

## **E-vote 2011 System Architecture Administration system**

**V 1.1**

*Source Code, High Level Architecture Documentation and Common Criteria Documentation Copyright (C) 2010-2011 and ownership belongs to The Norwegian Ministry of Local Government and Regional Development and EDB ErgoGroup 67 AS (“Licensor”)*

*The Norwegian Ministry of Local Government and Regional Development has the right to use, modify (whether by itself or by the use of contractors) and copy the software for the sole purposes of performing Norwegian Public Sector Elections, including to install and run the code on the necessary number of locations centrally and in any number of counties and municipalities, and to allow access to the solution from anywhere in the world by persons who have the right to participate in Norwegian national or local elections. This also applies to elections to the Longyearbyen Community Council at Svalbard and any possible future public elections in Norway arranged by the Election Authorities.*

*EDB Ergo Group 67 AS (or whom it appoints) has the right, inside and outside of Norway to use, copy, modify and enhance the materials, as well as a right of licensing and transfer, internally and externally, either by itself or with the assistance of a third party, as part of the further development and customization of its own standard solutions or delivered together with its own standard solutions.*

*The Norwegian Ministry of Local Government and Regional Development and EDB ErgoGroup AS hereby grant to you (any third party) the right to copy, modify, inspect, compile, debug and run the software for the sole purpose of testing, reviewing or evaluating the code or the system solely for non-commercial purposes. Any other use of the source code (or parts of it) for any other purpose (including but not limited to any commercial purposes) by any third party is subject to EDB ErgoGroup 67 AS’ prior written approval.*

E-vote 2011	Version: 1.1
Main System Architecture	Date: 11.03.2011

## Change history

<b>Date</b>	<b>Version</b>	<b>Description</b>	<b>Author</b>
11.11.2010	0.1	Initial	
17.11.2010	0.2	Added tables to data model	
03.12.2010	0.3	Reorganized	
06.12.2010	0.4	Corrected some details that were not correct, elaborated some sections, created new sections and moved interface stuff onto "interface document".	
09.12.2010	1.0	First published version	
11.03.2011	1.1	Updates and minor corrections.	

E-vote 2011	Version: 1.1
Main System Architecture	Date: 11.03.2011

# Contents

## Contents

1. Architectural goal and scope	4
2. Overall process for the administration system	5
3. Essential use cases	6
3.1 UC 0.1 – Definition of roles	6
3.2 UC 0.2 – Configuration	6
3.3 UC 0.3 / UC 0.4 – Electoral Roll and exception process ER	6
3.4 UC 1.1/1.2 – List proposals	7
3.5 UC 3.1 – Registration of pVotes in ER	8
3.6 UC 3.2 – Manual registration of p-vote results	8
3.7 UC 3.3 – Electronic counting of pVotes	8
3.8 UC 3.4 – Counting of eVotes	8
3.9 UC 3.5 – Approval of p-votes and ballots	9
3.10 UC 4.1 Reporting to SSB	9
3.11 UC 4.2 – Settlement	9
3.12 UC 5.1 – Reporting (including election protocols)	10
4. Module architecture	11
4.1 List proposals	11
4.2 Election configuration	12
4.3 Electoral roll	13
4.4 Counting	14
5. Data model	15

E-vote 2011	Version: 1.1
Main System Architecture	Date: 11.03.2011

## 1. Architectural goal and scope

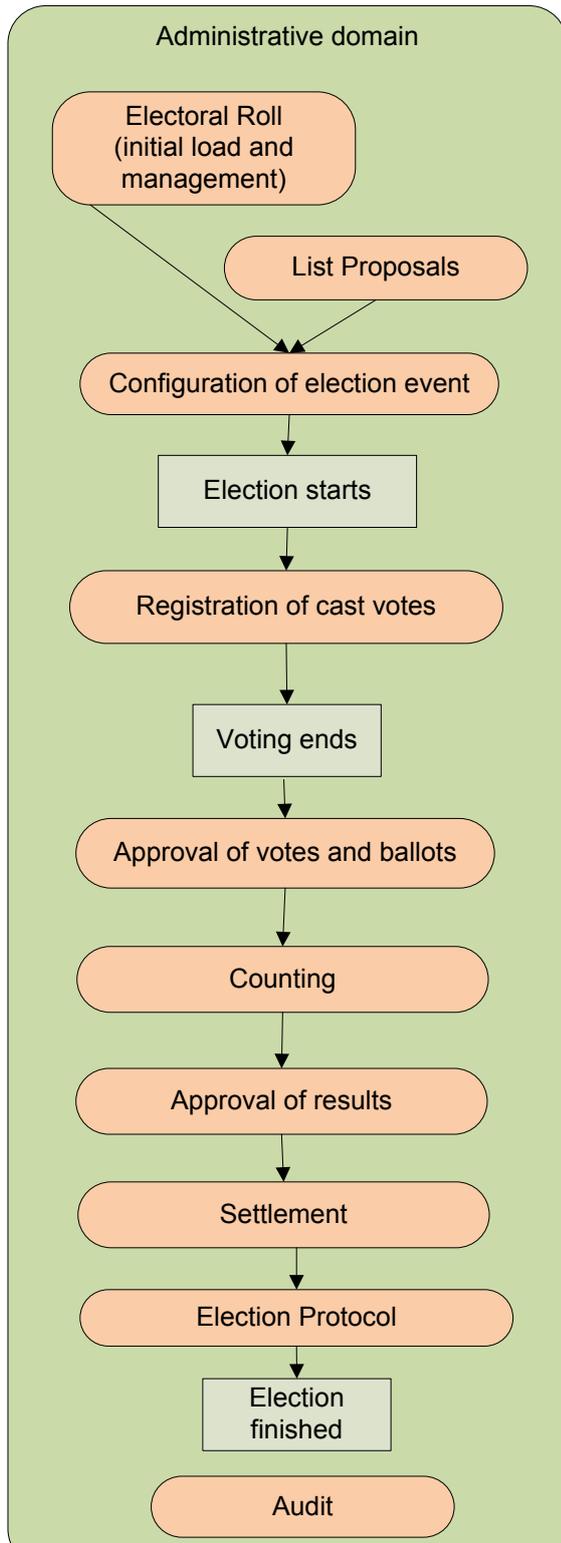
The administration system is built in a web-based open source platform intended for full public review. The system is built around a centralized solution allowing for remote access for the voting authorities. The system is accessible for de-localized subsystems through web services and through transfer of standardized election files (EML). The system is intended to operate without interruption and contain sufficient capacity and fail-over to complete an election without interruption.

