Projet de fin d'étude: Scrutin.app

Maxime Lalisse

February 17, 2025

1 Introduction

Le vote par internet est un besoin émergent ayant certains avantages par rapport au vote papier traditionnel. Notamment, il diminue drastiquement les coûts en infrastructure et permet de voter à distance. En revanche, c'est une technologie de nature radicalement différente. C'est un domaine de recherche très actif, notamment autour de sa sécurité. Du fait de sa nouveauté, son usage est strictement encadré par la loi, notamment pour les élections politiques où il est encore rarement proposé. En revanche, on y a souvent recours pour les élections non politiques, telles que les élection professionnelles ou d'association. De fait, il est très souhaitable de proposer des outils aussi fiables et sécurisés que possible.

De nombreuses entreprises s'intéressent au sujet (Voxaly, Scytl, AssemblyVoting, etc.), tout comme de nombreux pays via des projets publics et/ou de recherche (CHVote et Swiss Post en Suisse, iVoting en Estonie, etc.).

En France, le sujet est notamment traité à travers le projet Belenios (CNRS, INRIA, LORIA), ainsi que ses nombreuses variantes (Belenios-CaI, Belenios-RF, etc.). Les équipes de recherche les plus actives autour de Belenios, notamment Pesto et Caramba au LORIA, contribuent également à travers des résultats liés à la vérification formelle des protocoles de vote électronique, ainsi qu'à la formalisation de leurs propriétés de sécurité (confidentialité et vérifiabilité) et ont publié un livre sur le sujet [7].

Scrutin [4] est une plateforme et une application web et mobile libre basée sur Belenios.

2 Contexte

2.1 Belenios

Belenios est à la fois un protocole [3] basé sur Helios [1], une implémentation de ce protocole [8] composée d'un client, d'un serveur web et d'un outil en ligne de commande, ainsi qu'une plateforme de vote [9]. Les propriétés de sécurité recherchées (confidentialité et vérifiabilité) sont formalisées et prouvées au niveau du protocole.

Belenios est un dérivé d'Helios [1], offrant une vérifiabilité de bout-en-bout via deux mécanismes: la **vérifiabilité individuelle** (je peux vérifier mon vote) et la **vérifiabilité universelle** (je peux vérifier que le résultat correspond bien à l'ensemble des votes). Les votes sont anonymisés dans le processus, ce qui permet le **secret du vote**.

Belenios donne aussi son nom à un ensemble de variantes (Belenios-RF, Belenios-CaI, ...). Un système très similaire à Belenios a été utilisé pour les élections législatives de 2022, pour les résidents à l'étranger ne pouvant pas facilement se rendre dans un bureau de vote.

2.2 Scrutin

Scrutin [4] est un prototype d'application web et mobile (Android, iOS) utilisant le protocole Belenios.

Un objectif de Scrutin est d'améliorer l'expérience utilisateur tout en restant une alternative aussi compatible que possible avec Belenios.

Scrutin est développé en OCaml (ReScript) et en React Native. Il utilise une implémentation du protocole Belenios en TypeScript [5].

En revanche, Scrutin n'existe qu'à l'état de prototype et a été peu utilisé en conditions réelles.

2.2.1 Interface utilisateur

Scrutin repose à la fois sur un protocole robuste (Belenios) et sur le travail de designers UI 1 (Marne Strazielle, LQDN) et UX 2 (Thibaud Frère, freelance).

