored simultaneously on the computers of all participants. In order to disrupt the voting or tamper with the results, all the computers involved in the voting must be hacked.
- **Voting transparency.** Distributed consensus allows to provide a single version of voting results agreed by all participants.
- **Confidentiality of voting results.** Application of cryptographic algorithms for encryption of results allows to exclude fraud with voting results.

13.3 Is voting anonymous?

Yes. Moreover, you can customize the visibility of participants during the poll creation process. Depending on the settings you choose, participants may or may not see each other's email addresses.

13.4 How is a blockchainbased voting arranged?

In general, blockchain voting is similar to conventional voting: the same concepts, approaches and processes apply. When you create a vote, you operate on commonly known parameters, and when you participate in a vote, you simply answer the questions on the agenda in the client application.

The difference lies in the format of working with the data: all voting data is not stored centrally on any server, but is decentralized to the computers of voting participants. Along with the use of cryptographic algorithms, this makes voting as transparent and secure as possible.

13.5 How is the voting procedure performed?

The voting procedure consists of three stages:

1. Creation of a voting:

- The administrator determines the agenda and voting parameters;
- The system generates a MainPublicKey for voting.

2. Performing of a voting:

- The participant answers the questions on the agenda;
- The participant answers the questions on the agenda;

3. Summing up the voting results:

- The system encrypts the participant's ballot. The system summarizes the voting results by summing up the encrypted answers in each participant's ballot;
- The system decodes the results and displays them for you to see.

13.6 Can a user change answers during voting?

Yes. You can change your answer choices as many times as you like before voting ends. The system will accept the last option on your ballot for processing.

13.7 When will results be available?

As a rule, the results are available 510 minutes after the end of voting: about how long it takes the WE.Vote system to calculate and decrypt the results.

13.8 Are voting results encrypted?

Yes. The system encrypts both your ballot and the results of your vote.

13.9 Can a voting itself be changed after its start?

No. Once a vote is started, no one can make any changes to its parameters or agenda. As a participant, you can only change the answer choices you have chosen for the agenda.

13.10 Can voting results be faked?

It is practically impossible. Information about voting parameters and results is transmitted, stored and processed only in encrypted form. Blockchain technology is used both to store data and to control its integrity during transmission by means of hash sums. To tamper with or alter the results of a blockchain vote, at least 51% of participants' computers would have to be hacked and the data tampered with during the voting process itself. However, it would be impossible to decrypt the results: the compromised data would not pass integrity checks.

GLOSSARY

Blockchain

A decentralized, distributed and public digital ledger that is used to record in such way that any involved record cannot be altered retroactively, without the alteration of all subsequent blocks

Homomorphic encryption

A form of encryption that allows you to perform certain mathematical actions with encrypted text and get an encrypted result that corresponds to the result of operations performed with open text

Cryptographic data integrity protection

A security mechanism through data encryption for secure storage and protection of information from unwanted users.

Consensus

The way of getting an agreed result by a group of participants

Mainnet

Main network, where the main functions are performed in the form of transaction receipt and transfer, release and storage of tokens

Node

Network blockchain node, processing transactions, forming blocks and implementing the consensus algorithm

Private key

A string combination of characters for signing transactions and accessing tokens, stored privately. The private key is inextricably linked to the public key

Public key

String combination of characters, inextricably linked to the private key. The public key is attached to transactions to confirm the correctness of the user's signature made on the private key

Server

The main system node containing the Node, DecryptService, CryptoProvider, Backend + Database, API

Smart contract

Computer algorithm designed to form, control and provide information about the agreement between the participants

Transaction

Transmission of «facts» produced by members in the network to initiate any action

Participant

A blockchain participant who send transactions to the net for getting approve

Hash

A unique configuration of the symbols (letters and digits), it is a result of the hash function performing over the data according with the specified algorithm. Hash uniquely identifies the object

DKG (Distributed Key Generation)

A cryptographic process in which multiple parties participate in the calculation of a common set of public and private keys

MPC protocol (MultiParty Computation)

A cryptographic protocol that allows multiple participants to perform a calculation based on every participant's secret input data so that no participant can get any information about someone else's secret input data

ZeroKnowledge Range Proofs

An interactive cryptographic protocol that allows one of the interacting parties ("The verifier") to verify the validity of any statement (usually mathematical) without any other information from the other party ("The prover")

OFFICIAL RESOURCES

- Official site of the blockchainplatform [Waves Enterprise](#)
- [Github](#) project
- Official site of the blockchainplatform [Waves](#)