SwissiVi

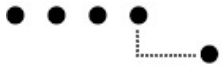
Haute école spécialisée bernoise
Technique et informatique

Andrea Pellegrini, Philémon von Bergen

Module de projet 2

Dr. Eric Dubuis, Dr. Rolf Haenni, Reto Eric Koenig

Table des matières



1 Historique des modifications

Révision	Date	Commentaire	Auteur
1	6 octobre 2011	Première édition	vonbp1, pella1
2	7 octobre 2011 au 17 janvier 2012	Mises à jour	vonbp1, pella1
3	17 janvier 2012	Relecture finale	vonbp1, pella1

Table 1 – Historique des modifications

2 Introduction

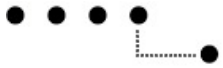
Dans le cadre du module de projet 2, nous avons reçu le mandat de contribuer au projet de recherche sur le vote électronique en réalisant un système de simulation qui pourra être utilisé lors de présentation du projet d'e-voting.
Nous avons décidé de baptiser notre partie du projet SwissiVi. « Swiss » parce que le projet est développé et serait utilisé en Suisse, et « iVi » en référence aux initiales e (se prononçant « i » en anglais) et V (prononcé « Vi ») de « e-Voting ».

Le travail principal du présent projet consiste à élaborer les spécifications pour le système de vote électronique fonctionnant de la manière suivante.
Le votant accède à un site internet (plateforme de vote). Ce site rassemble tous les objets pour lesquels le votant doit s'exprimer. Pour chacun de ces objets, un code-barres contenant la réponse choisie par le votant est mis à disposition. Le votant doit alors scanner ce code-barres à l'aide d'un petit appareil (l'appareil de vote) dans lequel il aura préalablement inséré sa carte de légitimation de vote (une smartcard nommée carte de vote). L'appareil va lire le contenu du code-barres scanné, il va l'afficher sur l'écran et demander à l'utilisateur de confirmer son choix. Si le votant accepte, il va crypter le fichier et le transférer sur la carte de vote qui, elle, va le signer numériquement.
Le fichier contenant la réponse se trouvant maintenant sur la carte de vote devra ensuite être transféré sur l'ordinateur et sera envoyé sur une plate-forme de récolte des votes (Bulletin-Board).

Une documentation déjà existante décrit le concept ci-dessus plus en détails. Ce n'est pas l'objet de la présente documentation, celle-ci se rapportant plutôt à la partie de simulation de l'appareil de vote et à la mise en place de la plateforme de vote. Pour rester consistant avec les autres documentations écrites en allemand, nous allons utiliser les traductions suivantes :
– « plateforme de vote » pour « Wahlplatform »
– « appareil de vote » pour « Wahlgerät »
– « carte de vote » pour « Wahlkarte »

3 Portée du projet

Dans le cadre de ce projet, deux parties bien distinctes vont être analysées :
– le simulateur de l'appareil et de la carte de vote
– la plateforme de vote



3.1 Le simulateur

Une partie du projet consiste en la réalisation des spécifications de l'appareil de vote et de la carte de vote. Comme nous ne disposons pas encore du matériel adéquat, l'appareil et la carte doivent être simulés. Cela se fera à l'aide de smartphones, plus précisément à l'aide de deux applications pour le système d'exploitation Android. Un smartphone ne pourrait pas être utilisé comme appareil de vote directement, car il n'offre pas la sécurité d'un appareil de vote dédié. Le but de son utilisation est simplement de démontrer le déroulement d'un vote lors de présentations.

3.2 La plateforme de vote

La seconde partie du projet consiste en la réalisation des spécifications de la plateforme de vote. Celle-ci consistera en un site internet sur lequel on retrouvera toutes les informations nécessaires pour les votes.
Ce site internet devra mettre à disposition les objets d'une votation et leurs codes-barres correspondants. Il devra également prendre des élections en charge.
Le site comprendra également une partie où l'on pourra uploader le fichier généré par le simulateur, afin d'afficher le résultat de la votation. En réalité, cela devrait se faire par l'intermédiaire d'un canal anonyme, mais cela ne fait pas partie de ce projet. Par conséquent, les spécifications de ce canal anonyme ne seront pas étudiées ici.

4 Spécifications du simulateur

4.1 Déroulement d'un vote

Pour signer le vote afin que les autorités puissent contrôler si la personne ayant voté y est autorisée, le votant doit insérer sa carte de vote dans son appareil de vote. Il doit ensuite confirmer qu'il est bien le détenteur de cette carte en introduisant son code PIN. Ces deux actions peuvent se faire à différents moments dans le cycle du déroulement d'un vote. Dans ce chapitre, nous allons aborder les avantages et les inconvénients de quelques-unes de ces possibilités.

4.1.1 Insertion de la carte de vote au début du processus de vote

Cette solution propose l'insertion de la carte et ensuite du code PIN avant toute utilisation de l'appareil de vote, autrement dit, l'appareil est inutilisable sans la carte. L'insertion de la carte allume automatiquement l'appareil, en l'enlevant il s'éteint automatiquement.

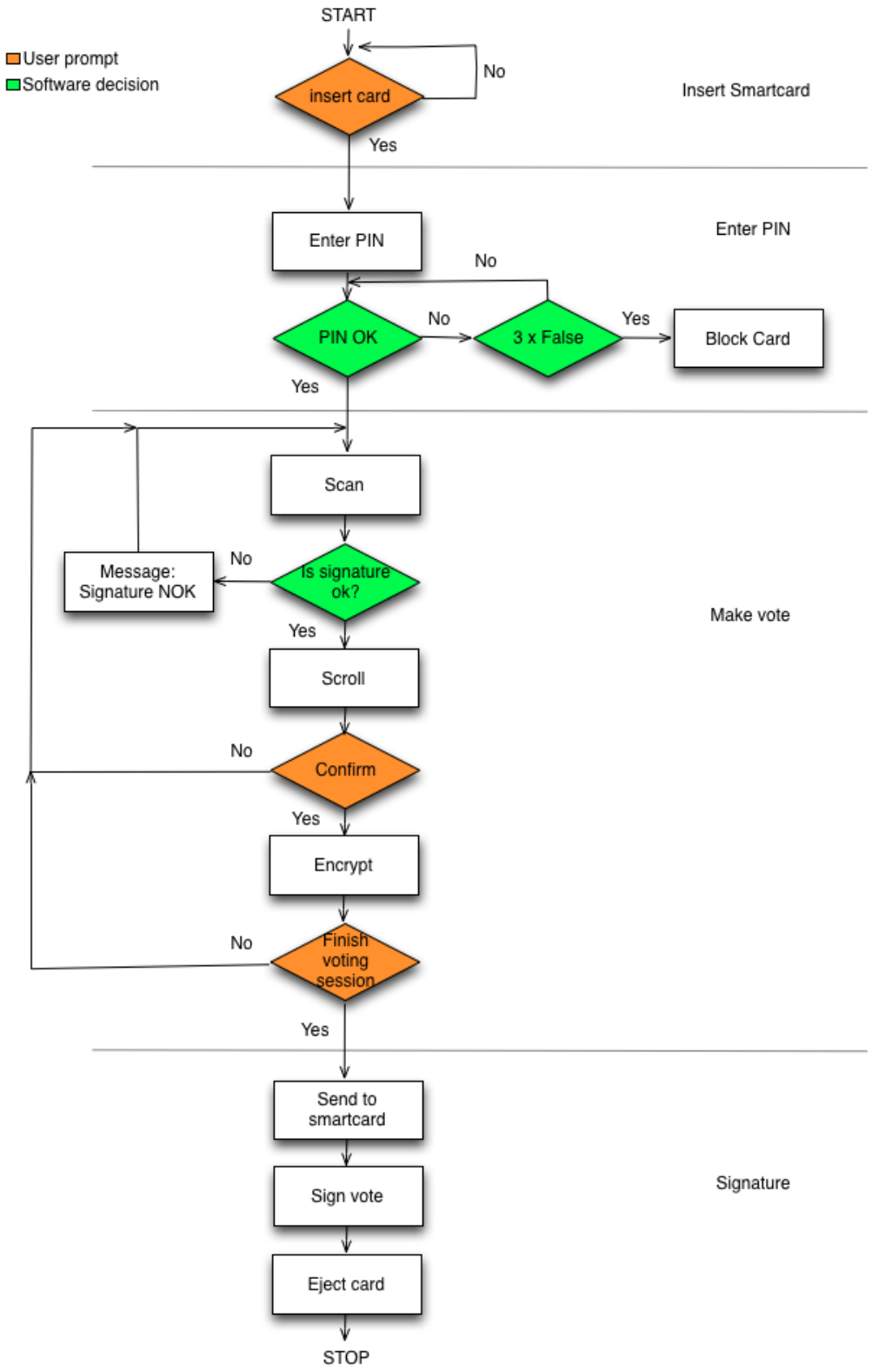


Figure 1 – Diagramme de déroulement d'un vote : Insertion de la carte et du PIN au début

Ci-dessous un tableau qui en présente les avantages et les inconvénients :

Avantages
Impression de sécurité : la confirmation demandée par l'appareil de vote est protégée par mon code PIN
Inconvénients
Pas de possibilité de faire des essais sans insérer la carte de vote
Le retrait de la carte peut être fait avant que la signature soit terminée dû à l'effet psychologique « J'ai confirmé mon choix, donc je peux retirer ma carte »
Si l'utilisateur laisse l'appareil de vote sans surveillance avec la carte insérée et que le PIN a déjà été entré, n'importe quelle personne qui trouve l'appareil peut faire le vote à sa place.

Table 2 – Avantages et inconvénients

4.1.2 Insertion de la carte de vote au début du processus de vote, mais introduction du PIN à la fin

Cette solution propose l'insertion de la carte tout au début du processus de vote, mais, contrairement à la version précédente, le code PIN est indiqué seulement à la fin, quand tous les votes ont été faits (voir figure 2). L'insertion de la carte allume automatiquement l'appareil de vote, en l'enlevant, il s'éteint automatiquement.

Ci-dessous un tableau qui en présente les avantages et les inconvénients :

Avantages
L'insertion du code PIN ne se fait qu'au moment où la carte va vraiment être utilisée.
Si je laisse l'appareil de vote sans surveillance avec la carte insérée, personne ne peut voter à ma place.
Inconvénients
Quand on arrive à la question : « Finish voting session » et que le votant a fini de voter, il se peut que l'utilisateur retire la carte avant d'entrer le code PIN. Ceci est pourtant moins probable que dans le scénario 1 où l'utilisateur a déjà entré le code PIN au début.

Table 3 – Avantages et inconvénients

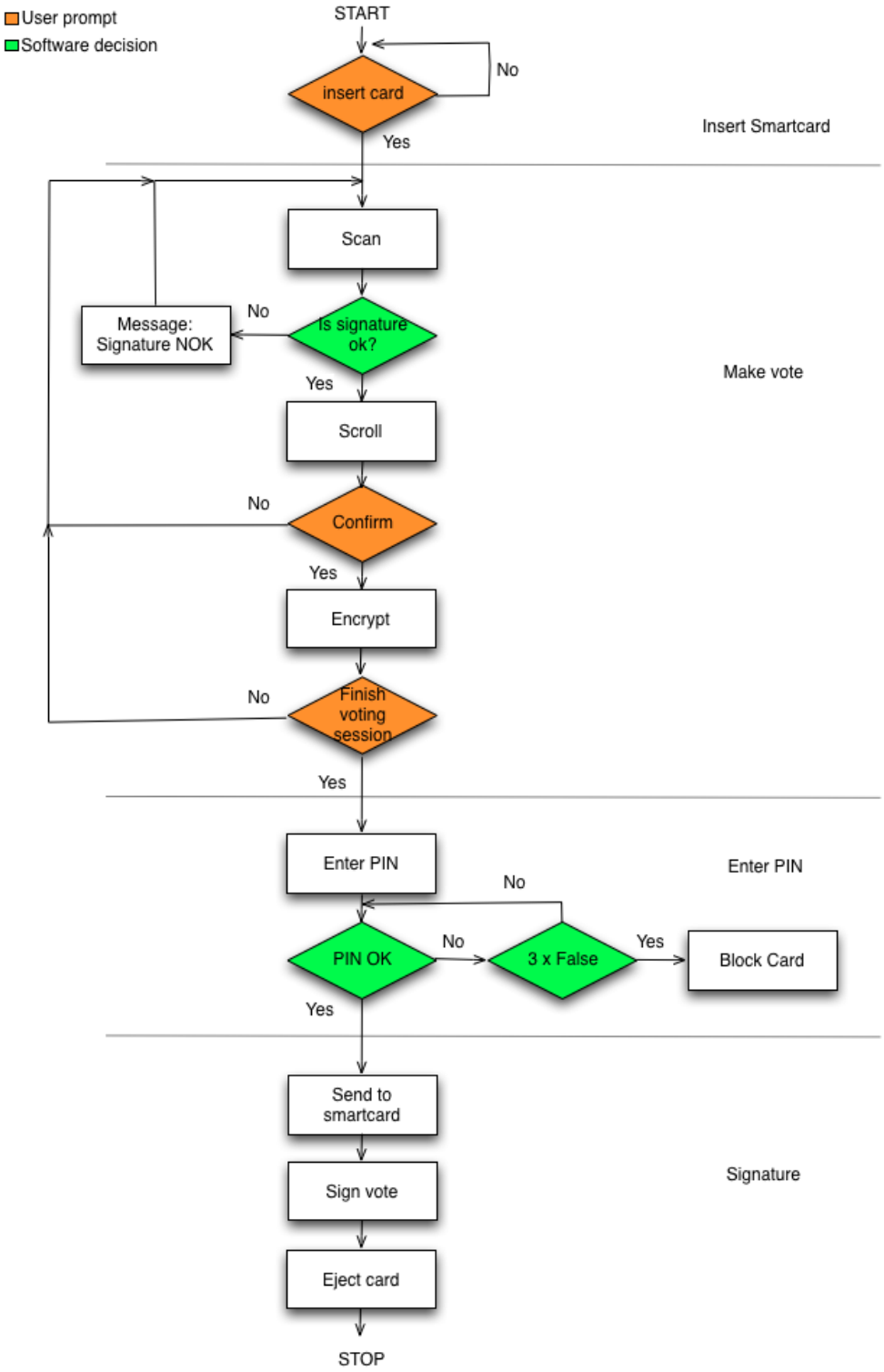


Figure 2 – Diagramme de déroulement d'un vote : Insertion de la carte au début et introduction du PIN à la fin

4.1.3 Insertion de la carte de vote lorsque tous les votes ont été confirmés

Cette solution propose l'insertion de la carte lorsque le votant a effectué et confirmé tous ses choix (voir figure 3). Le retrait de la carte éteint l'appareil de vote.

