

# Election Markup Language (EML) Specification Version 7.0

## Committee Specification 01

27 October 2011

### Specification URIs

#### This version:

<http://docs.oasis-open.org/election/eml/v7.0/cs01/eml-v7.0-cs01.pdf> (Authoritative)  
<http://docs.oasis-open.org/election/eml/v7.0/cs01/eml-v7.0-cs01.html>  
<http://docs.oasis-open.org/election/eml/v7.0/cs01/eml-v7.0-cs01.doc>

#### Previous version:

N/A

#### Latest version:

<http://docs.oasis-open.org/election/eml/v7.0/eml-v7.0.pdf> (Authoritative)  
<http://docs.oasis-open.org/election/eml/v7.0/eml-v7.0.html>  
<http://docs.oasis-open.org/election/eml/v7.0/eml-v7.0.doc>

#### Technical Committee:

OASIS Election and Voter Services TC

#### Chair:

John Borrás ([johnaborras@yahoo.co.uk](mailto:johnaborras@yahoo.co.uk)), Individual

#### Editors:

John Borrás ([johnaborras@yahoo.co.uk](mailto:johnaborras@yahoo.co.uk)), Individual  
David Webber ([david.webber@oracle.com](mailto:david.webber@oracle.com)), Oracle

#### Additional artifacts:

This prose specification is one component of a Work Product which also includes:

- XML schemas:  
<http://docs.oasis-open.org/election/eml/v7.0/cs01/Schemas/>  
<http://docs.oasis-open.org/election/eml/v7.0/cs01/external/>
- XML Dictionary and spreadsheet artifacts:  
<http://docs.oasis-open.org/election/eml/v7.0/cs01/dictionary/>

#### Related work:

This specification replaces or supersedes:

- *Election Markup Language (EML) Specification Version 6.0*. 19 August 2010. OASIS Committee Specification 01. <http://docs.oasis-open.org/election/eml/v6.0/cs01/EML-Specification-v6.0-cs01.html>

#### Declared XML namespace:

`urn:oasis:names:tc:evs:schema:eml`

#### Abstract:

This document describes the background and purpose of the Election Markup Language, the electoral processes from which it derives its structure and the security and audit mechanisms it is designed to support. It also provides an explanation of the core schemas used throughout, definitions of the simple and complex datatypes, plus the EML schemas themselves. It also

covers the conventions used in the specification and the use of namespaces, as well as guidance on the constraints, extensibility, and splitting of messages.

**Status:**

This document was last revised or approved by the OASIS Election and Voter Services TC on the above date. The level of approval is also listed above. Check the “Latest version” location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee’s email list. Others should send comments to the Technical Committee by using the “Send A Comment” button on the Technical Committee’s web page at <http://www.oasis-open.org/committees/election/>.

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 Technical Committee web page (<http://www.oasis-open.org/committees/election/ipr.php>).

**Citation format:**

When referencing this specification the following citation format should be used:

**[EML-v7.0]**

*Election Markup Language (EML) Specification Version 7.0*. 27 October 2011. OASIS Committee Specification 01. <http://docs.oasis-open.org/election/eml/v7.0/cs01/eml-v7.0-cs01.html>.

---

## Notices

Copyright © OASIS Open 2011. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full [Policy](#) may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

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' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on 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 implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of [OASIS](#), the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

---

# Table of Contents

1	Introduction.....	6
1.1	Terminology.....	6
1.2	Normative References.....	6
1.3	Non-Normative References.....	6
1.4	Background.....	6
1.5	Overview of the Document.....	7
1.6	Changes in this Version.....	8
2	Requirements.....	9
2.1	Business Drivers.....	9
2.2	Technical Drivers.....	9
2.3	The E&VS Technical Committee.....	9
2.4	Challenge and Scope.....	10
2.5	Documentation Set.....	11
2.6	Voting Terminology.....	12
3	High Level Election Processes.....	14
3.1	Outline.....	15
3.2	Process Descriptions.....	16
3.2.1	The Candidate Nomination Process.....	16
3.2.2	The Options Nomination Process.....	17
3.2.3	The Voter Registration.....	18
3.2.4	The Electronic Ballot Delivery Process.....	20
3.2.5	The Voting Process.....	20
3.2.6	The Vote Report Process.....	22
3.2.7	The Auditing Process.....	23
3.3	Data Requirements.....	24
4	Schema Outline.....	27
4.1	Structure.....	27
4.2	Viewing Schemas.....	27
4.3	IDs.....	27
4.4	Displaying Messages.....	28
4.5	EML Message Validation.....	30
4.6	Namespaces.....	31
4.7	Extensibility.....	31
4.8	Additional Constraints.....	31
4.9	Metadata.....	31
4.10	Splitting of Messages.....	31
4.11	Error Messages.....	32
4.12	All Schemas.....	32
4.12.1	XML Well-Formedness or Schema Validation Error.....	32
4.12.2	Seal Errors.....	32
4.12.3	EML Additional Rules.....	32
5	Schema Descriptions.....	35
5.1	Overview.....	35

5.2 EML Core Components .....	36
5.2.1 Complex Data Types .....	36
5.3 Message Headers .....	36
5.4 Message Schemas .....	36
Election Event (110) .....	37
Response (130) .....	38
GeoDistrict (150) .....	38
Candidate Nomination (210) .....	38
Response to Nomination (220).....	39
Candidate List (230) .....	39
Voter Registration (310) .....	39
Election List (330).....	39
Polling Information (340) .....	40
Outgoing Generic Communication (350a).....	40
Incoming Generic Communication (350b).....	40
Internal Generic (350c).....	41
Outgoing Channel Options (360a).....	41
Incoming Channel Options (360b).....	41
Ballots (410) .....	41
Authentication (420) .....	41
Authentication Response (430) .....	42
Cast Vote (440) .....	42
Retrieve Vote (445) .....	42
Vote Confirmation (450) .....	42
Votes (460) .....	42
VToken Log (470).....	43
Audit Log (480) .....	43
Ballot Delivery (505) .....	44
Count (510).....	44
Result (520) .....	45
Statistics (530).....	45
Options Nomination (610) .....	45
Options Nomination Response (620) .....	45
Options List (630) .....	46
6 Conformance .....	47
Appendix A. Acknowledgements .....	48
Appendix B. Other Considerations .....	49
B.1 Security.....	49
B.2 Internet Voting Security Concerns .....	56
B.3 The Timestamp Schema .....	59
B.4 W3C XML Digital Signature.....	60
Appendix C. Processing using Schematron or CAM .....	62
Appendix D. Revision History .....	64

---

# 1 Introduction

## 1.1 Terminology

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

## 1.2 Normative References

- [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.

## 1.3 Non-Normative References

- [xNAL] *Extensible Name and Address Language (xNAL) draft 4.0*, May 2011 OASIS Committee Specification Draft 04  
[www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=ciq](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ciq)
- [UK's APD] Address and Personal Details Fragment v2.0 Technology Policy Team, e-Government Unit, Cabinet Office UK, 21 March 2005  
[http://interim.cabinetoffice.gov.uk/govtalk/schemasstandards/xmlschemas/schemalibrary/address\\_and\\_personal\\_details.aspx](http://interim.cabinetoffice.gov.uk/govtalk/schemasstandards/xmlschemas/schemalibrary/address_and_personal_details.aspx)
- [XML] Extensible Markup Language (XML) 1.0 (Third Edition) Tim Bray et al, Worldwide Web Consortium, 4 February 2004 <http://www.w3.org/TR/REC-xml>
- [XML-DSig] XML-Signature Syntax and Processing Donald Eastlake et al, Worldwide Web Consortium, 10 June 2008 <http://www.w3.org/TR/xmlsig-core/>
- [IEEE/P1622/D1] Draft Standard for Electronic Distribution of Blank Ballots for Voting Systems, IEEE/P1622 committee, 13 June 2011

The text in the remainder of this section **1 Introduction** is for information only and is neither normative nor part of the Election Markup Language. For the purpose of this document the term “e-voting” is used to refer to any part of e-enabled elections or referendums, it does not refer just to the casting of votes using electronic means.

## 1.4 Background

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 original objective in 2001 was to introduce a uniform and reliable way to allow systems involved in the election process to interact. The overall focus today provides a rich standard that is:

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

- 42 • **Technology agnostic:** technologically stable and forward deployable with backward feature  
43 compatibility

44 The primary deliverable of the committee is the Election Markup Language (EML). This is a set of data  
45 and message definitions described as XML schemas along with a dictionary of core terms and structures  
46 that enable predictable and consistent foundation mechanisms. The messages that form EML are  
47 intended for transfer between systems. It is not intended that all aspects of an election system will have a  
48 corresponding schema. EML is flexible enough to be used for elections and referendums that are  
49 primarily paper-based or that are fully e-enabled.

50 At present EML includes specifications for:

- 51 • Candidate Nomination, Response to Nomination and Approved Candidate Lists
- 52 • Referendum Options Nomination, Response to Nomination and Approved Options Lists
- 53 • Voter Registration information, including eligible voter lists
- 54 • Various communications between voters and election officials, such as polling information, election  
55 notices, district boundaries, polling places, facilities and services provided, eligibility, blank ballot  
56 forms, etc.
- 57 • Ballot information (races, contests, issues, candidates, etc.)
- 58 • Voter Authentication
- 59 • Vote Casting and Vote Confirmation
- 60 • Election counts, statistics and results
- 61 • Audit information pertinent to some of the other defined data and interfaces

62 This document and its accompanying set of schemas and other artifacts do not claim to satisfy the final  
63 requirements of any and all registration or election systems. The specification represents our best current  
64 efforts, knowledge and experience with election systems since 2001. It is incumbent on the users of this  
65 document to identify any requirement gaps, mistakes, inconsistencies or missing data and to propose  
66 corrections or enhancements to the OASIS Election and Voter Services Technical Committee.

## 67 **1.5 Overview of the Document**

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

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

75 **Section 3** describes two complementary high-level process models of an election exercise, based on the  
76 human and technical views of the processes involved. It is intended to identify all the generic steps  
77 involved in the process and highlight all the areas where standardized data is to be exchanged or  
78 referenced. The discussions in this section presents details of how the messages and data formats  
79 detailed in the EML specifications themselves can be used to achieve the goals of open interoperability  
80 between system components. Also contained in this Section are high-level data models showing the  
81 relationships of the data used in the election processes.

82 **Section 4** provides an overview of the approach that has been taken to creating the XML schemas.

83 **Section 5** provides descriptions of the core elements, data types and schemas developed to date.

84 **Section 6** provides the conformance criteria required for implementations to claim conformance to the  
85 EML specification.

86 **Appendices** provide information on internet voting security concerns; use of the EML defined TimeStamp  
87 schema; the W3C Digital Signature technology; and Acknowledgements and a Revision History of this  
88 document.

## 89 **1.6 Changes in this Version**

90 The changes from EML v6.0 that this new version introduces are as follows:

- 91 • updates applied to match requirements for the USA's UOCAVA (Uniform and Overseas Citizens  
92 Assistance in Voting Act) application;
- 93 • updates applied to match new requirements from the Australia Election Commission;
- 94 • update to the dSig schema for latest W3C 2002/2008 specification;
- 95 • update external reference from xNAL v3.0 to xNAL v4.0;
- 96 • new Conformance criteria.

---

## 97 2 Requirements

### 98 2.1 Business Drivers

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

105 Society has become network oriented and citizens, used to the high degree of flexibility in the services  
106 provided by the private sector and in the Internet in particular, are now beginning to set demanding  
107 standards for the delivery of services by governments using modern electronic information systems.

108 The implementation of e-enabled elections and referendums has become globally widespread allowing  
109 increased access to information in the voting process for citizens everywhere and offering the scope for  
110 better verification and oversight for election supervision procedures. Allowing better access to information  
111 with consistent transparency and verification of results across the whole election process helps foster  
112 greater engagement and participation of voters throughout the whole democratic process itself. This also  
113 requires that standards ensure that the process is clear, robust and precisely understood so that  
114 confidence in the results is ensured. Access to a standard process also allows solution vendors to  
115 participate in an open marketplace that stimulates cost effective delivery and adoption of new technology  
116 without obsolescing existing investments.

117 However, it is recognized that more traditional verification methods and oversight will continue to be vital  
118 and in fact more so with the use of technology. Strong democracy requires participation from citizens and  
119 continuous independent monitoring of processes, procedures and outcomes. The OASIS EML standard  
120 seeks to facilitate precisely that transparency, access and involvement for citizens to the election process,  
121 end to end.

### 122 2.2 Technical Drivers

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

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

### 133 2.3 The E&VS Technical Committee

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

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

141 The Committee's mission statement is:

142 *“To develop a standard for the structured interchange of data among hardware, software, and service*  
143 *providers who engage in any aspect of providing election or voter services, be they partly paper-based or*  
144 *fully e-enabled, to public or private organizations. The services performed for such elections and*  
145 *referenda include but are not limited to:*

- 146 • *candidate nomination,*
- 147 • *referendum options nomination,*
- 148 • *voter registration,*
- 149 • *polling places, districting and boundaries*
- 150 • *various communications between voters and elections officials,*
- 151 • *ballot information*
- 152 • *ballot form(s) delivery*
- 153 • *voter authentication*
- 154 • *vote casting and vote confirmation*
- 155 • *election counts, statistics and results.”*

156 The primary function of an e-voting system is to capture voter preferences reliably, securely and report  
157 them accurately with legally requirements for privacy met correctly. Capture is a function that occurs  
158 between ‘a voter’ (individual person) and ‘an e-voting system’ (machine). It is critical that any election  
159 system be able to prove that a voter’s choice is captured correctly and anonymously, and that the vote is  
160 not subject to tampering, manipulation or other frauds.

161 In addition to the business and technical requirements, the committee was faced with the additional  
162 challenges of producing a specification that is:

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

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

## 174 **2.4 Challenge and Scope**

175 The goal of the committee has been to develop an Election Markup Language (EML) for end-to-end use  
176 within the election process. This is a set of data and message definitions described as a set of XML  
177 schemas and covering a wide range of transactions that occurs end-to-end during various phases and  
178 stages of the life cycle of an election. To achieve this, the committee decided that it required a common  
179 terminology and definition of election processes that could be understood internationally. The committee  
180 therefore started by defining the generic election process models described here.

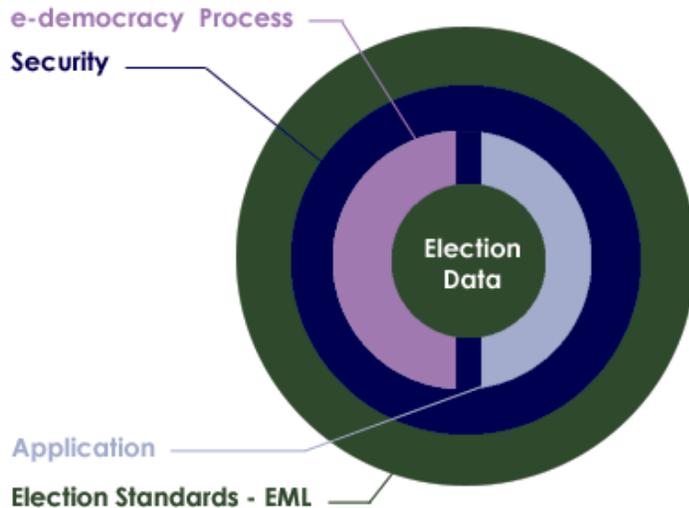
181 These processes are illustrative, covering the vast majority of election types and forming a basis for  
182 defining the Election Markup Language itself. EML has been designed such that elections that do not  
183 follow this process model should still be able to use EML as a basis for the exchange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

185 EML is focused on defining open, secure, standardized and interoperable interfaces between  
186 components of election systems and thereby providing transparent and secure interfaces between  
187 various parts of an election system. The scope of election security, integrity and audit included in these  
188 interface descriptions and the related discussions are intended to cover security issues pertinent only to  
189 the standardized interfaces and not to the internal or external security requirements of the various  
190 components of election systems.

191 The security requirement for the election system design, implementation or evaluation must be placed  
192 within the context of the vulnerabilities and threats analysis of a particular election scenario. As such the  
193 references to security within EML are not to be taken as comprehensive requirements for all election  
194 systems in all election scenarios, nor as recommendations of sufficiency of approach when addressing all  
195 the security aspects of election system design, implementation or evaluation. In fact, the data security  
196 mechanisms described in this document are all optional, enabling compliance with EML without regard for  
197 system security at all. It is anticipated that implementers may develop a complementary document for a  
198 specific election scenario, which refines the security issues defined in this document and determines their  
199 specific strategy and approach by leveraging what EML provides.

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

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



209  
210 *Figure 1A: e-Voting Components Relationship Overview*

## 211 2.5 Documentation Set

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

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

221 The document set comprises:

- 222 • **Specification:** This document. A general and global study of the electoral process. This introduces  
223 the transition from a complete manual election management process to a digitally enabled end-to-end  
224 election system by defining the data structures of content to be exchanged and or produced and  
225 where these data structures are needed, and describe how those exchanges and artifacts are  
226 encoded as XML schemas.

