D-voting - Front-end development for an e-voting platform

Capucine Berger - Badr Larhdir

Semester project
Final presentation

DEDIS Lab - Prof. Bryan Ford
Supervisor: Noémien Kocher

EPFL - 27.06.22
Content

1. Introduction to the platform
2. Current status of the front-end
3. Features and implementations
4. Improvements and future work
5. Demo
Introduction to the platform

- The D-voting platform
- Front-end architecture
- Previous status of the project (as of February)
The D-voting platform

- Voting platform based on the E-voting system on DELA blockchain

- Two backends, one that implements all the blockchain logic, the other to authenticate and sign all the communication between the front-end and the proxies

- A front-end to interact with the whole system (voting, monitor an election, managing user roles & nodes etc.)
Front-end architecture
[February] Compatibility issue
<table>
<thead>
<tr>
<th>Title</th>
<th>Status</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC class representatives</td>
<td>Open</td>
<td></td>
</tr>
<tr>
<td>SV class representatives</td>
<td>Results available</td>
<td></td>
</tr>
<tr>
<td>MA class representatives</td>
<td>Ballots shuffled</td>
<td></td>
</tr>
<tr>
<td>EL class representatives</td>
<td>Closed</td>
<td></td>
</tr>
<tr>
<td>AR class representatives</td>
<td>Canceled</td>
<td></td>
</tr>
</tbody>
</table>
Create a new election by filling out the information below or by uploading a json file

Option 1

Election title:

Add a candidate:

- Max
- Mike
- Dave

Option 2

Choose a json file from your computer:

Browse... no file selected.

Create election

ADD A USER

<table>
<thead>
<tr>
<th>login</th>
<th>role</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>23123</td>
<td>operator</td>
<td>DELETE</td>
</tr>
<tr>
<td>111112</td>
<td>operator</td>
<td>DELETE</td>
</tr>
<tr>
<td>123567</td>
<td>admin</td>
<td>DELETE</td>
</tr>
<tr>
<td>295747</td>
<td>admin</td>
<td>DELETE</td>
</tr>
</tbody>
</table>

© 2021 DEDIS LAB - https://github.com/dedis/dela
Current status of the front-end

- Update of the transactions
- Update of the API calls
Election flow

Status of API transactions:

- Create election form (new v2 !)
- Initialize and Setup the nodes
- Cast a ballot
- Compute & Get results
Update of the API Calls

**DK2: DKG setup**

<table>
<thead>
<tr>
<th>URL</th>
<th>/evoting/services/dkg/actors/{ElectionID}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>PUT</td>
</tr>
<tr>
<td>Input</td>
<td>application/json</td>
</tr>
</tbody>
</table>

Return:

200 OK  text/plain

```json
{
  "Action": "setup",
  "Proxy": ""
}
```

**DK3: DKG get info**

<table>
<thead>
<tr>
<th>URL</th>
<th>/evoting/services/dkg/actors/{ElectionID}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>GET</td>
</tr>
<tr>
<td>Input</td>
<td></td>
</tr>
</tbody>
</table>

Return:

200 OK  application/json

```json
{
  "Status": "<int>",
  "Error": {
    "Title": "",
    "Code": "<uint>",
    "Message": "",
    "Args": {}
  }
}
```
 Compatibility issues are resolved
Features and implementations

- Create election
- Manage election
- Cast ballot
- Election results
- Manage user roles and nodes
- UX testing
Create election

- Complete Election form with:
  - Subjects/ Sub-subjects
  - Ranks/Selects/Texts
  - I/O Form validation

- Preview the election

- Upload and export election in JSON with validation

Modal to create a select question
Manage election

- Auditing an election (including node initialization and setup):

**Actors**
- Smart contract
- DKG service
- Neff shuffle service

**ACTION**
- Initialize nodes
- Setup
- Open
- Close
- Shuffle
- Compute public shares
- Combine shares

**STATUS**
- Created
- Node initialized
- Setup
- Opened
- Closed
- Ballot shuffled
- Ballot decrypted
- Results available
Cast vote

D-voting - front-end: development for an e-voting platform

Capucine Berger - Badr Larhdir

16
Election results

See the results of an election

Results

Total number of votes: 2

Life on the campus

Rate the course

How did you find the provided material, from 1 (bad) to 5 (excellent)?

1: 50.00%
2: 0.00%
3: 0.00%
4: 50.00%
5: 0.00%

How did you find the teaching?

bad: 50.00%
normal: 0.00%
good: 50.00%

Who were the two best TAs?

Noémien: 100.00%
Pierluca: 100.00%
Manage user roles and nodes

- Manage user roles
- Manage the nodes

Please give the sciper of the user

Sciper
admin

Add user
Cancel

Please enter the addresses of the node and the proxy

Node: 123.123...
Proxy: https://...

Add
Cancel
UX testing

- 13 persons completed the testing survey:
  - ~30-40 minutes/persons
- The website was not online yet:
  - Had to be physically present
  - Used the font-end with the mocked API
- [Process] Asked them to accomplish the following actions:
  - Login
  - Manage user roles
  - Create an election and open it
  - Vote on another election
  - Make the results of the election available
  - Fill the Google form
UX testing

(a) Admin page. (b) Election create page. (c) Vote page. (d) Election page.

Figure 4.6: Results of the question "On a scale from 1 to 5 (1=not at all, 5=very much), was the interface easy to use?", for each of the pages. All users gave at least a 3 to all pages, and most of them rated them a 5.
Improvements and future work
Improvements and Future work

- Ticket to verify if ballot was taken into account
- Support for 🇫🇷 🇩🇪: translation of the UI and the ballots
- Open election only to a subset of the users (e.g. by section)
- Facilitate further the election creation
  - Modal on Subject creation
  - Tooltips for the page
- Optimise the web-backend transactions
Demo

https://dvoting-dev.dedis.ch/