Ci-dessous un tableau qui en présente les avantages et les inconvénients :

Avantages
la carte n'est insérée qu'au moment où elle est vraiment utilisée, c'est à dire pour la signature des votes
Mise en confiance en cas de scan d'un mauvais code-barres : l'erreur n'a pas pu être enregistrée, car je n'ai pas inséré la carte
Inconvénients
Une personne peut préparer le vote et demander à une autre personne d'insérer sa carte pour signer ces votes
Le votant doit allumer l'appareil de vote avec un bouton dédié
L'insertion de la carte va engendrer la signature de tous les fichiers se trouvant sur l'appareil de vote. Il se peut que des fichiers d'un test précédent se trouvent alors encore sur l'appareil et vont par conséquent être signés. Prenons un exemple pour ce cas : Alice vote pour les objets X,Y,Z mais n'introduit pas sa carte, ses fichiers restent donc sur l'appareil. Bob arrive et reprend le même appareil. Il vote seulement pour l'objet X. Lorsqu'il insère la carte, il va aussi signer les objets Y et Z d'Alice.

Table 4 – Avantages et inconvénients

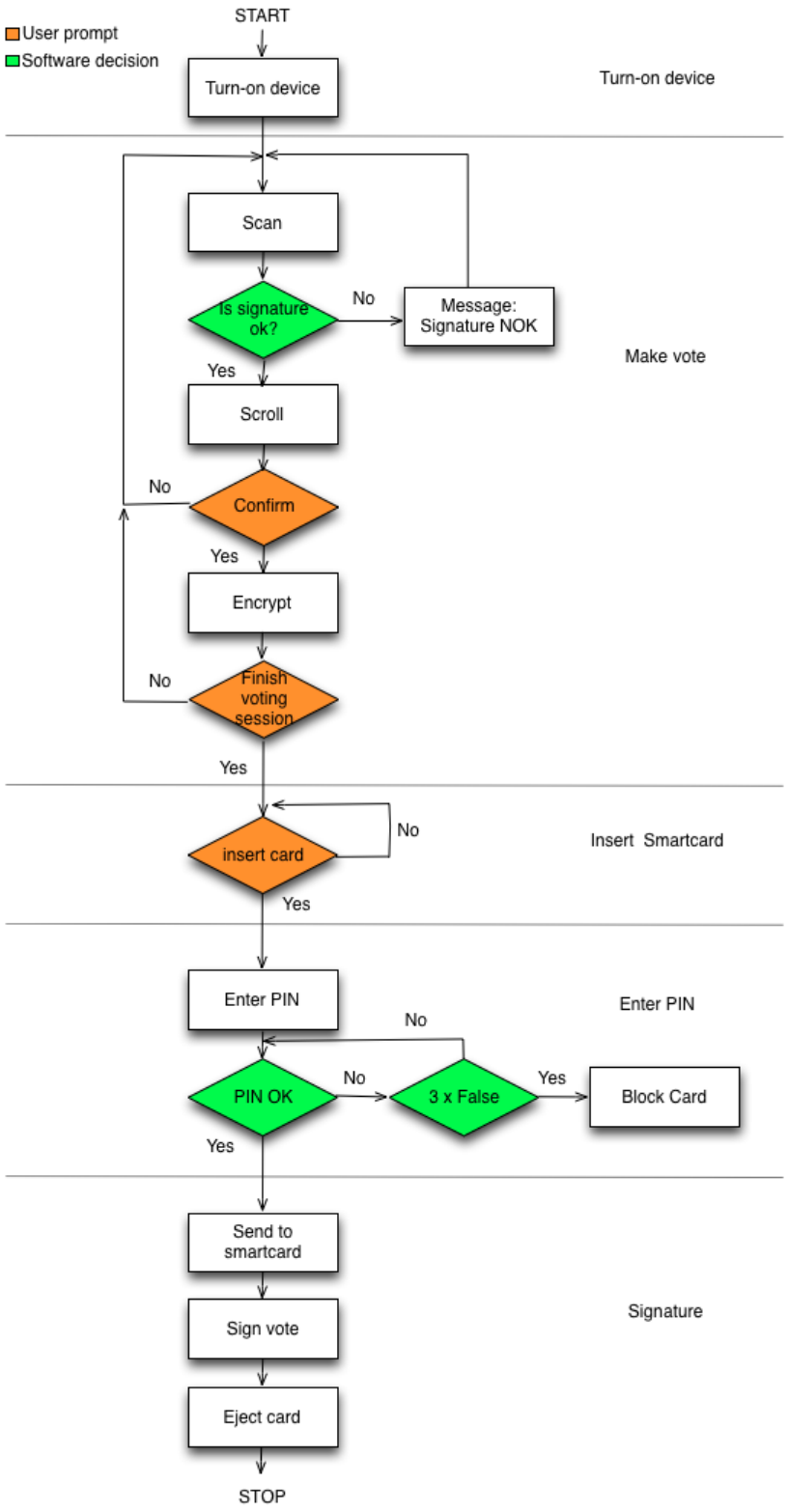


Figure 3 – Diagramme de déroulement d'un vote : Insertion de la carte et du PIN à la fin

4.1.4 Conclusion

Il est difficile de définir quelle est la meilleure des trois solutions. Encore une fois, le but est de faciliter la tâche à l'utilisateur. Des essais pratiques réalisés par différents utilisateurs pourraient influencer le choix. La solution la plus adaptée au niveau technique nous semble être la deuxième solution. Par conséquent, c'est celle-là que nous allons implémenter. Toutefois, les autres solutions restent absolument accessibles.

4.2 Nombre des codes-barres

Au cours du projet, nous avons remarqué qu'il ne sera certainement pas possible d'inclure le résultat d'une élection dans un seul code-barres (sujet traité plus loin au chapitre 5.5). Par conséquent, nous avons dû adapter le déroulement de façon à ce qu'il puisse gérer le scan de plusieurs codes-barres. A cet effet, les deux variantes suivantes ont été proposées.

4.2.1 Code-barres contenant les données générales

Une solution serait par exemple de créer un code-barres qui contienne toutes les informations générales des votations/élections, c'est à dire l'identification et la description des objets de vote, les objets de votes, la valeur $\hat{g}$ (utilisée pour la cryptographie) et la signature qui permet de vérifier l'intégrité de quelques-unes de ces données.
L'utilisateur devrait donc scanner ce code en premier, puis il pourrait scanner ses réponses.

Par conséquent, le bloc « Make vote » doit être adapté pour pouvoir accepter deux code-barres. Ci-dessous, voici un diagramme de flux représentant le bloc « Make vote » modifié pour cette solution (voir figure 4). Cette variante peut être intégrée dans les trois solutions présentées au chapitre précédent.

Cette solution présente toutefois certains inconvénients :
La votation la plus grande en Suisse peut contenir jusqu'à 90 candidats. Même en appliquant cette technique, un seul code-barres ne suffit pas pour contenir toutes les données.
Un autre inconvénient de cette solution est qu'il peut y avoir une grande différence de taille entre le code-barres contenant les données générales et le code-barres contenant la réponse.
Enfin, du point de vue de l'utilisateur, il n'est pas forcément logique de devoir faire un scan au début, qui du point de vue de l'utilisateur, ne contient rien.

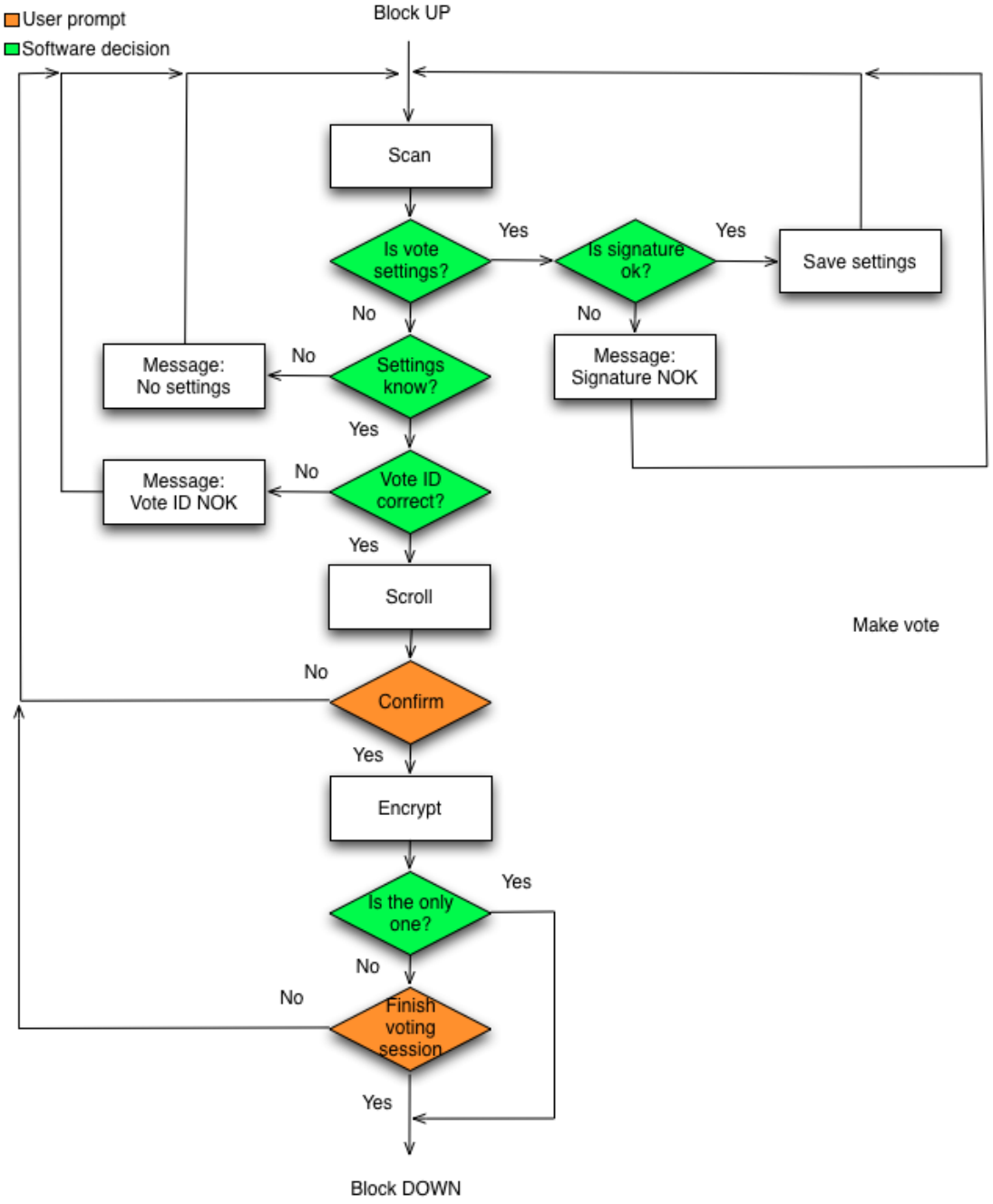


Figure 4 – Variante du bloc « Make vote » avec la prise en charge d'un deuxième code-barres contenant les données générales du vote.

4.2.2 Code-barres en plusieurs parties

Une autre solution pour résoudre le problème de capacité serait de créer deux codes de taille égale sans appliquer la solution précédente. Le premier code contiendrait donc les données générales plus le début de la réponse, et les codes suivants contiendraient le reste de la réponse. Cela impliquerait la modification du bloc « Make vote » de la façon suivante :

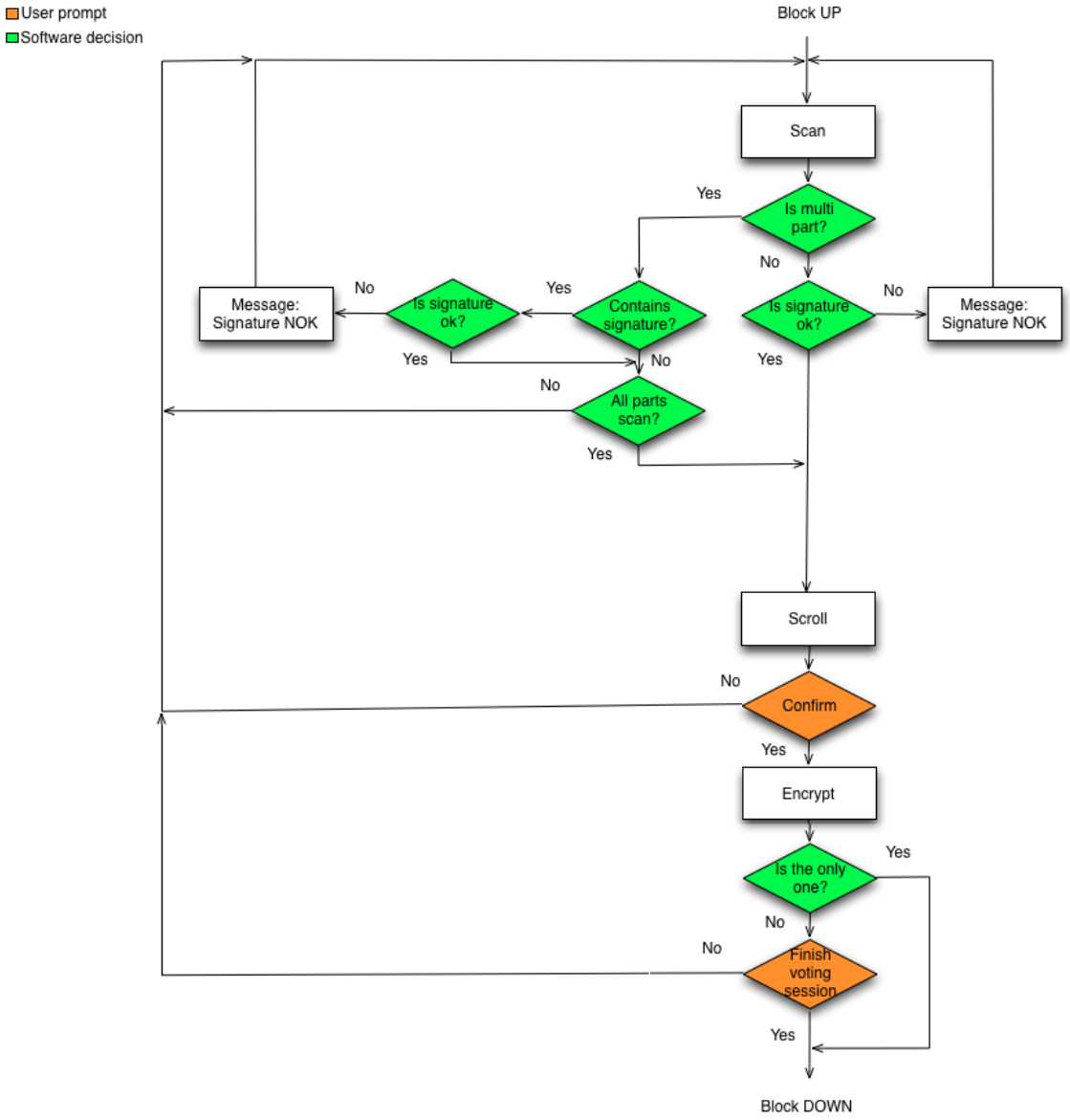


Figure 5 – Variante du bloc « Make vote » avec la prise en charge de plusieurs codes-barres.

