



Statewide Voter Registration System Project

California Secretary of State

II.4 – VoteCal System EMS Integration and Data Exchange Specifications Document

FINAL v6.0

June 29, 2016

REVISION SUMMARY

VERSION #	CHANGE DATE	AUTHOR	COMMENT
0.1	12/17/2013	Matt Kaiser	Initial Draft
0.2	02/06/2014	Phil Spencer	Updated following pre-submission review
0.3	02/14/2014	Matt Kaiser	Updates following pre-submission review
0.4	02/14/2014	Tulika Srivastava	QM Review
0.5	02/17/2014	Tina Moore	QA Review
1.0	02/24/2014	Greg Darrah	Final Review and Submission
1.1	04/14/2014	Matt Kaiser	Updates from formal review CTS
1.2	04/14/2014	Tulika Srivastava	QM Review
1.3	04/14/2014	Tina Moore	QA Review
2.0	04/18/2014	Mike Rockenstein	Final Review and Submission
2.1	07/24/2014	Matt Kaiser	Updates for III.6; updated related deliverable table, Appendix B, and Appendix D
2.2	07/28/2014	Phil Spencer	Updated section 6.1.1 to indicate “ChangeWorkflowState” operation as driver for changes to the message state and to indicate the acknowledgment of the data received Added detail for the use of Validation and Application faults in Section 6.2 Added Enumeration type to parameter types listing in Section 6.2
2.3	08/12/2014	Tulika Srivastava	QM Review
2.4	08/13/2014	Tina Moore	QA Review
3.0	08/14/2014	Greg Darrah	Final Review and Submission
3.1	05/15/2015	Phillip Spencer	Updated synchronization check process with details and updated various synch check diagrams. Removed voter activity from the synch check process as per Change Request #14.
3.2	06/01/2015	Tulika Srivastava	QM Review
3.3	06/02/2015	Tina Moore	QA Review
4.0	06/03/2015	Mike Rockenstein	Final Review and Submission
4.1	05/07/2015	Matt Kaiser	Updates to Appendix B for V.3 SDTSSC submission
4.2	09/23/2015	Mike Rahimi	QM Review
4.3	09/23/2015	Tina Moore	QA Review

5.0	09/23/2015	Mike Rockenstein	Final Review and Submission
5.1	05/09/2016	Terri Foster	Updated states to remove “END” and change “SENT” to “DATA SENT”
6.0	06/29/2016	Mike Rockenstein	Final Review and Submission

Document Approval

Signatures below indicate approval of the attached document.

Project Name: Statewide Voter Registration System Project

Document Name: II.4 – VoteCal System EMS Integration and Data Exchange Specifications Document

Version Number: 6.0

Deliverable Author: Matt Kaiser

Submitted To: Kimberly L. Gauthier, Susan Lapsley, Karey Hart, Sean Malone

Date Submitted: June 29, 2016

Date Response Due: August 3, 2016

Signature:

Kimberly L. Gauthier (VoteCal Project Sponsor)

Date

TABLE OF CONTENTS

1	INTRODUCTION	1
1.1	PURPOSE.....	1
1.2	SCOPE.....	1
1.3	SECURITY REQUIREMENTS.....	1
1.4	INTENDED USE AND AUDIENCE	1
1.5	DOCUMENT OVERVIEW	2
1.6	DEFINITIONS AND ACRONYMS.....	3
1.7	ASSUMPTIONS, DEPENDENCIES, AND CONSTRAINTS	4
1.8	REFERENCED DOCUMENTS	4
1.9	RELATED DELIVERABLES AND WORK PRODUCTS.....	4
	DOCUMENT MAINTENANCE.....	7
2	DATABASE	7
2.1	VOTECAL DATABASE	8
2.2	EMS DATABASE.....	8
3	DATA ELEMENTS AND STANDARDS	8
3.1	VOTECAL DATA ELEMENTS	8
3.2	VOTECAL STANDARD CODE VALUES	8
4	DATA VALIDATION	9
5	BUSINESS PROCESS RULES	10
6	MESSAGE STRUCTURE AND CONTENT	10
6.1	MESSAGING OVERVIEW	10
6.1.1	Notification Messages.....	13
6.1.2	Transactional Messages.....	14
6.2	MESSAGE STRUCTURE	14
7	TRANSACTION TIMING	16
8	EMS COMPLIANCE AND SYNCHRONIZATION	16
8.1	EMS COMPLIANCE	17
8.2	SYNCHRONIZATION OVERVIEW	17
8.2.1	SOS Initiated Synchronization	18
8.2.2	EMS Initiated Synchronization	19
8.2.3	EMS Synchronization Differences Search.....	19
8.3	SYNCHRONIZATION MESSAGES	20
8.4	HASHING SPECIFICATIONS	21
8.5	SYNCHRONIZATION PROCESS TRAINING FOR SOS STAFF	22
9	SECURITY AND NETWORK CONNECTIVITY	22
9.1	COMMUNICATION EXCHANGE	22
9.2	IP ADDRESSING AND PORT NUMBERING	23
9.3	COMMUNICATION PATH ENCRYPTION.....	24

9.4 AUTHENTICATION24

APPENDIX A.....26

APPENDIX B – STANDARD CODE VALUES.....27

APPENDIX C – DEFINITIONS AND ACRONYMS28

APPENDIX D – EMS REQUIREMENTS30

LIST OF TABLES

Table 1 – VoteCal Project Team Use of the EMSIS	1
Table 2 – Related Deliverables and Work Products	5
Table 3 – Definitions and Acronyms	28

LIST OF FIGURES

Figure 1 – EMS Integration Document Map	12
Figure 2 – Notification Services	13
Figure 3 – Notification Messages	15
Figure 4 – SOS Initiated Synch Process	18
Figure 5 – EMS Initiated Synch Process	19
Figure 6 – Synch Differences Search Process	20
Figure 7 – EMS to VoteCal Communication Path	23

1 Introduction

The *Deliverable II.4, VoteCal System EMS Integration and Data Exchange Specifications Document* (EMSIS) describes the specifics of the integration of VoteCal with the two Election Management System (EMS) Remediation Services contractors' software applications that will be in use by the 58 California counties' elections officials' staff. Each EMS must be modified to accommodate the detailed solution design, database design, and data standardization of VoteCal as defined within the EMSIS.

1.1 Purpose

The primary purpose of the EMSIS is to provide specifications for EMS integration with VoteCal. This approach includes providing documentation of the VoteCal database, listing the VoteCal data elements, identifying data elements subject to standardization, providing values for standardized data elements, describing the process and rules for validation of data, providing specifications for integration services available to the EMS, describing transaction timing, defining data synchronization and compliance services, and documenting the security and network infrastructure utilized by VoteCal. The EMSIS describes the VoteCal database, EMS integration specifications, and any additional terms, acronyms, roles, or other items newly introduced via this Deliverable.

