ErgoGroup

E-vote 2011
Security Architecture
Election Administration System
V 1.2

Change history

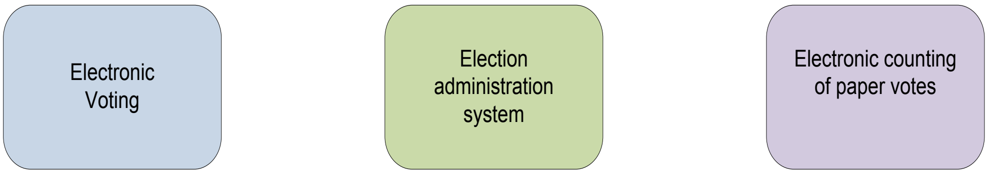
Date	Version	Description	Author
18.11.2010	0.1	Initial	
07.12.2010	0.5	First draft version for review	
13.01.2011	1.0	First complete version	
20.01.2011	1.1	Corrected description of election and area context. Updated the list of user roles.	
11.03.2011	1.2	Updated description of user roles. Some other minor updates.	

Contents

1. Introduction

This document presents the security architecture of the Election Administration System (Admin) which is part of the E-vote 2011 project. By *security architecture*, we mean the conceptual, logical, and physical design which is used to protect the systems and the information they protect from compromise.

The E-Vote 2011solution consists of three different but interacting systems:


Electronic Voting
Election administration system
Electronic counting of paper votes

Each system can operate in its own environment, but require interaction to satisfy common requirements such as configuration, authentication/authorization, logging and reporting. The overall security architecture is described in the document "Security Architecture - General Overview".

2. Conceptual design

Conceptual Design is the view of the security solution *from the user point of view*. It seeks to define the characteristics of information access and authentication that are part of the user experience.

2.1 Security domains and interfaces

The Election Administration domain includes three separate domains as illustrated below.

