Election Markup Language

Version 3 06th February 2003

Document identifier:

EML v3

Location:

http://www.oasis-open.org/committees/election/index.shtml

Editor:

Office of the e-Envoy, UK

Contributors:

John Ross
Paul Spencer
Charbel Aoun

Abstract:

This document contains a high-level overview of the processes within an e-voting system and the data requirements of the flows between those processes. It also addresses security issues relating to the exchange of data, and also provides a glossary of terms to ensure a full understanding by readers of the document. The approved schemes and schema descriptions are also provided.

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 election systems to interact with each other. 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"
- **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 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
- Voter Registration information, including eligible voter lists
- Various communications between voters and election officials, such polling information, election notices, etc.
- Logical 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

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. It covers the conventions used in the specification and the use of IDs, namespaces and displaying messages.

Section 6 provides descriptions of the schemas developed to date. It provides an explanation of the core schemas used throughout, definitions of the simple and complex datatypes, plus the EML schemas themselves.

Appendices: The document concludes with a number of Appendices including a glossary of voting terminology, particularly useful as it indicates some of the issues that arise when attempting to normalize the requirements and even nomenclature of elections internationally.

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 extensions 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 Internet 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.

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, the first XML specification of its kind, and when implemented 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:

- Keep each voter's choice an inviolable secret.
- Allow each eligible voter to vote only once, and only for those offices for which he/she is authorized to cast a vote.
- Do not permit tampering with voting system, nor the exchange of gold for votes.
- Report all votes accurately
- The voting system shall remain operable throughout each election.
- 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. e.g. proportional representation or "first past the post".
- 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 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. 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 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 e-voting system. 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.

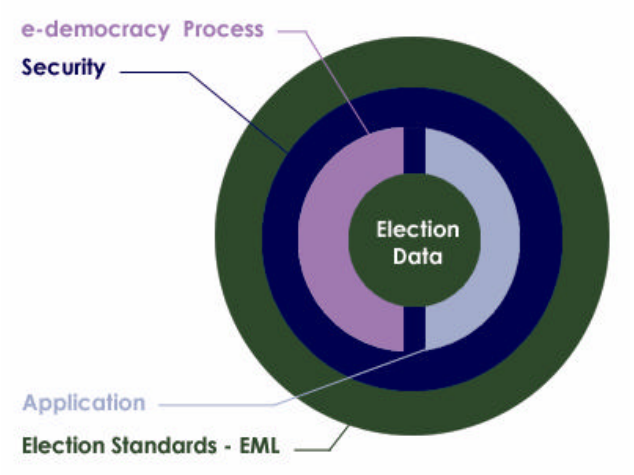


Figure 1A: Relationship overview

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 the vast amount 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 will be 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 needed. An EML schema is introduced and clearly marked.

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.

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	v1.0	
EML310	v2.0, v2.1	
EML320	v1.0, v2.0	v2.0
EML330		v1.1
EML340		v1.0
EML350		v1.0
EML360		v1.3

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 appendix A for the terms used by the committee in this document

Our approach was to regard elections as involving Contests between Candidates or 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**. The model below uses a British context to describe our approach in general terms.

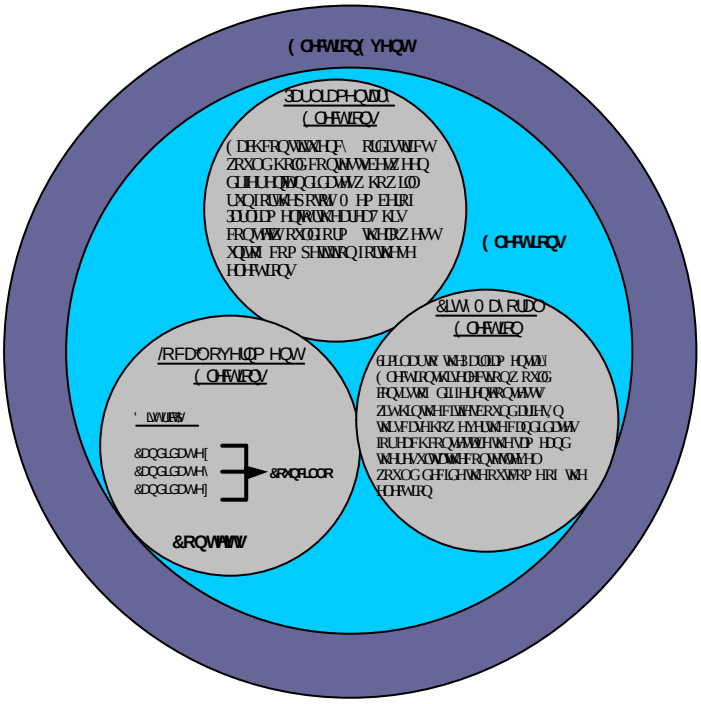
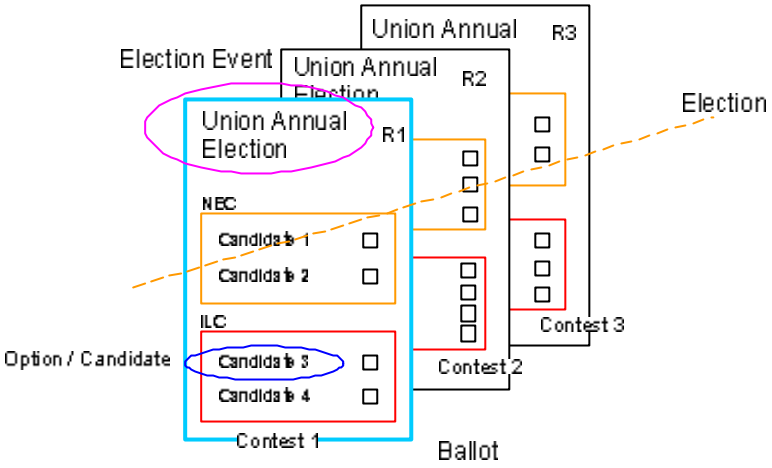


Figure 1B: The Election Hierarchy

In the detailed example below, 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 Liason 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 **options** (or **candidates**).

Figure 1c below shows the three **ballots** (one for each region). The **ballot** is personal to the voter and presents the **options** available to that voter. It also allows choices to be made. During the election exercise, each voter in region 1 receives only the region 1 ballot. This ballot will contain the **candidates** for the (R1) contest for each of the two **elections.**



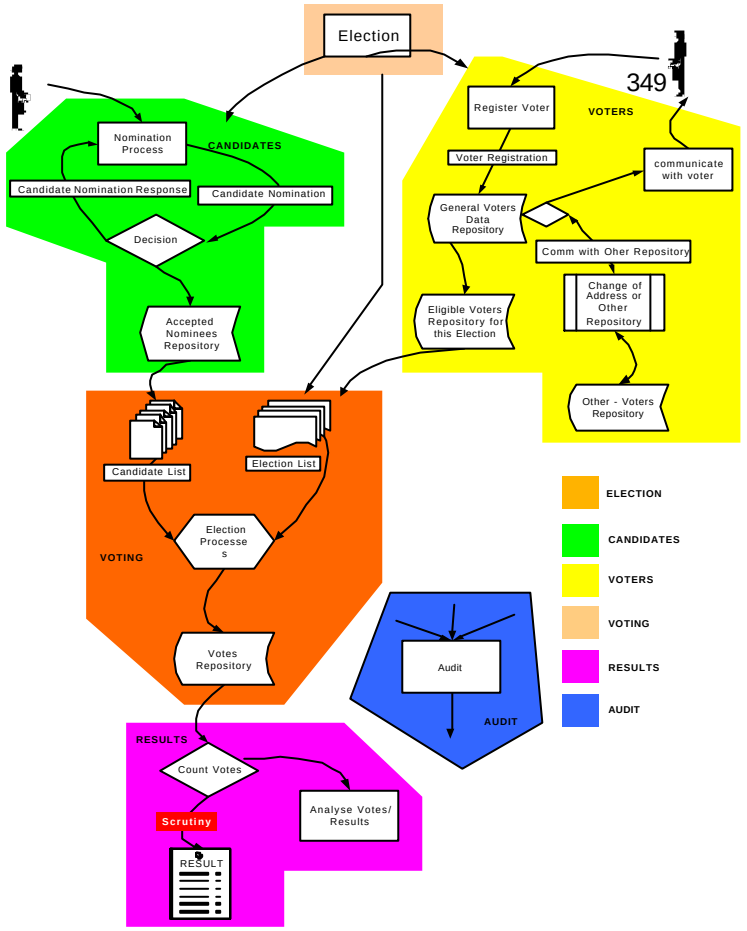
339

340 ***Figure1C: Union annual election***

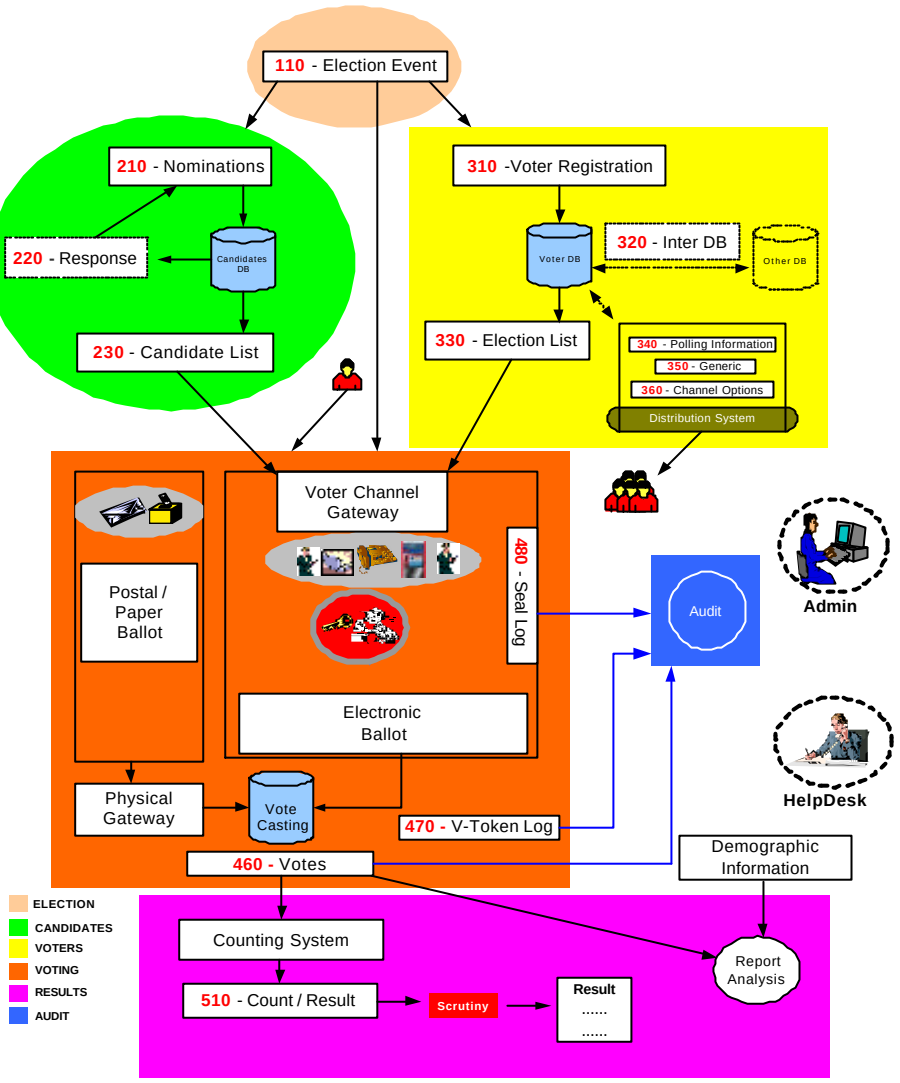
3 High-Level Election Process

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

3.1 Figure 2A: High Level Model – The Human View



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 schema is specified to support each process, this ensures consistency with all the figures and the schemas required:

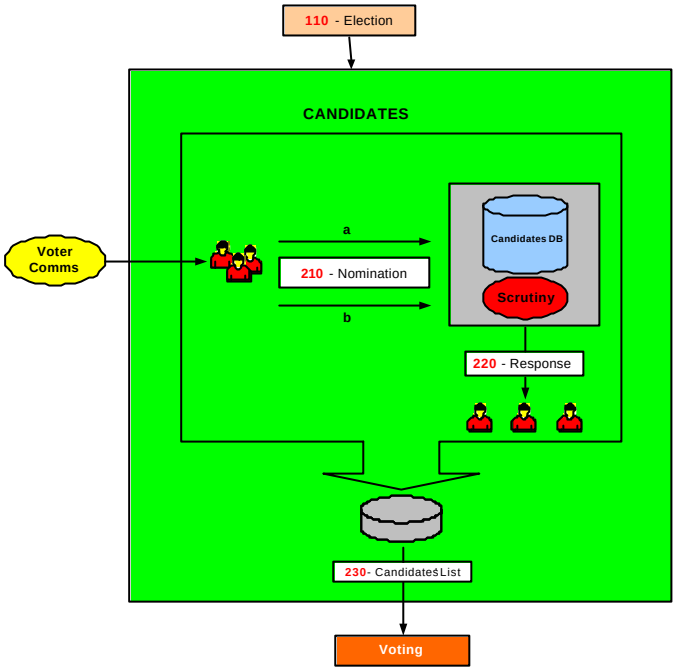
- Pre election
  - Election (100)
  - Candidates (200)
  - Voters (300)
- Election
  - Voting (400)
- Post election
  - Results (500)
  - Audit
  - Analysis

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

- Administration Interface
- Help Desk

3.4 Process Descriptions

Figure 2C: The Candidate Nomination Process



This is the process of approving nominees as eligible candidates for certain positions in an election. Schemas **210, 220** are specifically applicable to candidates' nominations and do not apply for issues like surveys, referendums.

Irrespective of local regulations covering the nomination process, or the form in which a candidate's nomination is to be presented, i.e. (written/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 voters population eligible to vote 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 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 letter specifying Mr. X as his or her nominee for the position in question.

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:

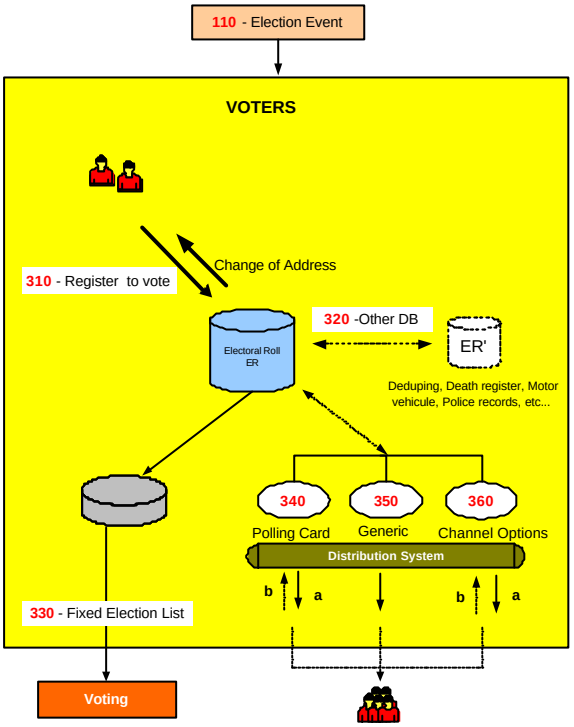
- Pay fees,
- Have x number of endorsers,
- Be of a certain age,
- Be a citizen more than x number of years,
- Etc.

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.

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 contests and occurrence on the final ballot(s).

Figure 2D: 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 DB.

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.

