Kiezen op Afstand

(English: Distant Voting)

Specification of the Functional Requirements

This functional design document is made by LogicaCMG BV for the Dutch ministry of internal affairs

English translation by **Martijn Warnier**[1]
February 22, 2005

Initial **DRAFT**

not for distribution

[1] http://www.cs.ru.nl/~warnier Email:warnier@cs.ru.nl

About the translation

Abreviations have not been translated, when deemed useful additional translations are given in footnotes. Footnotes added by the translator are marked with MW (Martijn Warnier).
For the moment pictures are omitted, unless they are absolutely necessary for the understanding of the main text.

Typeset in LaTeX 2_ε.

Contents

1 Introduction **9**
- 1.1 Purpose . . . 9
- 1.2 Customer . . . 9
- 1.3 Relation to other documents . . . 9
- 1.4 Version control . . . 10
- 1.5 Version . . . 10
- 1.6 Reading guide . . . 10
- 1.7 Application area . . . 10

2 Functional Overview **13**
- 2.1 KR (Voter register) . . . 14
- 2.2 VSL (Virtual polling station) . . . 18
- 2.3 ESB (Electronic voting boot) . . . 20
- 2.4 TSM (Telephonic voting machine) . . . 22
- 2.5 WSM (Web voting machine) . . . 23

3 Role and steps in the voting process **25**
- 3.1 Data maintainer . . . 25
- 3.2 Chairman . . . 25
  - 3.2.1 Tasks of the chairman before polling . . . 25
  - 3.2.2 Tasks of the chairman during polling . . . 25
  - 3.2.3 Tasks of the chairman after polling . . . 25
- 3.3 The voter . . . 25
  - 3.3.1 Voting via PC . . . 25
  - 3.3.2 Voting via telephone . . . 25

4 States and functions of the system **27**
- 4.1 Initial state . . . 28
- 4.2 State “Preparations” . . . 28
  - 4.2.1 Properties . . . 28
  - 4.2.2 Transitions . . . 28
- 4.3 State “Ready for opening” . . . 28
  - 4.3.1 Properties . . . 28
  - 4.3.2 Transitions . . . 28
- 4.4 State “Opening” . . . 28
  - 4.4.1 Properties . . . 28

4.4.2 Transitions . . . 28
4.5 State "Blocked" . . . 28
4.5.1 Properties . . . 28
4.5.2 Transitions . . . 28
4.6 State "Interrupted" . . . 28
4.6.1 Properties . . . 28
4.6.2 Transitions . . . 28
4.7 State "Ready to continue" . . . 28
4.7.1 Properties . . . 28
4.7.2 Transitions . . . 28
4.8 State "Closed" . . . 28
4.8.1 Properties . . . 28
4.8.2 Transitions . . . 28
4.9 State "Ready for opening votes" . . . 28
4.9.1 Properties . . . 28
4.9.2 Transitions . . . 28
4.10 State "Votes opened" . . . 28
4.11 Data flow diagrams . . . 28
4.11.1 Reading voter register . . . 28
4.11.2 Adding/removing voters . . . 28
4.11.3 Loading/getting candidate lists . . . 28
4.11.4 Initialization . . . 28
4.11.5 Blocking . . . 28
4.11.6 Interrupting . . . 28
4.11.7 Re-initialization . . . 28
4.11.8 Preparations for opening votes . . . 28
4.11.9 Opening votes . . . 28

5 Interface voting machine **29**
5.1 General . . . 29
5.2 Statistics . . . 29
5.3 Status tasks . . . 29
5.3.1 Initialization . . . 29
5.3.2 Re-initialization . . . 29
5.3.3 Opening voting . . . 29
5.3.4 Interrupting voting . . . 29
5.3.5 Blocking voting . . . 29
5.3.6 Continuing voting . . . 29
5.3.7 Closing voting . . . 29
5.3.8 Status question . . . 29
5.4 Messages for voting . . . 29
5.4.1 Verification voter . . . 29
5.4.2 Verification candidate . . . 29
5.4.3 Casting vote . . . 29
5.4.4 Signaling fatal errors . . . 29

6 Interchangings format BKZ **31**
6.1 General . . . 31
6.2 Delivering voter data . . . 31
6.3 Complementary voter date . . . 31
6.4 Removing voter data . . . 31
6.5 Delivering candidate lists . . . 31

7 Data model **33**
7.1 Entity relation diagram . . . 33
7.2 Fingerprints . . . 33
7.3 System parameters . . . 33

8 Audit logs, accountability and evaluation **35**
8.1 Audit log . . . 35
8.1.1 General . . . 35
8.1.2 Maintainer actions . . . 35
8.1.3 Status transitions . . . 35
8.1.4 Voting . . . 35
8.2 Accountability data . . . 35
8.2.1 Rapports from the audit log . . . 35
8.2.2 Status rapport . . . 35
8.2.3 Additional counting . . . 35
8.2.4 Results counting . . . 35

