E-vote 2011

Contractor Solution Specification

Project: E-vote 2011

Change log

Version	Date	Author	Description/changes
0.1	26.10.09		First version

CONTENT

1. Tenderer Solution Proposal

1.1. Overview of Proposed Functional Solution

For the first prototype, we focused our efforts on the compliance with the zero trust requirement. The zero trust requirement means that the e-vote solution cannot rely on the voter's PC security so the encryption of the vote cannot be done in the voter's environment (where a virus or Trojan can modify the content before the encryption takes place). Therefore, we focused our efforts in looking for alternatives to how to cast the vote (understanding that the rest of the process would not change –substantially- from the one implemented by our internet voting product).

We wanted to come up with a new solution for the voter to cast the vote. The "encrypted ballot" solution seemed unfriendly to the voters, so we looked for a different approach. We discarded the "images based interface" due to accessibility issues and some small security issues.

Finally, we presented to you a solution combining three channels: postal, internet and phone. This solution overcomes the issues related to the zero trust requirement. However, it raised two security issues:

- First, the vote needs to be decrypted in the IVR server in order to be read to the voter
- Second, the system reads the content of the vote to the voter, which may raise concerns about vote privacy.

To minimize the first issue we defined a process so the vote decryption and the server private key are only memory information, so the private information is never stored in a persistent device (except the private key of the server which is in the HSM). However, we still have a small thread of an internal attack, but it can be minimized to a residual risk through server administration policies and processes.

The second issue was more difficult to solve, but we did some internal tests and what was perceived by our security experts like an issue was not raised by any of our "test" voters, so we thought that maybe it was not a real issue after all. But anyway, those issues were relevant for you.

We have carefully assessed your comments raised in the evaluation of that solution. For this second prototype we started working in a new solution again. After some work we decided to offer you two different solutions, the main difference between them is the use of an IVR server in one of them and not in the other.

Non IVR Solution: We will insert a green box where there are differences between our two solutions.

First of all, we decided to remove the security matrix as we realized that it was not a familiar solution for the voters. The solution is based on two different personalized data that is printed in each voter card: a unique identifier number (the voter ID) and the voter's encrypted ballot codes (a 4 digit code for each party contesting the election. The list of party codes is unique per voter).

Security comment: The voter ID will be created by the EMIS as part of the electoral roll process. It will have one control digit. The private list of party codes will be created in the isolated environment for each voter (using random algorithms) and stored there.

The voter interface will not change substantially from the one presented in our previous prototype (some usability changes and improvements, but using the same philosophy). The big changes are on the last page where we no longer ask for the security matrix position and neither for the telephone number. Instead, we ask the voter to insert the party code of the party he/she has already selected in the web interface. The system will show a final page asking the user to call a free number from the Ministry in order to confirm the vote. When the voter calls that free number, the IVR will ask the voter for his voter ID printed on his voter card, and will tell the voter the party code recorded in the web interface. Therefore, the voter can confirm that his vote has been recorded properly (this solution can allow the Ministry –or the owner of each process- to make this incoming call optional or mandatory as preferred).

Non IVR Solution: In our proposal without IVR, when the voter casts his vote, the web interface requests his voter ID instead of asking for the party code. The system stores the vote and shows the voter a confirmation page with the party code of the selected party. This way the voter can check that the vote has been properly stored by checking the party code on the screen with the information printed on his voter card.

What happens in the server once the voter sends his/her vote through internet? Let's review it for each proposed solution:

In the IVR solution

- The server receives from an identified voter (identified by his/her eID through SAML integration with the identification server – and therefore by the voter ID in the electoral roll database) a vote (casted through the web interface) and a party code (inserted in the last page).
- The system creates a hash for those components (voter ID, selected party and party code) and compares it with a list of valid hashes for that voter (as generated in the isolated environment when the party codes are generated for each voter).
- If the hash matches any of the valid ones in the voter's hashes list (that means, the party selected and the code inserted matches), the system encrypts all the information with the public key of the election and stores it in the solution database digitally signed with a application server private key (created by the administrator in the HSM and stored there).
- The system also stores the party code (not encrypted) and linked to that voter ID, so that the IVR can read it to the voter once he/she has inserted the valid voter ID.

All the votes are registered in the database, signed by the application server and encrypted using the public key for the election. Once the election process is closed, those signed and encrypted votes are moved to the isolated environment, where the vote is decrypted (once the election private key has been rebuilt) and using the only copy of party codes and voters ID tables, the votes are confirmed.

In the NON IVR solution

The server receives for an identified voter (identified by his/her eID through SAML integration with the identification server – and therefore by the voter ID in the electoral roll database) a vote (casted through the web interface). The system checks that the voter ID inserted in the web and the one related to his/her eID identification matches (ensuring that the voter has his/her digital authentication and his/her voter card), the system then accesses the vote to get the corresponding party code from the tables of parties and voters exported from the isolated environment and sends it back to the voter (so he/she can confirm that his/her vote has been correctly stored with the correct intention)

Security comment: The main difference between the two solutions is where the party codes information for each voter is stored. In the first solution the information is only available in the isolated environment (and of course, on the voter cards) and only the hash of the party code plus the party plus the voter ID, is exported. In the second solution the information needs to be stored in the solution architecture (of course it will be stored encrypted, and with the maximum security levels, but in the solution server architecture).

Whichever the solution, the votes are stored in the system infrastructure encrypted and linked to a voter ID. Once the p-voters have been marked in the electoral roll, the system removes those e-votes from voters that have been marked as p-voters, and transmits the rest to the removable storage device to be transferred to the isolated environment.

Once the votes are transferred to the isolated environment the process continues as explained in our product description before, and from the isolated environment, back to the system servers to consolidate results.

The system server mixes the e-votes (from the isolated environment) and the p-votes (from the e-counting or the manual count) results to proceed with the next election steps.

Security comment: To account for potential court petitions, we suggest that a backup of each election process Database, the tokens to rejoin the election private key (with their passwords) and a back up of the isolated environment should be kept by the election administrator in order to allow a judge to restore all the information to allow a voter to know how his/her vote was counted in the election. The system administrator should rerun all the process with only the voter's vote in order to get the information to the judge.

Security review:

Data security

- Encryption – All election data shall be stored and remain encrypted using the combination of RSA 2048, AES 256 bit and SHA2 algorithm.
- Isolated server environment – This isolated server shall serve as the only environment where e-votes can be decrypted, validated and counted. In addition, in this environment will take place the generation of public and private election keys as well as calculation of the random parties' codes and the hash value of all party codes to be exported to the system infrastructure.
- Secret sharing algorithm – To prevent any unauthorized and single person from decrypting the e-votes, a secret sharing algorithm will be used to divide the private key into several parts and distributed to a number of authorized election officers. This would require some or all of the elections officers (or a selected combination of them) to agree and combine their parts of the secret key before a decryption process can be executed.
- Voter unique party code - To protect the integrity of the vote, a unique party code shall be generated for every voter. A voter shall be provided a unique party code that

corresponds to the voter. A hash calculation mechanism is then implemented to validate every party code a voter voted for (optional).

- IVR (Interactive Voice Response) – An IVR system shall be implemented to enable the voters to confirm the vote cast and ensure that the vote has not been manipulated and that the right vote has been recorded

Logical security

- Server, Routers and Switches Hardening - The proposed solution shall include a server hardening process that will involve the creation of a baseline for the security on all election system platforms including the PCs to be used on polling stations (controlled environment).
- Perimeter security – To prevent security breaches coming from outside and inside the elections system infrastructure, a robust perimeter security shall be implemented and are as follows:
    - DMZ Network and Network Segregation
    - External network firewall
    - Internal network firewall
    - Intrusion prevention system
    - DOS/DDOS mitigation hardware
    - VPN connection for connections coming from polling stations
    - SSL communication channel between the voter and the election system coming from uncontrolled environments

Application security

- *Source code fortification:* To prevent application related security threats such as SQL injection, Blind SQL injection, Cross-site-scripting, Phishing, Cross-site-request-forgery, Link injection, etc, the election system shall implement a source code fortification process prior to deployment. A rigorous testing of all source codes will be executed to identify and resolve badly coded source codes as well as those that contain hazardous characters. Additional measures such as filtering user inputs will also be implemented to prevent an attacker to input malicious parameters that could possible compromise the election application system. In addition, this process shall address all application threats mentioned in Open Web Application Security Project (OWASP) and Web Application Security Consortium (WASC).
- Application firewall: to ensure that only authorized http methods are executed on the application level and hazardous methods are blocked. It employs an auto-adaptive approach to application delivery security, where the security policy is automatically updated based on observed traffic patterns. This simplified approach to configuration prevents the possibility of application level attacks such as XSS, SQL injection, CSRF, Link injection caused by badly coded source codes.
- Application server hardening: A hardening process shall be implemented on the application server level. The hardening process shall involve the creation of a baseline for the security on all election application system. To effectively protect the application on the application server level, a solid and sophisticated security baseline for all application servers in the election system shall be implemented, that includes removing unnecessary services, disabling of default user, guest and admin accounts and to ensure that all accounts requires authentication and a strong password to access the application server

Access control

- Logical access control: The proposed solution shall implement access controls to restrict access to election systems to authorized personnel only.

- *Role-Based Access Control and Separation of Duties*

  To control access to the election system and to prevent one person to hold a role to execute critical tasks, the election system shall implement Role-based access control. In RBC, the assignment of permission to perform a particular operation in the election system is meaningful, because the operations are granular with meaning within the election application. RBAC has been shown to be particularly well suited to separation of duties (SOD) requirements, which ensure that two or more people must be involved in authorizing critical operations. An underlying principle of SOD is that no individual shall be able to execute a breach of security through dual privilege. By extension, no person may hold a role that exercises election systems related tasks over another.

  The same principle of access control applied to the election application shall be implemented to all other devices in the election infrastructure. Default system administrator accounts shall not be used on a day-to-day operation instead a user account with configured rights and permission according to its role and tasks in the election system shall be provided.

Audit

- *Log and Audit management:* The proposed solution of Indra offers a robust log management method to ensure that an end-to-end transaction is logged and documented and shall be verifiable for whatever purpose in the future. The logging will ensure that non private information is never logged, so will always maximize the privacy of the vote.
  - *Audit Mechanism:* An audit system shall be implemented to be able to verify and confirm all transactions and activities involving the whole election system. Auditors can easily perform audit functions using the data generated by the logging mechanism of the election system in all levels
  - *Event log* –The security solution shall implement logging mechanism on all levels of the whole election system, such as, network activities (source, destination, ports, services, etc), election application transactions (authentication, recording, counting, etc)
  - *Event log monitor* – A monitoring system shall be implemented and shall continuously run on the background on all levels of the election system to be able to detect in real-time any anomalous behavior and shall send alarms or information of such anomalous behavior to authorized personnel immediately

1.2. Functionality included in the Proposed Solution – free of charge for the Principal

Our proposal includes electronic poll books to be used in the advance polling stations and the Election Day polling stations. These EPBs (workstations) will work with or without connection to the server, but will allow the system to get access to the real list of p-voters in a much faster way than processing the electoral roll books coming from each polling station (a manual task to read the books and mark the voters in the system to prevent vote duplication will take a long time).

2. Elaboration of General Requirements

Elaboration of Requirement GR3.10
Indra will provide the necessary services to provide support to the customer for: user training, system support and configuration of the central system.

1. **User training**

   Indra propose to deliver the training courses for the users groups of the following organizations: KRD, the municipalities testing the system in 2010, the municipalities piloting the system in 2011, representatives of the political parties (functionality related to submission of lists proposals only).

   The training courses will be focused on the use of the different system modules from a user's point of view. They will be supplemented with a general overview of the processes and the associated roles and responsibilities.

   Indra suggests the training programme to follow a "train the trainers" strategy. We recommend that the maximum number of attendees per training session should not exceed 15.

   The training sessions will be divided into presentations and hands on sessions. Indra will set up the training centre in Oslo. This training centre will be equipped with the necessary installations to be able to carry out the hands-on sessions of the training courses on the exact replica of the real environment (including a small scale e-counting centre). In addition, users will be able to log into the same system remotely during a period of time after the training is performed. This way the users will be able to practice on the system functionalities corresponding to their assigned role.

   The training courses will be comprised of training sessions on site plus high quality support documentation. Beside the copy of the presentations given, users will be given high quality written training manuals. All written training material will be in Norwegian.

   The working process for the development of the training documentation will be iterative with Indra working on the training documentation development and in paralell the Customer reviewing and providing comments and or approval of each piece of material.

   As a minimum, training material will consist of the following sections:

   - Introduction
   - System overview and process workflows
   - Roles and responsibilities
   - Tasks descriptions:
     - Role 1:
       - Objectives of your role
       - Process steps
     - ….
     - Role N
   - Security and contingency procedures
   - Key election milestones
   - Frequently Asked Questions (FAQ's)

In addition to the above, user administrators will be provided with detailed written instructions on system configuration and administration tasks.

Trainees will be given the help desk contact information for resolution of process/system queries and/or potential technical issues.

Training session will be customized per user profile:

PROFILE	TRAINING SESSIONS FOCCUSSED ON
Election administrators	• System configuration • Upload of basic election data and electoral register data • Managing roles and profiles • Technical support
Election staff - EMIS system	• EMIS central system functionality: electoral roll, nominations, generation of voter cards and ballot papers • Advance Poll procedures • Election day procedures
Supervisors of polling committee staff (Advance voting and Election day)	• Procedures at polling stations both advance and election day
Election staff - Advance voting	• Procedures for advance votes processing
Electoral Committee – E-voting	• The electoral committe procedures (election's key) • The e-voting process • The election count procedures including merging of p/e-voters • The reporting of results procedures

For the e-counting system:

ROLE	TRAINING SESSIONS FOCCUSSED ON
Election staff (Electoral authority and Supervisors)	• The e-counting system: General Description, functions and operations, users, user's interfaces • Procedures during the count • User interfaces: o Election System Manager o Count progress o Ballot boxes approval o Management of anomalies o Technical district polling stations o Generation and approval of election results • Transmission of results to SSB • Statistics and Motebok

Scanning of ballot papers supervisor	• Scanner queue management • Scanner operation
Verification of marks supervisor	• Verification of voter marks • Verification of other written marks • Verification of doubtful ballot papers

2. **System support**

Indra will provide system support via help desks as requested by the Customer. System support will be provided for the election staff managing the election administration system (EMIS), for the election staff managing the e-voting system and for the election staff managing the e-counting system.

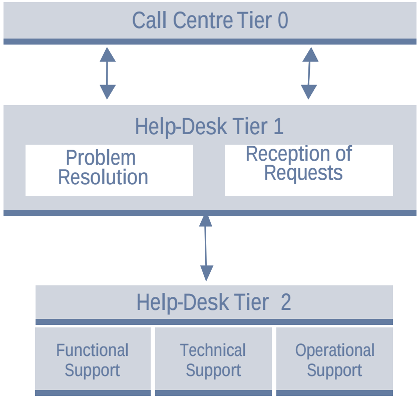
System support for e-voting system will be available from the 1st of July of 2011 until the polls are closed. This support will be available 24x7 during that period.

System support for election staff managing the election administration system (EMIS) will be available from the 1st of March of 2011 until the count is finished and results are approved by the Municipal and County Electoral Committees. Hours of operation will be 07:00 to 16:00.

System support for election staff managing the e-counting system will be available during one week which will cover the period of time from system installation to count finished and results are approved by the Municipal and County Electoral Committees. Hours of operation will be 08:00 to 16:00.

All help desks will implement a second tier support scheme:

- The Call Centre will route the user call to the appropriate supervisor.
- The Tier 1 support will be provided by the two supervisors who are specifically trained to resolve technical and operational issues that may arise
- A Second Tier support team, comprised by Indra technical staff involved in the system development and service design, will provide responses to any issues which cannot be resolved by the supervisors.



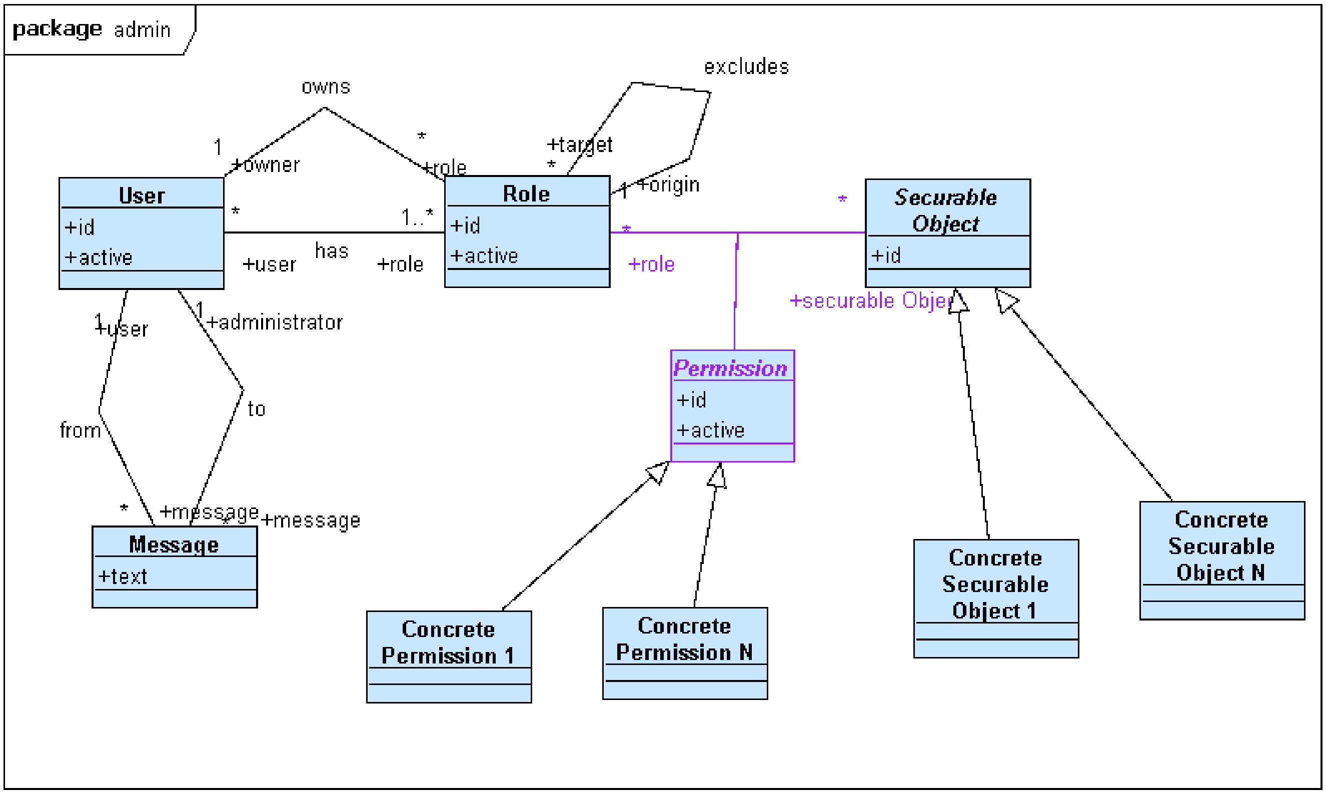
3. **Configuration of the central system**
   Indra will provide technical support during the configuration central system. Indra will have available a team of technicians to support the Customer technical team in the system configuration tasks. The technicians with be as follows: one architecture, one software applications and one communications specialists.