As part of preparation of this Deliverable, Team CGI solicited and incorporated input and comments on the Deliverable content from the EMS Remediation Services contractor representatives.

1.2 Scope

The scope of the EMSIS is limited to specifying the requirements for EMS integration with VoteCal, and the methodology by which an EMS will integrate with VoteCal.

1.3 Security Requirements

The VoteCal project considers the EMSIS a non-confidential project document; no special security considerations are required to protect the content of this document in its present form.

1.4 Intended Use and Audience

The intended audience for the EMSIS is the VoteCal Project Team. The VoteCal Project Team uses the document as described in Table 1 – VoteCal Project Team Use of the EMSIS.

Table 1 – VoteCal Project Team Use of the EMSIS

TEAM	DOCUMENT USE
Team CGI	
Infrastructure	Uses the EMSIS to validate all components are defined to provide integration with an EMS, and update the <i>VoteCal System Technical Architecture Documentation (Deliverable II.6)</i> (TAD) as necessary.
Architecture/	Uses the EMSIS to validate all systems are defined and designed to provide

TEAM	DOCUMENT USE
Development	integration with an EMS, and update the TAD and <i>VoteCal System Detailed System Design Specifications (Deliverable II.3) (DSD)</i> as necessary.
Test	Uses the EMSIS to begin developing the system functional, integration, regression, and performance test cases.
Project Management	Uses the EMSIS to verify that the VoteCal solution satisfies the requirements.
SOS VoteCal Project Team	
Business Subject Matter Experts (SMEs)	Uses the EMSIS to verify that business requirements presented are included in the VoteCal solution.
Technical	Uses the EMSIS to verify specific business and technical requirements and that the VoteCal solution meets the described business needs.
Test	Uses the EMSIS to begin developing the detailed user acceptance test cases to validate VoteCal.
Management	Uses the EMSIS to verify the VoteCal solution meets the overall requirements.
Independent Verification and Validation (IV&V)	Uses the EMSIS to cross-check and verify known requirements are included in the VoteCal solution.
EMS Remediation Services Contractors	
EMS Remediation Services Contractor	Uses the EMSIS to remediate its EMS to integrate with VoteCal.

Resources involved in the development, review, and approval of the EMSIS are required to have knowledge and experience with the VoteCal functional requirements, general knowledge of overall relational database design fundamentals, general knowledge of elections systems, practices, and processes in the counties and Secretary of State (SOS), and general knowledge of web-based transactional processing.

1.5 Document Overview

The EMSIS describes the specifics of the integration of VoteCal with the EMS Remediation Services contractors' software applications anticipated to be in use by the 58 California counties' elections officials' staff. Each EMS must be modified to accommodate the detailed solution design, database design, and data standardization of VoteCal as defined within the EMSIS.

The EMSIS defines all known system functions, integration specifications and data standards related to EMS remediation requirements, so that the EMS Remediation Services contractors can make the required modifications to their system for integration with VoteCal.

The EMSIS does not attempt to define what modifications are necessary to the EMS application and how those modifications would be accomplished, but does define services used to integrate with VoteCal.

The EMSIS is organized into the following sections:

- **Section 1: Introduction** – This describes the purpose, document scope, security requirements, intended audience, document overview, definitions and acronyms, referenced documents, related deliverables and work products, document maintenance procedures, and assumptions, dependencies, and constraints related to this Deliverable.
- **Section 2: Database** – This provides an understanding of the VoteCal database as it relates to which data elements will be stored, how the data elements will be stored, and the relationship between VoteCal records and standard values.
- **Section 3: Data Elements and Standards** – This describes and lists the data elements in VoteCal and identifies data elements that are subject to standardization within the EMSs. In addition, standard values are listed for the data elements subject to standardization.
- **Section 4: Data Validation** – This describes how data will be validated when transmitted between VoteCal and the EMS.
- **Section 5: Business Process Rules** – This describes business process rules to enforce data standardization and data validation within VoteCal and the EMSs.
- **Section 6: Message Structure and Contents** – This describes the types, structure, and contents of messages exchanged between VoteCal and the EMSs.
- **Section 7: Transaction Timing** – This describes the sequence and timing of transactions in VoteCal and the EMSs.
- **Section 8: EMS Compliance and Synchronization** – This contains information that defines and describes how each EMS achieves ongoing compliance with system of record data and data standards. The synchronization process describes the methodology and processes related to the synchronization of data between VoteCal and the EMSs, representing a component of the ongoing compliance mechanism.
- **Section 9: Security and Network Connectivity** – This describes the security and network connectivity implementation between VoteCal and the EMSs.
- **Appendix A** – This appendix has been intentionally removed.
- **Appendix B: Standard Code Values** – This appendix provides a complete listing of standard and common data values included in the VoteCal solution.
- **Appendix C: Definitions and Acronyms** – This appendix provides a detailed list of the definitions and acronyms used in the EMSIS which the *VoteCal Project Glossary* does not contain.
- **Appendix D: EMS Requirements** – This appendix provides a list of EMS requirements included in the VoteCal solution, cross-referenced with remediation specifications.

1.6 Definitions and Acronyms

The EMSIS includes definitions and acronyms as cited in the *VoteCal Project Glossary*. Appendix C – Definitions and Acronyms lists additional definitions and acronyms specifically related to this deliverable.

1.7 Assumptions, Dependencies, and Constraints

The following are the known assumptions for the EMSIS:

- At the VoteCal Project Team's discretion, the EMSIS may be conditionally accepted with outstanding deficiencies, provided that these deficiencies do not have any material impact on the content or applicability of the EMSIS.
- There is no known expectation for file exchange between VoteCal and the EMS. All information exchange is through messaging services. Therefore, no file naming standards and file structures are included in the EMSIS.
- The intent of data standardization is to implement a VoteCal system that will store and maintain distinct standard values in the VoteCal database, capture and store data as it is provided from the source and minimize the amount of interruptions or manual procedures needed to complete a registration process.
- Team CGI will work with the EMS Remediation Services contractors as needed to support their use and validation of the EMSIS during the EMS design, development, and testing activities.

There are no other known assumptions, dependencies, or constraints at this time.

1.8 Referenced Documents

The documents, excluding Team CGI Deliverables or work products, were used in the creation of the EMSIS or referenced in subsequent sections of the EMSIS include:

- *State of California Secretary of State, VoteCal Project, System Integrator (SI) contract 12S10048, Statement of Work (SOW)*
- *VoteCal Document Management Plan (Version 1.0) (DMP)*
- *VoteCal Logical Roles (Version 3.0)*
- *VoteCal Project Glossary (Version 3.0)*

1.9 Related Deliverables and Work Products

Table 2 – Related Deliverables and Work Products lists the primary VoteCal Project Team Deliverables and work products that are either used to develop the content of the EMSIS, or the EMSIS is used in their development.