9 User interface TSM **37**
9.1 General . . . 38
9.2 Scenario . . . 38
9.3 Status voting station . . . 38
9.3.1 Preparations . . . 38
9.3.2 Ready for opening . . . 38
9.3.3 Open . . . 38
9.3.4 Blocked . . . 38
9.3.5 Interrupted . . . 38
9.3.6 Ready to continue . . . 38
9.3.7 Closed . . . 38
9.4 Input voter code and access code . . . 38
9.5 Verification voter . . . 38
9.5.1 Verification failed . . . 38
9.5.2 Election not open . . . 38
9.5.3 OK . . . 38
9.5.4 Invalid credentials . . . 38
9.5.5 Account locked . . . 38
9.5.6 Already voted . . . 38
9.6 Input candidate code . . . 38
9.7 Verification candidate code . . . 38
9.7.1 Verification failed . . . 38
9.7.2 Election not open . . . 38

9.7.3 OK . . . 38
9.7.4 Invalid candidate . . . 38
9.8 Repeating and confirming candidate code . . . 38
9.9 Deposit vote . . . 38
9.9.1 Voting failed . . . 38
9.9.2 Election not open . . . 38
9.9.3 OK . . . 38
9.9.4 Invalid credentials . . . 38
9.9.5 Account locked . . . 38
9.9.6 Already voted . . . 38
9.10 Messages . . . 38
9.11 Counter statistics . . . 38
9.11.1 Direct counters . . . 38
9.11.2 Derived votes . . . 38
9.11.3 Implementation . . . 38

10 Web interface **39**
10.1 General . . . 39
10.2 Main structure . . . 39
10.3 Welcome screen . . . 39
10.4 Explanation screen . . . 39
10.5 Identification screen . . . 39
10.6 Chose candidate screen . . . 39
10.7 Confirmation screen candidate . . . 39
10.8 Result screen . . . 39
10.9 Overview messages . . . 39

A Glossary **41**

B Rapports **43**
B.1 Status overview . . . 43
B.2 Rapports audit log . . . 43
B.3 Results counting . . . 43

C FRS - Deliverables table **45**

D Performance and capacity demands **47**

List of Figures

2.1 Functional components of the KOA system 13

Chapter 1

Introduction

1.1 Purpose

This document describes the functionality of the Experiment Distant Voting[1] (KOA system). Its purpose is twofold: to give the customer insight in the (internal) working of the system and to form as a basis for technical specifications and testing of the system.

This document will be given to the customer for approval.

1.2 Customer

The customer is the Dutch ministry of internal affairs[2](BZK).

1.3 Relation to other documents

The functional demands of BZK are documented in the initial proposal. Because BZK chose for a service, the functional demands are restricted to the essential conditions involving this service only. In the initial proposal LogicaCGM has proposed a sketch of the to be realized system, including a view of the system from the user's perspective(Voter,Chairman,data maintainer). This description can be found (in slightly altered form) in Chapter 3 and 4 of this document.

The instructions for the voting stations (including screens) are not included in this document. They can be found in the document "Werkinstructie voorzitter" [1]. The same holds for the instructions for the data maintainer, these are included in the document "Werkinstructie Databeheerder" [2]. Only those rapports that are generated by the KOA system as a means of verifying its workings are included in Appendix B.

[1]Dutch: Experiment Kiezen op Afstand. MW
[2]Dutch: Ministerie van Binnenlandse Zaken en Koninkrijksrelaties. MW

1.4 Version control

Version control for this document is done according to the internal rules of the LogicaCGM quality system Cortex[3].

1.5 Version

This version conforms to the system after a first user test.

1.6 Reading guide

- Chapter 3 and 4 contain a description of the (to be realized) system in the form of a functional model and a description of the roles(Voter, Chairman, data maintainer) of the system.
- Chapter 5 contains a point wise description of the functionality of the system in terms of the different states the system can have.
- Chapter 6 contains a functional description of the relation between the TSM and the VSL.
- Chapter 7 contains a functional specification of the files that are used to exchange data (voter registration and candidate lists) between the KOA and the customer (the Dutch ministry of internal affairs).
- Chapter 8 contains the data model, as far as related to the functional specification.
- Chapter 9 contains a lists of logging data that is used for audits and accountability.
- Chapter 10 contains a detailed description of the call flow of the TSM
- The appendices contain a glossary, screen and rapport layouts and the deliverables.

1.7 Application area

The system has been designed for the experiment of BKZ. The specifications leave room at several points for a follow up (see RFP, chapter 1) and can be extended to a national level. Depending on the demands in different situations, additional features and/or changes to the system will be required.

The system, together with organizational and procedural measures, can provide a voting service that meets the specifications and safety rules of the RFP knock-out demand 1 and demand 8.

