EML Process & Data Requirements

Version 4.0d

03 September 2004

Document identifier:

EML v4.0d Process and Data Requirements

Editor:

eGovernment Unit, Cabinet Office, UK

Contributors:

John Ross
Paul Spencer
John Borras
Farah Ahmed

Abstract:

This document describes the background and purpose of the Election Markup Language, the electoral processes from which it derives its structure and the security and audit mechanisms it is designed to support.

The relating document entitled 'EML v4.0d Schema Descriptions' lists the schemas and schema descriptions to be used in conjunction with this specification.

Status:

This document is updated periodically on no particular schedule. Committee members should send comments on this specification to the election@lists.oasis-open.org list. Others should subscribe to and send comments to the election-services-comment@lists.oasis-open.org. To subscribe, send an email message to election-comment-request@lists.oasis-open.org with the word "subscribe" as the body of the message.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Election and Voter Services TC web page (http://www.oasis-open.org/committees/election/).

Table of Contents

1 Executive Summary

OASIS, the XML interoperability consortium, formed the Election and Voter Services Technical Committee in the spring of 2001 to develop standards for election and voter services information using XML. The committee's mission statement is, in part, to:

"Develop a standard for the structured interchange among hardware, software, and service providers who engage in any aspect of providing election or voter services to public or private organizations..."

The objective is to introduce a uniform and reliable way to allow systems involved in the election process to interact. The overall effort attempts to address the challenges of developing a standard that is:

- **Multinational**: Our aim is to have these standards adopted globally.
- **Flexible**: Effective across the different voting regimes (e.g. proportional representation or 'first past the post') and voting channels (e.g. Internet, SMS, postal or traditional paper ballot).
- **Multilingual**: Flexible enough to accommodate the various languages and dialects and vocabularies.
- **Adaptable**: Resilient enough to support elections in both the private and public sectors.
- **Secure**: Able to secure the relevant data and interfaces from any attempt at corruption, as appropriate to the different requirements of varying election rules.

The primary deliverable of the committee is the Election Markup Language (EML). This is a set of data and message definitions described as XML schemas. At present EML includes specifications for:

- Candidate Nomination, Response to Nomination and Approved Candidate Lists
- Referendum Options Nomination, Response to Nomination and Approved Options Lists
- Voter Registration information, including eligible voter lists
- Various communications between voters and election officials, such as polling information, election notices, etc.
- Ballot information (races, contests, candidates, etc.)
- Voter Authentication
- Vote Casting and Vote Confirmation
- Election counts and results
- Audit information pertinent to some of the other defined data and interfaces

EML is flexible enough to be used for elections and referendums that are primarily paper-based or that are fully e-enabled.

1.1 Overview of the Document

To help establish context for the specifics contained in the XML schemas that make up EML, the committee also developed a generic election process model. This model identifies the components and processes common to many elections and election systems, and describes how EML can be used to standardize the information exchanged between those components.

Section 2 outlines the business and technical needs the committee is attempting to meet, the challenges and scope of the effort, and introduces some of the key framing concepts and terminology used in the remainder of the document.

Section 3 describes two complementary high-level process models of an election exercise, based on the human and technical views of the processes involved. It is intended to identify all the generic steps involved in the process and highlight all the areas where data is to be exchanged. The discussions in this section present details of how the messages and data formats detailed in the EML specifications themselves can be used to achieve the goals of open interoperability between system components.

Section 4 presents a discussion of the some of the common security requirements faced in different election scenarios, a possible security model, and the mechanisms that are available in the EML specifications to help address those requirements. The scope of election security, integrity and audit included in these interface descriptions and the related discussions are intended to cover security issues pertinent only to the standardised interfaces and not to the internal security requirements within the various components of election systems.

The security requirement for the election system design, implementation or evaluation must be placed with the context of the vulnerabilities and threats analysis of a particular election scenario. As such the references to security within EML are not to be taken as comprehensive requirements for all election systems in all election scenarios, nor as recommendations of sufficiency or approach when addressing all the security aspects of election system design, implementation or evaluation.

Section 5 provides an overview of the approach that has been taken to creating the XML schemas.

Section 6 provides information as to the location of the descriptions of the schemas developed to date.

Appendices provide information on internet voting security concerns, TimeStamp schema, W3C Digital Signature and a revision history.

2 Introduction

2.1 Business Drivers

Voting is one of the most critical features in our democratic process. In addition to providing for the orderly transfer of power, it also cements the citizen's trust and confidence in an organization or government when it operates efficiently. In the past, changes in the election process have proceeded deliberately and judiciously, often entailing lengthy debates over even the most minute detail. These changes have been approached with caution because discrepancies with the election system threaten the very principles that make our society democratic.

Times are changing. Society is becoming more and more web oriented and citizens, used to the high degree of flexibility in the services provided by the private sector and in the Internet in particular, are now beginning to set demanding standards for the delivery of services by governments using modern electronic delivery methods.

Internet voting is seen as a logical extension of Internet applications in commerce and government and in the wake of the United States 2000 general elections is among those solutions being seriously considered to replace older less reliable election systems.

The implementation of electronic voting would allow increased access to the voting process for millions of potential voters. Higher levels of voter participation will lend greater legitimacy to the electoral process and should help to reverse the trend towards voter apathy that is fast becoming a feature of many democracies. However, it has to be recognized that the use of technology will not by itself correct this trend. Greater engagement of voters throughout the whole democratic process is also required.

However, it is recognized that more traditional voting methods will exist for some time to come, so a means is needed to make these more efficient and integrate them with electronic methods.

2.2 Technical Drivers

In the election industry today, there are a number of different services vendors around the world, all integrating different levels of automation, operating on different platforms and employing different architectures. With the global focus on e-voting systems and initiatives, the need for a consistent, auditable, automated election system has never been greater.

The introduction of open standards for election solutions is intended to enable election officials around the world to build upon existing infrastructure investments to evolve their systems as new technologies emerge. This will simplify the election process in a way that was never possible before. Open election standards will aim to instill confidence in the democratic process among citizens and government leaders alike, particularly within emerging democracies where the responsible implementation of the new technology is critical.

2.3 The E&VS Committee

OASIS, the XML interoperability consortium, formed the Election and Voter Services Technical Committee to standardize election and voter services information using XML. The committee is focused on delivering a **reliable, accurate and trusted** XML specification (Election Markup Language (EML)) for the structured interchange of data among hardware, software and service vendors who provide election systems and services.

EML is the first XML specification of its kind. When implemented, it can provide a uniform, secure and verifiable way to allow e-voting systems to interact as new global election processes evolve and are adopted.

The Committee's mission statement is:

"Develop a standard for the structured interchange of data among hardware, software, and service providers who engage in any aspect of providing election or voter services to public or private organizations. The services performed for such elections include but are not limited to voter role/membership maintenance (new voter registration, membership and dues collection, change of address tracking, etc.), citizen/membership credentialing, redistricting, requests for absentee/expatriate ballots, election calendaring, logistics management (polling place management), election notification, ballot delivery and tabulation, election results reporting and demographics."

The primary function of an electronic voting system is to capture voter preferences reliably and report them accurately. Capture is a function that occurs between 'a voter' (individual person) and 'an e-voting system' (machine). It is critical that any election system be able to prove that a voter's choice is captured correctly and anonymously, and that the vote is not subject to tampering.

Dr. Michael Ian Shamos, a PhD Researcher who worked on 50 different voting systems since 1980 and reviewed the election statutes in half the US states, summarized a list of fundamental requirements, or 'six commandments', for electronic voting systems:

1. Keep each voter's choice an inviolable secret.
2. Allow each eligible voter to vote only once, and only for those offices for which he/she is authorized to cast a vote.
3. Do not permit tampering with voting system, nor the exchange of gold for votes.
4. Report all votes accurately
5. The voting system shall remain operable throughout each election.
6. Keep an audit trail to detect any breach of [2] and [4] but without violating [1].

In addition to these business and technical requirements, the committee was faced with the additional challenges of specifying a requirement that was:

- Multinational – our aim is to have these standards adopted globally
- Effective across the different voting regimes – for example, proportional representation or 'first past the post', preferential voting, additional member system
- Multilingual – our standards will need to be flexible enough to accommodate the various languages and dialects and vocabularies
- Adaptable – our aim is to provide a specification that is resilient enough to support elections in both the private and public sectors
- Secure – the standards must provide security that protects election data and detects any attempt to corrupt it.

The Committee followed these guidelines and operated under the general premise that any data exchange standards must be evaluated with constant reference to the public trust.

2.4 Challenge and Scope

The goal of the committee is to develop an Election Markup Language (EML). This is a set of data and message definitions described as a set of XML schemas and covering a wide range of transactions that occur during an election. To achieve this, the committee decided that it required a common terminology and definition of election processes that could be understood internationally. The committee therefore started by defining the generic election process models described here.

These processes are illustrative, covering the vast majority of election types and forming a basis for defining the Election Markup Language itself. EML has been designed such that elections that do not follow this process model should still be able to use EML as a basis for the exchange of election-related messages.

EML is focussed on defining open, secure, standardised and interoperable interfaces between components of election systems. Thus providing transparent and secure interfaces between various parts of an election system. The scope of election security, integrity and audit included in these interface descriptions and the related discussions are intended to cover security issues pertinent only to the standardised interfaces and not to the internal or external security requirements of the various components of election systems.

The security requirement for the election system design, implementation or evaluation must be placed within the context of the vulnerabilities and threats analysis of a particular election scenario. As such the references to security within EML are not to be taken as comprehensive requirements for all election systems in all election scenarios, nor as recommendations of sufficiency of approach when addressing all the security aspects of election system design, implementation or evaluation. In fact, the data security mechanisms described in this document are all optional, enabling compliance with EML without regard for system security at all.

A complementary document may be defined for a specific election scenario, which refines the security issues defined in this document.

EML is meant to assist and enable the election process and does not require any changes to traditional methods of conducting elections. The extensibility of EML makes it possible to adjust to various e-democracy processes without affecting the process, as it simply enables the exchange of data between the various election processes in a standardized way.

The solution outlined in this document is non-proprietary and will work as a template for any election scenario using electronic systems for all or part of the process. The objective is to introduce a uniform and reliable way to allow election systems to interact with each other. The proposed standard is intended to reinforce public confidence in the election process and to facilitate the job of democracy builders by introducing guidelines for the selection or evaluation of future election systems.

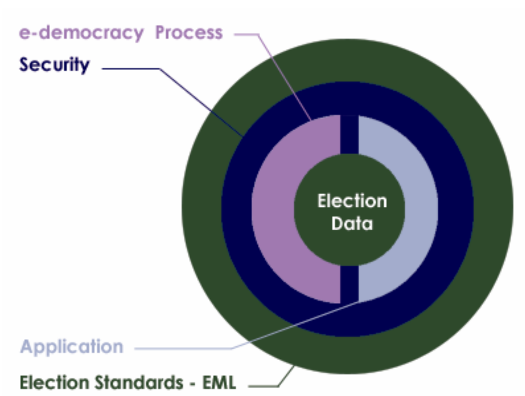


Figure 1A: Relationship overview

2.5 Documentation Set

To meet our objectives, the committee has defined a process model that reflects the generic processes for running elections in a number of different international jurisdictions. The processes are illustrative, covering a large number of election types and scenarios.

The next step was then to isolate all the individual data items that are required to make each of these processes function. From this point, our approach has been to use EML as a simple and standard way of exchanging this data across different electronic platforms. Elections that do not follow the process model can still use EML as a basis for the exchange of election-related messages at interface points that are more appropriate to their specific election processes.

The EML specification is being used in a number of pilots to test it's effectiveness across a number of different international jurisdictions. The committee document set will include:

- **Voting Processes:** A general and global study of the electoral process. This introduces the transition from a complete human process by defining the data structure to be exchanged and where they are needed.
- **Data Requirements:** A data dictionary defining the data used in the processes and required to be handled by the XML schemas.
- **EML Specifications:** This consists of a library of XML schemas used in EML. The XML schemas define the formal structures of the election data that needs to be exchanged.
- **Report on Alternative methods of EML Localisation:** EML provides a set of constraints common to most types of elections worldwide. Each specific election type will require additional constraints, for example, to enforce the use of a seal or to ensure that a cast vote is anonymous. This document describes alternative mechanisms for expressing these constraints and recommends the use of schemas using the Schematron language to supplement the EML schemas for this purpose.