The system facilitates all management of activities related to running an election. This includes preparing, performing and settling an election. In addition the system handles authorization and authentication, reporting and auditing. The administration system is also the main interface towards the electoral roll.

The system interfaces with the other subsystems that handle collection of votes: Electronic Voting and Electronic Counting of paper votes. The administration system also supports manual counting of votes and can be used as a standalone support system for purely manual elections. For selected elections, the administration system also facilitates list proposals for parties.

The administration system and the list proposal functionality are integrated with ID-porten. In addition, The equipment used by election officials has a certificate for SSL client authentication. This SSL client authentication is used to verify that the device (i.e. the election officials PC) is in a managed group of "secure" PCs. Computers used by party representatives for list proposals are external to the system domain. These computes are therefore not secured with SSL client authentication.

## 2. Overall process for the administration system



### Electoral Roll

The electoral roll decides who's allowed to vote at the different election events. This component also includes both internal and external updates of voters that are allowed to vote. There is one electoral roll for each election event defined in the system.

### List Proposals

All functionality prior to an approved list with candidates is handled by the component.

### Configuration of election and area structure

All configurations of the geographical structure and all election event parameters

### Registration of cast votes

Registration of cast votes including mark-offs in electoral roll.

### Approval of cast votes and ballots

Different types of votes submitted must be approved, and the ballots must be approved before counting.

### Counting

This is collection and registration of counting results from the eVoting platform, eCounting of pVotes and manually counted ballots.

### Approval of counting results

Final and preliminary counts are approved.

### Settlement

This component is where the official calculation of seats and distribution of mandates is done.

### Election Protocol

This component handles all functionality evolving the official protocols created throughout an election event.

### Audit

Contain functionality to audit all actions in the system after the election is closed.

E-vote 2011	Version: 1.1
Main System Architecture	Date: 11.03.2011

### 3. Essential use cases

The administrative system covers a large range of functions, the most important are:

#### 3.1 UC 0.1 – Definition of roles

The system provides a comprehensive and flexible way to control access to all functionality within the system. The key elements in this are:

1. **Securable objects.** All actions in the system are defines as securable object to which access is controlled. This also applies to screens and even single fields in a screen, when deemed necessary.
2. **Roles.** Roles can be created that has access to securable objects. A set of access rights is normally group into one role which corresponds to a user group of the system. New roles may also be created that contains other already existing roles.
3. **Users.** Users are created in the system and “tied” to a social security number, which is the one that identifies all users. ID-porten provide a Central Authentication Infrastructure, that authenticate the users and returns their social security number.

The above structure is named “Role Based Access Control”, RBAC.

When access to functionality is controlled this is actually done by combining two dimensions:

- The rights given by the RBAC
- The rights determined by where the user is located in the election hierarchy.

#### 3.2 UC 0.2 – Configuration

The use case contains flexible functionality for setting up an election.

The central election administrative system has functionality for defining election details such as election hierarchy, geographical hierarchy, parties and candidates, election dates and other relevant formalities. Management of parties and candidates are described more in detail in the section “Admin System - List Proposals” below.

When the election configuration is done, the configuration is frozen after an approval procedure. At this point the configuration is versioned and extracted from the configuration database and into EML-files. The EML-files are digitally signed to preserve their integrity while being transferred onto other system components. The EML-files are then made available for download by authorized users at the eVoting and pVoting sites.

If, for some reason, it is deemed necessary to change an approved configuration, the changed configuration will be exported into EML-files and digitally signed. The versioning of the configuration will be preserved in the EML-file.

The configuration is stored in three EML files which are digitally signed separately:

1. The Candidate list file containing parties and their candidates
2. The configuration of the complete election event (i.e. counties, municipalities and voting districts)
3. The option list file, which is only relevant for referendums.

#### 3.3 UC 0.3 / UC 0.4 – Electoral Roll and exception process ER

This is functionality for building and maintaining the electoral roll for an election event, such as voter eligibility, address, and other voting restrictions. In addition functionality to mark-off that voter has cast a vote in the election is supported.