- Vote secret:

[3]This does not hold for this English translation. MW

it is impossible to connect any given Voter to a valid vote, to ensure confidentiality of the vote;

- Uniqueness:

  every valid Voter can only vote once and this vote will exactly be counted once in the end result;

- Valid voters:

  Only voters who have the legal right to vote should be allowed to vote;

- Integrity:

  the end result of the ballot can not be influenced in any other way than by casting a legal vote.

- Accountability:

  the system generates all the constitutionally required information (to be able to verify the end result)

- Recounting:

  conform the constitutional demands a recount is possible

- Availability:

  Legally allowed voters should, as much as possible, be able to cast their votes. The guidelines from the European commission with respect to the availability of the web sites of the government and the contents thereof have to be taken into consideration (see appendix 8, "Toegankelijkheid van publieke websites en de inhoud daarvan"[4],COM (2001) 529 final, and the resolution of the European Parliament concerning availability of web sites
  (P5_TAPROV(2002)0325));

- Transparency for the Voter

  the Voter should be able to understand and trust the voting process

Bibliography

[1] 45.ECF2651.015. Handleiding Voorzitter, Versie A(4). Technical report, LogicaCMG, August 2003.

[2] 45.ECF2651.015. Werkinstructie databeheerder, Versie A(2). Technical report, LogicaCMG, June 2003.

[4]English: Availability of public web sites and the content thereof. MW

Chapter 2

Functional Overview

The most important components of the KOA system are displayed in Figure 2.1.

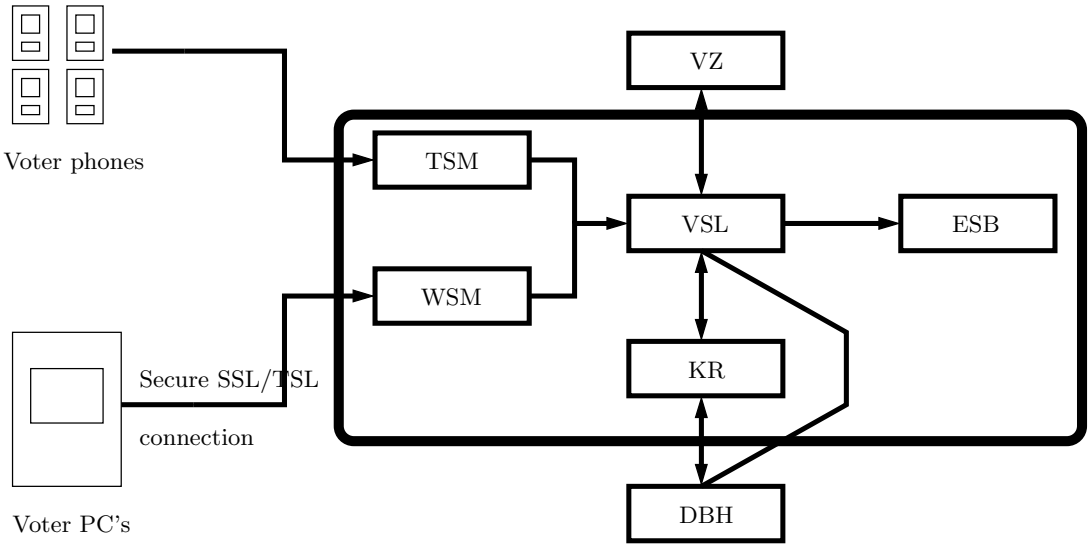

Figure 2.1: Functional components of the KOA system

- **The virtual polling station (VSL)**. This is the central controlling function of the KOA system. On behalf of the Experiment the KOA contains one VSL; one VSL can serve multiple voting circles[1]. Multiple voting machines can be connected to a VSL, this ensures that the different modules of the system (voting via the internet, via phone and possibly, in the future, other systems) can be realized. All voting machines are connected to the VSL in the same manner.
- **The electronic voting box (ESB)**. Each voting circle has one ESB; all casted votes are encrypted and stored in the ESB.
- **The voting register (KR)**. Before the start of the elections all the (anonymous) information of legal voters is stored in the KR. In addition

[1]Dutch: Kieskringen. MW

the KR registers if a voter already casted his/her vote. There is only one KR, even if the Experiment will be extended with multiple virtual polling stations.

- **The web voting machine (WSM)**. This component realizes the link between the PC of the Voter who wants to vote via the internet and the virtual polling station. The Experiment contains one WSM per VSL.
- **The telephonic voting machine (TSM)**. This component realizes the link between the telephone of the Voter who wishes to vote per telephone and the virtual polling station.

The Chairman of the polling station (VZ) communicates on behalf of his role via a secure connection with the VSL. The chairman does not play a role in the actual casting of a vote by the Voter; he only controls the VSL and receives from the VSL the rapports that are required to perform his function. Each VSL can have multiple Chairmen, but only one Chairman is active at any given moment. The KOA system does not make a distinction between the Chairman and the (other) members of the polling station; only one role has been defined for the members of the polling station. In this document "Chairman" refers to "the persons who, acting on behalf of the polling station, performs an action on the system". Only in one instance (changing the status of the System) two members of the polling station have to give their approval (see Section 3.2.1).