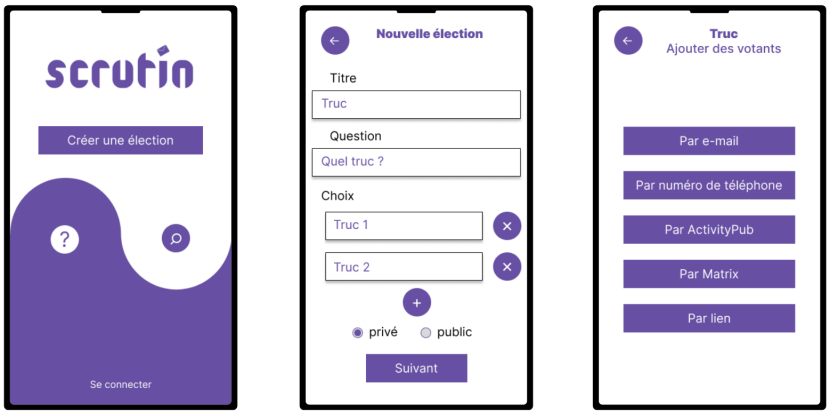


Figure 1: User interface

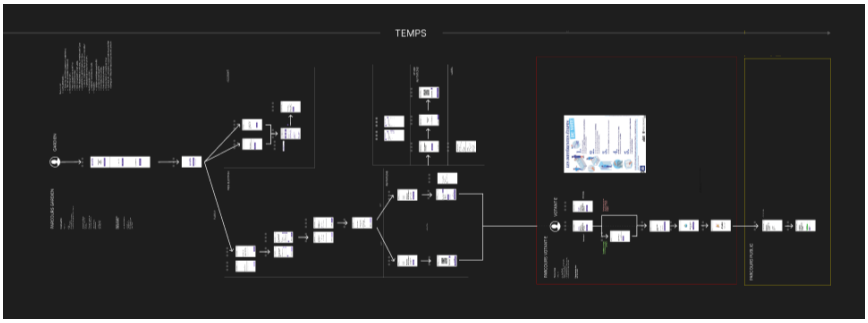


Figure 2: User eXperience

2.2.2 Jugement majoritaire

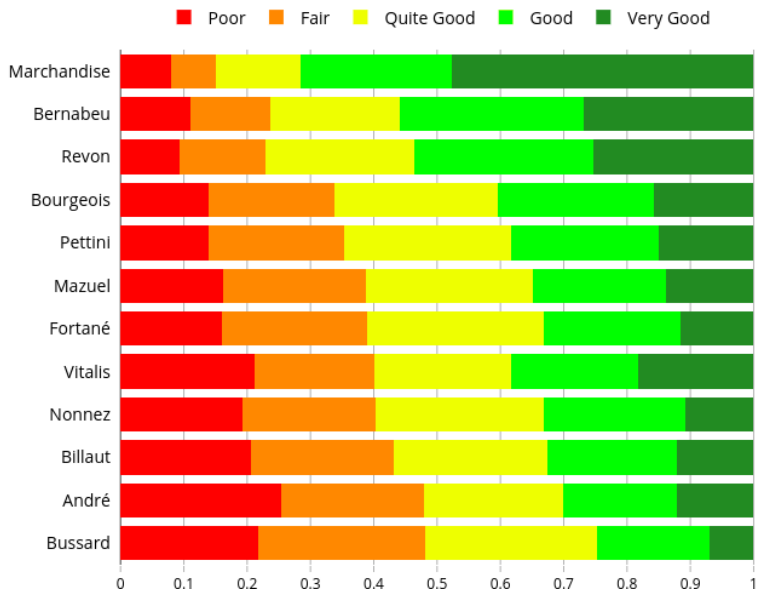


Figure 3: Jugement majoritaire

Le jugement majoritaire [2] 3 est une méthode de vote où chaque électeur exprime une évaluation sur chaque candidat, et où le vainqueur est celui qui obtient la meilleure appréciation collective. Au moment du comptage, on regarde l'appréciation la plus "centrale" pour chaque option.

C'est l'une des deux méthodes de vote supportés dans Scrutin, avec le scrutin uninominal (Belenios supportant plus de méthodes de vote).

2.3 Projet de fin d'étude

L'objectif de ce projet de fin d'études (PFE) est de consolider l'application en l'appliquant à des cas d'usage réels. Cela permettra de recueillir des retours afin d'élaborer une feuille de route des développements souhaitables.