The ER is initially created by importing a file from the national population register. For a defined period prior to the election event, daily updates from SKD (“Skattedirektoratet”) are automatically imported. In addition a management interface is provided for manual corrections. The eVote platform supports a function where voters that not find themselves in the electoral roll, can apply to be added. These request forms are forwarded to the

E-vote 2011	Version: 1.1
Main System Architecture	Date: 11.03.2011

Admin system, and can be processed by the right election officials.

Prior to an election that supports eVoting, the electoral roll is exported to the eVote platform for generation of personal return codes.

During the voting period the Admin system supports the following access to the electoral roll:

- A web-service is provided that enables the eVoting platform to see if a voter is included in the electoral roll, i.e. is allowed to vote, in the relevant contest(s).
- A poll-book application that enables the election officials at election day to see if a voter is in the electoral roll and to mark-off that votes have been casted.

During the counting period of the election, the Admin system supports export of all mark-offs of cast paper votes, so that the counting of electronic votes can be done according to the specified rules. After final counting of electronic votes, the Admin system also supports import of the eVote mark-offs in the electoral roll done in the eVoting platform.

### 3.4 UC 1.1/1.2 – List proposals

The use case contains functionality for submission, processing and approval of parties- and candidate lists. This module consists of a public and administrative interface. The public module is a web application available to any party or organization who wishes to be represented in an election. The applicant can submit a list of candidates and a signature list that endorses the party of organization according to laws and regulations. The public module is implemented in an external facing application environment for security reasons, and the administrative module is part of the administrative system interface.

The election officials use the administrative interface to manage the list proposals process and make the final approval of the lists. After the approval process is finished, the list proposals convert to election configuration for parties and candidates and are transferred to other system as part of the election configuration.

When a party wants to propose a candidate list, they follow this process:

1. Contact their election official (listeforslagsansvarlig)
2. The election official create an empty list proposal and a restricted user (listeforslagstiller) with access to that list
3. The representative of the party logs into the system (by means of DIFI CAI) and enters the role of party representative for the actual party. The party's proposal for candidates is entered into the system (either manually or by uploading an Excel sheet).
4. The rest of the procedure depends whether the party is already registered or not:
  - a. If the party is already registered, they add the name and information of at least two representatives that shall sign the candidate list. As the DIFI CAI currently does not support digital signatures, the candidate list with handwritten signatures must be forwarded to the election official by post. When receiving the paper-based candidate list with signatures, the "listeforslag ansvarlig" will manually verify the correctness of the signatures, and that the information entered in the system is according to the paper-based original.
  - b. If the party is not a registered one, the group or party must provide a list of signatures from people supporting the creation of the new party. The system supports download of a paper-based template that shall be used for providing the signatures. This paper-template is tailored for being ICR processed so that it is likely that the identity of the signatures can be automatically detected by the system. When the new party/group has the required number of signatures, they send the lists of signatures to the election official for scanning at the mail reception center. The operator at the mail reception center may upload the scanned TIFF-image to the system (it must be scanned according to the rules set forth in the procedures). When the system receives a scanned signature list, it will ICR the lists and tries to identify the persons on the list from the electoral roll. Some will be identified and automatically added to the system, and some must be manually added by the "listeforslag ansvarlig".

The final approved lists of candidates will be digitally signed by the Admin system and made available for

E-vote 2011	Version: 1.1
Main System Architecture	Date: 11.03.2011

download by the eVoting and eCounting of pVotes systems as part of the election configuration.

### 3.5 UC 3.1 – Registration of pVotes in electoral roll

As described in 1.3.3 the Admin system provides a poll book application that can be used when voters are submitting pVotes. The main functional elements of this poll book application:

1. **Receiving votes in advance.** When receiving votes in advance, each casting of a vote shall be registered and stored in the system. A “preliminary” mark-off in the electoral roll is created in the system. When the advanced vote later on is controlled, the casting may be rejected or accepted. In the latter case a final mark-offs is set in the electoral roll.
2. **Receiving votes in special cover.** When receiving votes that shall be put in special cover, or so called “foreign votes”, each casting of such votes shall be registered and stored in the system. When these votes later on are controlled, the mark off in the electoral roll is done.
3. **Registering of casted votes.** When voters want to cast their vote on election day, the electoral roll may be searched to find the voter. If the voter is allowed to cast a vote, the mark off in the electoral roll is done.
4. **Printing polling cards.** The poll book application support print out of duplicate polling cards. (Please note that these polling cards lack the return codes used for eVoting).
- 5.

### 3.6 UC 3.2 – Manual registration of p-vote results

This interface supports manually registering and comparing tallied results, such as number of voters that voted for a party in a districted. The administration system has a web application for entering manual counts. These counts can be entered several times, and individually compared with each other.

The interface also support manual registering of the corrections applied to corrected ballots.

An approval procedure is also supported, which enables the counting results to be part of the Settlement process.

### 3.7 UC 3.3 – Electronic counting of pVotes

This use case was originally to be provided by the scanning system. However, during the specification phase, a lot of the functionality was moved onto the Admin system. The scanning system provides the scanning of the ballots, and OCR interprets the content of the ballots. The scanning system also does the counting for ballots that are not corrected, but corrected ballots are transferred onto the Admin system for further processing.