- 227 • **Data Requirements:** A data dictionary defining the data used in the processes and required to be  
228 handled by the XML schemas. The data dictionary is provided in both XML and spreadsheet formats.  
229 In addition there are data models available in the 'EML v7.0 Data Models' file.
- 230 • **EML Schemas:** This consists of a library of the XML schemas used in EML. The XML schemas  
231 define the formal structures of the election data that needs to be processed throughout an election.
- 232 • **EML Core Components Dictionary:** A dictionary containing full definitions of the elements and data  
233 types used by the EML Core schema. The core dictionary is provided in both XML and spreadsheet  
234 formats.
- 235 • **Templates:** for each schema a template is provided that facilitates generation of localizations of the  
236 main schema structure, creation of test case examples and implementation documentation. This  
237 aims to reduce implementer's costs of development and integration.
- 238 • **Models:** one illustrative model is provided for schema 505 and more can be generated from the CAM  
239 templates as desired to suit project localization documentation needs.

## 240 2.6 Voting Terminology

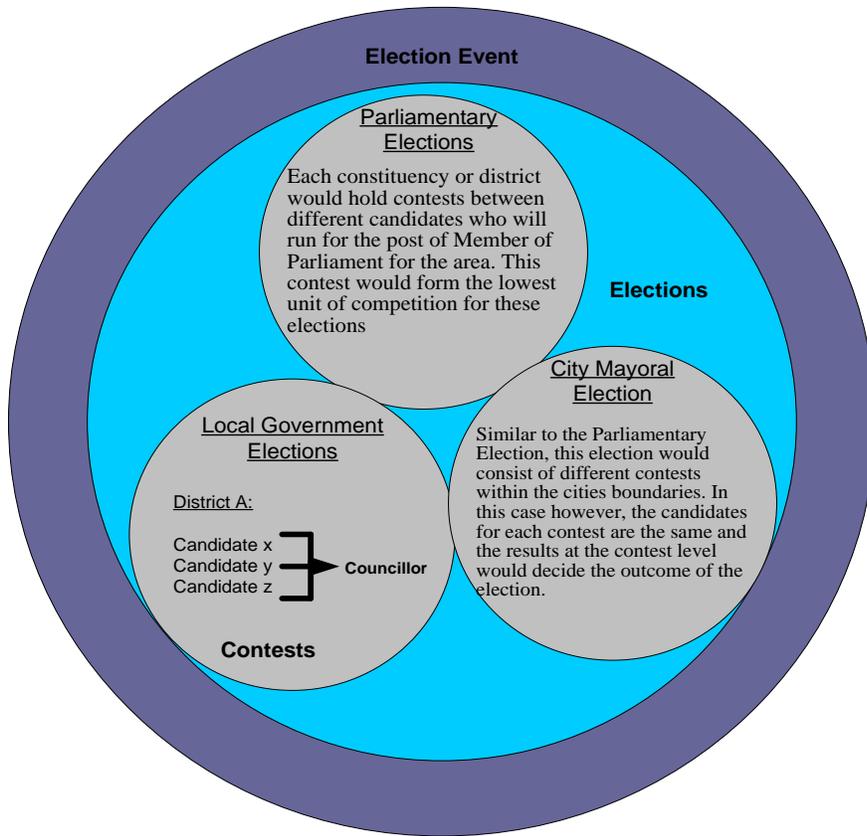
241 At the outset of our work, it was clear that the committee would need to rationalize the different terms that  
242 are commonly used to describe the election process.

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

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

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

252

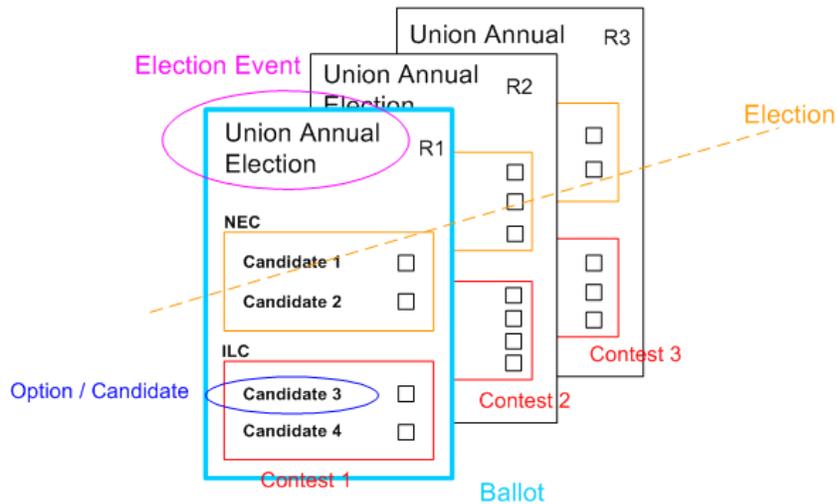


253

254 *Figure 12B: The Election Hierarchy*

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

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



263

264 *Figure 13C: Union Annual Election Event*

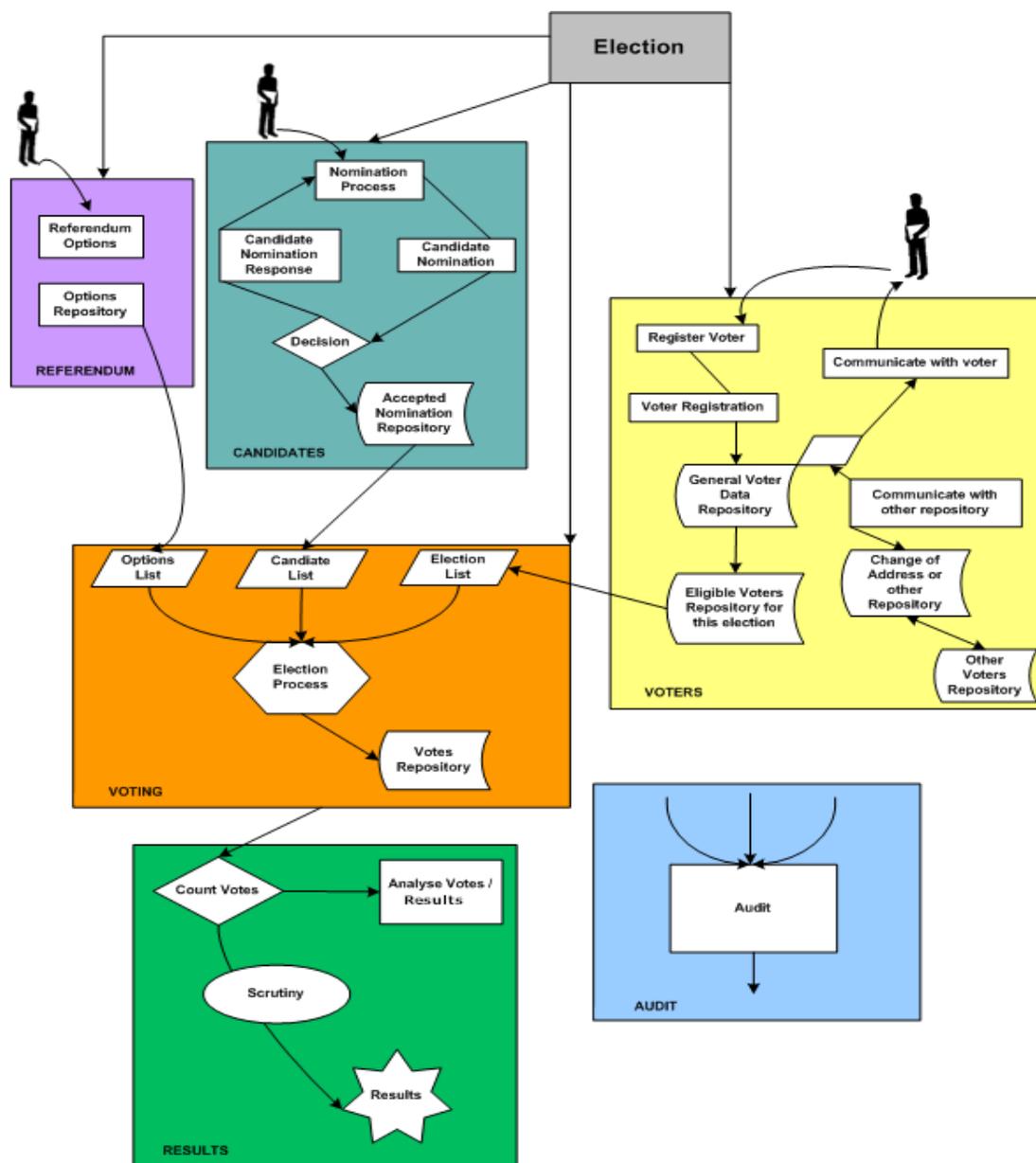
265

### 3 High Level Election Processes

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

269 First two diagrams are presented (Figures 2a and 2B below) that illustrate these process models and then  
270 the section continues by providing details pertaining to the models and illustrative real world processes  
271 they introduce.

272 *Figure 2A: High Level Model – Human View*

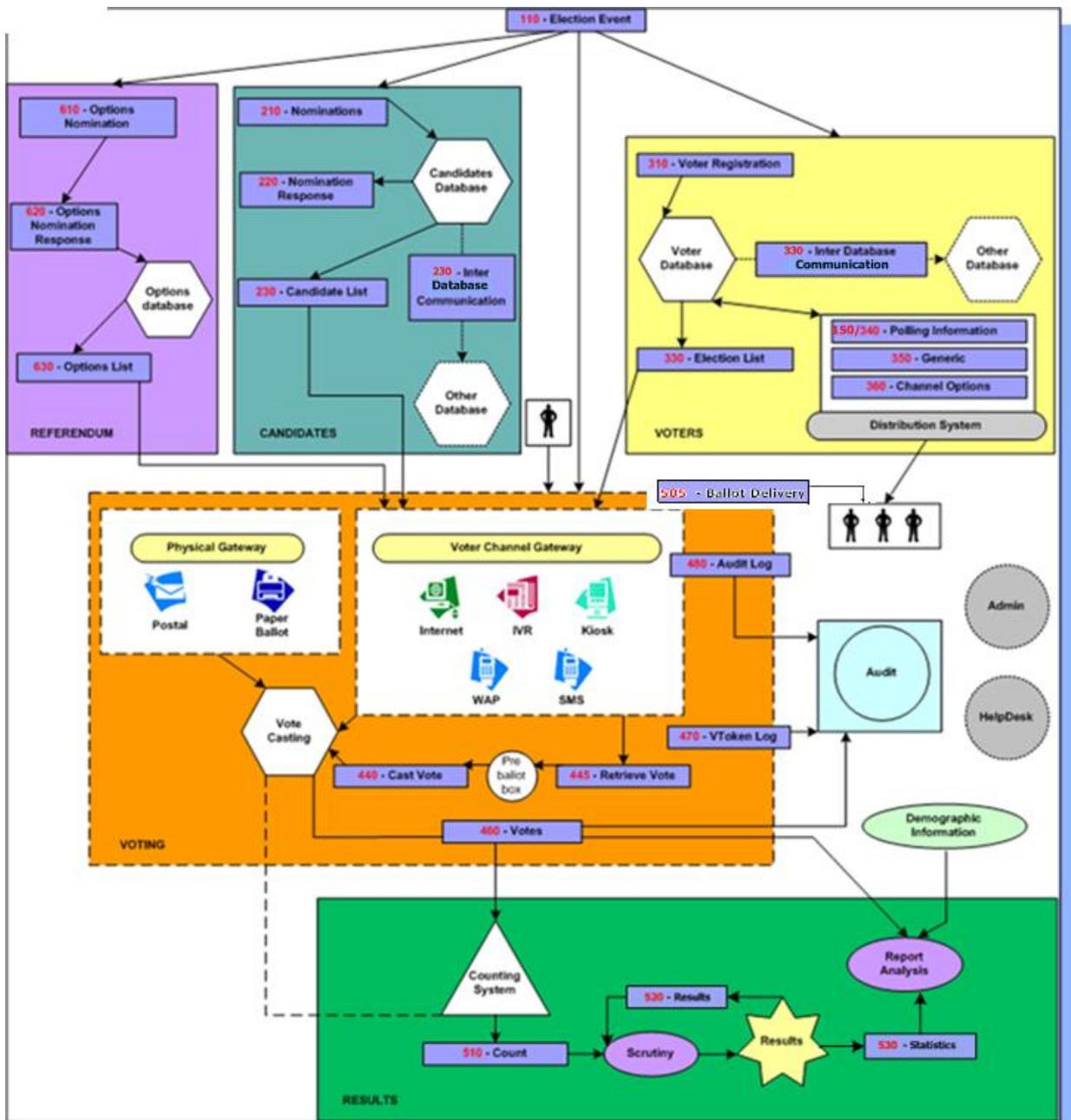


273

274

275

276 *Figure 2B: High Level Model – Technical View*



277  
278

### 279 3.1 Outline

280 This *high-level process model* is derived from real world election experience and is incorporates  
281 knowledge gained over the last few years of refining and improving the specification for EML.

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

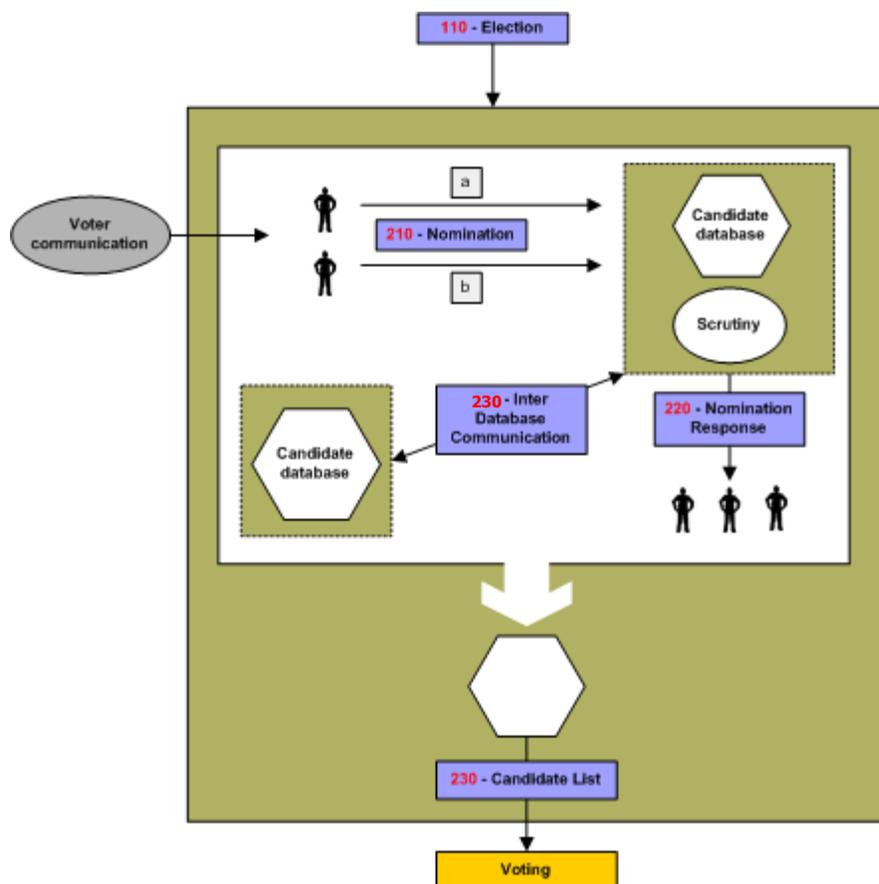
- 286 • Pre election
- 287 • Election (100)

- 288 • Candidates (200)
- 289 • Options (600)
- 290 • Voters (300)
- 291 • Election
- 292 • Voting (400)
- 293 • Post election
- 294 • Results/Reporting/Management (500)
- 295 • Audit
- 296 • Analysis
- 297 Some functions belong to the whole process and not to a specific part:
- 298 • Administration Interface
- 299 • Help Desk

## 300 3.2 Process Descriptions

### 301 3.2.1 The Candidate Nomination Process

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



304  
 305 *Figure 2C: The Candidate Nomination Process*

306

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

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

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

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

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

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

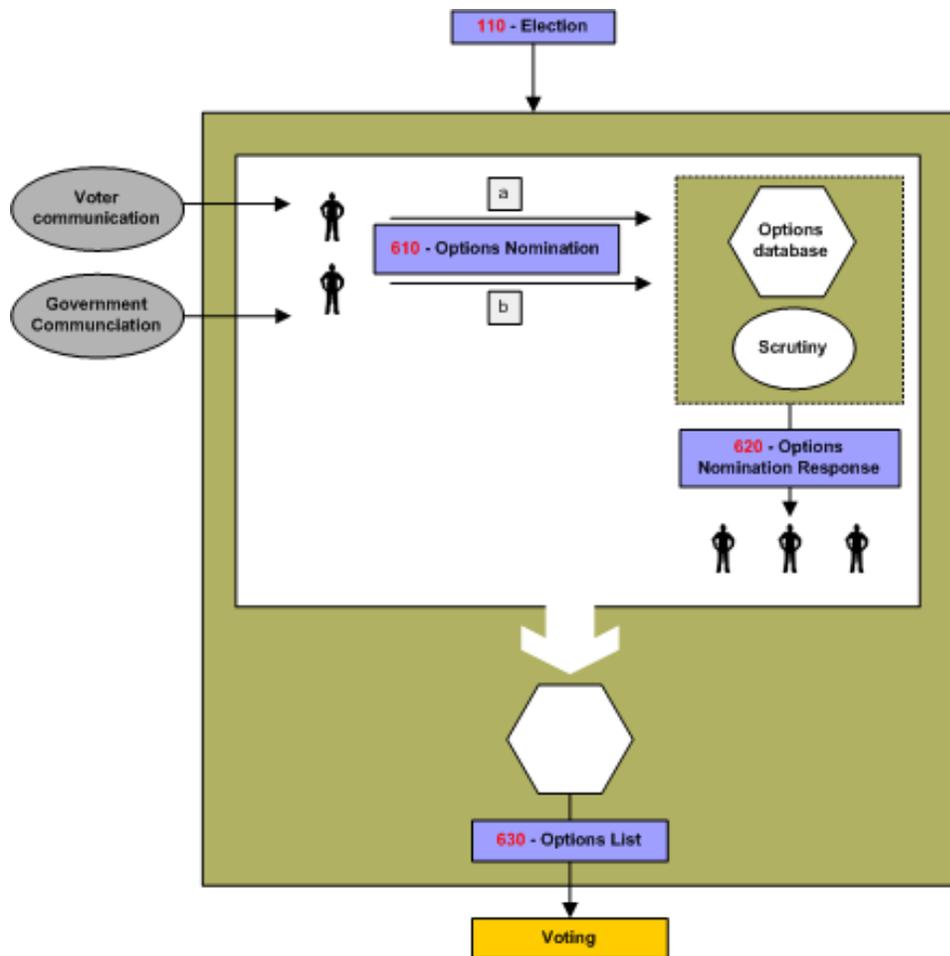
332 Schema 330 allows election officials to enquire of other jurisdictions whether a particular candidate is  
333 standing in more than one contest.