The Voter of the voting service will at any given time only communicate with one of the voting machines of the KOA system. Since the voter registration is centralized (in the KR), the Voter does not have to say beforehand which module he/she will use for castings his/her vote. Because of the dimensionality of the system it is preferred to have some indication as to how many users will roughly use each separate module.

The systems data maintainer (DBH) has the following functions:

- reading the candidate lists into the system and delivering the generated candidate codes to the customer. This function takes place before the actual voting has started;
- loading the KR with the anonymous records of legally allowed voters (before the start of the actual voting) and redelivering the file with voter codes to the customer;
- maintaining the voting register after the start of the election (adding voters and stopping the registration of voters). This can also be done during the elections.

Not included in the diagram are facilities for maintaining the system and facilities dealing with audits of the system. Audit logs are described in Chapter 9.

2.1 KR (Voter register)

The Voter (or voting) register contains all data related to legally allowed voters. The Voter register consists of three parts: data to authenticate the Voter, data

concerning the casting of the vote and a table that is used when filling the register.

In order to verify if a voter is legally allowed to vote the following is recorded:

- The voter code which the Voter uses to show that he/she is a legally allowed voter at the moment that he/she wants to cast his/her vote. This voter code is unique and is generated by the KOA system at the moment that all the data is loaded. The voter code is generated in such a way that it fulfills the 11-prove of the (Dutch) social security number[2]. A mistake by the Voter in one digit never results in a valid voter code; a mistake in two numbers only in less than 10 %. In combination with the 5 digit access code of the Voter this results in a chance of less then 1 in 10^6 that a Voter votes with another voter code than his/her own.

- The access code chosen by the Voter. This code is delivered and stored in encrypted form. For the encryption a one-way encryption algorithm is used.

- Indication "removed". This indication is off by default. Only if after starting the election the voter is withdrawn this option can be set to "true"; physically removing the Voter is unwanted, because there might be actions in the logs which refer to the Voter and removing the Voter can thus lead to inconsistencies.

- Voter circle and voter district. The voter circle is used to validate the candidate codes. When a vote is casted both the voter circle and voter district are stored with the vote. This ensures that a result per voter circle can be generated. Results per voter district are also possible, although this function will not be realized for the actual Experiment.

Data concerning the casting of the vote:

- An indication if the Voter has already casted a vote or not.

- At which date and time a vote was casted using which module.

- The delivered transaction code.

- A registration of a failed attempt of a Voter to log in, consisting of:

  - The actual counter of the number of attempts.
  - The total number of attempts.
  - The time of the last attempt.

For filling the KR a table is used (The Voter code allocation register or SAR[3]):

[2]Dutch: de 11-proef van het Sofi nummer. MW
[3]Dutch: het Stemcode Allocatie Register. MW

- The Voter identification. This is an anonymous, unique authentication of the legally allowed voter (delivered by the Customer).

- The Voter code that is generated for the voter (see above).

The SAR ensures that the system will always generate the same voter code for an individual Voter, even if the same voter is delivered to the system multiple times.

The KR is filled with an import file from the Customer. The Customer has to maintain its own register of legally allowed voters. It is required that the KR contains an exact copy of the data of the Customer, therefore an import file is always used. In addition, each imported file is re-exported and delivered back to the Customer for audit purposes.

The import function can work in three different ways:

1. In "replace mode" the import file contains a full copy of all the data of a voter circle. When loading this file, the complete internal copy of the voter circle is deleted and then the import file is processed fully. Thus in case corrections are desired, the full file has to be delivered again. The system guarantees that each legally allowed voter will again have the same voter code as before.

2. In "add" mode no data is deleted, in stead only new voters are registered. This mode is meant for voters who registered after the beginning of the elections.

3. In "replace" mode earlier registered voters are marked as no longer allowed to vote. To maintain system integrity, voters are not physically removed from the system. This functionality is used when a voter is no longer allowed to vote after the start of the elections. It only effects voters that haven't cast their vote yet; votes that are already cast (and thus (anonymously) stored in the system) can not be removed.

As long as an election hasn't started the "replace" mode is used, in order to keep the probability of errors as small as possible. The "add" and "remove" mode are defined because the registration and election periods overlap. The system is designed in such a way that replacing files (with a maximum of 600.000 voters) can be processed within one working day and add/remove files (with a maximum of 30.000 voters) can be processed within one hour.

Other functions of the KR are:

- Initialization: Here the systems checks if all indications that a particular voter has casted a vote and all other data is properly removed. This function is called from the VSL when the elections start.

- Interrupt/continue. When the system is interrupted all data is stored on a removable medium. In addition, an electronic fingerprint of the voter register is calculated that can be used to verify that the state of the system

is identical when the system continues to the interrupted system. This fingerprint is delivered via the VSL to the Chairman.