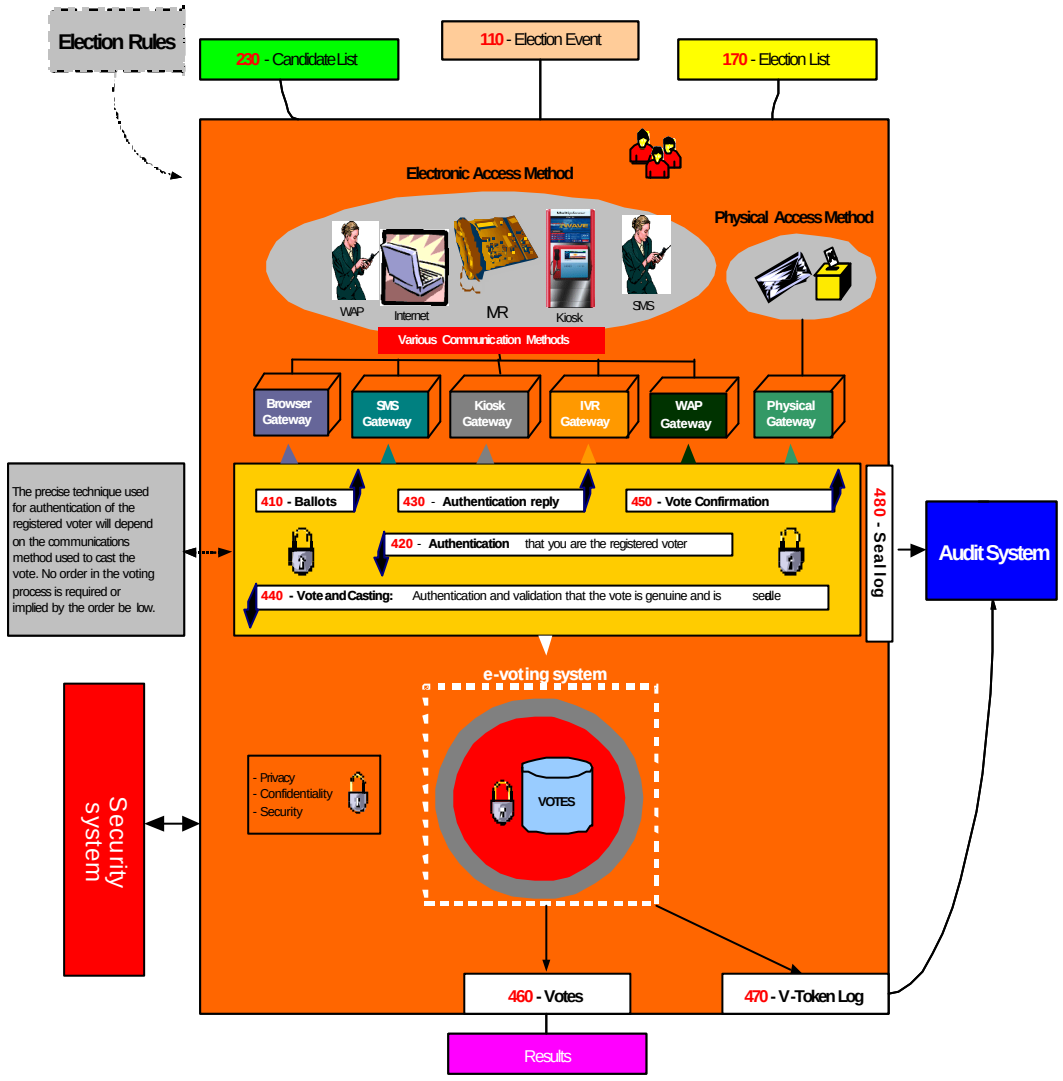
At a certain date, a subset of the voters DB is fixed from which the election list is generated [**Election List 330**] contains some subset of the eligible voters, perhaps grouped by polling district or voting channel.

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.

Figure 2E: 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, polling place or paper ballot (the “physical access methods”), or the voter can vote using “electronic access methods” where he/she will 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.

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

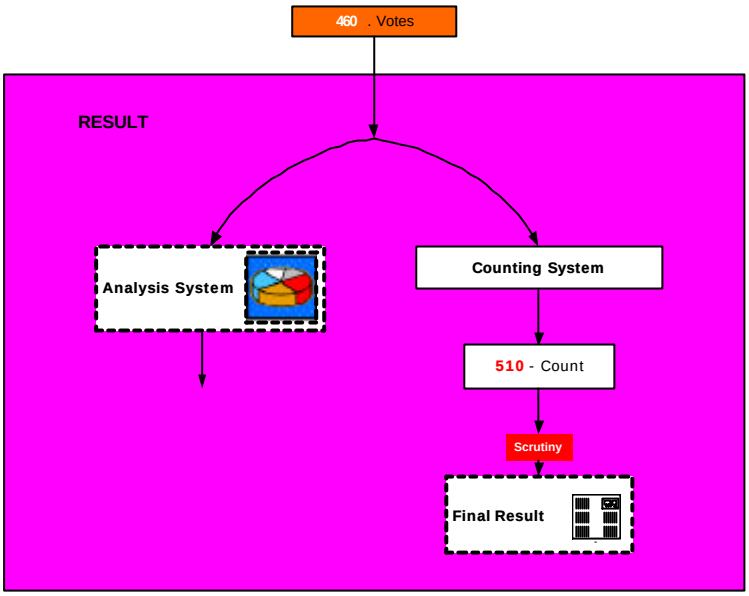
In some scenarios the right to vote may need to be qualified. This may occur if the voter’s right to vote is challenged or if the voter is given the temporary right to vote. In this case the vote needs to be cast by a voter with a qualified V-token. The reason for the qualification shall always be

present in a qualified V-token and the qualification may need to be investigated before the vote is counted as legitimate.

The V-Token and qualified V-token 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 **170** for exchanging such information between e-systems. However, a mapping process may be required in the e-voting system to map the various raw input data into output data for one ballot for one voter. This document uses the term election rules to define how this mapping is to be done in a particular election. When a precise election rule is needed is it identified by the election rule ID.

The current document assumes election rules themselves are implementation specific, thus by specifying the election rule ID the e-voting system can do the necessary mapping between voter, candidate, election and bylaws of the election to produce the ballot. Other issues that can be identified as affecting the election rules are geographical or organizational boundaries.

Figure 2F: The Vote Reporting Process



Two of the post election items are the result and the audit report. Audit is discussed in the next part.

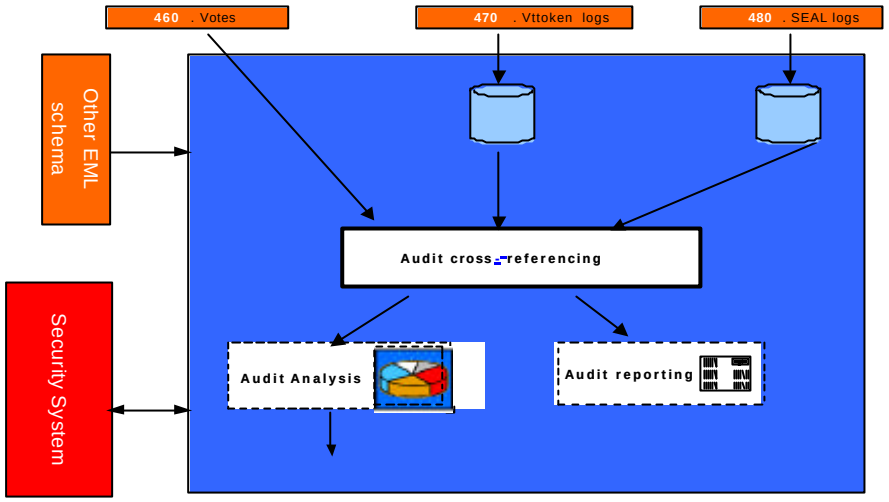
The voting system should communicate a bulk of data representing the votes to the counting system or the analysis system-using schema **460**. The result by itself, which is the compilation of the **460,** is to be communicated by the schema **510**.

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

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 Vote and Count schemas. For example, schemas for messages that report election results to the Press.

472 ***Figure 2G: Auditing System***



473

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

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

479 Schemas **460, 470**, **480** from the voting process provide input data to the audit process.
480 Depending on the audit requirements additional data from other processes may be required. In
481 particular, the security process may provide additional data about all the issued V-Tokens and
482 qualified V-Tokens (see Figure 3a: Voting system security).

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

486 • Eligible voters

487 • Issued V-Tokens or qualified V-Tokens.

488 The audit process can collate the total number of V-Tokens and qualified V-Tokens provided by
489 the security system with the total number reported by the voting system using schema **460** and
490 **470.**

491 The security system and sealing mechanism should be implemented so that trust can be placed
492 in the seal and hence the sealed data. This implies that the seal should be performed as close to
493 the user submission of the vote as technically possible. The count of the spoiled and unspoiled
494 votes from **460** can then be cross-checked against the count of the number of trusted seals from
495 **480**. This collation confirms that the total number of votes presented by the output of the e-voting
496 system in **460** is consistent with the total number of submitted votes with seals.

497 The above collation between trusted data provided by the security process and data provided by
498 the voting process prove that no legitimate votes have been lost by the voting system. It also
499 proves that there is consistency between the number of eligible voters and the spoiled, unspoiled
500 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 proves 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 result is contested. The number of votes from the voting system using schema **460** can be verified by collating the total votes as calculated by the audit system (using schema **480),** with the totals from the counting system. Then either rerunning 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** provides 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 a 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 the EML 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:

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 V-token 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 v-token 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 take 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.

V-token: the means by which a voter proves to an e-voting system that he/she has the right to vote in a contest.

V-token Qualified: the means by which a V-token can be qualified. The reason for the qualification is always appended to a V-token that is qualified. For example, a qualified V-token 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 V-token) 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 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 V-token, 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 V-token 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 V-token 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 V-token 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 identify 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 V-token may also vary depending on legal requirements of a country or electoral rules used in a particular contest. Also, the threats to misuse of v-tokens 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 V-token components must allow for various data types of authentication information to be contained within it.

It must be possible to prove that a V-token is associated with 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 to do so.

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 V-token may need to be qualified with a reason. In this document this is called a V-token Qualified. Before a vote is considered legitimate and counted the reason for the qualification must be have been suitable scrutinized, which could be done by the voting officials.

4.4.3 Protecting exchanges with remote voters:

The V-token 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 V-token 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 (V-token(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 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-reputability 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 must be 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 V-token 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 V-Token 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 V-token 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 V-token.

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

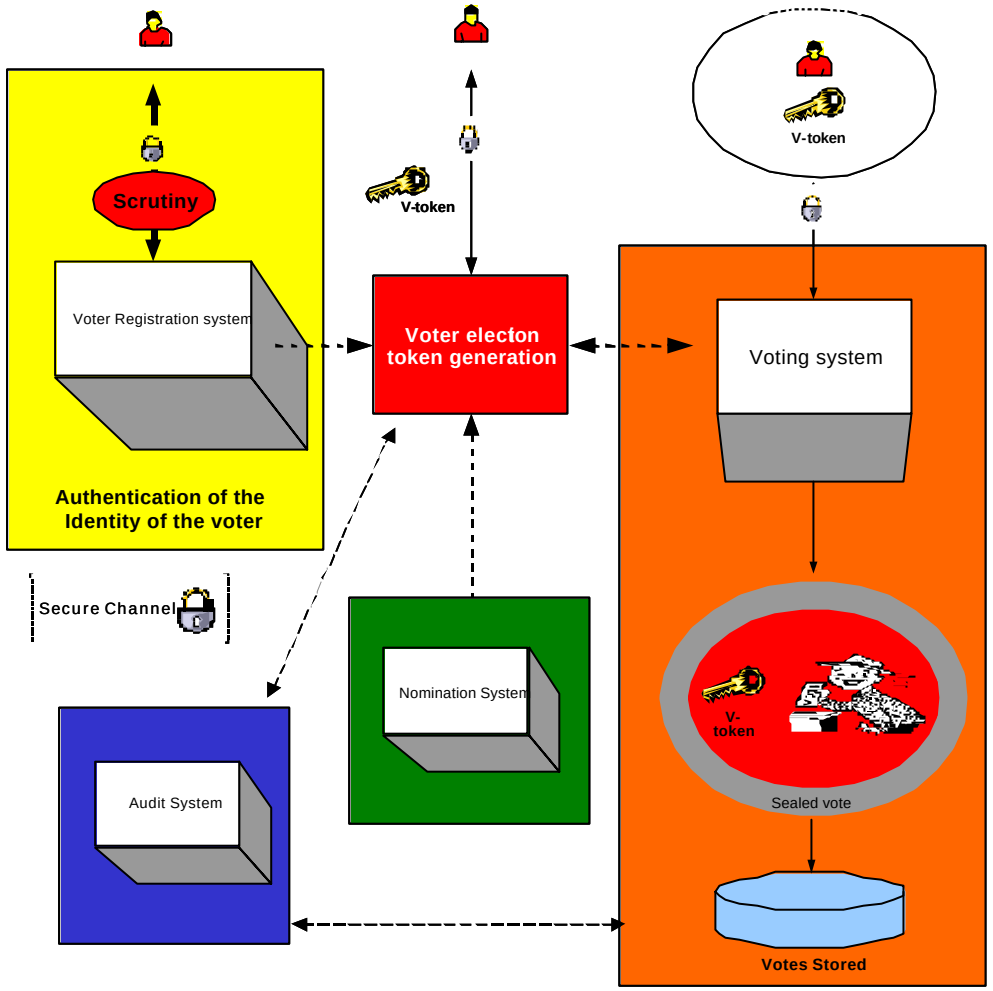
4.4.7 Vote counting accuracy

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 V-token/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 ballot 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 the some component of the system does not have malicious software which will attempt to alter the ballot choices as represented to the voter or alter the voter's selection?

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 a 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.

There is a third category of schema document within EML - the EML externals. This schema 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 standard schemas for XML [3], XLink [4] and 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 `xsd: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 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 in the Process document. 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.

Some identifiers in certain elements are mandatory as shown here:

Element	ID Opt/Man
BallotName	O
CandidateName	M
ContestName	M

ElectionEventName	M
ElectionName	M
LocationName	O
OptionName	M
ReportingUnitName	O
VoterName	O

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:

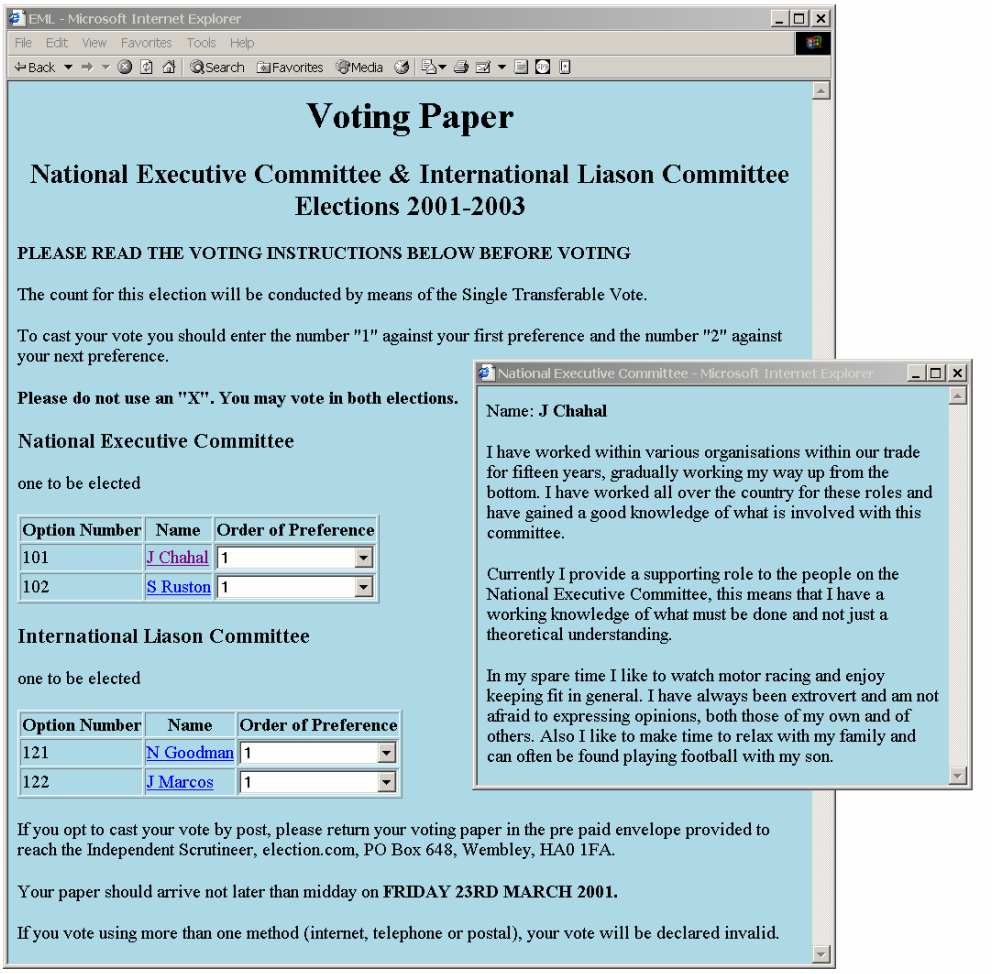
```
<?xml version="1.0" encoding="UTF-8"?>
<EML
  Id="410"
  SchemaVersion="0.1"
  xml:lang="en"
  xmlns="http://www.govtalk.gov.uk/temp/voting"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.govtalk.gov.uk/temp/voting
                      ..\schemas\ballot.xsd">
  <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. The xml:lang attribute on the `EML` element indicates that the message content is in English. Other `Display` elements can be added to cover other formats. 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 [4]), even when using a generic stylesheet.

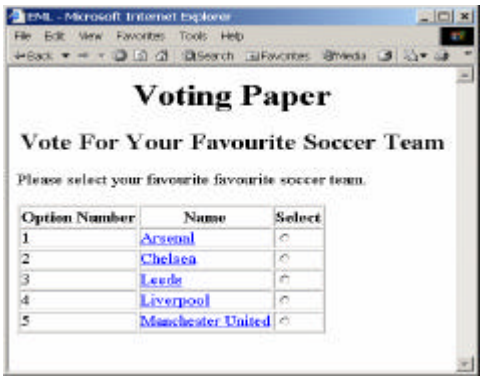
When displaying messages in HTML, the expectation is that generic stylesheets will cover most cases, with the stylesheet output being embedded in a web page generated from an application-specific template. Similarly, voice applications might have specific welcome and sign-off messages, while using a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

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



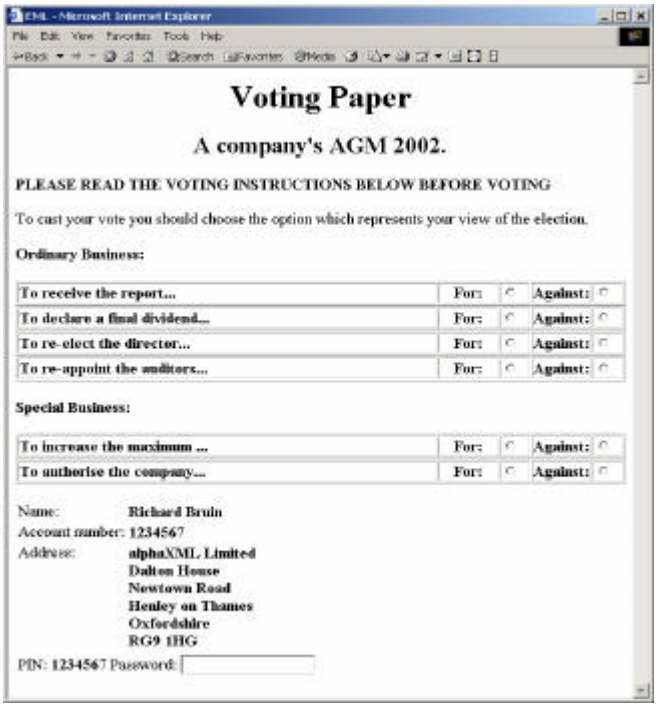
912

913 ***Screen shot of the ballot for scenario 2***



914

915 ***Screen shot of the ballot for scenario 3***



Screen shot of the ballot for scenario 4

5.4 Namespaces

The message schemas and the core schema are associated with the namespace `urn:oasis:names:tc:evs:schema:eml`. Use is also made of the external namespaces for XLink and xNAL, identified here using the prefixes `xlink:` and `xnal:`.

Version 2 of xNAL will have a namespace when it is released, and this will be used here. Currently, an invalid namespace is being used and the `elementFormDefault` has been set to "qualified" so that references to it can be qualified.

5.5 Extensibility

Various elements allow extensibility through the use of the xsd:any element. This is used both for display information (for example, allowing the sending of HTML in a message) and for local extensibility. Note that careless use of this extensibility mechanism could reduce interoperability.

5.6 Conventions

Within this specification, the following conventions are used throughout:

- Element and attribute names are shown in `Courier` font.
- ***Editorial comments are shown like this.***
- Diagrams are shown as generated by XML Spy v4.3, which was also used to generate the schemas and samples. Note that XML Spy will cross out an element that has a as the result of an `xsd:restriction`. It does not do the same where an `xsd:choice` has a

936 maxOccurs value of zero. This has been reported as a bug to Altova. This affects
937 diagrams where the VoterIdentificationStructure is restricted by not allowing
938 either a VToken or VTokenQualified. In these cases, this restriction of the maxOccurs
939 is mentioned in the accompanying text.

940 • Elements and attributes in schemas are identified by partial XPath expressions. Enough
941 of a path is used to identify the item without putting in a full path.

6 Schema Descriptions

This section describes the schemas that make up EML. For data types and elements with complex content, diagrams of the structure are shown. These are expanded to show the complete structure other than where an element is accessed by reference or corresponds to a data type described elsewhere. If the element is derived from a type (rather than being an exact correspondence), the derived structure is shown.

6.1 Core

The core schema contains elements and data types that are used throughout the e-voting schemas.

The choice between defining an element or a data type for a reusable message component is a significant design issue. It is widely accepted as good practice to use element declarations when there is good reason to always refer to an element by the same name and there is no expectation of a need to derive new definitions. In all other cases, data type declarations are preferable. The term *schema component* is used to refer to elements and data types collectively.

When defining a complete markup language, limiting the use of elements and types can restrict further development of the language. For that reason, both data types and elements are defined in EML. Only where an element is an example of a primitive or derived data type defined in XML Schema part 2 [7] is no explicit data type defined within EML.

In use, it is expected that, for example:

- a voting token will always have an element name `VToken` and so will use the element name;
- an address might be an `ElectoralAddress` or a `MailingAddress`, and so will specify a new element based on the data type; and
- within voter identification some elements will usually need to be made mandatory and so a schema will specify a new element based on the `VoterIdentificationStructure` data type.

Currently, the name and address data types are taken from the xNAL schemas as mentioned previously. Investigation is needed to evaluate other schemas for inclusion, embodying agreed definitions for widely used data types such as email addresses and telephone numbers.

The following schema components are defined in `emlcore.xsd`. In the descriptions that follow, element definitions are not shown where they are an example of an obviously-named data type.

Elements	Complex Data Types	Simple Data Types

`Affiliation`
`BallotName`
`CandidateName`
`ContestName`
`ElectionEventN`
	`AuditInformationStructure`
	`BallotNameStructure`
	`CandidateNameStructure`
	`ContactDetailsStructure`
		`ElectionRuleIdType`
		`EmailType`
		`TelephoneNumberType`

ame ElectionName ElectionRuleId ElectionStatement EML EventEnd EventStart LocationName MaxVotes MinVotes OptionName Profile Proposer ReportingUnitName ScrutinyRequirement Seal VoterName VotingChannel VotingMethod VToken VTokenQualified	ContestNameStructure ElectionEventNameStructure ElectionNameStructure EmailStructure IncomingGenericCommunicationStructure LocationNameStructure MessagesStructure OptionNameStructure OutgoingGenericCommunicationStructure ProcessingUnitStructure ProposerStructure ReportingUnitStructure ScrutinyRequirementStructure SealStructure TelephoneStructure VoterIdentificationStructure VoterInformationStructure VoterNameStructure VTokenQualifiedStructure VTokenStructure	VotingChannelType VotingMethodType YesNoType

6.2 Simple Data Types

6.2.1 ElectionRuledType

The election rule ID is used to identify a rule governing an election. For example, a professional society may have a rule that, within a single election event, only a certain class of membership is entitled to vote in one election. The ID can be described as either an `xsd:NMTOKEN` (intended when it references a known document or database) or a URI.

6.2.2 EmailType

This is a string with a maximum length of 129 characters and a pattern `[^@]+@[^@]+`. This allows any characters except the @ symbol, followed by an @ symbol and another set of characters excluding this symbol.

6.2.3 TelephoneNumberType

Since this must allow for various styles of international telephone number, the pattern has been kept simple. The pattern is \+?[0-9\(\)\-\s]{1,35}. This allows an optional plus sign, then between 1 and 35 characters with a combination of digits, brackets, the dash symbol and white space.