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

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

### 337 **3.2.2 The Options Nomination Process**

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



341

342 *Figure 2D: Referendum Options Nomination Process*

343

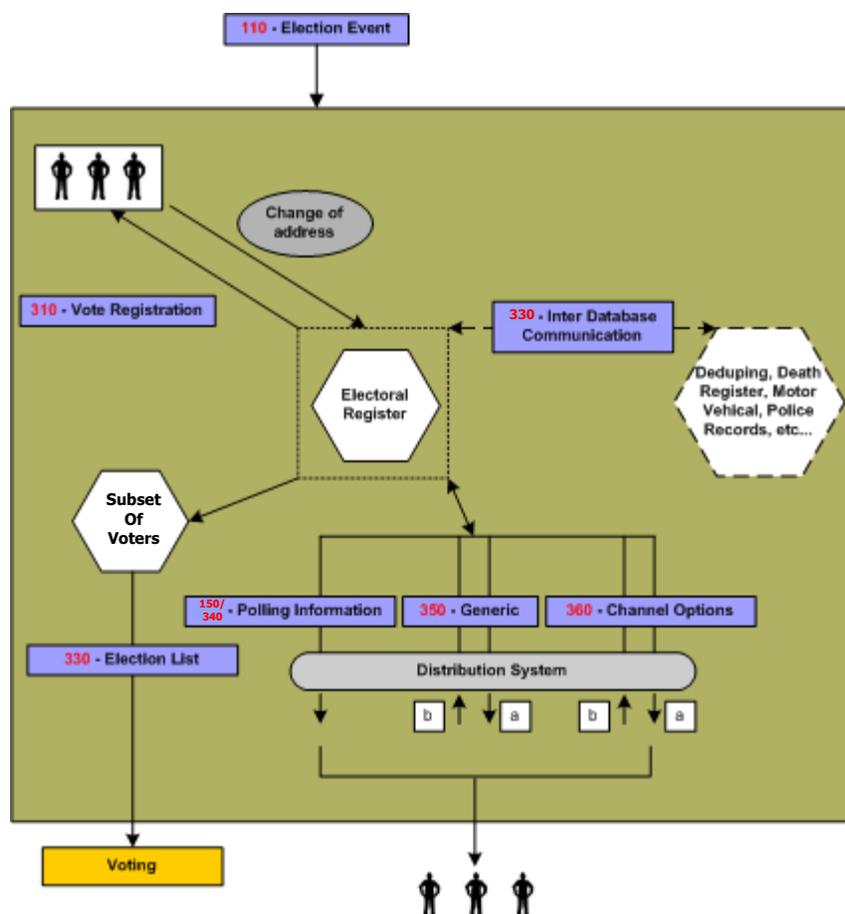
344 The nomination can be received in a number of ways including direct from government institutions or from  
 345 citizens or businesses, and schema 610 handles the receipt of nominations.

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

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

### 350 **3.2.3 The Voter Registration**

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



353  
354 *Figure 2E: Voter Registration*

355  
356 The centre of this process is the Electoral Roll Database or the Voters' Database. The input into this  
357 database is the outcome of communications between 'a voter' and 'an Election Authority'. The subject of  
358 this correspondence can vary from adding a voter to modifying a voter; deletion of a voter is considered  
359 as part of modification.

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

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

367 At a certain date, a subset of the voters' database is fixed from which the election list is generated.  
368 Schemas contain some subset of the eligible voters, perhaps grouped by polling district or voting channel.

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

- 371
- 372 • Channel options
  - 373 • Polling Information
  - 374 • Generic.

375 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.

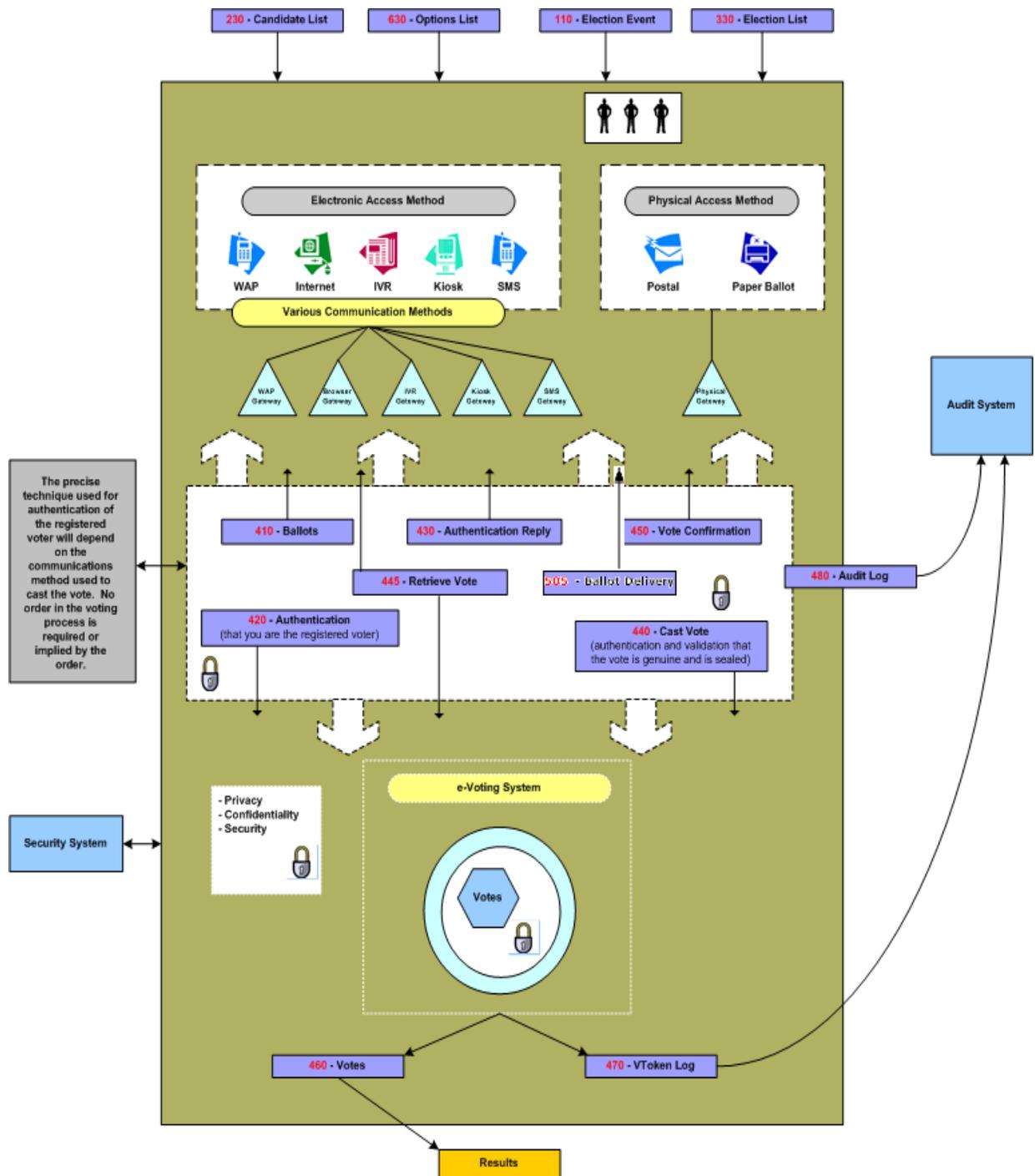
376 **3.2.4 The Electronic Ballot Delivery Process**

377 This is the process that enables election officials to make available electronically blank ballot forms to  
378 overseas and uniformed voters in a manner directly akin to absentee voting. This obviates the problem  
379 that distributing blank paper ballots to these voters using conventional postal mail can involve  
380 considerable delays, imperiling their timely return. The process involves providing export formats for the  
381 election information needed to facilitate construction and delivery of electronic blank ballot form to  
382 Internet-accessible ballot delivery systems (BDSs).

383 Whilst the process is primarily used by local election officials, it can also be useful in achieving other  
384 goals, such as permitting consistent export of information between voter registration database systems  
385 (VRDBs) and BDSs. It also provides the opportunity for overseas and uniformed voters to track the status  
386 of their ballot. The blank ballot forms are distributed using schema 505. There is no provision in this  
387 schema for the electronic return of these ballot forms, nor for verifying the voter's entitlement to vote. If  
388 those processes are performed then the other appropriate voting process schemas should be used. The  
389 position of this process in the overall voting process is depicted in Fig 2F below.

390 **3.2.5 The Voting Process**

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



392  
393 *Figure 2F: The Voting Process*

394  
395 We have assumed various systems would be involved in providing the voting process and regard each  
396 system as an independent entity.  
397 As this figure shows, the voter will be voting using a choice of physical channels such as postal or paper  
398 ballot (the 'physical access methods'), or the voter can vote using 'electronic access methods' where  
399 he/she can utilize a number of possible e-voting channels.  
400 Each channel may have a gateway acting as the translator between the voter terminal and the voting  
401 system. Typically, these gateways are in proprietary environments. The following schemas are to be used

402 when interfacing to such gateways: 410, 420, 430, 440 and 450. These schemas should function  
 403 irrespective of the application or the supplier's favored choice of technology.

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

406 Where a voter's right to vote in any particular contest needs to be determined, this is defined by the  
 407 parameters of his VToken. See Section B.1 for more information on security and the VToken.

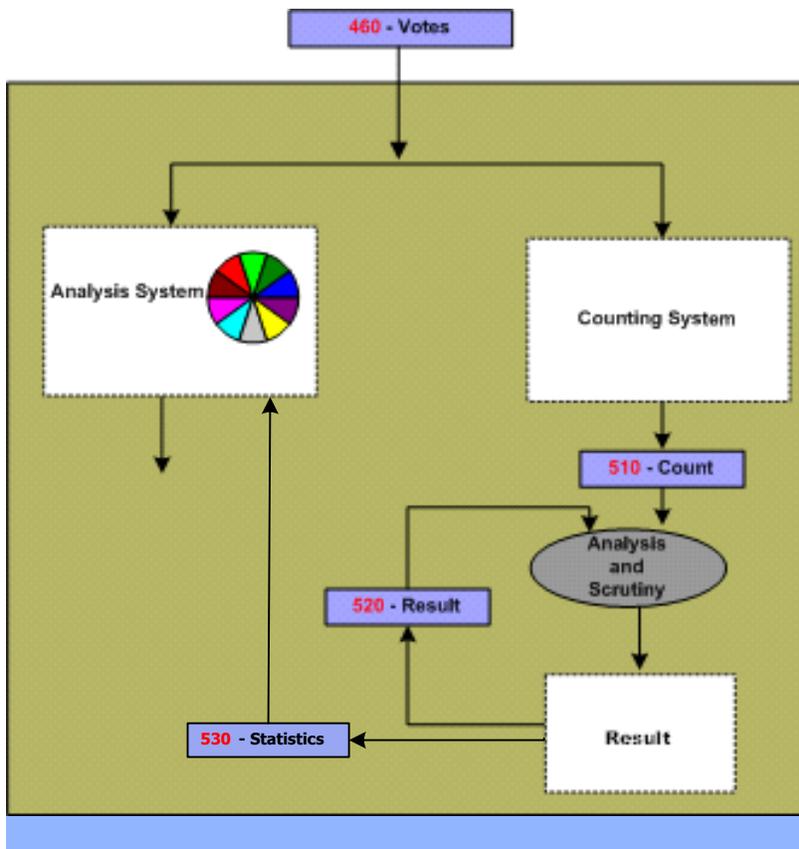
408 In some scenarios the right to vote may need to be qualified. This may occur if the voter's right to vote is  
 409 challenged or if the voter is given the temporary right to vote. In this case the vote needs to be cast by a  
 410 voter with a Qualified VToken. The reason for the qualification shall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

413 In some jurisdictions, eg Australia, where voting is compulsory there is a need to report if an elector has  
 414 been excused for not voting and what was the reason for the excuse. This information can be recorded in  
 415 schema 330.

416 To create balloting information, input data is needed about the election, the options/candidates available  
 417 and the eligible voters; see schemas 230, 110 and 330 for exchanging such information between e-  
 418 systems.

419 **3.2.6 The Vote Report Process**

420 Two of the post election items are the Final or Interim Result and the Audit Report. Audit is discussed in  
 421 3.2.6.



422  
 423 *Figure 2G: The Vote Reporting Process*  
 424

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

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

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

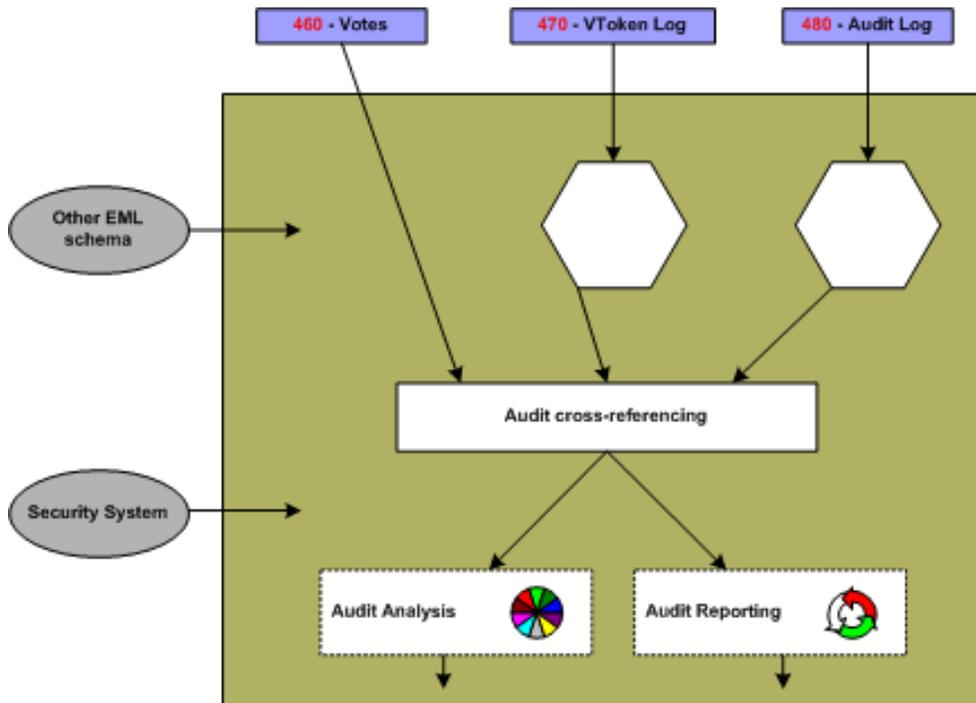
435 The votes schema 460 also feeds into a variety of analysis systems, which can be used to provide for  
 436 demographic, statistical or other types of election reports. The output of these analysis systems is outside  
 437 the scope of this document.

438 Schemas 510 and 520 allow for Simulation and Extrapolation of final or interim Counts and Results.  
 439 Simulation being the facility to forecast the result of a contest based on the result of another contest.  
 440 Extrapolation is the facility to forecast the final result of a contest based on the count so far.

441 Schema 530 allows for a variety of statistics to be extracted from the results and passed to analysis  
 442 systems and other media outlets.