   These technicians will work hand in hand with the Customer’s team starting with the walk-through of the stages in the configuration process and continuing with the configuration itself.

3. Elaboration of Use Case Requirements

3.1. Use Case 0.1 Definition of Roles

Definition of roles is provided through the implementation of a web application accessible to administrators (central and local) and general users, based on the provision of a set of services for the manipulation of the following conceptual domain model (fragment) for access rights management:



Elaboration of Requirement F 0.1.1

We will fully comply with this requirement through the implementation of the **User** and **Message** Domain Objects and their associations, supported by the associated manipulation services. Details regarding the full set of properties to be included in those Domain Objects must be defined during analysis phase of the project. During the system definition, the Ministry together with Indra will have to define the possibility to create several authentication methods. In general terms, the system will allow the generation of new authentication methods, but only when they inherit from the ones already in existence and no changes in the software or the communications are needed.

Elaboration of Requirement F 0.1.2

We will fully comply with this requirement through the implementation of the association **has** that links User and Role Domain Objects, supported by the associated manipulation services.

Elaboration of Requirement F 0.1.3

We will fully comply with this requirement through the implementation of the **Role** Domain Object and their associations (**owns**) and association classes (**Permission**), supported by the associated manipulation services.

Details regarding the full set of properties to be included the Domain Objects and the particular Permissions (notice that the association class **Permission** is abstract) must be defined during analysis phase of the project. Implementation of the **excludes** association supports flagging of mutually exclusive Roles

Elaboration of Requirement F 0.1.4
We will fully comply with this requirement through the implementation of the associative class **Permission** that links Role and Securable Object Domain Objects, supported by the associated manipulation services. Details regarding the full set of properties to be included the Domain Objects and the particular Permissions (notice that the association class **Permission** is abstract) must be defined during analysis phase of the project.

Elaboration of Requirement F 0.1.5
We will fully comply with this requirement through the implementation of the associative class **Permission** that links Role and Securable Object Domain Objects, supported by the associated manipulation services. Details regarding the full set of properties to be included the Domain Objects and the particular Permissions (notice that the association class **Permission** is abstract) must be defined during analysis phase of the project.

Elaboration of Requirement F 0.1.6
We will fully comply with this requirement through the implementation of the association class **Permission**, supported by the associated manipulation services. Details regarding the full set of properties to be included in this Domain Object and the particular Permissions (notice that the association class **Permission** is abstract), including the design of a levelled hierarchy of permissions, must be defined during analysis phase of the project.

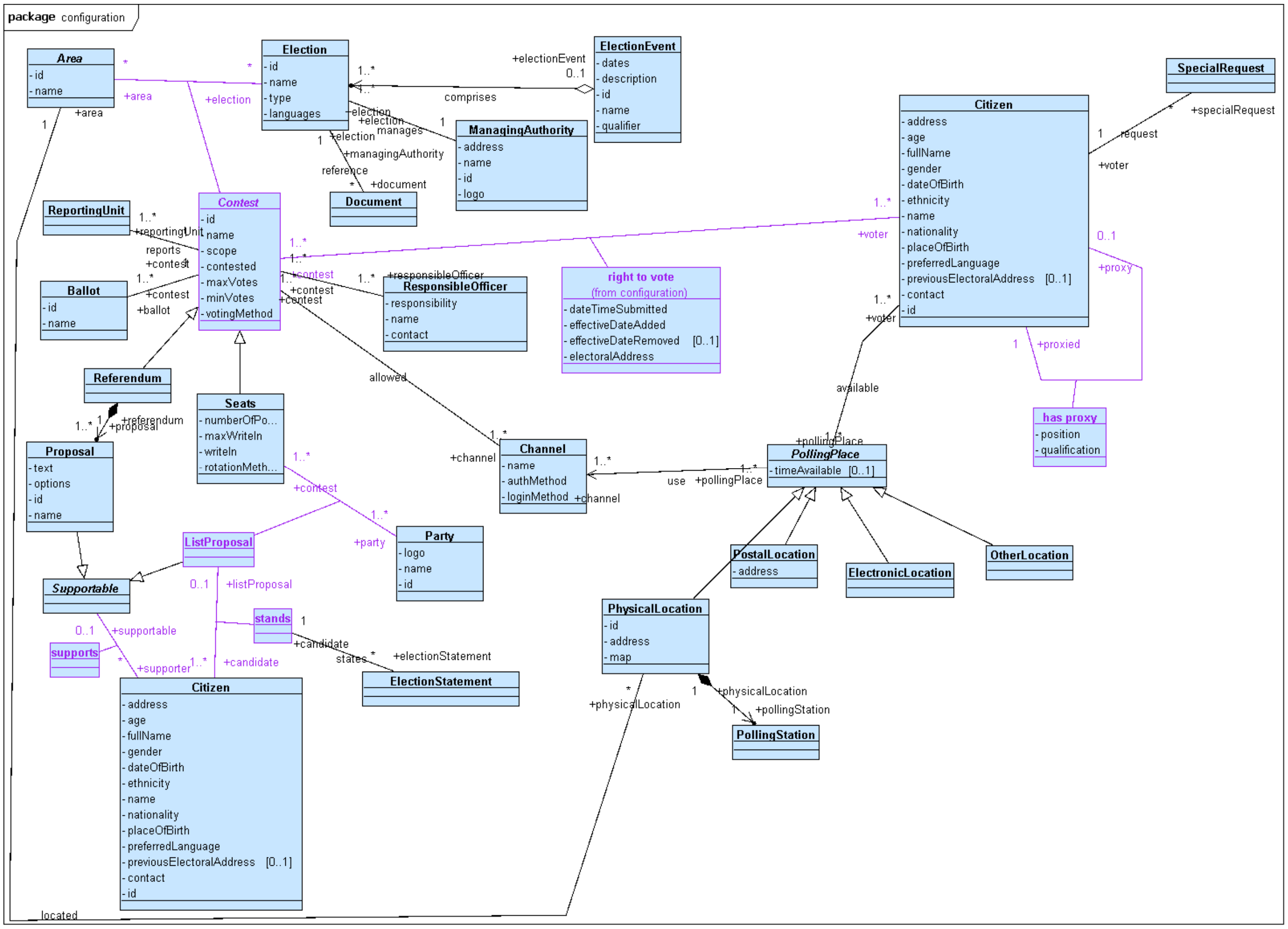
Elaboration of Requirement F 0.1.7
We will fully comply with this requirement through the implementation of the Class **Securable Object**, and its related associations, supported by the associated manipulation services. Details regarding the full set of properties to be included in this Domain Object and the particular Securable Objects (notice that the class **Securable Object** is abstract), including the design of an inheritance hierarchy of Securable Objects in terms of Permissions and Roles, must be defined during analysis phase of the project

Elaboration of Requirement F 0.1.8
We will fully comply with this requirement through the implementation of the finder services required

Elaboration of Requirement F 0.1.9
We will fully comply with this requirement through the implementation of the finder services required

3.2. Use Case 0.2 Configuration of the Election System

Configuration of the Election System is provided through the implementation of an application accessible to administrators (central and local), based on the provision of a set of services for the manipulation of the following conceptual domain model (fragment) for election configuration management:



Elaboration of Requirement F 0.2.1

We will fully comply with this requirement through the provision of a service for searching configurations in a repository in order to be modified and the provision of an application to easily specify the configuration of an Election Event.

Elaboration of Requirement F 0.2.2

We will fully comply with this requirement through the provision of a service for searching configurations in a repository in order to be modified and the provision of an application to easily specify the configuration of an Election Event.

Elaboration of Requirement F 0.2.3

We will fully comply with this requirement through the provision of a set of services to manipulate the Domain Objects shown in relation with Election System Configuration. During the analysis phase of the project, the Domain Objects' properties and associations must be refined and customized in order to conveniently support the different types of elections. Namely:

1. Support for EML in relation with the different type of elections (General Election, County Election, Municipality Election, Referendums, and Sami Election) will be provided transparently to the administrator (matching between Domain Objects –classes, associations, roles, …- and the elements of

EML). This support will be tailored to the specific needs of the KRD, and extension and/or modification of the initial support will be subject to the maintenance software agreements.

2. Configurable information (groups/parties, languages, error messages, etc.) should be designed during the design phase of the project in order to be reusable and easily manageable by the Administrators. An initial set of information will be provided, and extension and/or modification of this initial set will be subject to the maintenance software agreements.
3. This subject has its own section.
4. This subject has its own section.
5. Initial workflow specifications covering the processes regarding List Proposal presentation and approval and Electoral Roll management will be provided as part of the project. Administrators will be also provided with a configuration tool to perform parametrization of several subjects affecting this workflow specifications (for example, particular roles involved in approvals –steps in the workflow-, date ranges during which actions are allowed/disallowed, …) These workflows will be specified initially following the EML specifications/recommendations in relation with the processing of the messages involving these business processes. Substantial modification of these initially provided workflow specifications will be subject to the software maintenance agreements.
6. Initial implementation of the Sainte-Laguë modified method will be provided as part of the project. The design of the system will allow modification through configuration the particular rule to use in any election. Additional rules (D'Hont, …) can also be implemented as part of the software maintenance agreements.
7. Initial templates for the generation of ballot papers, polling cards, polling books, etc. will be provided as part of the project. These templates will include the layout of the document and the set of processing instructions (data base connections, queries, etc.) and parameters needed to render an instance of it. Additional templates can also be implemented as part of the software maintenance agreements.
8. Web-based interfaces will be provided between the Electoral System and several external systems. These web-services specifications will be based in the WSDL standard for the operational definition and EML will be the preferred standard in order to specify the message data structure to exchange in every operation and to identify the operations to include in the interfaces; provision of a client and/or server as part of the Electoral System will be defined as part of the design in phase of the project. The external systems involved in this project are:
    - SKD (provider of the base for the Electoral Roll); a web-service based server(s) will be provided by the Electoral System for the reception of the registered voters;
    - SSB; a web-service based client will be provided by the Electoral System in order to provide the information ready to be published (concerning preliminary and final results of the electoral process);
    - Brønnøysund Register Centre; a client web-service will be provided as part of the Electoral System in order to access the functionality provided by this register in relation to the parties.

    Additional support for external systems (support for new external systems and/or improved support for the already considered external systems) can be included subject to the software maintenance agreements.
9. All the configuration information will be subject to naming conventions and file structure locations that allow packaging and exporting and importing into the configuration tool.

<u>Elaboration of Requirement F 0.2.4</u>
We will fully comply with this requirement. Validation functionality will be provided by the system in order to check a given configuration by an administrator. The configuration tool will provide browsing facilities to inspect and modify a given configuration bundle.

Elaboration of Requirement F 0.2.6

We will fully comply with this requirement. During the design phase the particular mechanism of approval for a given configuration file will we decided. A workflow of approval is the most obvious option. The approval should be included in the configuration bundle in order to prevent execution of non-approved configuration bundles by the Electoral System (for example, using some kind of electronic signature available only to authorized personnel).

Elaboration of Requirement F 0.2.8

We will fully comply with this requirement. This date limit can be enforced as part of the workflow definition to be implemented by the Electoral System; after that date limit, modification of a given configuration bundle won't be allowed.

Elaboration of Requirement F 0.2.9

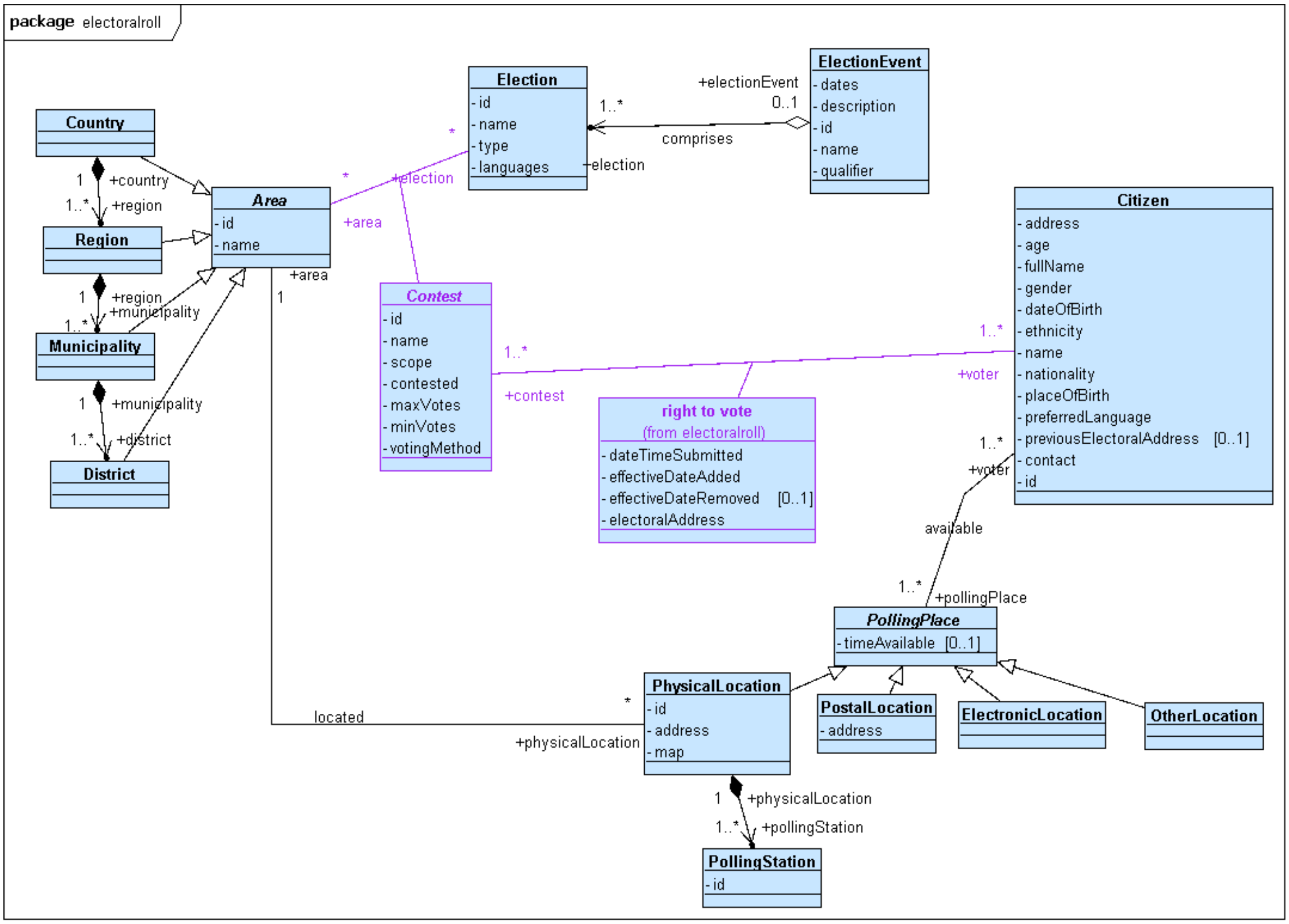
We will fully comply with this requirement. The configuration application will allow opening an already existing configuration bundle in order to modify its contents and save the modified configuration as a different one. After this operation, the recently saved configuration bundle won't be validated nor approved, even if the original one was validated and/or approved.

3.3. Use Case 0.3 Electoral Roll

Electoral Roll functionality is provided through:

- implementation of an automatic mechanism (based on FTP, s/FTP, web-service, … to be defined with the Ministry and SKD) for receiving and processing the SKD information used as a base for the Electoral Roll, and
- implementation of a web application accessible to all the voters in order to check the correctness of the information included in the Electoral Roll and functionality to request corrections.

Both software elements are based on the provision of a set of services for the manipulation of the following conceptual domain model (fragment) for the Electoral Roll:



Elaboration of Requirement F 0.3.2

We will fully comply with this requirement through the implementation of the mechanism to receive the SKD information and the implementation of services to manipulate the Domain Objects (mainly but not limited to) **Election**, **Citizen**, **right to vote**, **Contest**, **PollingPlace** (and particular subclasses) and **Area** (and particular subclasses) and related associations. Details regarding the full set of properties to be included in every Domain Object, must be defined during analysis phase of the project. The particular mechanism to be used to receive the information from SKD (FTP, s/FTP, web-service, etc.) must be defined during the design phase of the project, in cooperation with SKD and Ministry.

Elaboration of Requirement F 0.3.4

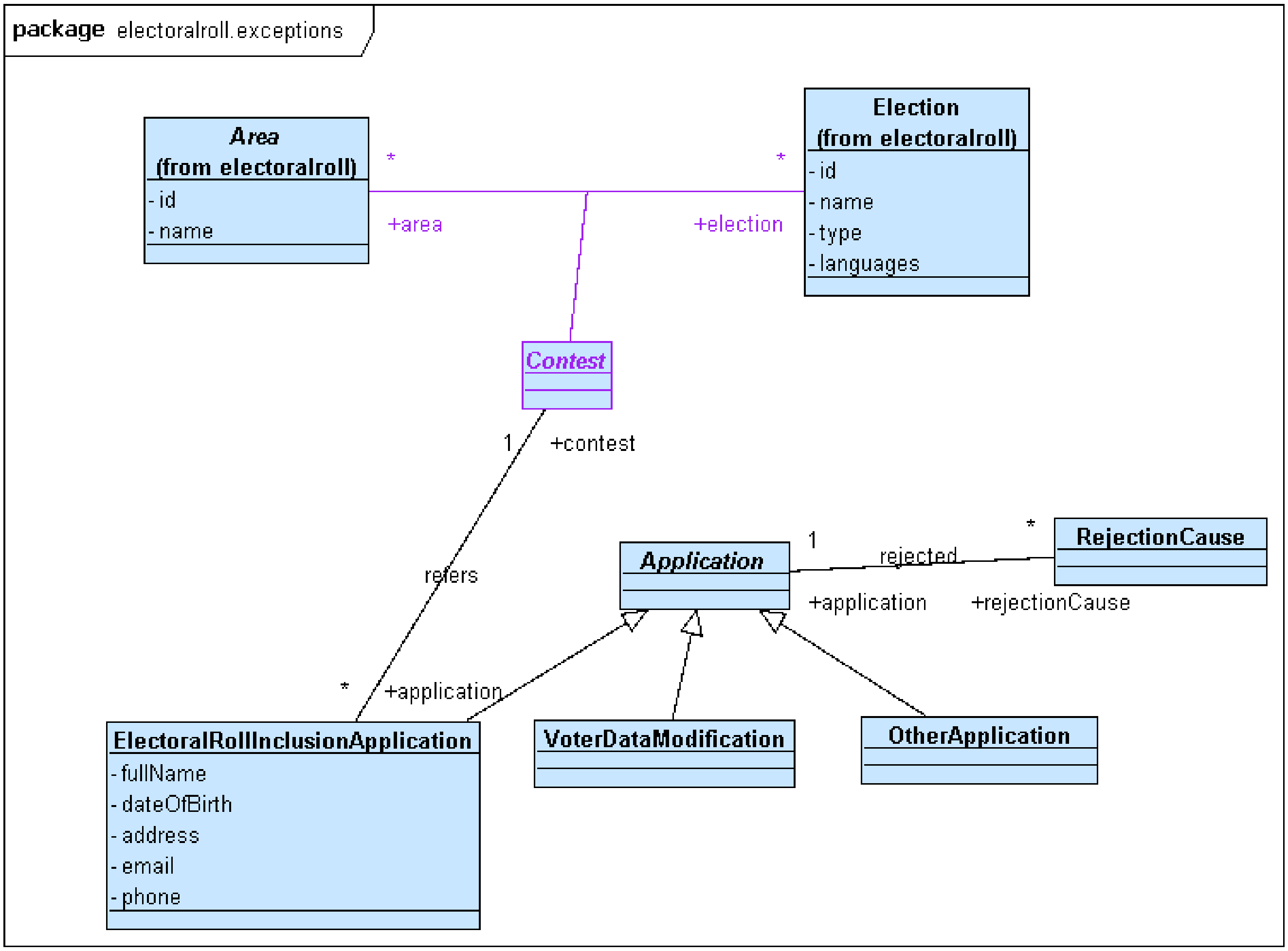
We will fully comply with this requirement. In order to be able to optimally process the transactions provided by the SKD, a time-stamp is assumed to be provided as part of each transaction. This time-stamp allows the system to discard those transactions already processed or older than the last transaction processed in the last batch.