Cette solution offre l'avantage de ne pas devoir conserver des données d'un objet de vote à l'autre, comme cela serait le cas si on scannait les données générales au départ.

Un autre avantage est que cette solution implémente déjà la possibilité de scanner plusieurs codes et pas seulement deux. Cette solution est donc extensible.

4.2.3 Conclusion

La deuxième solution est visiblement plus flexible et plus avantageuse, raison pour laquelle nous avons décidé d'implémenter cette solution, c'est-à-dire la variante **code-barres en plusieurs parties**.

4.3 Sauvegarde des fichiers de vote

Une autre question se pose : « Où faut-il sauvegarder les fichiers, sur l'appareil de vote ou sur la carte de vote ? ».
La réponse est relativement logique. Les réponses aux objets de vote sont quelque chose de personnel tout comme la carte de vote. L'appareil de vote, quant à lui, est un objet qui peut être partagé. Il est, par conséquent, plus sensé de placer les fichiers sur la carte, puisqu'ils sont tout deux des éléments personnels.
Cela sous-entend donc qu'il faut supprimer les fichiers sur l'appareil. Le moment le plus approprié est certainement lors de la signature des fichiers de vote par la carte. Les fichiers seraient donc déplacés (et non copiés) de l'appareil à la carte de vote.
Cette solution offre les avantages suivants :

Avantages
Les fichiers du dernier votant ayant utilisé l'appareil de vote ne peuvent plus être réutilisés par le votant suivant.
La carte peut signer tous les fichiers présents sur l'appareil de vote

Table 5 – Avantages du déplacement des fichiers sur la carte de vote

A ceci, il peut être intéressant d'ajouter la suppression des votes lors du retrait de la carte même s'ils n'ont pas été signés. Ceci offre l'avantage de ne pas laisser de votes non signés sur l'appareil qui pourraient être utilisés par le prochain votant.

Dans le cas où un utilisateur reproduit une votation déjà existante sur sa carte de vote, un message doit lui indiquer, lors du déplacement du fichier de l'appareil vers la carte, qu'il a déjà voté pour cet objet . Il peut alors décider de remplacer le vote existant ou de le garder.

4.4 Caractéristiques de l'appareil de vote

L'appareil de vote devra comporter plusieurs éléments. Tout d'abord, il y aura besoin d'un clavier numérique (touches 0 à 9) qui permettront d'entrer le code PIN. De plus on aura besoin de deux autres boutons, un pour valider le choix, un autre pour l'annuler.
Afin de minimiser le prix de l'appareil, il a été décidé de le faire le plus simple possible. Par conséquent, l'écran sera limité à un certain nombre de lignes (prenons par exemple deux lignes). Tous les textes ne sont pas forcément affichables sur deux lignes, par conséquent, il nous faut encore deux boutons qui permettent de faire défiler les textes sur l'écran.

L'appareil devra comporter une caméra capable de lire les codes-barres. Notre recommandation au niveau de la qualité de la caméra est une résolution de minimum 3.2 Mégapixels. Les tests ont permis de démontrer qu'avec une caméra de ce type, les codes-barres de type QR-code peuvent être lu relativement facilement jusqu'à la version 18. Une résolution supérieure est, bien entendu, un plus, car elle permettrait de lire des codes de plus haute capacité. L'élément déterminant sera le prix.

Les tests de lecture des codes-barres ont été réalisés avec un smartphone possédant un grand écran affichant la zone scannée par la caméra, et permettant ainsi un positionnement optimal. Cela ne pourra pas se faire sur l'appareil de vote qui ne comportera qu'un écran capable d'afficher du texte. Comment l'utilisateur va-t-il donc savoir à quelle distance positionner l'appareil pour réaliser le scan ? En effet, il risque d'être soit trop près, donc le code ne sera pas lisible en entier, soit trop loin et il se pourrait qu'il y ait deux codes visibles en même temps.
Pour éviter ce problème, une solution serait d'intégrer une diode lumineuse de couleur éclairant la zone scannée par la caméra. De cette façon l'utilisateur pourra savoir s'il scanne la bonne région de l'écran.
Un autre composant qui pourra être utile est un générateur de signaux sonores. Ainsi, quand le scan a été réalisé correctement, l'appareil peut en informer l'utilisateur à l'aide d'un bip.

L'appareil devra également comporter un lecteur de carte où l'utilisateur pourra introduire sa carte de votant. Une variante à cette solution serait d'insérer une puce NFC dans la carte et dans l'appareil de vote et d'utiliser cette technologie pour la communication entre ces deux éléments. Cette idée devra toutefois être approfondie le moment venu, car il n'est pas certain que suffisamment d'énergie puisse être transportée à la carte pour lui permettre d'exécuter les calculs de signature numérique (en partant d'un point de vue que la carte de vote doit rester un composant passif).

Voici ci-dessous un exemple de ce à quoi pourrait ressembler l'appareil de vote. Cette image est inspirée de l'appareil utilisé par PostFinance pour les paiements en ligne.

Figure 6 – Exemple d'appareil de vote

4.5 Essais propres au système de simulation

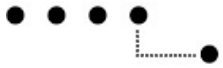
Dans le cadre de ce projet, il était également prévu de nous confronter aux technologies qui devront être utilisées pour la réalisation de ce projet lors du travail de bachelor.

4.5.1 Développement d'applications sur Android

Afin d'acquérir un peu d'expérience dans le développement d'applications pour téléphones portables équipés d'un système Android, nous avons décidé de réaliser une application qui simule le déroulement comme il aura lieu sur l'appareil de vote, sans pour autant implémenter toute la logique de calcul.
Le résultat est livré en annexe de ce document, au format informatique. Il est constitué d'une application au format APK et du code source de l'application.

4.5.2 Communication entre les smartphones

L'idée pour la simulation de l'appareil de vote et de la carte de vote est de développer deux applications séparées se trouvant finalement sur deux smartphones différents. Le principe de « near field communication » permettrait ensuite de déplacer les fichiers d'un smartphone (appareil de vote) à l'autre (carte de vote).
Etant donné qu'il n'a pas été possible de se procurer deux téléphones prenant en charge la technologie NFC, nous n'avons pas pu faire de test dans le cadre de ce projet. Nous avons toutefois essayé de récolter quelques connaissances théoriques sur cette technologie à l'aide de tutoriels.



5 Spécifications de la plateforme de vote

La plateforme de vote est le site internet mettant à disposition les objets de vote ainsi que les codes-barres correspondants. C'est sur cette interface que le votant se déplacera pour prendre connaissance des objets de vote ou des candidats aux élections.
La règle numéro une pour cette plateforme est qu'elle doit être aussi simple et conviviale que possible pour le votant.

5.1 Type de votes pris en charge

Au niveau suisse, différents types de votes existent. En voici une liste :

Type	Fonctionnement
Vote oui / non	Possibilité d'accepter ou de refuser l'objet
Initiative avec contre-projet	Possibilité d'accepter ou de refuser l'initiative, d'accepter ou de refuser le contre-projet, ainsi que d'indiquer sa préférence si l'initiative et le contre-projet sont acceptés
Elections du Conseil National	Le nombre de candidats éligibles varie en fonction du canton, et chaque candidat peut recevoir jusqu'à deux voix
Elections du Conseil des Etats	2 candidats par canton (1 par demi-canton) et une seule voix possible par candidat
Elections des gouvernements cantonaux	Dépend du canton
Elections des parlements cantonaux	Dépend du canton
Elections municipales	Dépend de la municipalité
Elections communales	Dépend de la commune
Elections paroissiales	Dépend de la commune

Table 6 – Types de votes en Suisse

La question qui se pose alors est la suivante : « Faut-il implémenter une logique de contrôle du vote pour chacun de ces cas ? ».

Il a été décidé de prendre en charge, pour ce projet, les votations oui/non, les initiatives et les élections au niveau fédéral, cantonal et communal. L'élection avec le plus de candidats élus à lieu au Tessin. Pour les élections cantonales, jusqu'à 90 candidats peuvent être élus. Notre système doit donc gérer une élection avec 90 candidats.

5.2 Identification des votes

L'identification des votes consiste à attribuer à chaque objet de vote ou session d'élections un identifiant unique de façon à ce qu'il puisse facilement être reconnu. Cet identifiant devra être passé dans le code-barres et ajouté dans le fichier crypté par l'appareil de vote, afin que le Bulletin-Board sache de quel objet il s'agit.
Une variante serait de générer cet identifiant de façon à ce qu'il contienne un certain nombre d'informations comme la date, le niveau auquel il s'applique (fédéral, cantonal, communal), s'il s'agit d'une votation, d'une élection ou d'une initiative, etc...
L'autre variante est d'incrémenter une valeur à chaque votation.
Les deux possibilités sont utilisables. Les informations nécessaires à l'appareil de vote sont déjà de toute façon transmises dans le code-barres par un autre moyen, la première variante n'offre donc pas plus d'avantages que la deuxième. L'administrateur est libre de décider ce qu'il préfère utiliser, il faut seulement respecter la règle qu'un identifiant ne doit en aucun cas être utilisé plusieurs fois.

5.3 Déroulement d'une élection

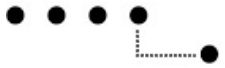
Lors d'une élection, le votant doit élire des candidats. Dans certaines élections, il peut attribuer plusieurs voix au même candidat.
On peut concevoir deux principes selon lesquels l'électeur peut atteindre son but.

5.3.1 Un code-barres par candidat

Selon ce principe, chaque candidat posséderait son propre code-barres. La liste complète posséderait également un code-barres. L'électeur devrait alors scanner le code-barres de chaque candidat qu'il désire élire, ou de la liste, si tel est son choix. Il pourrait également scanner le même code-barres à plusieurs reprises pour attribuer plusieurs voix au même candidat.
Ce principe présente un grand inconvénient de convivialité pour l'électeur. Il doit à chaque fois scanner, puis confirmer son choix, et cela pour chaque candidat. Le risque d'oublier pour qui il a déjà voté est également grand.
Ce système offre cependant l'avantage de la confidentialité. En effet, l'ordinateur (ou le support non sécurisé utilisé pour afficher la plateforme) n'a aucun moyen de savoir quel code-barres est scanné.

5.3.2 Le concept du panier à commission

Le principe du panier à commission (shopping cart) est un principe très répandu et connu par beaucoup d'utilisateurs. Il permet d'ajouter des articles dans son panier sur une boutique en ligne. Dans le contexte du vote électronique, cela signifierait que l'électeur peut ajouter



les candidats et la liste qu'il veut à son panier. Un code-barres est alors généré, contenant l'ensemble du choix fait par l'électeur.
Cette solution simplifie beaucoup l'utilisation et la compréhension de la plateforme, le système de panier à commission étant très répandu sur Internet. Elle permet également les modifications en cours de route, ce qui n'était pas possible avec le système précédent (une fois que le code est scanné et confirmé, il ne peut plus être modifié).
L'inconvénient de ce principe réside dans le fait que le support (non sécurisé) affichant la plateforme a connaissance du choix réalisé par l'électeur.

5.3.3 Conclusion

La deuxième solution peut être améliorée de la façon suivante : à chaque modification du panier, un nouveau code-barres est généré. De cette façon, le support ne peut pas savoir quel code-barres à finalement été scanné.
D'autre part, l'intérêt d'un pirate réside plus dans le fait de modifier le résultat d'une élection que dans le fait de savoir ce qu'une personne précise a voté. A cela s'ajoute le fait que le pirate ne peut jamais être sûr que la personne qu'il espionne est bien la personne qu'il désire espionner. On n'est jamais sûr de qui se trouve derrière un ordinateur.
L'inconvénient présenté par la première solution est grand et risque de décourager l'utilisateur par sa complexité. Les inconvénients présentés par la deuxième solutions sont relatifs, raison pour laquelle, notre proposition est de tout de même utiliser le concept du panier à commissions.

5.4 Choix du type de codes-barres

L'interface entre la plateforme et l'appareil de vote consiste en un code-barres. Cette solution offre la possibilité de choisir le résultat d'un vote sans que l'ordinateur (ou le support non sécurisé utilisé pour afficher la plateforme) n'en sache rien.
Pour des élections, beaucoup de données doivent être encodées dans le code-barres. Pour cette raison, le code-barres conventionnel (code-barres à une dimension) ne convient pas, car il ne permet d'encoder que peu de caractères. Notre choix s'est donc porté vers un code-barres bidimensionnel qui a des capacités bien plus importantes.
Il en existe plusieurs types. Le premier se nomme Data Matrix et peut contenir jusqu'à 2335 caractères alphanumériques. Il a été développé à l'origine aux Etats-Unis. Son concurrent, créé au Japon, se nomme le QR-code (Quick Response). Il peut, quant à lui, stocker jusqu'à 4296 caractères alphanumériques.
Microsoft a également développé sa variante de code-barres nommé « High Capacity Color Barcode ». Elle offre l'avantage de pouvoir encoder plus de données que le QR-code sur une même surface, grâce à l'utilisation des couleurs. Ce type de code n'est cependant pas encore très répandu et le moyen mis à disposition par Microsoft pour créer des codes se base sur

un principe différent de celui des autres codes bidimensionnels. Les données à encoder sont placées sur un serveur, et un HCCB-code contenant l'url de ce serveur est généré. L'application pour téléphone portable recherche donc les données sur ce serveur après lecture du code-barres. Cette méthode n'est pas applicable pour notre projet, car l'appareil de vote ne doit pas communiquer à travers internet. Ce type de code n'entre donc pas en ligne de compte aussi longtemps qu'il n'existe pas de solution pour stocker les données directement dans le code-barres. Peut-être qu'avec le temps, Microsoft mettra un outil permettant de faire cela à disposition, mais pour notre projet, nous ne pouvons pas encore l'utiliser.