2.6 Conformance

To conform to this specification, a system must implement all parts of this specification that are relevant to the interfaces for which conformance is claimed. The required schema set will normally be part of the purchasing criteria and should indicate schema version numbers. For example, in the future, the specification for an election list system might specify that a conforming system must accept and generate XML messages conforming to the following schemas:

Schema	Accept	Generate
EML110	v4.0, v3.0	
EML310	v4.0, v3.0	
EML330		v4.0
EML340		v4.0
EML350		v4.0
EML360		v4.0

A conforming system will then conform to the relevant parts of this specification and the accompanying schemas.

2.7 Terminology

At the outset of our work, it was clear that the committee would need to rationalize the different terms that are commonly used to describe the election process.

Terms used to describe the election process, such as ballot and candidate, carry different meanings in different countries – even those speaking the same language. In order to develop a universal standard, it is essential to create universal definitions for the different elements of the election process. See the Data Dictionary for the terms used by the committee in this document

Our approach was to regard elections as involving **Contests** between **Candidates** or **Referendum Options** which aggregate to give results in different **Elections**.

In practice however, electoral authorities would often run a number of different elections during a defined time period. This phenomenon is captured in our terminology as an **Election Event**. Figure 1B uses a British context to describe our approach in general terms.

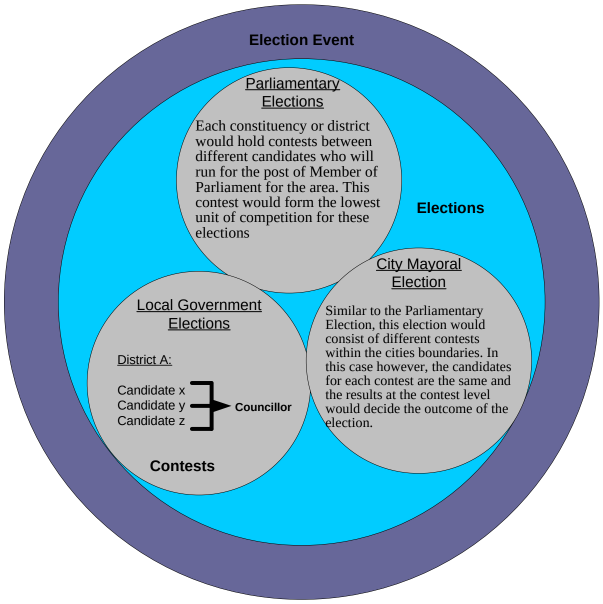


__Figure 1B: The Election Hierarchy__

In Figure 1C, there is an **Election Event** called the 'Union Annual Election'. This comprises two **Elections**, one for the National Executive Committee (NEC) and one for the International Liaison Committee (ILC). Three positions are being selected for each committee; as a result, each **Election** is made up of three **Contests**. In region 1 (R1), the **Contest** for each **Election** has two **Candidates**.

Figure 1C shows the three **Ballots** (one for each region). The **Ballot** is personal to the voter and presents the **Candidates** available to that voter. It also allows choices to be made. During the election exercise, each voter in region 1 (R1) receives only the region 1 ballot. This ballot will contain the **Candidates** for the R1 contest for each of the two **Elections.**

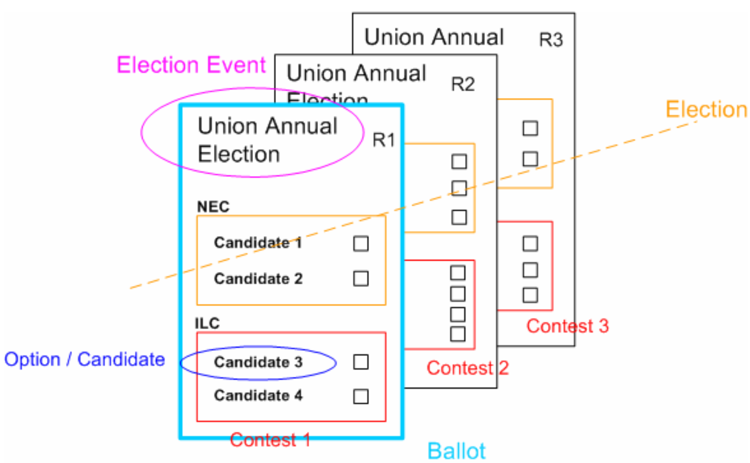


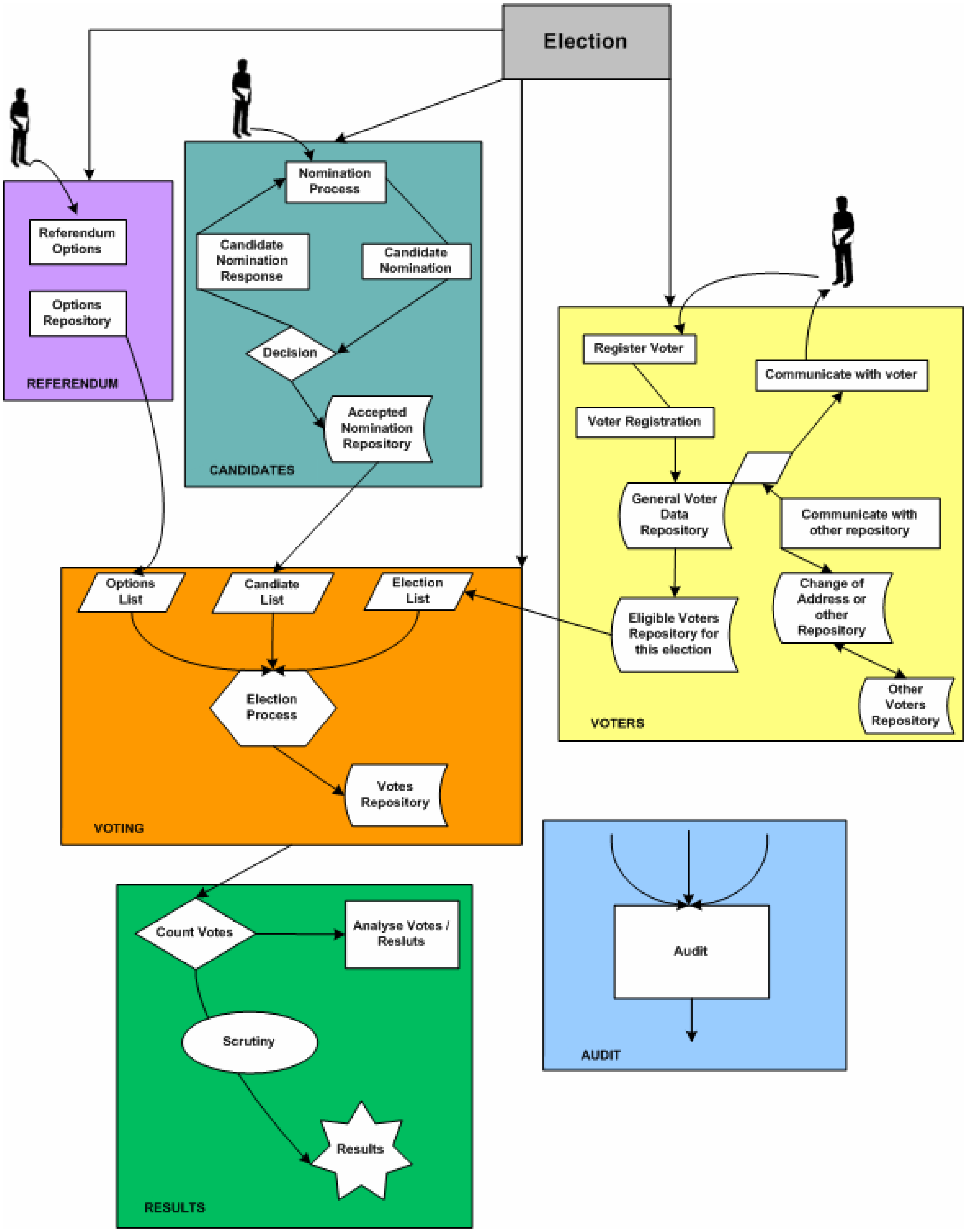
__Figure 1C: Union Annual Election__

3 High-Level Election Process

Section 3 describes two complementary high level process models of an election exercise, based on the human and technical views of the processes involved. It is intended to identify all the generic steps involved in the process and all the areas where data is to be exchanged highlight all the areas where data is to be exchanged.

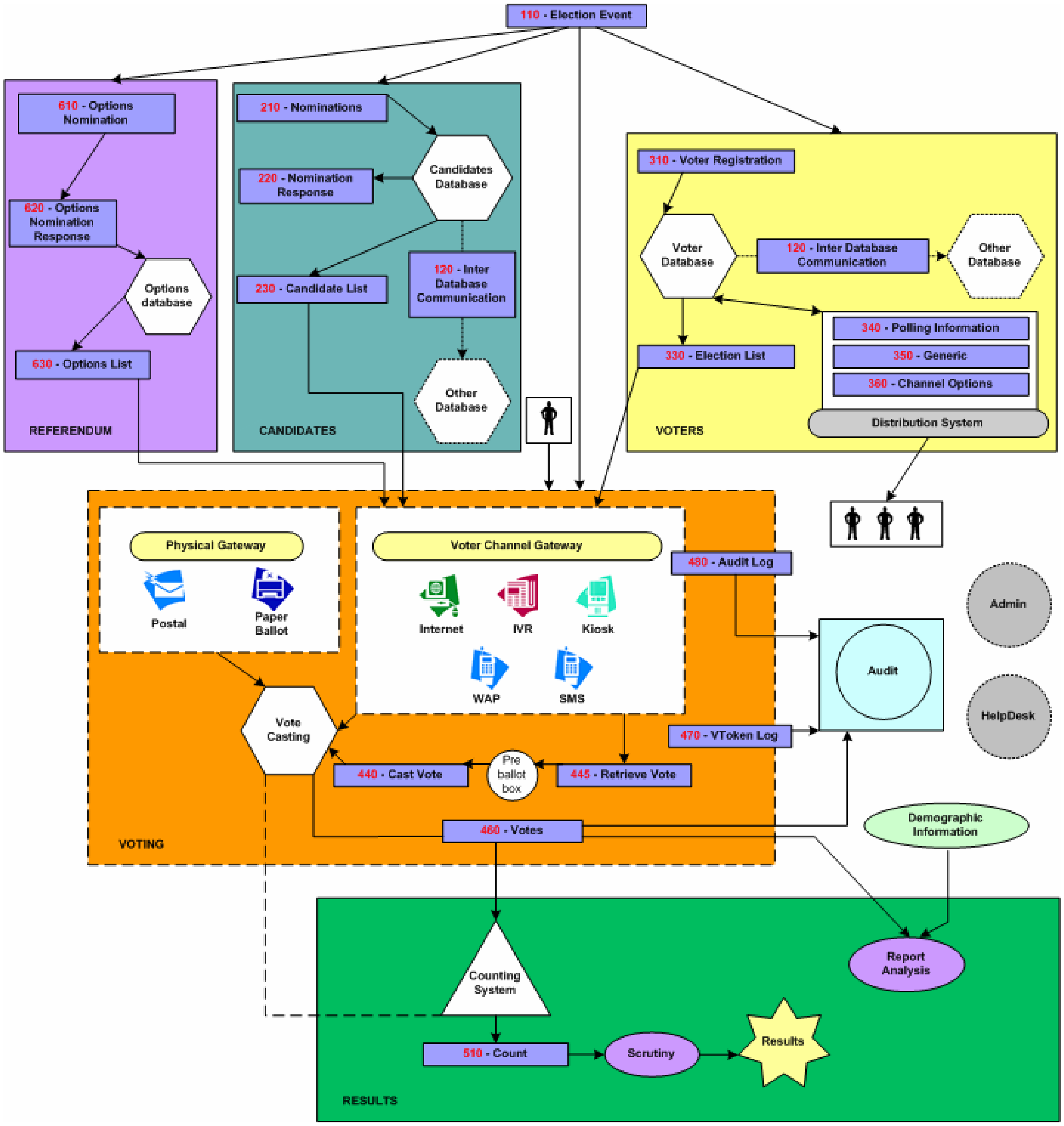
329

3.1 Figure 2A: High Level Model – Human View



330

3.2 Figure 2B: High-Level Model – Technical View



3.3 Outline

This *high-level process model* is derived from real world election experience and is designed to accommodate all the feedback and input from the members of this committee.