Table 2 – Related Deliverables and Work Products

DELIVERABLE/WORK PRODUCT	RELATIONSHIP	STATUS	VERSION # AND DATE
<i>I.4 – VoteCal Software Version Control and System Configuration Management Plan (SVCSCMP)</i>	The SVCSCMP documents the change control and release management process. The EMS compliance process in the EMSIS relies on the processes in the SVCSCMP.	Accepted	v5.2 04/21/2015
<i>II.1 – VoteCal System Requirements Specifications (SysRS)</i>	The SysRS describe the general architectural design, interface specifications, database description, and processing functions of VoteCal. This document provides the specifications for database design and messaging in the EMSIS.	Accepted	v4.0 06/03/2015
<i>II.2 – VoteCal System Functional Specifications (SysFS)</i>	The SysFS documents specifications for the user application interfaces, business processing logic, data flows, processes, reporting/querying capability, and pre-defined reports and extracts. SysFS design artifacts are used to meet the requirements documented in the functional specifications.	Accepted	v4.0 06/03/2015
II.3 – DSD	The DSD details the technical design for VoteCal. This document provides the specifications for database design, messaging, security, and network connectivity in the EMSIS.	Accepted	v3.0 06/03/2015
<i>II.5 – VoteCal System Detailed Requirements Traceability Matrix (RTM)</i>	The RTM defines the Requirements Traceability Matrix report(s) that are provided for VoteCal. Requirements from the functional specifications are included as part of the RTM for traceability to project artifacts. The RTM will provide cross-referencing for the data elements and data standards sections of the EMSIS.	Accepted	v3.0 08/07/2015
II.6 – TAD	The TAD describes the logical, physical, and implementation details of the entire VoteCal solution. This document will	Accepted	v3.0 06/03/2015

DELIVERABLE/WORK PRODUCT	RELATIONSHIP	STATUS	VERSION # AND DATE
	provide the specifications for database design, messaging, security, and network connectivity for the EMSIS.		
<i>II.7 – VoteCal System Data Model and Data Dictionary (DMDD)</i>	The DMDD defines and documents all the data elements and relationships among them and how the data will be represented and accessed. This document will be developed in concurrence with the EMSIS related to data elements and data standards.	Accepted	v4.0 06/03/2015
<i>II.8 – VoteCal System Data Integration Plan (DIP)</i>	The DIP describes the sequence of steps in data integration, including the integration of multiple records from different counties into a single record for each voter. The EMSIS will define data elements and data standards, and validation methodology that will be used in the mapping and load of county data during implementation.	Accepted	v.2.1 05/15/2014
<i>II.9 – VoteCal System Training Plan (STrgP)</i>	The STrgP describes the materials and methodologies used to train the staff to use VoteCal. The EMSIS identifies post-implementation responsibilities of the SOS staff. The EMSIS provides input to the STrgP regarding these responsibilities.	Accepted	v.1.1 07/08/2014
<i>III.3 – Acceptance Test Plan for Certification of EMS Data Integration and Compliance (CETP)</i>	The CETP describes the activities to test the integration of each county EMS with VoteCal. The plan is used to define the tests necessary to demonstrate EMS compliance with VoteCal data standards and validation.	Accepted	v1.1 05/19/2014
<i>IV.1 – VoteCal System Pilot County Data Integration Completion and Report (PDIR)</i>	The PDIR documents the integration effort, all deficiencies identified during integration, and correction of deficiencies for pilot counties. The report is used to document compliance with or deficiencies in EMS compliance with VoteCal data standards and validation.	To be delivered in Phase IV – Testing	N/A

DELIVERABLE/WORK PRODUCT	RELATIONSHIP	STATUS	VERSION # AND DATE
<i>VI.4 – VoteCal System Remaining County Data Integration Completed and Tested for Compliance and Successful Integration.</i>	The <i>VoteCal System Remaining County Data Integration Completed and Tested for Compliance and Successful Integration</i> documents the integration effort, all deficiencies identified during integration, and correction of deficiencies for counties other than pilot counties. The report is used to document compliance with or deficiencies in EMS compliance with VoteCal data standards and validation.	To be delivered in Phase VI – Deployment and Cutover	N/A
<i>(EMS) II.3 – Proposed EMS Remediation Solution Walkthroughs</i>	The Proposed EMS Remediation Solution Walkthroughs are presentations for EMS Remediation Services contractors' clients on the proposed remediation solution components designed to meet all specifications detailed in the EMSIS. The EMSIS will provide content and direction for the EMS Remediation Services contractors in development of these presentations.	The EMS Remediation Services contractors have responsibility for the content of this plan.	N/A

Document Maintenance

Team CGI reviews and updates the EMSIS using the “Deliverable Management” process defined in the *VoteCal Project Management Plan*, as needed and in accordance with the *Integrated Project Schedule*. Any update to the EMSIS follows approved document management process as documented in the SVCSCMP. When Team CGI changes the document, the author updates the revision history with a brief description of the change, the date of the change, and updated version number. The VoteCal Project Team maintains versions of the EMSIS throughout the life of the project in the VoteCal Project Library.

2 Database

This section provides an understanding of the VoteCal database as it relates to which data elements will be stored, how the data elements will be stored, and the relationship between VoteCal records and standard values.

The VoteCal standard code values are presented in Appendix B – Standard Code Values. Those values are stored in tables in the VoteCal database, and are mirrored in the EMS. The DMDD defines the relationships between VoteCal voter registration records and applicable sets of standard code values associated with VoteCal voter registration records.

References to database deliverables other than this Deliverable are included for information only, and are not required for the purpose of EMS remediation.

2.1 VoteCal Database

The VoteCal database exists within the VoteCal architecture as defined by the TAD. The EMSs do not connect directly to the VoteCal database. The VoteCal database is updated by the EMSs utilizing a set of integration services as described in Section 6 – Message Structure and Content and Appendix D – EMS Requirements.

The VoteCal data elements, how the data elements will be stored, and the relationship between VoteCal records and standard values are presented separately in the DMDD.

2.2 EMS Database

The EMS database exists within the EMSs architecture as defined by the EMS Remediation Services contractors. VoteCal does not connect directly to the EMS database. The EMS databases will be updated with VoteCal data utilizing a set of integration services as described in Section 6 – Message Structure and Content and Appendix D – EMS Requirements.

3 Data Elements and Standards

This section describes and lists the data elements in VoteCal and identifies data elements that are subject to standardization within the EMS software applications. In addition, standard values are listed for the data elements subject to standardization.