443 **3.2.7 The Auditing Process**

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



447  
 448 *Figure 2H: Auditing System*

449

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

452 Schemas 460, 470, 480 from the voting process provide input data to the audit process. Depending on  
 453 the audit requirements additional data from other processes may be required. In particular, the security

454 process may provide additional data about all the issued VTokens and Qualified VTokens (see Figure 3D:  
455 Voting system security).

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

- 459 • Eligible voters
- 460 • Issued VTokens or Qualified VTokens.

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

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

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

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

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

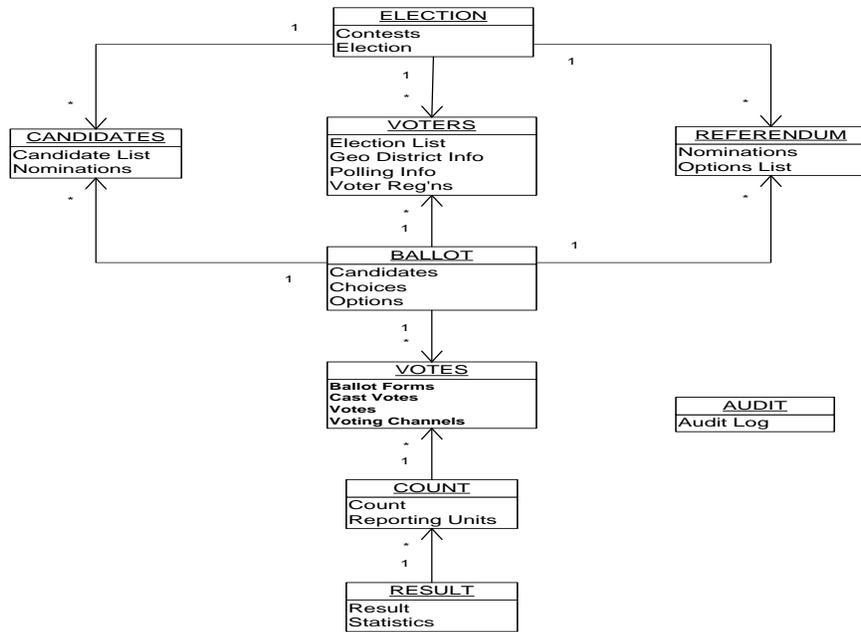
481 There is also the requirement for the election officer to be provided with a mechanism that allows for  
482 multiple observers to witness all the voting process. How this is achieved is dependant on the  
483 implementation of the system and procedures adopted. However, the seals and channel information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

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

### 492 **3.3 Data Requirements**

493 Shown below at Fig 2i is a high-level data model of the data used in the above processes. Further lower-  
494 level data models are available in the 'EML v7.0 Data Models' file and all the data are defined in 'EML  
495 v7.0 Data Dictionary'. Fig 2j below shows the mapping between the data entities and the EML schemas.

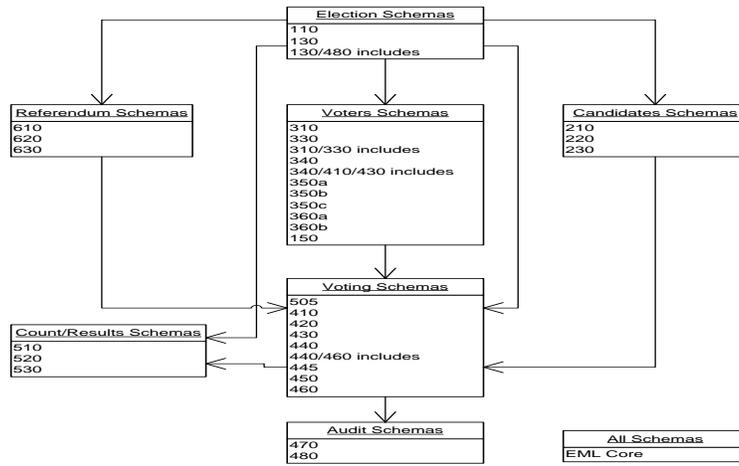
**EML v7.0 Top Level Data Model**



496  
497  
498

Figure 2i: High-level Data Model

**EML\_v7.0 Entity/Schema Mapping**



499  
500  
501  
502

*Fig 2j Entity/Schema Mapping*

---

## 503 4 Schema Outline

### 504 4.1 Structure

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

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

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

519 As well as these, several external schemas are used. The W3C has defined a standard XML signature  
520 [5]. OASIS has defined schemas for the extensible Name and Address language [1]. As part of the  
521 definition of EML, the committee has defined a schema for the Timestamp used within EML. All these  
522 schemas use their appropriate namespaces, and are accessed using xs:import directives.

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

526 As an international specification, EML is generic in nature, and so needs to be tailored for specific  
527 scenarios. An example of this is the draft IEEE/P1622/D1 standard [6] which specifies the electronic data  
528 interchange formats for the distribution of blank ballot forms, primarily to satisfy the needs of the USA's  
529 UOCAVA (Uniform and Overseas Citizens Assistance in Voting Act) and MOVE (Military and Overseas  
530 Voter Empowerment) Acts. However whilst this standard has been developed specifically for the USA it  
531 may be relevant and appropriate to other jurisdictions around the world with similar requirements.

532 Some aspects of the language are indicated in EML as required for all scenarios and so can be used  
533 unchanged. Some aspects (such as the ability to identify a voter easily from their vote) are required in  
534 some scenarios but prohibited in others, so EML defines them as optional. Where they are prohibited,  
535 their use must be changed from an optional to prohibited classification, and where they are mandatory,  
536 their use must be changed from an optional to required classification.

### 537 4.2 Viewing Schemas

538 EML schemas are supplied as text documents. For viewing the structure of the schemas, we recommend  
539 the use of one of the many schema development tools available. Many of these provide graphical  
540 displays.

### 541 4.3 IDs

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

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

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

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

## 551 4.4 Displaying Messages

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

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

```
557 <?xml version="1.0" encoding="UTF-8"?>
558 <EML
559   Id="410"
560   SchemaVersion="7.0"
561   xml:lang="en"
562   xmlns="http://www.govtalk.gov.uk/temp/voting"
563   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
564   xsi:schemaLocation="http://www.govtalk.gov.uk/temp/voting
565     ..\schemas\ballot.xs">
566   <Display Format="html">
567     <Stylesheet Type="text/xsl">../stylesheets/ballot.xsl</Stylesheet>
568     <Stylesheet Type="text/css">../stylesheets/eml.css</Stylesheet>
569   </Display>
570   <Ballots>
571     ...
```

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

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

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

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

**Voting Paper**

**National Executive Committee & International Liason Committee Elections 2001-2003**

**PLEASE READ THE VOTING INSTRUCTIONS BELOW BEFORE VOTING**

The count for this election will be conducted by means of the Single Transferable Vote.

To cast your vote you should enter the number "1" against your first preference and the number "2" against your next preference.

**Please do not use an "X". You may vote in both elections.**

**National Executive Committee**

one to be elected

Option Number	Name	Order of Preference
101	<a href="#">J Chahal</a>	1
102	<a href="#">S Ruston</a>	1

**International Liason Committee**

one to be elected

Option Number	Name	Order of Preference
121	<a href="#">N Goodman</a>	1
122	<a href="#">J Marcos</a>	1

If you opt to cast your vote by post, please return your voting paper in the pre paid envelope provided to reach the Independent Scrutineer, election.com, PO Box 648, Wembley, HA0 1FA.

Your paper should arrive not later than midday on **FRIDAY 23RD MARCH 2001**.

If you vote using more than one method (internet, telephone or postal), your vote will be declared invalid.

**Name: J Chahal**

I have worked within various organisations within our trade for fifteen years, gradually working my way up from the bottom. I have worked all over the country for these roles and have gained a good knowledge of what is involved with this committee.

Currently I provide a supporting role to the people on the National Executive Committee, this means that I have a working knowledge of what must be done and not just a theoretical understanding.

In my spare time I like to watch motor racing and enjoy keeping fit in general. I have always been extrovert and am not afraid of expressing opinions, both those of my own and of others. Also I like to make time to relax with my family and can often be found playing football with my son.

590

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

592

**Voting Paper**

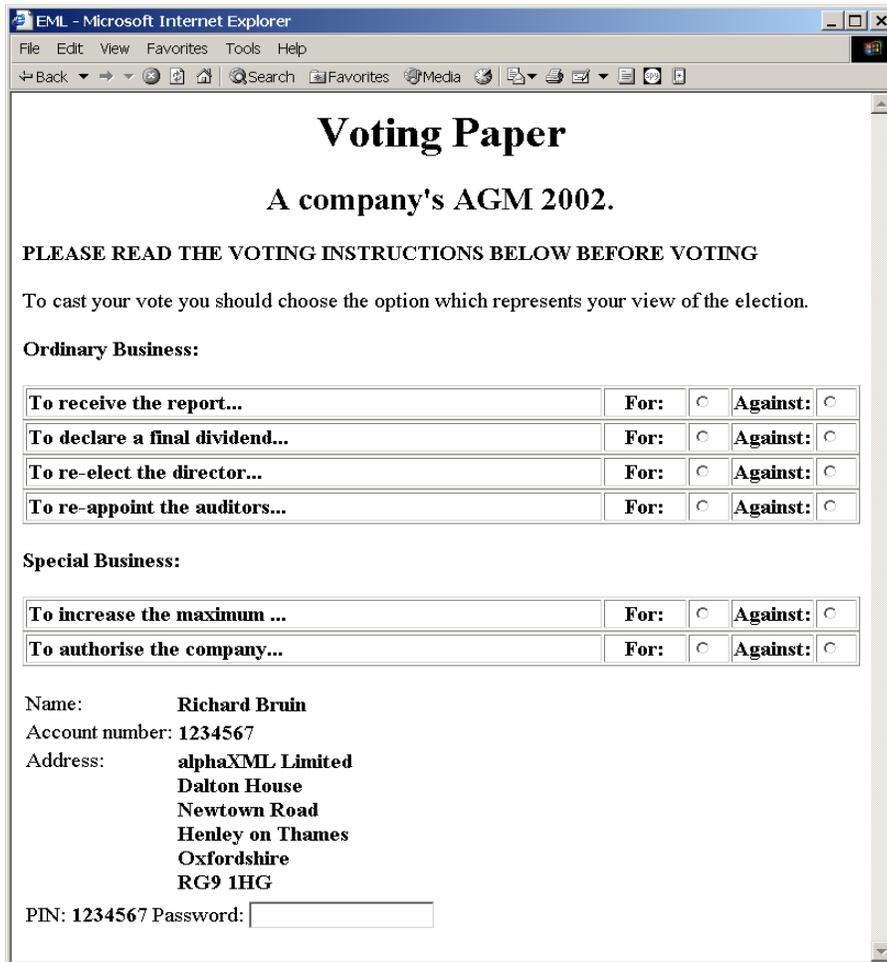
**Vote For Your Favourite Soccer Team**

Please select your favourite favourite soccer team.

Option Number	Name	Select
1	<a href="#">Arsenal</a>	<input type="radio"/>
2	<a href="#">Chelsea</a>	<input type="radio"/>
3	<a href="#">Leeds</a>	<input type="radio"/>
4	<a href="#">Liverpool</a>	<input type="radio"/>
5	<a href="#">Manchester United</a>	<input type="radio"/>

593

594 Figure 3B: Screen shot of the ballot for scenario 2  
595



596  
597 Figure 3C: Screen shot of the ballot for scenario 3

## 598 4.5 EML Message Validation

599 It is up to each specific system implementation whether it uses these schemas for validation of EML  
600 messages for either testing or live use. The recommended approach is to validate incoming messages  
601 against the EML schemas (with the application-specific EML externals schema), then further validate  
602 against the relevant Schematron schema or OASIS CAM template. The first stage requires the use of an  
603 XML processor (parser) that conforms to W3C XML Schema. The second stage requires either an XSLT  
604 processor or a dedicated Schematron or CAM processor.

605 However, an implementation may choose to:

- 606 • modify the EML schemas to incorporate those application-specific constraints that can be  
607 represented in W3C XML Schema;
- 608 • not validate the rules that are encoded as templates schemas (Schematron or CAM);
- 609 • not perform any validation; or
- 610 • develop some alternative backend validation.

## 611 4.6 Namespaces

612 The message schemas and the core schema are associated with the namespace  
613 `urn:oasis:names:tc:evs:schema:eml`. This is defined using the prefix `eml`.

614 The XML Schema namespace `http://www.w3c.org/2001/XMLSchema` is identified by the prefix `xs`  
615 and the XML Schema Instance namespace `http://www.w3.org/2001/XMLSchema-instance` by  
616 the prefix `xsi`.

617 The namespace for the W3C XML Signature Syntax and Processing is  
618 `http://www.w3.org/2000/09/xmldsig#` and is identified by the prefix `xmlns:ds`.

619 Use is also made of namespaces for the Extensible Name and Address Language (xNAL). The  
620 Extensible Name Language namespace `urn:oasis:names:tc:ciq:xnl:4` is identified by the prefix  
621 `xNL`, and the Extensible Address Language namespace `urn:oasis:names:tc:ciq:xal:4` by the  
622 prefix `xAL`.

## 623 4.7 Extensibility

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

## 627 4.8 Additional Constraints

628 The EML schemas provide a set of constraints common to most types of elections worldwide. Each  
629 specific election type will require additional constraints, for example, to enforce the use of a seal or to  
630 ensure that a cast vote is anonymous. It is recommended that these additional constraints be expressed  
631 using the Schematron language although other validators, e.g. OASIS CAM, can be used. This allows  
632 additional constraints to be described without altering or interacting with the EML schemas. Any  
633 document that is valid to a localization expressed in Schematron must also be a valid EML document.

## 634 4.9 Metadata

635 Some messages need information relating to the issuing of them, such as the issue date, who issued  
636 them, etc. This is most likely to be a requirement for the 330 message but is equally applicable to 130,  
637 230, 350a and several others. For that reason, it is useful to make this optional information available in  
638 the header. The information usually consists of: managing authority, date of issue, start of list period  
639 (used for changes to the list to indicate the start of the period for which changes are being shown), end of  
640 list period (i.e. the date of the snapshot of the list).

## 641 4.10 Splitting of Messages

642 There is sometimes a need to split long messages into several parts. By their nature, each of these  
643 messages will contain a small amount of background information and a single element type that is  
644 repeated many times. For example, the 330-electionlist message can have many VoterDetails elements.

645 When a message is split, each part must be a complete, valid EML document. This will contain all the  
646 elements required by EML and the specific application. Those parts outside the repeated element that  
647 relate to the message as a whole, such as the TransactionId, must have the same values in each part  
648 message. The values of those elements and attributes that relate to an individual part message, such as  
649 the SequenceNumber, may vary between the individual part messages. Information in the EML element  
650 indicates the sequence number of the message and the number of messages in the sequence. Each  
651 message in the sequence must contain the same TransactionId, and must indicate the repeated element  
652 according to the table below. Only the messages shown in the table may be split in this way.

653

Message	Repeated Element
150-geographicdistrict	ElectionDistricts

330-electionlist	VoterDetails
340-pollinginformation	Polling
410-ballots	Ballot
460-votes	CastVote
470-vtokenlog	VTokens
480-auditlog	LoggedSeal
505-ballotdelivery	LocalityBoundaries

654 For ease of implementation, a message that can be split may contain the elements used for splitting even  
655 if the entire message is sent in one piece. In this case, the values of SequenceNumber and  
656 NumberInSequence will both be "1".

## 657 4.11 Error Messages

658 The 130 schema is used to define a message for reporting errors in EML messages.

659 Error messages are given codes. These fall into one of five series:

1000	XML well-formedness or Schema validation error
2000	Seal error
3000	EML rule error
4000	Localization rule error
5000	System specific error

660 If the error type is not message-specific (or is a general rule applying to several schemas), the series  
661 reference above is used. If it is message-specific, the last three digits of the error series (and possibly a  
662 final alpha character) reflect the message type. A three digit error code is appended to the series code,  
663 separated by a hyphen.

664 An error code relating to a localisation applicable to all message types could therefore be 4000-001. One  
665 specific to the localization of schema 110 could be 4110-002.

## 666 4.12 All Schemas

### 667 4.12.1 XML Well-Formedness or Schema Validation Error

Error code	Error Description
1000-001	Message is not well-formed
1000-002	Message is not valid

### 668 4.12.2 Seal Errors

Error code	Error Description
2000-001	The Seal does not match the data

### 669 4.12.3 EML Additional Rules

670 The following rules apply to messages regardless of localization. One of the two rules on splitting will  
671 apply to each message type as described in the table below.

Error Code	Error Description
3000-001	If there are processing units in the

Error Code	Error Description
	AuditInformation, one must have the role of sender
3000-002	If there are processing units in the AuditInformation, one must have the role of receiver
3000-003	This message must not contain the elements used for splitting
3000-004	The value of the Id attribute of the EML element is incorrect
3000-005	The message type must match the Id attribute of the EML element
3000-006	All messages that are split must include the correct sequenced element name.