For clarity, the whole process can be divided into 3 major areas, pre election, election, post election; each area involves one or more election processes. This document allocates a range of numbers for each process. One or more XML schemas are specified to support each process, this ensures consistency with all the figures and the schemas required:

- Pre election
  - Election (100)
  - Candidates (200)
  - Options (600)
  - Voters (300)
- Election
  - Voting (400)
- Post election
  - Results (500)
  - Audit
  - Analysis

Some functions belong to the whole process and not to a specific part:

- Administration Interface
- Help Desk

3.4 Process Descriptions

3.4.1 The Candidate Nomination Process

This is the process of approving nominees as eligible candidates for certain positions in an election. A candidate in this context can be a named individual or a party.

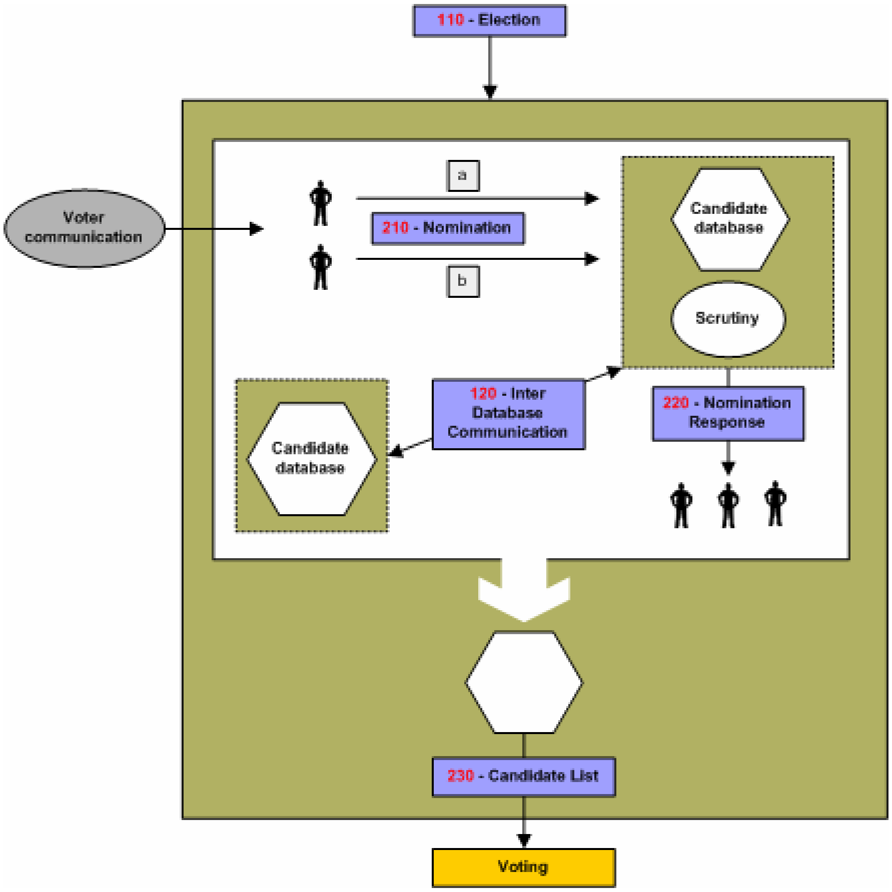


Figure 2C: The Candidate Nomination Process

Irrespective of local regulations covering the nomination process, or the form in which a candidate's nomination is to be presented, (e.g. written or verbal), the committee anticipates that the process will conform to the following format:

- Voter Communications [350-Generic] declaring the opening of nominations will be used to reach the population eligible to nominate candidates for a position x in an election y.
- Interested parties will respond in the proper way satisfying the rules of nomination for this election with the objective of becoming running candidates. The response message conforms to schema **210**.
- A nomination for an individual candidate can be achieved in one of two ways:
  - A Nominee will reply by attaching to his nomination a list of x number of endorsers with their signature.
  - Each endorser will send a message specifying Mr. X as his or her nominee for the position in question. Mr X will signal his agreement to stand.

Note that nomination and the candidate's agreement to stand might be combined in a single message or sent as two messages, each conforming to schema **210**.

The election officer(s) of this specific election will scrutinize those replies by making sure the requirements are fully met. Requirements for nomination vary from one election type to another, for example some elections require the nominee to:

- Pay fees,
- Have x number of endorsers,
- Be of a certain age,
- Be a citizen more than x number of years,
- Not stand for election in more than one contest at a time,
- Etc.

Schema **210** provides mechanisms to identify and convey scrutiny data but since the laws of nomination vary extensively between election scenarios, no specific scrutiny data is enumerated.

Schema **120** allows election officials to enquire of other jurisdictions whether a particular candidate is standing in more than one contest.

Nominees will be notified of the result of the scrutiny using a message conforming to schema **220**.

The outcome of this process is a list of accepted candidates that will be communicated using a message conforming to schema **230**. It will be used to construct the list of candidates for each contest.

3.4.2 The Options Nomination Process

This is the process of approving the options to be presented to voters in a referendum. The options can be a straight choice, e.g. YES or NO, to a single question, or can be more complex involving choices to a number of questions and/or preferences of choice.

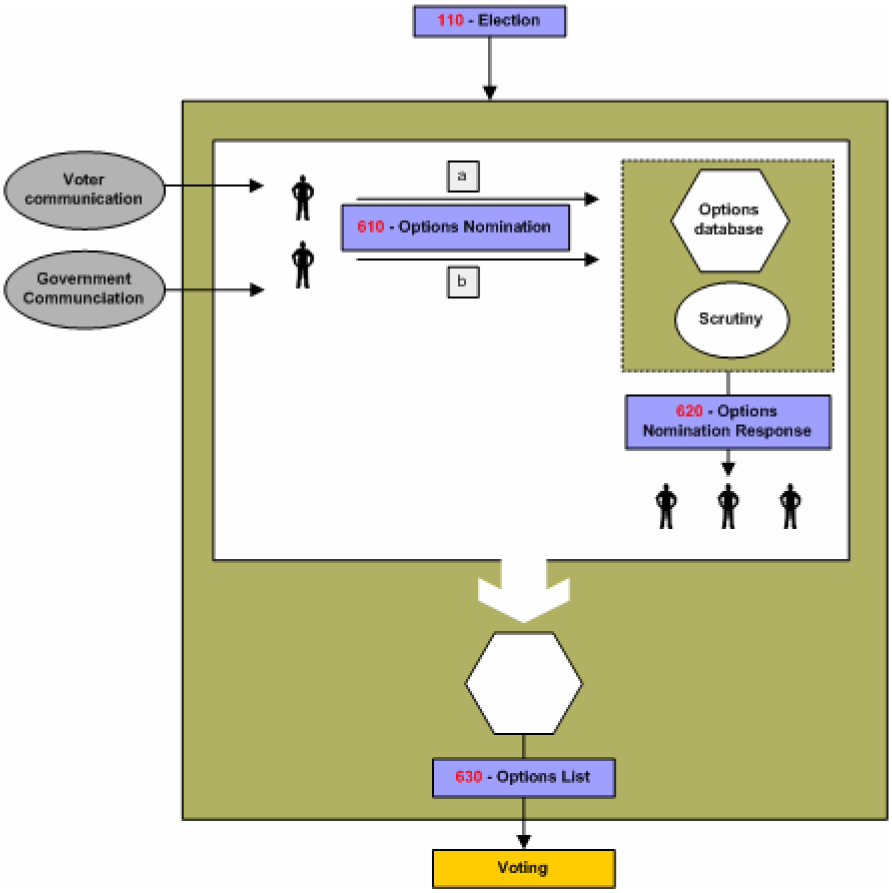


Figure 2D: Referendum Options Nomination Process

The nomination can be received in a number of ways including direct from government institutions or from citizens or businesses, and schema **610** handles the receipt of nominations.

Nominees may be notified of the result of any scrutiny of their nomination using a message conforming to schema **620**.

The outcome of this process is a list of accepted options that will be communicated using a message conforming to schema **630**. It will be used to construct the list of referendum questions for each contest.

3.4.3 The Voter Registration

This is the process of recording a person's entitlement to vote on a voter registration system. A key part of this process is the identification of the person.

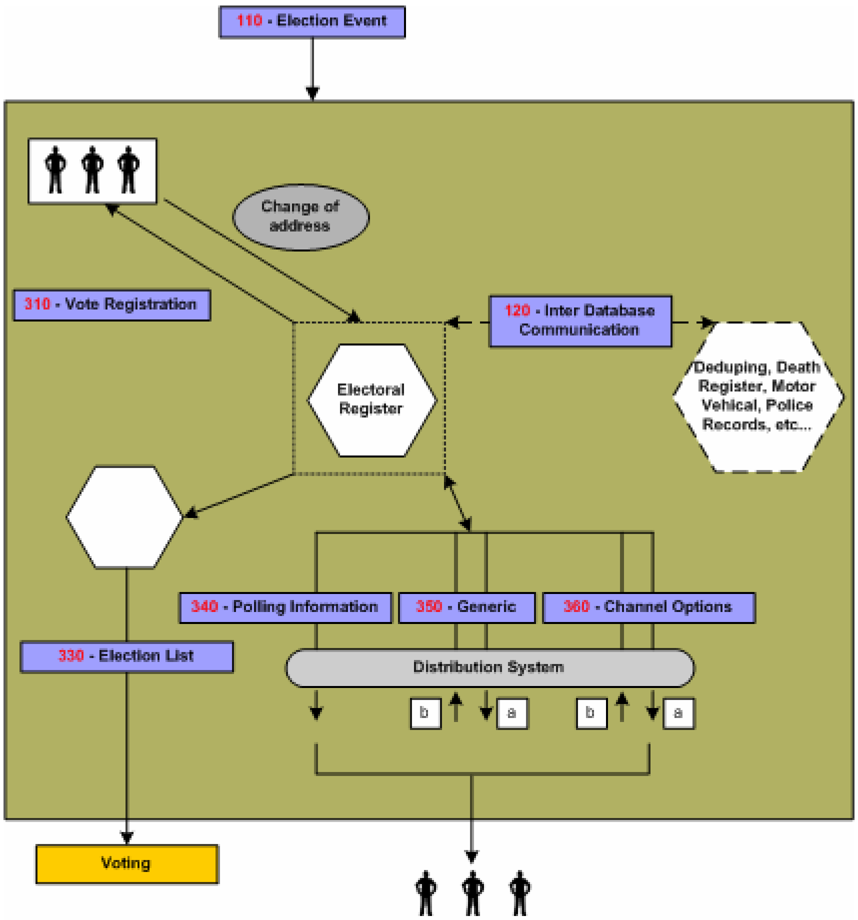


Figure 2E: Voter Registration

The centre of this process is the Electoral Roll Database or the Voters' Database. The input into this database is the outcome of communications between '*a voter*' and '*an Election Authority*'. The subject of this correspondence can vary from adding a voter to modifying a voter; deletion of a voter is considered as part of modification.

This schema of data exchange is recommended irrelevant of the method a voter uses to supply his information. For example, a voter could register online or simply by completing a voter's form and posting the signed form. In the latter case, this schema is to be followed when converting the paper form into the electoral database.