When the elections are continued either the removable medium or the internal database can be used. In both instances a comparison to the afore mentioned fingerprint, which was generated directly after an interrupt event, is performed. If the fingerprints don't match the Chairman has the final say if the elections can continue.

- Verification. Here the system checks if the voter code is part of the register, if the voter is legally allowed to vote and if the access code matches; if all these requirements are met the system checks if the Voter has already casted a vote or not. To check the access code, it is encrypted with the same one-way encryption function as mentioned before and matched with the stored encrypted code. For security reasons the system does not differentiate between the situations "access code is not present" and "access code not right". The situation "maximum number of attempts reached" *is* displayed individually, together with the time that this situation remains.

  This functionality can be used when a transaction is started to authenticate the Voter and to verify that the Voter has casted a vote.

  The KR further registers how many times authentication fails. After a certain number of attempts all further attempts are blocked for a certain period. The parameters of this function can be changed. The default is a maximum of three attempts and a blocking period of an hour.

- Storing if a voter has casted a vote. This can only be done after an actual vote by a Voter.

- Cleaning: all data concerning the vote transactions are removed.

Checking if the maximum number of authentication attempts is reached is done before the actual authentication itself and uses three data items per Voter: the actual counter, the total number of failed authentication attempts and the time of the last attempt. It works as follows:

- The first check is if the maximum number of attempts is reached (actual counter $\geq$ maximum). If this is the case the system checks if the blocking period is still active (time last attempt + interval > current time). If this is the case the attempt fails (code "Voter blocked"). This is logged.

- If the blocking period ends the actual counter is restored to zero and the time of blocking is deleted.

- Then the actual authentication is performed. If authentication succeeds the status "success" is returned. This is not logged.

- In case authentication fails the actual counter and the total counter are incremented and the time of the last attempt is stored. If in this case the maximum is reached the status becomes once again "Voter blocked"; otherwise the result is "authentication failed". This is also logged.

The total counter per Voter (not to be confused with the counter of the whole register) will thus never be returned to zero. It services to signal systematic attempts to tamper with the data of a certain voter over a longer period (see Section 8.1.4).

2.2 VSL (Virtual polling station)

The VSL is responsible for operating the election process. The VSL distinguishes functions on behalf of the Chairman of the polling station and functions for the operational part of the voting process.

The Chairman uses a specially prepared secure laptop to securely communicate with the VSL. The functions of Chairman are:

- Initialization. This facilitates the preparation of the voting process. The VSL module controls the initialization of all the components and guards the incoming confirmations of the separate components of the KOA system. If all components have confirmed to the Chairman that they are ready to perform their task the election can start.

- Opening the election. This can only happen when the system is properly initialized. At the start of the elections the VSL gets the public key of an RSA key pair from the Chairman. This key is used to encrypt all incoming votes.

- Interrupting the elections. From this moment onwards the VSL refuses any incoming commands from the voting machines. All components receive a command to halt their activities. Both the ESB and the KR confirm this by sending an electronic fingerprint (see Section 2.1 and Section 2.3). The VSL bundles these and returns them to the Chairman.

- Re-initializing. This can only happen if the elections are interrupted. When re-initialization is performed the Chairman can chose to do this using a removable media (which can be necessary when a backup system has to be used) or it can be done using the existing database. In both cases a new fingerprint is generated which is compared to the one that was generated at the moment that the elections where interrupted. If the fingerprints don't match the Chairman has to decide if the elections can continue.

- Continuing the elections. This is only possible if the system has been interrupted and re-initialized using one of methods explained above. The public key of the Chairman is retransmitted and compared to the original public key; this prevents that the votes in the ESB are encrypted with multiple keys. The VSL then clears the system for continuing the elections.

- Closing the elections. From this moment onwards the VSL refuses to accept any new votes casted from one of the voting machines. All components of the system receive a command to stop their activities. The

ESB and KR confirm this by sending a fingerprint to the VSL (see Section 2.1 and Section 2.3). The VSL bundles these and delivers them to the Chairman. Voters receive the message that the elections are closed.

- Opening the votes[4]. This can only happen if the elections are closed and this action can only be performed once during any given election (excluding the case where a state of the system is restored using a back up and the election process continues with the appropriate action). With this action the Chairman has to use his private key. This private key is send to the ESB which uses it to decrypt the votes and transform them to a specially formatted file. This forms the basis for the counting process, that is started at this moment. The result (in rapport form) is delivered to the Chairman.

- Requesting status. This results in a standard rapport of the progress of the elections (See Section 8.2.2). This rapport also services as a main interface for using the other functions of the system.

- Requesting incidents. If an incident occurs within the system that's detected and forms a threat (or can become a threat) for the election process, then this is logged within the system which can be requested by the Chairman.