3 Du prototype aux premières applications réelles

Scrutin est encore un projet peu mature, n'ayant jamais été utilisé en conditions réelles. Afin de dépasser le stade de projet étudiant (*in vitro*), il doit se confronter à des conditions réelles (*in vivo*). Pour ce faire, nous l'avons testé sur plusieurs cas d'usage.

Nous avons eu l'occasion d'utiliser Scrutin pour l'élection du Conseil d'Administration d'une association comptant 24 membres actifs : **Deuxfleurs** (`https://deuxfleurs.fr`). L'intérêt de cette élection est que l'enjeu est faible (il y a autant de sièges que de candidats) et qu'elle repose sur le jugement majoritaire, ce qui permet de tester ce mode de scrutin alternatif.

Ces expériences permettront de recueillir des retours utilisateurs, d'identifier les points à améliorer et d'élaborer une feuille de route pour les développements futurs.

3.1 Préparatifs

3.1.1 Emails

Nous utilisions un prestataire externe pour l'envoi de nos emails (sendgrid), or, Deuxfleurs souhaite (à raison) utiliser sa propre infrastructure. Nous avons donc paramétré un client SMTP avec la configuration de l'association. Il a fallu activer le mode "pool" pour avoir un envoi fiable, sinon une partie des emails était perdue ce qui avait des conséquences assez désastreuses car il n'est généralement pas possible de renvoyer les invitations.

3.1.2 Invitations

En plus du fonctionnement normal de Belenios où les invitations sont générées durant la phase de "Setup", il est aussi possible dans Scrutin de créer des élections en mode "ouvert". Dans ce cas, il sera possible de rajouter des bulletins de vote en cours d'élection, en y associant un pseudo, de manière similaire à Framadate. C'est évidemment un fonctionnement dégradé au niveau de la sécurité (au niveau de la vérifiabilité de l'éligibilité et la prévention du *ballot stuffing*). En revanche, cela permet d'autres cas d'usage, tels

que partager une élection dans un groupe (e.g. Whatsapp) sans avoir à inviter les membres individuellement. Dans ce mode "ouvert", les participants doivent vérifier que les noms correspondent bien aux membres du groupe, en sachant que des votants malicieux peuvent toujours essayer de voter pour les absents en usurpant leur nom. Ce mode d'invitation fait le pari que dans la plupart des cas, les votants n'essayeront pas de tricher. Nous en reparlerons à la fin de ce document.

3.1.3 Jugement majoritaire

Avec Belenios, il existe deux manières de faire une élection au jugement majoritaire: soit en utilisant une question en mode "mixnet" (tant que le nombre de mentions et de candidats n'est pas trop grand), soit en utilisant plusieurs questions en mode "homomorphe" (une question par candidat, où il faut simplement choisir une mention par les mentions possibles). Dans Scrutin, nous utilisons la seconde option.

Nous avons développé un nouveau parcours de création d'élection proposant l'option de faire une élection en mode jugement majoritaire. Il suffit de renseigner la liste des candidats, toutes les questions étant créées automatiquement. Une limitation est que dans ce mode, il n'est pas possible d'ajouter d'autres questions.

Au dépouillement, une logique particulière sera appliquée pour calculer la médiane, correspondant à la mention minimale que plus de la moitié des électeurs donnent à ce candidat.

