Projet de fin d'étude: Scrutin.app

Maxime Lalisse

February 28, 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 logistiques 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 France. En revanche, on y a souvent recours pour les élections non politiques, telles que les élections 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, Assembly Voting, 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-RF, Belenios-VS, 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 [8].

Scrutin [5] est un logiciel libre de vote électronique, développé comme un projet personnel et basé sur le protocole Belenios. Son principal objectif est d'offrir une expérience simple et intuitive sans compromettre la sécurité. Il vise également à être disponible sur mobile (Android, iOS) ainsi que sur le web.

Ce document est le rapport d'un projet de fin d'études (PFE) réalisé dans le cadre de la seconde année du Master d'informatique, mention Internet of

Things, de l'Université de Lille. Ce PFE porte sur le développement de Scrutin et son application à des cas d'usage réels.

Nous présenterons le contexte dans une première partie, puis nous ferons un retour sur l'utilisation de Scrutin dans des conditions réelles. Ensuite, nous établirons une feuille de route des développements futurs souhaitables avant de conclure.

2 Contexte

2.1 Belenios

Belenios est à la fois un protocole [4] basé sur Helios [1], une implémentation de ce protocole [9] composée d'un client, d'un serveur web et d'un outil en ligne de commande, ainsi qu'une plateforme de vote [10]. 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 [3].

2.2 Scrutin

Scrutin [5] est un logiciel libre de vote électronique, développé comme un projet personnel et basé sur le protocole Belenios. Il vise à être aussi compatible que possible avec Belenios lui-même. C'est un logiciel expérimental à plusieurs niveaux, notamment en ce qui concerne :

- **Son interface utilisateur**, afin de simplifier l'expérience de vote et de création d'élections.
- **Son usage sur mobile**, via des applications natives dédiées.
- **La gestion des invitations**, afin de s'adapter à des scénarios variés.

Scrutin est développé en OCaml (ReScript), un langage fonctionnel fortement typé, ce qui réduit les bugs, ainsi qu'en React Native, ce qui lui permet d'être disponible en tant qu'application mobile. Il utilise une implémentation du protocole Belenios en TypeScript [6].

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 Marne Strazielle (LQDN) (voir un exemple sur la figure 1). Un travail sur l'eXperience Utilisateur (UX) a été effectué par Thibaud Frère (freelance).

2.2.2 Jugement majoritaire

Le jugement majoritaire [2] (illustré dans la figure 2) est une méthode de vote où chaque électeur exprime une évaluation sur chaque candidat, et où

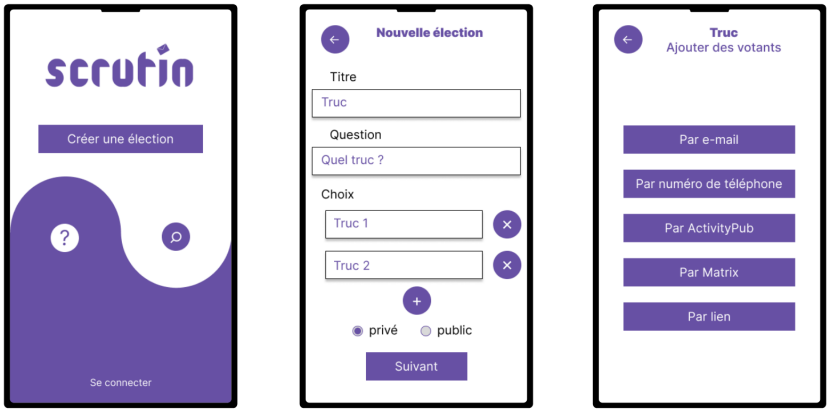

Figure 1: User interface

le vainqueur est celui qui obtient la meilleure appréciation collective. Au moment du comptage, on regarde l'appréciation la plus "centrale" pour chaque option.

Le jugement majoritaire aide à élire le candidat le plus apprécié par la majorité, plutôt que celui qui divise mais obtient juste assez de votes pour gagner.