The second function of the VSL module concerns the operational part of the elections. In order to achieve this the voting machines (TSM, WSM) communicate with the VSL module. The functions in this interface consist of:

- Verification if a voter is legally allowed to vote: this ensures that a voter is legally allowed to vote. The question consists of the access code and the voter code. The VSL module checks if the election is not interrupted or closed and answers the question by querying the KR (See Section 5.4.1 for possible responds).

- Verification candidate: this ensures for a voting machine that a candidate code is correct for a specific Voter, i.e., if the candidate code is contained within the candidate list that corresponds to the voter circle of this particular Voter.

- Casting a vote. the question consists of a voter code and an access code, together with the choice (candidate code) of the Voter. The VSL checks the same items as during verification; if the Voter is indeed allowed to vote, the choice is encrypted and stored in the ESB. The status of the Voter is corrected to prevent that double votes are possible. A transaction code is generated that is stored in the KR and passed on to the Voter as confirmation that a successful vote has been cast. This transaction code is a random number that is generated in the same manner and confirms to the same criteria as the voter code. Using the transaction code is not

[4]The System indicates this with "opnemen stemmen" (English: loading votes) in stead of "lichten stemmen" (English: Opening votes)

mandatory; a system parameter determines if a transaction code is passed on to the user. This does not concern the internal working of the system. The transaction code will always be generated and e.g. passed on to the TSM. The TSM has a similar parameter that decides if the transaction code will be read out aloud or not.

Besides these functions the VSL passes status commands (initialization, elections interrupted, elections continued, elections closed) to the voter machines.

For security reasons each candidate gets 1000[5] different codes assigned to him/her. The candidate lists that are send to the Voter contain one of these codes. This makes it possible to generate a large number of different candidate lists. This security measure is used for voting via telephone; If a connection is monitored it is not easy to match the candidate code to any specific candidate without a complete list of all candidate codes. The complete list can not be generated form any given candidate list. The VSL generates these codes when the candidate lists are imported (see Section 6.5) and maintains them. When a vote is stored this code is translated back to a position at the original candidate list.

It is crucial in this setting that all data concerning the casting of a single vote are performed in one transaction by the voting machines. This simplifies the processing of the votes; it is not necessary to keep track of the fact that the vote is in progress. In case the Voter tries to cast multiple votes at once (e.g. via telephone and internet) there will always be one first vote. This vote will be stored; the second attempt will fail because the Voter already casted a vote.

2.3 ESB (Electronic voting boot)

All votes are encrypted and stored in the electronic voting boot (ESB). There is one ESb per voter circle. Each casted vote is stored once, it contains the contents of the candidate code of the candidate that received the vote (or a fictive candidate with semantics "blanc vote"[6]. The name and initials of the candidate, the position on the candidate list, the list number and the political party of the candidate are added as well. This added data is not really necessary, but in cases of emergency it can be used to obtain the results of the elections form a removable media, outside the context of the virtual polling station. The voting district is also stored together with the vote, in order to (in a later version) give election results per voter district.

Within the system blanc votes are realized as an always present list with only one candidate. For security reasons the rest of the blanc votes are treated as much as possible as ordinary candidates.

The votes are encrypted when they are stored using the public key of the Chairman of the polling station (see Section 2.2). This guarantees two things:

[5] The number 1000 is a system parameter, that can be changed

[6] Under Dutch law it is possible to cast a vote without voting for a candidate, this is called a blanc vote (Dutch: blanco stem). MW

that the votes can only be opened by the Chairman of the polling station and that the corresponding VSL is responsible for the casted votes. Random data is added to the votes when they are encrypted. This ensures that votes within the same voter district and for the same candidate have a different encryption result for each vote, making it impossible to interpret encrypted votes.

The ESB has the following functions:

- Initialization. The system checks if the storage medium is empty. The VSL calls this function when the elections are started.
- Interrupt/continue. When the system is interrupted, the votes are stored on a removable device. A fingerprint of the stored votes is made which ensures that the data is the same when the elections are continued. This fingerprint is returned via the VSL to the Chairman.

  When the elections continue the database can be restored from the removable device (re-initialization) or the database can be used in its current form. In both cases the previously generated fingerprint is compared to a freshly generated one. When the fingerprints don't match the Chairman has the final say in the continuation of the elections.
- Deposit/cast vote.
- Opening the votes: The ESB receives the private key of the Chairman (via the VSL). This key is used to decrypt the votes in a random order (to making tracing voters by the order in which they voted impossible). The (decrypted) votes are then stored in a specially formatted 'counting' file. The VSL module uses this file to count the votes and deliver the result of the elections to the Chairman.
- Recounting. This is not a function of the ESB itself, but the ESB can be used to facilitate recounting. The next diagram clarifies this:

  [Diagram omitted,MW]

  This diagram clarifies that recounting is possible in several different ways:

  1. A backup of the votes is stored within the system. This backup can be used to count the votes after which the whole process can be repeated.
  2. The votes can be exported in encrypted form. A separate system can then decrypt the votes and count them. The fingerprint of this file can be compared to the fingerprint of the that is stored in the audit log in order to ensure that the votes are the same ones as the ones that are actually casted. This forms, within the context of the KOA system, the most logical interpretation of the lawfully required possibility of a recount: a facility that allows a separate system to count the votes with a check that it are indeed the same votes that were actually casted.