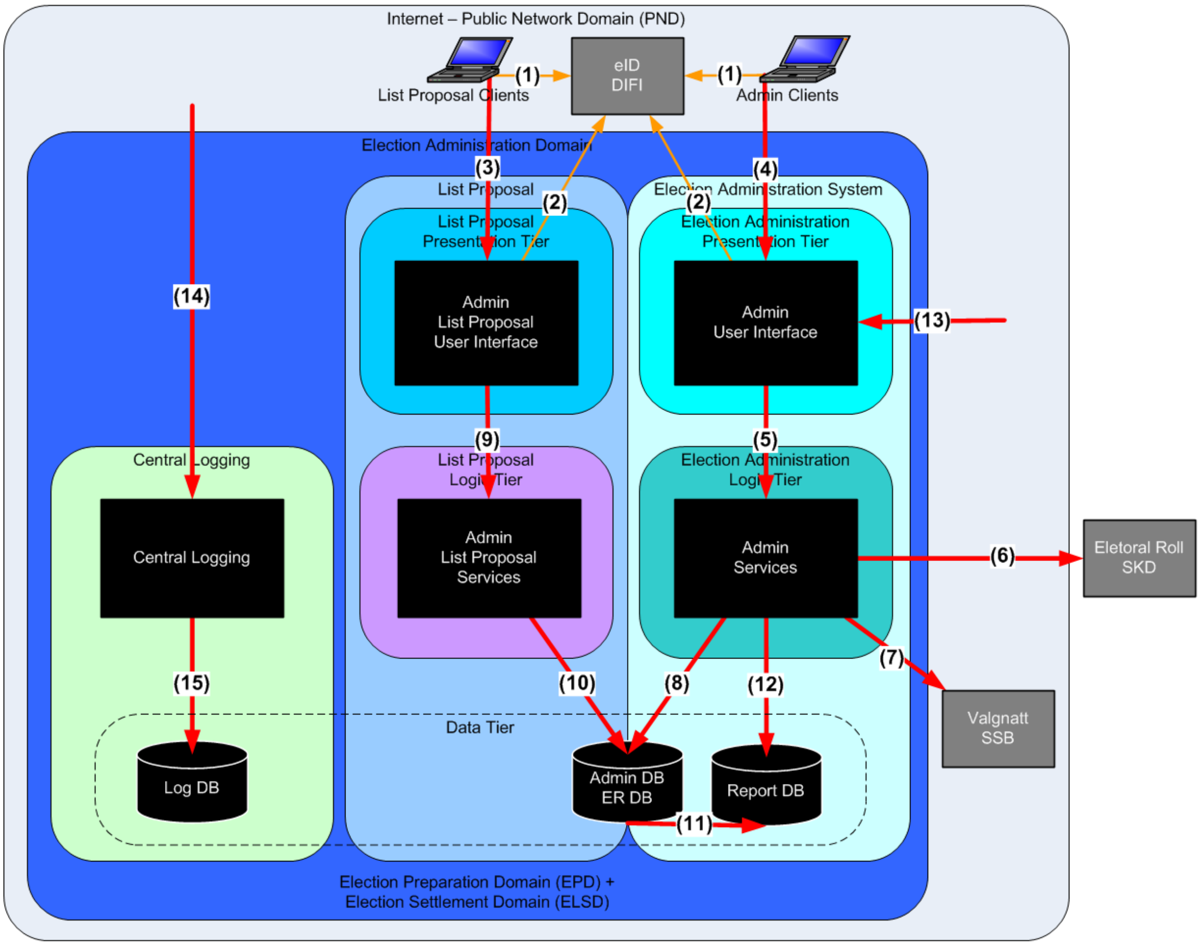


Figure 1: Election Administration Domain

- **Election Administration System** (Admin) is the main part split in a three tier architecture (Admin Presentation, Logic, and Data tier)
- **The List Proposal** collect list proposals from the public and share database with Admin
- **Central Logging** collect audit - and system logs from all systems in the E-Vote 2011solution (including rest of Admin)

The interfaces to the system and between the subsystems are (numbered in the figure above):

1. **User authentication**. Web interface over HTTPS. Possible redirected from election service.
2. **Verify user authentication**. Web service over mutual authenticated connection.
3. **Public web interface for list proposals** over HTTPS.
4. **Web interface for election administration** over mutual authenticated HTTPS connection.
5. **Service calls in Admin** (EJB calls)
6. **Get initially load and daily updates of the electoral roll** from Norwegian Revenue Service (In Norwegian: Skattedirektoratet - SKD).
7. **Send preliminary and final reports of election results** to Central Bureau of Statistics (In Norwegian: Statistisk Sentralbyrå - SSB).
8. **Database connection from Admin**
9. **Service calls in list proposal** (EJB calls)
10. **Database connection from list proposals**
11. **Database replication for reporting**
12. Read only **database connection to report database**
13. **Interfaces for the eVoting Collection - and Paper Voting Domain**. The following interfaces is provided:
    a. Deliver election configuration in EML (XML) format from a web service or web page over a mutual authenticated HTTPS connection. The configuration EML files are digitally signed by the Election Administration system.
    b. Receive ballot count results in EML (XML) format with a web service or web page over a mutual authenticated HTTPS connection. The counting EML files are digitally signed.
    c. Check if a voter is eligible to cast a vote in an election with a web service over a mutual authenticated HTTPS connection.
    d. Deliver a record based file for the whole electoral roll with mark-offs from a web service over a mutual authenticated HTTPS connection. The electoral roll copy is used by eVoting to check the final electronic ballots and to filter out all voters which have cast a paper vote from the final electronic count.
14. **Collection of audit - and system logs** from all election systems through rsyslog service (UDP/514).The audit log data is immutabilized before it is sent.
15. **Connection to central log database**

2.2 User profile and description

There are three groups of users in the Admin system: Election administrators/officials, Political party officials, and Operators in the pVoting and eVoting environments.

The election officials (centrally and at the municipalities) access the system to perform various tasks: election configuration, update electoral roll, run reports, approve list proposals from parties, count elections and perform settlement of elections. Election officials at the poll stations use the system to verify voter eligibility in the electoral roll and to mark-off votes in the electoral roll. The political party representatives use the system in advance of the elections to submit party and candidate proposals. The operators in the pVoting and eVoting environments access the Admin system to get the election configuration and report the counting results. The eVoting system also queries the election roll and downloads export of it through web services.

Selected users have access to add/change roles and users for the Admin system. Each user role has a set of “accesses”. These accesses defines which “securable objects” the user has permission to access. Any set of accesses could be used to define a new user role.

Users and roles (and most of the other data structures) are organised per election event in the system database. One root election event (“Admin-event”) and one system administrator role (“Systemansvarlig”) is predefined in the system. This system administration role is the only one with permissions to create/delete election events, manage global text resources, and manage cryptographic keys. On the first start up of the system, one system administrator user is defined during the bootstrap process. This first user is assigned the system administration role in the root election event.