Another potential communication or exchange of data is with other databases such as those used by another election authority, government body, etc. Database exchanges will be required in some election scenarios; examples include geographical and organizational boundary changes.

At a certain date, a subset of the voters' database is fixed from which the election list is generated. Schema **330** contains some subset of the eligible voters, perhaps grouped by polling district or voting channel.

It is here that we introduce the concept of voter communications. Under this category we divided them into three possible types of communications:

- Channel options
- Polling Information

- Generic.

The communication method between the Election Authority and the voters is outside the scope of this document, so is the application itself. This document does specify the data needed to be exchanged.

3.4.4 The Voting Process

This is the process that involves the authentication of the voter and the casting of an individual vote.

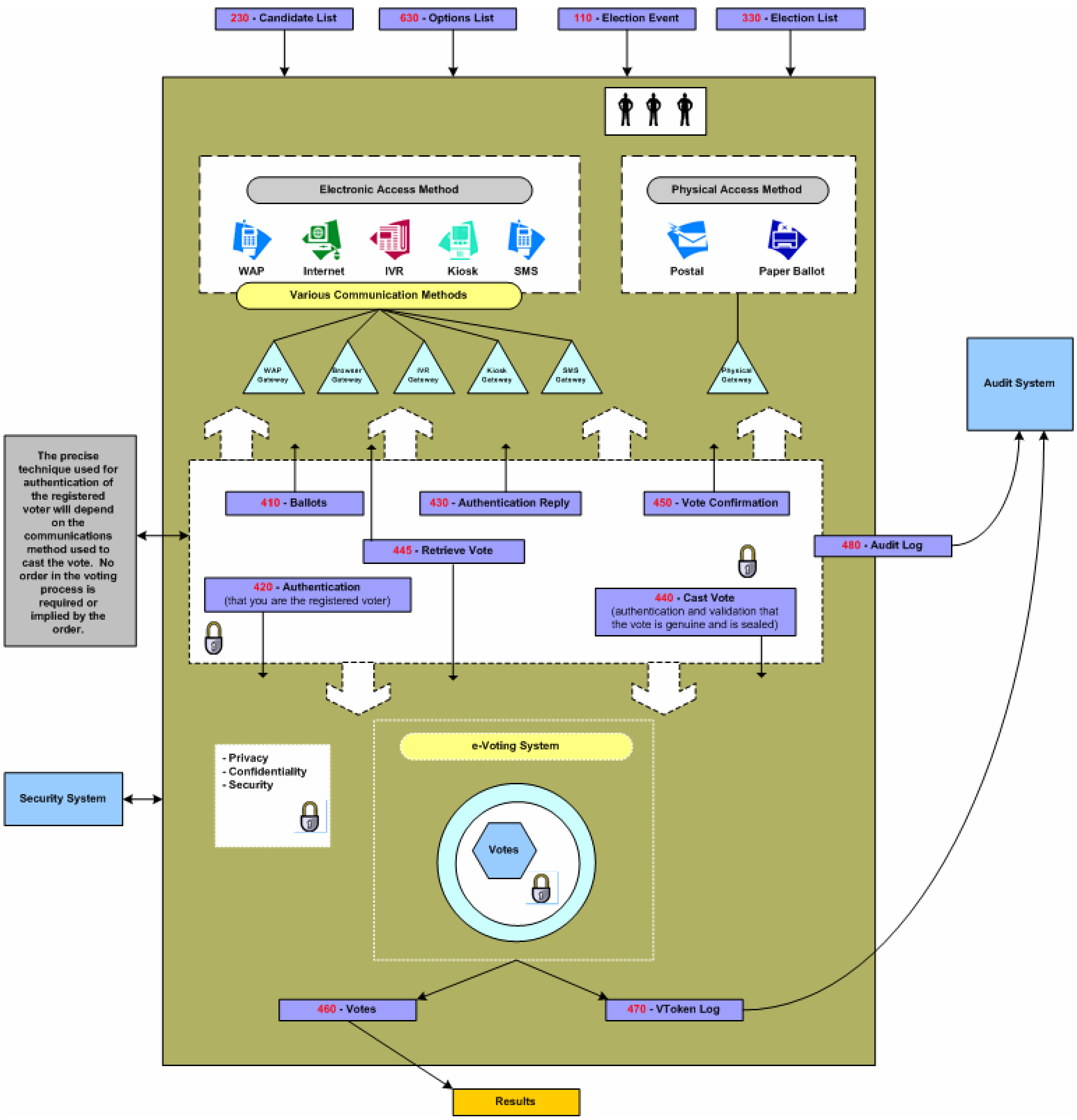


Figure 2F: The Voting Process

We assumed various systems would be involved in providing the voting process and regard each system as an independent entity.

As this figure shows, the voter will be voting using a choice of physical channels such as postal or paper ballot (the 'physical access methods'), or the voter can vote using 'electronic access methods' where he/she can utilize a number of possible e-voting channels.

Each channel may have a gateway acting as the translator between the voter terminal and the voting system. Typically, these gateways are in proprietary environments. The following schemas are to be used when interfacing to such gateways: **410**, **420**, **430**, **440** and **450**. These schemas should function irrespective of the application or the supplier's favored choice of technology.

When a pre-ballot box is required in a scenario, schema **445** can be used to retrieve and amend votes before they are counted.

Where a voter's right to vote in any particular contest needs to be determined, this is defined by the parameters of his VToken. See Section 4 for more information on security and the VToken.

In some scenarios the right to vote may need to be qualified. This may occur if the voter's right to vote is challenged or if the voter is given the temporary right to vote. In this case the vote needs to be cast by a voter with a Qualified VToken. The reason for the qualification shall always be present in a Qualified VToken and the qualification may need to be investigated before the vote is counted as legitimate. The VToken and Qualified VToken are part of schemas 420, 440, 450, 460 and 470.

To create balloting information, input data is needed about the election, the options/candidates available and the eligible voters; see schemas **230**, **110** and **120** for exchanging such information between e-systems.

3.4.5 The Vote Reporting Process

Two of the post election items are the Final Result and the Audit Report. Audit is discussed in 3.4.6.

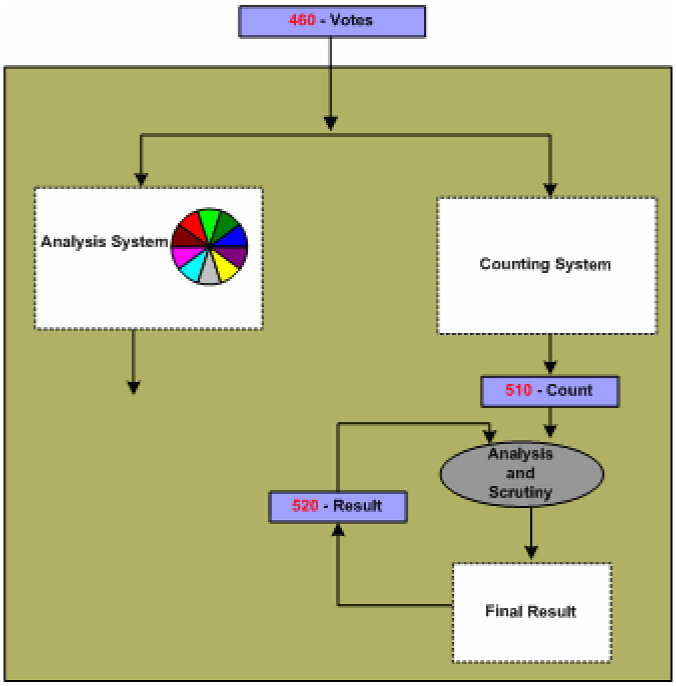


Figure 2G: The Vote Reporting Process

The voting system should communicate a bulk of data representing the votes to the counting system or the analysis system-using schema **460**. The count of these, which is the compilation of the **460,** is to be communicated by the schema **510**.

Recount can be very simply accommodated by a re-run of the schema **460**, on the same or another counting system.

Some voting methods, such as the additional member system (AMS), combine the result of one election with the votes of another to create a result. For an election run under the AMS, the results of the ‘first past the post’ (FPP) election can be communicated using a message conforming to schema **520**. This schema can only be used for communicating the results of elections using simple voting methods such as FPP, and is not intended as a general purpose results schema.

The votes schema **460** also feeds into an analysis system, which is used to provide for demographic or other types of election reports. The output of the analysis system is outside the scope of this document.

Further schemas may be developed that make use of the Votes and Count schemas. For example schemas for messages that report election results to the media.

3.4.6 The Auditing System

Audit is the process by which a legal body consisting of election officers and candidates' representatives can examine the processes used to collect and count the vote, thereby proving the authenticity of the result.

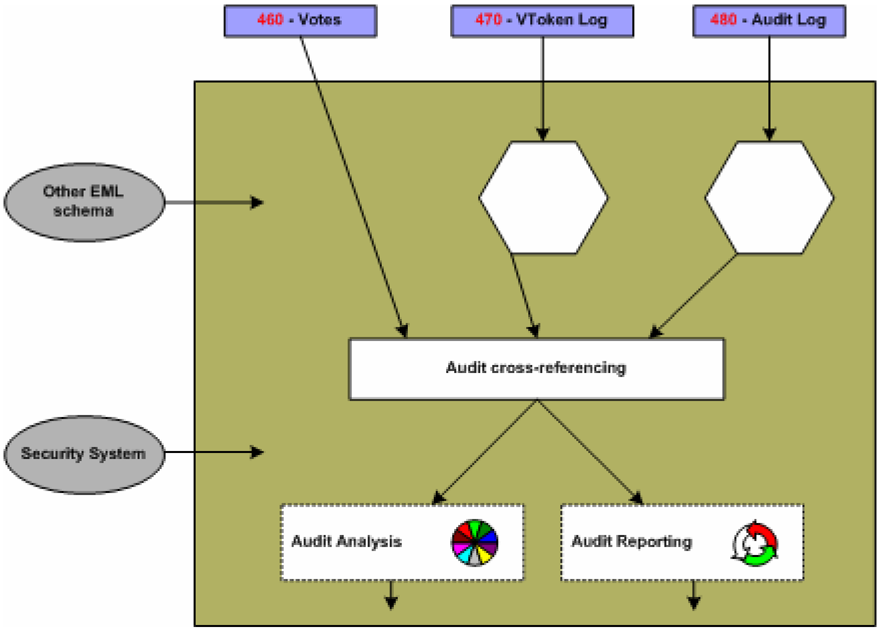


Figure 2H: Auditing System

A requirement is for the election officer to be able to account for all the ballots. A count of ballots issued should match the total ballots cast, spoiled and unused.

Schemas **460, 470**, **480** from the voting process provide input data to the audit process. Depending on the audit requirements additional data from other processes may be required. In particular, the security process may provide additional data about all the issued VTokens and Qualified VTokens (see Figure 3A: Voting system security).

The security process ensures that the right to cast a vote is dictated by the presence of a VToken, thus in order to provide accountability for all ballots as per the requirement above, reliable data from the security system is required on the total number of:

- Eligible voters
- Issued VTokens or Qualified VTokens.

The audit process can collate the total number of VTokens and Qualified VTokens provided by the security system with the total number reported by the voting system using schema **460** and **470.**

The security system and sealing mechanism should be implemented so that trust can be placed in the seal and hence the sealed data. This implies that the seal should be performed as close to the user submission of the vote as technically possible. The count of the spoiled and unspoiled votes from **460** can then be cross-checked against the count of the number of trusted seals from **480**. This correlation confirms that the total number of votes presented by the output of the e-voting system in **460** is consistent with the total number of submitted votes with seals.

The above correlation between trusted data provided by the security process and data provided by the voting process proves that no legitimate votes have been lost by the voting system. It also proves that there is consistency between the number of eligible voters and the spoiled, unspoiled and unused votes as recorded by the e-voting system.

Another requirement is for the election officer to be able to prove that voted ballots received and counted are secure from any alteration. This requirement is met because each vote cast is sealed; the seal can be verified by the audit system and to prove that no alterations have been made since the vote was sealed.