D'autres types de codes à deux dimensions existent encore, mais aucun d'eux n'a réellement réussi à percer. Notre choix s'est donc finalement porté vers le QR-code, car il permet de stocker plus de données que la DataMatrix. Ce code est également plus répandu en Europe.

5.4.1 Le QR-code

Le QR-code a deux paramètres principaux : la version du symbole (Symbol version) et la taille du module (Module Size). La version est définie par la quantité de données à encoder et le niveau de correction d'erreur désiré. Les versions sont numérotées de 1 à 40. Chacune définit le nombre de lignes et de colonnes du code-barres.
Le second paramètre est la résolution de l'imprimante (ou de l'écran) et celle du scanner, ou de la caméra. Celles-ci vont définir la taille du module qui est la surface en millimètres carrés de chaque point.
Pour ce qui concerne la correction d'erreur, il existe quatre niveaux de correction :
– L avec un taux de redondance de 7%
– M avec un taux de redondance de 15%
– Q avec un taux de redondance de 25%
– H avec un taux de redondance de 30%

5.5 Contenu des codes-barres

Pour notre application, le code-barres devra contenir un certain nombre d'informations générales en plus des informations dépendantes du choix du votant.
Ces informations sont les suivantes :
– l'identification de l'événement de vote
– la description de l'événement de vote
– la valeur $\hat{g}$ pour le système cryptographique selon El-Ghamal
– la ou les questions se rapportant aux objets de vote
– la signature des éléments cités ci-dessus
La signature citée en 5è place est la signature cryptographique de tous les éléments précédents. L'identification de l'événement de vote et la description de l'événement de vote sont

générés par l'Etat et seront contrôlés par l'Etat lorsque le vote sera placé sur le Bulletin-Board. Ces données ne doivent donc pas être modifiées par un hacker, sans quoi le vote ne sera pas valide.
La valeur $\hat{g}$ sera utilisée pour signer le vote. Cette signature sera contrôlée sur le Bulletin-Board. Il faut donc que le $\hat{g}$ ait la même valeur sur le Bulletin-Board et sur l'appareil de vote. Par conséquent, cette valeur ne doit pas être modifiée par un hacker, sans quoi le vote serait considéré comme non valable.
La question de l'objet de vote ne doit pas être modifiable non plus. Un hacker pourrait avoir l'idée de modifier la tournure de la question de façon à ce qu'on réponde par oui au lieu de répondre par non. Le votant serait ainsi trompé.
En signant numériquement chacun de ces éléments, on s'assure qu'il n'y a pas eu de modification.

5.5.1 Essais avec le QR-code

Pour faire un essai, nous avons pris les données suivantes :
- 20 caractères pour le numéro d'identification du vote
- 50 caractères pour la désignation du vote
- 320 bits pour la signature électronique du vote
- 1024 bits pour $\hat{g}$
- 90 personnes à élire (correspondant aux élections cantonales du Tessin)
- une moyenne de 20 caractères pour le nom des candidats et 10 caractères pour le numéro d'identification

Cela nous donne un moyenne d'environ 2940 caractères ce qui correspond à environ 23 000 bits.
Différents essais ont montré qu'avec un smartphone équipé d'une caméra de 3.2 méga pixels, la version 18 du QR-code était la limite. En-dessus de cette version, la caméra n'est plus en état de reconnaître rapidement le code-barres. Avec un niveau de correction d'erreur L, cela représente 5768 bits.
Par conséquent, une compression des données est nécessaire. Si celle-ci ne sera pas suffisante, il faudra prévoir de scanner plusieurs codes.

Pour la compression, la signature des données générales ainsi que la valeur $\hat{g}$ ne doivent pas entrer en ligne de compte. Comme ce sont des valeurs aléatoires, elles ne seront pas compressées davantage. Elles ne peuvent qu'influencer négativement la compression des autres données.

Une première approche a été réalisée avec la compression LZW. Elle permettait d'atteindre une diminution d'un peu moins de 35%. Cependant, nous n'avons pas trouvé de bibliothèque LZW permettant d'obtenir une bit string à injecter dans le code-barres.
Par conséquent, il a été décidé de se tourner vers une bibliothèque Zip (la compression Zip

utilise LZW) qui elle, permettait d'obtenir une bit string sous forme de bytes array.
Il faut savoir que le QR-code a été conçu pour recevoir des informations numériques, alpha-numériques, sous forme de bytes de 8 bits, ou alors de caractères Kanji (alphabet japonais). Il n'est donc pas possible d'injecter directement une bit string dans un QR-code mais cela doit se faire par l'intermédiaire d'un tableau d'octets qui eux sont compatibles avec le QR-code.
Du côté de l'application qui permet la lecture sur le smartphone, la lecture de bytes posait également des problèmes. Le résultat était uniquement livré sous forme de chaînes de caractères encodées au format ISO-8859-1. Nos superviseurs se sont donc occupés de contacter les développeurs de ZXing, afin qu'ils implémentent une solution à notre problème. Celle-ci devrait apparaître dans la prochaine mise à jour de l'application de ZXing.
Avec la compression Zip et la mise à jour du côté du décodeur, le problème devrait être résolu. Il faudra cependant tout de même compter avec le scan de plusieurs code-barres, car même avec une compression de 35%, on dépasse la contenance d'un QR-code de version 18.

Source des informations :
Documentation sur le QR-Code :
http ://www.denso-wave.com/qrcode/aboutqr-e.html
Documentation sur le HCCB-code :
http ://tag.microsoft.com

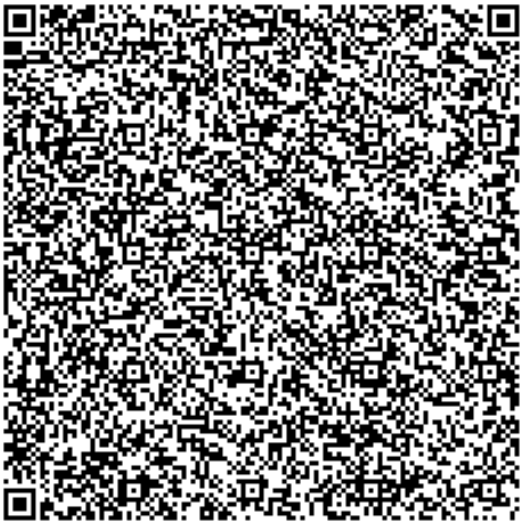
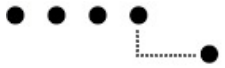


Figure 7 – Exemple d'un résultat d'une élection : Liste du votant, 9288 bits



5.6 Interfaces graphiques

Ce sujet a également présenté un vif intérêt auprès des personnes impliquées dans le projet d'e-voting. Cela a ses raisons. Les interfaces graphiques de la plateforme de vote forment la partie à laquelle l'utilisateur sera confronté. Elles doivent donc être les plus intuitives possibles pour que le votant comprenne rapidement le fonctionnement et ne soit pas découragé par la complexité de la chose.
Après beaucoup de réflexion, de discussions et d'idées, nous sommes arrivés au concept que nous décrirons ci-après. Les phases intermédiaires ne seront pas décrites ici, seuls certains aspects seront discutés plus en détails. Les schémas intermédiaires sont disponibles dans l'appendice.

5.6.1 Déroulement général

L'ordre des pages et le contenu a déjà provoqué beaucoup d'idées, pour finalement en arriver à la conclusion que l'utilisateur ne doit pas changer de page entre les différents votes. Tout doit être visible d'un seul coup d'oeil sans quoi le risque d'oubli pourrait devenir conséquent. Toutefois, il faut aussi que l'utilisateur puisse configurer quelques paramètres qui lui sont propres tels que son canton et sa commune d'habitation, sa langue, etc. D'autre part, l'aspect graphique joue également un grand rôle. Il ne serait donc certainement pas judicieux d'arriver directement sur la page de vote, mais plutôt sur une page d'accueil contenant un bref aperçu de quoi traite le site, ainsi qu'une bienvenue.
Notre idée est que l'utilisateur choisisse déjà la langue de l'interface et son canton de provenance sur cette page, mais cela d'une façon intuitive, c'est-à-dire en choisissant son canton sur une carte de la Suisse.

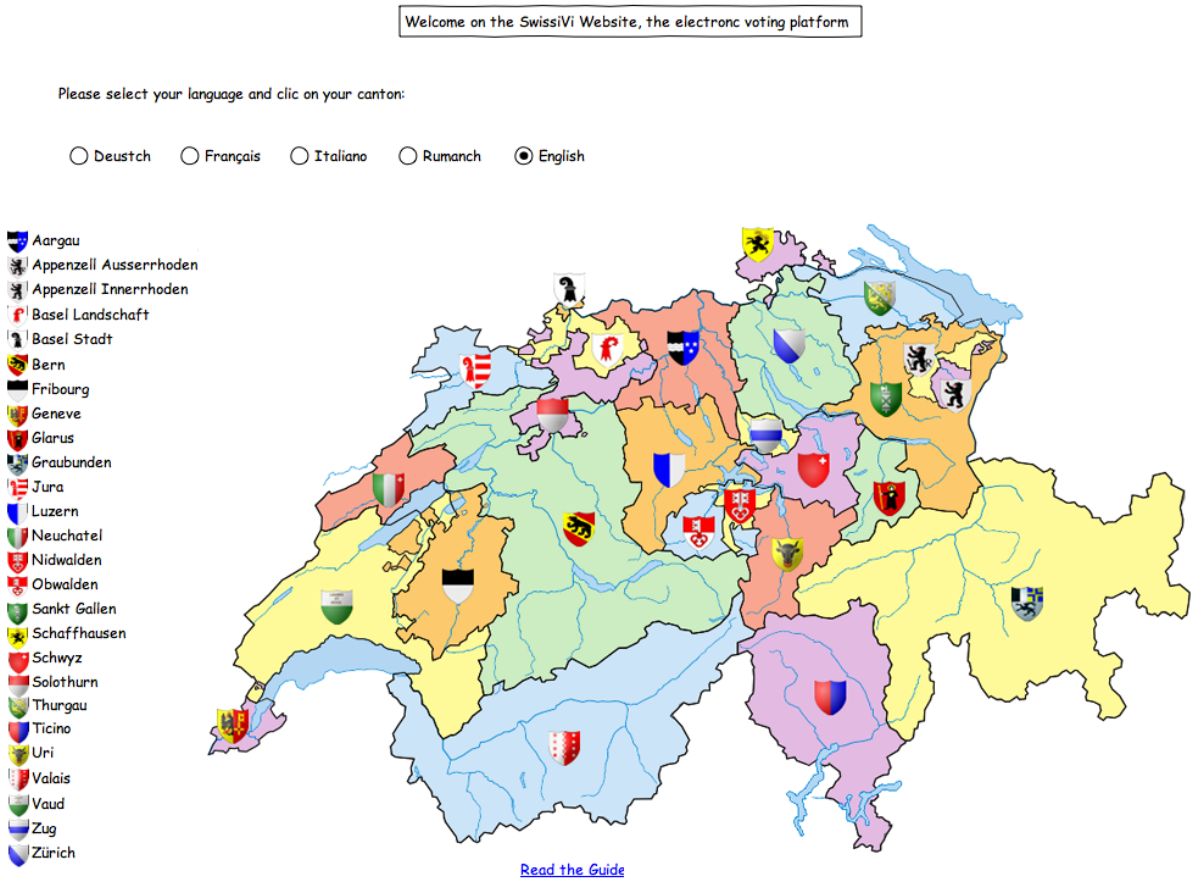


Figure 8 – Page d'accueil

Un fois que le canton a été choisi, la page de vote proprement dite s'affiche. Etant donné que l'utilisateur n'a toujours pas indiqué de commune d'habitation, un pop-up bloque l'accès à la page tant que cette information n'est pas fournie. Le nombre de communes en Suisse s'élevant à presque 2500, nous ne pouvons pas obliger l'utilisateur à chercher sa commune parmi une liste. Nous avons donc pensé à un champ à auto-complétion dans lequel l'utilisateur commence à taper les premières lettres du nom de la commune. Un certain nombre de propositions lui sont alors faites. Etant donné qu'il a déjà choisi son canton sur la page précédente, seules les communes du canton sélectionné lui sont indiquées. Il a toutefois la possibilité de changer de canton à l'aide d'un menu déroulant.