The root election event has a predefined a set of the most common user roles needed for running an election. When the system administrator creates a new election event, she could chose to copy all existing user roles from another election event (e.g. the root election event) to the new event.

A new election event manager role ("Valghendelseansvarlig") is created automatically in every new election event. The system manager who creates a new election event is assigned the election event manager role in that event.

Below is a list of a few of the default user roles which in the root election event medio February 2011.

User role	General use
Systemansvarlig	System manager in the root election event. Predefined in system database. First user created during system bootstrap process. Only role with permission to create new election events.
Valghendelseansvarlig	Election event manager. Created automatically when a new event is created. The system administrator who creates the event is the first user assigned to this role.
Lokal konfigurasjonsansvarlig	Local configuration manager. Configuration of area hierarchy within hers municipality (commune), local election board ("Stemmestyre"), and counting methods.
Lokal konfigurasjonsgodkjenner	Local configuration approver.
Listeforslagsansvarlig	List proposals responsible in municipality (commune). Create users for external parties/groups who wish to submit list proposals. Create political parties and manage the list proposals.
Listeforslagsstiller	List proposal submitter. Existing or new political party managing their proposed list through the public part of the system.
eTellingsansvarlig	E-voting counting manager. Export electoral roll information to the e-voting system for return code generation and cleansing. Verify import of counting results from the e-voting system.
Manntallsfører	Electoral roll responsible in municipality (commune).
Forhåndsstemmemottaker	Receiver of advance votes. Add voter to the person register (not the electoral roll).
Prøveansvarlig forhåndsstemmegivning	Advance votes approver. Approves or disapproves advance and late received votes. Mark voter in electoral roll for approved votes.
Stemmemottaker valgting	Receiver of votes at polling sites. Mark voter in electoral roll.
Stemmemottaker valgting særskilt	Receiver of provisional ballots ("Særskilt omslag") at polling sites. Add voter to the person register (not the electoral roll).
Korreksjonsansvarlig	Electoral roll correction manager. Could remove the marking of voters in the electoral roll.
Prøveansvarlig valgting særskilt stemmegivning	Provisional ballots approver. Approves or disapproves Provisional ballots. Mark voter in electoral roll for approved votes.
Opptellingsansvarlig forhåndsstemmer / Opptellingsansvarlig valgtingstemmer - Stemmestyre, - Valgstyre, - Fylkesvalgstyre / Opptellingsansvarlig stemmegivninger valgting	Counting responsible on different levels.
Godkjenner av foreslått forkastede stemmesedler / Resultatgodkjenner valgtingstemmer - Stemmestyre, - Valgstyre, - Fylkes	Approval responsible on different levels.
Skanneroperatør / - administrator	Users in eCounting of pVotes domain.

2.3 Authentication

Authentication in Admin with MinID is described in the document “Security Architecture - General Overview”. This is in principle a single-sign-on system for many government services, but the configuration does not allow single-sign on towards the election systems.

Client used for accessing the Admin web application (except public list proposal) is further authenticated based on the calling IP address and provided SSL client certificate. Before use of the Admin client, the client certificate must be installed by municipal system administrator. The Admin client PC should be configured in a secure way, be well patched, and have an updated anti-virus system running.

Inactivity session timeout in Admin is configurable. It will be set to be 30 min.

Authentication in the Admin web application could be change to require security level 4 provided by the new national electronic ID service (“ID-porten”). Users will then use their Buypass smartcard in the authentication process.

2.4 Access control

Only predefined users could log on to the Admin web application. Users and user roles are defined in the Admin system. Users with special roles (e.g. “Systemansvarlig” and “Valghendelseansvarlig”) have permission to create, edit, and delete other roles and users in the Admin web application. Users and user roles are created for each election event.

Access to page components, entities, and tasks in the Admin application is secured by a role based access control (RBAC) system. A user might have access to several user roles, but she might only possess one role at the time. The RBAC system supports exclusive roles. If a user is assigned to such a exclusive roles, she could not be assigned to any other roles.

All functionality in the application is implemented as methods in EJBs. These EJBs implements the service layer between the frontend and backend of the system. The system configuration of the backend (`ejb-jar.xml`) binds an interceptor (`SecurityInterceptor`) to every EJB in this service layer. The interceptor invokes the RBAC and the system configuration guarantees that access control is performed on every call through the service layer.