Thus, the Admin system provides the following functionality for this use case:

- Receive the OCR interpreted results from the scanning system and store them into the counting tables in the database.
- Process all the corrections on corrected ballots.
- Presents the counting results for the election officials and enable them to compare results.
- Enable the election official to correct the results if necessary.
- Provide an approval procedure for the counting results which enables the counting results to be part of the Settlement process.

### 3.8 UC 3.4 – Counting of eVotes

The main process on counting of eVotes is done in the eVoting platform, and is not described here. However, the following functionality is provided in the Admin system to support the counting of eVotes process:

- An initial export of the electoral roll with mark-offs for casted pVotes is done to enable the eVoting platform to do the initial counting of eVotes.
- The counting results from eVotes are imported and stored in the counting tables in the database.
- When each counting authority for pVotes has finished all electoral roll mark-offs, they may ask for the final counting results for eVotes. The electoral roll with mark-offs are exported to the eVoting platform

E-vote 2011	Version: 1.1
Main System Architecture	Date: 11.03.2011

for the relevant counting authority.

- The corresponding final counting results are imported into the Admin system. Note that also in this case the corrections done on corrected ballots are not tallied in the eVoting platform, but the corrections are transferred individually onto the Admin system.
- The Admin system processes all the corrections on corrected ballots.
- Presents the counting results for the election officials.
- Provide an approval procedure for the counting results which enables the counting results to be part of the Settlement process.

### **3.9 UC 3.5 – Approval of p-votes and ballots**

The Admin system provides functionality to approve or reject casting of pVotes done in advance or pVotes that has been put into special covers. A one-by-one procedure is supported as well as a procedure for so called “negative proofing”. In the latter case, some castings may be individually rejected, and the rest of the castings within a batch will be approved.

The Admin system also supports handling of questionable ballots. They may be rejected or approved. If rejected each ballot must be registered with a rejection reason.

A change order that has been implemented implies that the images of the questionable ballots are transferred from the scanning stations onto the Admin system for viewing and processing here.

### **3.10 UC 4.1 Reporting to SSB**

The Admin system provides SSB with information through a web-service interface. Some of the information is regarded as general data and is provided prior to the election. However, the main information reported is during election day.

The party codes and the names of the parties are manually maintained in the system. SSB is the master for these data.

The following information is reported:

- Prior to the election:
  - Contact information for all election authorities.
  - All parties that is eligible for each contest and their candidates
- During election day
  - Preliminary counting results for the municipality as well as the county election.
  - Final counting results for the municipality as well as the county election.
- After the election (some days after the election closes)
  - Corrected counting results

Please note that the reporting done several months after the election, that among others contains the distribution of seats and information on candidates, requires a lot of information that is not available in the system. Thus, this reporting to SSB will not be done automatically, but has to be done by manually using the SSB web-interface for this purpose.

### **3.11 UC 4.2 – Settlement**

This aim of the settlement process is to distribute seats and mandates based on the election results. The functionality in the Admin system that supports the settlement process is:

- Show the counting status for each contest (includes the counting results for each counting district), so that it can be determined if settlement can be done or not.

E-vote 2011	Version: 1.1
Main System Architecture	Date: 11.03.2011

- If the prerequisites for settlement are fulfilled, find the approved counting results for each district that belongs to the contest for all relevant vote types and prepare the counting results for settlement. Preparation includes preprocessing of all corrected ballots according to the settlement algorithm.
- Calculate the distribution of seats according to the correct algorithm and store all necessary data that is required for the settlement protocols.
- Calculate the candidates that are selected according to the correct algorithm, and store all the data that is required for the settlement protocol.
- Provide feedback to the election administrator that shows the result of the settlement.

### 3.12 UC 5.1 – Reporting (including election protocols)

The reporting use case consists of a general reporting module for creating and running election related reports - In particular, officially required reports such as election protocols as well as electoral roll reports. In addition several reports may be created to support different statistical purposes.

The reporting module is implemented based on standard open source solutions. It includes JasperReports which is a generic reporting engine. The tool iReport is used to design and implement the reports. In addition, there is an engine to scheduled and execute reports. There are two different types of reports that can be generated:

- So called application reports, which is tailor made to suit a specific operation or screen. (In these cases the reporting tool is used to generate a PDF that displays the result of the previous action or the result of an operation. The PDF may be viewed, saved or printed. These types of reports are highly dependent on the actual application context and are thus managed as part of the source control itself. Example of an application report is instant printing of duplicate polling cards.
- General and statistical reports. These are managed and run by a general asynchronous reporting solution. A specific work flow is created as part of this general solution to support approval of new reporting templates. Examples of these types of reports are electoral roll reports and election protocols.

## 4. Module architecture

### 4.1 List proposals

The 'List proposal collection' and 'List proposal administration' modules share a common code base deployed at different servers. They also share a common database, but the public collection part connects to the database with a restricted user which only has access to the relevant tables.