672

	3000-003	3000-006
110	✓	
130	✓	
150		✓
210	✓	
220	✓	
230	✓	
310	✓	
330		✓
340		✓
350a	✓	
350b	✓	
350c	✓	
360a	✓	
360b	✓	
410		✓
420	✓	
430	✓	
440	✓	
445	✓	
450		✓
460		✓
470		✓
480		✓
505		✓
510	✓	
520	✓	
530	✓	
610	✓	

<b>620</b>	✓	
<b>630</b>	✓	

673

674

## 5 Schema Descriptions

675

### 5.1 Overview

676

The following table presents a high-level overview of the EML schemas. Further explanations are given

677

in the following sub-paragraphs.

Schema Name	Purpose
EML 110 – election event	Information about an election or set of elections. It is usually used to communicate information from the election organizers
EML 130 – response	Report error response. Contains details of the message received that was in error.
EML 150 – geographic district	Allow use of geographic mapping systems to describe the election districts and boundaries and balloting
EML 210 – candidate nomination	Used to nominate candidates or parties, consenting or withdrawing
EML 220 – response to nomination	Use to confirm whether the candidate's nomination has been accepted.
EML 230 – candidate list	Contest and candidates details
EML 310 – voter registration	Used to register voters for an election
EML 330 – voter election list	Details of actual voters for an election and ballot entitlement
EML 340 – polling information	Notification to voter of an election, their eligibility and how to vote
EML 350a – outgoing generic	Provides a common structure for communications to the voter.
EML 350b – incoming generic	Provides a common structure for communications from the voter.
EML 350c – internal generic	Provides a common structure for systems communications.
EML 360a – outgoing channel	Used for messages offering a set of voting channels to the voter
EML 360b – incoming channel	Used for messages defining a preferred voting channels of the voter
EML 410 – ballot	Describes the actual ballot to be used for an election
EML 420 – voter authentication	Used for voter authentication during a voting process
EML 430 – authentication response	Indicates whether authentication succeeded; may present the ballot to the user
EML 440 – cast vote	Actual record of vote cast
EML 445 – retrieve vote	For systems that include a pre-ballot box from which votes can be retrieved and confirmed
EML 450 – confirm vote	Show whether a vote has been accepted and provide a reference number or rejected.
EML 460 – votes group	Group of votes being transferred for counting
EML 470 – vtoken log	Add voting tokens to an audit log and reporting Voter actions to an Election Management System.
EML 480 – audit log	Documents access to voting records and reason
EML 505 – ballot delivery	Electronic delivery of blank ballot forms to overseas and uniformed voters
EML 510 – count	Results of election contest(s) and counts
EML 520 – result	Communicating specific result details on candidates and elections
EML 530 – statistics	Provide statistical information about EML 510 counts and results
EML 610 – options nomination	Used to nominate the choice of options that will be included in a referendum.
EML 620 – options nomination	Confirms whether the options nomination has been accepted.

response

EML 630 – options list

Use to transfer lists of proposals for a referendum

EML Core

Defines the core definitions of the content model reused across the EML schemas

## 678 **5.2 EML Core Components**

679 The EML Core schema contains elements and data types that are used throughout all the EML schemas.

680 For details see the EML core dictionary that is provided as separate files in XML and spreadsheet  
681 formats. The core components are included in the EML Core schema that is imported into each EML  
682 schema.

683 The dictionary shows items in sequence and denotes their CCTS (Core Components Technical  
684 Specification) classification based on their usage within EML structures. Those marked as BBIE (Basic  
685 Business Information Entity) are atomic pieces (element), while those marked as ABIE are Aggregate  
686 entities consisting of more than one component (elements structure), while ASBIE equate to XML  
687 attributes values for the associated BBIE elements. For complete discussion of Core Components  
688 concepts see the UN/CEFACT Core Components specification  
689 ([http://www.unece.org/cefact/ebxml/CCTS\\_V2-01\\_Final.pdf](http://www.unece.org/cefact/ebxml/CCTS_V2-01_Final.pdf) ).

690 Related to classification of content type is the difference between Schema elements and types and  
691 specifically Schema `xsd:complexType` usage and this is discussed next.

### 692 **5.2.1 Complex Data Types**

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

698 When defining a complete mark-up language, limiting the use of elements and types can restrict further  
699 development of the language. For that reason, both data types and elements are defined in EML. Only  
700 where an element is an example of a primitive or derived data type defined in XML Schema Descriptions  
701 is no explicit data type defined within EML.

702 In use, it is expected that, for example:

- 703 • A voting token will always have an element name `VToken` and so will use the element name.
- 704 • A logo or a map have similar definitions, so both use the `PictureDataStructure`. There is no  
705 `PictureData` element.
- 706 • Within voter identification, some elements will usually need to be made mandatory and so a schema  
707 will specify a new element based on the `VoterIdentificationStructure` data type.

## 708 **5.3 Message Headers**

709 Each EML messages has a header included in it that describes the message and lists the history of  
710 changes that have been made to it. The original EML schemas did not group this information in a header,  
711 but had it as part of the main body of schema. By grouping it together in a header it makes the use and  
712 management of schemas simpler.

713 Now exchanges can readily use a consistent set of header details across a system implementation.

## 714 **5.4 Message Schemas**

715 This section describes the EML messages and how the message specifications change for this  
716 application. It uses the element and attribute names from the schemas.

## 717 Election Event (110)

718 This schema is used for messages providing information about an election or set of elections. It is usually  
719 used to communicate information from the election organisers to those providing the election service.

720 The message therefore provides information about the election event, all elections within that event, all  
721 contests for each election and other general election management information.

722 For the election event, the information includes the ID and name of the event, possibly with a qualifier on  
723 the event. This qualifier is used when an event has several local organisers. For example, for a UK  
724 general election, each constituency organizes its own contests. The election event is therefore the  
725 general election, whilst the qualifier would indicate the constituency. Other information regarding an  
726 election event comprises the languages to be used, the start and end dates of the event, potentially a list  
727 of external documents that are applicable (such as the rules governing the election), a description and  
728 information about the managing authority.

729 The managing authority can be indicated for the event, each election, each contest within the election and  
730 each reporting unit.

731 An election can have a number of dates associated with it. For example, there is likely to be a period  
732 allowed for nomination of candidates and a date when the list of eligible voters is fixed. Each date can be  
733 expressed as a single date when something happens, a start date, an end date, or both start and end  
734 dates. These dates can be either just a date or both a date and time using the subset of the ISO 8601  
735 format supported by XML Schema.

736 Like the event, an election can have both a managing authority and referenced documents. Finally, there  
737 is a `Messages` element for additional information.

738 A contest has a name and ID. It can also have reporting unit identifiers. A contest may need to specify its  
739 geographical area independently from its name, for which purpose the `Area` element is provided. Each  
740 contest can specify the voting channels allowed. In general, the list of possible channels will be further  
741 restricted as part of a local customization. Each channel can specify several methods for authenticating  
742 the voter, such as PIN and password, and a response method, indicating the type of response to be given  
743 to a cast vote. Finally, facilities are provided to indicate the dates and times when the channel will be  
744 available to the voter.

745 As described previously, a contest can indicate its managing authority. It may also indicate the position  
746 (such as 'President') for which votes are being cast. The `Description` allows for additional text describing  
747 the contest. Each contest indicates the voting method being used, whilst the `CountingAlgorithm`  
748 indicates the method of counting (such as the d'Hondt or Meeks method) that will be used. The minimum  
749 and maximum number of votes to be cast by each voter can also be indicated.

750 A list of polling places can be provided. These can be either physical locations for people to go to vote,  
751 postal addresses for postal votes or electronic locations. An 'other location' is also allowed for cases  
752 where these do not meet the requirements. A location can also say when it will be available. This is  
753 intended for mobile polling stations that will only be available at a given address for a part of the voting  
754 period.

755 Finally, a `Messages` element allows for additional information that might be communicated to the voter  
756 later through other messages.

## 757 Additional Rules

Error Code	Error Description
3110-001	The allowed channels must not be declared at both the election event level and the contest level.

## 758 **Response (130)**

759 Some messages have a defined response message that provides useful information. However, there is a  
760 need for a more general response, either to indicate that a message has been accepted, or to indicate the  
761 reasons for rejection.

762 The message includes information to identify the message to which the response applies (by using the  
763 same transaction id in the `EML` element and, if necessary, including the sequence number of the message  
764 to which the response applies in the `Response` element), with information on the entity raising the  
765 message, whether the message was accepted and information about the errors if it was not. The desired  
766 language for a display message can also be included to allow a downstream processor to substitute a  
767 language-specific error message if required.

768 If the message is reporting an error, the location of the error within the message can be indicated.  
769 Usually, this will be an XPath to the location of the error. However, errors detected by an XML parser may  
770 be in a different format, such as a line number.

771 Note that a single response can be raised for a series of sub-messages with the same transaction ID.  
772 This allows indication, for example, that a sub-message was missing.

## 773 **Additional Rules**

Error Code	Error Description
3130-001	If the message is not accepted, there must be an <code>Errors</code> element

## 774 **GeoDistrict (150)**

775 This schema allows the use of geographic mapping systems to describe election districts and their  
776 boundaries by providing information to voters to help their understanding of where and when they should  
777 go to cast their vote. For example information relating to the streets and polling places within a district, the  
778 name by which the district is identified to voters and physical features and landmarks describing a specific  
779 polling place to be used in elections.

780 Supplementary information about the districts and polling places to assist voters can also be recorded, for  
781 example detailed descriptions in one or more language that describes the district, the political area or  
782 legislature that the district belongs to, the Authority that is responsible for managing elections in the  
783 district, and access and facilities details about a specific polling place to be used in elections.

784 This set of authorized information can be made available by any number of organisations through a  
785 variety of different outlets.

## 786 **Candidate Nomination (210)**

787 Messages conforming to this schema are used for four purposes:

- 788 1. nominating candidates in an election;
- 789 2. nominating parties in an election;
- 790 3. consenting to be nominated; or
- 791 4. withdrawing a nomination.

792 Candidate consent can be combined in a single message with a nomination of the candidate or party or  
793 sent separately.

794 Note that the message does not cover nomination for referendums.

795 The election and contest must be specified. When a candidate is being nominated, there must be  
796 information about the candidate and one or more proposers. The candidate must supply a name.  
797 Optionally, the candidate can provide contact information, an affiliation (e.g. a political party) and textual  
798 profiles and election statements. These two items use the `MessagesStructure` to allow text in multiple  
799 languages. There is also scope to add additional information defined by the election organiser.

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

802 If a party is being nominated, the primary proposer will be the contact. Information on candidates in a  
 803 party list can also be provided.

804 Candidates, either individuals or on a party list, must define the action being taken and may provide  
 805 scrutiny information. The scrutiny requirements indicate how the candidate has met any conditions for  
 806 standing in this election. This could include indicating that a deposit has been paid or providing a  
 807 reference to prove that he or she lives in the appropriate area. This information can be signed  
 808 independently of the complete message.

## 809 **Response to Nomination (220)**

810 This message is sent from the election organiser to the candidate or nomination authority for a party to  
 811 say whether the nomination has been accepted. Along with the acceptance information and the basic  
 812 information of election, contest and party and candidate names, the candidate's contact details and  
 813 affiliation can be included and a remark explaining the decision.

## 814 **Additional Rules**

Error Code	Error Description
3220-001	If the nomination has not been accepted, a reason for rejection is required in the Remark element

## 815 **Candidate List (230)**

816 This schema is used for messages transferring candidate lists for specified contests. It has the election  
 817 event, election and contest identifiers, and optionally the event dates and a contest description. The list  
 818 itself can be either a list of candidates, each with a name, address, optional affiliation and other useful  
 819 data, or a list of parties. In the latter case, contact information and a list of candidates under a party list  
 820 system can also be included.

## 821 **Voter Registration (310)**

822 This schema is used for messages registering voters. It uses the `VoterIdentificationStructure`.  
 823 The `VoterInformationStructure` is used unchanged. Proof of ID can be provided.

824 There is the facility for the transmission channel (for example a trusted web site) to add the time of  
 825 transmission.

## 826 **Additional Rules**

Error Code	Error Description
3310-001	The Proxy must not have a VToken or VTokenQualified

## 827 **Election List (330)**

828 This schema is primarily used for messages communicating the list of eligible voters for an election or set  
 829 of elections. It can also be used for any other purpose, eg the transfer of voter information, to report voter  
 830 eligibility, to report ballot types for ballot delivery purposes, entitlement checking such as poll book  
 831 applications or ballot delivery systems. Partial lists are allowed through the use of the Qualifier, Blocked  
 832 and VoterGroup elements. So, for example, a list of postal voters or a list of proxies can be produced.  
 833 The schema can also be used for filtered lists such as a list of postal proxies. These lists sometimes do  
 834 not contain any names meeting the filter so empty lists are allowed.

835 For each voter, information is provided about the voter himself or herself, and optionally about the  
836 elections and contests in which the voter can participate. The information about the voter is the same as  
837 that defined in the 310-voter-registration schema. Added to this can be a list of elections, each identifying  
838 the election and the contest in which this voter is eligible to vote, and the polling places available. Any  
839 voter can have a Blocked element set against them with an optional Reason and Channel. This allows a  
840 list to be produced for a polling place indicating those that have already voted by another means or who  
841 have registered for a postal vote. It can also be used if the complete electoral register must be transmitted  
842 (perhaps as a fraud prevention measure) but some people on the register are no longer eligible to vote.

## 843 **Additional Rules**

Error Code	Error Description
3330-002	The polling district can only be included for either the voter or the election.
3330-003	The polling place can only be included for either the voter or the election.

## 844 **Polling Information (340)**

845 The polling information message defined by this schema is sent to a voter to provide details of how to  
846 vote. It can also be sent to a distributor, so multiple sets of information are allowed. In the case of SMS  
847 voting, ballot information may also be required, so this can be included. Either one or several sets of  
848 polling information may be sent to each voter for any election event.

849 Some information about the voter and any proxy may be included, for example to print on a polling card.  
850 This can also include a mailing address for a distributor to use.

851 Information about the elections and contests is included for the benefit of the voter. For each voting  
852 channel, this includes where to vote (which could be a polling station, address for postal voting, URL for  
853 Internet voting, phone number for SMS voting etc) and the times that votes can be placed. Use of the  
854 DisplayOrder attribute on these allows the display or printing of information to be tailored from within the  
855 XML message.

856 Ballot information may be included if required. This is a subset of the information defined in the 410-  
857 ballots schema. In this case, it is likely that the short code for a candidate will be used for SMS voting. It is  
858 possible that an expected response code will be provided as well. Both the short code and expected  
859 response code may be tailored to the individual voter as part of a security mechanism.

## 860 **Outgoing Generic Communication (350a)**

861 This schema provides a common structure for communications to the voter. Individual message types can  
862 be designed based on extensions of this schema.

863 The voter must always provide a name and might provide one or more identifiers. These are shown as a  
864 restriction of the VoterIdentificationStructure, the restriction being to leave out the VToken and  
865 VTokenQualified. Contact details are also required, and it is expected that at least one of the allowed  
866 contact methods will be included. Inclusion of proxy information is optional.

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

## 869 **Incoming Generic Communication (350b)**

870 This schema provides a common structure for communications from the voter. Individual message types  
871 can be designed based on extensions of this schema.

872 The voter's name must be provided and there can be one or more identifiers. These are shown as a  
873 restriction of the VoterIdentificationStructure, the restriction being to leave out the VToken and

874 `VTokenQualified`. Contact details are also required, and it is expected that at least one of the allowed  
875 contact methods will be included. Inclusion of proxy information is optional.

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

### 878 **Internal Generic (350c)**

879 This schema provides a common structure for communications between those involved in organizing an  
880 election. Individual message types can be designed based on extensions of this schema.

881 There are optional `To` and `From` elements, which can contain any EML elements. It is expected that  
882 these will usually be a responsible officer or a person's name and contact information.

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

### 885 **Outgoing Channel Options (360a)**

886 This schema is used for messages offering a set of voting channels to the voter. It is an extension of  
887 schema 350a. A message conforming to this schema will include a list of allowed channels, either to  
888 request general preferences or for a specific election event or election within the event.

### 889 **Incoming Channel Options (360b)**

890 This schema is used for messages indicating one or more preferred voting channels. It may be sent in  
891 response to 360a or as an unsolicited message if this is supported within the relevant jurisdiction.

892 It is an extension of schema 350b, and indicates preferred voting channels in order of preference.

### 893 **Ballots (410)**

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

896 In the simplest case, a distributor can be sent information about the election event and a ballot ID to  
897 indicate the ballot to print. It can also contain the full ballot details for automated ballot generation and  
898 delivery applications.

899 In other cases, the full information about the elections will be sent with either an election rule ID to identify  
900 the voters to whom that election applies or a set of voter names and contact information. If the ballot is  
901 being sent directly to the voter, this information is not required. Since printed ballot papers are likely to  
902 require a unique identifier printed on them, the range to be used for each ballot type can be defined.

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

906 A contest must have its identifier and a list of choices for which the voter can vote. A voter can vote for a  
907 candidate, an affiliation (possibly with a list of candidates) or a referendum proposal. There is also a set of  
908 optional information that will be required in some circumstances. Some of this is for display to the voter  
909 (`HowToVote` and `Messages`) and some controls the ballot and voting process (`Rotation`,  
910 `VotingMethod`, `MaxVotes`, `MinVotes`, `MaxWriteIn`).