3.1 VoteCal Data Elements

VoteCal data elements are defined as tables and columns in the DMDD. Integration with an EMS requires only a subset of VoteCal data elements, and VoteCal integration services provide a layer of data management between an EMS database and the VoteCal database. The data elements utilized by VoteCal integration services are included in data contracts which group together data elements according to the function for which the data elements are used, as opposed to how the data elements are organized in the structure of the database. For example, the data elements involved in the function of registering a voter are grouped together in a single voter data contract, and include data elements that are stored in several tables and columns in the database.

Data contracts and the data elements included in those data contracts that are used by VoteCal integration services are defined in the *VoteCal Service Design Common Contracts* document. The specific data contracts utilized by VoteCal and an EMS in VoteCal integration services are identified as “Applicable to EMS Integration” in the *VoteCal Service Design Common Contracts* document.

Each data contract includes one or more properties. A property of a data contract is equivalent to a data element. Therefore, the properties of data contracts identified as applicable to EMS integration in the *VoteCal Service Design Common Contracts* document delineate VoteCal data elements for integration with an EMS.

3.2 VoteCal Standard Code Values

Please reference Appendix B – Standard Code Values for the list of data elements subject to standardization, and the standard, common values for those data elements subject to standardization. The VoteCal standard code values are maintained by the SOS using the VoteCal user interface as defined in the DSD.

4 Data Validation

This section describes the VoteCal data validation actions that enforce data standardization, validation, and integrity between VoteCal and the EMSs. Data validations enforce standards to ensure that the data meets the expected format, size, and content. The following lists the commonly used types of validations performed by VoteCal:

- **Required Data** – This validation indicates when a parameter is mandatory to the operation. Required data is indicated in the design specifications when the “nullable” attribute marked with a “no”. Optional data is marked with a “yes” for the “nullable” attribute.
- **Allowable or Valid Values** – This validation ensures the provided data exists in the list of acceptable values.
- **Data Type** – This validation ensures the provided data meets the classification as determined by the interface.
- **Size** – This validation ensures the length of the string does not exceed the specified limit. VoteCal does not constrain the minimum size of data but instead provides an indicator for required or optional fields (see Required Data bullet above).
- **Format** – This validation ensures the format of the data meets the provided specifications such as phone numbers or dates with separators.
- **Character Types** – This validation ensures the use of allowable character types such as alphabetic, alphanumeric, and special characters.

Each VoteCal EMS integration service has an operation or set of operations that are utilized to perform an action such as search, retrieve, add, or update data. Each operation contains a set of parameters used to communicate data. The VoteCal application intrinsically supplies validated data to the EMSs and the VoteCal EMS integration services apply validations and business rules to the data received from the EMSs. VoteCal applies data validations at both the operation level and the parameter level.

The business rules work product for each VoteCal EMS integration service identified in Appendix D – EMS Requirements contains the data validations specific to that service. For example, the data validations for the Jury Wheel integration service are located in the Rules: JuryWheelIntgSvc file. This spreadsheet contains a column for the data validations labeled “Validations”. Any parameter or operation that does not contain a data validation merely contains a dash. Each data validation is identified with “#V” followed by a sequential number to indicate a unique validation for a given parameter or operation. Any messages associated to the validations are provided in the column labeled “Messages”. These messages also utilize the same validation indicator of “#V” followed by a number to align the message to its corresponding validation. Additionally, the comments column in the spreadsheet indicates when the validation occurs in the Web Services Definition Language (WSDL) layer. If the WSDL is not indicated as the validation source, then the validation occurs in the application code behind the WSDL technical interface.

5 Business Process Rules

This section describes the business process rules that enforce data standardization and data validation between VoteCal and the EMSs. Business process rules or plainly, business rules, communicate the actions taken to process the information provided to VoteCal by the EMSs.

The business rules work product for each VoteCal EMS integration service contains the business rules specific to that service. For example, the business rules representing the Jury Wheel integration service is the Rules: JuryWheelIntgSvc file. This spreadsheet has a column specifically for the business rules labeled “Business Rule”. The VoteCal EMS integration services apply business rules to the data received at both the operation level and the parameter level. Any parameter or operation that does not contain a business rule merely contains a dash. Each business rule is identified with “#BR” followed by a sequential number to indicate a unique rule for a given parameter or operation.

6 Message Structure and Content

This section describes the types, structure, and contents of messages provided by the VoteCal services to the EMSs.

6.1 Messaging Overview

This section provides an overview of the messaging process between VoteCal and the EMSs. The following description elaborates on the integration architecture documented in the TAD as well as the service designs in the DSD. The TAD describes the service bus layer as the interfacing component of VoteCal that hosts the services that communicate with the EMSs. Within this layer, VoteCal utilizes integration services to provide the messaging capability between VoteCal and the EMSs. VoteCal implements these integration services as web services in the integration layer of the VoteCal application. VoteCal defines two categories of integration service messages as follows:

- **Notification Messages** – These include messages that require a response from the county EMSs as well as messages that provide informational value to the county EMSs. The services utilizing these messages are the EMSInboundIntgSvc service and the EMSOutboundIntgSvc service.
- **Transactional Messages** – These include messages that provide the EMSs with access to create, read, and update specific VoteCal data. The integration services catalog (EMS Integration Service Catalog spreadsheet) lists the services in this category.