Elaboration of Requirement P 0.3.1

We will fully comply with this requirement. In order to be able to comply with the required processing time / throughput, a time-stamp is assumed to be provided as part of each transaction provided by the SKD. This timestamp will allow the system to discard those transactions already processed and/or older than the most recently processed transaction in the last batch (incremental updates).

3.4. Use Case 0.4 Exception Process for Electoral Roll

Exception Processing for the Electoral Roll is provided through the implementation of a web application accessible to general public and allowed users, based on the provision of a set of services for the manipulation of the following conceptual domain model (fragment) for applying for modifications of the Electoral Roll and their approval/rejection by the Electoral Committee:



Elaboration of Requirement F 0.4.1

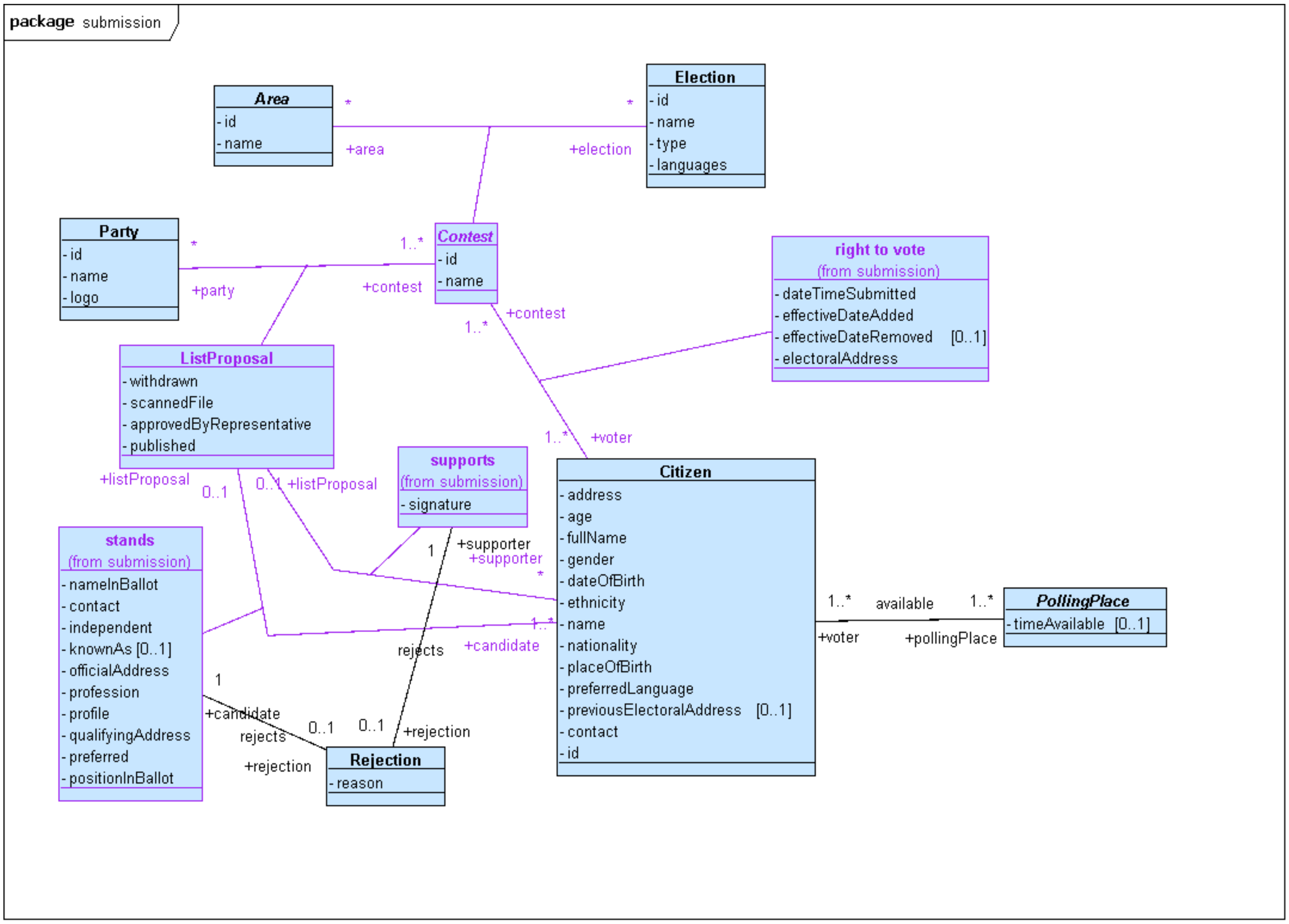
We will fully comply with this requirement through the implementation of the **Application** Domain Objects and their associations, supported by the associated manipulation services. Details regarding the full set of properties to be included in those Domain Objects and the set of particular subclasses (Application is an abstract class) must be defined during analysis phase of the project. **ElectoralRollInclusionApplication** is included as an example of such a particular subclass; in that Domain Object, elements (properties **email** and **phone**) are included to allow the communication with the applicant citizen.

Elaboration of Requirement F 0.4.4
We will fully comply with this requirement through the implementation of the **RejectionCause** Domain Objects and their associations, supported by the associated manipulation services. Details regarding the full set of properties to be included in those Domain Objects and the set of particular subclasses (Application is an abstract class) must be defined during analysis phase of the project. **ElectoralRollInclusionApplication** is included as an example of such a particular subclass; in that Domain Object, elements (properties **email** and **phone**) are included to allow the communication with the applicant citizen. The mechanism provided by the system in order to notify the Electoral Committee of a pending application will be defined during the design phase of the project (it can be, for instance, an email including a link to the web page showing the application)

Elaboration of Requirement F 0.4.5
We will fully comply with this requirement through the implementation of a web application restricted to the Electoral Committee in order to manipulate the Electoral Roll, in order to reflect the modifications corresponding to approved applications (voters inclusions, modifications, etc.).

3.5. Use Case 1.1 Submission of List Proposals

Submission of List Proposals functionality is provided through the implementation of a web application accessible to (allowed) Parties/Groups and Electoral Committees, based on the provision of a set of services for the manipulation of the following conceptual domain model (fragment) for presenting the List Proposals and their approval/rejection:



<u>Elaboration of Requirement F 1.1.1</u>

We will fully comply with this requirement through the implementation of the **Election**, **Contest**, **Party**, **ListProposal** and **stands** Domain Objects and their associations, supported by the associated manipulation services. Details regarding the full/final set of properties to be included in those Domain Objects must be defined during analysis phase of the project.

<u>Elaboration of Requirement F 1.1.2</u>

We will fully comply with this requirement through the implementation of file uploading facility in the web application for the candidates lists. The particular file formats supported will be specified during the design phase project. Additionally, support for handwritten character recognition will be provided in order to process, as automatically as possible, the lists of supporters after being scanned. Functionality to include manually supporters and or candidates, will also be provided (and only allowed to the Electoral Committee, in the former case). The particular mechanism(s) to notify the Electoral Committee of a list proposal withdrawal (e-mail, etc.) will be defined during the design phase of the project.

<u>Elaboration of Requirement F 1.1.4</u>

We will fully comply with this requirement through the implementation of the **stands** and **supports** Domain Objects and their associations, supported by the associated manipulation and checking services. Details regarding the full set of properties to be included in those Domain Objects must be defined during analysis phase of the project.

Elaboration of Requirement F 1.1.5
We will fully comply with this requirement through the implementation of the **stands** and **supports** Domain Objects and their associations, supported by the associated manipulation and checking services. Details regarding the full set of properties to be included in those Domain Objects must be defined during analysis phase of the project.

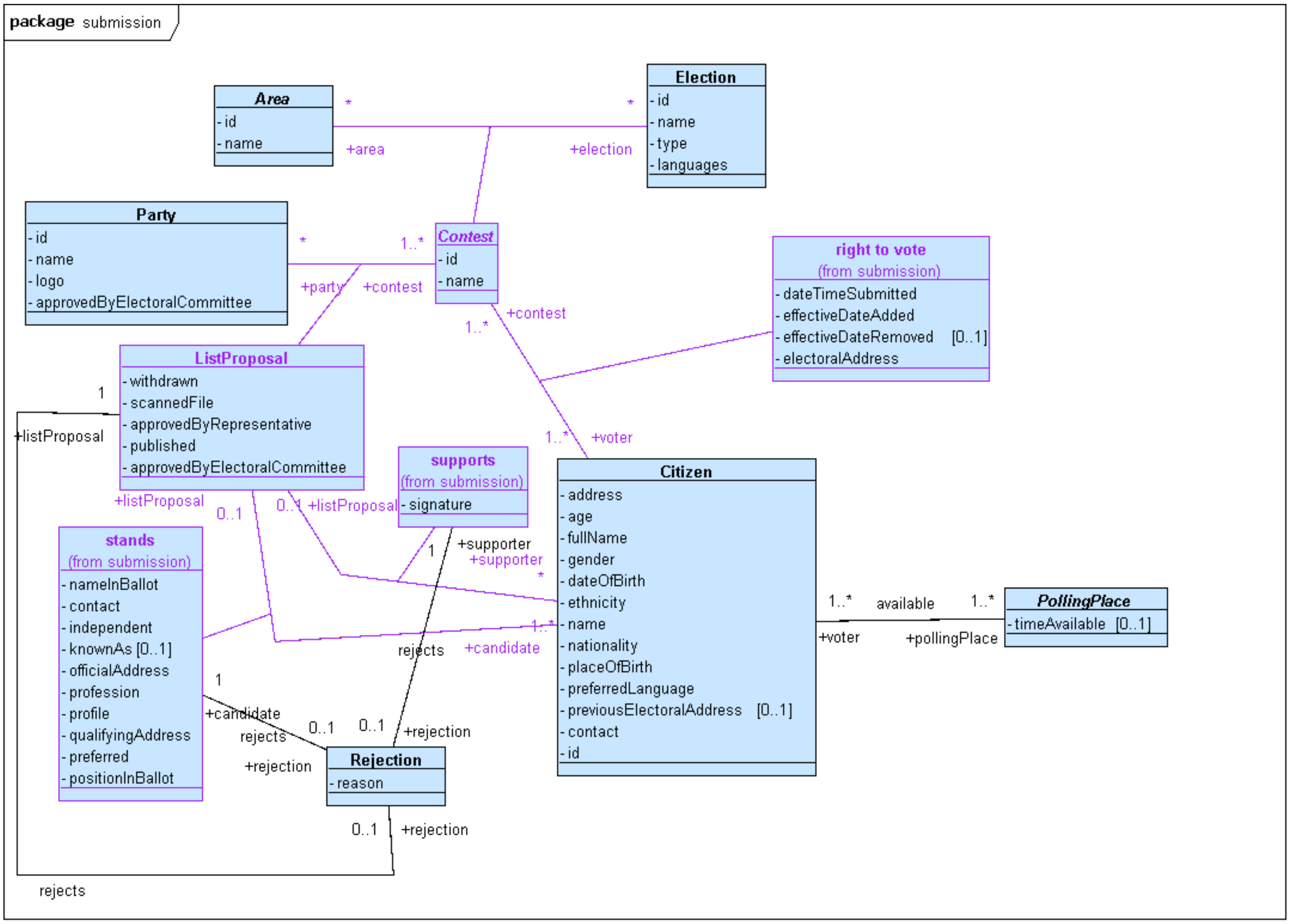
Elaboration of Requirement F 1.1.7
We will fully comply with this requirement through the implementation of **ListProposal** Domain Object and their properties (**approvedByRepresentative**), supported by the associated manipulation and checking services. Details regarding the full set of properties to be included in those Domain Objects must be defined during analysis phase of the project.

Elaboration of Requirement F 1.1.10
We will fully comply with this requirement through the implementation of **ListProposal** Domain Object and their properties (**published**), supported by the associated manipulation and checking services. Details regarding the full set of properties to be included in those Domain Objects must be defined during analysis phase of the project.

Elaboration of Requirement P 1.1.1
We will fully comply with this requirement.

3.6. Use Case 1.2 Processing List Proposals

Processing of List Proposals functionality is provided through the implementation of a web application accessible to (allowed) Parties/Groups and Electoral Committees, based on the provision of a set of services for the manipulation of the following conceptual domain model (fragment) for processing the List Proposals:



Elaboration of Requirement F 1.2.1

We will fully comply with this requirement, with the provision of a search service in the web application to display the List Proposals. Details regarding the full set of properties to be included in those Domain Objects must be defined during analysis phase of the project

Elaboration of Requirement F 1.2.2

We will fully comply with this requirement, with the provision of manipulation services through a web application to associate **Rejection**s to **ListProposal**s and mark **ListProposal**s as approved by Electoral Committee.

Elaboration of Requirement F 1.2.5

We will fully comply with this requirement, with the provision of a query service through a web application available to the general public, in order to display the information relating to approved **Parti**es and published **ListProposal**s, and a pairing service available to the Electoral Committee to mark as approved the **Parti**es and **ListProposals**

Elaboration of Requirement F 1.2.6

We will fully comply with this requirement through the implementation of **ListProposal, stands** and **supports** Domain Objects, supported by the associated manipulation and checking services. Details regarding the full set of properties to be included in those Domain Objects must be defined during analysis phase of the project, including properties and associations aimed at supporting versioning of those Domain Objects, if needed.

3.7. Use Case 2.1 E-voting

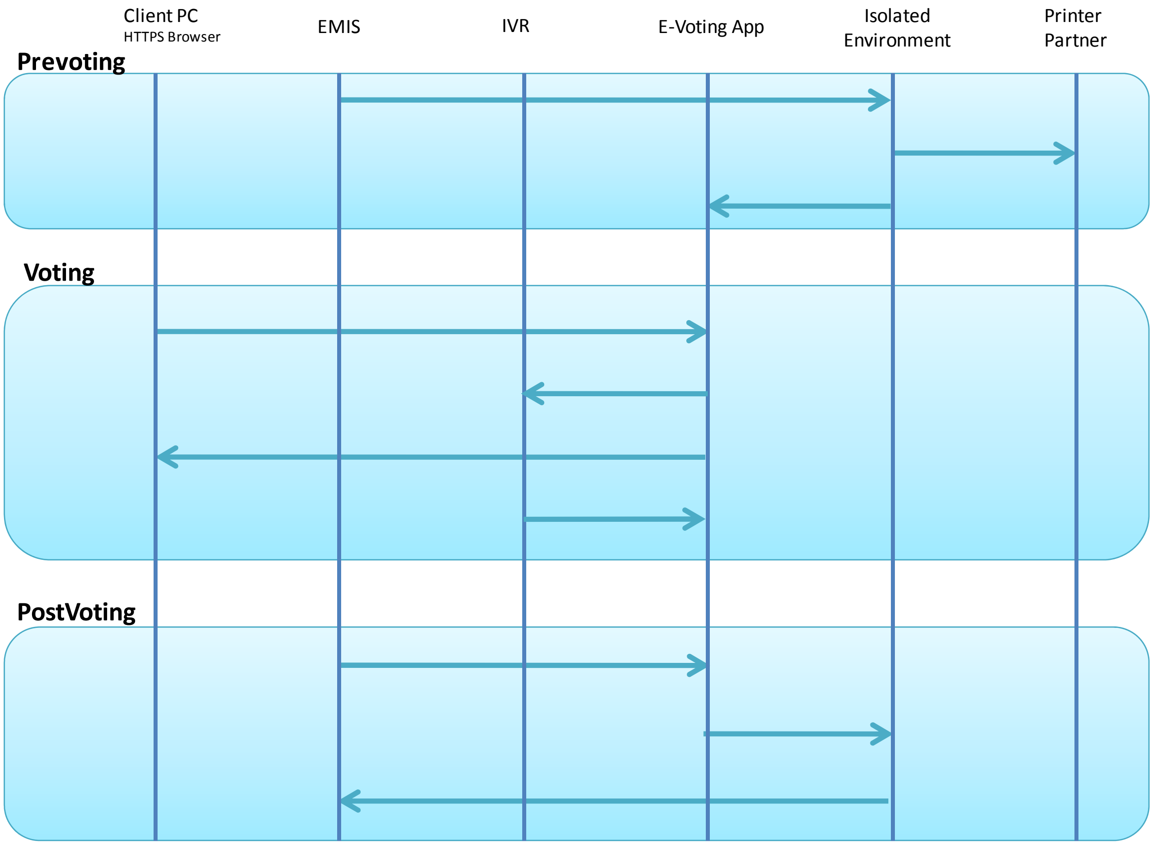
As we explained in our introduction, this use case is the one that have changed more during all the offering process with you, and the one with the biggest changes for this last version. It's part of our effort explain the Ministry team that our solution is not yet a closed solution, so the proposal we are submitting in this document is open in his functionality and processes to be checked and improved (if needed) during the initial phases of our project.

We have worked in this release in a way to allow the system confirm the voter that it have received the correct vote intent without actually showing him/her the content of that vote. After searching for more complex ways to do that we realized that we needed something simple and easy to understand by voter, so we have finished really near where we started… in the "encrypted ballot"!

We kept most of our web interface similar to the one used in the previous prototype with small improvements to try to address some of the comments we received from you. But he made big structural changes to the security solution, which affected mainly to the last page in the website environment and of course to all the processes in the servers.

Let's review the processes with the systems involved in each one. As we are presenting two different prototypes with two different processes we will do this task twice to highlight the differences. Please note that this is a high level review of the process with no a system definition objective but a first process review.