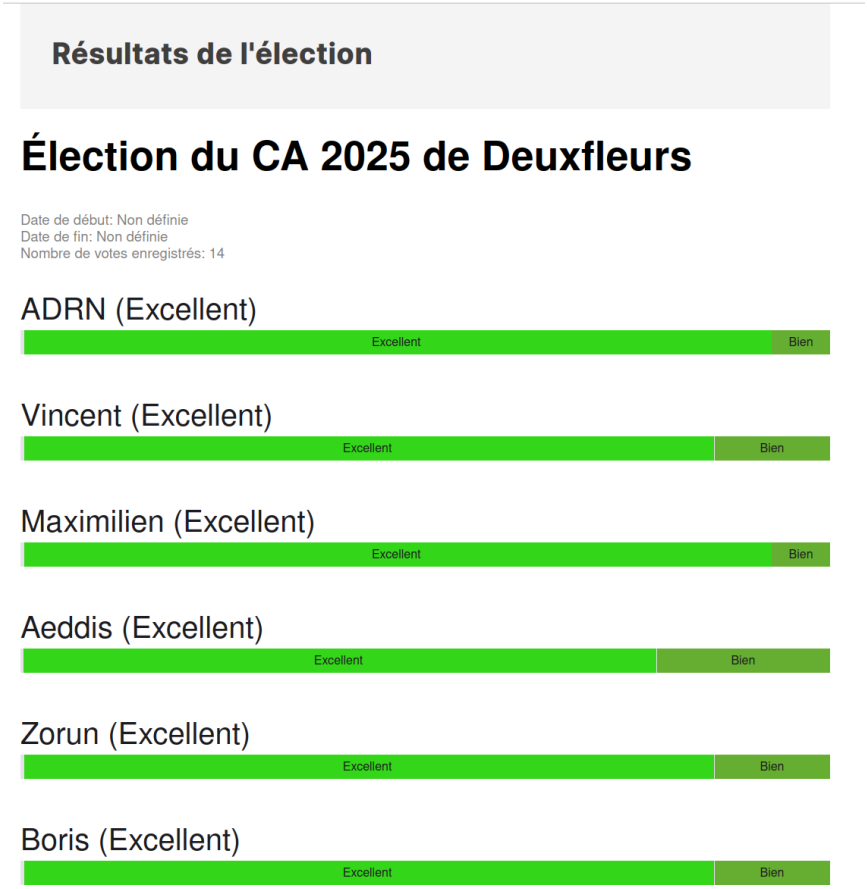
3.2 Déroulement de l'élection

On nous a fourni une liste de candidats ainsi qu'une liste de votants. Nous avons créé l'élection et sécurisé notre clé de trustee.

Nous avons créé l'élection, invitant au passage tous les participants par email, en utilisant la configuration et l'infrastructure de l'association.

Afin de pallier un éventuel problème lors de l'envoi des invitations (si par exemple des électeurs ne recevaient pas leur email), l'élection était en mode "ouvert" (même si le lien d'accès était resté caché), permettant aux éventuels votants n'ayant pas reçu d'invitation de se rajouter si besoin. Nous n'avons pas eu besoin d'utiliser cette fonctionnalité, tous les membres ayant bien reçu leur invitation.

14 personnes ont voté. Aucun problème particulier ne nous a été remonté.



3.3 Perspectives

Ce résultat étant encourageant, nous avons conçu une **feuille de route** détaillant les améliorations à apporter au logiciel.

Cette feuille de route est disponible sur GitLab et contient les catégories suivantes :

- Interface
- Système d'invitation
- Sécurité
- Nouvelles fonctionnalités

4 Conclusion

Ce projet de fin d'études a apporté plus de structure au projet, en le dotant d'une feuille de route détaillée. Il m'a aussi appris à mieux m'organiser et à prendre du recul sur le code. La création de ce document a d'ailleurs été un *byproduct* intéressant.

Bien que l'élection choisie ait eu peu d'enjeu (il y avait autant de places que de candidats), cette expérience m'a donné confiance pour proposer Scrutin à d'autres associations. Elle a aussi permis d'identifier des améliorations souhaitables, notamment en ce qui concerne le système d'invitation et les types de scrutin.

4.1 Remerciements

Je tiens à remercier l'université de Lille d'avoir accepté ce sujet et Adrien Luxey-Bitri de l'avoir supervisé. Ses nombreux conseils ont été précieux et m'ont aidé tant sur le plan rédactionnel que dans mon organisation.