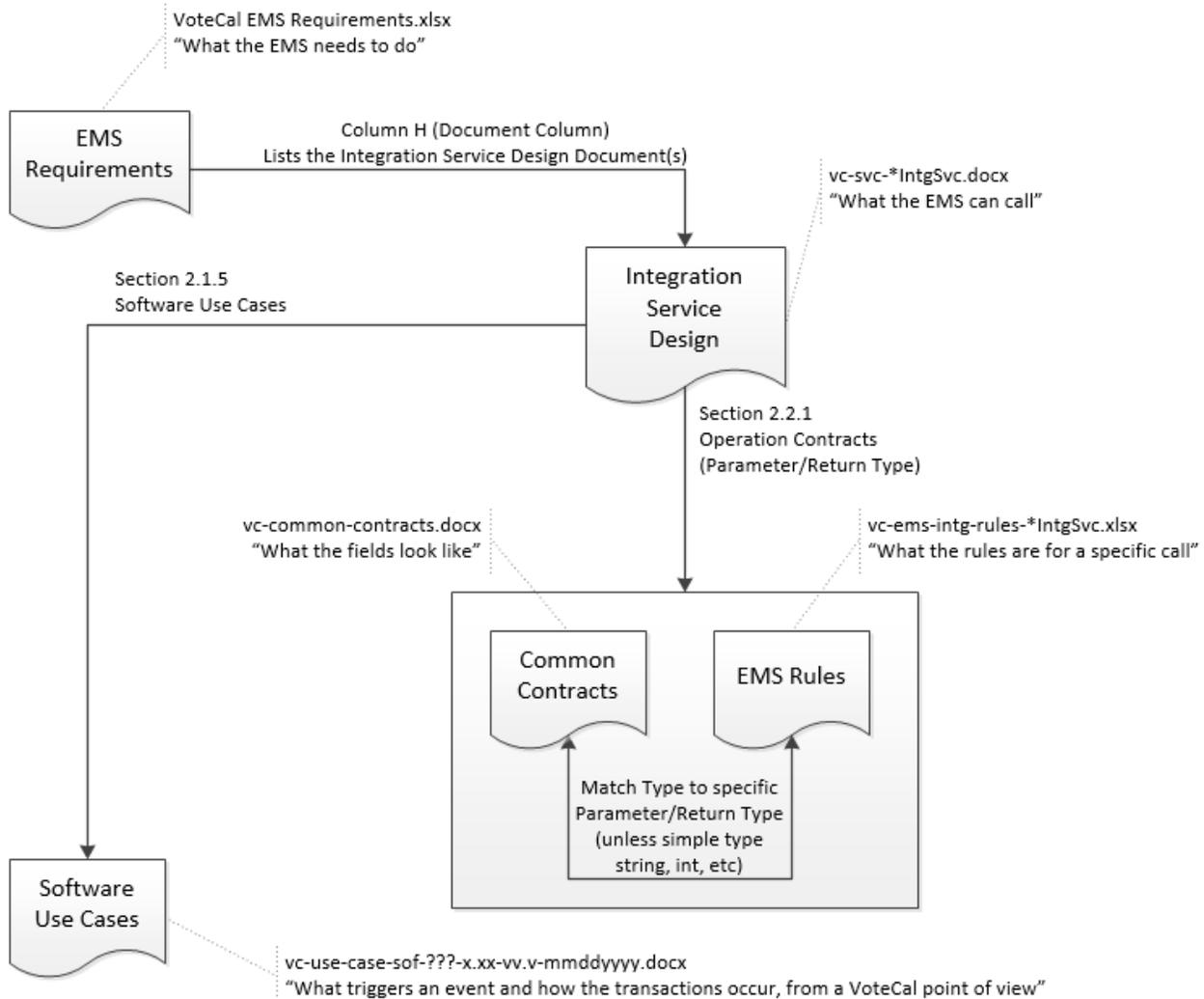
The following set of work products describes the services providing the messaging capability in VoteCal:

- **VoteCal EMS Integration Service Catalog** – This document lists the integration services, their operations, a description of the operation, mode of operation (synchronous or asynchronous), time-out period in seconds, and the timing (on-demand or scheduled).
- **VoteCal EMS Integration Rules** – This set of documents provides the business rules and validations that apply to the messages for services that integrate VoteCal with the EMSs. The VoteCal EMS Integration Rules consist of a Microsoft Excel spreadsheet for each service in the EMS Integration Service Catalog.

- **VoteCal Integration Service Designs** – These documents provide the detail on the services that integrate VoteCal with other applications or systems. The VoteCal Integration Services Catalog contains the list of the integration services that apply to the EMS Integration with VoteCal.
- **VoteCal Service Design Common Contracts** – This document contains the message structure for VoteCal services. VoteCal utilizes a label under the data contract header that states “Applicable to EMS Integration” to indicate the data contracts that are specific to the EMS integration with VoteCal. This document also contains the payload structures for notification-based messaging services.
- **Web Services Definition Language Files** – These files provide the technical endpoint details for a VoteCal integration service thus defining how an EMS must communicate with VoteCal to consume these services. These files describe the operations and messages for a particular integration service as well as its concrete network protocol and message format.

Figure 1 – EMS Integration Document Map provides a mapping of the design documents, their relationships and usage.

Figure 1 – EMS Integration Document Map



Appendix D – EMS Requirements references the requirements that the EMS use to locate the specific service(s) necessary to satisfy that requirement. In addition, the VoteCal EMS Integration Service Catalog provides a list of the services available to the EMS. VoteCal provides an integration service design to describe these services. Each EMS integration service design points to specific software use cases that dictate under what business scenarios the EMS should use that EMS integration service. In addition, the EMS integration service design provides the business functions, also known as operations, which the service provides to the EMS. Each operation lists the parameters that the operation accepts when called, as well as the return value type.

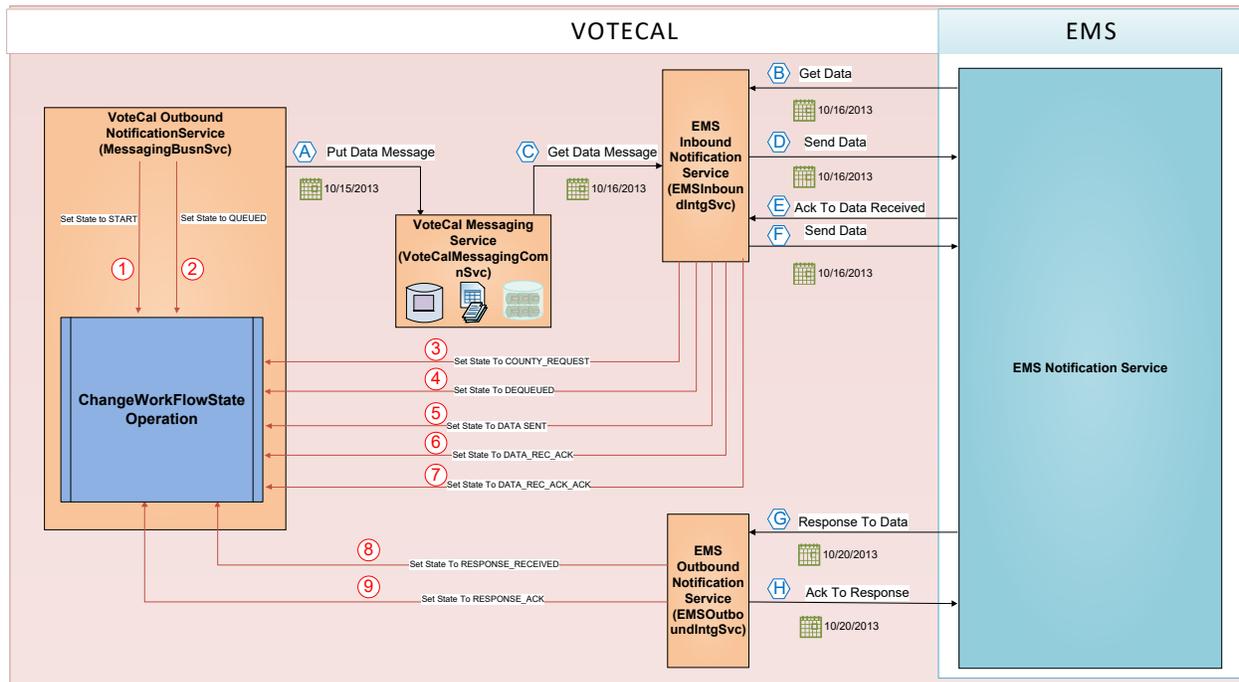
The return value type provided with each operation in the service design indicates the type of data returned on successful completion of the operation. When the provided data does not pass data validations or business rules, a fault is generated and provided back to the EMS. The data validations and business rules are captured in rules documents corresponding to each EMS integration service. Please refer to Section 4 – Data Validation for more information on EMS data validation in VoteCal. VoteCal provides the EMS Integration Rules documents to indicate