Prototype with IVR



Prevoting

1) The EMIS will send to the Isolated environment the voters involved in the election with all the data needed to print the voter cards. Also includes the public key of every printer in the system, the evoting server public key and the key pair for the isolated environment (different from the election key pair and stored in a security token).

a. The Isolated environment will create the random party codes for the system and the random party codes for each voter.
b. The process will print the voter cards including for each voter his/her voterId and the randomly generated party codes for that voter.
c. The isolated environment creates a list of hashes per voter using as origin information the voterId, the party code for the system and the party code for the voter.
d. The isolated environment creates the election key pair, splits the private key in pieces and stores them in tokens protected by passwords.

2) The isolated environment stores in a removable data storage device the information to be sent to the printer partner (all the voter cards to be printed in that printer). This information will be encrypted with the public key for that printer and signed by the isolated environment with its private key (using the security token created in 1).

3) The isolated environment stores in a removable data storage device the information to be sent to the evoting app. This information includes the list of hashes per voter and the election public key. All the information is encrypted with the evoting server public key and signed by the isolated environment with its private key (using the security token created in 1).

Voting

1) The voter sends a vote through https to the evoting server (previously the voter has been identified through his/her digital Id and usgin SAML against the identification server. Also the evoting server asked the EMIS about the elections active for that voter and have presented to the voter the voting interface, including the random list of parties and the possibility to make a personal vote). Sending the vote includes the voter identification, the party selected, the personal preferences if any and the final party code inserted in the final screen.

a. The evoting server creates a hash from the voterId (received from the EMIS using the voter digital identification), the system party code selected by the voter in the web, and the voter's party code inserted in the las page. The system compares that hash against the list of hashes for that voter, knowing if the party selected in the web site and the final voter's party code inserted matched correctly. If not, fires a message to the voter (with a maximum of 3 retries)
b. The server encrypts all the information using the election public key and stores it in the database, related to the voter and with information regarding the vote channel (that is controlled or uncontrolled environment).

2) The e-counting server send to the IVR server the voterID and the voter's party code, so it can be accessed by the IVR when the voter makes his/her call.

3) The e-counting server tells the voter in the website that the vote has been correctly registered and that he/she needs to call the IVR to confirm the vote content using the voterID (the IVR will read to the voter the party code in his/her voter card)

4) The IVR server will notify the evoting server when the voter has confirmed the vote through the phone. Also it will notify to the evoting the vote cancellation if it happens.

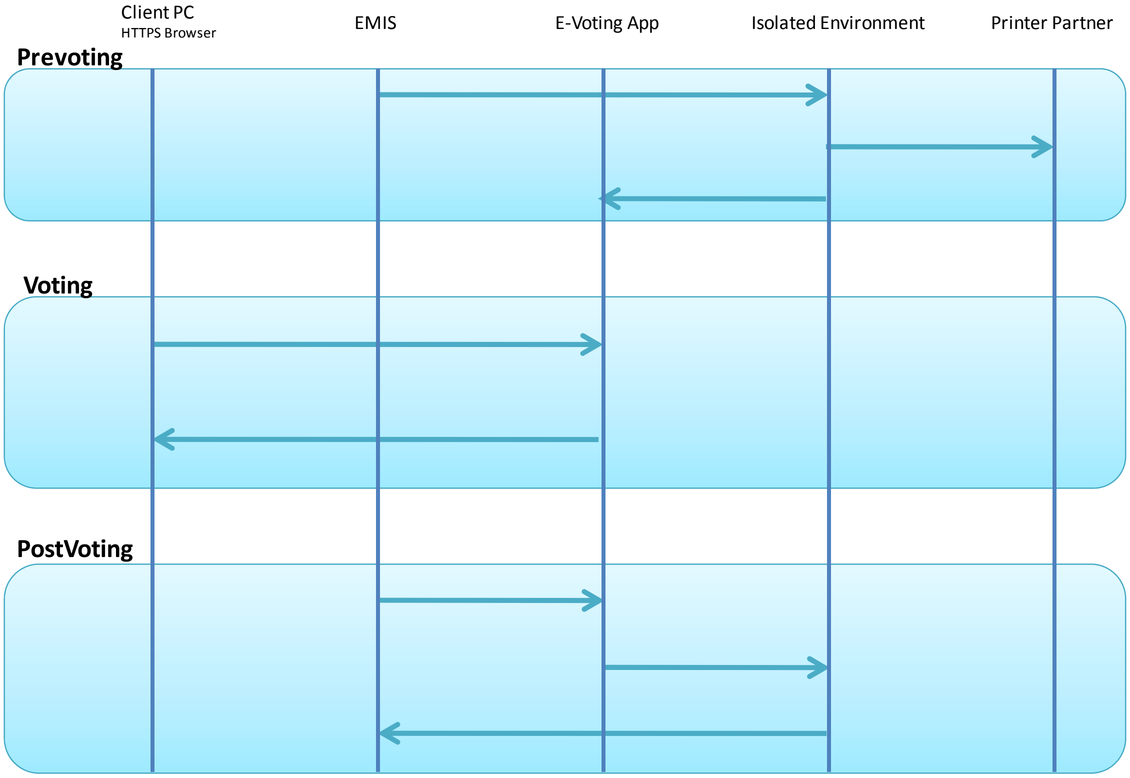
PostVoting

1) After the EMIS has received all the p-voters lists from the EPBs in the polling stations, it wil send them to the e-voting server.

2) The evoting server applies all the rules to the votes stored, removing those voters in the p-vote voters list, but also keeping the last e-vote from a controlled environment for each voter if exists, or the last e-vote from uncontrolled environment if no controlled environment vote exists. The system will store the selected votes (those to be counted) in a removable data storage device (signed with his private key and encrypted with the isolated environment public key)

3) The isolated environment checks the signature in the received information, decrypts it and starts the process to rejoin the election private key by asking for tokens to the election officers. When the election private key is available, the system start to decrypt the votes, check the voters party codes and keep every valid vote in a final results list (created in a random order and mixed again after the process has been finished). The final list of votes (just party votes and preferences, with no link to any voter) will be then encrypted with the public key of the EMIS, signed with the isolated environment private key and store in a removable data storage device, to be sent to the EMIS, where the calculation of results activities will continue.

Prototype without IVR



Prevoting

1) The EMIS will send to the Isolated environment the voters involved in the election with all the data needed to print the voter cards. Also includes the public key of every printer in the system, the evoting server public key and the key pair for the isolated environment (different from the election key pair and stored in a security token).

a. The Isolated environment will create the random party codes for the system and the random party codes for each voter.
b. The process will print the voter cards including for each voter his/her voterId and the randomly generated party codes for that voter.
c. The isolated environment creates the election key pair, splits the private key in pieces and stores them in tokens protected by passwords.

2) The isolated environment stores in a removable data storage device the information to be sent to the printer partner (all the voter cards to be printed in that printer). This information will be encrypted with the public key for that printer and signed by the isolated environment with its private key (using the security token created in 1).

3) The isolated environment stores in a removable data storage device the information to be sent to the evoting app. This information includes the list valid party codes per voter (not related to any party) and the election public key. All the information is encrypted with the evoting server public key and signed by the isolated environment with its private key (using the security token created in 1).

Voting

1) The voter sends a vote through https to the evoting server (previously the voter has been identified through his/her digital Id and usgin SAML against the identification server. Also the evoting server asked the EMIS about the elections active for that voter and have presented to the voter the voting interface, including the random list of parties and the possibility to make a personal vote). Sending the vote includes the voter identification, the party selected, the personal preferences if any and the voterID inserted in the final screen (from the voters card).

a. The evoting server compares the voterId inserted in the website with the voterId (received from the EMIS using the voter digital identification) knowing if the voter identified thorugh the eID has his/her voter card and has inserted the voterId correctly. If not, fires a message to the voter (with a maximum of 3 retries)
b. The server encrypts all the information using the election public key and stores it in the database, related to the voter and with information regarding the vote channel (that is controlled or uncontrolled environment).

2) The e-counting server tells the voter in the website that the vote has been correctly registered and that the vote he/she has selected correspond with the code XXXX in his/her voter card. By using this final step the voter will know that he/she has sent the vote to the real election server and also that it has received his/her real intention to vote.

Post Voting

1) After the EMIS has received all the p-voters lists from the EPBs in the polling stations, it wil send them to the e-voting server.

2) The evoting server applies all the rules to the votes stored, removing those voters in the p-vote voters list, but also keeping the last e-vote from a controlled environment for each voter if exists, or the last e-vote from uncontrolled environment if no controlled environment vote exists. The system will store the selected votes (those to be counted) in a removable data storage device (signed with his private key and encrypted with the isolated environment public key)

3) The isolated environment checks the signature in the received information, decrypts it and starts the process to rejoin the election private key by asking for tokens to the election officers. When the election private key is available, the system start to decrypt the votes, check the voters party codes and keep every valid vote in a final results list (created in a random order and mixed again after the process has been finished). The final list of votes (just party votes and preferences, with no link to any voter) will be then encrypted with the public key of the EMIS, signed with the isolated environment private key and store in a removable data storage device, to be sent to the EMIS, where the calculation of results activities will continue.

Elaboration of Requirement F 2.1.1

The evoting system will be real time connected to the EMIS ensuring a perfect matching in the electoral roll for each election and the users authentication and access to the them.

This real time connection will allow the system even allow the voters to check which elections can they actually vote, including last minute changes in the electoral roll.

Elaboration of Requirement F 2.1.2

First screen after a valid authentication will show the elections active to one voter. Therefore, the voter can select the one he/she wants to vote.

Once the vote has been sent and the final confirmation screen shown, the voter will have a button to go that active elections screen to proceed with other elections votes.

Elaboration of Requirement F 2.1.3

The voter user interface, developed completely in html, will cover this requierement. Please check also our prototype.

Elaboration of Requirement F 2.1.4

The voter user interface, developed completely in html, will cover this requierement. Please check also our prototype.

Elaboration of Requirement F 2.1.5

Please refer to the introduction in this use case.

Elaboration of Requirement F 2.1.6

Please refer to the introduction in this use case.

Elaboration of Requirement F 2.1.7

The evoting system will be real time connected to the EMIS ensuring a perfect matching in the electoral roll for each election and the users authentication and access to the them.

We fully cover this requirement.

Elaboration of Requirement F 2.1.8

The evoting system will be real time connected to the EMIS ensuring a perfect matching in the electoral roll for each election and the users authentication and access to the them.

We fully cover this requirement.

Elaboration of Requirement P 2.1.1

Our solution architecture is built over standard J2EE infrastructure and scales horizontally in the application server tier.

Also, for the national deployment we recommend a virtual data center infrastructure, so new servers can be deployed in a peak situation.

Please refer to our architecture and infrastructure proposal for further details.

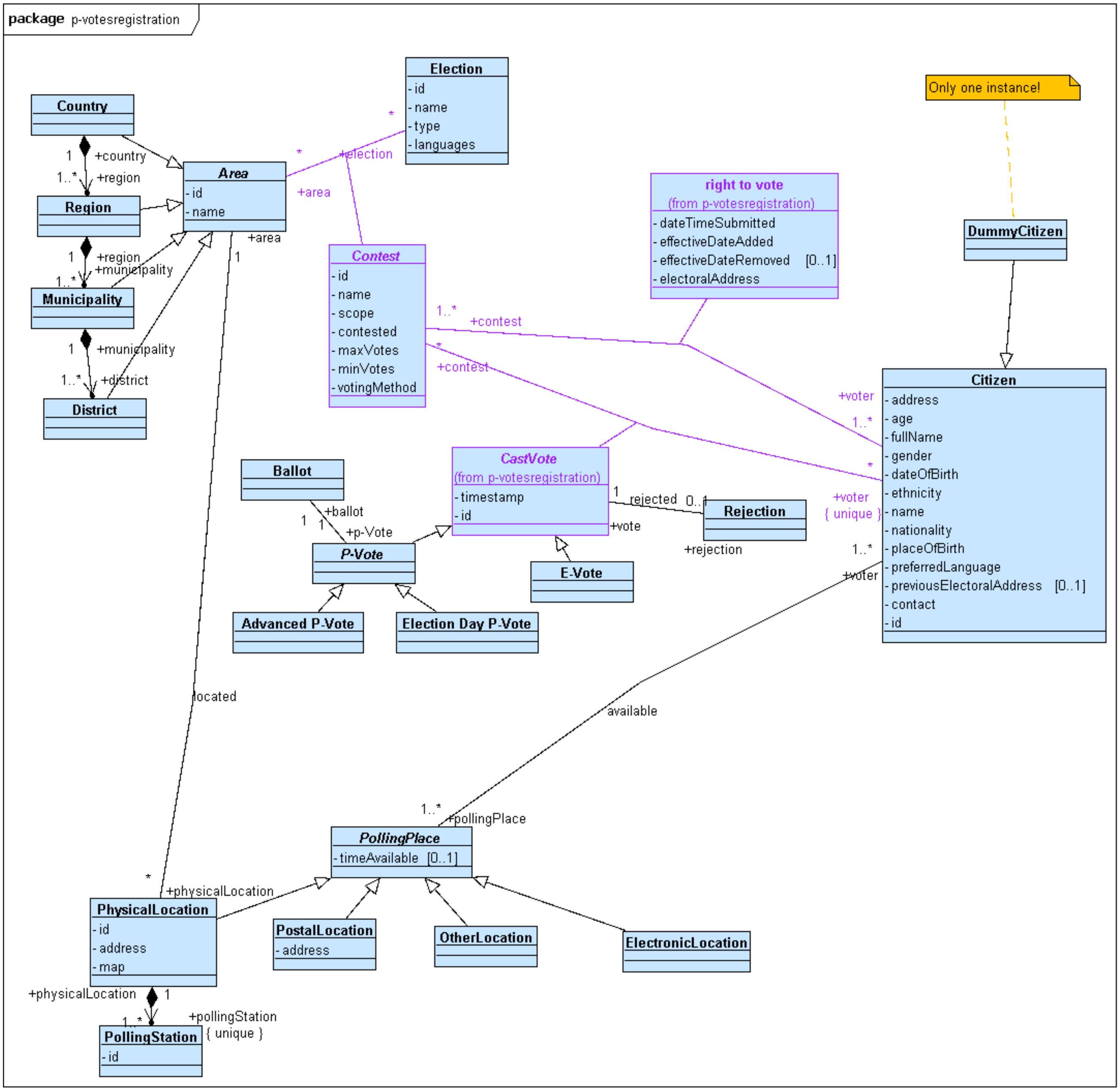
Elaboration of Requirement NF 2.1.1

We fully cover this requirement.

The active elections screen will show always all the active elections, not showing any data about previous votes in any of them.

3.8. Use Case 3.1 Registration of p-votes in Electoral Roll

Registration of p-votes in the Electoral Roll functionality is provided through the implementation of a web application accessible to (allowed) Polling and Electoral Committees, based on the provision of a set of services for the manipulation of the following conceptual domain model (fragment) for p-votes registration:



Elaboration of Requirement F 3.1.1

We will fully comply with this requirement. Namely, the web application will provide services to check:

- the voter (Citizen) can be found in the system (that is, is included in the Electoral Roll) and is not an instance of DummyCitizen;
- the voter (Citizen) has the right to vote in a contest (right to vote association class end) in a given municipality;
- the voter (Citizen) has cast an advance paper vote (Advanced P-Vote) and the contest end of both association classes (right to vote and CastVote) are attached to the same Contest instance.

All those checks can be supported with the Domain Objects **Citizen**, **Contest**, **right to vote** and **CastVote** (and subclasses) and their associations. Details regarding the full set of properties to be included in those Domain Objects must be defined during analysis phase of the project.

<u>Elaboration of Requirement F 3.1.2</u>
We will fully comply with this requirement. Namely, the web application will provide a service to create **Advanced P-Vote** instances, for a given **Citizen** in a given **Contest** (please note that **CastVote** is an abstract class). The web application will also provide a service to check if an **Advanced P-Vote** has already casted by that voter (**Citizen**) in a given **Contest**.

<u>Elaboration of Requirement F 3.1.4</u>
We will fully comply with this requirement. Namely, the web application will provide services to register:

- votes from a voter (**Citizen**) that can’t be found in the system (that is, is not included in the Electoral Roll); the casted vote (subclass of **CastVote**) will be associated then to the **DummyCitizen**; this **DummyCitizen** can be implemented as a **Citizen** with an special id.
- Votes from avoter (**Citizen**) that hasn’t the right to vote in a **contest** end) in a given municipality: that is, the **contest** end of the **right to vote** association class and the **contest** end of the CastVote association class for a given **Citizen,** are attached to different **Contest** instances;
- As many votes (**CastVote** subclasses) as wanted, for a given **Contest** and a given **Citizen** (but they can be **rejected** associating a **Rejection** to the **CastVote**, if deemed not valid by the Polling/Electoral Committee)

All those registration services can be supported with the Domain Objects **Citizen**, **Contest**, **right to vote** and **CastVote** (and subclasses) and their associations. Details regarding the full set of properties to be included in those Domain Objects must be defined during analysis phase of the project.

<u>Elaboration of Requirement F 3.1.7</u>
We will fully comply with this requirement

3.9. Use Case 3.2 Manual registration of p-vote results

<u>Elaboration of Requirement F 3.2.1</u>
The system will allow the upload of election results in various formats (csv, xls, etc.). These formats will be defined during the first stage of the project.

The data to be upload will be the preliminary results (only votes for party lists) and the final results (votes for party lists and votes for individual candidates).

Since the system implements a role-base authentication scheme, each electoral authority (whether Municipal or County) will have access to this functionality for the election type they have authority over.

<u>Elaboration of Requirement F 3.2.2</u>
The system allows for the manual input of relevant information as recorded in each Motebok stemmestyre. Examples of this information is as follows:

- Total number of voters
- Number of ballot papers in the ballot box
- Number of valid votes

- Number of ballot papers in special envelopes(fremmede)
- Blank ballot papers
- Doubtful papers (tvilsomme)

Elaboration of Requirement F 3.2.4
Once all results are input, both preliminary and final for advance and election day, the system calculates the p-votes election results. The user can visualize the results per kret and aggregated per electoral district/municipality/county (depending on the election type). Results are calculated based on votes to party lists and also on personal votes to specific candidates. Differences between preliminary and final results are also presented to the user.

Elaboration of Requirement F 3.2.5
Access to information on election results and approval of election results is only be granted to members of the Electoral Committee. This will be implemented through the role-based authentication scheme.

Elaboration of Requirement F 3.2.8
The system presents the user with the preliminary and the final results per kret. The user of the Electoral Committee has to confirm the final results based on the comparison with the preliminary ones. If they do not balance, the confirmation of the kret results are put on hold until those votes are counted again (either manually or with the e-counting system) and a second set of results is input into the system. The user is then able to compare the re-count results with the preliminary ones and take a decision base on that comparison.