3. The decrypted votes can also be exported. This exported file can directly be loaded into a separate system that can then count the votes.

2.4 TSM (Telephonic voting machine)

The TSM forms a so called Voice Response system. It asks questions to the Voter in a predetermined order. The Voter answers by pressing the keys of his/her phone.

A system with a menu is not used, because the Customer demanded that the votes per telephone should be protected against eaves dropping. It is assumed that the Voter has his/her overview of the candidate lists ready for use.

The scenario for a Voter that uses this facility goes as follows:

- The Voter hears a short welcome message as confirmation that he/she has dialed the correct telephone number.

- Then the Voter is asked to first give his voter code and then the access code that is chosen by the Voter him/herself. It is required that this access code is numerical and has a predetermined length. This length is chosen to be 5 digits[7]. The access code is used to ensure that somebody who intercepts the voter code can't vote for that Voter.

  The TSM checks the incoming data with the VSL. If the data is incorrect the Voter has two more tries before the connection is closed.

- Then the Voter is asked to give the number of the candidate he/she wants to vote for. Again, because of security concerns an acknowledgment is given by reading the candidate code back to the user. This only confirms that the Voter voted for the correct person. One could also speak the name of the candidate, but this ignores the security restrictions (i.e. eaves dropping).

  If the Voter does not confirm his/her vote then he/she can continue with a new vote (phase 3) or cancel the process.

- The TSM sends the voter data to the VSL and receives a transaction code. This code is read out to the Voter (depending on the system parameter that allows this, see Section 2.2). Then the connection is closed. At the moment that the VSL receives the vote it is final (unless a technical error occurs when the vote is stored). So this is still the case if the Voter disconnects before the transaction code is received.

[7] The 5 digit code forms a compromise. A longer code gives more security, but makes the chance of errors bigger. Banks use a 4 digit PIN, but this is in combination with a physical token (the bank card). A relevant example forms the Dutch taxing bureau, which also uses a 5 digit PIN for online transactions.

The rule that the access code is restricted to 5 digits is also used for voting via the PC, because one of the functional requirements states that the Voter can chose at the last moment which voting facility he/she will use.

The Chairman's status commands are redirected to the TSM via the VSL. The TSM becomes active at the moment that the initialization command is received; until this moment each call receives a short standard message stating that the elections haven't started yet. The same holds when the system is interrupted ("Elections interrupted") and when the TSM has no live connection with the VSL ("The voting service is not accessible at this moment"). After the elections have ended the standard message is substituted for a message which states that the elections have closed.

The TSM checks at three moments if the VSL is accessible (and the elections are open): when the Voter is verified (stage 2), when the candidate is checked and when the vote is casted. The Voter will be warned at moments that he/she him/herself undertakes an action; the dialog will never be interrupted with a warning that the voting station is closed or the elections are interrupted.

2.5 WSM (Web voting machine)

The Web Voting machine (WSM) is a standard web-server. All web-pages of the WSM conform, as far as appropriate, to the demands of the Customer[8]. The following guidelines have been used for implementation of the WSM:

- The interface has been kept as sober as possible. This ensures that a minimum of data has to be transfered, which ensures that voters with a slow connection can still use this facility.
- The requirements for the browser are kept to a minimum. The most important requirement is that the browser should support encrypted connections; almost all browsers conform to this. The browser also has to allow session cookies (cookies which are removed when the connection is aborted). This is almost always the case, even for PC's behind a firewall[9].

  Users that try to connect to the standard http port are immediately redirected to the secure pages.

The menu structure of the web-interface is as follows:

1. The Voter first sees a couple of pages which briefly explain the voting process, then a verification screen is displayed which requires the Voter to fill the voter code and in his/her (self chosen) access code.

   The WSM verifies these with the VSL. If the data is incorrect the voters can try a predetermined number of times to obtain; if the verifications keeps on failing an error message is displayed. A new attempt is no longer possible, the voter has to log in again. It is also possible that the polling station is not open yet or that the voter is blocked; in these cases an appropriate error message is displayed and the Voter cannot try to log in again.

[8] See Section 8 of the initial proposal.

[9] SSL version 3.0 or TLS (the official name for SSL V3.1) is used for the secure connection. All communication uses the standard port 443 (for https)

If it turns out, during verification, that the Voter has already voted then a message is displayed which states this together with the earlier acquired transaction code. This ensures that the Voter can verify if his/her vote has been registered, e.g. because the connection was interrupted before the transaction code was displayed during an earlier attempt.

2. In stage 2 the Voter sees a screen that allows him/her to enter the candidate code. It is assumed that the Voter has his/her list with candidate codes within reach. This candidate code list also lists the candidate code for a blanc vote. This implementation was chosen to keep the interfaces via the web and phone as closely related as possible.