### 911 **Authentication (420)**

912 The authentication message defined by this schema may be used to authenticate a user during the voting  
913 process. Depending on the type of election, a voter's authentication may be required. The precise  
914 mechanism used may be channel and implementation specific, and can be indicated using the  
915 `LoginMethod` element. In some public elections the voter must be anonymous; in which case the prime  
916 method used for authentication is the voting token. The voting token can contain the information required  
917 to authenticate the voter's right to vote in a specific election or contest, without revealing the identity of the

918 person voting. Either the `VToken` or the `VTokenQualified` must always be present in an authenticated  
919 message. The `VotingChannel` identifies the channel by which the voter has been authenticated.

## 920 **Authentication Response (430)**

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

## 924 **Cast Vote (440)**

925 This message represents a cast vote, which comprises an optional voting token (which may be qualified)  
926 to ensure that the vote is being cast by an authorized voter, information about the election event, each  
927 election within the event and the vote or votes being cast in each election, an optional reference to the  
928 ballot used, the identifier of the reporting unit if applicable and a set of optional audit information.

929 For each election, the contest is identified, with a set of, possibly sealed, votes. The votes are sealed at  
930 this level if there is a chance that the message will be divided, for example so that votes in different  
931 elections can be counted in different locations.

932 The selection of candidates, affiliations or a referendum option uses the `Selection` element. If an  
933 election requires preferences to be expressed between candidates, multiple `Selection` elements will be  
934 used, each of these having a suitable `Value` attribute. Some elections allow write-in candidates, and  
935 these are handled in a similar way. Preferences can also be expressed between parties, using the  
936 `Affiliation` element. The `PersonalIdentifier` is used in elections where each voter is given an  
937 individual list of codes to indicate their selection.

938 A more complex election might request the voter to vote for a party and then express a preferences of  
939 candidates within the party. In this case, the `Affiliation` element is used to indicate the party  
940 selected, and multiple `CandidateIdentifier` elements, each with a `Value` attribute are used to  
941 express candidate preferences.

942 Preferences in a referendum are handled in the same way as they are for candidates and parties, using  
943 the `ReferendumOptionIdentifier`.

## 944 **Retrieve Vote (445)**

945 This message is used for voting systems that include a pre-ballot box from which votes can be retrieved  
946 and amended before being counted. When a vote is retrieved, it should be deleted from the pre-ballot  
947 box.

## 948 **Vote Confirmation (450)**

949 The vote confirmation message can be used to show whether a vote has been accepted and provide a  
950 reference number in case of future queries. Some voting mechanisms require multiple  
951 `ConfirmationReference` elements. If the vote is rejected, the `Remark` element can be used to show a  
952 reason.

## 953 **Votes (460)**

954 This schema is used to define a message comprising a set of votes being transferred for counting. It is a  
955 set of `CastVote` elements from schema 440 with the addition of the `ProposedRejection` and  
956 `ProposedUncounted` elements and audit information for the voting system. If a vote is rejected, for  
957 example, because a voter has chosen to spoil a ballot paper, many authorities will want to count that vote  
958 as having been cast. The `UncountedVotes` element is reserved for those cases where that record is not  
959 required, for example when the result is thought to be fraudulent. A `ProposedRejection` or  
960 `ProposedUncounted` element must have a `ReasonCode` attribute, and may have a `Reason` attribute to  
961 describe the code. They may also have an `Objection` attribute. This indicates that someone has  
962 objected to this vote being rejected or the proposal that it should not be counted.

963 **VToken Log (470)**

964 The message defined by this schema is primarily used to add voting tokens (which may be qualified) to  
 965 an audit log. It can also be used to track voter actions, such as receipt of a particular blank ballot, and  
 966 then validation of ballot delivery/receipt tracking to an EMS. The *VToken* or *VTokenQualified* is  
 967 extended by the addition of a *Status* attribute with a value of *voted* or *unvoted* for the *VToken* and  
 968 *voted*, *unvoted* and *withdrawn* for the *VTokenQualified*. In addition to sending single tokens as they  
 969 are used, the schema can be used to validate a message sending multiple tokens optionally grouped by  
 970 voting channel. This might be used instead of sending tokens as they used or, for example, to send the  
 971 unused tokens at the end of an election. The *Update* element can be used to indicate that an existing log  
 972 is being updated rather than the message containing a complete new log. The logging system can also  
 973 be identified for audit purposes.

974 **Audit Log (480)**

975 The message defined by this schema is used to log the use of each seal with associated information for  
 976 audit purposes.

977 An audit log message can be transmitted individually as the message causing the log entry is sent or  
 978 received, or the logs can be stored, and several seals logged at once. Ideally, every device that can  
 979 create or consume a message will create a log entry so that pairs of entries can be matched. The most  
 980 important messages to log are those associated with the voting process itself, and these are shown  
 981 below.

982 When used in this message, the *Response* element will not have an *AuditInformation* child.

	<i>Originating Device</i>	<i>Gateway</i>	<i>Voting System</i>	<i>Counting System</i>	<i>Vtoken Logging System</i>	<i>Seal Logging System</i>	<i>Other</i>	<i>Notes</i>
130								4
410	next receiver	receiver	sender					
420	previous sender	sender	receiver					
430	next receiver	receiver	sender				sender / receiver	3
440	previous sender	sender	receiver					
445	previous sender	sender	receiver					
450	next receiver	receiver	sender					
460			sender	receiver				
470			sender	sender	receiver		sender	
480	sender	sender	sender	sender	sender	receiver	sender	2
510				sender			receiver	
520				sender			sender / receiver	

**Notes:**

1. In some cases (e.g. a kiosk) there may be no gateway involved. In this case, the values in the Gateway column apply to the Originating Device.
2. Creators and receivers of 480 (audit log) messages may not be required to log the seals. In particular, if an audit log message is sent per seal created or received, the seal on the 480 message must not be logged.
- 3 "Other" may be the sender when the message is sent to a printer. In this case, the receiver will also be an "Other".
4. An audit log should only be created when the message is used to communicate an error. Most devices can send or receive 130 messages.

983  
984

985 The message may contain the name and ID of the event, election and contest. It can also indicate  
986 whether this is an update to an existing log or a new log. Following the logged seals, a text message can  
987 be added as well as audit information for the audit logging message itself.

988 Each seal being logged must indicate whether the device sending the log was the sender or receiver of  
989 the sealed message. It may be accompanied by the voting token associated with the seal and possibly  
990 additional audit information. This will be the audit information from the message being logged with  
991 additional information about the message. Most of this is common to all message types, but some  
992 message types require specific audit information. One of these is the 130-response message. When this  
993 is used to convey an error, almost the complete message payload (the `Response` element and its  
994 contents apart from the audit information) is logged with the usual message-independent data.

## 995 **Ballot Delivery (505)**

996 The message defined by this schema can be used to serve two basic purposes: (a) for use in indexing  
997 from overseas and uniformed voters' jurisdictional information to their corresponding electronic blank  
998 ballots which may be pre-constructed prior to the election and stored externally, or (b) for use in dynamic  
999 constructing of generically-formatted blank ballots, minus jurisdiction-specific formatting details.

1000 The schema contains two major components:

- 1001 a) a structure that describes various election authority details;
- 1002 b) a series of linked structures that describe information on:
  - 1003 i. voting location information (localities, locality boundaries, districts, polling locations);
  - 1004 ii. information on the elections in each ward or precinct;
  - 1005 iii. contests and propositions in corresponding elections including vote variation information  
1006 and the order in which the contests appear on the ballot, and candidates in the contests.

1007 The election information and linked structures are populated from exports from the voter registration  
1008 database system (VRDB) and the Election Management System (EMS). The contest information links to  
1009 a ballot ID, which can be used by an Internet-accessible ballot delivery system (BDS) for either of the  
1010 following:

- 1011 a) for dynamically constructing the ballot with the election and contest information contained in the  
1012 505 schema, or
- 1013 b) for pointing to a pre-constructed electronic blank ballot located elsewhere.

1014 Accordingly, the voting location information in the schema 505 will point to contests, propositions, and  
1015 candidates, which will point to a ballot ID. For pointing to pre-constructed electronic blank ballots located  
1016 elsewhere, the ballot ID can be loaded with an identifier of the ballot or a URL to its location.

## 1017 **Count (510)**

1018 The count message defined by this schema is used to communicate the results of one or more contests  
1019 that make up one or more elections within an election event. It may also be used to communicate the  
1020 count of a single reporting unit for amalgamation into a complete count.

1021 The message includes the election event identifier, and for each election, the election identifier, an  
1022 optional reference to the election rule being used and information concerning the set of contests.

1023 In some cases, reporting for a contest may be required at a lower level (for example, for each county in a  
1024 state). For this reason, reporting may be done at the level of the reporting unit, the total votes, or for a  
1025 total vote and the breakdown according to the multiple reporting units.

1026 Each contest indicates its identifier, and optionally the counting system and the maximum number of  
1027 votes that each voter could cast. The key information is that about the votes cast for each of the choices  
1028 available and the numbers of abstentions and rejected and uncounted votes. If a vote is rejected, for  
1029 example, because a voter has chosen to spoil a ballot paper, many authorities will want to count that vote  
1030 as having been cast. The `UncountedVotes` element is reserved for those cases where that record is not  
1031 required, for example when the result is thought to be fraudulent. Both the `UncountedVotes` and

1032 RejectedVotes elements have Reason (optional) and ReasonCode (mandatory) attributes to indicate  
 1033 why the votes were treated as they have been. The former is a textual description, and the latter a code.  
 1034 For each choice available to the voter, the identifier and number of valid votes are mandatory. The other  
 1035 information provided depends on the type of election. For example, the Value attribute of the Selection  
 1036 element can be used to indicate whether a candidate was a first or second choice in an election run under  
 1037 the single transferable vote system. In the simplest cases, the identifier for the candidate (perhaps with  
 1038 the party), the party or the referendum option is given. If the voter was able to vote for a party and provide  
 1039 a preference for candidates within the party, the AffiliationIdentifier element is used, and multiple  
 1040 CandidateIdentifier elements may be used, each with a Count attribute. This count is the result of  
 1041 whatever algorithm has been used to calculate the ranking of the candidates.  
 1042 This schema allows for Simulation and Extrapolation of Counts and subsequently Results. Simulation  
 1043 being the facility to forecast the result of a contest based on the result of another contest. Extrapolation is  
 1044 the facility to forecast the final result of a contest based on the count so far.

## 1045 **Result (520)**

1046 Messages described by this schema can be used to communicate the results of simple election types.  
 1047 One specific use is to provide an input into the calculation algorithm for elections using the additional  
 1048 member system.  
 1049 The main part of the schema is held within the Selection element. This allows a choice of candidate,  
 1050 affiliation or referendum option identifiers to be defined with the position that choice achieved (first,  
 1051 second etc). Optionally, the number of votes can be shown. A candidate can be associated with his or her  
 1052 affiliation if required. Write in candidates will be shown in the same way as other candidates, although  
 1053 they will only have an Id attribute if this is assigned in the election system after the votes are cast.  
 1054 This schema allows for Simulation and Extrapolation of Results using data from Counts. Simulation being  
 1055 the facility to forecast the result of a contest based on the result of another contest. Extrapolation is the  
 1056 facility to forecast the final result of a contest based on the count so far.

## 1057 **Statistics (530)**

1058 This schema allows for a variety of statistical information to be made available about the counts and  
 1059 results captured in the Counts 510 schema. For example statistics about attendance and votes at each  
 1060 district and county level or by which voting channels have been used.  
 1061 The statistics can be made available through any type of outlet be it Web, TV, SMS etc. and to any type  
 1062 of organization eg news agencies, political parties.

## 1063 **Options Nomination (610)**

1064 This schema is used to submit proposals, for example for a referendum or company AGM. It uses the  
 1065 generic Proposal element to define the proposal itself. One of more proposers can be named and may  
 1066 sign the nomination.

## 1067 **Options Nomination Response (620)**

1068 This message is sent from the election organiser to the proposer to say whether the nomination has been  
 1069 accepted. Along with the acceptance information and the basic information of election, contest and  
 1070 identifier for the proposal, a remark can be made explaining the decision.

## 1071 **Additional Rules**

Error Code	Error Description
3620-001	If the nomination has not been accepted, a reason for rejection is required in the Remark element

1072 **Options List (630)**

1073 This schema is used for messages transferring lists of proposals for a referendum. It may identify the  
1074 election event, and provides details about the election. Each proposal in a referendum counts as an  
1075 election, so each election identified will hold a single proposal.

1076

1077

---

1078 **6 Conformance**

1079 To conform to this specification, a system **MUST** implement all parts of this specification that are relevant  
1080 to the interfaces for which conformance is claimed. The required schema set will normally be part of the  
1081 conformance criteria and should indicate schema version numbers. For example, the specification for an  
1082 election list system might specify that a conforming system must accept and generate XML messages  
1083 conforming to the following illustrative capability matrix:

Schema	Accept	Generate
EML110	V5.0, V6.0, V7.0	
EML310	V5.0, V6.0, V7.0	
EML330		V7.0
EML340		V7.0
EML350		V7.0
EML360		V7.0

1084  
1085 The data being exchanged **SHOULD** be validated for accuracy using both the XSD schema and the CAM  
1086 templates to ensure that it is conformant.  
1087 We **RECOMMEND** that the business process flows and associated message exchanges are fully  
1088 documented.  
1089 By adhering to these conformance criteria, a system will then be fully compliant with the relevant parts of  
1090 the EML specification and the accompanying schemas.

1091

---

## Appendix A. Acknowledgements

1092 The following individuals have participated in the creation of this specification and are gratefully  
1093 acknowledged:

1094 **Participants:**

1095 The following individuals in addition to the editors have participated in the creation of this Specification  
1096 and are gratefully acknowledged:

1097 **Participants:**

1098 Joseph Hall, University of California, Berkeley

1099 John Ross, Secstan

1100 Sven Rubben, IBM

1101 Peter Zelechowski, ES&S

1102 Richard Cardone, IBM

1103 Neal McBurnett, Individual

1104 John Wack, NIST

1105 Carmelo Montanez-Rivera, NIST

---

## 1106 **Appendix B. Other Considerations**

### 1107 **B.1 Security**

1108 This section presents a general discussion of many of the security considerations commonly found in  
1109 many election environments. As presented previously, these standards apply at EML interface points and  
1110 define data security mechanisms at such interface points. This document is not intended to provide a  
1111 complete description, nor a set of requirements for, secure election systems. In fact, the data security  
1112 mechanisms described in this document are all optional, enabling compliance with these standards  
1113 without regard for system security at all.

1114 This discussion is included here simply to show how the information passed through the various  
1115 interfaces described in these standards could be secured and used to help meet some of the  
1116 requirements commonly found in some elections scenarios.

### 1117 **Basic Security Requirements**

1118 The security governing an election starts before the actual vote casting. It is not only a matter of securing  
1119 the location where the votes are stored. An intensive analysis into security related concerns and possible  
1120 threats that could in one way or another affect the election event resulted in the following:

- 1121 • Security considerations of e-voting systems include:
- 1122 • Authentication
- 1123 • Privacy/Confidentiality
- 1124 • Integrity
- 1125 • Non-repudiation

### 1126 **Authentication**

1127 This is checking the truth of a claim of identity or right to vote. It aims to answer questions such as “Who  
1128 are you and do you have the right to vote?”

1129 There are two aspects of authentication in e-voting systems:

- 1130 • Checking a claim of identity
- 1131 • Checking a right to vote.

1132 In some e-voting scenarios the two aspects of authentication, checking a claim of identity and checking a  
1133 right to vote, may be closely linked. Having checked the identity of the voter, a list of authorized voters  
1134 may be used to check the right to vote.

1135 In other scenarios the voter’s identity must remain private and must not be revealed by a ballot. In which  
1136 case some systems may provide a clear separation between checking of the claim of identity, which may  
1137 be done some time before the ballot takes place, from checking the right to vote at the time of the vote is  
1138 cast. Alternatively, other mechanism may be used to ensure the privacy of the voter’s identity on cast  
1139 votes (i.e. by anonymizing the ballot).

1140 In the physical voting world, authentication of identity is made by using verifiable characteristics of the  
1141 voter like handwritten signatures, address, etc and physical evidence like physical IDs; driver’s license,  
1142 employee ID, Passport etc, all of this can be termed a physical ‘credential’. This is often done at the time  
1143 an electoral register is set up, which can be well before the actual ballot takes place.

1144 Checking the authenticity of the right to vote may be performed at various stages in the process. Initial  
1145 authenticity checks may be done related to the voter’s identity during registration.

1146 Where an election scenario demands anonymity of the voter and privacy of the voter’s ballot, the identity  
1147 of the voter and the cast votes must be separated at some time within the voting process. This can be  
1148 done in several ways by a voting system including, but not restricted to, the following options:

1149 Authentication of the right to vote by itself does not reveal a voter's identity, but does verify he has a  
1150 legitimate right to vote (e.g. the VToken data provides authentication of the right to vote but has  
1151 anonymous properties as to the identification of the person voting).