6.2.4 VotingChannelType

This type exists to hold the possible enumerations for the channel through which a vote is cast. These are:

- `SMS`
- `WAP`
- `digitalTV`
- `internet`
- `kiosk`
- `polling`
- `postal`
- `telephone`
- `other`

If `other` is used, it is assumed that those managing the election will have a common understanding of the channel in use.

6.2.5 VotingMethodType

The VotingMethod type holds the enumerated values for the type of election (such as *first past the post* or *single transferable vote*). The full set of enumerations is:

- `FPP`
- `OPV`
- `SPV`
- `STV`
- `additonalmember`
- `approval`
- `block`
- `partylist`
- `supplementary`
- `other`

6.2.5.1 YesNoType

This is a simple enumeration of `yes` and no and is used for elements and attributes that can only take these binary values.

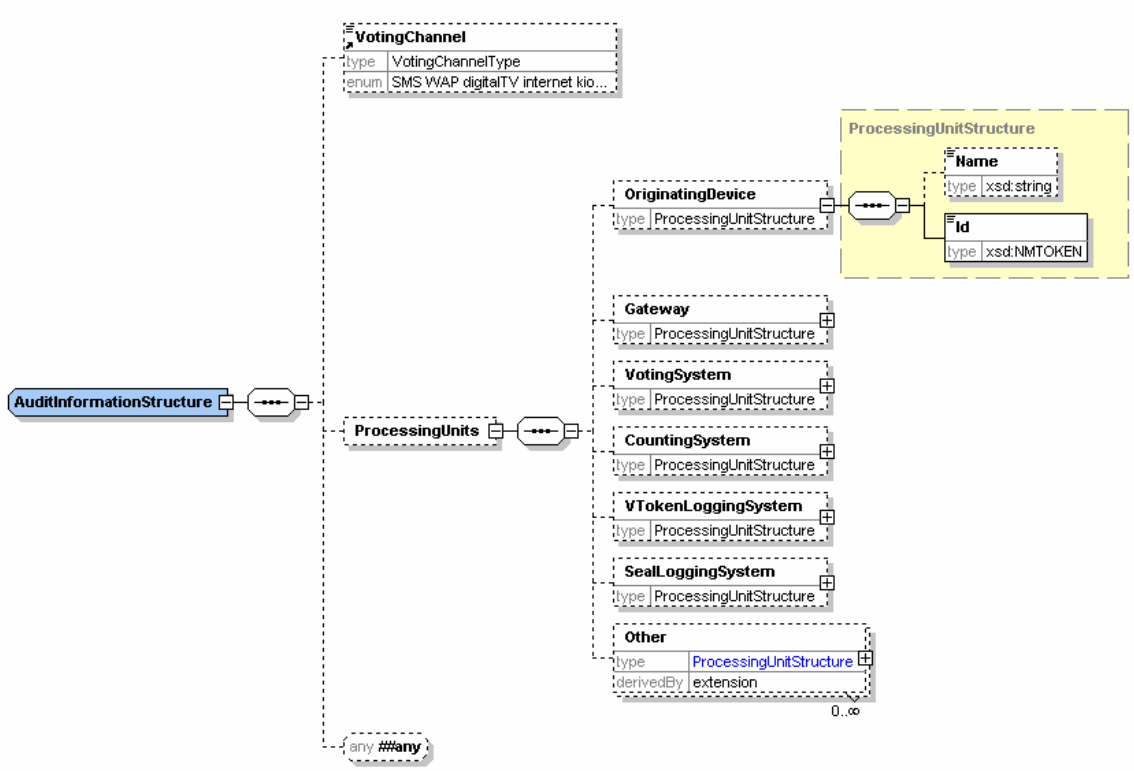
6.3 Complex Data Types

6.3.1.1 AuditInformationStructure

This data type contains information that might be required for auditing a cast vote. This comprises information regarding the channel used for the casting of the vote and IDs for devices used n the voting process (for example, a phone number for an SMS vote or the IP address of a gateway). All the fields are optional, and the intention is that elements will be derived from this data type by just including the information relevant to a specific part of the voting process.

Each named device type device has a mandatory `Id` and an optional `Name`. There is also provision for a device type `Other`. As well as the `Name` and `Id`, this has a `Type` attribute. This allows devices other than those shown in the generic voting process to be identified.

An `any` element is included for extensibility.



6.3.1.2 BallotNameStructure

The ballot name structure defines a string with two optional attributes: `Id` and `DisplayOrder`.

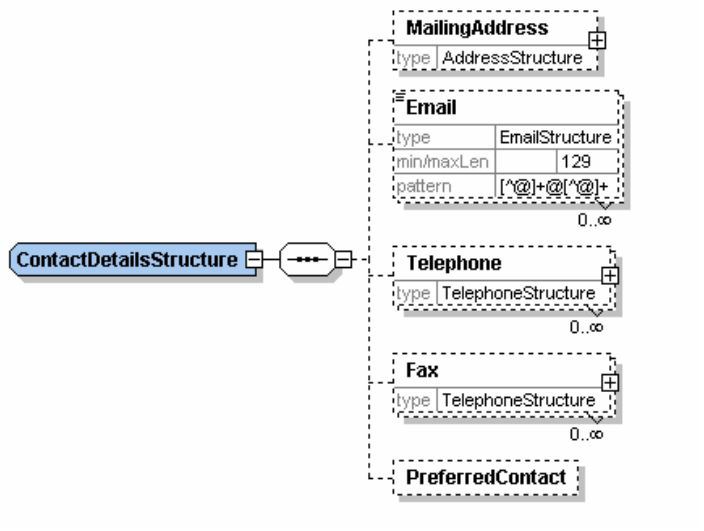
6.3.1.3 CandidateNameStructure

The candidate name structure defines a string with a mandatory `Id` and optional `DisplayOrder` attribute.

6.3.1.4 ContactDetailsStructure

This data type allows for a set of contact details. Each can be qualified through attributes as shown in the descriptions of e.g. `EmailStructure` below. The `PreferredContact` is an XLink to a definition of the preferred means of contact. The destination of this link could be part of this structure or could be elsewhere in this or another document. The use of this mechanism is illustrated in the scenario for voter registration for a UK Parliamentary Election.

As an example of the use of `PreferredContact` and the `Preferred` attributes on email addresses and phone and fax numbers, consider the case of an election officer needing to contact a person. The officer should take note of the preferred method of contact. If this is unsuitable, for example the preferred method is by post, but the need for contact is urgent, the officer might decide that the telephone is the appropriate contact method, see several phone numbers and use the one whose `Preferred` attribute has a value of `yes`. Thus the `PreferredContact` takes precedence over the `Preferred` attribute, the latter only being used when the former does not indicate a suitable contact method.



6.3.1.5 ContestNameStructure

The contest name structure defines a string with a mandatory `Id` and optional `DisplayOrder` attribute.

6.3.1.6 ElectionEventNameStructure

The election event name structure defines a string with a mandatory `Id` and optional `DisplayOrder` attribute.

6.3.1.7 EmailName

This is an extension of the `EmailType` and adds a `Preferred` attribute of type `YesNoType`. This indicates which of several email addresses is preferred.

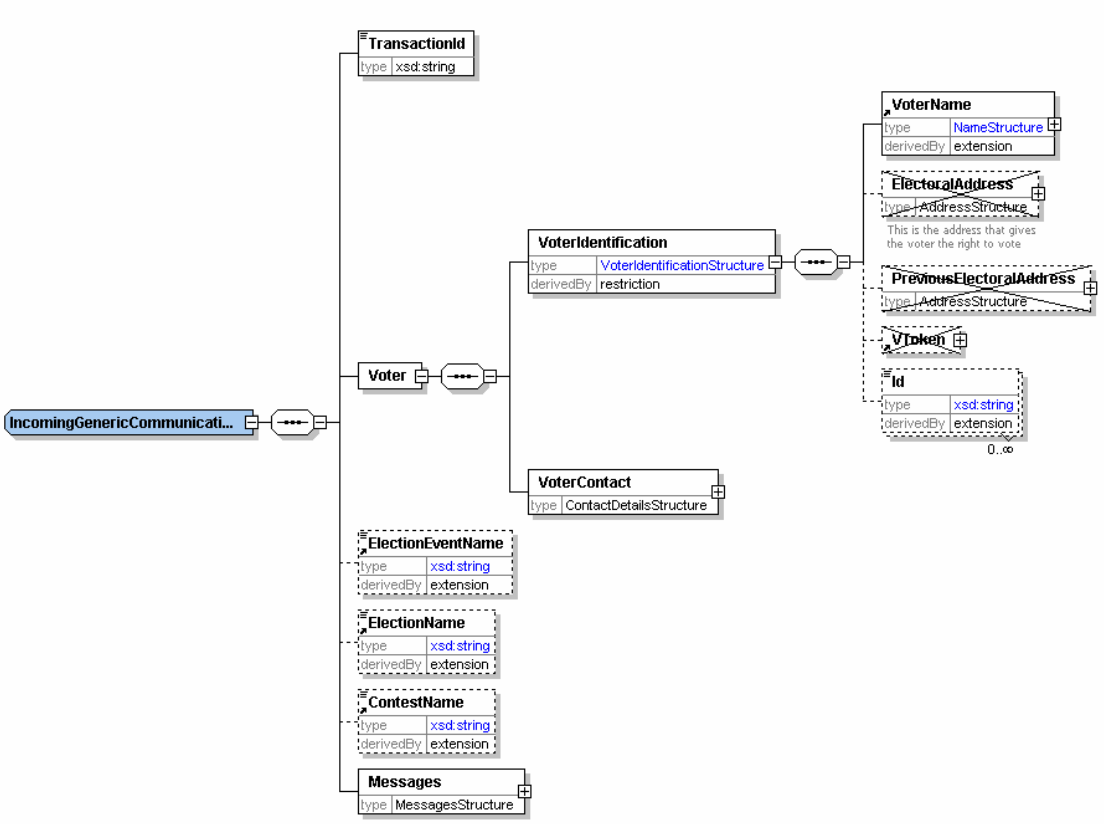
6.3.1.8 IncomingGenericCommunicationStructure

This data type provides a common structure for incoming communications. Individual message types, such as that used for selecting a preferred voting channel (schema 360b) are based on extensions of this schema.

The `TransactionId` is used to reference an outgoing message to which this is a response or to provide a reference for a response.

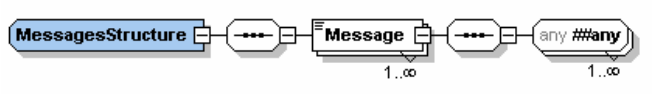
The voter must always provide a name and might provide one or more identifiers. These are shown as a restriction of the `VoterIdentificationStructure`. Contact details are also required, and it is expected that at least one of the allowed contact methods will be included.

The names of the election event, election and contest are optional. There is then an element in which a message can be placed in any of several different formats according to the channel being used.



6.3.1.9 MessagesStructure

This data type is used for general display information. The `Messages` element contains a `DisplayOrder` attribute. The `Message` element contains a `Format` attribute indicating the type of output intended (HTML, WAP, VoiceXML etc.).



6.3.1.10 OptionNameStructure

The option name structure defines the name of a candidate (when a person) or choice (when a resolution) and is a string with a mandatory `Id` and optional `DisplayOrder` attribute.

6.3.1.11 OutgoingGenericCommunicationStructure

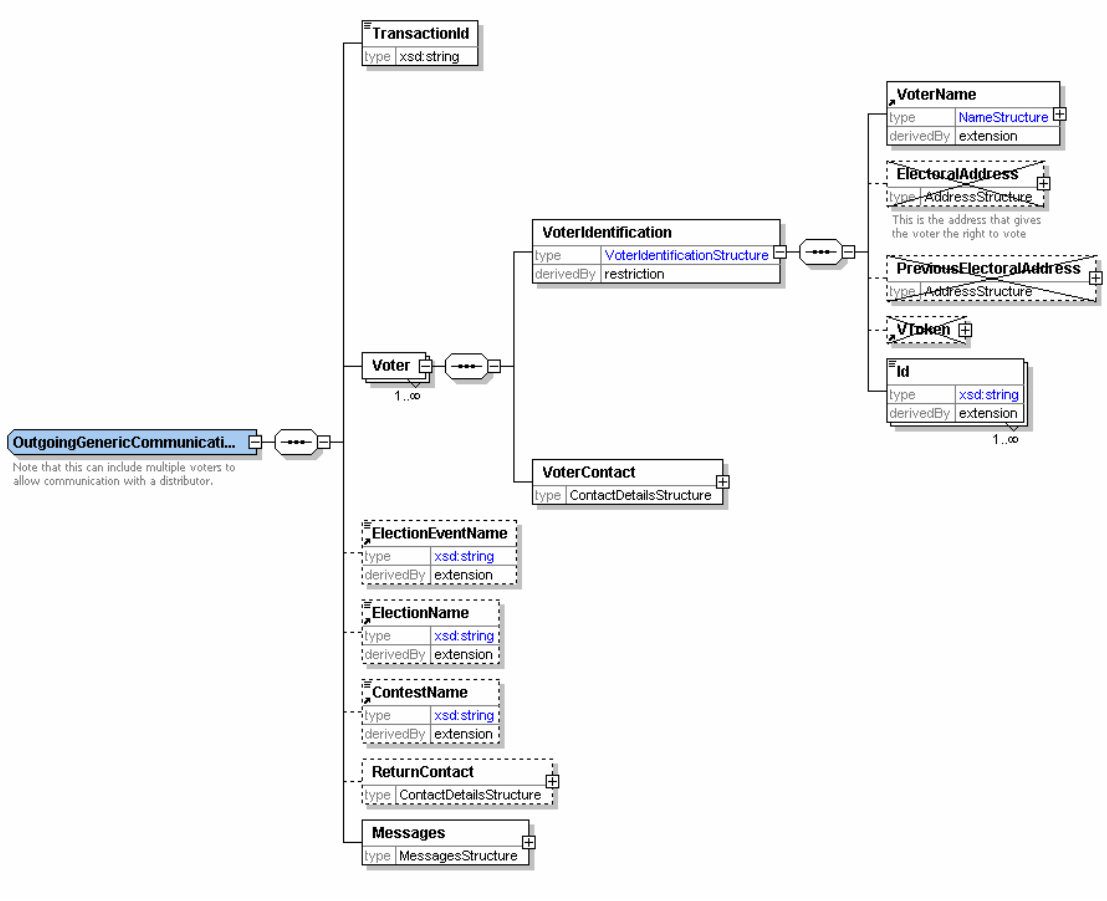
This data type provides a common structure for outgoing communications. Individual message types, such as that used for requesting the selection of a preferred voting channel (schema 360a) are based on extensions of this data type.

Unlike the schema for incoming communications, messages to multiple voters are allowed to enable this schema to be used to describe messages being sent to a distributor (such as a printer or email bureau).

The `TransactionId` is used to provide a reference to be used in a response or to reference an incoming message to which this is a response

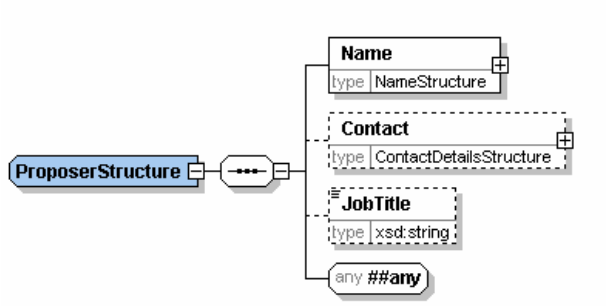
Each voter must have a name and one or more identifiers. These are shown as a restriction of the `VoterIdentificationStructure`. Contact details are also required, and it is expected that at least one of the allowed contact methods will be included.

The names of the election event, election and contest are optional. There may also be contact information provided to allow a reply. There is then an element in which a message can be placed in any of several different formats according to the channel being used.



7.1.2.12 ProposerStructure

A proposer proposes, seconds or endorses an option. A name is always required, and additional information might be needed.



6.3.1.12 ReportingUnitNameStructure

The reporting name structure defines a string with an optional `Id` and optional `DisplayOrder` attribute.

6.3.1.13 ScrutinyRequirementStructure

A scrutiny requirement has two parts, a `Type` attribute and a text value. The `Type` specifies a condition that a candidate must meet, such as an age or membership requirement or the payment of a fee. The text describes how that condition has been met. For example:

```
<ScrutinyRequirement Type="dateofbirth">8 June
1955</ScrutinyRequirement>
```

6.3.1.14 SealStructure

The seal is used to protect information such as a vote, voting token or complete message. The seal provides the means of proving that no alterations have been made to a message or individual parts of a message such as a vote or collection of votes, from when they were originally created by the voter. The seal may also be used to authenticate the identity of the system that collected a vote, and provide proof of the time at which the vote was cast.

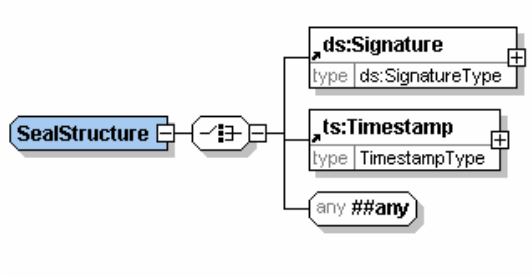
If a message is to be divided, each part must be separately sealed to protect the integrity of the data. For example, if votes in several elections are entered on a single ballot, and these votes are being counted in separate locations, each vote must be separately sealed.

A seal may be any structure which provides the required integrity characteristics, including:

- an XML signature (as defined in http://www.w3.org/2000/09/xmldsig)
- a time-stamp (see Appendix C)
- other mechanisms

The XML signature created by the voting system provides integrity and authentication of the identity of the system that collected the vote. The time-stamp provides integrity of the vote and proof of the time that the vote was cast.

The other mechanism may be used, for example a combinations of an authentication mechanism and timestamps that will provide integrity of the vote, authentication of the identity of the system that collected the vote, and proof of the time that the vote was cast.



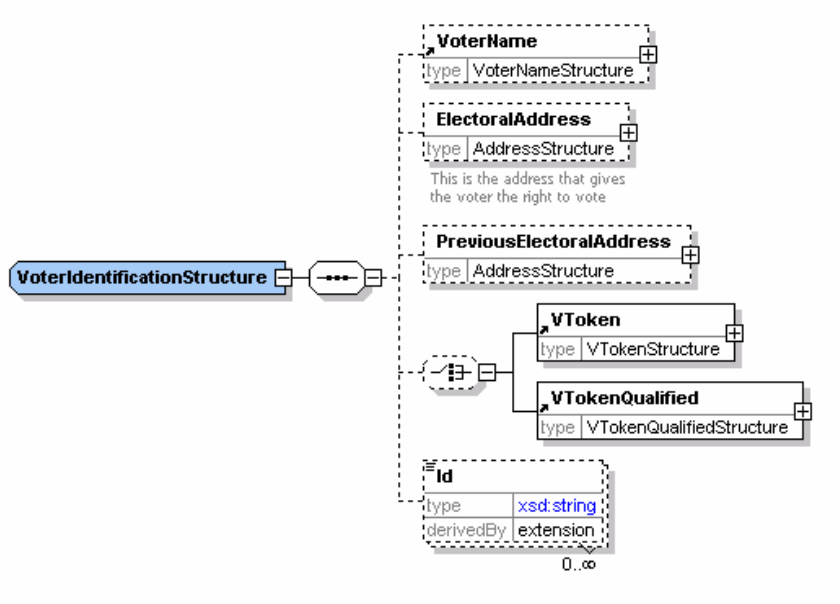
6.3.1.15 TelephoneStructure

This is an extension of the `TelephoneType` and adds the two attributes `Preferred` and `Mobile` of `YesNoType`. The `Preferred` attribute indicates which of several phone numbers or fax numbers is preferred.

6.3.1.16 VoterIdentificationStructure

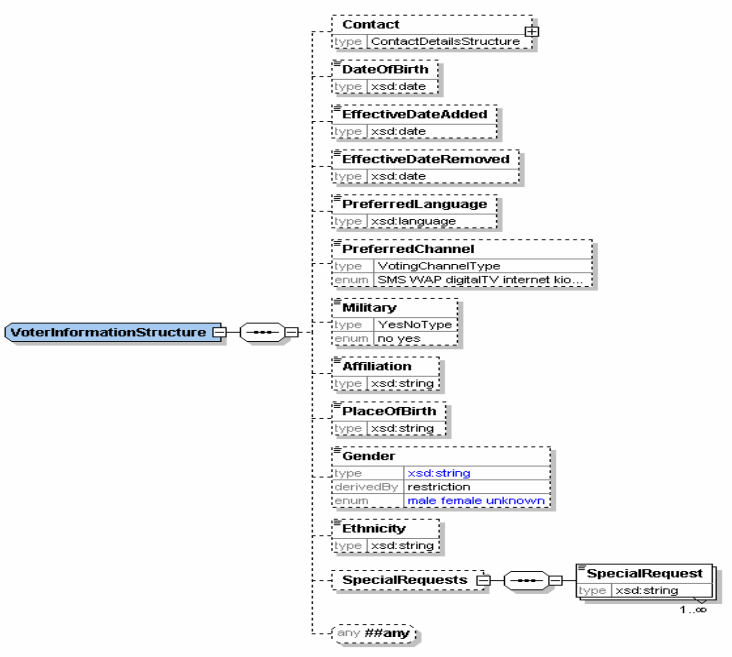
This is used wherever identification of a voter is required. It contains the voter's name and electoral address (using definitions from xNAL), the voting token (either normal or qualified (see section 7.1.2.19) and a number of identifiers (such as an electoral roll number). It may also include a previous electoral address if this is required (for example, because a voter has not been at his or her current address for more than a predefined period).

This has been produced as a complex data type rather than an element since it is expected that it will usually be restricted (for example, many uses will make the `VoterName` mandatory).



6.3.1.17 VoterInformationStructure

This contains more information about the voter. It contains all the information that would typically be included on an electoral roll other than that used for identification of the voter. It contains an xsd:any element for extensibility. This has been produced as a complex data type rather than an element since it is expected that it will usually be restricted.



1149

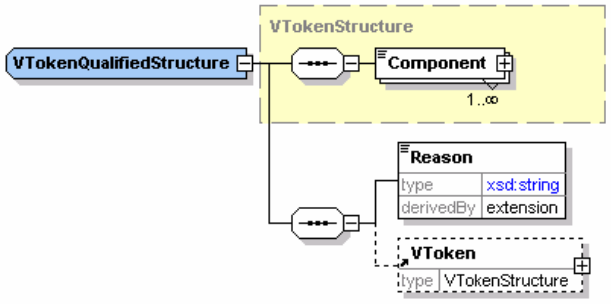
1150 6.3.1.18 VoterName

1151 The voter name structure defines a string with two optional attributes: `Id` and `DisplayOrder`.

1152 6.3.1.19 VTokenQualifiedStructure

1153 There are occasions when a normal VToken cannot be used. For example, if a voter is
1154 challenged, or an election officer claims the voter has already voted. In these circumstances a
1155 qualified VToken can be used and treated appropriately by the election system according to the
1156 election rules. For example, challenged votes might be ignored unless there were sufficient to
1157 alter the result of the election, in which case each vote would be investigated and counted if
1158 deemed correct to do so.

1159 The `VTokenQualifiedStructure` is therefore an extension of the `VTokenStructrue` to add
1160 the additional information required. This additional information comprises a reason for
1161 qualification (as a `Reason` element with a `Type` attribute and textual description) and possibly an
1162 original `VToken`.



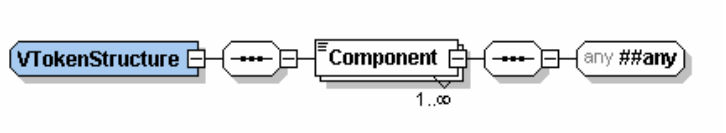
1163

6.3.1.20 VTokenStructure

The VToken contains the information required to authenticate the voter's right to vote in a specific election or contest. A VToken can consist of a continuous string of encoded or encrypted data, alternatively it may be constructed from several data components that a user may input a various stages during the voting process (such as PIN, password and other coded data elements). The totality of the VToken data proves that a person with the right to vote in the specific election has cast the vote.

Depending on the type of election, the voter may need to cast their votes anonymously, thus not providing a link to the voter's true identity. In this case the VToken data will not identify the actual person casting the vote, it just proves that the vote was cast by a person with the right to do so. Other election rules require a link to be maintained between a vote and a voter, in which case a link is maintained between the VToken data and the voter's identity.

The components of the VToken are identified by a Type attribute and may contain text or any markup from any namespace depending on the token type. The content could be defined further in separate schemas for specific types of token.



6.3.2 Elements

Elements are defined here if:

- their type is a generic EML type such as MessagesStructure rather than a specific type such as AuditInformationStructure;
- they are derived from an EML data type by extension or restriction; or
- they are of a data type defined in XML Schema part 2 [6].

6.3.2.1 Affiliation

This is a text string used to identify the affiliation (e.g. political party) of a candidate in an election.

6.3.2.2 ElectionStatement

This is the candidate's message to voters and is an extension of the MessagesStructure to allow multiple languages.

6.3.2.3 EML

This element is used as the document element for all Election Markup Language messages. It has three attributes: an Id that relates to the Id of the associated message in the Process document, a SchemaVersion that indicates the full version number of the schema with which the message was designed to comply, and an xml:lang that indicates the language of the message content.

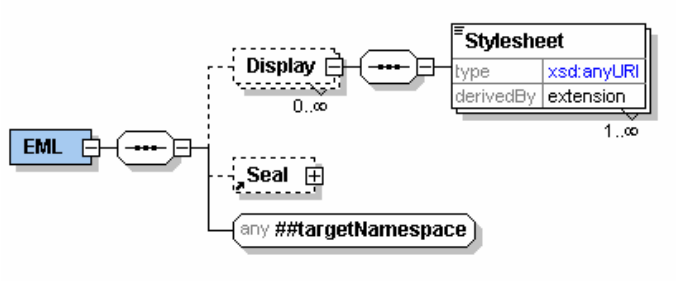
The EML element can contain multiple Display elements. These contain Stylesheet elements that indicate a MIME type (using the Type attribute) and a URI as the element value. The

`Display` element has a `Format` attribute that indicates the target channel for the display (such as HTML). The reason for having multiple `Display` elements is to allow the same message to be presented appropriately through different channels.

The `EML` element can also contain a `Seal` element. This is used to seal the complete message so that any tampering can be detected.

In general, there will only be a single `Stylesheet` element per `Display` element. More are allowed so that the output of an XSLT transformation to HTML can contain a reference to a CSS stylesheet to be used to display the transformed message.

Finally, the `EML` element can contain any other element from the EML namespace. These will be elements such as `Ballots` and `VoterRegistration` defined in the other schema documents that form the Election Markup Language.



6.3.2.4 EventEnd

This is the end date/time of the election event in `xsd:dateTime` format.

6.3.2.5 EventStart

This is the start date/time of the election event in `xsd:dateTime` format.

6.3.2.6 LocationName

The location name is a string with two optional attributes: `Id` and `DisplayOrder`.

6.3.2.7 MaxVotes

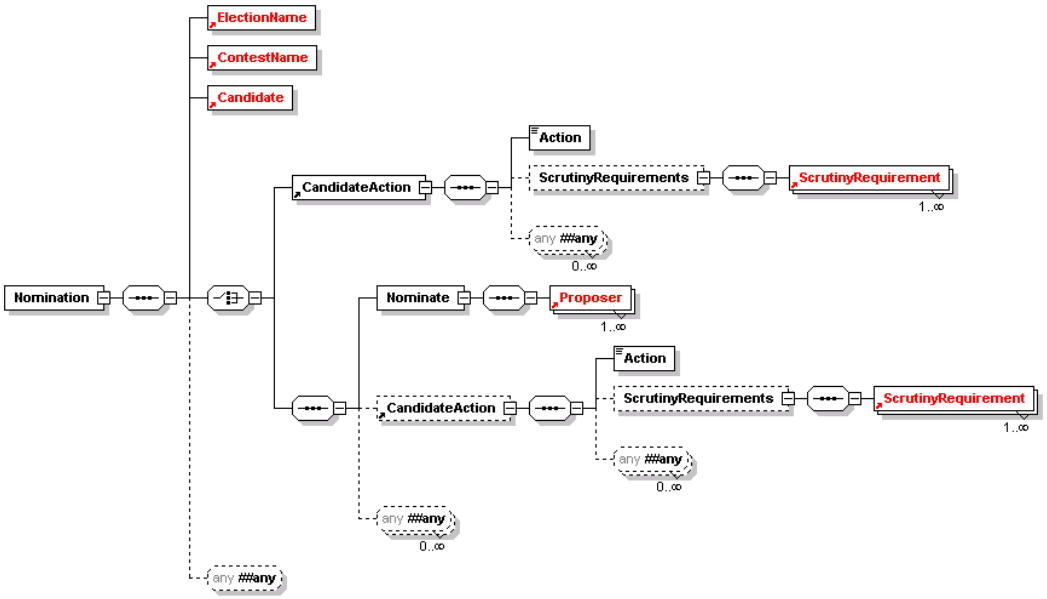
The maximum number of votes allowed (also known as the vote limit). This is an `xsd:positiveInteger` and defaults to a value of `1`.

6.3.2.8 MinVotes

The minimum number of votes allowed. This is an `xsd:nonNegativeInteger` and defaults to a value of `0`.