3. The vote is transfered to the VSL in stage 3, in order to verify if the candidate code is correct, i.e. is the candidate code equal to the number that is used for a candidate in the voter circle of that particular Voter. If this is correct the system displays a screen that contains the name of the list (political party) of the candidate and the name and place of residence of the candidate. From this screen the Voter can either confirm his/her vote or return to stage 2.

4. In stage 4 the WSM transfers the vote of the Voter to the VSL which confirms the vote by sending a transaction code back to the WSM. This transaction code is then displayed on the final screen (depending on the corresponding system parameter, see Section 2.2. At the moment that the vote is transfered to the VSL the vote is final (unless a technical error occurs when the vote is stored). Thus this is also the case if the Voter aborts the connection before the transaction code can be displayed.

The system checks at three different occasions, the same as with the TSM, if the polling station is open: when verifying the data of the Voter (stage 2), when the candidate is checked (stage 3) and when the actual vote is casted (stage 4). The Voter is only warned that the polling station has closed when he/she him/herself commits an action; so no pop-up windows (or the like) are used to warn the user that the elections have stopped or are interrupted.

All status commands of the Chairman are redirected to the WSM via the VSL. The WSM becomes active at the moment that the command to start the elections has been received; until this moment a standard message is displayed that states that the elections haven't started yet. When the elections are interrupted another message is displayed ("elections interrupted"). If the WSM can (temporarily) not connect to the VSL the message "the voting service is temporarily not available" is displayed.

When the elections are closed the standard message is replaced by a message which stated that the polling station is closed.

The WSM is only used for voting. A separate web-server has to be used for information purposes and other such activities.

Chapter 3

Role and steps in the voting process

3.1 Data maintainer

3.2 Chairman

3.2.1 Tasks of the chairman before polling

3.2.2 Tasks of the chairman during polling

3.2.3 Tasks of the chairman after polling

3.3 The voter

3.3.1 Voting via PC

3.3.2 Voting via telephone

Chapter 4

States and functions of the system

4.1 Initial state

4.2 State “Preparations”

4.2.1 Properties

4.2.2 Transitions

4.3 State “Ready for opening”

4.3.1 Properties

4.3.2 Transitions

4.4 State “Opening”

4.4.1 Properties

4.4.2 Transitions

4.5 State “Blocked”

4.5.1 Properties

4.5.2 Transitions

4.6 State “Interrupted”

4.6.1 Properties

4.6.2 Transitions

4.7 State “Ready to continue”

4.7.1 Properties

4.7.2 Transitions

4.8 State “Closed”

Chapter 5

Interface voting machine

5.1 General

5.2 Statistics

5.3 Status tasks

5.3.1 Initialization

5.3.2 Re-initialization

5.3.3 Opening voting

5.3.4 Interrupting voting

5.3.5 Blocking voting

5.3.6 Continuing voting

5.3.7 Closing voting

5.3.8 Status question

5.4 Messages for voting

5.4.1 Verification voter

5.4.2 Verification candidate

5.4.3 Casting vote

5.4.4 Signaling fatal errors

Chapter 6

Interchangings format BKZ

6.1 General

6.2 Delivering voter data

6.3 Complementary voter date

6.4 Removing voter data

6.5 Delivering candidate lists

Chapter 7

Data model

7.1 Entity relation diagram

7.2 Fingerprints

7.3 System parameters

Chapter 8

Audit logs, accountability and evaluation

8.1 Audit log

8.1.1 General

8.1.2 Maintainer actions

8.1.3 Status transitions

8.1.4 Voting

8.2 Accountability data

8.2.1 Rapports from the audit log

8.2.2 Status rapport

8.2.3 Additional counting

8.2.4 Results counting

Chapter 9

User interface TSM

9.1 General

9.2 Scenario

9.3 Status voting station

9.3.1 Preparations

9.3.2 Ready for opening

9.3.3 Open

9.3.4 Blocked

9.3.5 Interrupted

9.3.6 Ready to continue

9.3.7 Closed

9.4 Input voter code and access code

9.5 Verification voter

9.5.1 Verification failed

9.5.2 Election not open

9.5.3 OK

9.5.4 Invalid credentials

9.5.5 Account locked

9.5.6 Already voted

9.6 Input candidate code

9.7 Verification candidate code

9.7.1 Verification failed

9.7.2 Election not open

Chapter 10

Web interface

10.1 General

10.2 Main structure

10.3 Welcome screen

10.4 Explanation screen

10.5 Identification screen

10.6 Chose candidate screen

10.7 Confirmation screen candidate

10.8 Result screen

10.9 Overview messages

Appendix A

Glossary

Appendix B

Rapports

B.1 Status overview

B.2 Rapports audit log

B.3 Results counting

Appendix C

FRS - Deliverables table

Appendix D

Performance and capacity demands