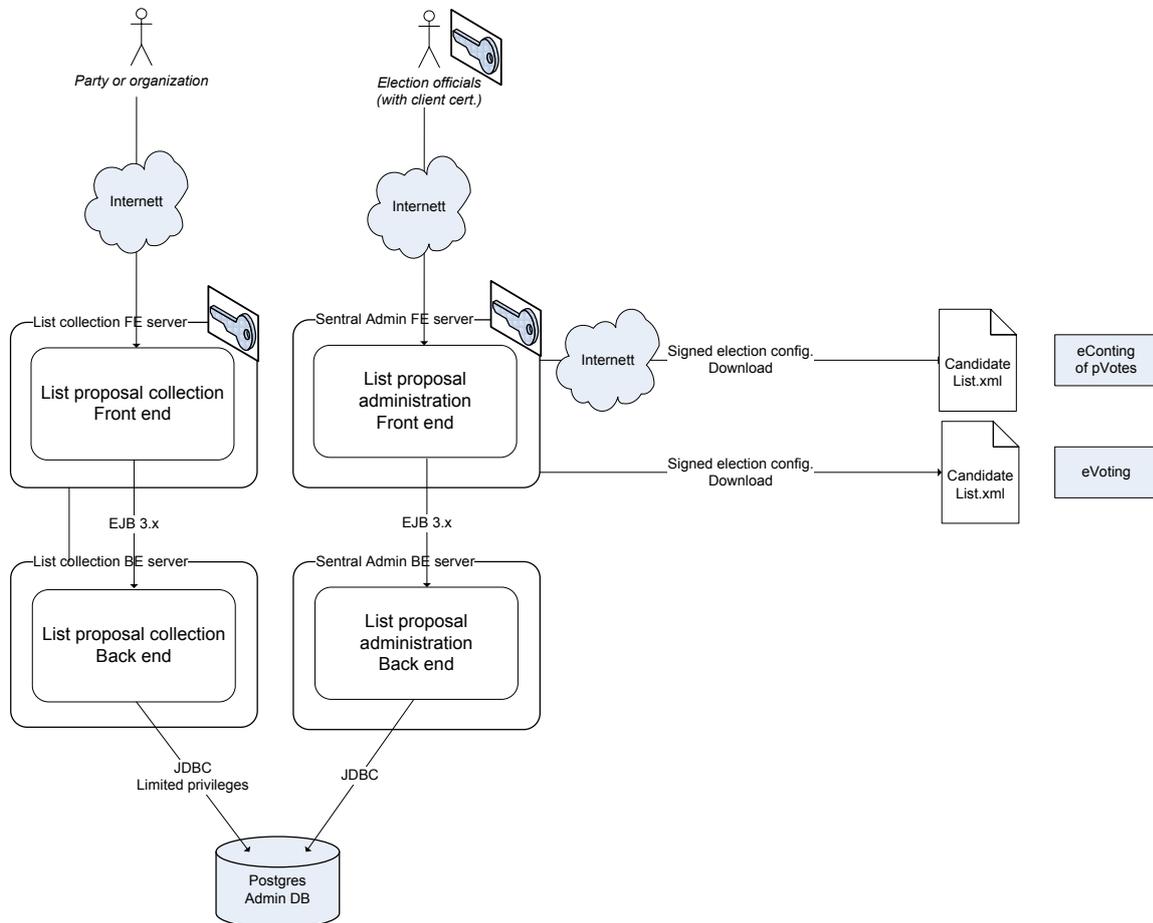


Figure 1: DFD List Proposals high level view 3 tier architecture

## 4.2 Election configuration

The election configuration module uses a web interface to define elections. Elections that are completely configured are exported as signed EML files (event, candidate list, option).