3.10. Use Case 3.3 Electronic counting of p-votes

Elaboration of Requirement F 3.3.1
File metadata will be pre-loaded into the e-counting system before the counting of votes starts. For each contest the metadata includes: election districts, krets within each district and corresponding identification information, number of registered voters per kret, parties/lists contesting the election, nominated candidates in each party/list.

The system allows for uploading of pre-election data by importing any standard file (xls, csv …).

Elaboration of Requirement F 3.3.2
The ballot papers are scanned using commercial scanners that generate the corresponding image file in standard .tiff format.

Alternative standard formats such as .jpeg can also be generated subject to software maintenance agreements.

Elaboration of Requirement F 3.3.3
Ballot paper recognition is performed as the ballot paper is being scanned. The system compares its format to the list of official ballot papers “templates”. If this comparison fails, the system flags the image of that particular ballot paper to a verification operator who will indicate on the screen whether it is an official ballot paper and the party/list it belongs to. The system recognizes over 99% of the ballot papers.

After the image is captured, the system utilizes a recognition engine to read the relevant information on the ballot paper scanned image.

If the system is not capable of interpreting the information read on the image with a pre-determined level of certainty, the ballot paper image is flagged to a verification operation together with what the system is interpreting. The operator can confirm or change that interpretation.

Ballot papers include a serial number which guarantees that each ballot paper is uniquely identified. Duplicated ballot papers are flagged to the electoral committee for adjudication.

The system stores both the digital image of the ballot paper together with the information read/verified/adjudicated on it.

Any ballot paper with Braille characters can be scanned by the scanners.

Elaboration of Requirement F 3.3.5
The e-counting system stores all count data plus ballot paper images in the e-counting servers.

Once all ballot papers are counted and dealt with, the Electoral Committee run the calculation of election results and the gereneration of the Motebok (Election Protocol).

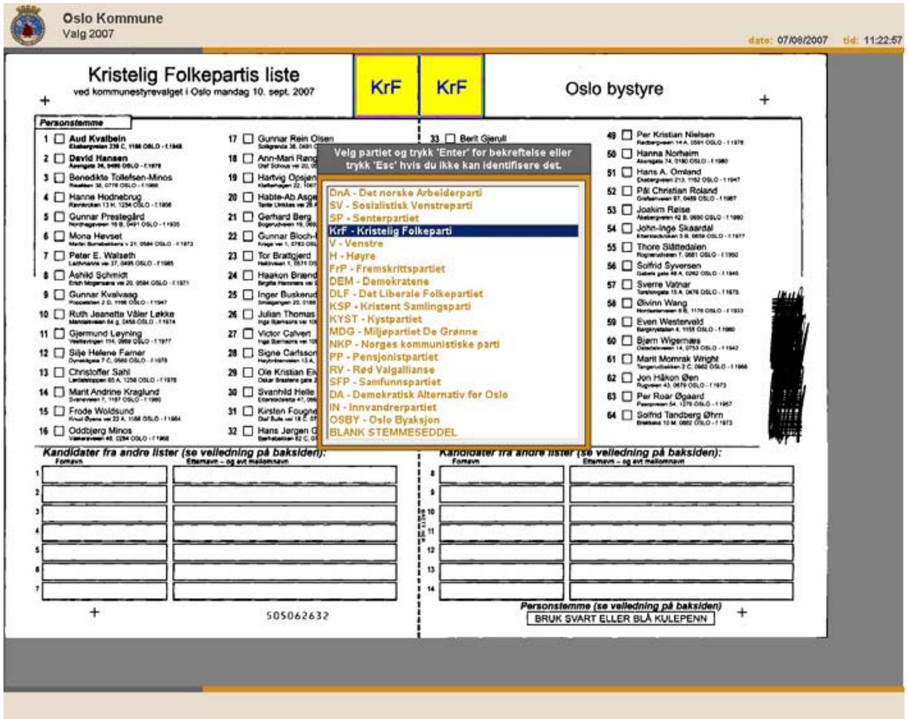
Elaboration of Requirement F 3.3.6
Once the polls are closed, votes are preliminary counted at the Polling Station. The Polling Committee will fill in the corresponding protocol on the PC screen and the preliminary results data. All that information is transmitted electronically to the e-counting servers.

When the ballot boxes arrive at the counting centre they are scanned and votes to party lists plus blanks are compared with the preliminary results transmitted from the polling stations. This comparison is a pre-condition to progress any further into the counting process.

The advance votes are scanned twice: one to provide the preliminary results and the second time to provide the final results.

Elaboration of Requirement F 3.3.9
The system presents the user with the preliminary and the final results per kret. The user of the Electoral Committee has to confirm the final results based on the comparison with the preliminary ones. If they do not balance, the confirmation of the kret results are put on hold until those votes are counted again (i.e. re-scan the ballot box) and a second set of results are in the system. The user is then able to compare the re-count results with the preliminary ones and take a decision base on that comparison.

Elaboration of Requirement F 3.3.12
When the system detects a doubtful ballot, its image is displayed to the verification operator together with what the system is actually interpreting. After assessment of the information, the votes can be accepted or rejected.
The system stores the image of the ballot paper together with the information read as corrected by the verification operator .



Oslo Kommune
Valg 2007

date: 07/08/2007 tid: 11:22:57

Kristelig Folkepartis liste
ved kommunestyrevalget i Oslo mandag 10. sept. 2007

KrF KrF

Oslo bystyre

Personstemme

1 □ Aud Kvalbein
2 □ David Hansen
3 □ Benedikte Tollefsen-Minos
4 □ Hanne Hodnebrug
5 □ Gunnar Prestegård
6 □ Mona Hevset
7 □ Peter E. Walseth
8 □ Åshild Schmidt
9 □ Gunnar Kvalvaag
10 □ Ruth Jeanette Våler Løkke
11 □ Gjermund Løyning
12 □ Silje Helene Farner
13 □ Christoffer Sahl
14 □ Marit Andrine Kraglund
15 □ Frode Woldsund
16 □ Oddbjørg Minos

17 □ Gunnar Rein Olsen
18 □ Ann-Mari Røng
19 □ Hartvig Opsjøn
20 □ Habte-Ab Asge
21 □ Gerhard Berg
22 □ Gunnar Bloch-
23 □ Tor Brattgjerd
24 □ Haakon Brænd
25 □ Inger Buskerud
26 □ Julian Thomas
27 □ Victor Calvert
28 □ Signe Carlsson
29 □ Ole Kristian Ei
30 □ Svanhild Helle
31 □ Kirsten Fougne
32 □ Hans Jørgen G

33 □ Berit Gjerull

49 □ Per Kristian Nielsen
50 □ Hanna Norheim
51 □ Hans A. Omland
52 □ Pål Christian Roland
53 □ Joakim Reise
54 □ John-Inge Skaardal
55 □ Thore Slåttedalen
56 □ Solfrid Syversen
57 □ Sverre Vatnar
58 □ Øivinn Wang
59 □ Even Westervold
60 □ Bjørn Wigernæs
61 □ Marit Momrak Wright
62 □ Jon Håkon Øen
63 □ Per Roar Øgaard
64 □ Solfrid Tandberg Øhrn

Velg partiet og trykk 'Enter' for bekreftelse eller trykk 'Esc' hvis du ikke kan identifisere det.

DnA - Det norske Arbeiderparti
SV - Sosialistisk Venstreparti
SP - Senterpartiet
KrF - Kristelig Folkeparti
V - Venstre
H - Høyre
FrP - Fremskrittspartiet
DEM - Demokratene
DLF - Det Liberale Folkepartiet
KSP - Kristent Samlingsparti
KYST - Kystpartiet
MDG - Miljøpartiet De Grønne
NKP - Norges kommunistiske parti
PP - Pensjonistpartiet
RV - Rød Valgallianse
SFP - Samfunnspartiet
DA - Demokratisk Alternativ for Oslo
IN - Innvandrerpartiet
OSBY - Oslo Byaksjon
BLANK STEMMESEDDEL

Kandidater fra andre lister (se veiledning på baksiden):

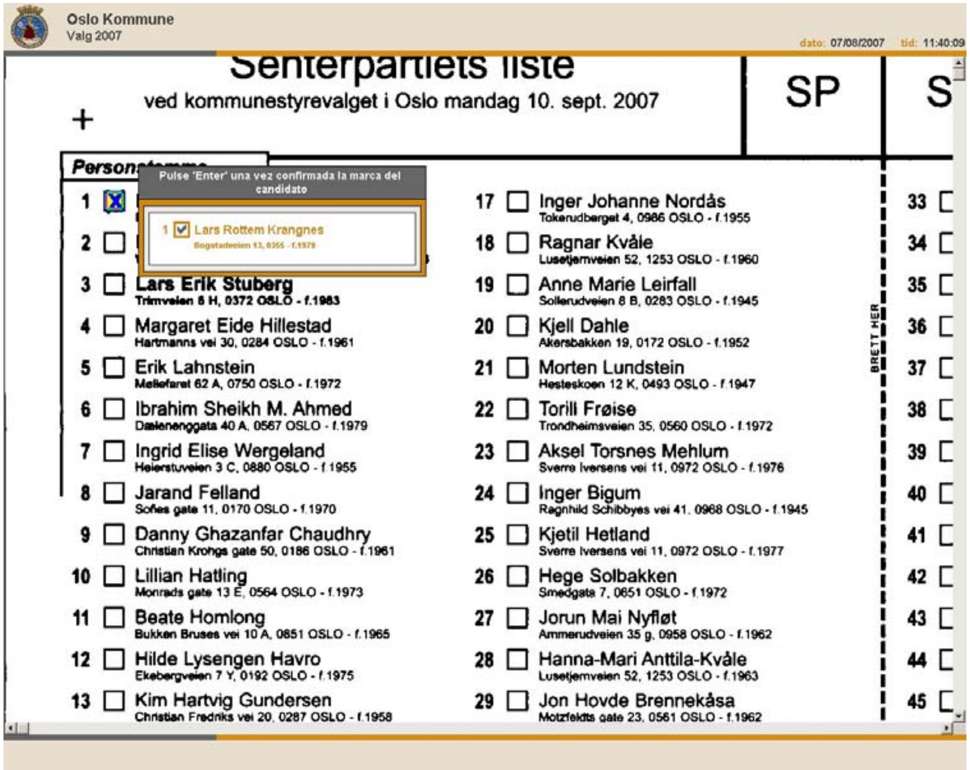
Fornavn	Etternavn – og evt mellomnavn
1	
2	
3	
4	
5	
6	
7	

Kandidater fra andre lister (se veiledning på baksiden):

Fornavn	Etternavn – og evt mellomnavn
8	
9	
10	
11	
12	
13	
14	

505062632

Personstemme (se veiledning på baksiden)
BRUK SVART ELLER BLÅ KULEPENN



Oslo Kommune
Valg 2007

dato: 07/08/2007 tid: 11:40:09

Senterpartiets liste
ved kommunestyrevalget i Oslo mandag 10. sept. 2007

SP S

Personstemme

Pulse 'Enter' una vez confirmada la marca del candidato

1 ☑ Lars Rottem Krangnes
Bogstadveien 13, 0355 - f.1979

1 ☒
2 □
3 □ Lars Erik Stuberg
Trimveien 6 H, 0372 OSLO - f.1983
4 □ Margaret Eide Hillestad
Hartmanns vei 30, 0284 OSLO - f.1961
5 □ Erik Lahnstein
Mellefaret 62 A, 0750 OSLO - f.1972
6 □ Ibrahim Sheikh M. Ahmed
Daelenenggata 40 A, 0567 OSLO - f.1979
7 □ Ingrid Elise Wergeland
Heierstuveien 3 C, 0880 OSLO - f.1955
8 □ Jarand Felland
Sofies gate 11, 0170 OSLO - f.1970
9 □ Danny Ghazanfar Chaudhry
Christian Krohgs gate 50, 0186 OSLO - f.1961
10 □ Lillian Hatling
Monrads gate 13 E, 0564 OSLO - f.1973
11 □ Beate Homlong
Bukken Bruses vei 10 A, 0851 OSLO - f.1965
12 □ Hilde Lysengen Havro
Ekebergveien 7 Y, 0192 OSLO - f.1975
13 □ Kim Hartvig Gundersen
Christian Fredriks vei 20, 0287 OSLO - f.1958

17 □ Inger Johanne Nordås
Tokerudberget 4, 0986 OSLO - f.1955
18 □ Ragnar Kvåle
Lusetjernveien 52, 1253 OSLO - f.1960
19 □ Anne Marie Leirfall
Sollerudveien 8 B, 0283 OSLO - f.1945
20 □ Kjell Dahle
Akersbakken 19, 0172 OSLO - f.1952
21 □ Morten Lundstein
Hesteskoen 12 K, 0493 OSLO - f.1947
22 □ Torill Frøise
Trondheimsveien 35, 0560 OSLO - f.1972
23 □ Aksel Torsnes Mehlum
Sverre Iversens vei 11, 0972 OSLO - f.1976
24 □ Inger Bigum
Ragnhild Schibbyes vei 41, 0968 OSLO - f.1945
25 □ Kjetil Hetland
Sverre Iversens vei 11, 0972 OSLO - f.1977
26 □ Hege Solbakken
Smedgata 7, 0651 OSLO - f.1972
27 □ Jorun Mai Nyfløt
Ammerudveien 35 g, 0958 OSLO - f.1962
28 □ Hanna-Mari Anttila-Kvåle
Lusetjernveien 52, 1253 OSLO - f.1963
29 □ Jon Hovde Brennekåsa
Motzfeldts gate 23, 0561 OSLO - f.1962

BRETT HER

33 □
34 □
35 □
36 □
37 □
38 □
39 □
40 □
41 □
42 □
43 □
44 □
45 □

3.11. Use Case 3.4 Counting of e-votes

Please, check the use case 2.1 (e-voting) description, where also the counting of e-votes has been described.

Elaboration of Requirement F 3.4.1
We fully cover this requirement as described in the description before.

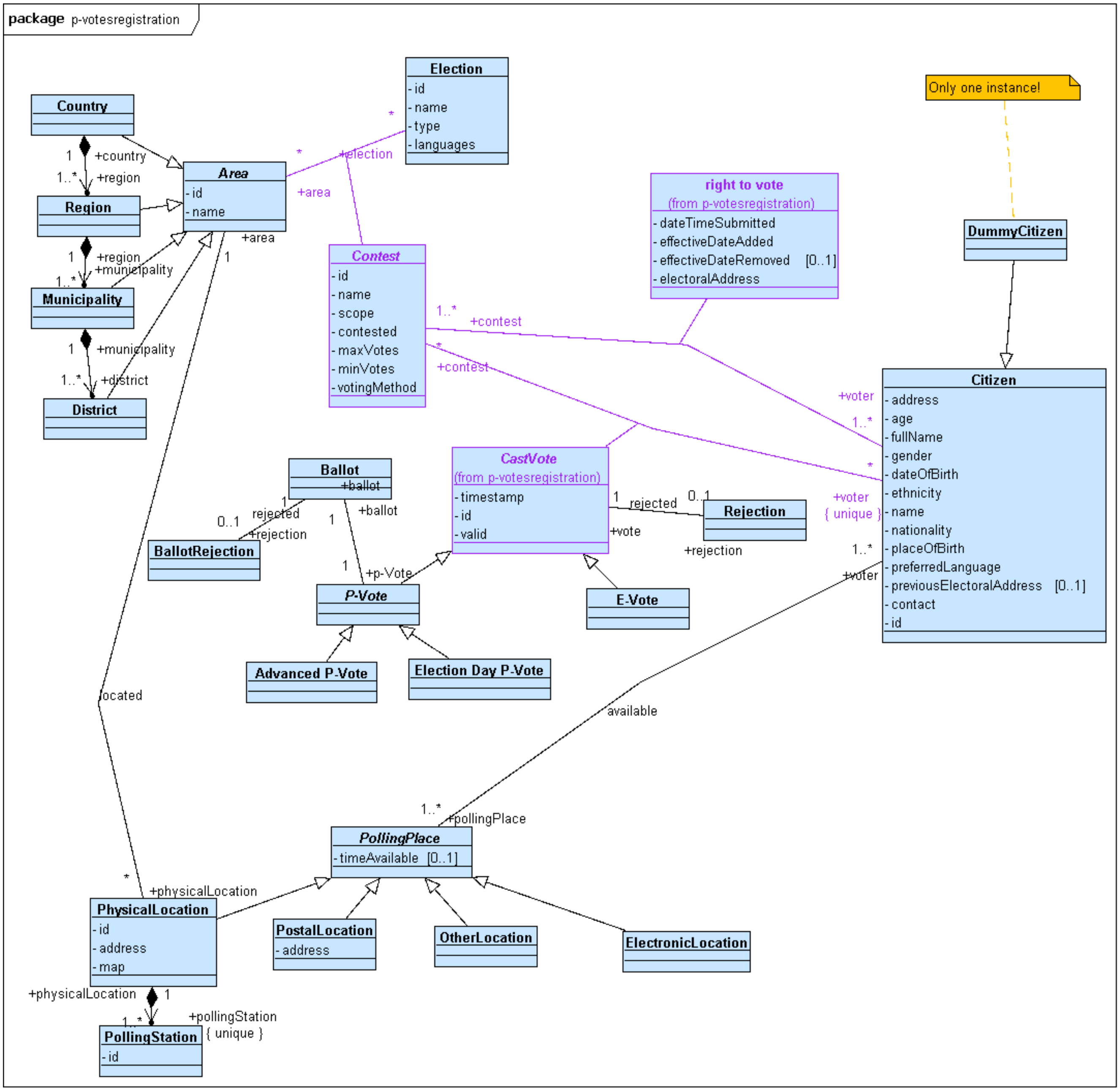
Regarding the small constituencies, the process that export the votes to the removable data store device to be sent to the isolated environment will check the number of votes per constituency and will alert the user, allowing (and forcing) the administrator to join constituencies when the minimum number of voters has not been reached.

Elaboration of Requirement P 3.4.1
Our solution architecture is built over standard J2EE infrastructure and scales horizontally in the isolated environment if needed.

Please refer to our architecture and infrastructure proposal for further details.

3.12. Use Case 3.5 Approval of p-votes and ballots

Approval of p-votes and ballots is provided through the implementation of a web application accessible to (allowed) Polling and Electoral Committees and County Electoral Committees, based on the provision of a set of services for the manipulation of the following conceptual domain model (fragment) for p-votes approval:



It can be noted that this conceptual model is (almost) identical to the already presented conceptual model for p-votes registration; the reason is that the approval of p-votes will be based on a set of services in order to check the validity of the previously registered p-votes.

This model supports several **Contest**s running in parallel, with a given voter (**Citizen**) casting votes for several of those **Contests** for which he/she has the right to vote.

Elaboration of Requirement F 3.5.1

We will fully comply with this requirement. Namely, the web application will provide a service to check **Advanced P-Vote** instances, for a given **Citizen** in a given **Contest** (please note that **CastVote** is an abstract class), in relation with the given **Citizen** inclusion in the Electoral Roll for a given Municipality (Contest), and in the absence of any other P-Vote instance precluding validity of the **Advance P-Vote** (please note that **P-Vote** is an abstract class, so it cannot be instantiated). The application will provide services to help check the validity of the Advance P-Vote regarding the conditions (§10-1 and §10-2) stated by the law, for the members of the Electoral Committees.