Nous proposons ce mode de vote 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**

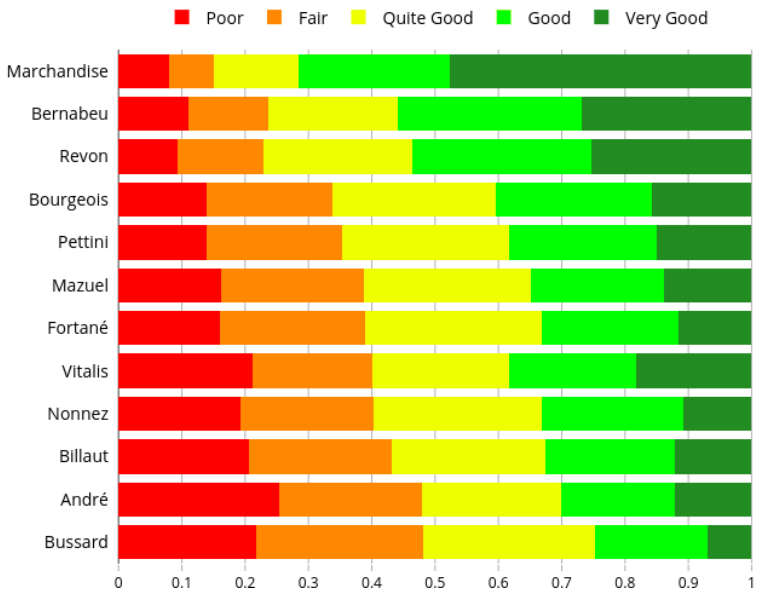

Figure 2: Jugement majoritaire

(`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

Le fonctionnement classique de Scrutin requiert que la liste des votants soit définie à l'avance, afin de leur envoyer des invitations par email. Les emails étant le canal par lequel les invitations sont envoyées, il est essentiel de ne pas en perdre, car il n'est généralement pas possible de les renvoyer.

Pour le prototype, nous utilisions un prestataire externe pour l'envoi des emails. Cependant, Deuxfleurs souhaitant utiliser sa propre infrastructure, nous avons dû développer un système robuste d'envoi d'emails.

3.1.2 Élections ouvertes

En plus du fonctionnement classique où les invitations sont générées durant la phase de mise en place, Scrutin permet aussi de créer des élections en mode "ouvert". Dans ce mode, l'identifiant utilisé pour voter est un pseudo choisi par le votant. C'est évidemment un fonctionnement dégradé en termes de sécurité (notamment pour la vérifiabilité de l'éligibilité et la prévention du bourrage d'urne).

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 repose sur l'hypothèse que tous les participants sont honnêtes. Nous en reparlerons à la fin de ce document.

3.1.3 Jugement majoritaire

Nous avons légèrement adapté le fonctionnement de Belenios dans notre approche du jugement majoritaire. Concrètement, nous utilisons une question par candidat, chaque question demandant de noter le candidat sur une échelle.

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

Lors du dépouillement, une logique spécifique est appliquée pour calculer la médiane, correspondant à la mention minimale que plus de la moitié des électeurs attribuent 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 la clé de chiffrement nécessaire au dépouillement.

Nous avons ensuite invité 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). Cela permettait

aux éventuels votants n'ayant pas reçu d'invitation de se rajouter si besoin. Cependant, nous n'avons pas eu à utiliser cette fonctionnalité, tous les membres ayant bien reçu leur invitation.

Les résultats sont affichés ci-dessous (figure 3). On peut voir, pour chaque candidat, le nombre de mentions reçues ainsi que la mention retenue (ici, tous les candidats ont reçu la mention Excellent).