under what circumstances a business rule violation occurs or a data validation fails. Please refer to Section 6.2 – Message Structure for details on the fault message structure returned to the EMS from VoteCal.

6.1.1 Notification Messages

VoteCal manages the messages that require action by the EMSs or provide informational value to the EMSs through the VoteCal notification services. VoteCal provides two integration web services to allow messaging capabilities between VoteCal and EMSs as shown in Figure 2 – Notification Services.

Figure 2 – Notification Services



The VoteCal outbound notification service exposes the “ChangeWorkflowState” operation as the input to the state machine for managing the point in the process (the state) that the message resides. In addition, this service is the entry point for the VoteCal application to send messages to the counties. When a message needs to be sent to the county, the VoteCal outbound notification service is invoked and the workflow state is set to “START” (shown as “1” in the figure). Then the message is saved or queued using the VoteCal messaging service (shown as “A” in the figure), and the state is changed to “QUEUED” (shown as “2” in the figure). At this point, the message is ready to be consumed by the county EMS.

The county EMS continually polls the queue using the EMS inbound notification service to retrieve messages (shown as “B” in the figure). At this point, the state is set to “COUNTY_REQUEST” (shown as “3” in the figure). The EMS inbound notification service retrieves the message from the queue using the VoteCal Messaging Service (shown as “C” in the figure) and the state is changed to “DEQUEUED” (shown as “4” in the figure). The EMS inbound notification service sends the data to the EMS (shown as “D” in the figure), and again the state is updated to “DATA SENT” (shown as “5” in the figure). The EMS then saves the information and sends an acknowledgement that the data has been successfully retrieved

(shown as “E” in the figure). The state is moved to “DATA_REC_ACK” (shown as “6” in the figure). After this, VoteCal sends back confirmation (shown as “F” in the figure) that it got the acknowledgement that the data has been successfully retrieved. The state is moved to “DATA_REC_ACK_ACK” (shown as “7” in the figure).

For messages that require a response, the EMS supplies their response along with the pertinent data to the EMS outbound notification service (shown as “G” in the figure), and the state is set to “RESPONSE_RECEIVED” (shown as “8” in the figure). The EMS outbound notification service then provides an acknowledgement (shown as “H” in the figure). The workflow state is set to “RESPONSE_ACK” (shown as “9” in the figure). This is the final state of messages that require an EMS response.

The EMS inbound and outbound notification services use the *VoteCalMessage* common message structure described in the *VoteCal Service Design Common Contracts* document.

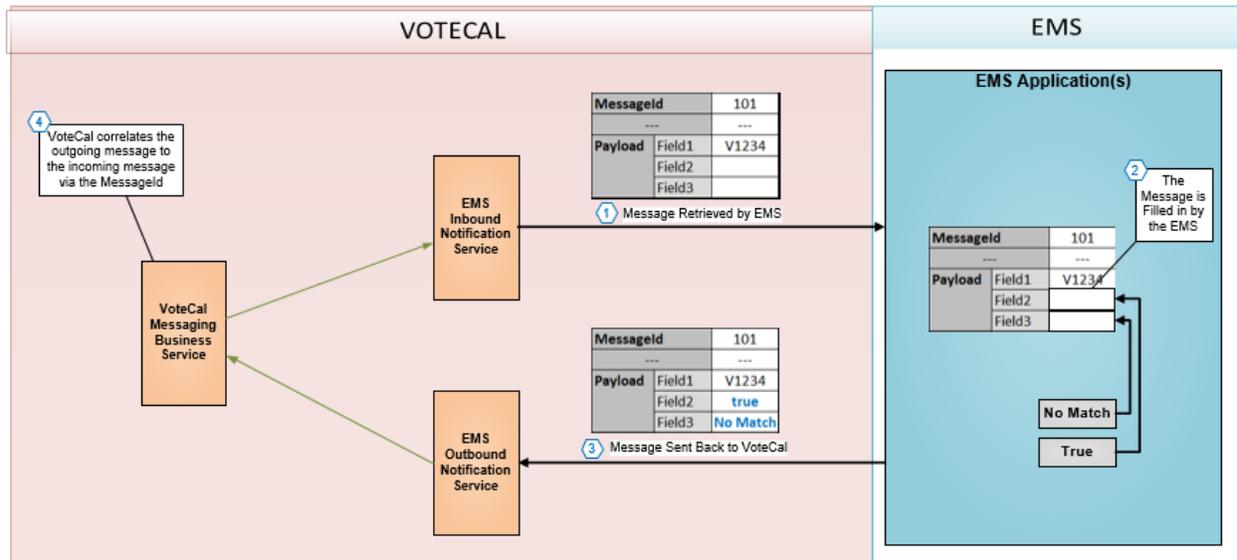
6.1.2 Transactional Messages

VoteCal provides a set of services to support specific business needs to create, retrieve, and update voter and other related data such as political party, district, or election information. Transactional services employ a request-reply pattern where each EMS initiates the request and VoteCal provides the reply. Many of these services act on a specific type of data such as voter participation history and typically provide create, read, and update capability with this data. The integration service catalog located in Appendix D – EMS Requirements lists these integration services along with their available operations.

6.2 Message Structure

The service design for each VoteCal EMS integration service provides the business functions also known as operations that the service provides to the EMS. Each operation lists the parameters that comprise the message structure that the operation expects and returns. These EMS integration services use the *CountyUserInfo* data contract as the last parameter of the operation to capture the county and county user information. Please refer to the *VoteCal Service Design Common Contracts* document for the details on the structure of the *CountyUserInfo* data contract.