Elaboration of Requirement F 3.5.2

We will fully comply with this requirement. Namely, the web application will provide a service to check the voter (**Citizen**) as marked off. This will be based on the implementation of services for the manipulation of the **valid** property in the **CastVote** domain object; the system will enforce only one **CastVote** can be set to valid for a given voter (**Citizen**) in a given **Contest**. This mechanism allows the system to support scenarios where the elector (**Citizen**) is casting votes on several simultaneously run **Contest**s –so several marks in the Electoral Roll must be supported for a given **Citizen-**.

Elaboration of Requirement F 3.5.3

We will fully comply with this requirement. Namely, the web application will provide a service to set the voter (**Citizen**) as marked off. This will be based on the implementation of services for the manipulation of the **valid** property in the **CastVote** domain object; the system will enforce only one **CastVote** can be set to valid for a given voter (**Citizen**) in a given **Contest**. This mechanism allows the system to support scenarios where the elector (**Citizen**) is casting votes on several simultaneously run **Contest**s –so several marks in the Electoral Roll must be supported for a given **Citizen-**.

Elaboration of Requirement F 3.5.4

We will fully comply with this requirement through the implementation of the **Ballot** and **BallotRejection** Domain Objects and their associations (**rejected**), supported by the associated manipulation services. Details regarding the full set of properties to be included in the Domain Objects must be defined during analysis phase of the project. Further categorization of the **BallotRejection** reasons will be considered during the analysis phase of the project.

Elaboration of Requirement F 3.5.5

We will fully comply with this requirement. Namely, the web application will provide a service to set the vote (**CastVote**) as not valid, creating an instance of **Rejection** associated to it. This will be based on the implementation of services for the manipulation of the **valid** property in the **CastVote** domain object, the **Rejection** domain object and the associations between them. Further categorization of the **Rejection** reasons will be considered during the analysis phase of the project.

Elaboration of Requirement F 3.5.7

We will fully comply with this requirement. Namely, the web application will provide a service to set a **Ballot** as not valid, creating an instance of **BallotRejection** associated to it. This will be based on the implementation of services for the manipulation of the **Ballot** and **BallotRejection** domain objects and the associations between them, enforcing that a **Ballot** rejected has a **BallotRejection** reason.

3.13. Use Case 4.1 Reporting to SSB

Elaboration of Requirement F 4.1.1
In addition to the election protocols, the Election System will generate all sorts of post-election reports and statistics for publication. The system will be able to produce reports on all administrative levels. Please note that the system will produce a series of pre-configured reports which will have to be agreed before hand with the customer. Through the "post-election reports" interface (web interface), the authorised user will be able to produce customised reports by choosing amongst different selection criteria.

Elaboration of Requirement F 4.1.2
On top of the post-election reports, all sorts of system reports will also be available for the customer and auditors. Information on system performance, authentication data including successes, retries, IPs of origin etc…, data on usage….can is obtained at any time during the election period and post election too. All reports including post-election reports, will be able to be exported as PDF, XLS, TXT, EML or XML as required by the customer.

Elaboration of Requirement F 4.1.3
Transmission of reports to the SSB will be carried out in the way agreed with the SSB (currently a web service is in used)

Elaboration of Requirement F 4.1.6
Transmission of reports to the SSB will be carried out in the way agreed with the SSB. Currently a web service is in used. Through the web service the system receives confirmation on receipt. This confirmation message is stored in the system for record purposes.

Elaboration of Requirement F 4.1.8
Transmission of reports to the SSB will be carried out in the way agreed with the SSB. Currently a web service is in used. Through the web service the system receives confirmation on receipt. Exception messages received are also stored in the system in the way as confirmation messages.

3.14. Use Case 4.2 Settlement

Elaboration of Requirement F 4.2.1
The system does not allow the calculation of final election results (distribution of seats and returning of members) until after the database of final p-vote results is merged with the database of final e-votes results. This is a pre-condition for this functionality.

Currently the calculation of seats and returning members is performed in stages and a lot of information is produced in each stage. This information allows the corresponding Electoral Committee to perform cross-checks at every stage.

Elaboration of Requirement F 4.2.2
The Election System calculates the final results and distribution of seats as specified in the RPA Chapter 11. The Election System has full flexibility to implement other counting algorithms, which will be done should the election legislation be changed.

Currently the calculation of final results and distribution of seats is implemented in steps as reflected in the following example screens. All information can be exported to external system or printed as required by the customer.

3.15. Use Case 5.1 Reporting

Reporting functionality is provided through the integration of a reporting suite in the proposed solution. The particular suite to be integrated must be defined during the design phase of the project. The functionality provided by this suite will be augmented with custom developments in order to fully comply with all the requirements. Available Open Source Software suites will be the preferred solution.

Elaboration of Requirement F 5.1.1
We will fully comply with this requirement. A relation with the Election/Referendum the report was/will be created for will be maintained in the reports repository in order to allow easy identification to the designer. Several implementations of this relationship can be sought: as metadata associated to the report specification, as report specification naming standards, … The particular form will be defined during the design phase of the project, as are dependent on the functionality offered by the reporting suite the solution will be based on.

Elaboration of Requirement F 5.1.2
We will fully comply with this requirement. The solution will provide the designer with an editor for the easy definition of reports and report templates, allowing several modes of operation (graphical, textual, …) The solution will provide support for configuration of the time and date the report will be run by the system. Details will be defined during the design phase of the project, as are dependent on the functionality offered by the reporting suite the solution will be based on.

Elaboration of Requirement F 5.1.3
We will fully comply with this requirement. Details will be defined during the design phase of the project, as are dependent on the functionality offered by the reporting suite the solution will be based on.

Elaboration of Requirement F 5.1.4
We will fully comply with this requirement.

Elaboration of Requirement F 5.1.5
We will fully comply with this requirement. The Electoral Committee will be provided with a web-based solution, based on the reporting suite, in order to allow the execution of the available reports for a given Election.

Elaboration of Requirement F 5.1.6
We will fully comply with this requirement.

Elaboration of Requirement F 5.1.7
We will fully comply with this requirement.

Elaboration of Requirement F 5.1.8
We will fully comply with this requirement. It is expected that custom development will be needed in order to support (but not limited to) EML export of the reports, some image export formats and approval, general e-mail delivery and SSB delivery of the report results and authorization of report results for data publishing. Details will be defined during the design phase of the project, as are dependent on the functionality offered by the reporting suite the solution will be based on.

Elaboration of Requirement F 5.1.9

We will fully comply with this requirement. A relation with the Election/Referendum the report was/will be created for will be maintained in the reports repository in order to allow easy identification to the designer. Several implementations of this relationship can be sought: as metadata associated to the report specification, as report specification naming standards, … The particular form will be defined during the design phase of the project, as are dependent on the functionality offered by the reporting suite the solution will be based on.

Elaboration of Requirement F 5.1.10

We will fully comply with this requirement. The solution will provide the designer with an editor for the easy modification of reports and report templates, allowing several modes of operation (graphical, textual, …). Details will be defined during the design phase of the project, as are dependent on the functionality offered by the reporting suite the solution will be based on.

3.16. Use Case 5.2 Auditing

Elaboration of Requirement F 5.2.1

- Event logs: The election system shall implement an audit system and log all significant events at all levels (logical, technical, application) that may include the following:
  - Election related transactions
  - Attacks on the operations of the election system including the election infrastructure as well as the election application
  - Events happening on the OS and network level
  - etc
- Audit function shall continuously monitor the logged transactions and shall detect anomalous behavior immediately.

Elaboration of Requirement F 5.2.2

Auditor’s access through the role based access control shall be provided according to its required access and permission, in which case, the auditor shall be able to filter, search through all audit logs among other audit tasks. The auditor shall have access to all levels of logs (application logs, OS logs, network logs, etc)

Elaboration of Requirement F 5.2.3

The election system is designed in such a way that all audit logs are converted into a readable report format and auditors with proper access permission shall be able to use this reporting mechanism.

Elaboration of Requirement F 5.2.4

The election system audit function shall continuously run on the background, monitoring the event logs and detecting anomalous activities and behavior. A real-time warning shall be created and promptly inform the election officers or auditors. In addition, auditors shall be able to create parameters and configuration to detect other abnormal behaviors they deem necessary.

Elaboration of Requirement F 5.2.5

A real-time warning shall be created and promptly inform the election officers or auditors. In addition, auditors shall be able to create parameters and configuration to detect other abnormal behaviors they deem necessary

Elaboration of Requirement F 5.2.6

A real-time warning shall be created and promptly inform the election officers or auditors. In addition, auditors shall be able to create parameters and configuration to detect other abnormal behaviors they deem necessary

Elaboration of Requirement P 5.2.1
The election solution proposed is designed in such a way that one activity performed by anyone such as auditing tasks shall not interrupt, disrupt or affect in anyway the performance of the election system and its platform as a whole.

Elaboration of Requirement P 5.2.2
The proposed solution is designed to withstand interactively with very minimal delay on all processes. Any audit tasks undertaken shall not in any way affect the performance of the whole election system nor shall delay the times when using the audit system.

3.17. Use Case 9.1 Authentication

Elaboration of Requirement F 9.1.3
We will fully comply with this requirement. A login screen will be provided as part of the system.

Elaboration of Requirement F 9.1.9
We will fully comply with this requirement.
For the internal users of the system, they will be provided with user/password pairs or a smart card with a certificate fully managed by the system. Examples of internal users of the system are the Electoral Committee members, Polling Committee members, System Administrators, etc.
External users of the system will be using the eID; validation of the user will be performed interfacing according to SAML 2.0 standard with DIFI operated infrastructure. Examples of external users are Voters, Political party representatives, etc.
In any case, once the user is validated, the set –if any- of allowed roles will be presented to the user in order to choose the one to be used in this system’s session. If the user is trying to access an application different from the e-voting web application, and no role is found for the user, the system will redirect the user to the screen where he/she can apply for access rights.

Integration with DIFI operated Authentication Infrastructure, following the SAML 2.0 standard specification, will be included as part of the project. Given the variability in low-level mechanisms allowed by the specification, integration with other SAML 2.0 compliant infrastructure can be implemented as part of the software maintenance agreement.

Elaboration of Requirement F 9.1.10
We will fully comply with this requirement. Once the user is validated, the set of available roles to that user will be presented, and he/she must select the one to use in this session.

Elaboration of Requirement F 9.1.11
We will fully comply with this requirement. The system can issue temporary voter credentials for those voters without eID; this credentials will be usable during a time range (configured in the system) and once used, will automatically become unusable. During the design phase of the project the particular support for this temporary voter credentials will be defined (for example, it could be and smart card PIN protected and containing the X.509 certificate identifying the user; this certificate can be set the validity range dates, that will render the certificate invalid automatically out of the allowed time frame; additionally the system will control the already used certificates in order to allow only one usage per certificate issued)

Elaboration of Requirement F 9.1.12

We will fully comply with this requirement. A login screen will be provided as part of the system.

Elaboration of Requirement F 9.1.13

We will fully comply with this requirement. The system can issue temporary voter credentials for those voters without eID; this credentials will be usable during a time range (configured in the system) and once used, will automatically become unusable. During the design phase of the project the particular support for this temporary voter credentials will be defined (for example, it could be and smart card PIN protected and containing the X.509 certificate identifying the user; this certificate can be set the validity range dates, that will render the certificate invalid automatically out of the allowed time frame; additionally the system will control the already used certificates in order to allow only one usage per certificate issued)

Elaboration of Requirement F 9.1.14

We will fully comply with this requirement. The system will provide the requested user interface elements in order to allow the user to present his/her credentials, be informed of the validity of the credentials and allow selection of the available roles to the user if successfully logged-in.

4. Elaboration of Accessibility and Usability Requirements

Elaboration of Requirement AU 1
The XHTML will be logically and semantically structured to facilitate the automated processing of contents and subsequent comprehension by the support devices.

In this sense, the contents structure will be perfectly legible in the absence of style sheets or script programming. Therefore, the content of these pages can be offered by different web browser devices or configurations.

Likewise, the contents will be adequately labelled in semantic terms. Each content piece will be treated with the correct system, using the different markers available in HTML language to obtain correctly formed structures that express the different information treatment degrees. This will be required to explain the relevance of headers (h1, h2,...) or the different element lists (ul, li, ...) will be marked, also highlighting some text fragments (strong) to facilitate their understanding through the support devices.

The information will be organised in a logical structure for the standard operations and reading comprehension, with a clear and simple treatment of the contents offered by the site. Therefore, full access will be facilitated to all types of objective users to the contents.

Elaboration of Requirement AU2
Based on the development of a web client based on W3C standards (XHTML and CCS), the product developed must operate in all browsers that comply with the said standards.

However, we are aware that some browsers make different interpretations of the standards, so that Indra can make the necessary adjustments to the coding, so that the use experience in any browser is satisfactory and highly similar between them.

Therefore, these statements can be used in different operating systems that are commonly used.

The combination of browsers and operating systems that are compatible and tested for the product are as follows, with the purpose of covering the largest web browser market share:

- **Windows:**
  - Internet Explorer, versions: 6, 7, 8 and higher.
  - Firefox, versions: 2, 3 and higher.
  - Chrome, versions: 2 and higher.
  - Opera, versions: 9 and higher.

- **MacOs:**
  - Safari, versions: 4 and higher.
  - Firefox, versions: 2, 3 and higher.

- **Linux:**
  - Lynx.
  - Firefox, versions: 2, 3 and higher.

Elaboration of Requirement AU3

The product developed will have a clear multi-language character. This is required to set forth a user interface design and a technological solution that will facilitate its evolution to new languages.

The user interface will be fully translated, including all of its elements (text, audio, video, images, attachments, etc.) for all languages offered.

Currently, a prototype is offered, which offers the English and Norwegian languages, although new languages can be added with the final product delivered.

Elaboration of Requirement AU4

The prototype solution presented does not offer the possibility of attaching files, but the client can always state the need for such a functionality.

Indra proposes solutions that are accessible to the different formats proposed (ODF or OOXML), considering the treatments required by original files, in order to improve their accessibility.

These good practices will be used and complemented when creating files in PDF format, so that these can also be adequately handled by the solutions assisting different disabilities.

Elaboration of Requirement AU5

Pages are considerably smaller in size, given the fact that the development mode is based on standards and forms recommended by W3C, so that each page will have a maximum of 200Kb in all cases.

In addition, the layers are separated between contents, presentation and logic, as well as the adequate use of cache mechanisms to minimise the download times and promote the use experience.

Elaboration of Requirement AU6

Accordingly, we are proposing a system that is independent from the platform where it is run, which does not require specific plug-ins and is not dependent on local information storage devices (i.e. cookies).

Elaboration of Requirement AU7

The design proposal is presented in the prototype - even when using a pilot prior to the project, with the corresponding fine tuning procedures - and it complies with a series of flexible design standards, so that the screen can be adequately adjusted to different sizes in proportion to the font size.

This type of design is focused on solving visual impairments, offering the access to the information with the font increase functionality, present in most market browsers.

The base resolution taken for the design is 1024x768px, so that the scale accepts greater resolutions with no problems. This resolution has been chosen because it is the most common one in the market, since 1024x768 and higher resolutions account for 95.5% of the market and only 4.5% of the market uses smaller resolutions.

Therefore, the area favours the accessibility when proposing better solutions from the user interface point of view, facilitating its use to any type of user.

Elaboration of Requirement AU8

The initial design is used as the base for the product interfaces that can be offered (and are offered):

- Mechanisms to increase and reduce the font size, in addition to the browser options, which will facilitate reading to those with a moderate visual impairment.

- Presentation options in various contrast degrees, so that the user can select different css solutions that offer different graphical results with the contrasts. Specific configurations can be offered for some particular visual impairment typologies, in relation to original and high contrast resolutions.

Elaboration of Requirement AU9
The web interface definition and development is set forth for browsing and interaction purposes with the operating system, regardless of the device used.

Therefore, good practices related to the use of a coherent and complete application are applied, regardless of the use of a mouse or other devices, avoiding the dependence on events that are directly dependant on specific devices.

In this sense, "onclick" and "onkeypress" events will be adequately complemented, which will be directly linked to specific devices; the operations would not be possible without these devices.

Elaboration of Requirement AU10
The standard system fonts will be used, always taking into account that the use of this type of fonts will allow us to use the scaling features correctly. For the design presented, uniform use of the Arial font is made.

Elaboration of Requirement AU11
The purpose of the proposal designed for the e-ID screen is to facilitate the identification of the user with the simplest and safest possible method.

Therefore, a screen offers an operation based on a standardised system, that is not only simple and safe, but also allows the identification of the user with a password, or e-ID.

This is a very common mechanism in different digital applications, with a series of additional mechanisms that facilitate its use, such as the language change features or access to help screens that describe the operation details.

Elaboration of Requirement AU12
The use of additional technologies to the standard W3C technologies are applied with the main purpose of facilitating the interaction with the user and improving the performance experience obtained.

However, take into account that it only uses javascript, in accordance with the ECMAScript standard, with a non-intrusive approach. Therefore, the product is fully operational in the absence of javascript, even though its presence facilitates the performance.

On the other hand, no browser plug-ins need to be downloaded, in the form of applets, flash, silverlight or any other additional technology.

Elaboration of Requirement AU13
In relation to the previous point, the use of non-intrusive javascript technologies is adopted, so that the application is fully functional without this technology.

This is directly applicable to the use through other assisted devices, which progressively incorporate the compatibility with this type of de facto standards, but can also fully access the information and provide the interaction with the product.

Elaboration of Requirement AU14

The latency time forecasted for the elections application will be less than the 2 seconds established. Therefore, all server systems will be optimised during the project to reduce the response times while defining and using all cache mechanisms (server and client) required to avoid the presence of unnecessary information loads.

In case the said times are exceptionally exceeded, an information system will be available to the user, which will state the progress of the activity at all times, making sure that the user does not feel that the system is down. Similar methods to those used in progress bars during loading or operation procedures will be used.

5. Elaboration of Security Requirements

Elaboration of Requirement OS 0.4

- The voter voted more than once in an uncontrolled environment:
  If the voter voted more than once using e-vote in an uncontrolled environment, the e-vote with the latest timestamp will take precedence at all times. The election system will implement a mechanism to query and identify the timestamp of the e-votes to ensure that only one valid vote is counted per voter per contest.

- The voter voted more than once in a controlled environment:
  If the voter voted more than once in a controlled environment, the e-vote with latest time stamp will take precedence at all times. The election system will implement a mechanism to query and identify the timestamp of the e-votes to ensure that only one valid vote is counted per voter per contest.

- The voter voted more than once using e-vote in an uncontrolled and controlled environment:
  If the voter voted twice using e-vote in an uncontrolled environment as well as in a controlled environment, the vote casted in a controlled environment will take precedence at all times. The election system will implement a mechanism to identify e-votes done in a controlled environment and e-votes done in an uncontrolled environment to ensure that only one valid vote is counted per voter per contest.