A further requirement is for the election officer to be provided with a mechanism to allow a recount when a result is contested. The number of votes from the voting system using schema **460** can be verified by correlating the total votes as calculated by the audit system (using schema **480),** with the totals from the counting system. Then either re-running the count or running the count on another implementation can verify an individual result.

There is also the requirement for the election officer to be provided with a mechanism that allows for multiple observers to witness all the voting process. How this is achieved in dependant on the implementation of the system and procedures adopted. However, the seals and channel information using schema **480** provide the ability to observe voting inputs per channel while voting is in progress without revealing the vote itself or the voter's identity. The final count of the seals can then be used to cross check the totals of the final result as described above.

The above defines some of the election data that can be verified by the audit system. However, ideally everything done by the various components of an election system should be independently verifiable. In the scope of EML this means that the audit system may need to be able to process all the standardized EML schemas. The audit system may in addition support proprietary interfaces of voting systems to enhance visibility and correctness of the election process.

3.5 Data Requirements

The data used in all the above processes are defined in 'EML v4.0 Data Dictionary'.

4 Security Considerations

This section presents a general discussion of many of the security considerations commonly found in many election environments. As presented previously, these standards apply at EML interface points and define data security mechanisms at such interface points. This document is not intended to provide a complete description, nor a set of requirements for, secure election systems. In fact, the data security mechanisms described in this document are all optional, enabling compliance with these standards without regard for system security at all.

This discussion is included here simply to show how the information passed through the various interfaces described in these standards could be secured and used to help meet some of the requirements commonly found in some elections scenarios.

4.1 Basic security requirements

The security governing an election starts before the actual vote casting. It is not only a matter of securing the location where the votes are stored. An intensive analysis into security related concerns and possible threats that could in one way or another affect the election event resulted in the following:

Security considerations of e-voting systems include:

- Authentication
- Privacy/Confidentiality
- Integrity
- Non-repudiation

4.1.1 Authentication

This is checking the truth of a claim of identity or right to vote. It aims to answer questions such as “Who are you and do you have the right to vote?”

There are two aspects of authentication in e-voting systems:

- Checking a claim of identity
- Checking a right to vote.

In some e-voting scenarios the two aspects of authentication, checking a claim of identity and checking a right to vote, may be closely linked. Having checked the identity of the voter, a list of authorized voters may be used to check the right to vote.

In other scenarios the voter’s identity must remain private and must not be revealed by a ballot. In which case some systems may provide a clear separation between checking of the claim of identity, which may be done some time before the ballot takes place, from checking the right to vote at the time of the vote is cast. Alternatively, other mechanism may be used to ensure the privacy of the voter’s identity on cast votes (i.e. by anonymizing the ballot).

In the physical voting world, authentication of identity is made by using verifiable characteristics of the voter like handwritten signatures, address, etc and physical evidence like physical IDs; driver’s license, employee ID, Passport etc, all of this can be termed a physical **credential.** This is often done at the time an electoral register is set up, which can be well before the actual ballot takes place.

Checking the authenticity of the right to vote may be performed at various stages in the process. Initial authenticity checks may be done related to the voter’s identity during registration.

Where an election scenario demands anonymity of the voter and privacy of the voter’s ballot, the identity of the voter and the cast votes must be separated at some time within the voting process.

This can be done in several ways by a voting system including, but not restricted to, the following options:

Authentication of the right to vote by itself does not reveal a voter's identity, but does verify he has a legitimate right to vote (e.g. the VToken data provides authentication of the right to vote but has anonymous properties as to the identification of the person voting).

An voter's identity and the right to vote are both validated (i.e. the VToken data has both 'voter identification' and 'right to vote' authentication properties) and then the cast votes are clearly separated from the identity of the voter (i.e. the voters identification occurs before the ballot is 'anonymized')

In all cases any verification of the authenticity that takes place after the voter has indicated his/her choices must preserve the privacy of those choices according to the laws of the jurisdiction and the election rules.

Finally, when counting and auditing votes it is necessary to be able to check that the votes were placed by those whose right to vote has been authenticated.

Public democratic elections in particular will place specific demands on the trust and quality of the authentication data. Because of this and because different implementations will use different mechanisms to provide the voter credential, precise mechanisms are outside the scope of this document.

4.1.2 Privacy/Confidentiality

This is concerned with ensuring information about voters and how votes are cast is not revealed except as necessary to count and audit the votes. In most cases, it must not be possible to find out how a particular voter voted. Also, before an election is completed, it should not be possible to obtain a count of how votes are being cast.

Where the user is remote from the voting system then there is a danger of voting information being revealed to someone listening in to the communications. This is commonly stopped by encrypting data as it passes over the communications network.

The other major threat to the confidentiality of votes is within the system that is collecting votes. It should not be possible for malicious software that can collect votes to infiltrate the voting system. Risks of malicious software may be reduced by physical controls, careful audit of the system operation and other means of protecting the voting systems.

Furthermore, the results of voting should not be accessible until the election is complete. Potential approaches to meeting this goal might include access control mechanisms, very careful procedural control over the voting system, and various methods of protecting the election data using encryption techniques.

4.1.3 Integrity

This is concerned with ensuring that ballot options and votes are correct and unaltered. Having established the choices within a particular ballot and the voter community to which these choices apply, the correct ballot information must be presented to each voter. Also, when a vote is placed it is important that the vote is kept correctly until required for counting and auditing purposes.

Using authentication check codes on information being sent to and from a remote voter's terminal over a communications network generally protects against attacks on the integrity of ballot information and votes. Integrity of the ballot and voting information held within computer systems may be protected to a degree by physical controls and careful audit of the system operation. However, much greater confidence in the integrity of voting information can be achieved by using digital signatures or some similar cryptographic protection to "seal" the data.

The fundamental challenge to be met is one of maintaining voter privacy and maintaining the integrity of the ballot.

4.1.4 Non-repudiation

Non-repudiation is a derivative of the identification problem. Identification in e-voting requires that the system provide some level of assurance that the persons representing themselves as valid participants (voters, election workers, etc.) are, in fact, who they claim to be. Non-repudiation requires that the system provides some level of assurance that the identified participant is not able to successfully assert that the actions attributed to them via the identification mechanism were, in fact, performed by someone else. The two requirements are related in that a system with a perfect identification mechanism and undisputable proof of all actions would leave no room for successful repudiation claims.

Non-repudiation also requires that the system provide assurance that data or actions properly associated with an identified participant can be shown to have remained unaltered once submitted or performed. For example, approved candidate lists should be verified as having come from an authorized election worker, and voted ballots from a valid voter. In both cases the system should also provide a way to ensure that the data has remained unchanged since the participant prepared it.

Non-repudiation is not only a technical quality of the system. It also requires a certain amount of pure policy, depending on the technology selected. For example, in a digital signature environment, signed data can be very reliably attributed to the holder of the private key(s), and can be shown to be subsequently unmodified. The policy behind the acceptance of these properties, however, must be very clear about the responsibilities of the private key holders and the required procedures for reporting lost or stolen private keys. Further, and especially in "mixed-mode" elections (where voters can chose between multiple methods of voting), it may often be desirable to introduce trusted time stamps into the election data stream, which could be used to help determine acceptance criteria between ballots, or help resolve issues with respect to the relative occurrence of particular events (e.g. ballot cast and lost keys reported). The presence of the time information itself would not necessarily enable automatic resolution of these types of issues, but by providing a clear ordering of events could provide data that can be fed into decisions to be made according to established election policy.

4.2 Terms

The following security terms are used in this document:

- **Identity Authentication**: the means by which a voter registration system checks the validity of the claimed identity.
- **Right to vote authentication:** the means by which the voting system checks the validity of a voter's right to vote.
- **VToken**: the means by which a voter proves to an e-voting system that he/she has the right to vote in a contest.
- **VToken Qualified**: the means by which a VToken can be qualified. The reason for the qualification is always appended to a VToken that is qualified. For example, a qualified VToken may be issued to a challenged voter.
- **Vote sealing**: the means by which the integrity of voting data (ballot choices, vote cast against a given VToken) can be protected (e.g. using a digital signature or other authentication code) so that it can be proved that a voter's authentication and one or more votes are related.

4.3 Specific Security Requirements

Electronic voting systems have some very specific security requirements that include:

- Only legitimate voters are allowed to vote (i.e. voters must be authenticated as having the right to cast a vote)
- Only one set of choices is allowed per voter, per contest
- The vote cannot be altered from the voter's intention
- The vote may not be observed until the proper time
- The voting system must be accountable and auditable
- Information used to authenticate the voter or his/her right to vote should be protected against misuse (e.g. passwords should be protected from copying)
- Voter privacy must be maintained according to the laws of the election jurisdiction. (Legal requirements of public elections in various countries conflict. Some countries require that the vote cannot be tracked back to the voter's identity, while others mandate that it must be possible to track every vote to a legitimate voter's identity)
- The casting options available to the voter must be genuine
- Proof that all genuine votes have been accurately counted.

There are some specific complications that arise with respect to security and electronic voting that include:

- Several technologies may be employed in the voting environment
- The voting environment may be made up of systems from multiple vendors
- A voter may have the option to vote through alternative delivery channels (i.e. physically presenting themselves at a poling station, by post, by electronic means)
- The voting systems need to be able to meet various national legal requirements and local voting rules for both private and public elections
- Need to verify that all votes are recorded properly without having access to the original input
- The mechanism used for voter authentication may vary depending on legal requirements of the contest, the voter registration and the e-voting systems for private and public elections
- The user may be voting from an insecure environment (e.g. a PC with no anti-virus checking or user access controls).

Objectives of this security architecture include:

- Be open
- Not to restrict the authentication mechanisms provided by e-voting systems
- Specify the security characteristic required of an implementation, allowing for freedom in its precise implementation.

4.4 Security Architecture

The architecture proposed here is designed to meet the security requirements and objectives detailed above, allowing for the security complications of e-voting systems listed.

The architecture is illustrated in figure 3a below, and consists of distinct areas:

- Voter identification and registration
- Right to vote authentication
- Protecting exchanges with remote voters
- Validating Right to Vote and contest vote sealing

- Vote confidentiality.
- Candidate list Integrity
- Vote counting accuracy
- Voting system security controls.

4.4.1 Voter identification and registration

The Voter identification and registration is used to identify an entity (e.g. person) for the purpose of registering the person has a right to vote in one or more contests, thus identifying legitimate voters. The security characteristics for voter identification are to be able to authenticate the identity of the legal person allowed to vote in a contest and to authenticate each person's voting rights. The precise method of voter identification is not defined here, as it will be specific to particular voting environments, and designed to meet specific legal requirements, private or public election and contest rules. The voter registration system may interact with the e-voting system and other systems to define how to authenticate a voter for a particular contest.

Voter identification and registration ensures that only legitimate voters are allowed to register for voting. Successful voter registration will eventually result in legitimate voters being given a means of proving their right to vote to the voting system in a contest. Depending on national requirements or specific voting rules/bylaws the voter may or may not need to be anonymous. If the voter is to be anonymous, then there must not be a way of identifying a person by the means used to authenticate a right to vote to the e-voting system. Right to vote authentication is the means of ensuring a person has the right to cast a vote, but it is not the identification of the person.

4.4.2 Right to vote Authentication

Proof of the right to vote is done by means of the VToken, which is generated for the purpose of authentication that the voter has a legitimate right to vote in a particular contest.

The security characteristic of the VToken and hence its precise contents may vary depend on the precise requirements of a contest, the supplier of the voter registration system, the e-voting system, the voting channel or other parts of the electoral environment. Thus, the content of the VToken will vary to accommodate a range of authentication mechanisms that could be used, including; pin and password, encoded or cryptographic based password, hardware tokens, digital signatures, etc.