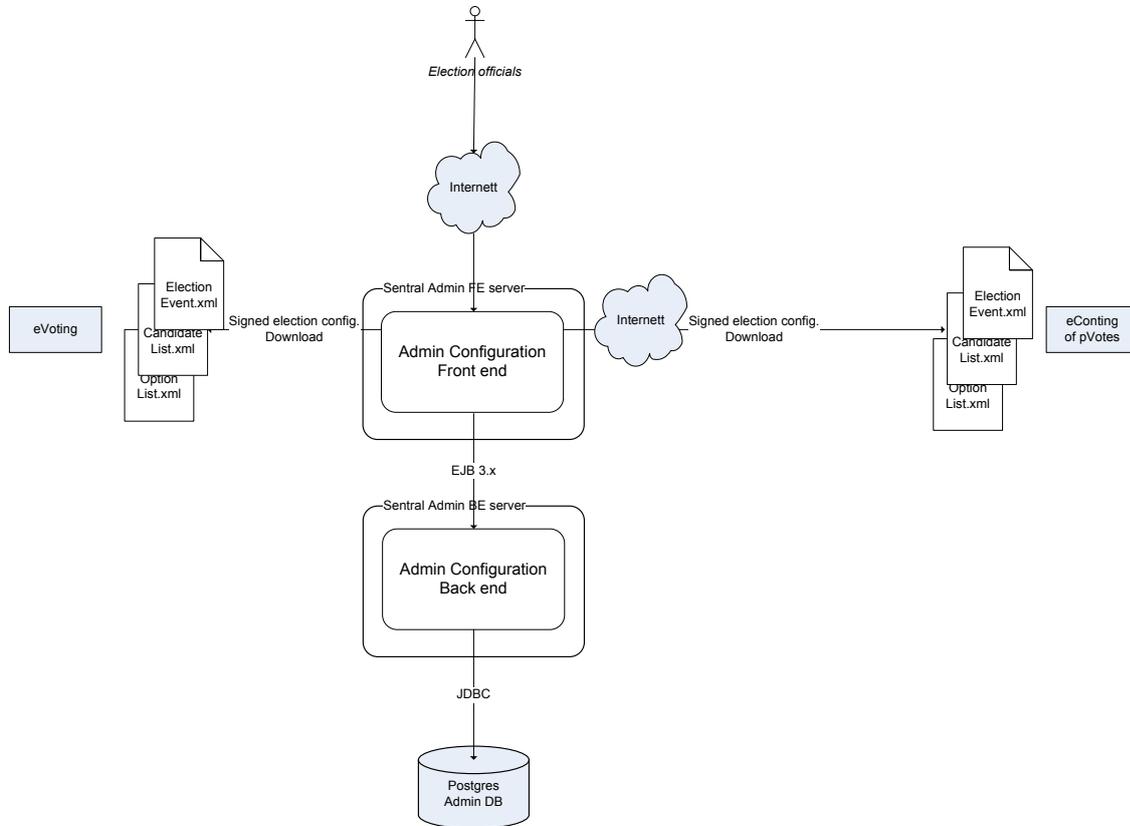


Figure 2: DFD Admin Configuration high level view

### 4.3 Electoral roll

The electoral roll is used in many settings

- Manage electoral roll (add/change/removed voters)
- By eVote to check voter eligibility
- By eVote to eliminate paper votes from eVoting
- During paper voting (advance and election day) to verify voter eligibility and mark-off.

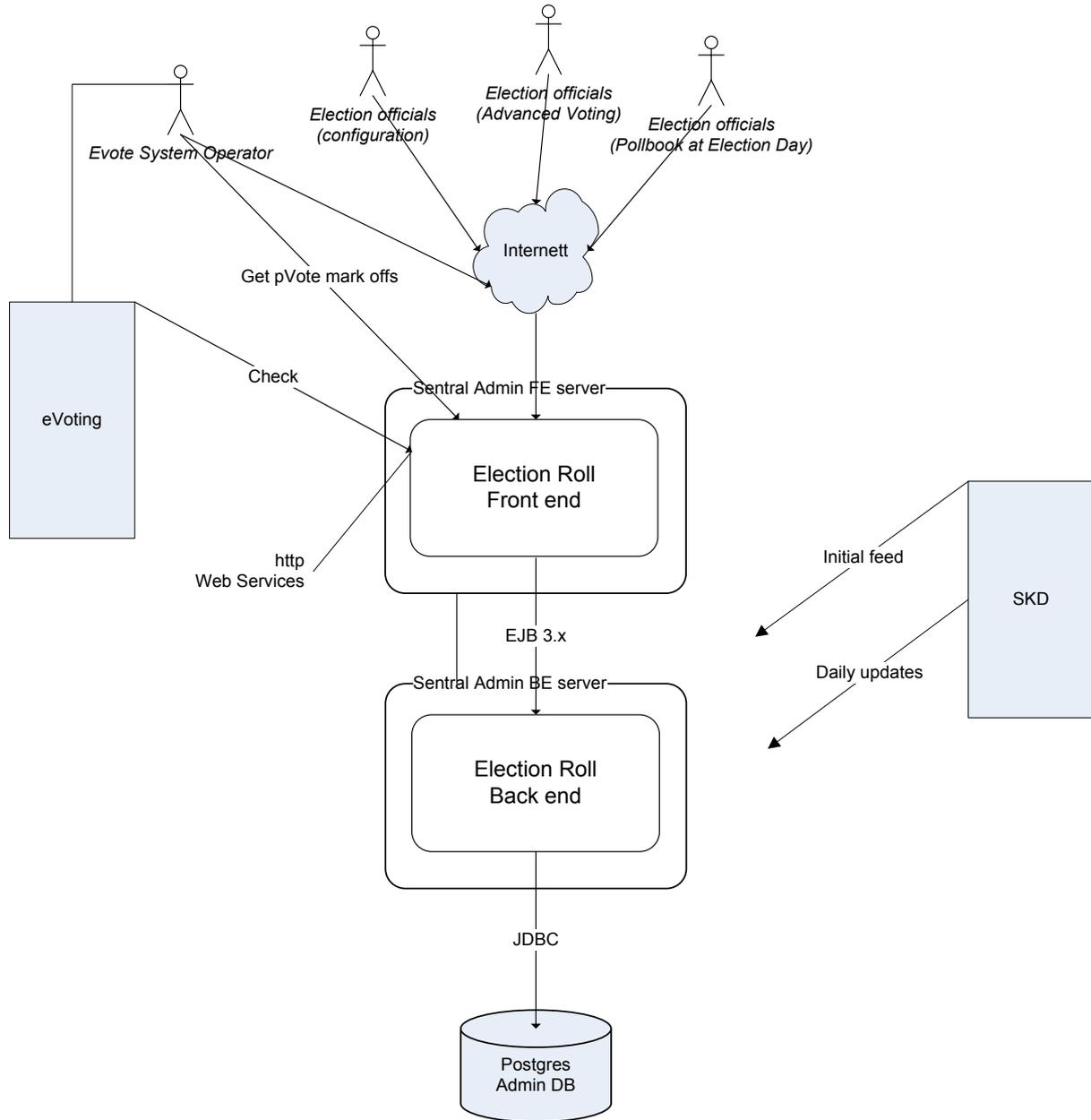


Figure 3: DFD Electoral Roll high level view

## 4.4 Counting

The Admin system collects counting results to be able to perform consolidation of counts from eVoting, pVoting and manually entered counts. How the counts are performed is dependent on the election configured. The counts also depend on the type of vote. Counts from external systems (eVoting and eCounting of pVotes) are received as signed EML files. Please note that corrections on the corrected ballots are transferred individually to the Admin system.