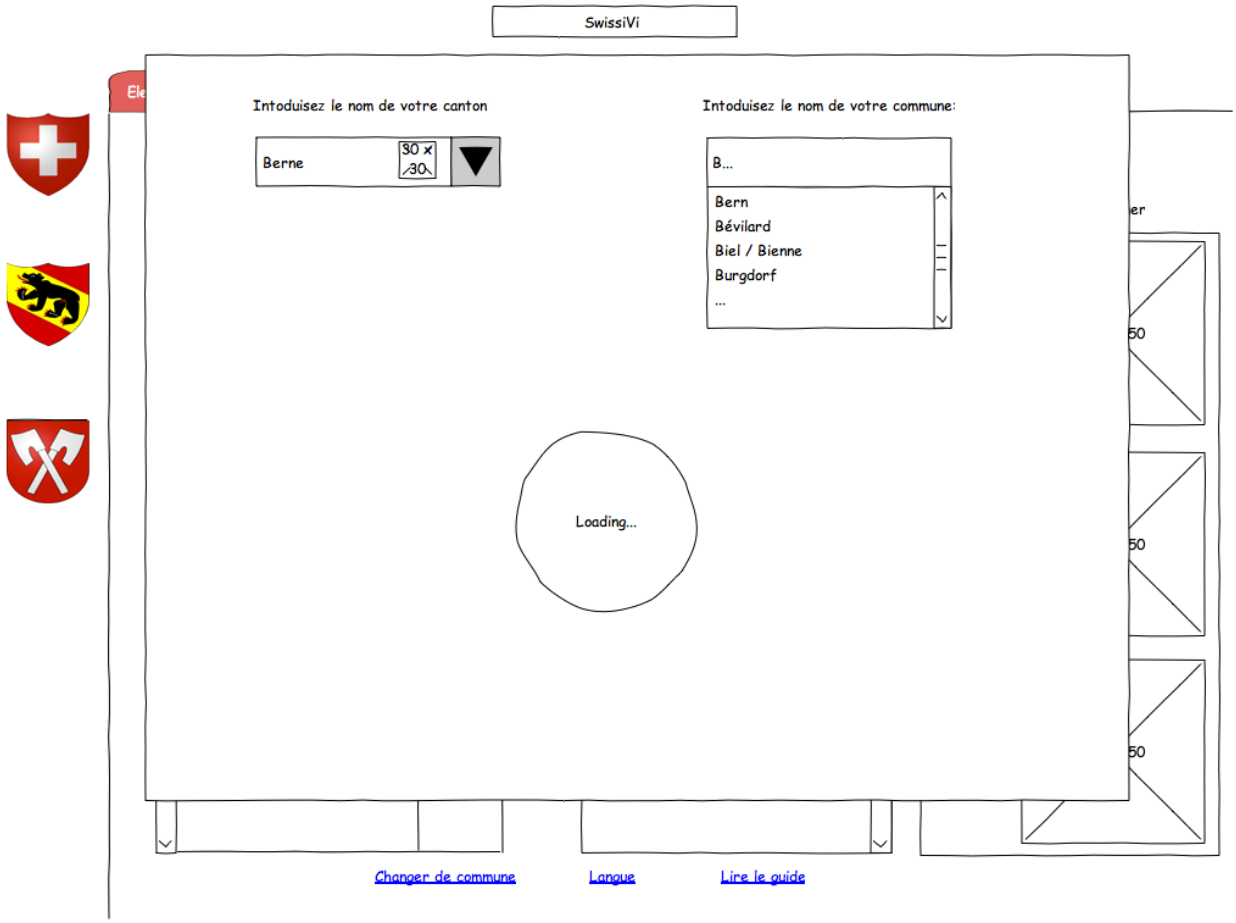


Figure 9 – Pop-up de sélection de commune

Pour éviter que l'utilisateur oublie certains objets, nous avons pensé lui afficher une vue d'ensemble de tous les éléments auxquels il doit donner une réponse. Pour conserver la structure actuelle des votes, les éléments sont ordonnés par niveaux : confédération, canton, commune. Cette vue d'ensemble s'affiche à l'intérieur du pop-up en dessous des champs de sélection de la commune et du canton.
Une fois tous les paramètres sélectionnés, l'utilisateur peut accéder à la page de vote en appuyant sur un bouton.

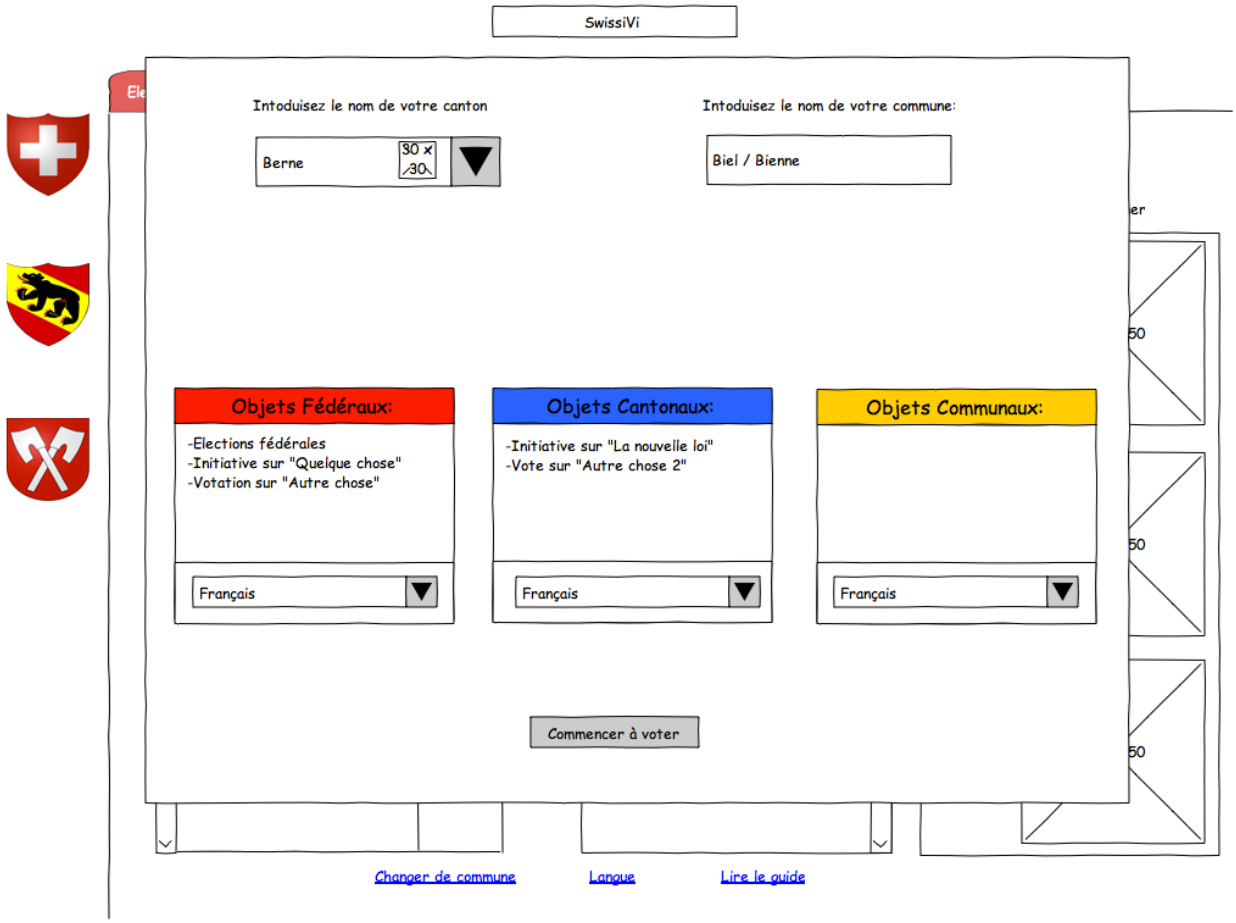


Figure 10 – Pop-up de sélection de commune avec vue d'ensemble

A ces paramètres s'ajoute encore le choix des langues. Il faut différencier la langue de l'interface (du site proprement dit) et celle des objets de votes. Prenons un exemple : un tessinois (ou plutôt un italophone) habitant dans le canton de Berne désire peut-être afficher le site en italien. Toutefois, les objets de votes du canton de Berne ne sont pas forcément traduits en italien, mais seulement en allemand et en français. Il doit donc pouvoir indiquer s'il préfère afficher les objets cantonaux en français ou en allemand. Ceci est paramétrable pour chaque niveau, fédéral, cantonal et communal en fonction des langues disponibles à ce niveau.

5.6.2 Interface de la page de vote et d'élection

La page de vote et d'élection est la page ou l'utilisateur va faire ses choix et scanner les codes-barres. Comme cité précédemment, il a été décidé que tous les éléments devaient se trouver sur la même page, ceci afin d'éviter à l'utilisateur de passer d'une page à l'autre et

d'avoir ainsi une meilleure vue d'ensemble. Une autre possibilité aurait été de guider l'utilisateur à l'aide d'une procédure « Suivant/Précédent » mais cette idée a été mise de côté, car elle offre moins de flexibilité à l'utilisateur pour passer d'un objet à un autre.

Afin de pouvoir afficher tous les objets de vote, il a été décidé de les répartir à nouveau dans les trois catégories classiques : confédération, canton, commune. Ces trois catégories apparaissent sur la gauche de la page. Dans chacune de ces trois catégories, plusieurs objets peuvent être disponibles. La navigation entre ces objets se fait à l'aide d'onglets situés au haut de la page.
Afin d'éviter que le votant oublie de consulter une des trois catégories, l'interface devra être construite pas à pas : d'abord la couche de la commune, qui sera ensuite masquée par la couche canton, puis finalement la couche confédération qui s'affiche par dessus les autres. Chacune de ces superpositions devra être visible par l'utilisateur, ainsi il sera conscient des différentes couches.

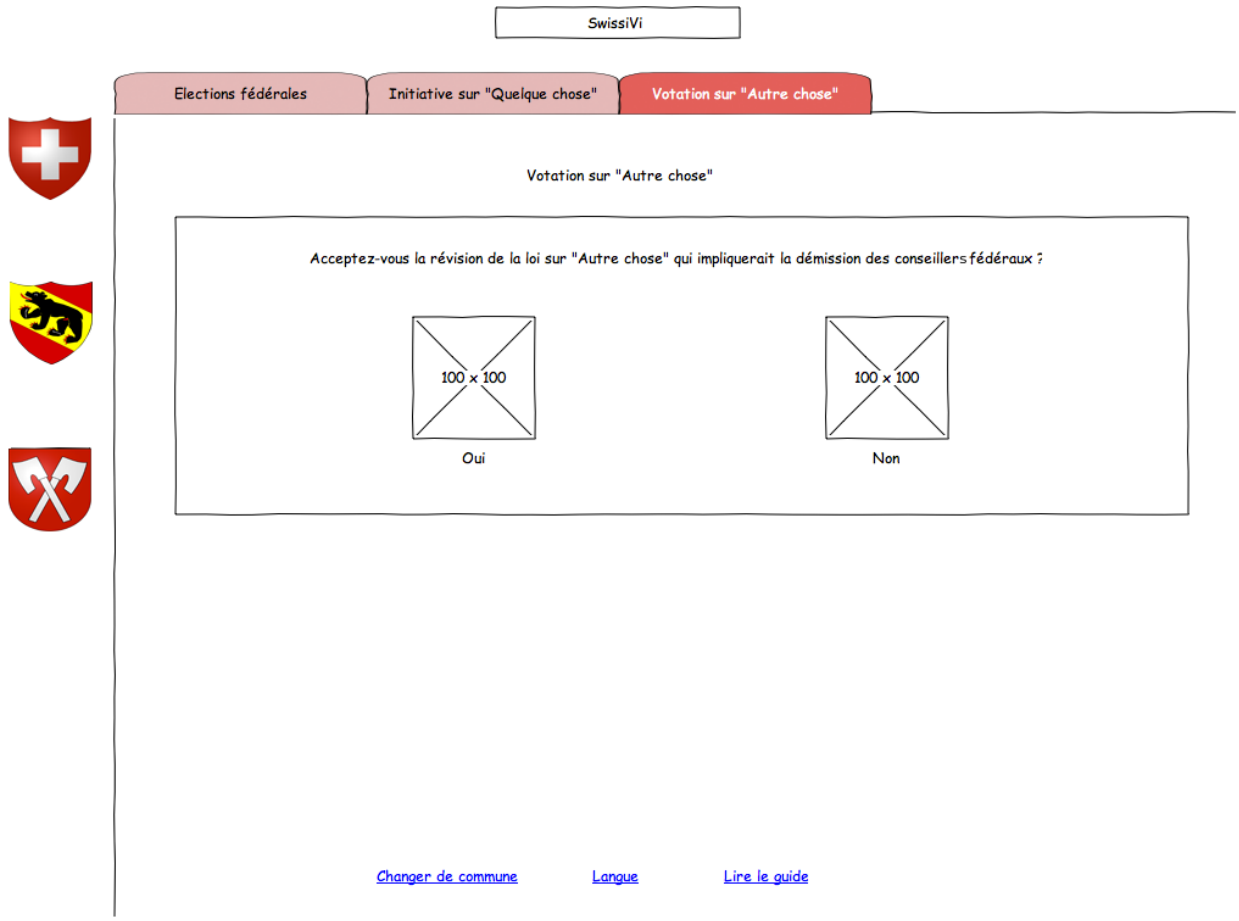


Figure 11 – Interface générale avec vote du type oui/non

Sur cette page, quelques liens permettent à l'utilisateur de modifier sa commune et son canton d'origine ainsi que la langue et d'afficher l'aide (dans un pop-up ou une nouvelle fenêtre) sans pour autant revenir à la page d'accueil. Le changement de la langue de l'interface nécessite, quant à lui, le retour à la page d'accueil.

5.6.3 Interface de l'onglet de votation du type oui/non

La partie représentant un vote du type oui/non n'est pas très compliquée. Il suffit d'afficher la question ainsi que deux codes-barres, un pour la réponse oui et un pour la réponse non (voir figure 11 ci-dessus).

5.6.4 Interface pour une initiative avec contre-projet

Pour la partie devant représenter une initiative avec un contre projet, il y avait deux solutions. La première reprenait la philosophie du vote de type oui/non, c'est à dire que pour chacune des trois questions, deux codes-barres étaient disponibles. Cette solution n'était toutefois pas satisfaisante, car elle oblige l'utilisateur à scanner trois codes-barres.
L'autre option est celle que nous avons choisie. Pour chaque question, un bouton radio apparaît où l'utilisateur peut sélectionner son choix. Le code-barres est ensuite généré dynamiquement en fonction du choix de l'utilisateur. Grâce à cette solution, l'utilisateur ne doit scanner qu'un seul code-barres.