The contents of the VToken may also depend on the requirements of a particular contest, which may mandate a particular method be used to identify the person and the voter. For example, if a country has a national identity card system, it could be used for the dual purpose of identifying the person and providing proof that the person is entitled to vote, provided the legal system (or the voting rules of a private election) allow a personal identity to be associated with a vote. However, this would not work for countries or private voting scenarios that require the voter to be anonymous. For such a contest the mechanism used to identify that a person has the right to cast a vote must not reveal the identity of the actual person, thus under such voting rules voter identity authentication and right to vote authentication do not use the same information or semantics.

The security characteristic required of the VToken may also vary depending on legal requirements of a country or electoral rules used in a particular contest. Also, the threats to misuse of VTokens will depend to a large degree on the voting channels used (e.g. physical presence at voting station, Internet, mobile phone). Bearing this in mind the XML schema of the VToken components must allow for various data types of authentication information to be contained within it.

It must be possible to prove that a VToken is associated with a vote cast and the rules of the contest are followed, such as only one vote being allowed per voter, per contest. Thus providing proof /non-repudiation that all votes were genuine, they were cast in accordance with the rules of the contest, that no vote has been altered in any way and that all the votes counted in a contest were valid when audited.

Depending on the legal requirements of a country or electoral rules a voter may be challenged as to the right to vote, or may be given a temporary right to vote. In such cases the VToken may need to be qualified with a reason. In this document this is called a VToken Qualified. Before a vote is considered legitimate and counted the reason for the qualification must have been suitably scrutinized, which could be done by the voting officials.

4.4.3 Protecting exchanges with remote voters

The VToken may be generated as part of the registration system, the e-voting system, or as interaction between various components of a voting environment, as illustrate in Figure 3a. The VToken will need to be provided securely to the voter so that this can be used to prove the right to vote.

The exchange of information when casting a vote must be protected by secure channels to ensure the confidentiality, integrity of voting data (VToken(s) and vote(s) cast) and that this is correctly delivered to the authenticated e-voting system. If the channel isn't inherently secure then this will require additional protection using other mechanisms. Possible mechanisms might include: a postal system with sealed envelopes, dedicated phone channel, secure e-mail, secure internet link (SSL), peer to peer server/client authentication and a seal.

Wherever technically possible the exchange of information should be secured and integrity guaranteed even if non-secure communications channels are used.

4.4.4 Validating Right to Vote and contest vote sealing

When a vote is cast, to ensure that it cannot be altered from the voter's intention, all the information used to authenticate the right to vote and define the vote cast must be sealed to ensure the integrity and non-repudiability of the vote. This seal may be implemented using several mechanisms ranging from digital signatures (XML and CMS), cryptographic seals, trusted timestamps and other undefined mechanisms. The seal provides the following security functions:

- The vote cannot be altered from the voter's intention
- The voting system is accountable and auditable.

The right to vote may be validated at the time the vote was cast. If votes are not checked for validity before sealing then the right to vote must be validated at the time that votes are subsequently counted. Also when counting, or otherwise checking votes, the validity of the seal must be checked.

If votes are sealed and recorded without being checked for validity at the time they were cast, then the time that the vote was cast must be included in the seal, so that they may be checked for validity before they are counted.

In some election scenarios it is required to audit a vote cast to a particular voter, in this case a record is also needed of the allocation of a VToken to a voter's identity. Such systems also provide non-repudiation of the voter's actions. In such cases a voter cannot claim to have not voted or to have voted a different way, or that his vote was not counted. In many election scenarios where this type of auditing is required, it must not be easy to associate a VToken to the Voter's identity, therefore this type of records must be under strict control and protected by security mechanism and procedures, such as; encryption, key escrow and security operating procedures.

4.4.5 Vote confidentiality

All cast votes must not be observed until the proper time, this requires confidentiality of the vote over the voting period, how this is achieved will vary from e-voting system to e-voting system.

Mechanism of vote confidentiality, range from trust in the e-voting systems internal security functions (processes and mechanisms) to encryption of the data, with key escrow tools.

4.4.6 Candidate list integrity

To ensure that the voter is present and that the candidate list is genuine, there must be a secure channel between the voting system and the person voting or the data must be sealed. The approach selected must ensure that there is no man-in-the-middle that can change a vote from what the voter intended. There are various ways this requirement can be met, ranging from the candidate list having unpredictable characteristics with a trusted path to convey that information to the voter, to trust placed in the complete ballot/vote delivery channel.

As an example, there may be a secure path to convey the VToken to the person entitled to vote, a way of ensuring that a voter is always presented with a genuine list of candidates might be to encode the candidate list as part of a sealed VToken.

In summary, there must be a way of ensuring the validity of the ballot options and voter selection.

4.4.7 Vote counting accuracy

Audit of the system must be able to prove that all vote casts were genuine and that all genuine votes were included within the vote count. Voters may need to be able to exercise that proof should they so desire. Thus auditing needs data that has non-repudiation characteristics, such as the VToken/vote sealing, see schema **470** and **480**.

4.4.8 Voting System Security

The overall operation of the voting systems and its physical environment must be secure. Appropriate procedural, physical and computing system controls must be in place to ensure that risks to the e-voting systems are met. There must be a documented security policy based upon a risk analysis, which identifies the security objectives and necessary security controls.

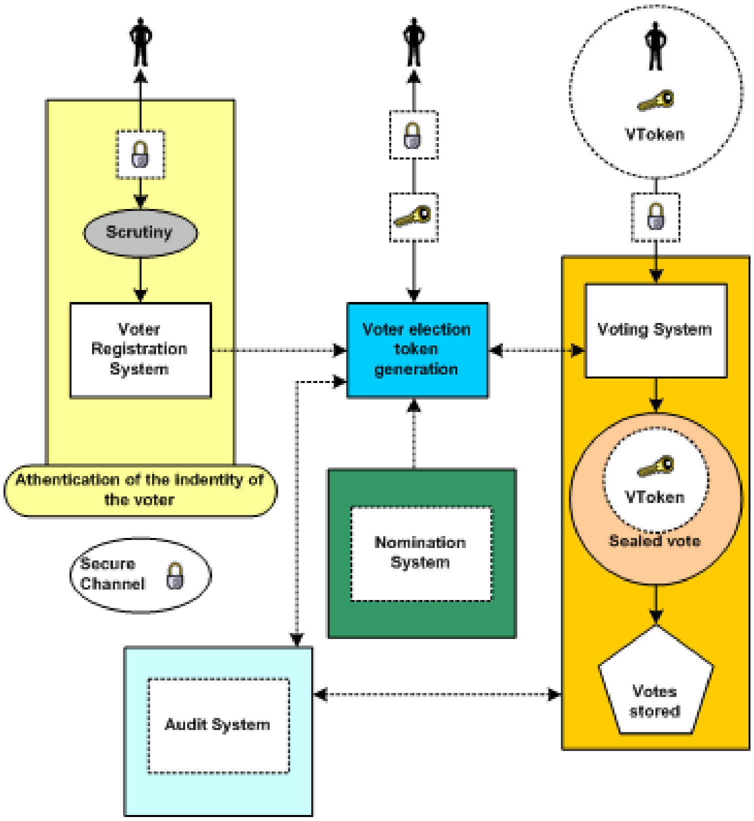


Figure 3A: Voting system security

4.5 Remote voting security concerns

Many new election systems are currently under evaluation. These systems tend to offer deployment options in which the communication between the voter and the election officials is carried out in an environment that is not completely under the control and monitoring of the election officials and/or election observers (e.g., the Internet, private network, telephones, cable TV networks, etc.). In these ’remote‘ or ’unattended‘ environments, several particular security concerns and questions like:

- How do I know that that the candidate information I am being presented with is the correct information?
- How do I know that my vote will be recorded properly?
- How do I know there isn't a man-in-the-middle who is going to alter my vote when I place it?
- How do I know that it is the genuine e-voting server I'm connected to that will record my vote rather than one impersonating it that's just going to throw my vote away?
- How do I know that some component of the system does not have malicious software which will attempt to alter the ballot choices as represented to me or alter my election?

The type and importance of a particular contest will have an effect on whether the above concerns exist and whether they do, or do not, represent a tangible threat to the voting process and its outcome. The table listed at Appendix B shows the concerns that have been identified as possibilities for one such remote or unattended environment (the Internet) that could be used in public election voting scenarios. The table shows how the concerns can be translated to technical threats and characterizes security services that may be used to counter such threats. Many of the items are not unique to the Internet, and can serve as a useful reference or starting point in developing similar threat analysis for other digital and/or unattended voting environments. How the security services are implemented in any particular environment or deployment is outside the scope of this document allowing freedom to the system providers.

5 Schema Outline

5.1 Structure

The Election Markup Language specification defines a vocabulary (the EML core) and message syntax (the individual message schemas). Thus most voting-related terms are defined as elements in the core with the message schemas referencing these definitions. The core also contains data type definitions so that types can be re-used with different names (for example, there is a common type to allow messages in different channel formats), or used as bases for deriving new definitions.

In some cases, two or more message schemas have large parts in common. For example, a voter authentication response message can contain a ballot that is almost identical to that used in the ballot message. When this occurs, the relevant declarations are included in a file whose file name includes the word ‘include’ and the number of the schemas in which it is used.

There is a third category of schema document within EML - the EML externals. This document contains definitions that are expected to be changed on a national basis. Currently this comprises the name and address elements, which are based on the OASIS Extensible Name and Address Language [1], but may be replaced by national standards such as those contained in the UK Government Address & Personal Details schemas [2]. Such changes can be made by replacing just this single file.

As well as these, several external schemas are used. The W3C has defined a standard XML signature [5]. OASIS has defined schemas for the extensible Name and Address Language (xNAL) [1]. As part of the definition of EML, the committee has defined a schema for the Timestamp used within EML. All these schemas use their appropriate namespaces, and are accessed using `xs:import` directives.

Each message (or message group) type is specified within a separate schema document. All messages use the `EML` element from the election core as their document element. Elements declared in the individual schema documents are used as descendents of the `EML` element.

5.2 IDs

XML elements may have an identifier which is represented as an `Id` attribute.

Each `schema` element has an `Id` attribute that relates to the message numbering scheme. Each message also carries this number.

Some items will have identifiers related to the voting process. For example, a voter might be associated with an electoral roll number or a reference on a company share register. These identifiers are coded as elements.

Other identifiers exist purely because of the various channels that can be used for voting (e.g. Internet, phone, postal, etc). In this case the identifiers are likely to be system generated and are coded as attributes.

5.3 Displaying Messages

Many e-voting messages are intended for some form of presentation to a user, be it through a browser, a mobile device, a telephone or another mechanism. These messages need to combine highly structured information (such as a list of the names of candidates in an election) with more loosely structured, often channel-dependent information (such as voting instructions).