The system defines a set of “securable objects”. For a user, the securable objects are the different permissions she could have. A user role is defined as a list of these securable objects. For methods in the service layer, the security objects are the different permissions they could require. The securable objects are basically strings which must be in place in the database. The securable objects beginning with e in the path is for securing service layer. Objects beginning with w are for GUI. The securable objects are hierarchical. Let’s say you have the following objects:

- e.results.counting.count
- e.results.counting.viewCount
- e.results.counting.reCount

If one were to have access to the securable object e.results.counting, it would also give access to the descending objects e.results.counting.count, e.results.counting.viewCount, e.results.counting.reCount.

All methods in the service layer require the current user to have at least one of the specified permissions. The required permissions are specified by annotation (`@SecObj`) of the whole service class or the individual methods. The implementation of the interceptor ensure that the developer must explicitly state required permissions for all functionality in the service layer - invocation of a method without annotation will result in a security exception.

The system defines an election hierarchy (Election group / Election / Contest) and an area hierarchy (Country / County / Municipality / Borough / Polling district / Polling place). When a user role is assigned to a user, the assignment must be connected to a specific place in these hierarchies. In addition to RBAC based on the securable objects, the access control could also be performed based on the election context and area context of the role the user possess. Individual parameters could be annotated (`@SecureEntity`) to enforce checking of the user’s access to a specific entity. The method `hasAccess()` will take into account the current election context and area context of that user. And the methods `hasMinAreaLevel()` and `hasMinElectionLevel()` could be used to check the context levels of a given user. There is no mandatory context based access control - the developers have to invoke these controls as the business logic demand.

2.5 Key Security Scenarios

Here is a description of some key security scenarios.

2.5.1 Key management

To support secure communication and signing, there is a separate key management service that is airgapped in the eVoting environment. This is used to generate the keys and certificates necessary for secure communication and to ensure integrity during exchange of data. The key management is documented in [Key legend] and [Bootstrap process].

The Admin system needs the following cryptographical keys:

1. SSL server certificate for https connections from the Admin clients
2. SSL client certificate for direct connection to ID-porten
3. Private key for immutable log signing
4. Private key for digitally signing of exported data files (e.g. election configuration, token, and electoral roll exports)

The SSL server certificate for the Admin system is generated and stored on the load balancer (i.e. BigIP) which terminates all external SSL connections to the system.

The SSL client certificate and the log signing key are common for all election events in the Admin system. These keys are installed during the first start up of the system.

The application private key for signing of exported data files is specific for each election event. These keys are installed by the election event manager through functionality in the Admin web application.

2.5.2 Approval and signing of election configuration

After the election configuration is done, an election official approves it. When the election configuration is approved, the Admin system generates a configuration package (zip) consisting of:

- Election event EML file (`ElectionEvent-<eventID>.xml`)
- Candidate list EML file (`CandidateList-<eventID>.xml`)
- One signature file for each of the EML files (`ElectionEvent-<eventID>.pem` and `CandidateList-<eventID>.pem`

The configuration files are digitally signed by the system with the installed Admin application private key.

2.5.3 Generating, signing, and encrypting of security tokens

Operators in the pVoting and eVoting domain must log on to Admin and download a security token file before they can use their application in these domains. The token file contains the user-id, assigned role, and expiration time. The token is signed by the Admin application private key. When users chose to download their token, they have to pick a personal password. The Admin system uses the given password to generate a symmetric key which is used to encrypt the token file.

When the token is presented to other systems, the user has to input her password. The other systems use the same algorithm to generate a symmetric key which is used to decrypt the file.

2.5.4 Verification of received counting results

The Admin system receives counting results in EML format from E-voting and P-voting. The files from E-voting are signed by the E-voting Counting module application key. The files from P-voting are signed by the personal key of the sending operator. Before the data is accepted in the Admin system, the signatures are verified. The process verifies that:

- the signature is correct and the file is not tampered with after signing
- the certificate is verified by a trusted certificate authority and was valid on the signing time

- the signing user/system has permission to report counting results for this part of the election

2.5.5 Registering and approving of counting results

The Admin application has functionality for registering of manually counts of paper votes. It is also possible to correct previously registered counting results. Before any counts are included in the total on a higher level, it has to be approved by a user with special permissions. The RBAC system supports exclusive roles. If a user is assigned to such a exclusive roles, she could not be assigned to any other roles. With this, the system can prevent a user which registers a counting result from approving the same result.