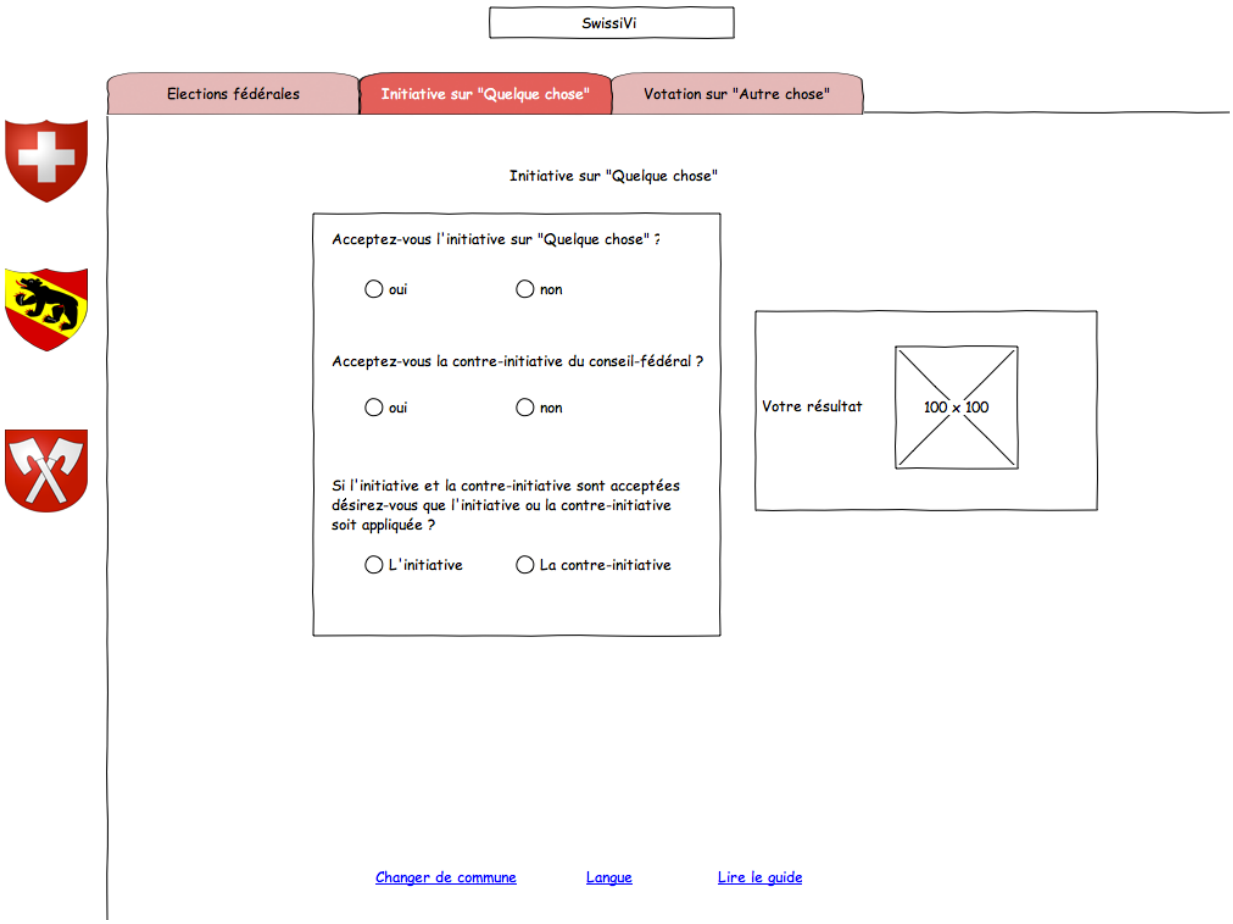


Figure 12 – Initiative avec contre-projet

5.6.5 Interface pour une élection

L'interface pour les élections a engendré beaucoup de discussions. L'idée de base était de créer trois parties : la partie avec les candidats disponibles, celle avec la sélection du votant, et la partie contenant les codes-barres. Cette idée est restée la même. Le problème a plutôt été de représenter clairement les manipulations à faire pour sélectionner des candidats. Comme nous l'avons cité dans un chapitre précédent, nous avons décidé de partir sur le principe du « Drag and Drop ».
Il faut savoir qu'il y a trois actions possibles, et en partie nécessaires, pour remplir un bulletin électoral. Premièrement, il est possible de choisir une liste complète. Deuxièmement, il est possible de choisir des candidats précis d'une liste. La troisième action n'est utile que lorsque l'électeur veut créer sa propre liste. Après avoir choisi ses candidats, il peut indiquer une liste à laquelle seront attribuées les voix qu'il n'a pas explicitement définies. La troisième action

est donc l'indication de cette liste.
Le premier et le deuxième point sont relativement compréhensibles. Si je tire le titre ou l'onglet d'une liste, toute la liste va être copiée. Si je tire un candidat, celui-ci sera copié. Cependant, si l'utilisateur tire une liste dans le but d'indiquer un numéro de liste à laquelle seront attribuées les voix qu'il n'a pas définies, comment faire pour différencier ce cas du premier ?
La décision prise après beaucoup de discussions a été d'utiliser la même manipulation que pour le premier cas en y ajoutant un message informatif. Si le votant tire une liste, un message est affiché lui demandant s'il désire copier tout le contenu de la liste ou s'il désire simplement indiquer la liste dans l'en-tête.
Trois boutons sont également disponibles en bas de la page :
– le bouton « Undo » annule la dernière modification
– le bouton « Redo » rétablit la dernière modification (après qu'elle ait été annulée)
– le bouton « Reset » vide la liste générée, après avoir demandé la confirmation de l'utilisateur.

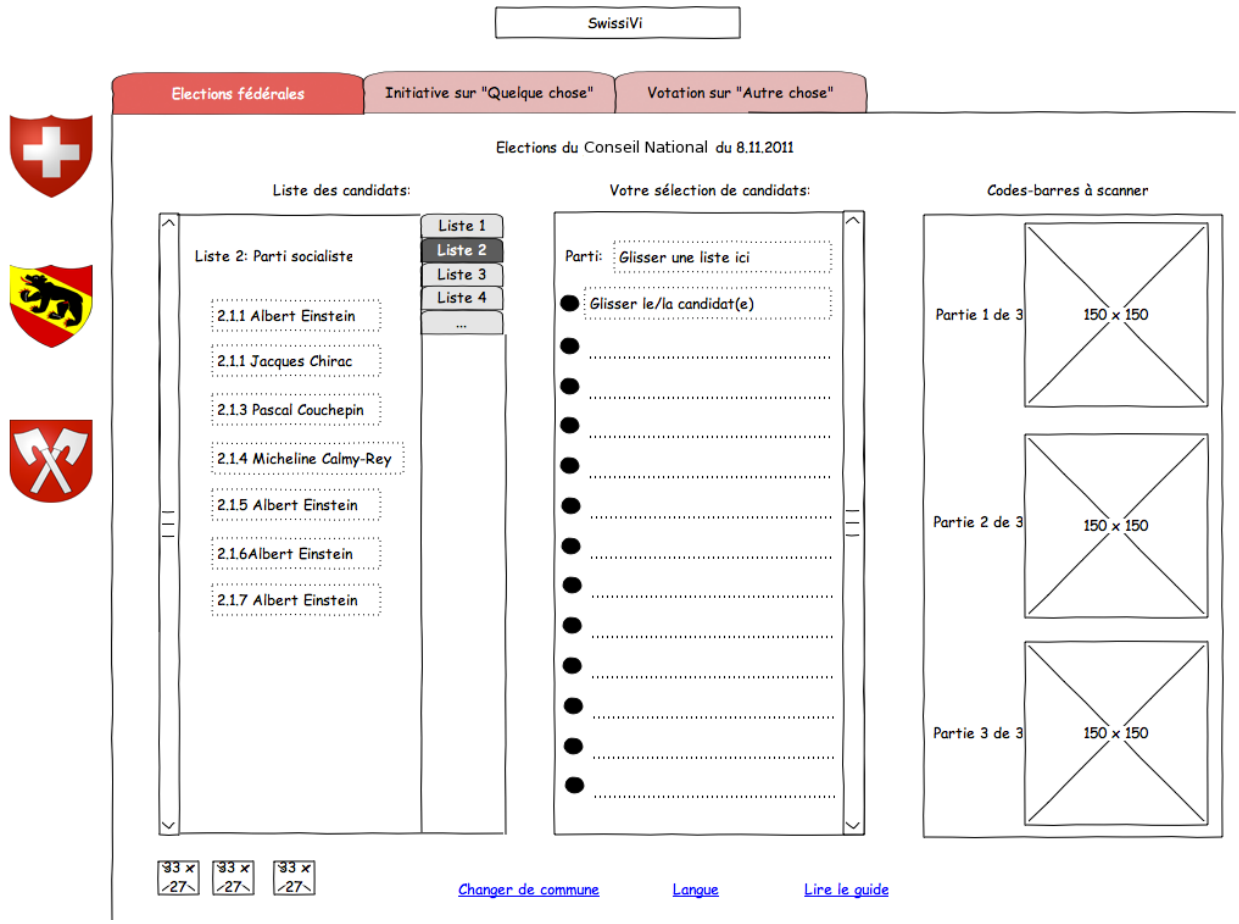
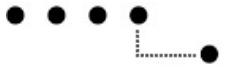


Figure 13 – Elections



Voici quelques règles complémentaires se rapportant au principe de Drag & Drop :
- une liste à puces indique le nombre total de candidats qui peuvent être choisis
- en tirant le titre ou l'onglet d'une liste et en le déposant dans la zone du bulletin vierge, le message dont nous avons parlé ci-dessus s'affiche
- si l'utilisateur choisit de remplacer les candidats, tous les candidats déjà choisis vont être remplacés même si la liste contient moins de candidats que le nombre de sièges disponibles, ceci afin d'éviter que l'utilisateur ajoute de façon inconsciente des candidats à la liste
- l'onglet d'une liste ne doit pas forcément être actif pour que la liste puisse être copiée
- si l'utilisateur glisse un candidat en-dessous d'autres candidats déjà choisi, le nouveau candidat sera ajouté à la suite des précédents et non à l'endroit ou l'utilisateur relâche la souris. L'ordre des candidats ne joue pas de rôle en Suisse
- si l'utilisateur glisse un candidat entre deux candidats précédemment choisis, un espace se crée entre ces deux candidats et c'est à cet endroit que le nouveau candidat sera copié. Cela permet à l'utilisateur d'organiser son bulletin comme il le désire, même si l'ordre des candidats ne joue pas de rôle
- pour effacer un candidat de sa sélection, l'utilisateur doit le tirer hors de la zone du bulletin vierge
- le remplacement de candidat n'est pas possible. Il faut d'abord effacer l'ancien candidat puis glisser le nouveau, ceci afin d'éviter des remplacements non désirés
- un message est affiché si l'utilisateur essaie de glisser plus de candidats qu'il y a de sièges L'utilisateur devra donc d'abord effacer un ancien candidat pour pouvoir insérer le nouveau. La variante de l'effacement automatique du dernier candidat pour ajouter le nouveau a été rejetée, car ce n'est pas forcément selon les désirs de l'utilisateur
- dès qu'un candidat apparaît dans la liste, les codes-barres sont générés
- lors de l'établissement d'une liste « à la main », l'indication d'une liste et du nom d'un parti n'est pas obligatoire
- le changement de commune réinitialise la liste des candidats, le changement de langue pas

5.6.6 Interface d'administration

L'interface d'administration sera utilisée pour créer de nouveaux objets de votes ou de nouvelles élections. Dans le cadre de ce projet, l'interface d'administration se limitera à une page de login et une possibilité d'uploader un fichier XML contenant les informations de l'objet à ajouter. Ce fichier XML sera ensuite décomposé et les données seront sauvegardées dans une base de données. Il s'agira donc de définir une syntaxe XML à utiliser pour les votations et les élections.

5.7 Choix des langages de programmation

Le principe de base est que le code doit s'exécuter du côté client afin d'éviter les requêtes au serveur. Seulement lors de la construction de la page les requêtes sont admises (et nécessaires). Pour la sélection de candidats et la génération des codes-barres, il ne doit pas y avoir de requêtes au serveur pour des raisons de confidentialité.
Deux principaux langages de programmation entrent donc en ligne de compte pour la plateforme de vote : le javascript avec les bibliothèques jQuery et jQuery UI, ou Google Web Toolkit. L'avantage du second est qu'il offre la possibilité de coder en Java. L'avantage du premier réside dans le fait qu'il est plus facilement modifiable au niveau du design. Dans les deux solutions, les outils de drag & drop sont déjà proposés, mais il faudra bien sûr les adapter. Notre préférence se porte tout de même vers jQuery, car il est plus facile d'y modifier certains paramètres.
Bien sûr, le langage HTML et le CSS seront également utilisés pour la représentation graphique des éléments.

Pour ce qui est du langage du côté serveur, avec accès à la base de données, nous avons pensé utiliser du PHP, car il ne s'agit pas ici d'un très grand projet du côté serveur.

A Versions antérieures des interfaces graphiques de la plateforme de vote

Cet appendice rassemble les deux premières versions des interfaces graphiques. La première version est présentée telle quelle. Dans la seconde version, quelques commentaires détaillent l'évolution par rapport à la première. La version finale a déjà été présentée au chapitre 5.6 et ne sera donc pas reprise ici.