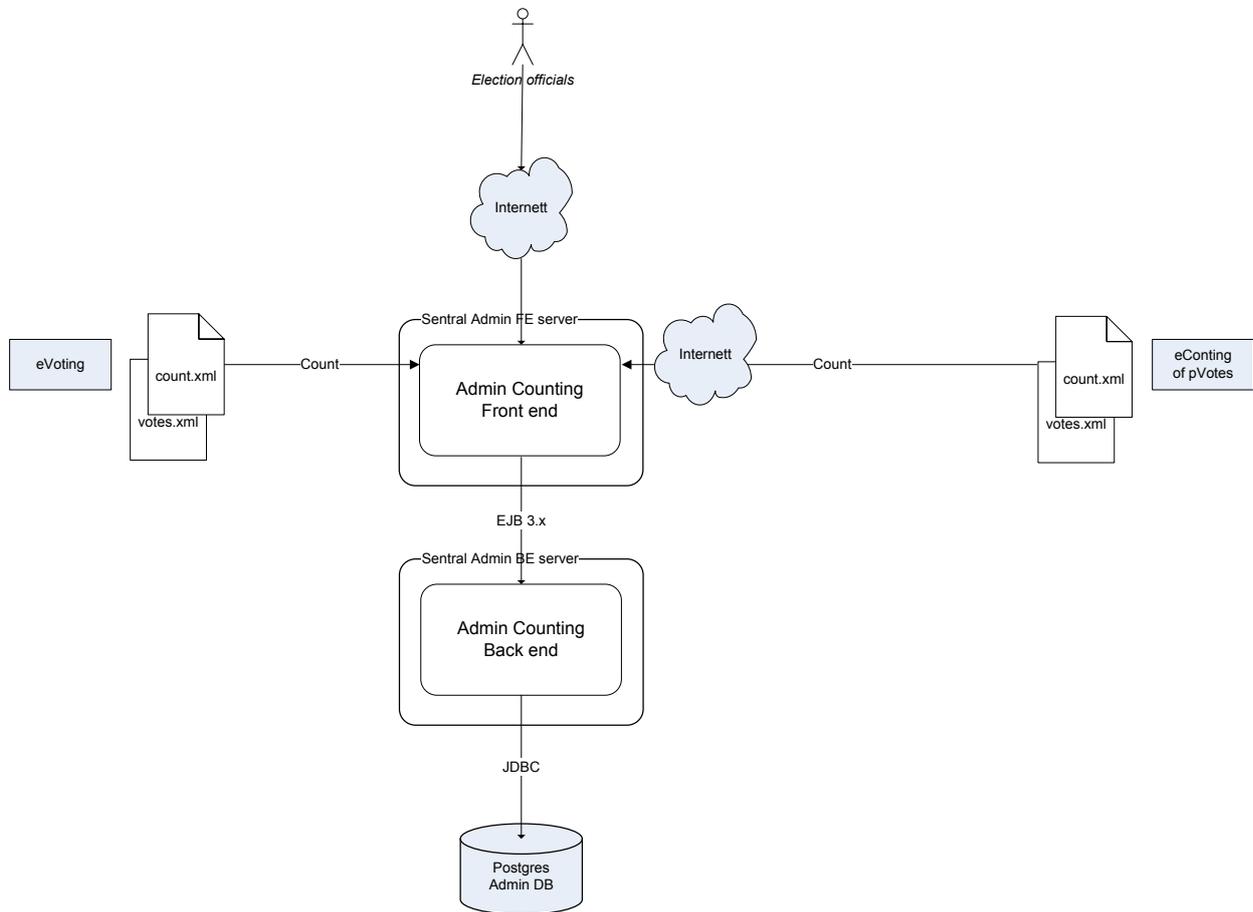


Figure 4: DFD Admin Counting high level view

E-vote 2011	Version: 1.1
Main System Architecture	Date: 11.03.2011

## 5. Data model

Table name	Description
aarsakskode	Codes from Skattedirektoratet / Det sentrale folkeregisteret
access	RBAC: Hierarchy of access to securable objects
access_type	RBAC: Categories of securable objects
affiliation	Affiliations on candidate lists / ballots
area_level	Description of levels in mv_area
ballot	Candidate lists / ballots (an approved candidate list represents a ballot)
ballot_count	Count of ballots. Ballots may be accumulated (possible for unmodified ballots) or counted individually (necessary for modified ballots)
ballot_rejected_count	Count of rejected ballots. Rejected ballots may be accumulated or counted individually, in the same way as accepted ballots
ballot_rejection	Reasons for rejection of ballots
ballot_status	Status for ballot / candidate list
ballot_text	Text linked to individual ballots
batch	Information on the latest batch
batch_number	The last assigned batch number in the election event
batch_status	Status for batch jobs
binary_data	Binary storage, with back reference to table/column/row containing binary_data pk
binary_data_number	The last assigned binary data number in the election event
borough	Boroughs
candidate	Candidates on candidate lists
candidate_audit	
candidate_count	Count of candidate votes
configuration	Configuration parameters
contest	Contests per election, e.g. "Oslo" or "Sør-Norge valgkrets"
contest_area	One or more areas contained in the contest (Norwegian "valgkrets")
contest_rel_area	
contest_report	Represents contest protocols with optional counts. The reporting is performed for a contest within the authority of the reporting unit
contest_report_text	Text for reporting unit contest protocols
contest_seat	Seats assigned by the modified Sainte Lagüe method
contest_status	Status for contest
contest_text	Text linked to individual contests
contest_voting	View for calculation of votings/markoffs per contest
count_qualifier	Qualifiers for separation of counts, e.g. preliminary and final count
country	Countries
county	Counties
election	Election within an election group
election_all_area	
election_day	Election event day(s)
election_event	Parents all data for one or more elections run within the same time frame
election_event_status	Status for election event