1223 ### 6.3.2.9 Profile

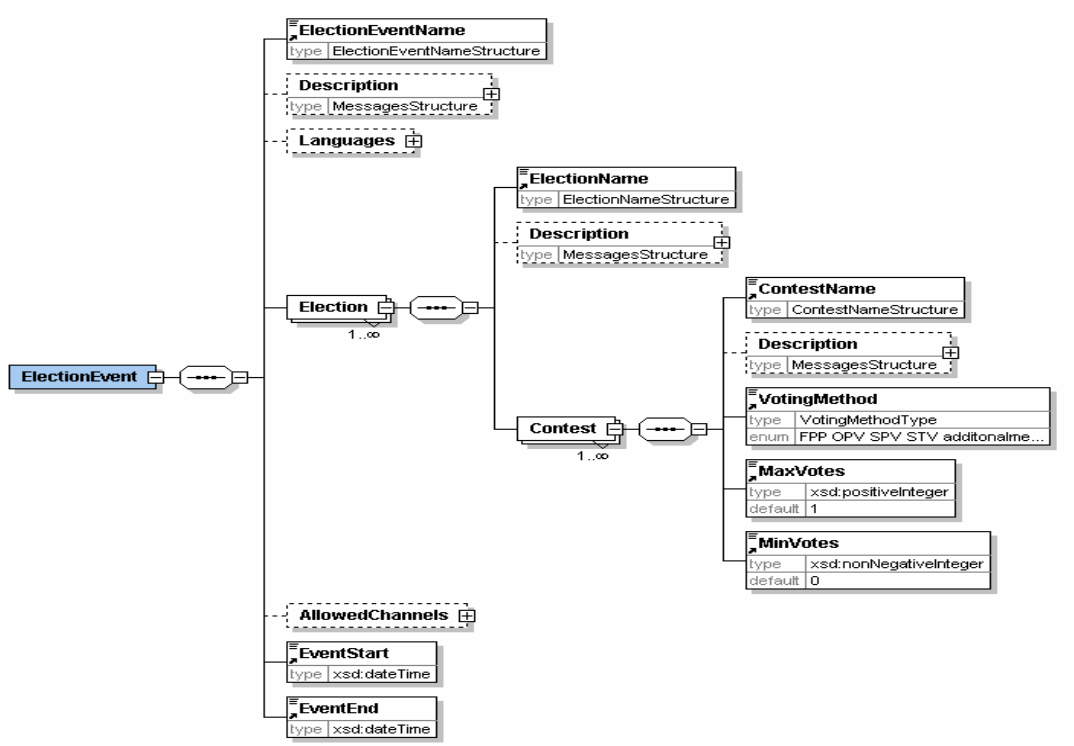
1224 This is the candidate's profile statement and is an extension of the `MessagesStructure` to allow
1225 multiple languages.



1226

6.4 EML Schemas

6.4.1 Election Event (110)



This schema is used for messages providing information about an election or set of elections. An event has a start and end date and time, a list of allowed voting channels, a list of the languages in which information is to be available and a set of one or more elections. Each election may have multiple contests, each of which can have a different voting method (e.g. *first past the post* or *single transferable vote*). Some voting methods will specify the maximum and minimum numbers of votes, but if these are omitted, they default to sensible values.

6.4.2 Nomination (210)

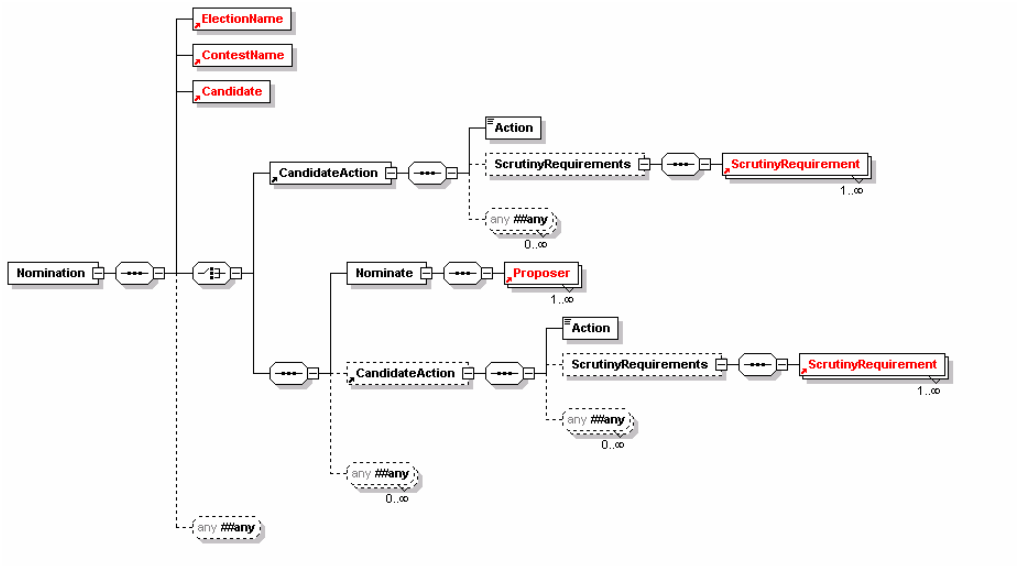
This schema is used for messages nominating candidates in an election. Note that it does not cover other forms of option nomination - only human candidates.

The election and contest must be specified as well as information about the candidate and one or more proposers. The candidate must supply name and contact information. The contact data is derived from the standard data type by making the address mandatory. Optionally, the candidate can provide an affiliation (e.g. a political party) and textual profiles and election statements. These two items extend the `MessagesStructure` to allow text in multiple languages. There is also scope to add additional information defined by the election organiser.

The proposers use the standard proposer declaration with a mandatory name and optional contact information and job title. Again, additional information can be required.

The scrutiny requirements indicate how the candidate has met any conditions for standing in this election.

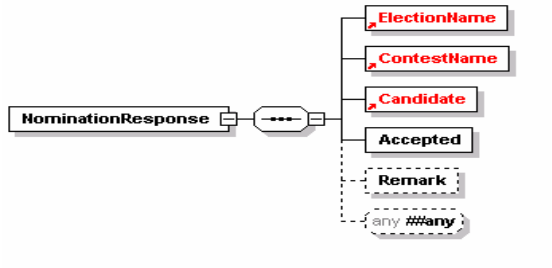
1249 Finally, there is scope to extend the schema by adding additional information to the nomination.



1250

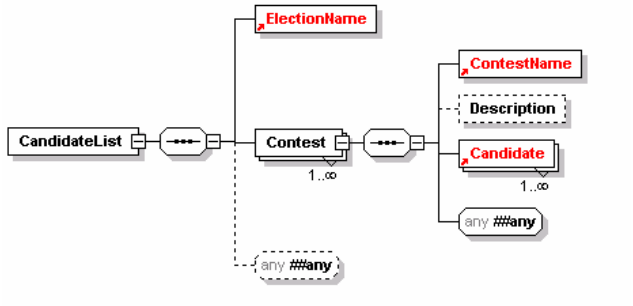
1251 ## 6.4.3 Nomination Response (220)

1252 This message is sent from the election organiser to the candidate to say whether the nomination
1253 has been accepted. Along with the acceptance information and the basic information of election,
1254 contest and candidate names, the candidates contact details and affiliation can be included and a
1255 remark explaining the decision.



1256 ## 6.4.4 Candidate List (230)

1257 This schema is used for messages transferring candidate lists for a specified contests. It has the
1258 election event name, contest name (with its ID), optionally a contest description and then a list of
1259 candidates, each with a name and optional affiliation.

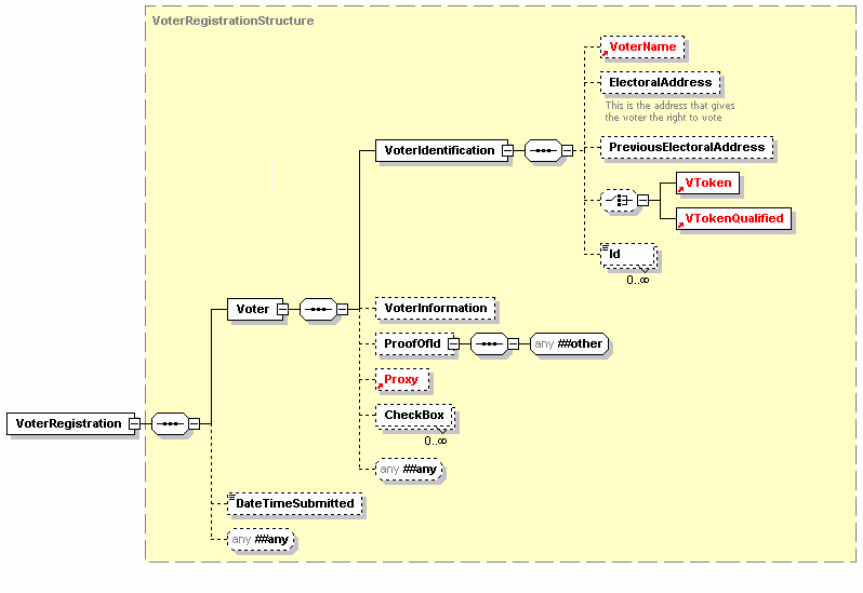


6.4.5 Voter Registration (310)

This schema is used for messages registering voters. It uses the `VoterIdentificationStructure` described in section 6.1.2.17, with the exception that no `VToken` or `VTokenQualified` is allowed. The `VoterInformationStructure` is used unchanged.

There is the facility to add a proof of ID and for the transmission channel (for example a trusted web site) to add the time of transmission.

This schema allows any additional data to be added to the message for appropriate local extensions.

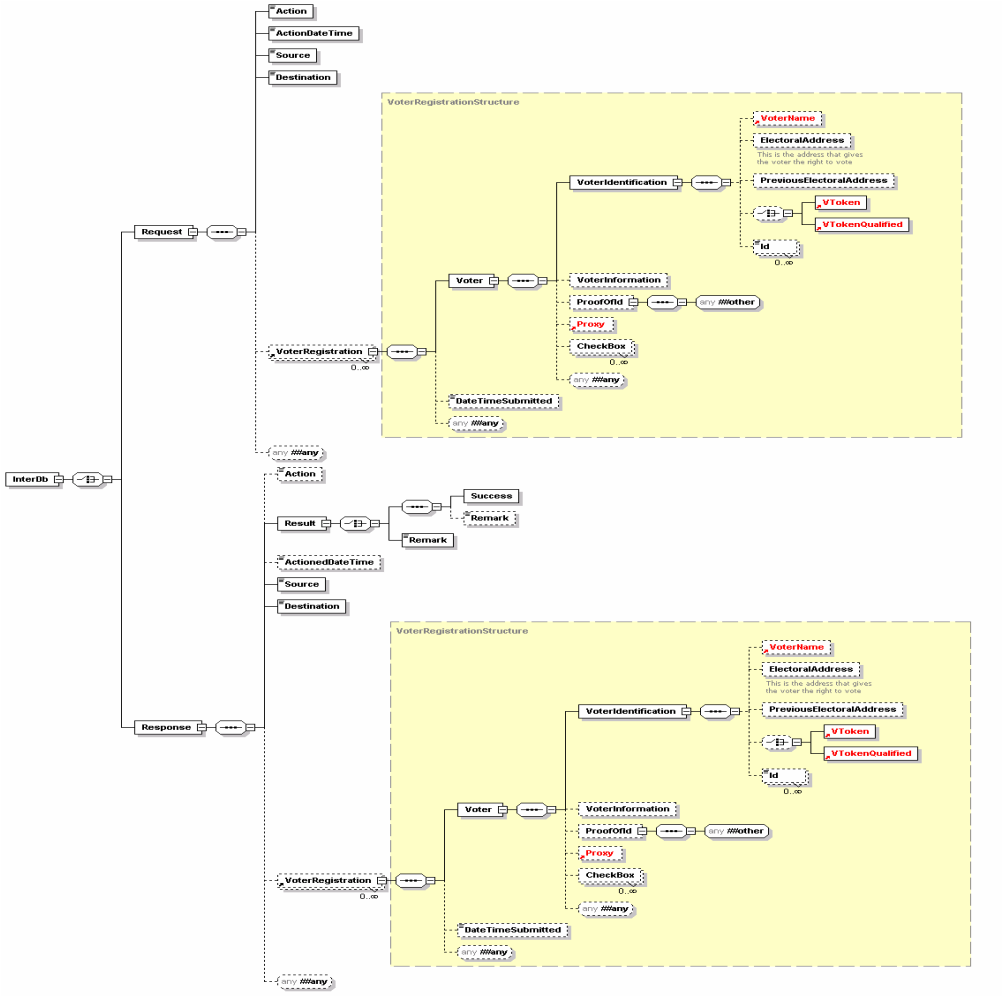


6.4.6 Inter Database Communications (320)

This schema is used for messages requesting services from other electoral list databases. This can, for example, be used to de-dupe databases. The schema is in two parts, so a message will be either a request or a response.

A request starts with an `Action` code and a `TransactionId` that can be used to correlate the response with the original request. The `ActionDateTime` is used to specify when the action should be carried out. The `Source` and `Destination` are used as identifiers (either string or URI) and then there is an optional list of voters. The message can also be extended through the `xsd:any` element.

A response has a similar structure. It could be that the `Action` code is no longer required, so this is now optional. The `TransactionID` must match that given in the request. The `Result` is either a binary `Success` flag or a remark or both. Again, there is a date and time, but in this case it is the date and time at which the action took place.

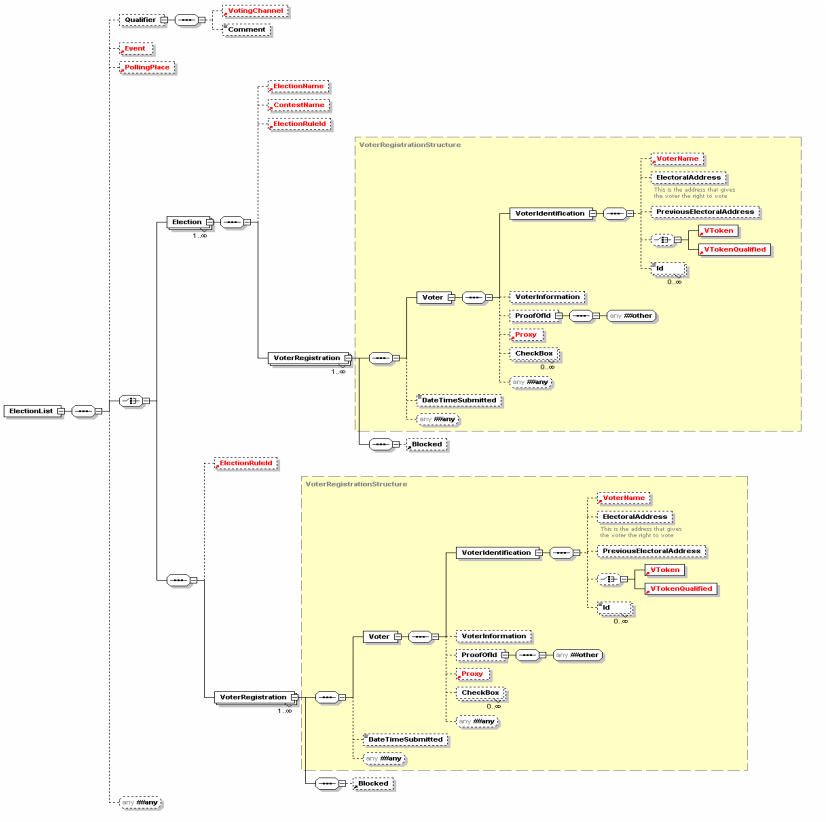


6.4.7 Election List (330)

This schema is used for messages communicating the list of eligible voters for an election event or election within the event. This choice is allowed as frequently the same population will be able to vote at all elections within an event, but on other occasions the elections will have different lists.

One choice is therefore to send in one message a sequence of the election event name and ID, followed by an election rule ID and a list of voter registrations. The election rule indicates which voters in the register will be able to vote in this election event.

1292 The other choice is to indicate the election, and optionally an individual contest, to which the voter
1293 list applies.



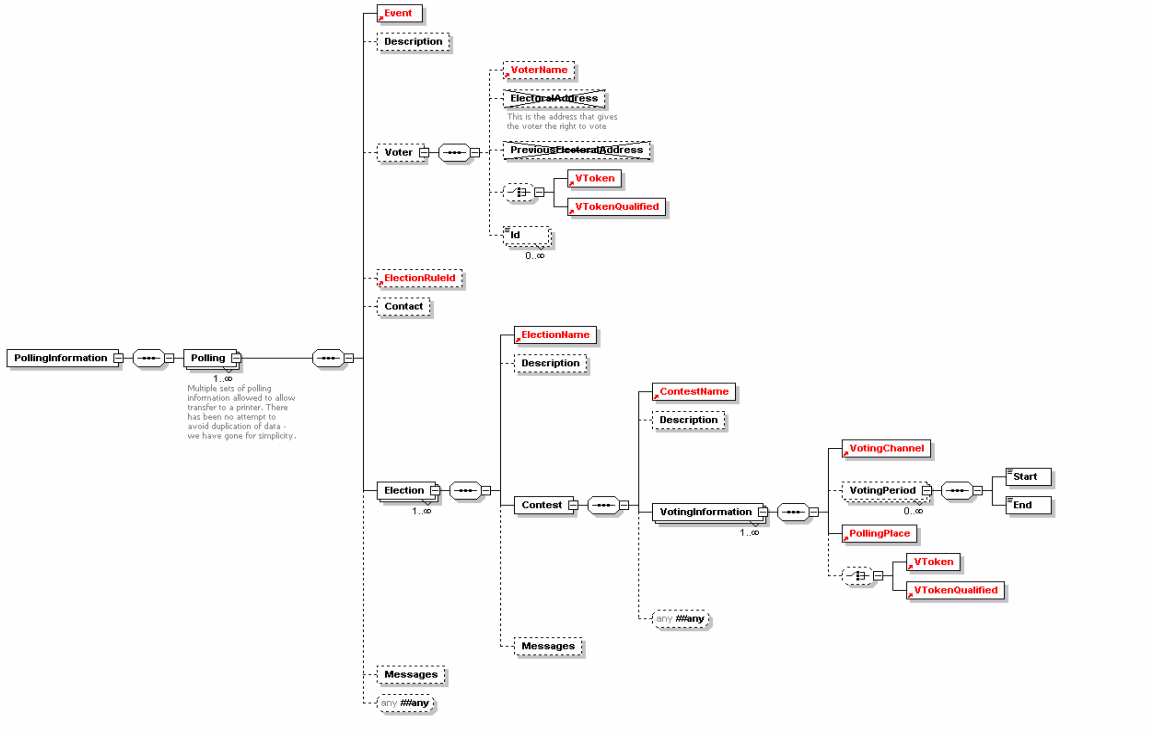
1294 ## 6.4.8 Polling Information (340)

1295 The polling information messages defined by this schema are sent to voters to provide them with
1296 details of how to vote. It can also be sent to a distributor, so multiple sets of information are
1297 allowed.

1298 One set of polling information may be sent to each voter for any election event, so the election
1299 event name is included, with the polling start and end time. Some information about the voter may
1300 be included, for example to print on a polling card.

1301 The `ElectionRuleId` can be included, and contact information for the benefit of a distributor.

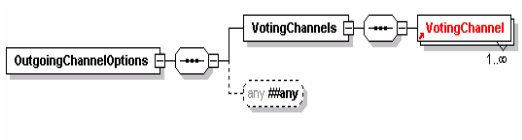
Information about the elections and contests is included for the benefit of the voter, and further messages might be added. Use of the `DisplayOrder` attribute on these allows the display or printing of information to be tailored from within the XML message.



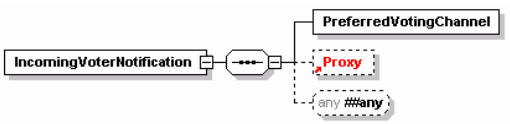
6.4.9 Generic Communication (350)

These two schemas (350a and 350b) extend the two corresponding data types by allowing any additional element to be appended.

6.4.10 Channel Options (360)



360a - Outgoing



360b - Incoming

These two schemas are used for messages offering a set of voting channels to the voter and to indicate a preferred channel. 360b may be sent as an unsolicited message if this is supported within the relevant jurisdiction.

Both are extensions of the corresponding generic communications data type. The outgoing message includes a list of allowed channels, and the incoming provides a single channel.

Either message can be extended in the normal way.

6.4.11 Ballots (410)

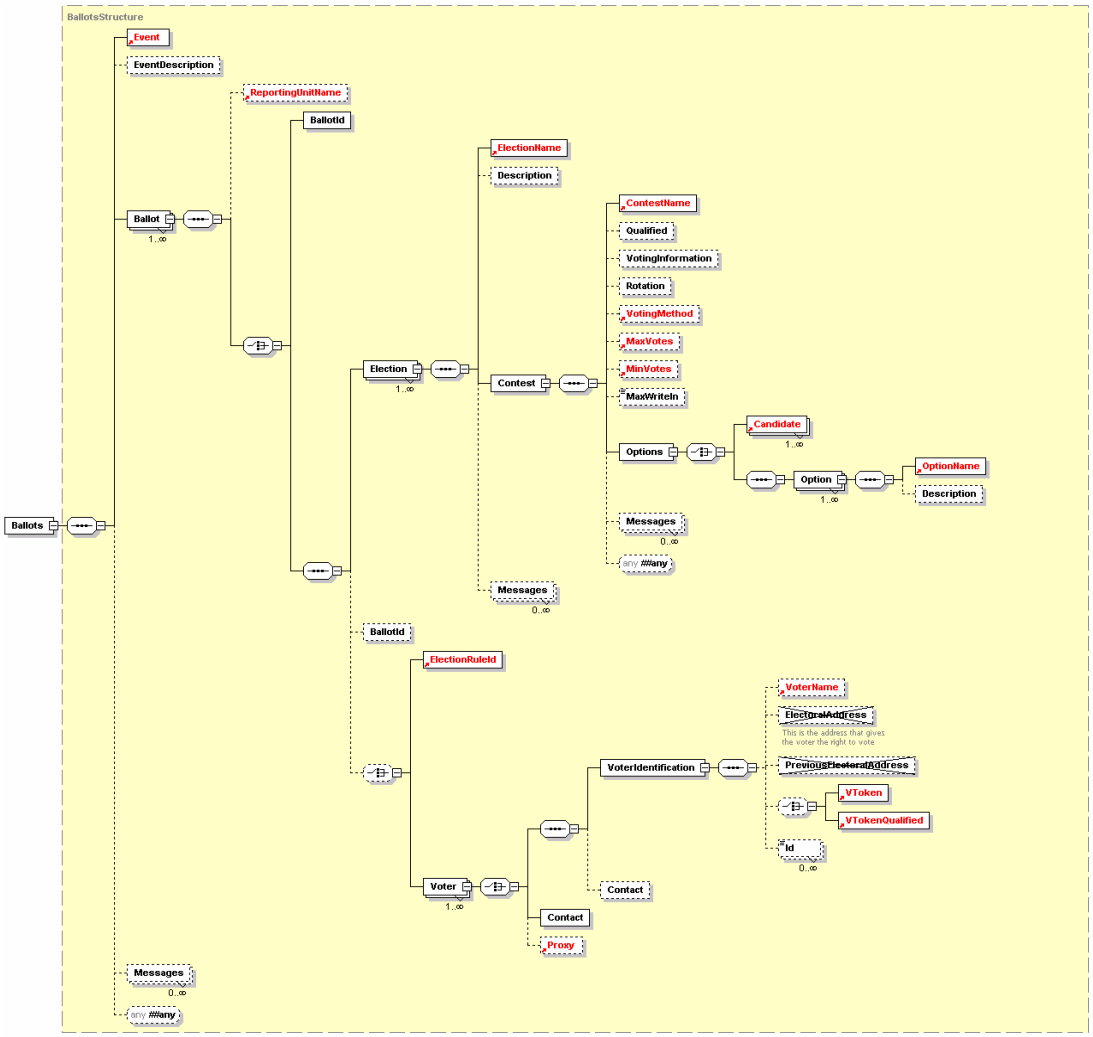
This schema is used for messages presenting the ballot to the voter or providing a distributor with the information required to print or display multiple ballots.

In the simplest case, a distributor can be sent information about the election event and a ballot ID to indicate the ballot to print.

In other cases, the full information about the elections will be sent with either an election rule ID to identify the voters to whom that election applies or a set of voter names and contact information. If the ballot is being sent directly to the voter, this information is not required.

The election information starts with the election event name and description. This is followed by information related to the contest and any other messages and information required. Note that each voter can only vote in a single contest per election, so only a single iteration of the `Contest` element is required.

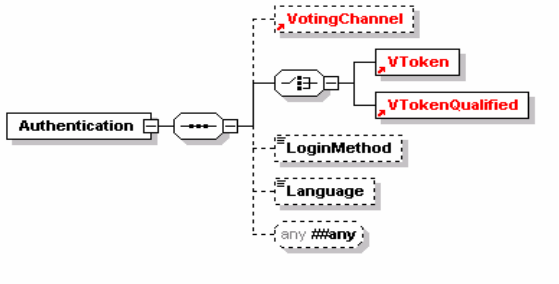