2.5.6 Audit log

There is also a logging infrastructure that collects data from each subsystem. Each subsystem generates their own immutable logs, but the central system in the Admin domain also receives copies of the logs continuously. In addition, the central system receives messages from the infrastructure for monitoring purposes.

3. Logical design

Logical Design is the view of the security solution *from the design team's point of view*. It seeks to define as far as possible technology-neutral design which meets the needs of the users.

3.1 Application architecture

The election administration system is one web application. A separate instance of the application is deployed and configured to provide the functionally needed for collecting list proposals from existing and new political parties. The public part of list proposals share database (but not database connection) with rest of the Admin system. The database connection from the public part of list proposals has the only the necessary privilege.

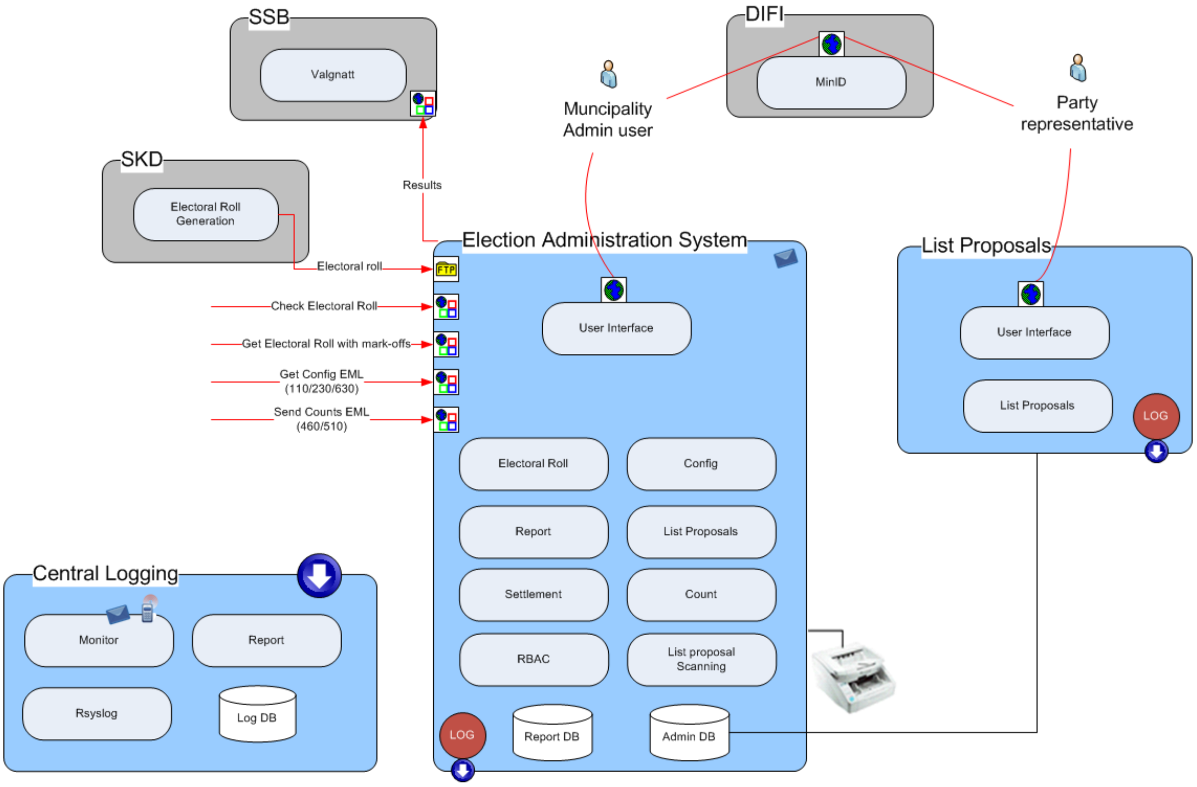


Figure 2: Admin logical design

3.2 Logging, auditing, and monitoring

The central log and audit is a separate system which collects log events from Admin and the other election systems.

In the Admin system every call to methods in the service layer (which is secured by the RBAC system) will be logged. In this way all use of functionality which requires authorization will be logged in the audit log. For the moment, the use of about 330 different methods will be logged from the Admin system.
[Ref.: redmine/issues/3233]

In addition start and stop of the servers and applications will be logged in system - and application logs (not in the audit log).