- The voter voted twice using the e-vote and the p-vote:
  At end of the Election Day and upon closing the voting period, all p-votes will be counted and all e-votes that contain a p-vote mark (signifying that the owner of the e-vote also voted in p-vote) will be discarded, ensuring that the p-vote takes precedence over e-votes at all times to ensure that only one valid vote per voter per contest is counted.

- The voter voted twice using the p-vote:
  Case 1:
  If the voter voted in advance using a p-vote and cast another p-vote in a special envelope on election day, the Electoral Committee will invalidate the second p-vote during the process of approval of p-votes and ballots on election night.

  Case 2:
  In normal operation, the system prevents a voter casting two p-votes on election day since the voter is marked off in the electronic electoral roll (central server) when he casts the first p-vote. The only exception to this is the case of loss of communication with the central server (electronic electoral roll). In this case the voter is marked off in the local copy of electoral roll. In polling stations with several computers (electronic poll books), there is a theoretical possibility of a voter casting two p-votes when in this period of time.
  Since we do not recommend stopping the voting process during periods of loss of communication, we would recommend splitting the electoral roll in volumes and loading one volume per computer in the same way as the paper poll books are split in volumes (A to M and N to Z for instance). This way, voters that would want to cast a second p-vote have to go through the same computer where he was marked off the first time.

  Indra would like to propose the following option: In the future when e-voting becomes the standard way of casting a vote and p-votes become residual, the Customer may want to assess the option of establishing that all p-votes on election day are cast in special envelopes to promote the concept of vote anywhere.

Elaboration of Requirement OS 0.6

All votes that will be stored in the central server will remain encrypted and can only be decrypted in an isolated environment server that is not connected to the election system or to any network in a physical and logical manner. The only way to decrypt the encrypted votes is using a private key that was generated together with a public key that was used to encrypt the votes.

The private key will be divided into several parts and will be distributed to a number of authorized election officers. These authorized election officers must agree and combine every part of the private key that they receive before any of them can decrypt the votes. Each of the officers will not be given any information as to which part of the private key they have.

RSA cryptography will be implemented using 2048 bits encryption key length.

This encryption mechanism and the divided private keys that is distributed to a number of authorized election officers will ensure that a check and balance is implemented and that no one person will be able to decrypt the votes at anytime.

Elaboration of Requirement OS 0.9

The election system shall implement High Availability setup in all levels:

- Redundant WAN connections
- Redundant firewalls, switches and load balancers
- Redundant servers for critical applications
- Redundant NIC on the servers
- Hardware RAID in each server with redundant hard disk
- UPS for the whole infrastructure
- Internally the databases will be replicated between the server managing the same role

Elaboration of Requirement OS 0.12

When the voter calls the IVR to confirm the vote, the voter will be asked to provide the voter's information (e.g. VoterID) and the IVR will then provide the information of the vote related to the VoterID (using the party code). The final process would be for the voter to confirm or discard the vote by pressing a certain key that will be provided by the IVR during the call. If the voter confirms the vote, the voter will be informed that the vote has been cast and properly recorded.

As an option Indra offers a separate information website where the voter will be able to check whether the vote is cast by introducing the VoterID, including the date and time when the vote was cast.

Elaboration of Requirement OS 0.12B

The election system shall be able to provide the e-voter the end-to-end election process after the election period, by demonstrating the process of collecting the encrypted e-vote, transferring the encrypted e-vote to the isolated environment by an authorized election officer, combining all the parts of the private key and decrypting the e-vote, validating the voterID and presenting to the e-voter the cast vote. This process shall provide the e-voter the proof that the e-vote was correctly counted as intended and that the encrypted vote was obtained in a controlled environment, which is the central server for the election system.

All items (encrypted e-votes, private keys, passwords, etc) required in this kind of situation shall be retained and stored in a secure and protected environment for at least 4 years (or more, depending on the legal requirement) to be able to demonstrate an end-to-end process to confirm the counting of every e-vote

Elaboration of Requirement OS 0.13
The election system is designed in such a way that the voter will be able to confirm whether his vote was successful or not and if the vote has been properly recorded or not. Upon voting, if the voter provided an invalid party code, the voter shall be asked to review the party codes provided in the voter's card, if the party code is valid, the election system will accept the vote. However, the voter will have to call the implemented IVR system to confirm the vote using his voterID. In this way, the voter is assured that his vote has been properly recorded and the voter's choice has been accurately represented.

Elaboration of Requirement OS 0.14
The election system cannot provide any information about the number of votes cast for any voting option until the voting process has ended and the encrypted votes are sent to the isolated environment for decryption.

Elaboration of Requirement OS 0.15
The election system will implement NTP (Network Time Protocol) across the whole system.

The NTP server will be a dedicated appliance stratum 1 level deriving accurate time directly from the atomic clocks aboard the GPS satellite system.

In this setup, typical attacks modifying the NTP queries trough Internet, and expiring the certificates, will be avoided.

Elaboration of Requirement OS 0.17
Election related data such as lists of candidates when sent out for ballot paper production shall be encrypted in form.

Elaboration of Requirement OS 0.18
The election system is designed to allow the voter to vote as many times as the voter wishes. The right to vote is retained at all times even if the voter aborts the voting process for technical or whatever reasons.

Elaboration of Requirement OS 1.2
The election system shall use the eID to be provided by the Norwegian Government. In addition, the VoterID included in the voter card will be used. The election system shall provide a VoterID for every voter that is totally unique to other voters.

Elaboration of Requirement OS 3.1
The election system shall ensure that all stored votes are, and will remain anonymous throughout and even after the election process by storing only the vote cast without the information of the voter. After a voter casts the vote and confirmed the vote through the IVR, the vote will then be encrypted and recorded, without the information of the voter. In this way, the votes stored and encrypted is anonymous and will remain anonymous and there will be no way to reconstruct a link between the vote and voter at any time.

Elaboration of Requirement OS 3.3
The election system shall implement a secure communication channel between the voter and the election system during the voting process. This means that a voter will have an encrypted communication channel while connected to the election system, preventing the information sent to the election system readable in clear text by

anyone who will try to hijack the communication. To ensure the security of the information during transmission, a Secure Socket Layer (SSL) 256-bit encryption key will be implemented.

Elaboration of Requirement OS 3.4

After the polling period, the encrypted votes are transferred to the isolated environment server to be decrypted using the divided and distributed private keys and counted.

To comply with the requirements in the regulations, the media containing the votes in the isolated environment shall be in custody and safely stored by the election authority

However, Indra would recommend to destroy the physical media containing the votes after the election process

Elaboration of Requirement OS 3.5

In order for the user to confirm the vote, the voter will have to call the implemented IVR using the phone number provided together with the voter's card. In this process the IVR will inform the voter of the casted vote, in this case the party code and will then ask the voter to confirm or cancel the said vote. If the voter cancels the vote, the vote will not be recorded and if the voter confirms the vote, only then will the vote be recorded.

However, it is important to take note that the IVR will not give the voter the information whether the vote was counted or not, as this process takes place before the actual counting of the votes and that all e-votes are to be counted in a completely isolated environment and not on the central server where the votes stay encrypted the whole time.

Elaboration of Requirement OS 3.6

The audit system shall include the voting activities executed by the voter but does not include any information that provides a link between the voter and the vote. Information such as date and time of the vote, voterID and other voting operation. The election audit logs will never store and contain any information related to votes.

Elaboration of Requirement OS 4.1

The election system shall implement the following controls to ensure the secrecy of all votes at all stages in the election process.

- Cookies will not be used on the client's browser at any time to prevent attacks that involves cookie manipulation
- Secure communication channel between the voter and the election system using SSL-256 bit encryption
- All votes to be transmitted to the central sever will be encrypted through an encrypted channel
- All votes stored in the central server will remain encrypted throughout the election process without the information of the voter linked to a certain vote
- All encrypted e-votes will only be decrypted in the isolated environment server
- To decrypt the e-votes, the authorized election officers who received parts of the divided and distributed private key must agree and must combine every part of the private key in order to decrypt the e-votes.

Elaboration of Requirement OS 4.2

Public and Private Keys

Cryptographic public key to be used to encrypt the votes will be generated in an isolated environment server and will be stored in the central server at all times. The private key pair of the public key will be divided and distributed to all authorized election officers and will stay in their possession using a smart card the whole time. But not one of them will be able to decrypt the votes alone

Party code hash value

Every voter will be provided with party codes, unique to each and every individual. These party codes will be used by every voter to vote for a certain party. Every voter will have his own party code that will be sent together with the voter's card. The election system will then calculate the hash value equivalent using SHA2 algorithm of all party codes sent to all voters and store it in the central server. During the voting process, after the voter votes for a party using the provided party code, the election system will validate the party code by recalculating the hash value and comparing the result to the stored hash values in the central server. If the hash value is valid, the voter will receive a message that the vote was successful and properly recorded, if the hash value is not valid, the voter will be informed that the party code used was invalid and will be asked to review the party code on the voter card. The hash values of party codes stored will not in any way be used to connect or relate a vote to a specific voter as this information will only be provided to authorize election officers. In case the hash file is stolen, it will remain unusable to the person who stole it as there is almost no way to re calculate the hash value back to its original state.

OS passwords

Operating system passwords will be stored and kept in a restricted storage where only authorized personnel will have access on it. Systems administrators will not be able to change administrator passwords without having to comply with policies and procedures that will be in place and without the approval of an authorized officer.

Election application system passwords

Election application passwords will be stored and kept in the same manner as the operating system passwords but in a different storage and access will be approved by a separate authorized personnel. Election application owners will not be able to change the passwords without having to comply with policies and procedures that will be implemented and approved by a separate authorized officer

Physical access

Physical access to the election devices and machines will be restricted to authorized personnel only. All access to and from the restricted area must comply with policies and procedures to be implemented and must be approved and logged.

Elaboration of Requirement OS 4.3

All information of voters will be stored in a protected system. The election system infrastructure will be designed to withstand any logical and physical attacks that might compromise the privacy and confidentiality of the voters.

Source code fortification: A thorough audit on all source codes within the election system will be done to identify, resolve or remove all badly coded source codes as well as those hazardous characters that might result to successful attacks such as SQL injection, Phishing, XSS, Link injection, etc. Source code fortification will ensure that the election system is protected from attacks that take advantage of badly coded source codes.

Network segregation: All publicly available servers (e.g. election web portals) will be located in the DMZ network, protected by an external firewall, an Intrusion prevention system (IPS) and DOS mitigation hardware. The application servers and Database servers will be located in a secure environment, logically separated from the external network and will be protected by another firewall and an access control rules and policies. In this case, only defined ports and services that are required in the process will be allowed and non-defined ports and services will be blocked and denied.

Physical access control: A physical security policy and procedure shall be put in place to all infrastructures involving the election system. Every physical access must comply with these policies and procedures and must be approved by authorized personnel. A clear description of the nature and purpose of every physical access must be documented and all entry/exit must be logged including the date and time.

Separation of duties: A role based access control system shall be implemented. OS administrators will be limited to OS level administration tasks, but will require an approval from authorized person before executing a critical and confidential action. The OS administrator will not have access to any election data as well as to election application system.

Election application administrators will be limited to administration tasks on the application level, but will require an approval from authorized personnel before executing critical or confidential actions. The application administrators will not have access to election related data at any time.

Elaboration of Requirement OS 4.7
The RSA 2048 bit encryption shall be implemented to encrypt the votes

The AES 256 bit key encryption shall be implemented to secure the communication between the voter and the election system

The SHA2 algorithm shall be used to generate the equivalent hash value of the party codes

Elaboration of Requirement OS 4.8
The election system shall implement SHA2 hashing algorithm to be used to calculate the equivalent hash value of the party codes to be provided to the voters.

Elaboration of Requirement OS 4.9
All e-votes will be stored in a hardened central server, removing all unnecessary services.

All votes that will be stored in the central server will remain encrypted using RSA 2048 and can only be decrypted in an isolated server that is not connected to the election system or to any network in a physical and logical manner. In addition, the only way to decrypt the encrypted votes is using a private key that was generated together with a public key.

The private key shall be divided into several parts and will be distributed to a number of authorized election officers. These authorized election officers must agree and combine every part of the private key that they receive before any of them can decrypt the votes. Each of the officers will not be given any information as to which part of the private key they have.

RSA cryptography will be implemented using 2048 bits encryption key length.

This encryption mechanism and the distributed private keys to a number of authorized election officers will ensure that a check and balance is implemented and that no one person will be able to decrypt the votes at anytime.

Elaboration of Requirement OS 4.11

Public and Private Keys
Cryptographic public key to be used to encrypt the votes will be generated in an isolated environment server and will be stored in the central server to be able to encrypt all the votes. The private key will be divided and

distributed to all authorized election officers and will stay in their possession the whole time. But not one of them will be able to encrypt the votes alone

Party code hash value

The Equivalent hash value of all party codes will be stored in the protected central server where logical and physical access are restricted and control. In addition, the hash value would be almost impossible to recalculate back to its original form. SHA2 algorithm is a one way hash and it would be very hard for any attacker to be able to know the equivalent party codes of every hash value.

OS passwords

Operating system passwords will be stored and kept in a restricted storage where only authorized personnel will have access on it. Systems administrators will not be able to change administrator passwords without having to comply with policies and procedures that will be in place and without the approval of an authorized officer.

Election application system passwords

Election application passwords will be stored and kept in the same manner as the operating system passwords but in a different storage and access will be approved by a separate authorized personnel. Election application owners will not be able to change the passwords without having to comply with policies and procedures that will be implemented and approved by a separate authorized officer

Physical access

Physical access to the location of these keys and passwords shall be restricted to authorized personnel only. All access to and from the restricted area must comply with policies and procedures to be implemented and must be approved and logged.

Elaboration of Requirement OS 5.1

Upon connecting to the election web portal, a welcome page that contains information of the means to verify the authenticity of the web portal will be presented to the voter. The election system shall implement a web certificate for the election web portal. The voter will be able to verify the authenticity of the election system once connected through the information provided by the web certificate in the web portal. To ensure the authenticity of the election web portal the voter is connected to, the web certificate will provide the following information:

- Owner and identity of the web site
- Provider of the certificate
- Digital certificate of the website
- The certificate validity, etc

In addition, the election system shall implement an IVR (Interactive Voice Response) system that enables the voter to confirm the cast vote after the voting process to prevent voters from voting in fake election web portal trying to impersonate and mimic the authentic election web portal.

Elaboration of Requirement OS 5.4

Local access to election system components will require a smart card. This will prevent an OS systems administrator with no access to election system components from making an unauthorized access. Only authorized personnel will be able to open any component of the election system through the role base access control mechanism. All other accounts (user or admin) that have no authorization will be limited to OS level access only.

Remote or on-line access to election system components will be done through a virtual private network or VPN. Prior to establishing a communication, an authentication process will occur forcing the user to provide the valid credentials using a smart card. The same role based access control mechanism shall be implemented to all users and ensure that only the required tasks and objects configured for every user shall be presented and available to the user.

Elaboration of Requirement OS 6.1

The election system shall implement policies and rules to prevent anyone from executing unauthorized codes on any platform involving the election system. The use of removable media (CD, DVD) and external drives (usb flash drives, usb external HD) shall not be allowed. The same policy shall automatically deny the execution of files such as .exe, .bat, .com and other executable codes.

Application users and administrators shall be limited tasks related to application levels and will not have the permission to run executable codes.

The OS users and administrators shall be restricted and limited to administrations tasks only and will not be allowed to open executable files. Default administrator accounts will be renamed and will not be used in day-to-day admin tasks of the election platform. A username identified to a user that is provided specific administration tasks shall be configured without including the right and permission to open executable files.

Elaboration of Requirement OS 6.2

The election system shall implement an automated inventory system of the system configuration of election platforms that will include the approved and certified installed. The generated inventory list shall be made available to any review process to be able to validate and prove that no changes have been implemented on the system configuration and that no unauthorized software has been installed.

Elaboration of Requirement OS 7.1

The election system shall implement a role based access control. All users will only have access to services that he is required and allowed to access which includes the jobs that is required of him. The user will not be able to view, modify or delete anything on the services that he does not have access to. Once the user is authenticated, he will only be presented the interface that he is allowed and nothing more.

Elaboration of Requirement OS 7.2

Using the role based access control mechanism, all users will only have access to services that he is required and allowed to access which includes the jobs that is required of him. The user will not be able to view, modify or delete anything on the services that he does not have access to. Once the user is authenticated, he will only be presented the interface that he is allowed and nothing more. No action can be executed prior to a successful authentication

Elaboration of Requirement OS 7.3

The election system shall be configured in such a way that it will be sufficiently flexible, granular type of access control system and any changes or creation of new roles will be implemented without affecting other services or the election system as a whole

Elaboration of Requirement OS 7.6

In the election system design, all votes shall stay completely anonymous in all the stages of the election process. The only scenario where the votes resides together with the voterID is when the votes are processed and decrypted in the isolated environment, and can only be executed by a team of authorized personnel that possess the different parts of the distributed private key. No single authorized personnel will be able to decrypt the votes.

Elaboration of Requirement OS 7.7
The election system shall implement an asymmetric algorithm using RSA 2048 bit key encryption. The private key shall be encrypted using symmetric algorithm and shall be divided into several parts using secret sharing algorithm and providing each authorized election officers its own unique part of the key where parts or all of them are needed in order to reconstruct the key. In this case, we considered that counting on all authorized officers to combine together the secret keys might be impractical and therefore a threshold scheme shall be used where some of the secret key parts will be sufficient enough to reconstruct the key.