A contest must have its name and ID and a list of options for which the voter can vote. There is also a set of optional information that will be required in some circumstances. Some of this is for display to the voter (`VotingInformation` and `Messages`) and some controls the ballot and voting process (`Rotation`, `VotingMethod`, `MaxVotes`, `MinVotes`, `MaxWriteIn`).



6.4.12 Authentication (420)

The authentication message defined by this schema may be used to authenticate a user during the voting process. Depending on the type of election, a voter's authentication may be required; the precise mechanism used may be channel and implementation specific. In some public elections the voter must be anonymous, in which case the prime method used for authentication is the voting token. The voting token can contain the information required to authenticate the voter's right to vote in a specific election or contest, without revealing the identity of the person voting. Either the VToken or the VTokenQualified must always be present in an authenticated message.

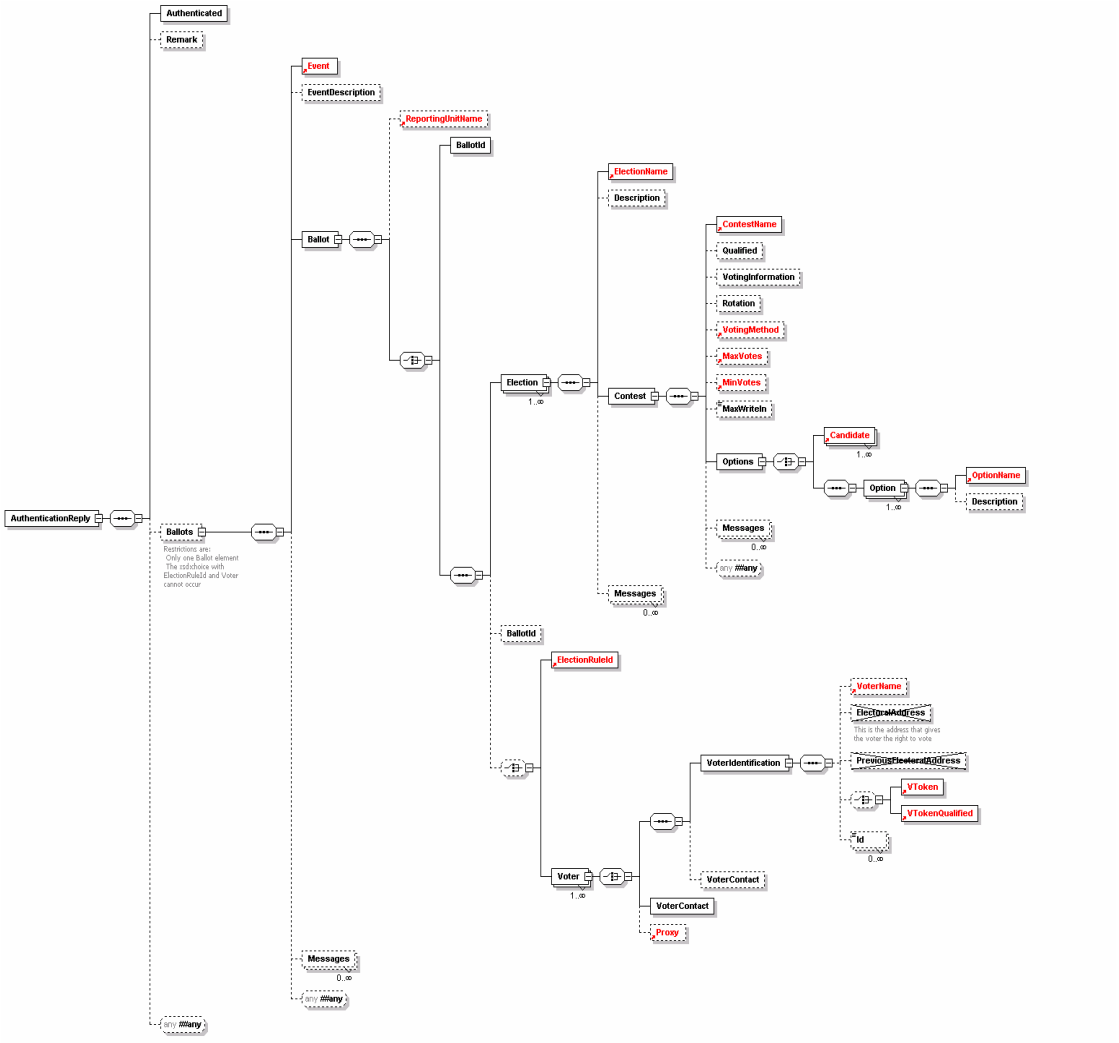
The other authentication elements are optional. The TransactionId is used to collate an authentication message with an authentication reply, the VotingChannel identifies the channel by which the voter has been authenticated, the LoginMethod allows additional information to be added about any channel specific authentication method used. Language and corresponds to the general description of that element.



1348

1349 ## 6.4.13 Authentication Reply (430)

1350 The authentication reply is a response to message 420. It indicates whether authentication
1351 succeeded using the `Authenticated` element, and might also present the ballot to the user.
1352 This is a restriction of the previous `Ballots` element to allow only a single ballot per reply.



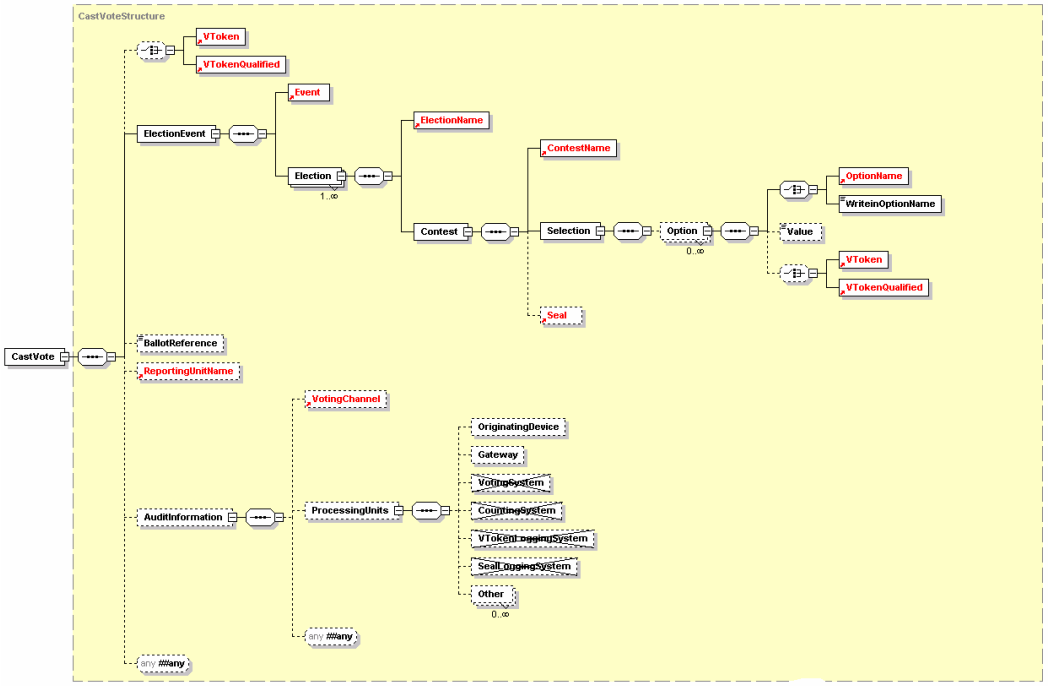
1353

6.4.14 Cast Vote (440)

This message represents a cast vote, which comprises an optional voting token (which may be qualified) to ensure authorisation, information about the votes themselves, the name and ID of the reporting unit if applicable and a set of optional audit information.

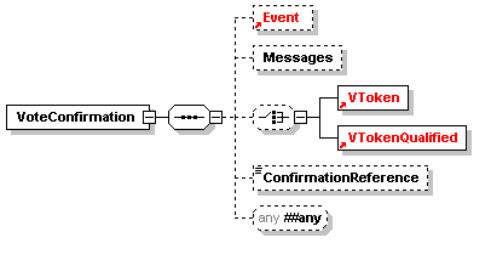
The election event is identified, together with a set of elections (if multiple elections were included on the same ballot). For each election, the contest is identified, with a set of, possibly sealed, votes. The votes are sealed at this level if there is a chance that the message will be divided, for example so that votes in different elections can be counted in different locations.

For each contest, one or more options is listed. For each of these, either the option name and ID is provided or a write-in option name for elections where this is allowed. This is accompanied by the value of the vote for that option, with an optional voting token (which, again, may be qualified). In some elections where it is only possible to vote for a single candidate, different voting tokens may be provided for each option. In this case, only the voting token is required.



6.4.15 Vote Confirmation (450)

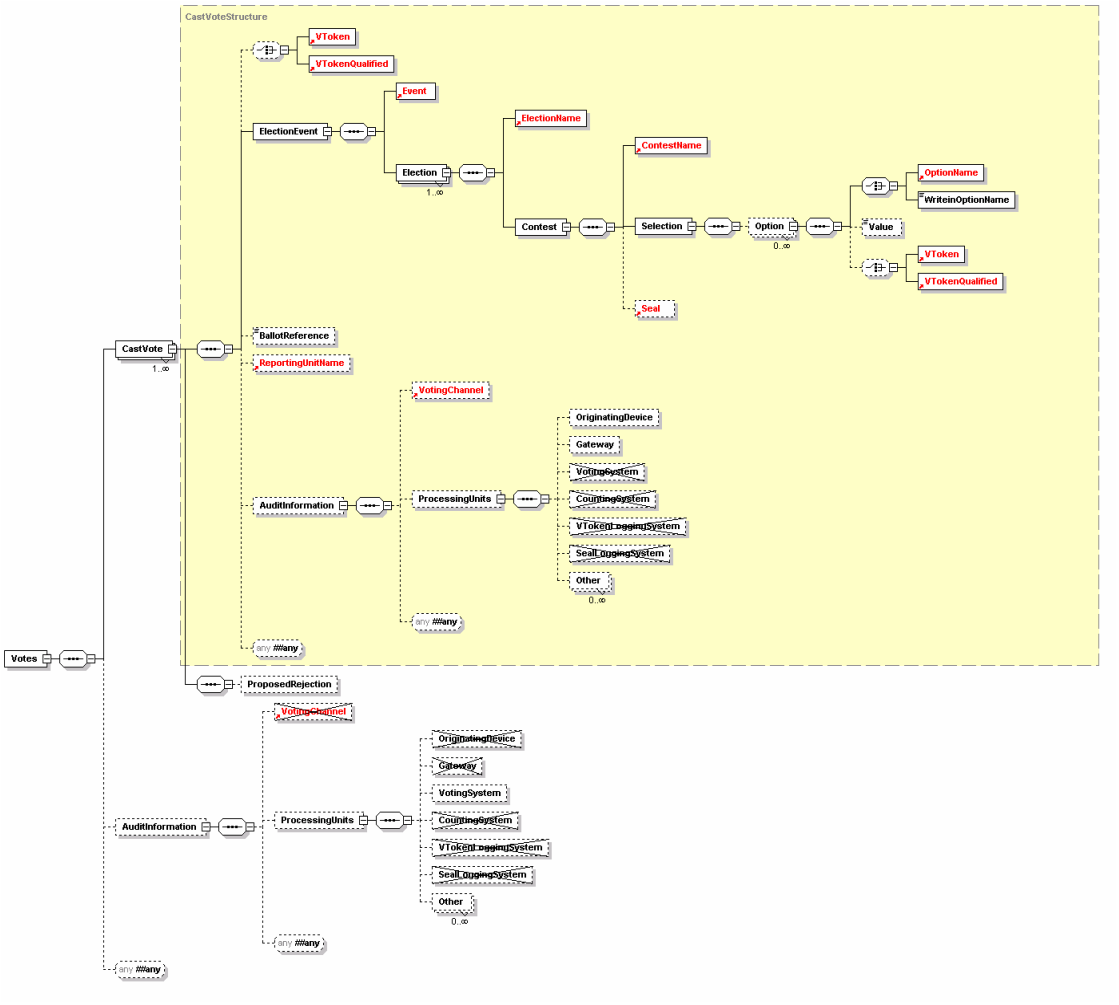
The vote confirmation message can be used to show that a vote has been accepted and provide a reference number in case of future queries. Display information can also be provided as well as additional structured information using `xsd:any`.



6.4.16 Votes (460)

This schema is used to define a message comprising a set of votes being transferred for counting. It is a set of `CastVote` elements from schema 440 with the addition of audit information for the voting system.

The message defined by this schema is used to add a voting token (which may be qualified) to an audit log. The `VToken` or `VTokenQualified` is extended by the addition of a `Status` attribute with a value of `voted` or `unvoted`. In addition to sending single tokens as they are used, the schema can be used to validate a message sending multiple tokens optionally grouped by voting channel. This might be used instead of sending tokens as they used or, for example, to send the unused tokens at the end of an election. The logging system can also be identified for audit purposes.



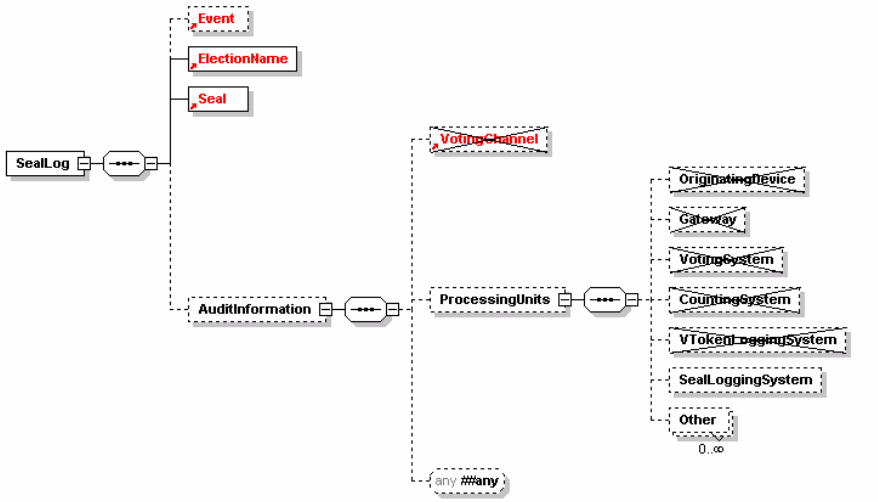
1383

1384

1385 ### 6.4.17 Seal Log (480)

1386 The message defined by this schema is used to log the use of each seal for audit purposes.
1387 There must be one message per seal, so, if multiple votes are sealed individually in one cast vote
1388 message, twp seal log messages must be generated.

1389 The message contains the name and ID of the election, the seal itself and possibly additional
1390 audit information as defined in section 7.1.2.



1391

1392 ### 6.4.18 Count (510)

1393 The count message defined by this schema is used to communicate the results of the sets of
1394 contests that makes up one or more elections within an election event. It may also be used to
1395 communicate the result of a single reporting unit for amalgamation into a complete result.

1396 The message therefore includes the election event name and ID, and for each election, the

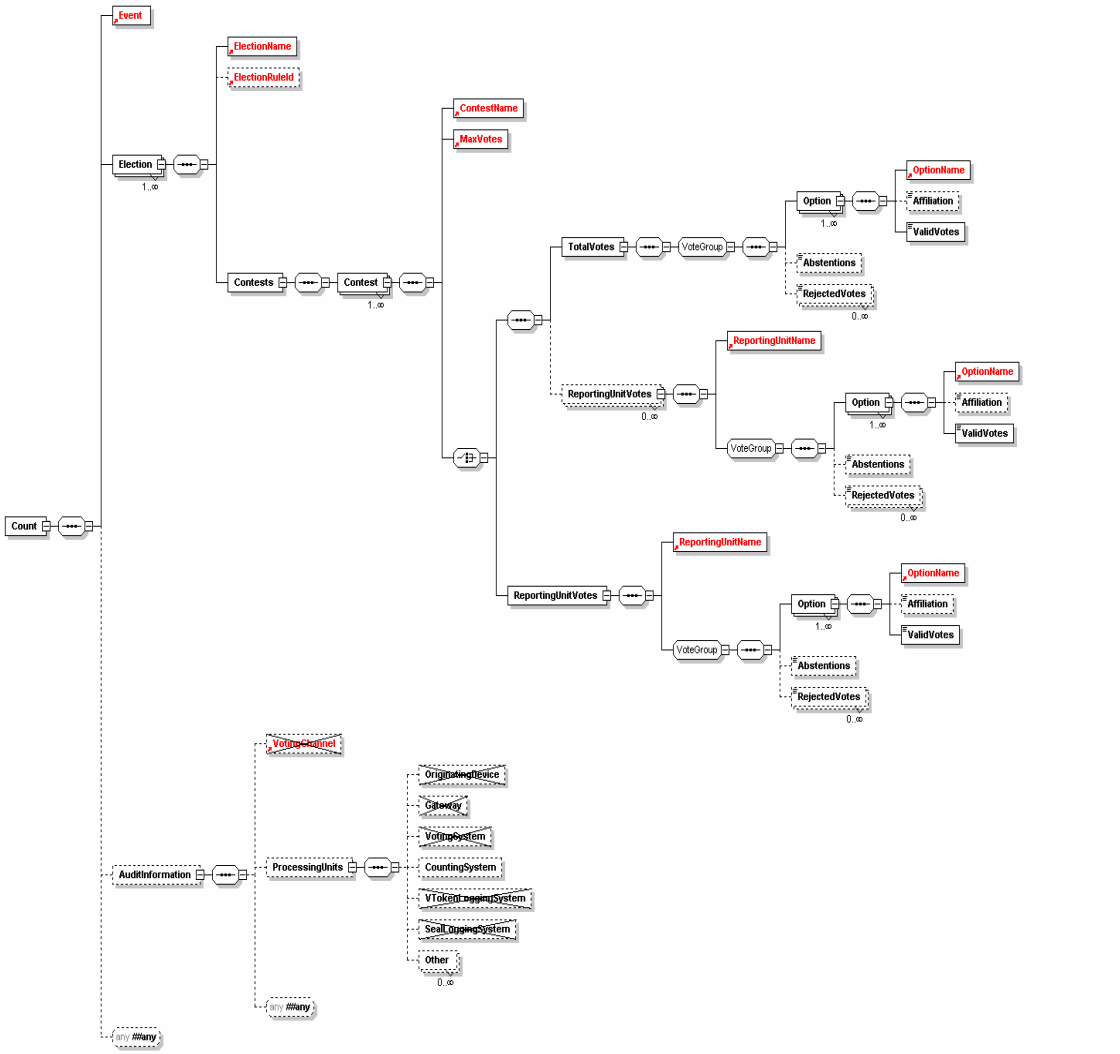
1397 election ID, a reference to the election rule being used and information concerning the set

1398 of contests. The counting system is may also be identified for audit purposes.

1399 In some cases, reporting for a contest may be required at a lower level (for example, for each
1400 county in a state). For this reason, reporting may be done at the level of the reporting unit, the
1401 total votes, or for a total vote and the breakdown according to the multiple reporting units.

1402 Each contest indicates its name and ID, the maximum number of votes that each voter could
1403 cast, information about the votes cast for each option and the numbers of abstentions and
1404 rejected votes. The `RejectedVotes` element has `Reason` (optional) and `ReasonCode`
1405 (mandatory) attributes to indicate why the votes were rejected. The former is a textual description,
1406 and the latter a code.

1407 For each option, the name, ID and number of valid votes is mandatory. These are optionally
1408 supplemented by an affiliation when the option is a (human) candidate.



1409

References

1 **eXtensible Name and Address (XNAL) Specifications and Description Document (v1.0)** ***Customer Information Quality Technical Committee*** **OASIS** ***8 May 2001*** **http://www.oasis-open.org/committees/ciq/xnal/xnal_spec.zip**

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 (Second Edition)** ***Tim Bray et al*** **Worldwide Web Consortium** ***6 October 2000*** **http://www.w3.org/TR/REC-xml**

4 **XML Linking Language (XLink) (v1.0)** ***Steve DeRose et al*** **Worldwide Web Consortium** ***27 June 2001*** **http://www.w3.org/TR/xlink/**

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

6 **Voice Extensible Markup Language (VoiceXML) Version 2.0** ***Scott McGlashan et al*** **Worldwide Web Consortium** ***23 October 2001*** **http://www.w3.org/TR/voicexml20**

7 **XML Schema Part 2: Datatypes** ***Paul V Biron et al*** **Worldwide Web Consortium 2 May 2001 http://www.w3.org/TR/xmlschema-2/**

Appendix A: Glossary/Terminology

E-VOTING TERMS

The table below contains a list of voting terms used within this process document. The entries in bold relate to core terms that have been centrally defined by the committee and are essential to understanding the use of terminology within this document.

Additional suggestions from committee members have also been included.

TERM	DEFINTION	ORIGIN
BALLOT	Appropriate to one voter and will contain the set of candidates or options for a particular contest within one or more elections.	E&VSTC
BALLOT FORMAT	A format for rendering a ballot	USA
BALLOT LAYOUT	A template for a physical ballot	USA
BALLOT MESSAGE	Fixed text, image, instructions, etc. that appears on a ballot page	USA
BALLOT STYLE	Unique combination of contest and candidates	USA
CANDIDATE	**An individual in standing in a contest or one of a set of proposal on an issue [See option]**	E&VSTC