Such messages start with one or more `Display` elements, such as:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<EML
  Id="410"
  SchemaVersion="0.1"
  xml:lang="en"
  xmlns="http://www.govtalk.gov.uk/temp/voting"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.govtalk.gov.uk/temp/voting
                      ..\schemas\ballot.xs">
  <Display Format="html">
    <Stylesheet Type="text/xsl">../stylesheets/ballot.xsl</Stylesheet>
    <Stylesheet Type="text/css">../stylesheets/eml.css</Stylesheet>
  </Display>
  <Ballots>
    ...
```

This example shows a `Display` element providing information to the receiving application about an XSL stylesheet which transforms the message into HTML for displaying the ballot in a Web browser. In the `Display` element in the example, the XSLT stylesheet reference is followed by a CSS stylesheet reference. In this case, the XSLT stylesheet referenced will pick up the reference to the CSS stylesheet as it transforms the message, and generate appropriate output to enable the displaying browser to apply that cascading stylesheet to the resulting HTML.

Not all information in a message will need to be displayed, and the creator of the message might have views on the order of display of the information. To allow stylesheets to remain generic, many elements in the schemas can have a `DisplayOrder` attribute. The values of these attributes determine the layout of the display (or the spoken voice if transforming to, for example, VoiceXML), even when using a generic stylesheet.

When displaying messages in HTML, the expectation is that generic stylesheets will cover most cases, with the stylesheet output being embedded in a web page generated from an application-specific template. Similarly, voice applications might have specific welcome and sign-off messages, while using a generic stylesheet to provide the bulk of the variable data.

The three screen shots show the effect of using the same XSL stylesheet on the ballots for various voting scenarios. In the first picture, clicking on the name of a candidate has popped up a window with additional details.

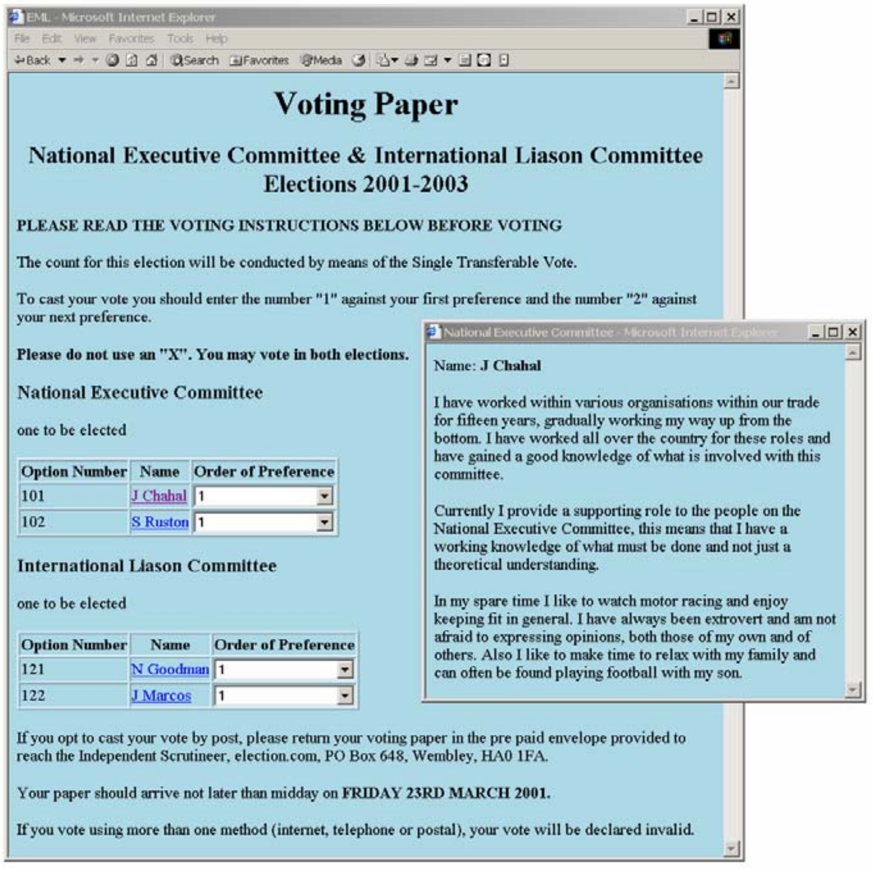


Figure 3A: Screen shot of the ballot for scenario 1

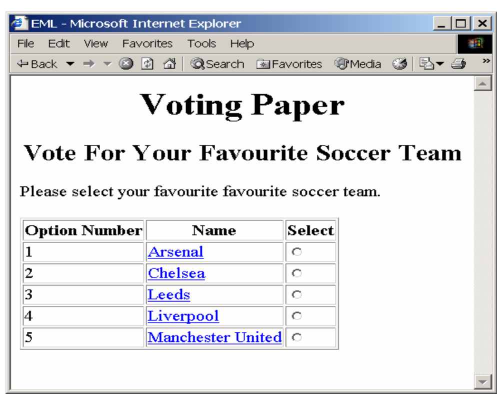


Figure 3B: Screen shot of the ballot for scenario 2

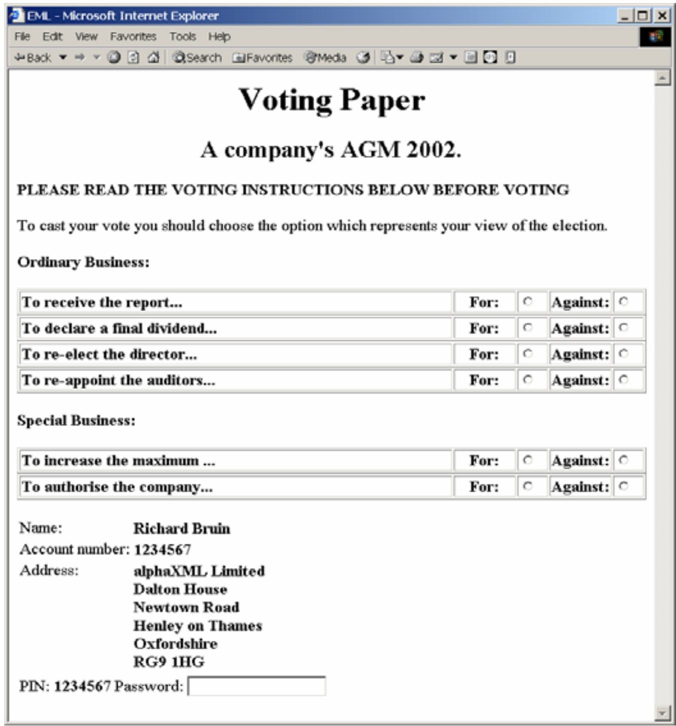


Figure 3C: Screen shot of the ballot for scenario 3

6 Schema Descriptions

Details on the description of schemas used in EML v4.0 can be found within the document ‘EML v4.0d Schema Descriptions’.

Appendix A: Internet Voting Security Concerns

Concerns raised on Internet voting		Resulting Technical Threats	Possible generic security service countermeasure
1.	Impersonation of the right to vote. The concern here is that a person attempts to impersonate to be a legitimate voter when he/she is not. The initial task of verifying that a person has the right to vote must be part of the voter registration process. A person must not be given the right to vote until after proper due diligence has been undertaken during voter registration that the person has a right to vote in a contest.	Inadequate, incorrect or improper identification of person during registration of voters	Trusted voter identification and registration using: Security Procedures. Best Practices. Secure communications channels. The voter registration authority must follow standard Security Operating Procedures (SOPs) which ensure due diligence has been done.
		Inadequate privacy of the exchange between the person and the electoral system during voter registration	Channel between voter and registration system must provide: Connection Confidentiality Connection Integrity
2	Voter is not presented with correct ballot information due to incorrect candidate identification.	Incorrect identification during candidate registration.	Trusted candidate identification and registration are needed using: - Security Procedures. - Best Practices. - Secure communications channels. - Authentication and identification of candidates The candidate registration must follow standard Security Operating Procedures (SOPs) which ensure due diligence has been done.
3	Registration system impersonation	Inadequate authentication of registration system	Channels to and from the registration system must provide point to point authentication.

4	Impersonation of a legitimate registered voter	Incorrect authentication at the time of casting vote.	Trusted voter authentication (i.e. the right to cast a vote in this contest)
		Inadequate privacy of the exchange between the voter and the electoral system when vote is cast.	Channel to provide: - Connection Confidentiality - Connection Integrity - Between voter and e-voting system
5	Obtaining the right to vote illegally from a legitimate voter. This may be by intimidation, theft or by any other means by which voting right has been obtained illegally. For example, by Stealing a voting card from a legitimate voter.	Stealing the voter's voting card (e.g. the VToken data).	Some secret data only known to the voter's is required to be presented at the time of casting a vote. Before a vote is counted as a valid vote proof must be provided that the voter's secret data was present at the time of casting the vote.
		Any means of getting a legitimate voter to reveal his VToken data.	
6	Voting system impersonation	Inadequate authentication of registration system	Channel to provide: Point to point authentication
		Inadequate authentication of voting casting point (e.g. polling station/ballot box)	Channel to provide: Point to point authentication
7	Voter is not presented with correct ballot information	Inadequate integrity of the ballot information Given to the user Held in the voting system	Trusted path to voter on ballot options
			Integrity of the ballot information
			Integrity of cast votes
		The casting options available to the voter are not genuine	Trusted path between voter and vote recording
		Trojan horse, man in the middle attack	Trusted path to voter on ballot options
8	How do I know the voting system records votes properly	Integrity of the voting system	Non-repudiation of the vote
			Non-repudiation the vote was cast by a genuine voter
			Audit of voting system
			Connection confidentiality
		Insecure channel between the voter and the vote casting point	Connection Integrity
			Connection Confidently

		Voter's intent is recorded accurately	Trusted path between voter and vote recording
			Non-repudiation of the vote recorded
		Proof that a genuine vote has been accurately counted	Audit
9	How can I be sure the voting system will not disclose whom I have voted for	Voter's identification is revealed	Voter's identification is anonymous
			Vote confidentiality
10	How can it be sure that my vote has been recorded	Loss of vote	Proof of vote submission
11	How can I be sure there is no man-in-the- middle that can alter my ballot	Vulnerable client environment; Trojan horses Virus	Physical security
			Procedural security
			Unpredictable Coded voting information
		Interception of communication	Integrity of communications channel between client and server system
12	All votes counted must be have been cast by a legitimate voter	Voter impersonation	Voter authentication
		Audit facility fails to provide adequate proof	Non-repudiation of the vote record
			Non-repudiation that legitimate voters have cast all votes.
		Breaking the vote counting mechanisms	Independent audit
13	Only one vote is allowed per voter, per contest	Voter impersonation at registration	User registration security Procedures Voter Identification
		Multiple registration applications	
		Multiple allocation of voters credentials	Voter authentication
14	The vote cannot be altered from the voter's intention	Vulnerable client environment; Trojan horses Virus	Trusted path from voter's intent to vote record
			Vote integrity
			Vote non-repudiation