E-vote 2011	Version: 1.1
Main System Architecture	Date: 11.03.2011

election_event_text	Text linked to individual election event
election_group	Grouping of elections within an election event, facilitating RBAC access to a subset of all elections via a single role
election_level	Description of levels in mv_election
election_party	
election_rel_area	
election_status	Status for election
election_text	Text linked to individual elections
election_type	Broad categorisation of election, e.g. Referendum
eligibility	View for calculation of eligible contests per voter
foreign_early_voting	View for report on foreign early voting
generic_communication	Incoming/outgoing generic communication from/to voters, e.g. applications for inclusion in the Electoral Roll / feedback on the applications
incoming_status	Status for incoming generic communication
key_domain	Signing domain for key in key ring
locale	Locales (languages) for text
locale_text	Translated text
marital_status	Marital status codes from <a href="http://ssb.no/stabas/">http://ssb.no/stabas/</a>
municipality	Municipalities
municipality_status	Status for municipality configuration
mv_area	Materialized view containing all nodes in the hierarchy formed by the tables country, county, municipality, borough, polling_district, and polling_place, facilitating RBAC access to any level in the hierarchy via a single field/pointer
mv_election	Materialized view containing all nodes in the hierarchy formed by the tables election_group, election, and contest, facilitating RBAC access to any level in the hierarchy via a single field/pointer
mv_election_reporting_units	Reporting unit types per node in the election hierarchy
opening_hours	Opening hours for polling places
operator	RBAC: Application user
operator_role	RBAC: Role granted to an operator, with additional constraints specified by links into the election and area hierarchies
operator_task	
party	Standard party list, including historical parties. Codes from Statistisk sentralbyrå
party_category	Categories of parties. Codes from Statistisk sentralbyrå
polling_district	Polling districts
polling_place	Polling places
polling_place_voting	
polling_station	Polling stations ("roder")
postal_code	Postal codes from <a href="http://epab.posten.no/Norsk/Nedlasting/NedlastingMeny.htm">http://epab.posten.no/Norsk/Nedlasting/NedlastingMeny.htm</a>
postal_code_category	Postal code categories from <a href="http://epab.posten.no/Norsk/Nedlasting/NedlastingMeny.htm">http://epab.posten.no/Norsk/Nedlasting/NedlastingMeny.htm</a>
proposer	Proposers (signers) of candidate lists
proposer_list	Scanned candidate lists
proposer_list_number	The last assigned proposer_list number, per ballot
proposer_list_status	Status for (scanned) list of proposers
proposer_role	Proposer (signer) roles

E-vote 2011	Version: 1.1
Main System Architecture	Date: 11.03.2011

referendum_option	Referendum options on options lists
referendum_option_count	Count of referendum option votes
report_count_category	Specifies the vote count categories which should be counted, and whether the preliminary count should be performed at the municipality level or at a lower level
reporting_unit	Election board, with area(s) of authority specified by links into the election and area hierarchies. E.g. an election board for "Kommunestyre- og fylkestingsvalg" at the municipality level may be linked to the election group representing both elections, and to the municipality. In this case, counts for both elections may be performed by the same election board
reporting_unit_type	Types of reporting units, used as templates to create reporting units
responsibility	Possible responsibilities for officers in reporting units
responsible_officer	Officers in reporting units
role	RBAC: Role
role_access	RBAC: Access to securable objects per role
role_access_all	
role_include	RBAC: Roles included per role
role_include_all	
signing_key	
spes_reg_type	Codes from Skattedirektoratet / Det sentrale folkeregisteret
standard_report	Standard report templates
statuskode	Codes from Skattedirektoratet / Det sentrale folkeregisteret
task	Task list
task_number	The last assigned task number in the election event
text_id	Text id for translation
translation	
vote_category	Categories for separation of votes, e.g. normal votes and write-ins
vote_count	Count of votes per polling district per vote count category, with optional underlying count of ballots. The polling district is within the authority of the reporting unit
vote_count_category	Categories for separation of voting counts, e.g. ordinary early votes and ordinary election day votes
vote_count_status	Status for vote_count
voter	Voter information
voter_audit	
voter_import_batch	Information on the latest voter import batch
voter_number	The last assigned voter number, per municipality
voting	Information on all votings per voter. Only one voting may be accepted within each election_group
voting_category	Categories for separation of cast votes, e.g. ordinary early votes and early votes from other municipalities
voting_number	The last assigned voting envelope number for the municipality, per election_group and voting category
voting_rejection	Reasons for rejection of votings