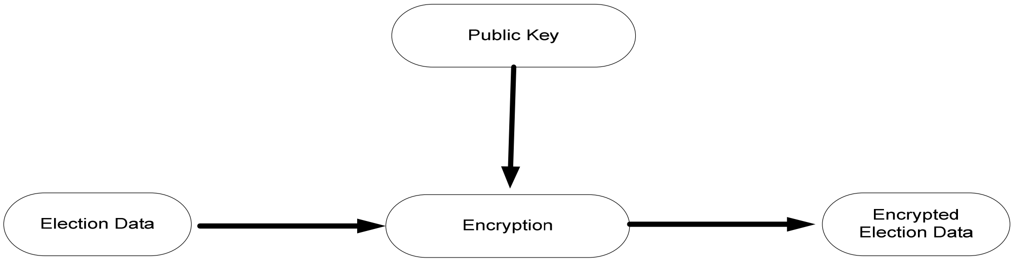
To elaborate, the election system shall implement the following process:

Public /Private key – Using RSA 2048, a public and public key shall be generated. The public key will be used to encrypt the election data. The private key shall be divided and distributed to a number of authorized personnel. To do this, see the next process:

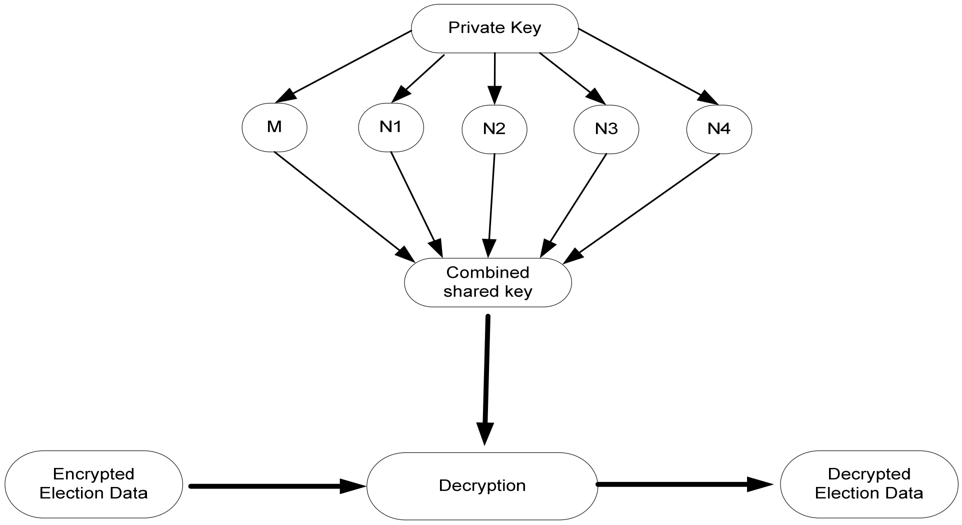
- Encrypt the private key – Using Bouncy Castle crypto package or the crypto++ library, a shared key using AES 256 bit algorithm shall be generated and used to encrypt the private key obtaining an encrypted private key.
- Sharing the secret key – To protect the easy decryption of the private key and avoid using it to decrypt the election data by one person only, the secret key to be used to decrypt and reveal the private key shall be divided into several parts using Secret sharing algorithm (Shamir's secret sharing scheme or Blakley's scheme). This method shall enable the distribution of some or all parts of the secret key. Each authorized election officer shall receive a share of the secret key. The secret key can be reconstructed only when a sufficient number of shares are combined together and individual shares of the key are of no use on their own.

To explain further, the election officer M gives a share of the secret to election officers N. Election officer M gives each of election officers N a share of the key in such a way that a group of election officers N (e.g. GN) together with election officer M can reconstruct the shared key, but no group of N that is fewer than GN shall be able to reconstruct the shared key. This ensures that even with the absence of some election officers N, the shared key can still be reconstructed. All these processes shall be integrated inside the proposed solution of Indra. See figure below:

Election data encryption workflow



Election data decryption workflow



Elaboration of Requirement OS 7.8
The election system will not require the implementation of an HSM. The RSA distributed key generation is a robust implementation of PKI that does not require a trusted party. In this case, a public and private key shall be generated in a totally isolated environment server that is in no way connected to other network physically or logically.

Elaboration of Requirement OS 7.9
As mentioned in OS 7.7, the election system shall encrypt the private using symmetric algorithm, preventing anyone from knowing the private key before it is to be used. The secret key used to encrypt which is the same key to be used to decrypt the private key shall be shared to a number of election officers using Secret Sharing Algorithm. The private key will only be revealed during the time which the election officers combine their shares of the secret key to decrypt the private key, which will then be used to decrypt the election data.

Elaboration of Requirement OS 7.11
The election system shall implement an SSL 256 bit AES encryption to provide a secure communication between the voter and the election system and prevent anyone from hijacking, intercepting and modifying election related data on the wire.

In addition, the election system shall implement a role based type of access control, providing only the required access specific to every user. Rights and permissions will be clearly defined and implemented to prevent any user from executing actions that the user is not allowed to execute. Access to election data shall require proper authentication and the right role with the right permission.

A separation of duties will be implemented to critical and/or confidential actions to ensure that no one person will be able to execute actions that will affect the election system or the confidential data within the election system

Elaboration of Requirement OS 7.14
The election system shall implement a role based access control mechanism, providing only the required access specific to every user. Rights and permissions will be clearly defined and implemented to prevent any user from executing actions that the user is not allowed to execute.

A separation of duties shall be implemented to critical and/or confidential actions to ensure that no one person will be able to execute actions that will affect the election system or the confidential data within the election system. Authorized personnel must have the approval of other authorized personnel and both must be authenticated to the election system prior to executing a critical or confidential action, leaving no one election officer to be able to execute these actions alone

Elaboration of Requirement OS 7.17
The election system shall implement a strong authentication system to all users including administrators, operators and auditors.

Local access to election system components will require a smart card.

Remote or on-line access to election system components will be done through a virtual private network or VPN. Prior to establishing a communication, an authentication process will occur forcing the user to provide the valid credentials using a smartcard.

Elaboration of Requirement OS 8.1
All acknowledge votes are stored in a redundant storage system, replicating data in real-time.

All acknowledge votes shall be stored and remain encrypted in the central server to prevent alteration.

Elaboration of Requirement OS 8.4
Voter to election system: The transfer of data between the voter to the election system is protected by 256 bit AES encryption and therefore not possible to hijack, intercept and modify the data on the wire

Central server to isolated server: E-votes will then be stored in the central server where it will remain encrypted throughout the whole election process. After the election process the encrypted e-votes will then be transferred to the isolated environment server by an authorized election officer manually. This process provides an end to end security to the data making it not possible to alter, delete or add votes to the encrypted data

Access control: All access to election data shall be logged including the user who accessed the data, date and time and the activities executed on a particular data

Elaboration of Requirement OS 8.5
Voter to election system: The transfer of data between the voter to the election system is protected by 256 bit AES encryption and therefore not possible to hijack, intercept and modify the data on the wire
Central server to isolated server: E-votes will then be stored in the central server where it will remain encrypted throughout the whole election process. After the election process the encrypted e-votes will then be transferred to the isolated environment server by an authorized election officer manually. This process provides an end to end security to the data making it not possible to alter, delete or add votes to the encrypted data during the transfer.

Elaboration of Requirement OS 8.6
Voter to election system: The transfer of data between the voters to the election system is protected by256 bit AES encryption and therefore not possible to hijack, intercept and modify the data on the wire. Authentication process shall be logged including the origin and the date and time of the authentication process.

Central server to isolated server: E-votes will then be stored in the central server where it will remain encrypted throughout the whole election process. After the election process the encrypted e-votes will then be transferred to the isolated environment server by an authorized election officer manually. This process provides an end to end security to the data making it not possible to alter, delete or add votes to the encrypted data during the transfer.

Elaboration of Requirement OS 8.7
Voters cast their votes using a specific party code that is unique to every voter. These party codes are provided together with the voter's card. Since this party code is assigned to a specific voter only, any manipulation of the vote and the party code shall be detected and the voter will be asked to review the voter cards. Take into consideration that the equivalent hash value of every party code is stored in the central server to validate the party code that the voter voted for.

Elaboration of Requirement OS 8.9
Access to recorded votes will not be allowed to anyone at anytime during the election process and therefore no changes can be done on the votes whether authorized or unauthorized in nature.

Elaboration of Requirement OS 8.10
In order for the user to confirm the vote, the voter will have to call the implemented IVR using the phone number provided together with the voter's card. In this process the IVR will inform the voter of the casted vote, in this case the party code, which is the shared secret between the system and the voter.

Receiving false messages will be avoided in this case due to the fact that the voter will be calling to the IVR system instead of the IVR system calling the voter that may be created by a fake IVR system

Elaboration of Requirement OS 8.12
The election system shall implement an isolated environment server that is totally separated from other networks physically and logically. It will be a standalone server where decryption and counting of e-votes will be done.

Elaboration of Requirement OS 8.13
During the e-vote process, recording of e-votes shall occur simultaneously together with the annotation or marking the voter in the electoral roll. If in any case, something failed in between, it is mandatory to redo the operation

During the p-vote process, when the voter introduce the ballot into the ballot box, the presiding officer will mark off the voter in the electoral roll as having voted in p-vote

Elaboration of Requirement OS 8.14
The election system shall implement a security control mechanism to communicate with external entities containing electoral roll and the parties that are going to provide the lists of candidates.

To enable the parties to provide the lists of candidates, a username and password shall be provided.

To enable a secure communication between the external entity containing electoral roll and the election system, a mutual certificate between the two parties or a VPN connection (LAN to LAN) shall be implemented

Elaboration of Requirement OS 8.15
To enable the election system to verify the integrity of p-vote results, a hash function algorithm shall be implemented. For every p-vote result, a hash equivalent value shall be calculated and shall be validated once received on the other end using the same process. In addition, a secure communication using an IPSEC tunnel

will be implemented. IPSEC also has the capability to verify the integrity of the communication between both sides.
The same implementation is to be used in other data transfers.

Elaboration of Requirement OS 8.16
To enable a secure communication between the external entity containing electoral roll and the election system, a mutual certificate between the two parties or a VPN connection (LAN to LAN) shall be implemented

Elaboration of Requirement OS 8.17
No function will be developed to reset the election system to its initial state after the polling phase has begun

Elaboration of Requirement OS 8.18
The election system will not in any way store the votes outside the controlled environment. All votes will be stored in a controlled environment and will remain encrypted until the whole election process ends

Elaboration of Requirement OS 8.19
The election system shall present each voter with the authentic ballot corresponding to the voter's constituency. In addition, the voter can verify the authenticity of the election web portal by doing the following:

Upon connecting to the election web portal, a welcome page that contains information of the means to verify the authenticity of the web portal will be presented to the voter. The election system shall implement a web certificate for the election web portal. The voter will be able to verify the authenticity of the election system once connected through the information provided by the web certificate in the web portal. To ensure the authenticity of the election web portal the voter is connected to, the web certificate will provide the following information:

- Owner and identity of the web site
- Provider of the certificate
- Digital certificate of the website
- The certificate validity, etc

Elaboration of Requirement OS 9.1
The election system shall implement a High-Availability configuration on all levels of the proposed solution, such as:

- Redundant WAN connections
- Redundant firewalls, switches and load balancers
- Redundant servers for critical applications
- Redundant NIC on the servers
- Hardware RAID in each server with redundant hard disk
- UPS for the whole infrastructure
- Internally the databases will be replicated between the server managing the same role

In case one of the systems fails, automatically the redundant system shall take over the failed function. In case a DOS/DDOS attack is launched, the proposed solution is designed to detect and automatically block such attacks.

Elaboration of Requirement OS 9.2
As mentioned in OS 9.1, the election system shall implement a high-availability setup in such a way that when one of the systems reboots (unintentionally or intentionally), the redundant system shall take over the function in real-time, thus ensuring the availability of the e-voting services at all times.

A backup configuration files shall be generated and stored in a secure manner so that in case a configuration of a certain system is corrupted, a prompt restoration of the said configuration shall be executed immediately.

Elaboration of Requirement OS 9.4
Monitoring system shall be implemented to monitor the services and features of the election system including the implemented security mechanisms to ensure that all systems are working properly. Any failure detected shall be reported in real-time. Election personnel shall monitor and check the functionalities of the election system and ensure that prompt action is executed once an alarm is received

Elaboration of Requirement OS 9.5
A DOS/DDOS mitigation hardware as well as an Intrusion Prevention System (IPS) shall be implemented to detect possible DOS/DDOS attacks. A reasonable number of connections coming from a specific IP will only be permitted at a given time and at reasonable size of packets. If a huge amount of packets is detected coming from a single source, the connection will be cut and blocked automatically. This will prevent the election system to use all of its resources replying to these malicious requests and deny the service to legitimate requests.

Elaboration of Requirement OS 10.1
Voters' register and lists of candidates shall be stored and maintained in a secure and controlled environment to ensure the authenticity, availability and integrity of these data. The security features and high availability implementation can be found in the document ¨Customer Technical Platform¨ in APPENDIX III

Elaboration of Requirement OS 11.5
The whole election system will be protected against all types of logical attacks as well as physical attacks

The election system will be protected with a hardened firewall, a DOS/DDOS mitigation hardware and an intrusion prevention system (IPS) configured in a stealth way. The stealth configuration will hide the IPS from anyone who tries to query on it and try to exploit it including DNS amplification attacks

Remote command execution attacks on the application level will not be possible as the election system will implement a source code fortification and removal of hazardous characters prior to deployment

DNS server shall implement the latest version of BIND (Berkeley Internet Domain Name) running on top a fully patched and secure linux operating system. To further strengthen the DNS server security and prevent attacks such as DNS cache poisoning and DNS ID spoofing, the following will be implemented:

- Physically separate external election DNS server from internal DNS servers
- Restrict zone transfers to authorized devices only
- Implement TSIG (Transaction Signatures) to digitally sign zone transfers and zone updates
- Restrict dynamic DNS updates (when possible)
- Not to broadcast BIND version used and hide as from any queries
- Harden the DNS server and remove all unnecessary services
- Deploy the DNS server in a dedicated machine

However, it will not be possible for the election system to mitigate DNS poisoning using arp poisoning on voters using an uncontrolled environment due to the fact that the election system does not have the authority or control over the machines that the voter will be using to vote. However, the proposed solution implements the 3rd channel (IVR) to confirm the vote.

Elaboration of Requirement OS 13.2

Monitoring system shall be implemented to monitor the services and features of the election system including the implemented security mechanisms to ensure that all systems are working properly. Any failure detected shall be reported in real-time. Election personnel shall monitor and check the functionalities of the election system and ensure that prompt action is executed once an alarm is received.

Elaboration of Requirement OS 13.3

All information retrieved from the audit system shall be converted into a readable format and will be provided to the election observers. This information will be focused on e-voting, p-voting, etc.

Elaboration of Requirement OS 13.4

Event logs: The election system shall implement an audit system and log all significant events at all levels (logical, technical, application) that may include the following:

- Election related transactions
- Attacks on the operations of the election system including the election infrastructure as well as the election application
- Events happening on the OS and network level
- etc

Audit function shall continuously monitor the logged transactions and shall detect anomalous behavior immediately.

Elaboration of Requirement OS 13.5

The election system shall implement an audit function that would provide an end-to-end solution for all transactions related to the election system, that includes recording of transactions, monitoring transactions and the ability to verify such transactions if needed.

Elaboration of Requirement OS 13.6

The election system audit function shall continuously run on the background, monitoring the event logs and detecting anomalous activities and behavior. A real-time warning shall be created and promptly inform the election officers or auditors. In addition, auditors shall be able to create parameters and configuration to detect other abnormal behaviors they deem necessary.

Elaboration of Requirement OS 13.8

As mentioned in OS 13.4, the election system shall implement an audit system that will be able to analyze and verify transaction logs in all levels especially on the election application system. The results shall be converted into a readable format and can be verified at any time if applicable legal provisions have been complied with

Elaboration of Requirement OS 13.9

The logging mechanism of the election system together with the audit system shall enable the possibility to cross-check and verify proper operations and accuracy of election results. As mentioned, all election related transactions shall be logged including possible voting fraud and an end-to-end proof of the authenticity of every counted votes

Elaboration of Requirement OS 13.10

The election system shall implement an audit function that would provide an end-to-end solution for all transactions related to the election system that includes recording and counting of e-votes as well as p-votes

The logging mechanism of the election system together with the audit system shall enable thpossibility to cross-check and verify proper operations and accuracy of election results. As mentioned, all election related transactions shall be logged including possible voting fraud and an end-to-end proof of the authenticity of the election results

Elaboration of Requirement OS 13.11
The audit system will be located in a controlled environment protected at all times against logical attacks as well as physical attacks that might result to records modification, deletion and removal.

Elaboration of Requirement OS 13.13
Since the proposed solution will implement a high-availability setup on server level, media level, hardware level and other levels of the election system, any failure or server crashes will not affect the functionalities as well as the integrity of the polling phase data. Additional information can be found in ¨Customer Technical Platform¨ in APPENDIX III

Elaboration of Requirement OS 14.1
The election system is designed in such a way that the electoral authority shall be able to test the election system to ensure its ingenuity and proper operations at any time. In addition, the election system shall implement an automated inventory system of the system configuration of election platforms that will include the approved and certified software installed.

In addition, Indra solution shall propose the process of hash calculation using SHA2 algorithm for all approved and certified versions of applications (e.g. election application) installed and provide them to the auditors, to enable them to verify and validate the authenticity and validity of the application running on the system. This will ensure that no newer or different versions aside from the approved and certified versions are installed in the election system.

Elaboration of Requirement OS 14.4
The election system shall implement an automated inventory system of the system configuration of election platforms that will include the approved and certified software installed. The generated inventory list shall be made available to any review process to be able to validate and prove that no changes have been implemented on the system configuration and that no unauthorized software has been installed.

6. Elaboration of External Interface Requirements

Elaboration of Requirement EIS 4

We will fully comply with this requirement. Given the diversity of low-level mechanisms supported by the SAML 2.0 standard, during the design phase of the project will be defined the mechanism to implement in order to support SSO with DIFI infrastructure or the Altinn portal. Only one mechanism will be supported as part of the project; additional mechanisms can be incorporated, if needed, subject to the software maintenance agreement.

Elaboration of Requirement EIS 5

We can fully comply with this requirement, should the customer require so. However at the moment it is not included in this proposal.

Elaboration of Requirement EIS6

We will fully comply with this requirement. During the analysis and design phases of the project, the EML messages available for this kind of import will be tailored to meet the requirements.

Elaboration of Requirement EIH1

We will fully comply with this requirement. All the commercial scanners support this kind of interfaces (TWAIN and ISIS)

Elaboration of Requirement EIH2

We will fully comply with this requirement. The PDFs provided by the system are fully compatible with the most number of industrial printers on the market.

7. Elaboration of Documentation Requirements

Elaboration of Requirement D 2

We will fully comply with this requirement. The set of technical documentation will be distributed in the following documents:

- Business Model: This document will describe the conceptual domain supported by the system and the set of processes and actors involved. UML will be the preferred formalism in order to document this knowledge. This document will serve as a base-line in order to allow for better decision making through the life of the system.
- Architectural Model: This model will describe the set of modules/services and the interfaces offered/required for each of them. UML, possibly including stereotypes, will be the preferred formalism for documenting this high-level design of the system.
- Detailed technical models: One model per module/service will describe the details of every part of the system. User interfaces will also be documented according to this formalism. UML, including stereotypes, will be the preferred way of documenting this low level design. Those stereotypes will be designed in such a way that they convey information about the technology used in the implementation

Elaboration of Requirement D 3

We will fully comply with this requirement. The set of installation and operation manuals will include:

- Installation guide, covering full installation of the software on the required hardware
- Configuration guide, covering full documentation of the system parameters, location and usage examples.
- Deployment plan, covering full documentation of the roll-out phases.
- Operation guide, covering the full set of operations of the system including information back-up and restore, important operations to be manually performed to support electoral processes, procedures to execute in the event of failure (for example, node or site switching), etc.

Elaboration of Requirement D 4

We will fully comply with this requirement. The set of documentation aimed at the users of the system will include:

- General description of the system’s functionality
- User guides, for each part –application- of the system; this will include a set of interactive presentations in order to show the most common operations as performed on the real user inferface.