15	The vote may not be observed until the proper time	Votes may be observed before the end of the contest	Voter confidentiality
16	The voting system must be accountable and auditable		Non-repudiation of vote data.
			Audit tools

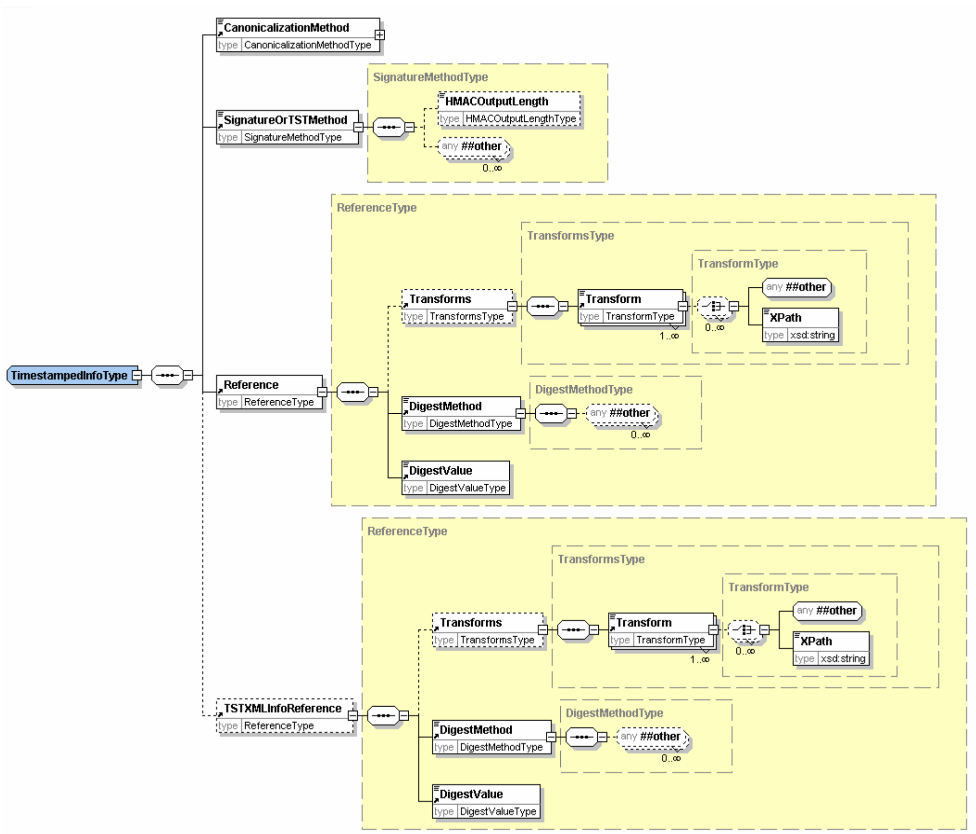
17	Identification and authentication information to and from the voter must be privacy protected	Loss of privacy	Channel to provide: Connection Confidentiality
18	The voter’s actual identity may need to be anonymous	Voter’s identification is revealed Denial of service attack	Voter’s identification is anonymous
19	Denied access to electronic voting station		This needs to be counted by engineering the system to provide survivability when under denial of service attack.

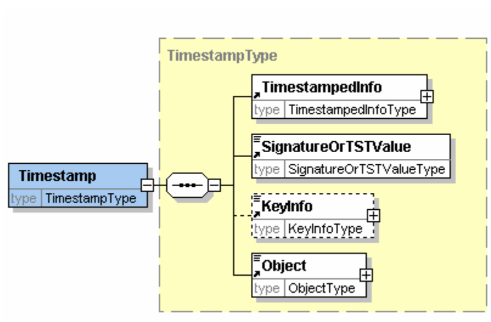
Appendix B: The Timestamp Schema

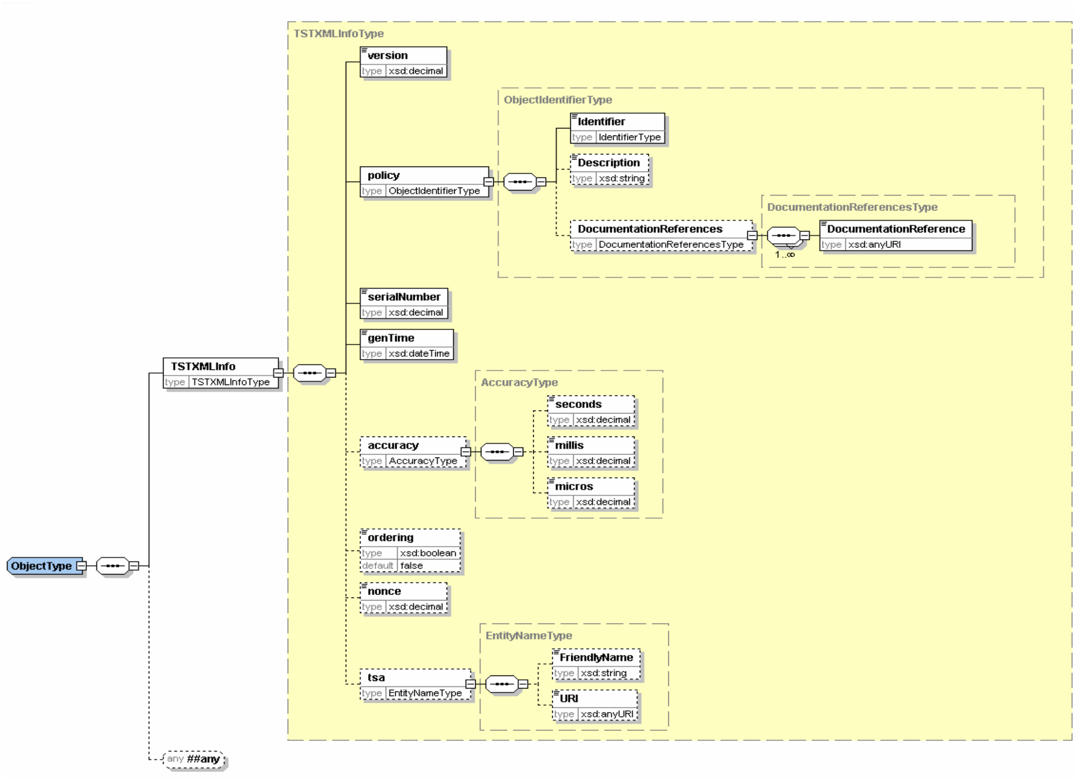
Although used as part of EML, this schema has been put in a separate namespace as it is not an integral part of the language.

A time-stamp binds a date and time to the sealed data. The time-stamp seal also protects the integrity of the data.

The structure of the time-stamp is similar to the structure of an XML Signature. The structure of the `Timestamp` element is shown here, followed by the detail of two of the four data types that are used to define its child elements.







The timestamp structure may be used in one of two ways either:

Using Internet RFC 3161 binary encoded time-stamp token with the time-stamp information repeated in XML,

Using a pure XML encoded time-stamp.

In the case of the RFC 3161 based time-stamp, the Timestamp structure is used as follows:

- within TimestampedInfo:
- `TSTOrSignatureMethod` identifies RFC 3161.
- `Reference` contains the URI reference of the voting data being time-stamped. The `DigestValue` sub element contains the digest of the voting data being time-stamped.
- `TSTXMLInfoReference` is not present in this case.
- `SignatureOrTSTValue` holds the RFC 3161 time-stamp token applied to the digest of `TimestampedInfo`. The `TimestampedInfo` is transformed to a canonical form using the method identified in `CanonicalizationMethod` before the digest algorithm is applied.
- `KeyInfo` contains any relevant certificate or key information.

`Object` contains the `TSTXMLInfo` element which is a copy of the information in `SignatureOrTSTValue` converted from RFC 3161 to XML encoding. The `TSTXMLInfo` element contains:

- version of time-stamp token format. This would be set to version 1
- the time-stamping policy applied by the authority issuing the time-stamp,
- the time-stamp token serial number,
- the time that the token was issued, the contents of this element indicate the time of the timestamp.

- optionally an indication as to whether the time-stamps are always issued in the order that requests are received
- optionally a nonce[1] given in the request for the time-stamp token,
- optionally the identity of the time-stamping authority

In the case of a pure XML encoded time-stamp, the Timestamp structure is used as follows:

- within TimestampedInfo,
- `TSTOrSignatureMethod` identifies the algorithm used to create the signature value.
- `Reference` contains the URI reference of the voting data being time-stamped. The `DigestValue` sub element contains the digest of the voting data being time-stamped.
- `TSTXMLInfoReference` must be present, and contains the URI reference of `TSTXMLInfo` as contained within the `Object` element. The `DigestValue` sub element contains the digest of the `TSTXMLInfo`.
- `SignatureOrTSTValue` contains the signature value calculated over the `TimestampedInfo` using the signature algorithm identified in `TSTOrSignatureMethod` having been transformed to a canonical form using the method identified in `CanonicalizationMethod`. This signature is created by the time-stamping authority.
- `KeyInfo` contains any relevant certificate or key information.

`Object` contains the XML encoded time-stamp information in an TSTXMLInfo element. The contents of TSTXMLInfo is the simular as for the case described above. However, in this case the information is directly signed by the time-stamping authority. The `TSTXMLInfo` element contains:

- version of time-stamp token format: This would be set to version 2
- the time-stamping policy applied by the authority issuing the time-stamp,
- the time-stamp token serial number,
- the time that the token was issued, this is the time of the timestamp.
- optionally an indication as to whether the time-stamps are always issued in the order that requests were received
- optionally a nonce given in the request for the time-stamp token,
- optionally the identity of the time-stamping authority.

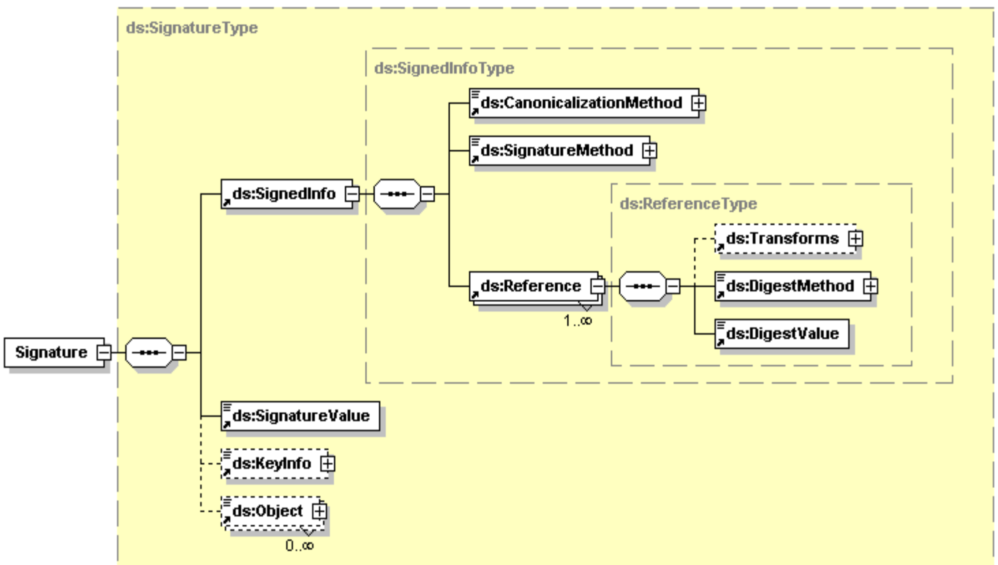
[1] A nonce is a parameter that varies over time and is used as a defence against a replay attack.

Appendix C: W3C XML Digital Signature

Some information on the digital signature is included here, but for full information refer to the Recommendation at [5].

An XML Signature consists of:

- `SignedInfo` which includes a sequence of references to the data being signed with the digest (eg. SHA-1 hash) of the data being signed
- `SignatureValue` which contains the signature value calculated over the SignedInfo using the signature algorithm identified in `SignatureMethod` having been transformed to a canonical form using the method identified in `CanonicalizationMethod`
- `KeyInfo` contains any relevant certificate or key information.
- `Object` can contain any other information relevant to the signature



Appendix E: Revision History

Rev	Date	What
V0.1a	2002-02-07	Draft e-voting schemas for internal comment
V0.2a	2002-02-13	Draft e-voting schemas for internal comment
V0.3a	2002-03-22	Draft e-voting schemas for public consultation comment
V0.4	2002-04-18	Draft Committee Specification version 2
V1.0	2002-04-29	Committee Specification for Technical Committee approval
V1.0	2002-05-13	Committee Specification
V2.0a	2002-06-13	Revised draft accommodating committee's comments
V2.0b	2002-07-15	Draft Committee Specification for Technical Committee approval
V2.0	2002-09-05	Committee Specification
V3.0a	2002-12-12	Draft Committee Specification
V3.0b	2003-02-06	Draft Committee Specification for Technical Committee approval
V3.0	2003-02-24	Committee Specification
V4.0a	2003-10-05	Revised draft accommodating requirements of Council of Europe Member States and UK pilots
V4.0b	2004-01-27	Draft Committee Specification
V4.0c	2004-03-09	Revised draft by placing Schema Description section in document of its own due to excessive size of v4.0b. Draft Committee Specification for Technical Committee approval.
V4.0d	2004-09-03	Draft Committee Specification for Technical Committee approval.

References

1. eXtensible Name and Address (XNAL) Specifications and Description Document (v2.0) Customer Information Quality Technical Committee OASIS July 2002 http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ciq

2. UK Online – Information Architecture – Address and Personal Details Fragment v1.1 Adrian Kent (ed) Office of the e-Envoy 1 March 2002 http://www.govtalk.gov.uk/interoperability/draftschema_schema.asp?schemaid=92

3. Extensible Markup Language (XML) 1.0 (Third Edition) Tim Bray et al Worldwide Web Consortium 4 February 2004 http://www.w3.org/TR/REC-xml

4. XML-Signature Syntax and Processing Donald Eastlake et al Worldwide Web Consortium 12 February 2002 http://www.w3.org/TR/xmldsig-core/

5. Voice Extensible Markup Language (VoiceXML) Version 2.0 Scott McGlashan et al Worldwide Web Consortium 16 March 2004 http://www.w3.org/TR/voicexml20

Notices

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification, can be obtained from the OASIS Executive Director.

OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to implement this specification. Please address the information to the OASIS Executive Director.