A.1 Version 1

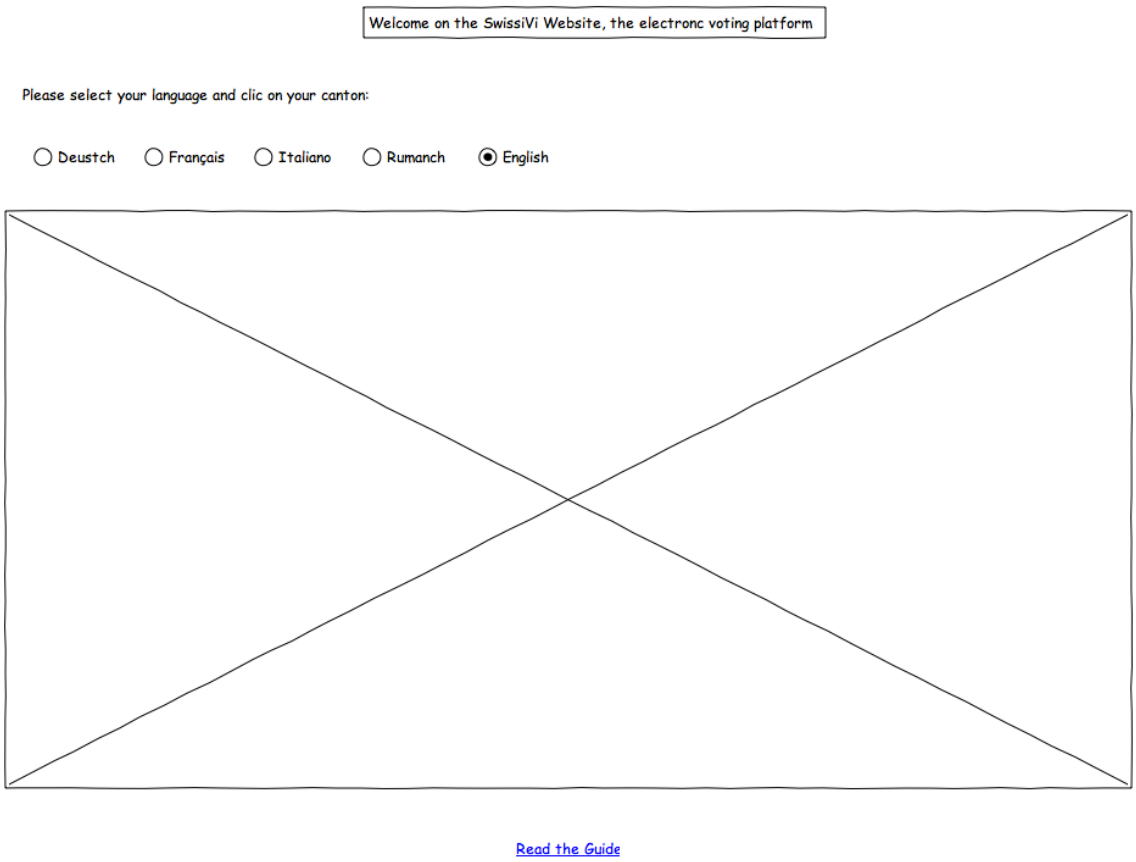


Figure 14 – Version 1 : Page d'accueil

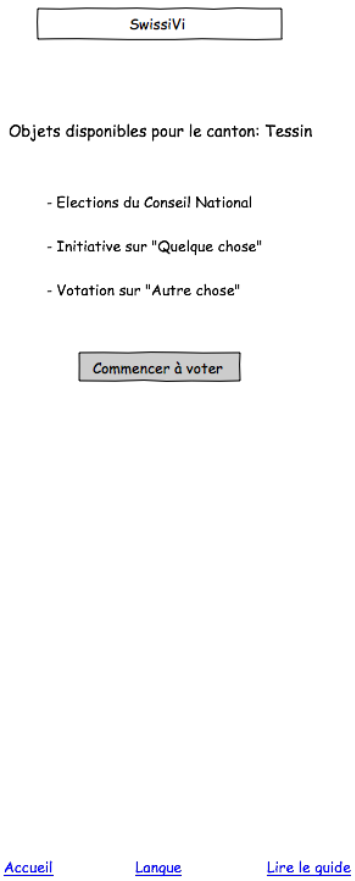


Figure 15 – Version 1 : Page avec vue d'ensemble des objets de vote

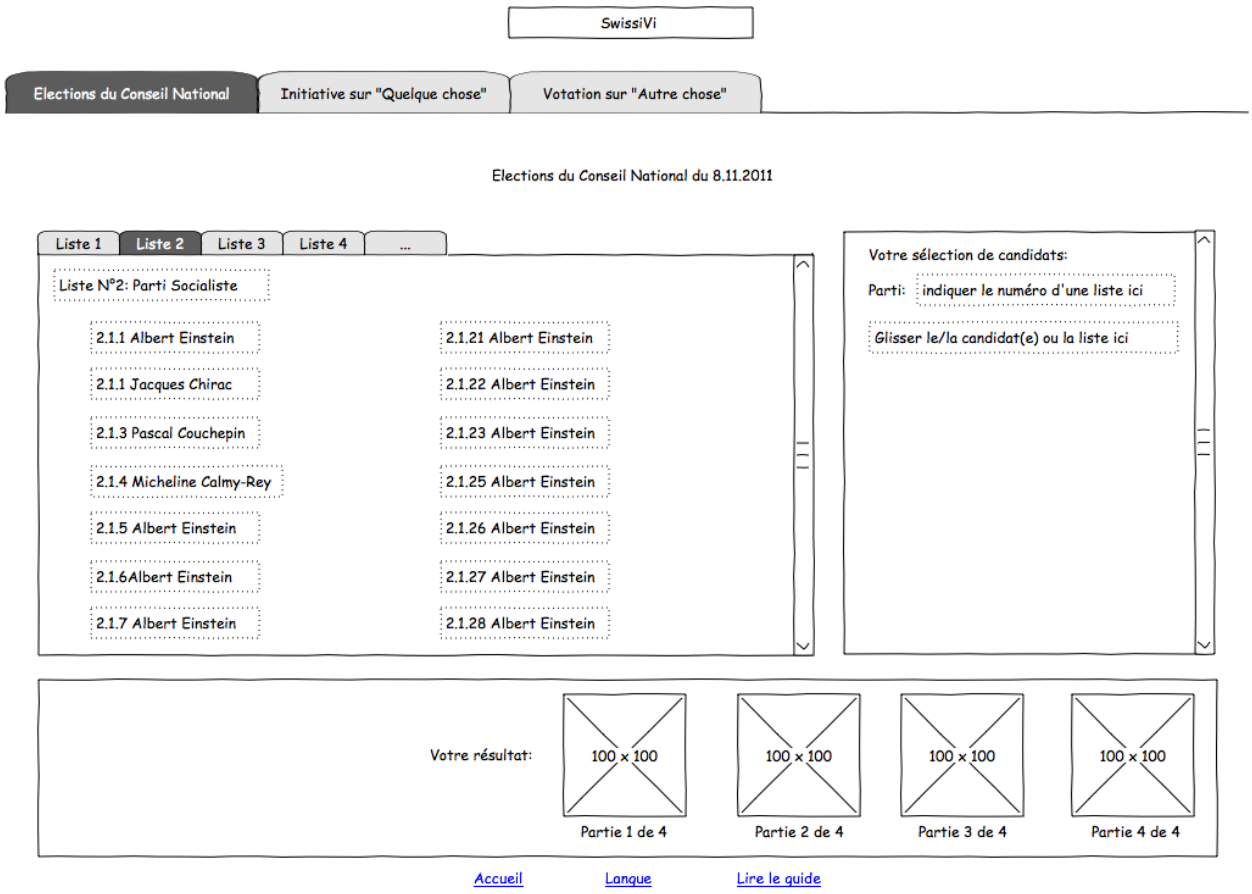


Figure 16 – Version 1 : Page pour une élection

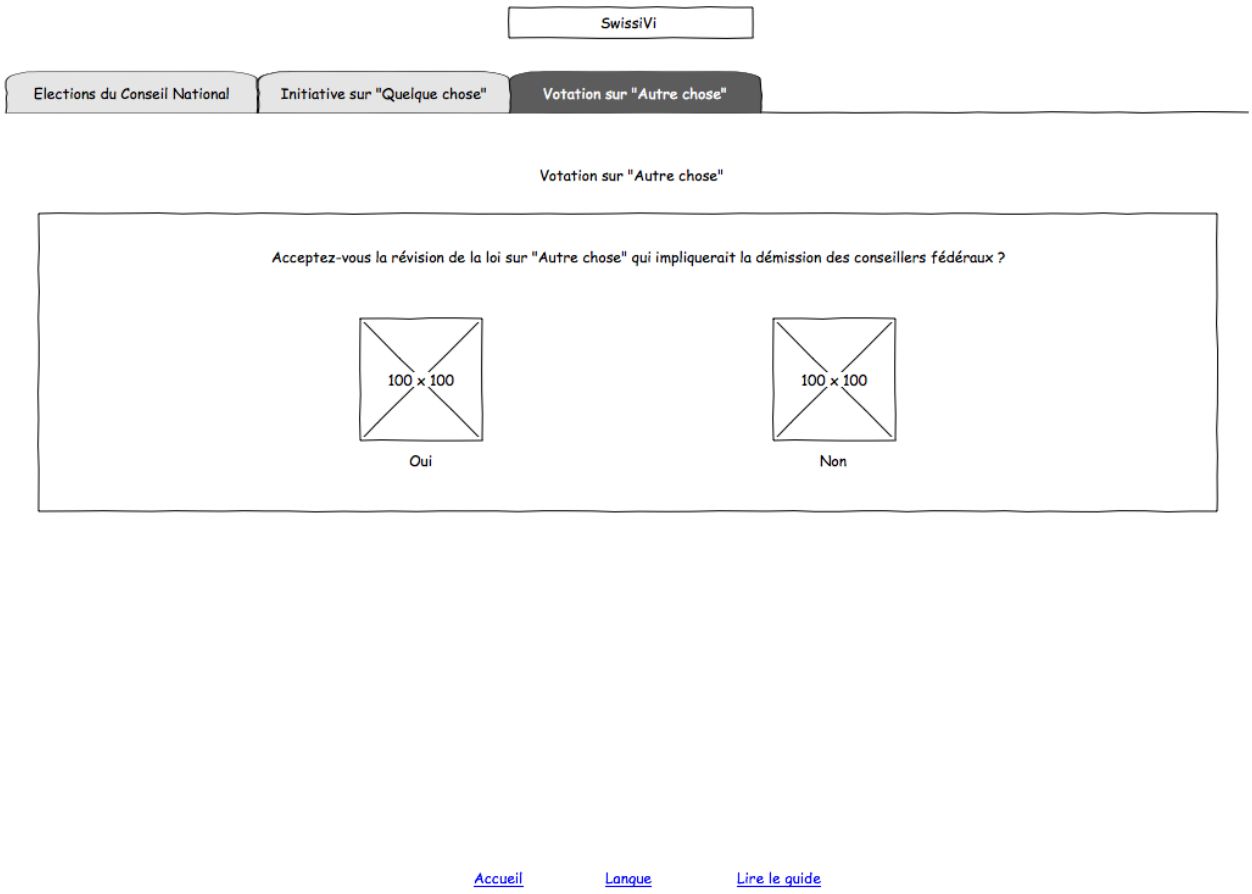


Figure 17 – Version 1 : Page pour une votation

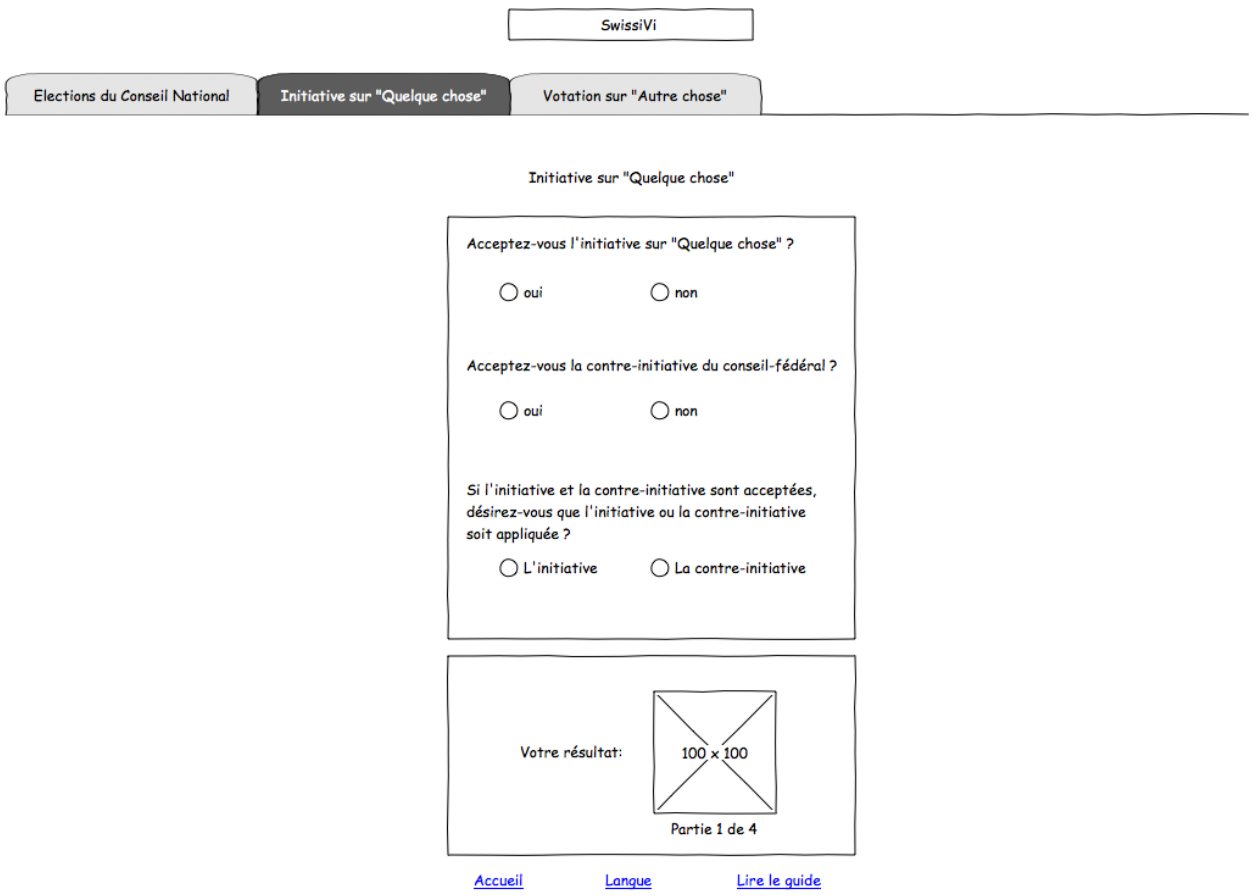


Figure 18 – Version 1 : Page pour une initiative

A.2 Version 2

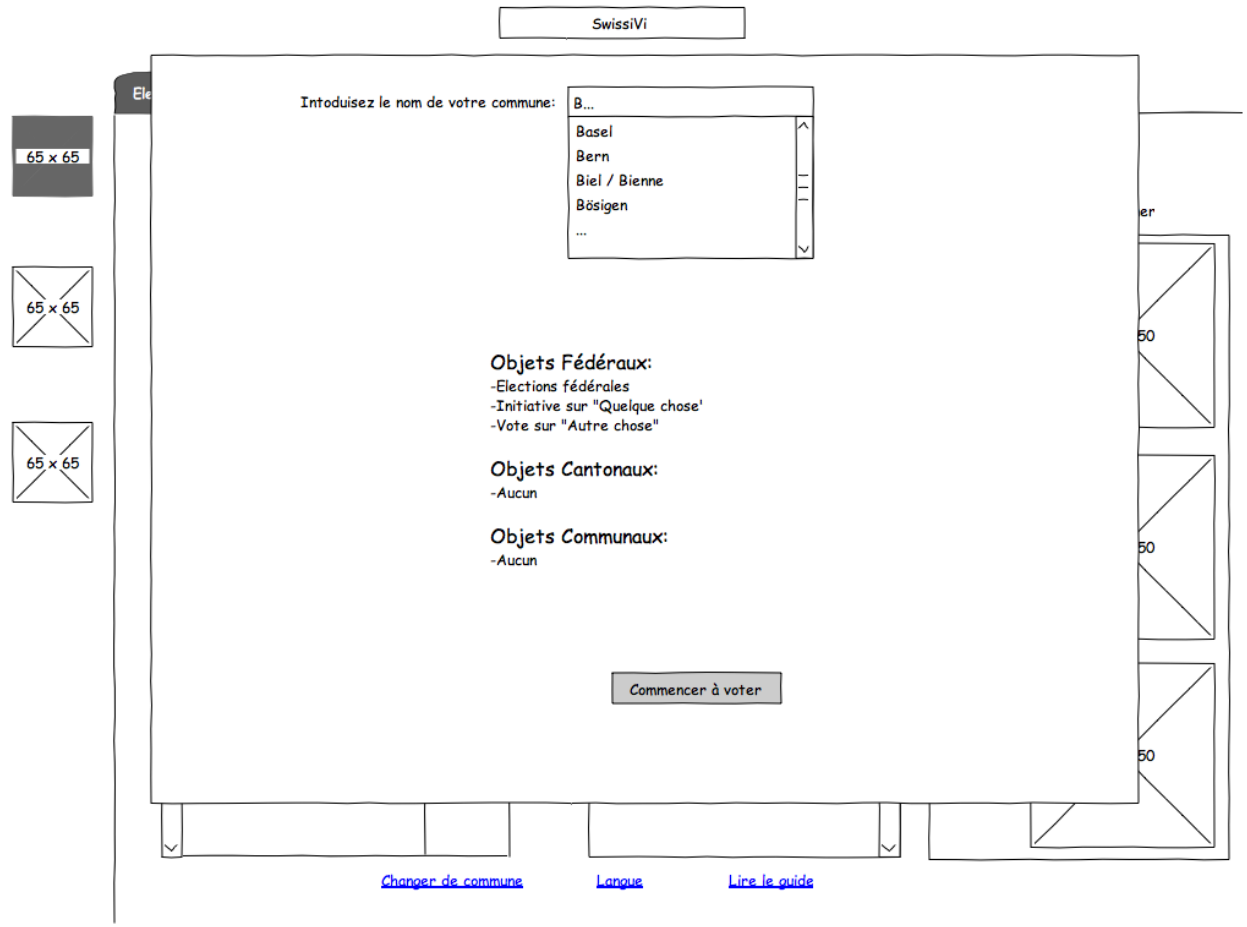


Figure 19 – Version 2 : Page avec vue d'ensemble des objets de vote pour la confédération, le canton et la commune

Le pop-up ci-dessus remplace la page avec la vue d'ensemble des objets de vote pour le canton sélectionné (figure 15). On y remarque l'ajout du champ de la commune. Ceci est fait à l'aide d'un champ à auto-complétion dans lequel l'utilisateur commence à taper les premières lettres du nom de la commune.