VoteCal utilizes a common structure for all notification messages from both the outbound (*EMSOutboundIntgSvc*) and inbound (*EMSInboundIntgSvc*) services. This structure has a common set of fields such as *CountyId*, *MessageId*, message type and status, and message created, acknowledged, and response dates that act to facilitate management and delivery of the message. These messages also contain a payload that represents the container for what is deemed the actual business data. For these notification messages, the structure of the payload is dependent on the message type. Figure 3 – Notification Messages illustrates the usage of the message as it traverses between EMS and VoteCal systems.

Figure 3 – Notification Messages

As shown in the diagram, messages retrieved by the EMS contain a message identifier, the MessageId, as well as a series of other supporting fields (shown as "----") and a payload. The payload contains fields with values prepopulated (e.g., Field1 in the diagram) as well as fields that are to be populated by the EMSs (e.g., Field2 and Field3 in the diagram). The diagram shows a message that requires a response by the EMS. The EMS fills in the necessary data and sends the message back to VoteCal. The VoteCal Messaging Business Service correlates the outgoing message (that was retrieved by the EMS) with the incoming message (that was sent by the EMS to VoteCal) using the MessageId field.

In circumstances when an error occurs or a business rule/data validation violation occurs, VoteCal responds with a subclass of a GenericFaultContract message. The GenericFaultContract is a base class for concrete faults: ValidationFault and ApplicationFault. The ValidationFault contains a List of ApplicationException and the ApplicationFault has only one ApplicationException. The applicationError is a container of the code and error description of the exception being reported to the client. Thus, the GenericFaultContract structure provides a coded value and a string message synchronously back to the EMS. For validation violations, the associated field is also passed in this message. Please refer to the *VoteCal Service Design Common Contracts* for the details of the GenericFaultContract data contract structure. In some instances, these rule violations or data validation failures may require the EMS to take further action to correct the conditions indicated. When this occurs, a notification message is placed in the queue for retrieval and processing by the EMS. The notification message types along with the service that initiates the message are listed in the VoteCal Messages Payload Location document (vc-ems-intg-messages-payload-location.xlsx). This list also indicates which messages require a response from the EMS or are informational in nature.

The EMSInboundIntgSvc and EMSOutboundIntgSvc service design documents contain the specific detail for the VoteCal inbound and outbound notification messaging structures. The EMS Messaging section of the *VoteCal Service Design Common Contracts* document contains the message structure (VoteCalMessage data contract) along with the associated payload structures that corresponds to each message type.

The structure of the transactional messages is located in their corresponding service design documents. For example, the RORIntgSvc design document contains the message structure for the Report of Registration (ROR) integration service.

The parameter types or return value types shown in the service designs or common contracts may be a primitive .NET data type or a VoteCal object. The following lists the data types shown in the VoteCal service designs:

- The value types **byte** (1-byte unsigned integer), **short** (2-byte signed integer), **int** (4-byte signed integer), and **long** (8-byte signed integer).
- The floating-point value types **single** (4-byte floating point) and **double** (8-byte floating point).
- The **decimal** type (16-byte decimal value).
- The **bool** (Boolean) type, which represents a truth value, typically the result of a relational or logical operation. Boolean values should be passed as true or false.
- The **DateTime** type, which represents a date and/or a time. The design indicates when just the date part is used. The default date and time structure follows this structure: The format for DateTime is: YYYY-MM-DDThh:mm:ss. The YYYY represents the century and year. The MM represents the two-digit year. The DD represents the two-digit day. The hh represents a two-digit hour using 24-hour time. The mm represents a two-digit minute. The ss represents a two-digit second. A sample literal value is 2014-02-13T10:03:51. If only a date is needed, then zeroes may be used for the time portion: hh:mm:ss.
- The **string** reference type, which represents a sequence of Unicode characters.
- The Enumeration type, which consists of an ordered set of named constants.

A special data type known as an IList may be used to indicate that a data type or object is provided as an array or list of those data types or objects. For example, an IList<Election> would be a list of election objects. Please reference the *VoteCal Service Design Common Contracts* document when a non-primitive type is declared in the service designs such as Election in the prior example.

7 Transaction Timing

This section describes the sequence and timing of transactions in VoteCal and the EMSs. The timing depends on whether the service is a notification-based service or a transactional service.

For notification-based or transactional messages, the timing is on a “per service basis”. The individual time-out periods are defined in the VoteCal EMS Integration Services Catalog document, and apply only to operations that expect message results from VoteCal (synchronous messages). In case the service operation times out, the integration service returns a timeout message.

8 EMS Compliance and Synchronization

This section contains information that defines and describes how each EMS achieves ongoing compliance with the system of record data and data standards. The synchronization process

describes the methodology and processes related to the synchronization of data between VoteCal and the EMSs, representing a component of the ongoing compliance mechanism.

8.1 EMS Compliance

The EMS achieves initial compliance with acceptance of the *Deliverable III.3, Acceptance Test Plan for Certification of EMS Data Integration and Compliance* and acceptance of *Deliverable IV.1 – VoteCal System Pilot County Data Integration Completion and Report* and *Deliverable VI.4 – VoteCal System Remaining County Data Integration Completed and Tested for Compliance and Successful Integration*. The EMSs achieve ongoing compliance using data standards, data validation, synchronization checks, and release management. Agreements on data standards between the EMSs and SOS establish the basis for ongoing compliance activities. The details of these data standards are described in Section 3 – Data Elements and Standards.

Based on the VoteCal data standards, data validation rules enforce the data types, lengths, and pre-defined valid values. The data validation performed by the VoteCal EMS integration services is described in Section 4 – Data Validation.

As documented in the SVCSCMP, EMS Remediation Services contractors are members of the VoteCal System Configuration Management and Release Team. Changes in the EMS solutions are communicated to the VoteCal System Configuration Management and Release Team. To support the VoteCal system configuration management, the *1.1 – EMS Remediation Plan* for DIMS and DFM describes the EMSs software configuration management activities including the collaboration and communication approach with the SOS and VoteCal Project team.

As another method for determining EMS compliance with the system of record data, the synchronization process identifies data differences between VoteCal and the county data stored in the EMSs. The methodology and processes related to the synchronization of data between VoteCal and the EMSs are described in the following sections.

The data validation and synchronization EMS compliance mechanisms are inherent to the VoteCal application and are described in this document. As a part of the Phase IV training activities, SOS technical staff and elections end-user training covers the use of VoteCal for message monitoring to evaluate data validation as well as the procedures to run a synchronization check.