CANDIDATE LIST	**A list of candidates or issues involved in a contest.**	E&VSTC
CAST VOTE	**This is a ballot containing the voters Preferences**	E&VSTC
CONSTITUENCY	The whole area to which the elective office relates and may include a number of POLLING DISTRICTS	UK
CONTEST	**A competition between a set of candidates for a particular post or on a particular issue**	E&VSTC
Election EVENT	**An election event is a series of elections that for some reason are grouped together into one event. For example they may be completely different elections but for logistic reason they are all run on the same day.**	E&VSTC
ELECTION	**An election is used in the traditional sense, such as a country's government election, local government election, or other local community elections. An election comprises a collection of related contests over a defined period of time. A series of elections may, or may not, be combined into one ballot for a voter within an election event.**	E&VSTC
FOOTER	Text, image, or other detail that appears immediately after	USA

TERM	DEFINTION	ORIGIN
	a contest or candidate listing	
HEADER	Text, image, or other detail that appears immediately before a contest or candidate listing	USA
ITEM	The thing voted upon whether it is an office, position-elect or referendum	USA
ITEM_TYPE	Describes the type of ITEM (such as first-past-the-post, plurality, proportional vote, etc	USA
POLL SITE INTERNET VOTING	This refers to the casting of ballots at public sites where election officials control the voting platform	US
REMOTE INTERNET VOTING	This refers to the casting of ballots at private sites, where the voter or a third party controls the voting client.	US
NON-VOTER	Someone either who is on the register but has not voted, or someone who is ineligible to vote on Age or other grounds	UK
OPTION	**The options are the choices presented to a voter for a particular contest and can comprise the list of candidates, choices, answers, etc.**	E&VSTC
PARTY AFFLIATION	Political party affiliation associated to a CONTEST or CANDIDATE	USA
POLLING DISTRICT	The smallest geographical entity within which the VOTERS are subdivided for registration and voting purposes	UK
POLLING DISTRICT	A specific geo-political area that defines a boundary for a BALLOT CONTEST	USA
POLLING DISTRICTS SPLIT	Unique combination of all DISTRICTS in a specific jurisdiction	USA
REPORTING UNIT	**A sub-unit within a CONTEST.**	**E&VSTC**
ROTATION	The concept of presenting candidates (for the same contest) in a different order for different ballots	USA
SELECTION	The CANDIDATE, answer, etc which is the option or choice for ELECTION	USA
SEQUENCE	Order in which a CANDIDATE or CONTEST appears on a BALLOT	USA
UNDERVOTE	Indicates whether it is allowable to VOTE for fewer than the allowable SELECTIONS	USA
VOTE	A positive act, which records the voter's choice of CANDIDATE but in such a way as to ensure the secrecy of the BALLOT	UK

TERM	DEFINTION	ORIGIN
VOTELIMIT	Defines the number of vacancies to be filled in a particular item	USA
VOTER	**A voter is someone who is on the election list**	E&VSTC
WRITEIN	Describes the number of write in CANDIDATES allowed	USA

E-VOTING PROCESS TERMINOLOGY

PROCESS	DEFINITION	ORIGIN/LINKS
REGISTER VOTER	This involves getting personal data onto the electoral roll	E&VSTC
CANDIDATE NOMINATION	The method of confirming eligibility to be a candidate in a contest and storing the relevant data.	E&VSTC
VOTING PROCESS	This involves the following two activities, the authentication of the voter and the casting of an individual vote.	E&VSTC
COUNTING PROCESS	The process of turning voted ballots into the results of a contest.	E&VSTC
VOTER IDENTIFICATION	The means by which a voter registration system identifies the entity (e.g human) entitled to vote.	E&VSTC
VOTER AUTHENTICATION	The means by which an e-voting system identifies that a voter has the right to cast a vote in a contest.	E&VSTC
VOTE SEALING	The means by which voter authentication and one or more vote can be proved to be related (e.g. possibly the a cryptographic way of sealing together a vote and proof the voter was legitimate).	E&VSTC

Appendix B: 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.***	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.
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 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	Incorrect identification during	Trusted candidate identification and

Concerns raised on Internet voting	Resulting Technical Threats	Possible generic security service countermeasure
presented with correct ballot information due to incorrect candidate identification.	candidate registration.	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

Concerns raised on Internet voting	Resulting Technical Threats	Possible generic security service countermeasure
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 V-token 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 V-token data.	
6:Voting system impersonation	Inadequate authentication of registration system Inadequate authentication of voting casting point (e.g. polling station/ballot box)	Channel to provide: Point to point authentication Channel to provide: Point to point authentication

Concerns raised on Internet voting	Resulting Technical Threats	Possible generic security service countermeasure
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

Concerns raised on Internet voting	Resulting Technical Threats	Possible generic security service countermeasure
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

Concerns raised on Internet voting	Resulting Technical Threats	Possible generic security service countermeasure
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	Voter's identification is anonymous
19: Denied access to electronic voting station	Denial of service attack	This needs to be counted by engineering the system to provide survivability when under denial of service attack.

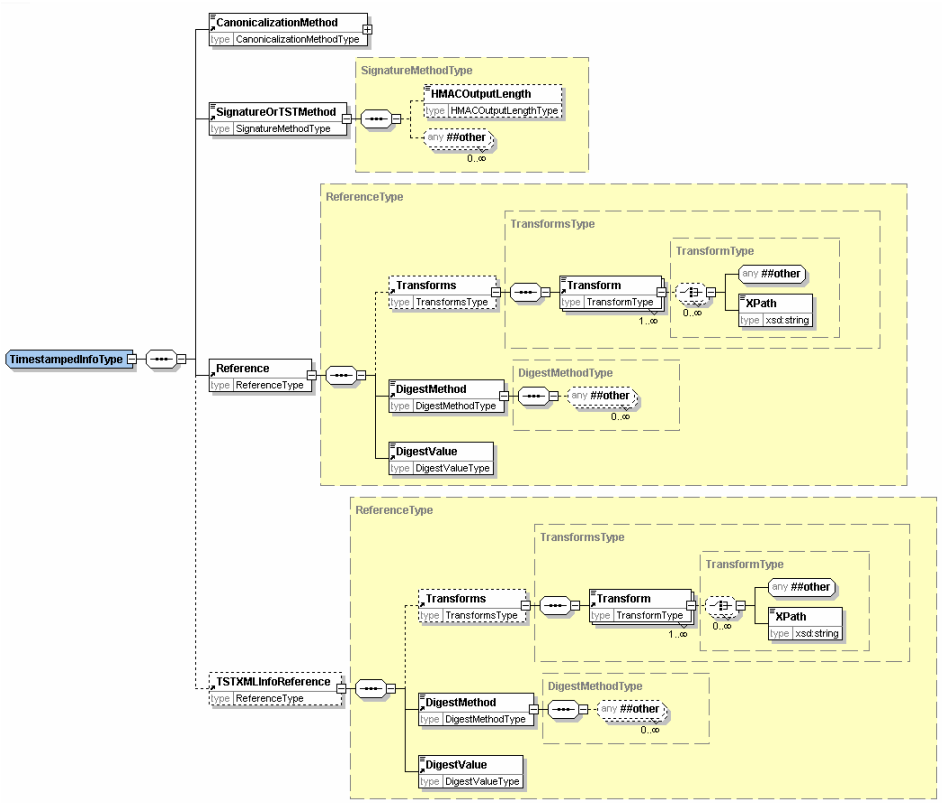
1436

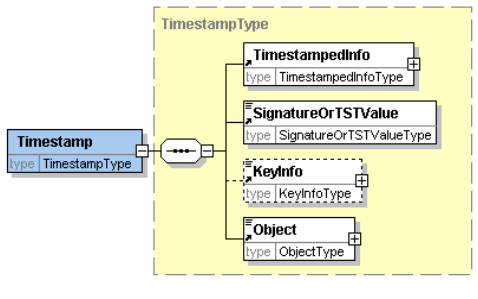
Appendix C: The Timestamp Schema

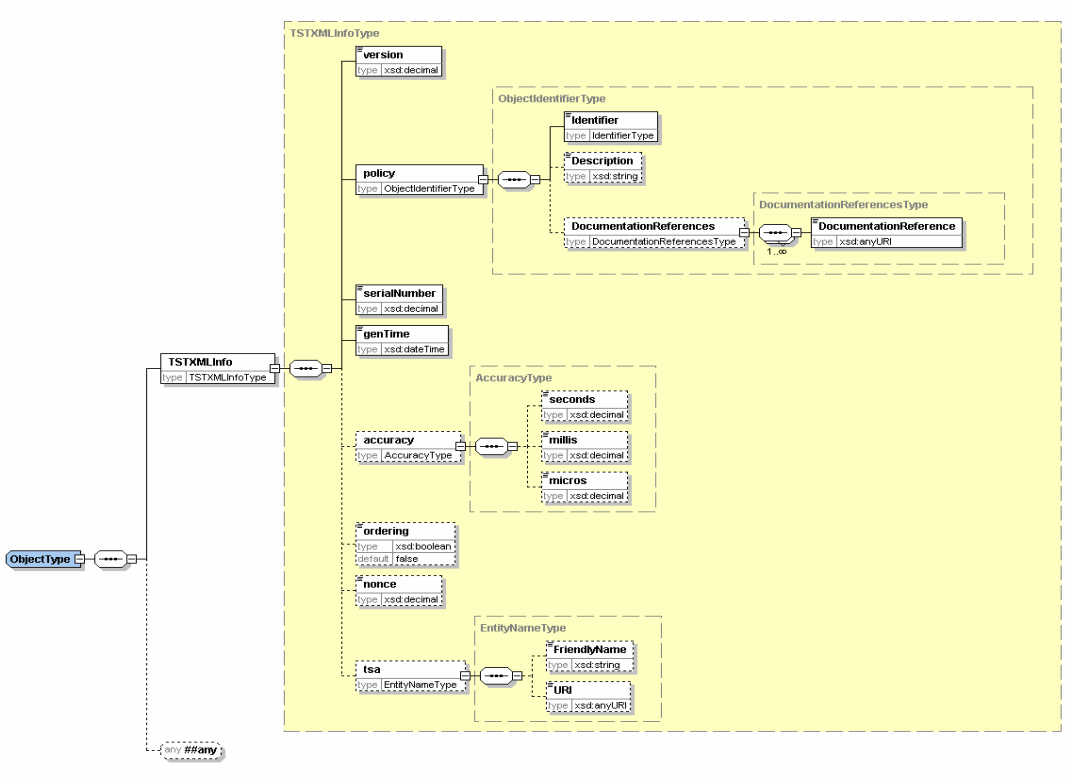
Although used as part of EML, this schema has been put in a separate namespace as it is not an integral part of the language.

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.







1447

1448 The timestamp structure may be used in one of two ways either:

1449 • Using Internet RFC 3161 binary encoded time-stamp token with the time-stamp information
1450 repeated in XML,

1451 • Using a pure XML encoded time-stamp.

1452 In the case of the RFC 3161 based time-stamp, the Timestamp structure is used as follows:

1453 • within `TimestampedInfo`:

1454 • `TSTOrSignatureMethod` identifies RFC 3161.

1455 • `Reference` contains the URI reference of the voting data being time-stamped. The
1456 `DigestValue` sub element contains the digest of the voting data being time-stamped.

1457 • `TSTXMLInfoReference` is not present in this case.

1458 • `SignatureOrTSTValue` holds the RFC 3161 time-stamp token applied to the digest of
1459 `TimestampedInfo`. The `TimestampedInfo` is transformed to a canonical form using the
1460 method identified in `CanonicalizationMethod` before the digest algorithm is applied.

1461 • `KeyInfo` contains any relevant certificate or key information.

1462 • `Object` contains the `TSTXMLInfo` element which is a copy of the information in
1463 `SignatureOrTSTValue` converted from RFC 3161 to XML encoding. The `TSTXMLInfo`
1464 element contains:

1465 o version of time-stamp token format. This would be set to version 1

1466 o 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 D: W3C XML Digital Signature

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

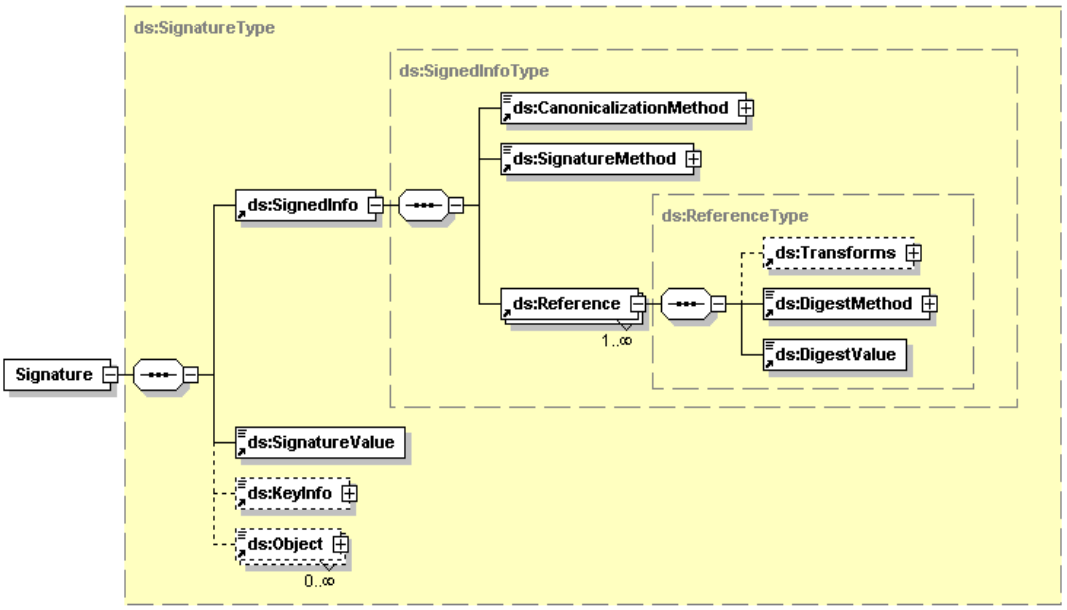
An XML Signature consists of:

`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



1514 # 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 for Technical Committee approval
V3	2003-02-06	Committee Specification

1515

Appendix F: 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.