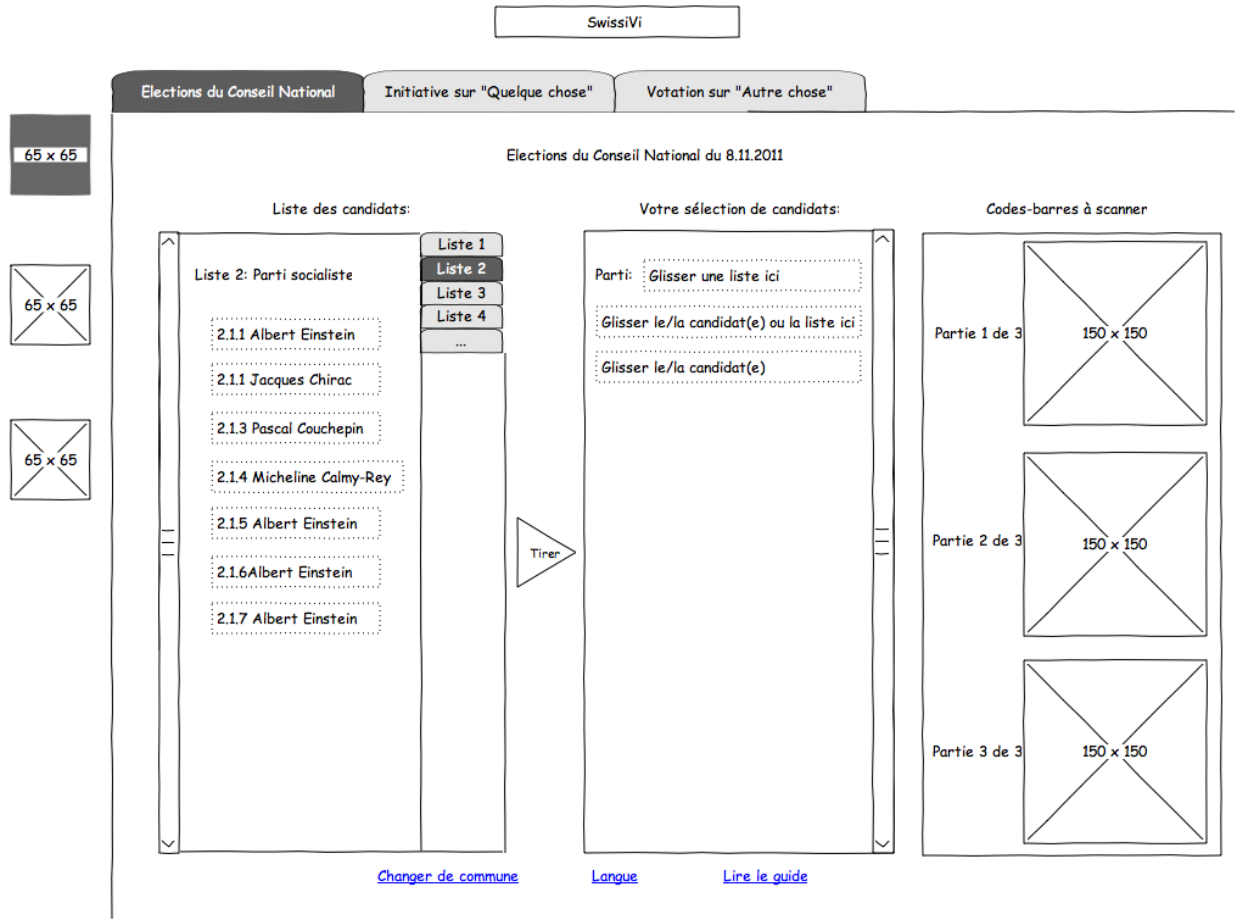


Figure 20 – Version 2 : Page pour un objet élection

Cette page est la mise à jour de la page des élections (figure 16). On y remarque l'ajout des trois niveaux, fédéral, communal et cantonal.
Le positionnement des éléments de la page a également été retravaillé afin de tirer profit de la largeur de l'écran plutôt que de sa hauteur.

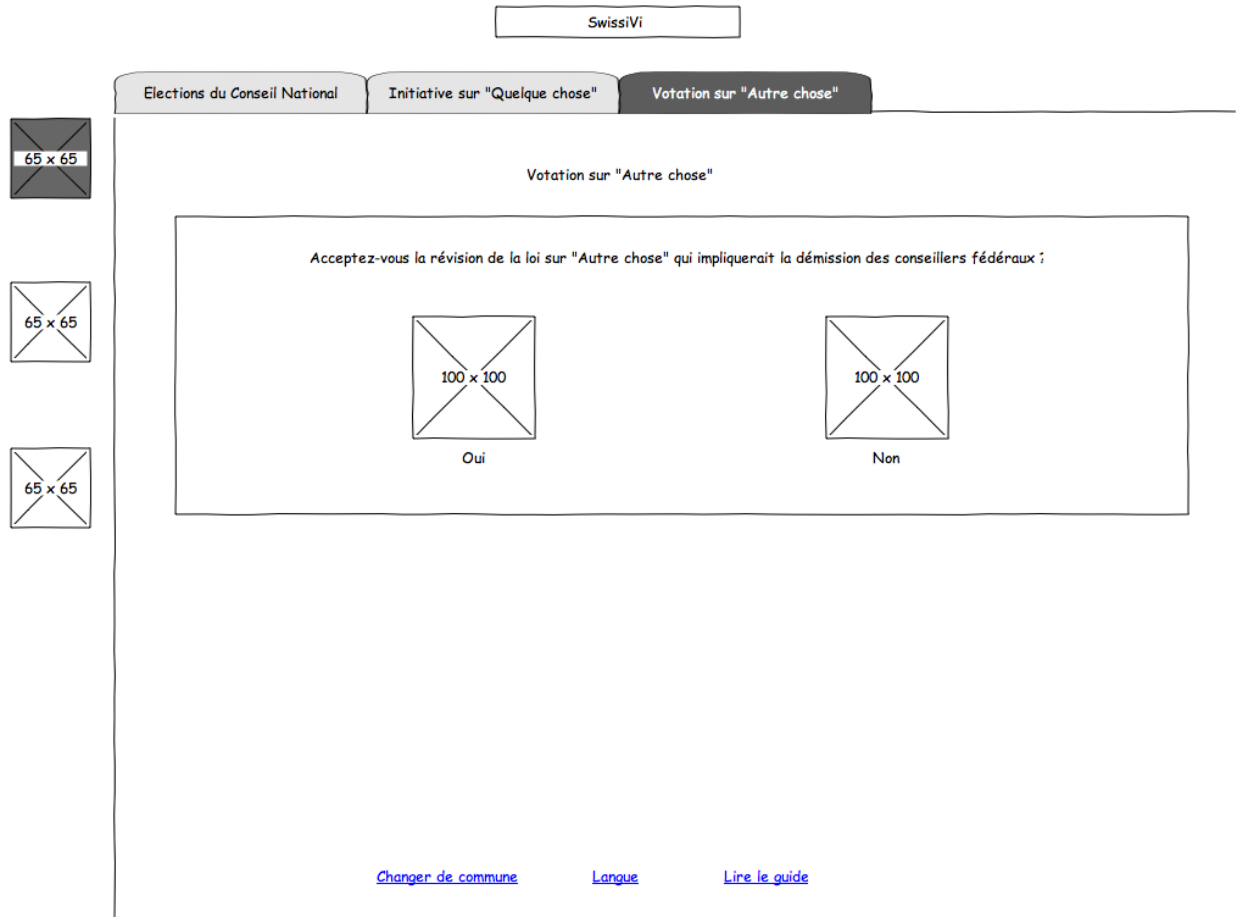


Figure 21 – Version 2 : Page pour une votation

La seule modification apportée à cette page par rapport à la version précédente (figure 17) est l'ajout de la navigation en trois niveaux sur la gauche.

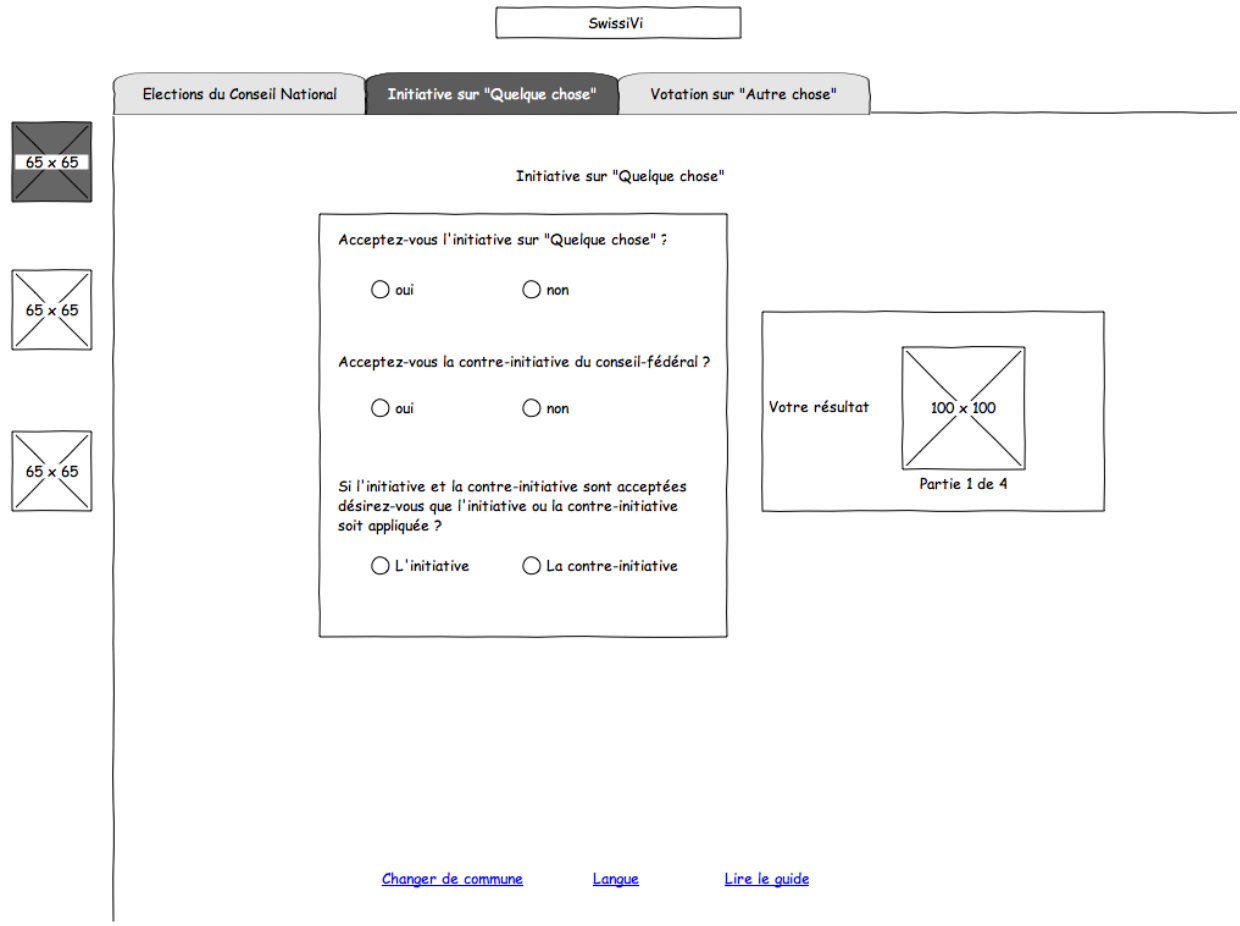


Figure 22 – Version 2 : Page pour une initiative

Cette page est la mise à jour de la page des initiatives (figure 18).
On y remarque de nouveau l'ajout de la navigation en trois niveaux ainsi que le repositionnemnt du code-barres.