1152 An voter's identity and the right to vote are both validated (i.e. the VToken data has both 'voter  
1153 identification' and 'right to vote' authentication properties) and then the cast votes are clearly separated  
1154 from the identity of the voter (i.e. the voters identification occurs before the ballot is 'anonymized')

1155 In all cases any verification of the authenticity that takes place after the voter has indicated his/her  
1156 choices must preserve the privacy of those choices according to the laws of the jurisdiction and the  
1157 election rules.

1158 Finally, when counting and auditing votes it is necessary to be able to check that the votes were placed  
1159 by those whose right to vote has been authenticated.

1160 Public democratic elections in particular will place specific demands on the trust and quality of the  
1161 authentication data. Because of this and because different implementations will use different mechanisms  
1162 to provide the voter credential, precise mechanisms are outside the scope of this document.

### 1163 **Privacy/Confidentiality**

1164 This is concerned with ensuring information about voters and how votes are cast is not revealed except  
1165 as necessary to count and audit the votes. In most cases, it must not be possible to find out how a  
1166 particular voter voted. Also, before an election is completed, it should not be possible to obtain a count of  
1167 how votes are being cast.

1168 Where the user is remote from the voting system then there is a danger of voting information being  
1169 revealed to someone listening in to the communications. This is commonly stopped by encrypting data as  
1170 it passes over the communications network.

1171 The other major threat to the confidentiality of votes is within the system that is collecting votes. It should  
1172 not be possible for malicious software that can collect votes to infiltrate the voting system. Risks of  
1173 malicious software may be reduced by physical controls, careful audit of the system operation and other  
1174 means of protecting the voting systems.

1175 Furthermore, the results of voting should not be accessible until the election is complete. Potential  
1176 approaches to meeting this goal might include access control mechanisms, very careful procedural  
1177 control over the voting system, and various methods of protecting the election data using encryption  
1178 techniques.

### 1179 **Integrity**

1180 This is concerned with ensuring that ballot options and votes are correct and unaltered. Having  
1181 established the choices within a particular ballot and the voter community to which these choices apply,  
1182 the correct ballot information must be presented to each voter. Also, when a vote is placed it is important  
1183 that the vote is kept correctly until required for counting and auditing purposes.

1184 Using authentication check codes on information being sent to and from a remote voter's terminal over a  
1185 communications network generally protects against attacks on the integrity of ballot information and  
1186 votes. Integrity of the ballot and voting information held within computer systems may be protected to a  
1187 degree by physical controls and careful audit of the system operation. However, much greater confidence  
1188 in the integrity of voting information can be achieved by using digital signatures or some similar  
1189 cryptographic protection to "seal" the data.

1190 The fundamental challenge to be met is one of maintaining voter privacy and maintaining the integrity of  
1191 the ballot.

### 1192 **Non-Repudiation**

1193 Non-repudiation is a derivative of the identification problem. Identification in e-voting requires that the  
1194 system provide some level of assurance that the persons representing themselves as valid participants  
1195 (voters, election workers, etc.) are, in fact, who they claim to be. Non-repudiation requires that the system  
1196 provides some level of assurance that the identified participant is not able to successfully assert that the  
1197 actions attributed to them via the identification mechanism were, in fact, performed by someone else. The

1198 two requirements are related in that a system with a perfect identification mechanism and undisputable  
1199 proof of all actions would leave no room for successful repudiation claims.

1200 Non-repudiation also requires that the system provide assurance that data or actions properly associated  
1201 with an identified participant can be shown to have remained unaltered once submitted or performed. For  
1202 example, approved candidate lists should be verified as having come from an authorized election worker,  
1203 and voted ballots from a valid voter. In both cases the system should also provide a way to ensure that  
1204 the data has remained unchanged since the participant prepared it.

1205 Non-repudiation is not only a technical quality of the system. It also requires a certain amount of pure  
1206 policy, depending on the technology selected. For example, in a digital signature environment, signed  
1207 data can be very reliably attributed to the holder of the private key(s), and can be shown to be  
1208 subsequently unmodified. The policy behind the acceptance of these properties, however, must be very  
1209 clear about the responsibilities of the private key holders and the required procedures for reporting lost or  
1210 stolen private keys. Further, and especially in “mixed-mode” elections (where voters can chose between  
1211 multiple methods of voting), it may often be desirable to introduce trusted time stamps into the election  
1212 data stream, which could be used to help determine acceptance criteria between ballots, or help resolve  
1213 issues with respect to the relative occurrence of particular events (e.g. ballot cast and lost keys reported).  
1214 The presence of the time information itself would not necessarily enable automatic resolution of these  
1215 types of issues, but by providing a clear ordering of events could provide data that can be fed into  
1216 decisions to be made according to established election policy.

## 1217 **Terms**

1218 The following security terms are used in this document:

- 1219 • Identity Authentication: the means by which a voter registration system checks the validity of the  
1220 claimed identity.
- 1221 • Right to vote authentication: the means by which the voting system checks the validity of a voter's  
1222 right to vote.
- 1223 • VToken: the means by which a voter proves to an e-voting system that he/she has the right to vote in  
1224 a contest.
- 1225 • VToken Qualified: the means by which a VToken can be qualified. The reason for the qualification is  
1226 always appended to a VToken that is qualified. For example, a qualified VToken may be issued to a  
1227 challenged voter.
- 1228 • Vote sealing: the means by which the integrity of voting data (ballot choices, vote cast against a given  
1229 VToken) can be protected (e.g. using a digital signature or other authentication code) so that it can be  
1230 proved that a voter's authentication and one or more votes are related.

## 1231 **Specific Security Requirements**

1232 Electronic voting systems have some very specific security requirements that include:

- 1233 • Only legitimate voters are allowed to vote (i.e. voters must be authenticated as having the right to  
1234 cast a vote)
- 1235 • Only one set of choices is allowed per voter, per contest
- 1236 • The vote cannot be altered from the voter's intention
- 1237 • The vote may not be observed until the proper time
- 1238 • The voting system must be accountable and auditable
- 1239 • Information used to authenticate the voter or his/her right to vote should be protected against misuse  
1240 (e.g. passwords should be protected from copying)
- 1241 • Voter privacy must be maintained according to the laws of the election jurisdiction. (Legal  
1242 requirements of public elections in various countries conflict. Some countries require that the vote  
1243 cannot be tracked back to the voter's identity, while others mandate that it must be possible to track  
1244 every vote to a legitimate voter's identity)
- 1245 • The casting options available to the voter must be genuine

- 1246 • Proof that all genuine votes have been accurately counted.
- 1247 There are some specific complications that arise with respect to security and electronic voting that  
1248 include:
- 1249 • Several technologies may be employed in the voting environment
- 1250 • The voting environment may be made up of systems from multiple vendors
- 1251 • A voter may have the option to vote through alternative delivery channels (i.e. physically presenting  
1252 themselves at a polling station, by post, by electronic means)
- 1253 • The voting systems need to be able to meet various national legal requirements and local voting rules  
1254 for both private and public elections
- 1255 • Need to verify that all votes are recorded properly without having access to the original input
- 1256 • The mechanism used for voter authentication may vary depending on legal requirements of the  
1257 contest, the voter registration and the e-voting systems for private and public elections
- 1258 • The user may be voting from an insecure environment (e.g. a PC with no anti-virus checking or user  
1259 access controls).
- 1260 In addition, the objectives of security architectures for electronic voting systems should include:
- 1261 • Being open
- 1262 • Not restricting the authentication mechanisms provided by e-voting systems
- 1263 • Specifying the security characteristic required of an implementation, allowing for freedom in its  
1264 precise implementation.
- 1265 • Providing the means to exercise security isolation and controls at interfaces between various election  
1266 processes, thereby providing the ability to implement isolated trusted logic processes to meet  
1267 dedicated functions of an election service. Process security isolation ensures that one voting sub-  
1268 process does not inadvertently affect another voting sub-process thereby undermining the whole  
1269 voting system.

## 1270 **Security Architecture**

1271 The architecture proposed here is designed to meet the security requirements and objectives detailed  
1272 above, allowing for the security complications of e-voting systems listed.

1273 The architecture is illustrated in figure 3a below, and consists of distinct areas:

- 1274 • Voter identification and registration
- 1275 • Right to vote authentication
- 1276 • Protecting exchanges with remote voters
- 1277 • Validating Right to Vote and contest vote sealing
- 1278 • Vote confidentiality.
- 1279 • Candidate list Integrity
- 1280 • Vote counting accuracy
- 1281 • Voting system security controls.

## 1282 **Voter identification and registration**

1283 The Voter identification and registration is used to identify an entity (e.g. person) for the purpose of  
1284 registering the person has a right to vote in one or more contests, thus identifying legitimate voters. The  
1285 security characteristics for voter identification are to be able to authenticate the identity of the legal person  
1286 allowed to vote in a contest and to authenticate each person's voting rights. The precise method of voter  
1287 identification is not defined here, as it will be specific to particular voting environments, and designed to  
1288 meet specific legal requirements, private or public election and contest rules. The voter registration  
1289 system may interact with the e-voting system and other systems to define how to authenticate a voter for  
1290 a particular contest.

1291 Voter identification and registration ensures that only legitimate voters are allowed to register for voting.  
1292 Successful voter registration will eventually result in legitimate voters being given a means of proving their  
1293 right to vote to the voting system in a contest. Depending on national requirements or specific voting  
1294 rules/bylaws the voter may or may not need to be anonymous. If the voter is to be anonymous, then there  
1295 must not be a way of identifying a person by the means used to authenticate a right to vote to the e-voting  
1296 system. Right to vote authentication is the means of ensuring a person has the right to cast a vote, but it  
1297 is not the identification of the person.

## 1298 **Right to vote authentication**

1299 Proof of the right to vote is done by means of the VToken, which is generated for the purpose of  
1300 authentication that the voter has a legitimate right to vote in a particular contest.

1301 The security characteristic of the VToken and hence its precise contents may vary depend on the precise  
1302 requirements of a contest, the supplier of the voter registration system, the e-voting system, the voting  
1303 channel or other parts of the electoral environment. Thus, the content of the VToken will vary to  
1304 accommodate a range of authentication mechanisms that could be used, including; pin and password,  
1305 encoded or cryptographic based password, hardware tokens, digital signatures, etc.

1306 The contents of the VToken may also depend on the requirements of a particular contest, which may  
1307 mandate a particular method be used to identify the person and the voter. For example, if a country has a  
1308 national identity card system, it could be used for the dual purpose of identifying the person and providing  
1309 proof that the person is entitled to vote, provided the legal system (or the voting rules of a private election)  
1310 allow a personal identity to be associated with a vote. However, this would not work for countries or  
1311 private voting scenarios that require the voter to be anonymous. For such a contest the mechanism used  
1312 to identify that a person has the right to cast a vote must not reveal the identity of the actual person, thus  
1313 under such voting rules voter identity authentication and right to vote authentication do not use the same  
1314 information or semantics.

1315 The security characteristic required of the VToken may also vary depending on legal requirements of a  
1316 country or electoral rules used in a particular contest. Also, the threats to misuse of VTokens will depend  
1317 to a large degree on the voting channels used (e.g. physical presence at voting station, Internet, mobile  
1318 phone). Bearing this in mind the XML schema of the VToken components must allow for various data  
1319 types of authentication information to be contained within it.

1320 It must be possible to prove that a VToken is associated with a vote cast and the rules of the contest are  
1321 followed, such as only one vote being allowed per voter, per contest. Thus providing proof /non-  
1322 repudiation that all votes were genuine, they were cast in accordance with the rules of the contest, that no  
1323 vote has been altered in any way and that all the votes counted in a contest were valid when audited.

1324 Depending on the legal requirements of a country or electoral rules a voter may be challenged as to the  
1325 right to vote, or may be given a temporary right to vote. In such cases the VToken may need to be  
1326 qualified with a reason. In this document this is called a VToken Qualified. Before a vote is considered  
1327 legitimate and counted the reason for the qualification must have been suitably scrutinized, which could  
1328 be done by the voting officials.

## 1329 **Protecting exchanges with remote voters**

1330 The VToken may be generated as part of the registration system, the e-voting system, or as interaction  
1331 between various components of a voting environment, as illustrate in Figure 3a. The VToken will need to  
1332 be provided securely to the voter so that this can be used to prove the right to vote.

1333 The exchange of information when casting a vote must be protected by secure channels to ensure the  
1334 confidentiality, integrity of voting data (VToken(s) and vote(s) cast) and that this is correctly delivered to  
1335 the authenticated e-voting system. If the channel isn't inherently secure then this will require additional  
1336 protection using other mechanisms. Possible mechanisms might include: a postal system with sealed  
1337 envelopes, dedicated phone channel, secure e-mail, secure internet link (SSL), peer to peer server/client  
1338 authentication and a seal.

1339 Wherever technically possible the exchange of information should be secured and integrity guaranteed  
1340 even if non-secure communications channels are used.

## 1341 **Validation right to vote and contest vote sealing**

1342 When a vote is cast, to ensure that it cannot be altered from the voter's intention, all the information used  
1343 to authenticate the right to vote and define the vote cast must be sealed to ensure the integrity and non-  
1344 repudiability of the vote. This seal may be implemented using several mechanisms ranging from digital  
1345 signatures (XML and CMS), cryptographic seals, trusted timestamps and other undefined mechanisms.  
1346 The seal provides the following security functions:

- 1347 • The vote cannot be altered from the voter's intention
- 1348 • The voting system is accountable and auditable.

1349 The right to vote may be validated at the time the vote was cast. If votes are not checked for validity  
1350 before sealing then the right to vote must be validated at the time that votes are subsequently counted.  
1351 Also when counting, or otherwise checking votes, the validity of the seal must be checked.

1352 If votes are sealed and recorded without being checked for validity at the time they were cast, then the  
1353 time that the vote was cast must be included in the seal, so that they may be checked for validity before  
1354 they are counted.

1355 In some election scenarios it is required to audit a vote cast to a particular voter, in this case a record is  
1356 also needed of the allocation of a VToken to a voter's identity. Such systems also provide non-repudiation  
1357 of the voter's actions. In such cases a voter cannot claim to have not voted or to have voted a different  
1358 way, or that his vote was not counted. In many election scenarios where this type of auditing is required, it  
1359 must not be easy to associate a VToken to the Voter's identity, therefore this type of records must be  
1360 under strict control and protected by security mechanism and procedures, such as; encryption, key  
1361 escrow and security operating procedures.

## 1362 **Vote Confidentiality**

1363 All cast votes must not be observed until the proper time. This requires confidentiality of the vote over the  
1364 voting period but how this is achieved will vary from e-voting system to e-voting system. Mechanism of  
1365 vote confidentiality, range from trust in the e-voting systems internal security functions (processes and  
1366 mechanisms) to encryption of the data, with key escrow tools.

## 1367 **Candidate List integrity**

1368 To ensure that the voter is present and that the candidate list is genuine, there must be a secure channel  
1369 between the voting system and the person voting or the data must be sealed. The approach selected  
1370 must ensure that there is no man-in-the-middle that can change a vote from what the voter intended.  
1371 There are various ways this requirement can be met, ranging from the candidate list having unpredictable  
1372 characteristics with a trusted path to convey that information to the voter, to trust placed in the complete  
1373 ballot/vote delivery channel.

1374 As an example, there may be a secure path to convey the VToken to the person entitled to vote, a way of  
1375 ensuring that a voter is always presented with a genuine list of candidates might be to encode the  
1376 candidate list as part of a sealed VToken.

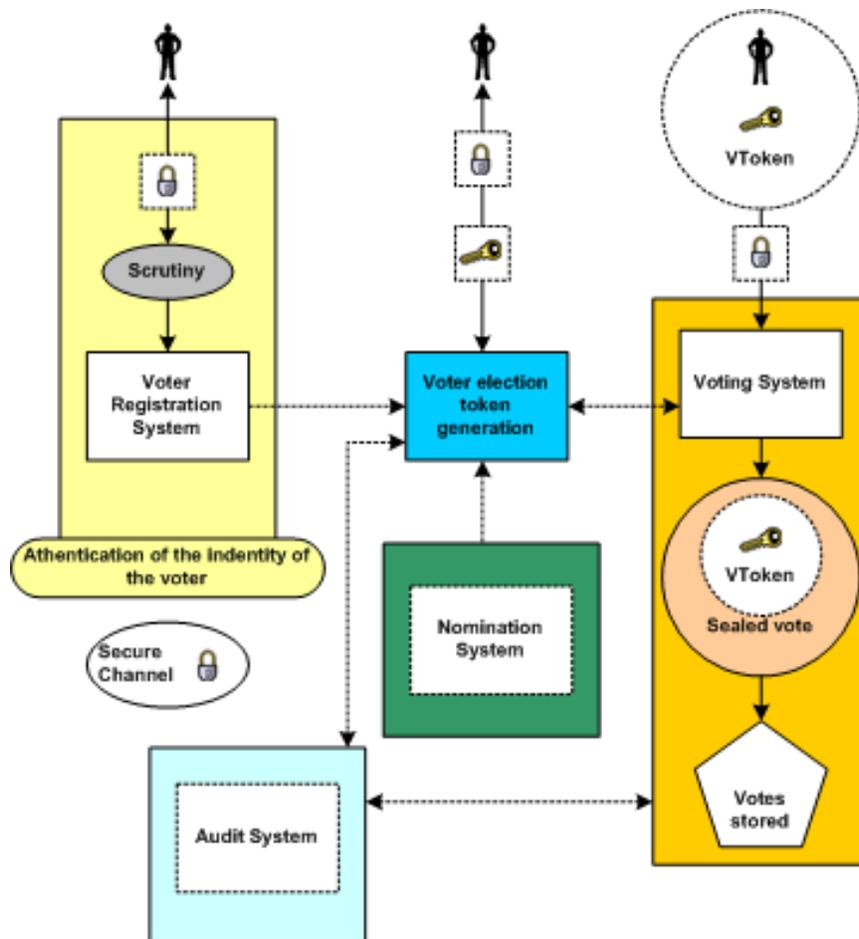
1377 In summary, there must be a way of ensuring the validity of the ballot options and voter selection.