Web service calls are not logged per se - only the function the services use. The daily updates of the electoral roll are not being logged explicitly. I'm still not sure if these updates use functionality which will be logged.

Authentication is handled by MinID at Difi. Logging in and logging out don't use any methods in the service layer and will consequently, for the moment, not be logged.

4. Physical design

Physical Design is the view of the security solution from the developer's point of view. It seeks to define the configuration of the physical components used to implement the technology solution and to provide a process for operation and support of the system.

4.1 Desktop environment

We have no influence on the clients used by the political parties' representatives accessing the public part of list proposals. They could use any supported web browser on any platform.

The municipal admin client, the poll station client, and the admin client at the scanning site are under municipal administration and - security policy. To access the election administration system, the user needs a supported web browser and an installed SSL client certificate. The required client certificate will be generated by the key management system in the E-voting environment and be distributed to system administrators in the different municipalities. The system administrator has to manually install the certificate on each of the PC which will be used to run the Admin web client.

4.2 Local area network environment

The systems for election administration are deployed at Brønnøysund service provider.

4.3 Wide area networking environment

The web application for election administration is accessed from internet. SSL server certificate are installed for each system and all connection is by HTTPS. All Admin clients (except for list proposals) have installed SSL client certificates. All connections from these clients are by mutual authenticated HTTPS.

4.4 Network components

The internet facing web servers are protected by firewalls. The firewall rules accept only HTTP traffic to these servers. Traffic to election administration system (except public list proposal) is only allowed from municipal owned IP addresses.

The firewalls also accept log traffic from the E-voting environment and systems for E-counting of p-votes down to the central log servers.

Log events from the systems for E-counting of p-votes are sent over internet (default port 514).

A load balancer is also deployed in front of the internet facing web servers. The Admin system (except public list proposals) is deployed in parallel servers for higher availability.

Firewalls or other filtering devices are also deployed between front end and backend of the Admin systems.

4.5 Solution topology

The Admin system is deployed in two data centres at Brønnøysund.