Merci aussi à Adrien et à l'association Deuxfleurs de s'être proposés pour tester Scrutin en conditions réelles.

Merci au LORIA et à l'équipe de Belenios, sans qui Scrutin n'existerait pas. En particulier, un grand merci à Stéphane Glondu pour avoir encadré le développement du vérificateur indépendant d'élection (devenu Sirona) lors de mon Projet Individuel (PJI).

4.2 Perspectives

J'espère que ce projet, ainsi que le code de Scrutin et de Sirona, pourront constituer des contributions utiles au projet Belenios, ainsi qu'à la recherche et aux communs en général.

En tant qu'implémentation (partielle) de Belenios, Sirona est disponible comme vérificateur indépendant d'élections créé par Belenios [6]. L'existence d'un vérificateur indépendant fait partie des recommandations de la CNIL. Le code source de Sirona peut également servir de ressource pour étudier le fonctionnement du protocole de Belenios.

Il faut néanmoins nuancer ces affirmations : Sirona reste un projet étudiant, et il convient d'être vigilant face aux éventuels bugs. Des mesures telles qu'un audit, une suite de tests et éventuellement du fuzzing doivent encore être mises en place.

Scrutin, quant à lui, propose une interface alternative à celle de Belenios et cherche à intégrer les retours utilisateurs afin d'être le plus simple possible

à utiliser, sans compromettre la sécurité.

Scrutin vise également à offrir plus de méthodes d'invitation, notamment la possibilité pour l'administrateur d'ajouter des invitations en cours d'élection – un besoin fréquemment remonté – ou encore d'envoyer les invitations par d'autres moyens que l'email. Scrutin propose également des élections de type ouvert, qui peuvent être partagées au sein d'un groupe sans nécessiter d'inviter individuellement chaque membre. Dans les contextes où les membres du groupe sont bienveillants, ce mode constitue une alternative maximisant l'ergonomie tout en conservant certaines capacités de vérification de l'éligibilité (qui a le droit de voter). Chaque votant peut ainsi associer un pseudonyme à son vote, et tous peuvent vérifier les pseudonymes à la fin de l'élection.

References

[1] Ben Adida. "Helios: Web-based Open-Audit Voting." In: *USENIX security symposium*. Vol. 17. 2008, pp. 335–348.

[2] Michel Balinski and Rida Laraki. *Majority judgment: measuring, ranking, and electing*. MIT press, 2011.

[3] Stéphane Glondu. *Belenios specification*. 2024. URL: `https://www.belenios.org/specification.pdf`.

[4] Maxime Lalisse. *Scrutin.app*. Last accessed: February 17, 2025. 2025. URL: `https://github.com/mjal/scrutin` (visited on 02/17/2025).

[5] Maxime Lalisse. *Sirona*. Last accessed: February 17, 2025. 2025. URL: `https://github.com/mjal/sirona` (visited on 02/17/2025).

[6] Maxime Lalisse. *Sirona Web Interface*. Last accessed: February 17, 2025. 2025. URL: `https://mjal.github.io/sirona/` (visited on 02/17/2025).

[7] Véronique Cortier et Pierrick Gaudry. *Le vote électronique: Les défis du secret et de la transparence*. Last accessed: February 17, 2025. 2022. URL: `https://livrevote.loria.fr/` (visited on 02/17/2025).

[8] Stéphane Glondu Véronique Cortier Pierrick Gaudry. *Code source de Belenios*. Last accessed: February 17, 2025. URL: `https://gitlab.inria.fr/belenios/belenios` (visited on 02/17/2025).

[9] Stéphane Glondu Véronique Cortier Pierrick Gaudry. *Platforme de vote de Belenios*. Last accessed: February 17, 2025. URL: `https://vote.belenios.org/` (visited on 02/17/2025).