## 1378 **Vote counting accuracy**

1379 Audit of the system must be able to prove that all vote casts were genuine and that all genuine votes  
1380 were included within the vote count. Voters may need to be able to exercise that proof should they so  
1381 desire. Thus auditing needs data that has non-repudiation characteristics, such as the VToken/vote  
1382 sealing, see schema 470 and 480.

## 1383 **Voting System Security**

1384 The overall operation of the voting systems and its physical environment must be secure. Appropriate  
1385 procedural, physical and computing system controls must be in place to ensure that risks to the e-voting  
1386 systems are met. There must be a documented security policy based upon a risk analysis, which  
1387 identifies the security objectives and necessary security controls.



1388  
1389 *Figure 3D: Voting system security*

1390 **Remote voting security concerns**

1391 Many new election systems are currently under evaluation. These systems tend to offer deployment  
1392 options in which the communication between the voter and the election officials is carried out in an  
1393 environment that is not completely under the control and monitoring of the election officials and/or  
1394 election observers (e.g., the Internet, private network, telephones, cable TV networks, etc.). In these  
1395 'remote' or 'unattended' environments, several particular security concerns and questions like:

- 1396 • How do I know that the candidate information I am being presented with is the correct  
1397 information?
- 1398 • How do I know that my vote will be recorded properly?
- 1399 • How do I know there isn't a man-in-the-middle who is going to alter my vote when I place it?
- 1400 • How do I know that it is the genuine e-voting server I'm connected to that will record my vote rather  
1401 than one impersonating it that's just going to throw my vote away?
- 1402 • How do I know that some component of the system does not have malicious software which will  
1403 attempt to alter the ballot choices as represented to me or alter my election?

1404 The type and importance of a particular contest will have an effect on whether the above concerns exist  
1405 and whether they do, or do not, represent a tangible threat to the voting process and its outcome. The  
1406 table listed at Appendix B2 shows the concerns that have been identified as possibilities for one such  
1407 remote or unattended environment (the Internet) that could be used in public election voting scenarios.  
1408 The table shows how the concerns can be translated to technical threats and characterizes security  
1409 services that may be used to counter such threats. Many of the items are not unique to the Internet, and  
1410 can serve as a useful reference or starting point in developing similar threat analysis for other digital

1411 and/or unattended voting environments. How the security services are implemented in any particular  
 1412 environment or deployment is outside the scope of this document allowing freedom to the system  
 1413 providers.

1414 **B.2 Internet Voting Security Concerns**

Concerns raised on Internet voting		Resulting Technical Threats	Possible generic security service countermeasure
1.	<p>Impersonation of the right to vote.</p> <p>The concern here is that a person attempts to impersonate to be a legitimate voter when he/she is not.</p> <p>The initial task of verifying that a person has the right to vote must be part of the voter registration process.</p>	<p>Inadequate, incorrect or improper identification of person during registration of voters</p>	<p>Trusted voter identification and registration using:</p> <p>Security Procedures.</p> <p>Best Practices.</p> <p>Secure communications channels.</p> <p>The voter registration authority must follow standard Security Operating Procedures (SOPs) which ensure due diligence has been done.</p>
	<p>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.</p>	<p>Inadequate privacy of the exchange between the person and the electoral system during voter registration</p>	<p>Channel between voter and registration system must provide:</p> <p>Connection Confidentiality</p> <p>Connection Integrity</p>
2.	<p>Voter is not presented with correct ballot information due to incorrect candidate identification.</p>	<p>Incorrect identification during candidate registration.</p>	<p>Trusted candidate identification and registration are needed using:</p> <ul style="list-style-type: none"> <li>- Security Procedures.</li> <li>- Best Practices.</li> <li>- Secure communications channels.</li> <li>- Authentication and identification of candidates</li> </ul> <p>The candidate registration must follow standard Security Operating Procedures (SOPs) which ensure due diligence has been done.</p>

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

8	How do I know the voting system records votes properly	Integrity of the voting system	Non-repudiation of the vote
			Non-repudiation the vote was cast by a genuine voter
			Audit of voting system
			Connection confidentiality
		Insecure channel between the voter and the vote casting point	Connection Integrity
			Connection Confidentially
		Voter's intent is recorded accurately	Trusted path between voter and vote recording
			Non-repudiation of the vote recorded
		Proof that a genuine vote has been accurately counted	Audit
9	How can I be sure the voting system will not disclose whom I have voted for	Voter's identification is revealed	Voter's identification is anonymous
			Vote confidentiality
10	How can it be sure that my vote has been recorded	Loss of vote	Proof of vote submission
11	How can I be sure there is no man-in-the- middle that can alter my ballot	Vulnerable client environment;	Physical security
			Procedural security
		Trojan horses	Unpredictable Coded voting information
		Virus	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
		Multiple registration applications	Procedures
		Multiple allocation of voters credentials	Voter Identification Voter authentication
14	The vote cannot be altered from the voter's intention	Vulnerable client	Trusted path from voter's intent to vote record

		environment; Trojan horses Virus	Vote integrity Vote non-repudiation
15	The vote may not be observed until the proper time	Votes may be observed before the end of the contest	Voter confidentiality
16	The voting system must be accountable and auditable		Non-repudiation of vote data. Audit tools
17	Identification and authentication information to and from the voter must be privacy protected	Loss of privacy	Channel to provide: Connection Confidentiality
18	The voter's actual identity may need to be anonymous	Voter's identification is revealed Denial of service attack	Voter's identification is anonymous
19	Denied access to electronic voting station		This needs to be counted by engineering the system to provide survivability when under denial of service attack.

### 1415 **B.3 The Timestamp Schema**

1416 Although used as part of EML, this schema has been put in a separate namespace as it is not an integral  
1417 part of the language. A time-stamp binds a date and time to the sealed data. The time-stamp seal also  
1418 protects the integrity of the data. The structure of the time-stamp is similar to the structure of an XML  
1419 Signature.

1420 The timestamp structure may be used in one of two ways either:

- 1421 • Using Internet RFC 3161 binary encoded time-stamp token with the time-stamp information repeated  
1422 in XML,
- 1423 • Using a pure XML encoded time-stamp.

1424 In the case of the RFC 3161 based time-stamp, the Timestamp structure is used as follows:

- 1425 • within TimestampedInfo:
- 1426 • TSTOrSignatureMethod identifies RFC 3161.
- 1427 • Reference contains the URI reference of the voting data being time-stamped. The DigestValue sub  
1428 element contains the digest of the voting data being time-stamped.
- 1429 • TSTXMLInfoReference is not present in this case.
- 1430 • SignatureOrTSTValue holds the RFC 3161 time-stamp token applied to the digest of  
1431 TimestampedInfo. The TimestampedInfo is transformed to a canonical form using the method  
1432 identified in CanonicalizationMethod before the digest algorithm is applied.
- 1433 • KeyInfo contains any relevant certificate or key information.

1434 Object contains the TSTXMLInfo element which is a copy of the information in SignatureOrTSTValue  
1435 converted from RFC 3161 to XML encoding. The TSTXMLInfo element contains:

- 1436 • the version of time-stamp token format. This would be set to version 1
- 1437 • the time-stamping policy applied by the authority issuing the time-stamp,
- 1438 • the time-stamp token serial number,

- 1439 • the time that the token was issued, the contents of this element indicate the time of the timestamp.
- 1440 • optionally an indication as to whether the time-stamps are always issued in the order that requests
- 1441 are received
- 1442 • optionally a nonce<sup>1</sup> given in the request for the time-stamp token,
- 1443 • optionally the identity of the time-stamping authority
- 1444 In the case of a pure XML encoded time-stamp, the Timestamp structure is used as follows:
- 1445 • within TimestampedInfo,
- 1446 • TSTOrSignatureMethod identifies the algorithm used to create the signature value.
- 1447 • Reference contains the URI reference of the voting data being time-stamped. The DigestValue sub
- 1448 element contains the digest of the voting data being time-stamped.
- 1449 • TSTXMLInfoReference must be present, and contains the URI reference of TSTXMLInfo as
- 1450 contained within the Object element. The DigestValue sub element contains the digest of the
- 1451 TSTXMLInfo.
- 1452 • SignatureOrTSTValue contains the signature value calculated over the TimestampedInfo using the
- 1453 signature algorithm identified in TSTOrSignatureMethod having been transformed to a canonical form
- 1454 using the method identified in CanonicalizationMethod. This signature is created by the time-stamping
- 1455 authority.
- 1456 • KeyInfo contains any relevant certificate or key information.
- 1457 Object contains the XML encoded time-stamp information in an TSTXMLInfo element. The contents of
- 1458 TSTXMLInfo is the similar as for the case described above. However, in this case the information is
- 1459 directly signed by the time-stamping authority. The TSTXMLInfo element contains:
- 1460 • version of time-stamp token format: This would be set to version 2
- 1461 • the time-stamping policy applied by the authority issuing the time-stamp,
- 1462 • the time-stamp token serial number,
- 1463 • the time that the token was issued, this is the time of the timestamp.
- 1464 • optionally an indication as to whether the time-stamps are always issued in the order that requests
- 1465 were received
- 1466 • optionally a nonce given in the request for the time-stamp token,
- 1467 • optionally the identity of the time-stamping authority.

## 1468 **B.4 W3C XML Digital Signature**

1469 Some information on the digital signature is included here, but for full information refer to the

1470 Recommendation at <http://www.w3.org/TR/xmlldsig-core/>

1471 An XML Signature consists of:

- 1472 • SignedInfo which includes a sequence of references to the data being signed with the digest (eg.
- 1473 SHA-1 hash) of the data being signed

---

<sup>1</sup> A nonce is a parameter that varies over time and is used as a defence against a replay attack.

- 1474 • SignatureValue which contains the signature value calculated over the SignedInfo using the signature  
1475 algorithm identified in SignatureMethod having been transformed to a canonical form using the  
1476 method identified in CanonicalizationMethod
- 1477 • KeyInfo contains any relevant certificate or key information.
- 1478 • Object can contain any other information relevant to the signature
  
- 1479

1480

## Appendix C. Processing using Schematron or CAM

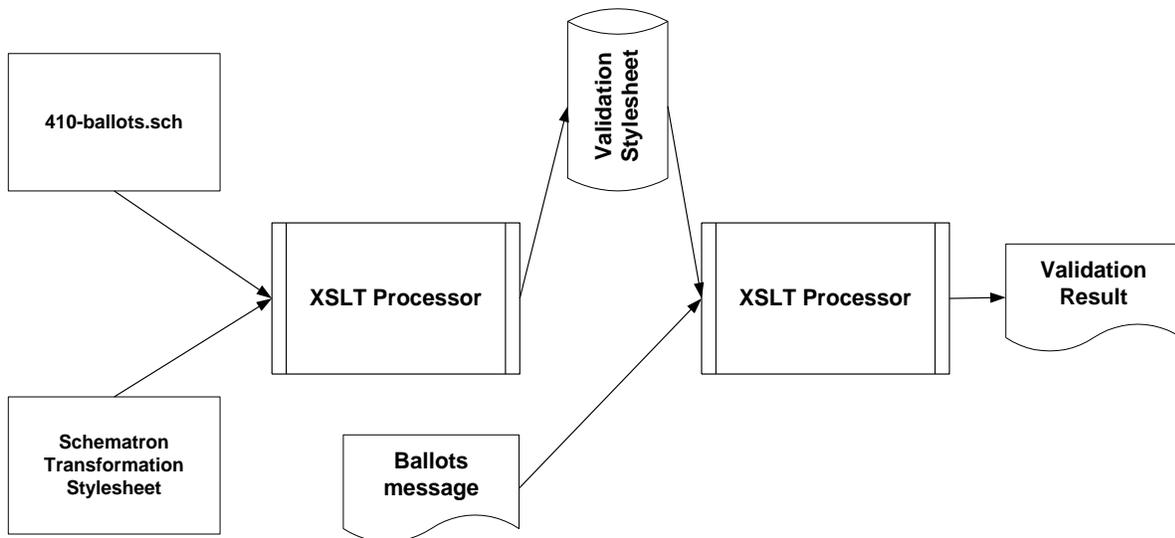
1481 This section gives a short introduction to how validation can be achieved using either Schematron  
1482 schemas or the OASIS CAM template approach. For Schematron this is done either using an XSLT  
1483 processor tool (such as Saxon), or by direct validation using the Schematron schemas and a dedicated  
1484 Schematron processor. For CAM templates this is using a conforming implementation toolkit such as the  
1485 camprocessor project on SourceForge.net as open source.

### 1486 Validation using Schematron Schemas

1487 A Schematron schema is an XML document that can be converted to XSLT using an XSLT stylesheet.  
1488 There is a published stylesheet (skeleton1-5.xslt) that can be used to achieve this. This produces an  
1489 HTML output from the validation. A separate stylesheet can be produced that will create an output to the  
1490 specification below. This stylesheet can import the skeleton and just over-ride those aspects where  
1491 changes are required.

1492 This stylesheet can be used once on each Schematron schema to produce the XSLT file that will be used  
1493 for validating a specific message type. This stylesheet is then used to transform the incoming EML  
1494 message into an error report based on the additional constraints.

1495 The process is shown in the diagram below.

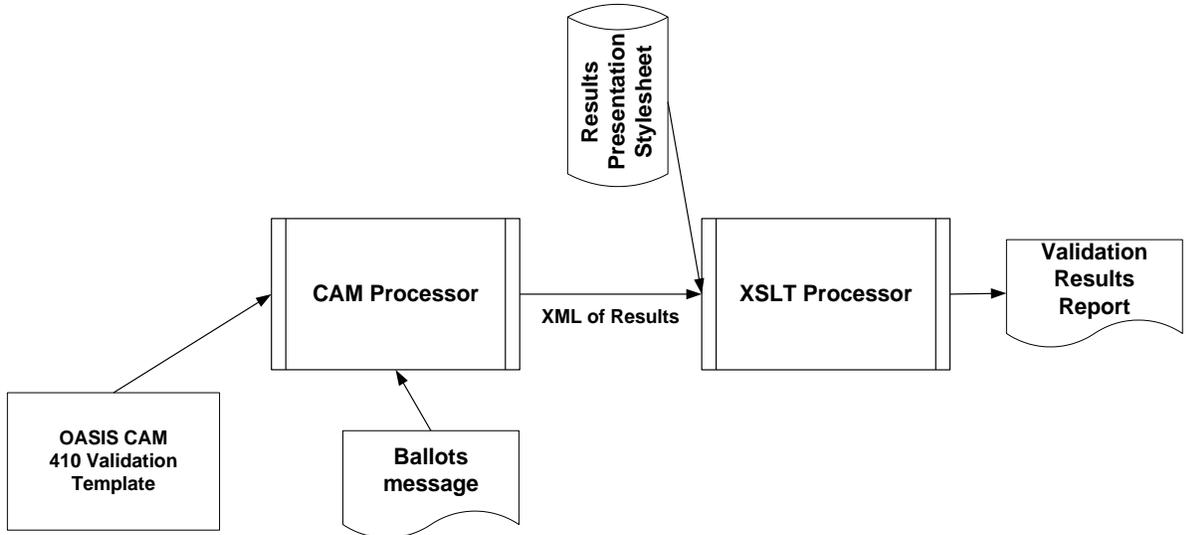


1496

### 1497 Validation using OASIS CAM Templates

1498 An OASIS CAM (Content Assembly Mechanism) Template is an XML document that provides the ability  
1499 to rapidly tailor the XSD schema structure definitions in the base EML standard to suit country  
1500 localizations and rules. The CAM template can then be used to validate the particular implementation  
1501 XML transactions. An open source toolkit is available that implements the OASIS CAM specification. A  
1502 default template can be generated using this toolkit by ingesting the particular EML XSD schema, and  
1503 then tailoring that to produce a country localization pick list and customizations of the content rules. The  
1504 toolkit will also allow the generation of realistic example XML test case instances, localization  
1505 documentation, models and dictionary content.

1506 Once test cases and templates are available then these can be validated using the CAM toolkit. The  
1507 process is shown in the diagram below.



1508

1509

---

## Appendix D. Revision History

1510

Revision	Date	Editors	Changes Made
01	17 June 2011	John Borrás (TC Chair), David Webber (Oracle)	Initial draft
02	24 June 2011	John Borrás (TC Chair), David Webber (Oracle)	Draft for approval as a Committee Specification Draft

1511