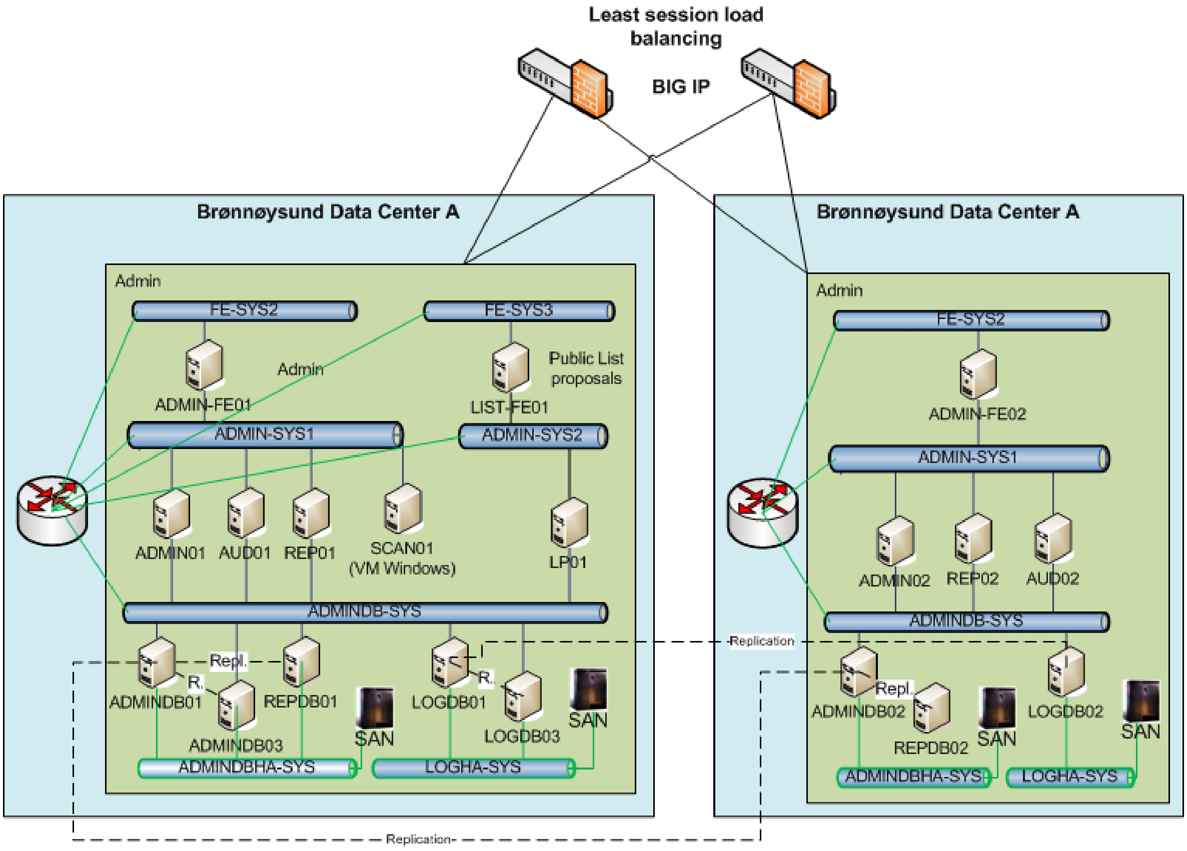


Figure 3: Operational environment for Admin

5. Self-protection of Admin

This section sum up how the Admin system protects itself from interference and tampering.

Only predefined authenticated users get access to the user interface of the system. For the main part of the users, network connection to the system requires a client certificate previously installed on the user's PC by a municipally system administrator. Users of the list proposal part of the application don't have such client certificates. Consequently this part of the application is deployed on a separate server with limited connection to the backend database.

The system offer a limited set of web services. Network connection to these web services is only available from machines with the required client certificate or in the same data centre. The integrity and authenticity of all files uploaded to these web services are secured with digital signatures from the originating systems (eVoting and eCounting). The internal system tasks responding to web services calls are assigned a user role with limited permissions.

Possible threats against the security of the system are assessed and security controls are implemented as counter-measures. The threat model process is documented in the Redmine issue tracking system. Some of the security controls are mention below with reference to the corresponding issue number in Redmine.

To ensure that only system administrators could modify important parts of the system, the RBAC system restrict access to administrative functions (#3614) and only a few system administrators can change the system security attributes (#3615).

All actions on behalf of a user in the Admin application require successfully identification and authentication of that user. To ensure this, all users are uniquely identified by their social security number (#3618) and t he whole Admin application is configured to require authentication with ID-porten (#3619).

Important events in the system are recorded in an audit log (#3611) which is secured from tampering (#2904). The audit logs are monitored for anomalities (#3085) and the systems use NTP (#3630) to enable correlation of the logs.

6. Domain isolation in Admin

This section sum up how the security domains maintained by the Admin system.

The Election Administration domain includes three separate domains; Election Administration System, List Proposal, and Central Logging. The Election Administration System and List Proposal are also split in a Presentation tier, Logical tier, and Data tier.

The separation between the domains is implemented by separate VLANs implemented on a firewall. The servers in central logging are connected to the same VLAN as database servers in the shared data tier. The rest of the domains have their own dedicated VLAN. A firewall restricts the connections to the necessary traffic.

7. Non-bypassibility of security functionality in Admin

This section sum up how the Admin system prevents bypass of the security functionality. The protection mechanisms should always be invoked to prevent that the application can be used to access protected data or resources in an unauthorised way.

Several security controls are in place to ensure that the security functionality of the Admin system could not be bypassed.

To ensure that the restriction on administrative functions could not be bypassed by attacking the clients, the Admin client PCs are hardened (#2797), they are promptly patched when new patches are available (#2798), they have screen lock configured (#2891), and there is a session timeout in the Admin application (#2890).

To ensure that the restriction on administrative functions could not be bypassed by malicious system administrators, direct system access in the production environment is restricted to a few people (#3067), login on central servers and all other direct system access is logged (#3068), and all changes to election configuration files are logged to an audit log (#3072).

To ensure that the restriction on administrative functions could not be bypassed by malicious non-administrative users, SQL injection attacks are prevented (#2786) and an unpredictable token is used to guard against CSRF attacks (#2781).

To ensure that the required identification and authentication could not be bypassed by breaking in on the servers, perimeter security only allow https requests from Internet (#2792), access is only allowed from registered IP ranges (#2795) and requires a valid client certificate (#2796), and serves are hardened (#2789) and promptly patched when new patches are available (#2788).