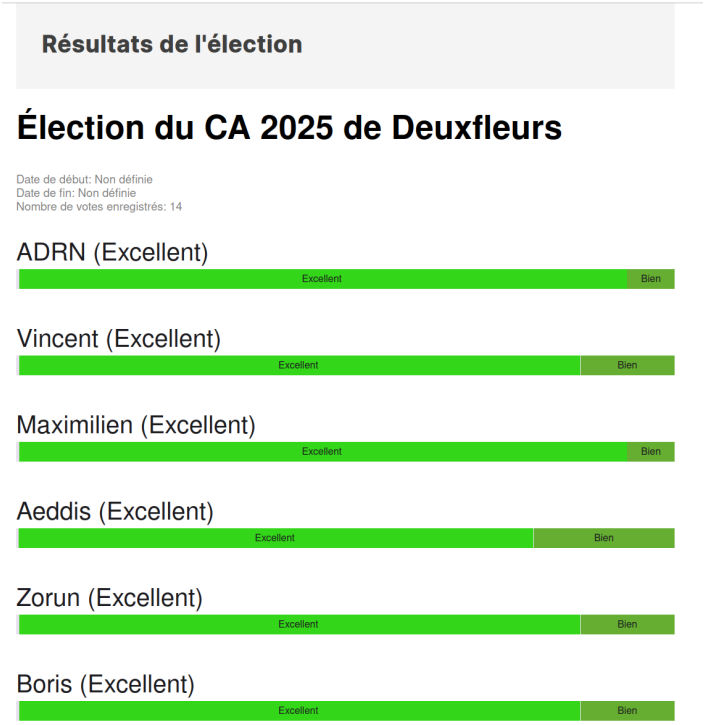

Figure 3: Résultats de l'élection

3.3 Feuille de route

Le résultat de cette expérience étant encourageant, nous avons conçu une **feuille de route** détaillant les améliorations à apporter au logiciel. Elle est découpée dans les sections suivantes.

- Système d'invitation
- Sécurité

- Nouvelles fonctionnalités
- Communication

3.3.1 Système d'invitation

Les élections de type "ouvert" manquent encore des fonctionnalité crutiales, telle que l'affichage du pseudo à coté du bulletin chiffré.

Pour les élections de type "fermé", il semble aussi souhaitable que l'administrateur de l'élection puisse aussi rajouter ou révoquer des invitations en cours de scrutin, si certains votants n'ont pas pu reçu leur invitation. Ces opérations seront visisbles pour tous les votants afin de rester aussi transparant que possible.

3.3.2 Sécurité

- Le hash du ballot devrait être affiché après avoir voté
- L'élection devrait pouvoir être verifié via Belenios et/ou Sirona, via un export au format .bel

3.3.3 Nouvelles fonctionnalités

La notion d'administrateur de l'élection est prévue, permettant de pouvoir notamment gérér les invitations après la création de l'élection. C'est une notion différente que celle de "trustee". L'administrateur pourrait également éventuellement modifier l'élection avant sa date de début.

La notion de compte utilisateur associé à un email pourrait également être pratique.

3.3.4 Communication

Au-delà des aspects techniques, la pérennité d'un logiciel repose aussi sur ce qui le soutient, notamment sa communication et sa communauté.

Nous souhaitons travailler sur un site web de présentation ainsi que sur une documentation claire et accessible.

De plus, nous comptons en parler au sein des communautés du logiciel libre, telles que Framalibre, Wikilibriste et LinuxFr.org, afin de mieux le faire connaître et encourager les contributions.

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 pourra constituer une contribution utile au projet Belenios, ainsi qu'à la recherche et aux communs en général.

Scrutin peut servir d'environnement d'expérimentation pour diverses fonctionnalités, que ce soit au niveau de l'interface, du système d'invitation ou même de la décentralisation. Son objectif est de rester le plus simple possible, sans compromettre la sécurité.

Sirona, développé en parallèle et dont nous avons peu parlé, est une implémentation alternative et partielle du protocole de Belenios. Sirona est également disponible comme vérificateur d'élections créé par Belenios [7]. Un tel vérificateur indépendant fait partie des recommandations de la CNIL. Nous espérons que ce projet pourra aussi être utile.

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] Véronique Cortier et al. "French 2022 legislatives elections: a verifiability experiment". In: *The E-Vote-ID Conference 2023*. 2023.

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

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

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

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

[8] 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).

[9] 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).

[10] 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).