8.2 Synchronization Overview

This section provides an overview of the synchronization process. The VoteCal synchronization process is one of several mechanisms utilized to ensure that EMS data complies with the VoteCal system of record and standards. Synchronization refers to the process that identifies differences between VoteCal and EMS data including VoteCal data not in the EMS and EMS data not in VoteCal. This process compares and identifies differences by the following areas of data:

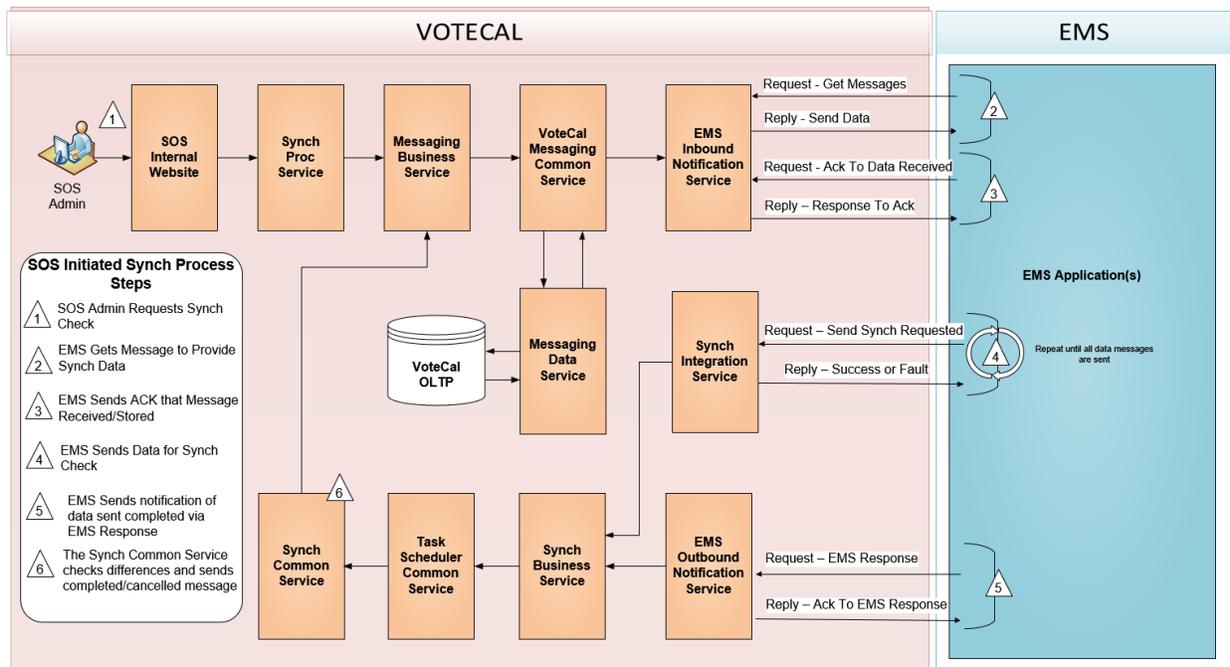
- **Voter** – This data includes voter registration, voter participation history, and images.
- **Precinct-District Data** – This data includes precinct, district, and precinct to district mapping.

The synchronization process is initiated by an authorized administrator either from the SOS or by the county through their EMS. In addition, the EMSs have access to search the differences from the synchronization check.

8.2.1 SOS Initiated Synchronization

An authorized SOS Administrator via the VoteCal user interface may initiate the synchronization check. Figure 4 – SOS Initiated Synch Process illustrates the synchronization process steps.

Figure 4 – SOS Initiated Synch Process



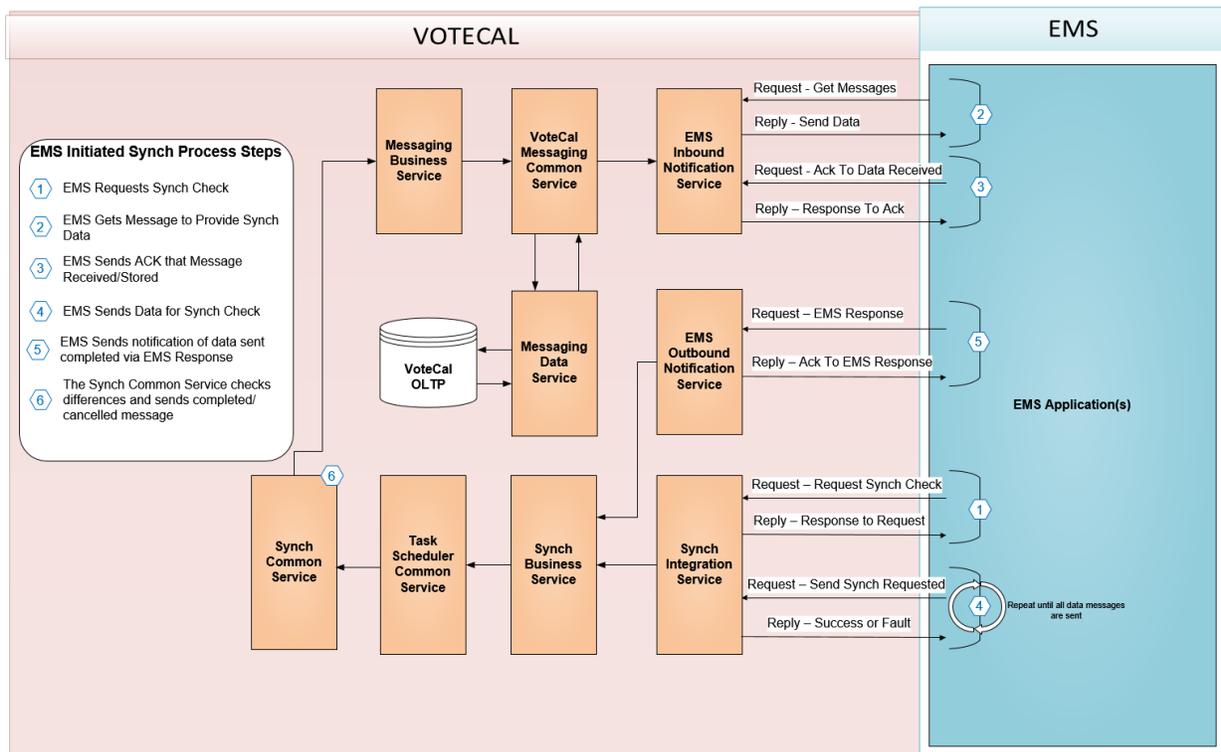
As shown in the diagram, when initiated by an authorized SOS Administrator, the Synch Proc Service (SynchronizationProcSvc) calls the Synchronization Business Service (SynchronizationBusnSvc) to send a notification message. This synchronization request message is queued which means it is persisted in the VoteCal Online Transaction Processing (OLTP) database until the EMS retrieves the message. Once the message is retrieved by the EMS using the EMS inbound notification service (EMSInboundSvc) and the acknowledgement is sent that the EMS has successfully retrieved and stored the message, the request for EMS data has been completed. Next, the EMS must assemble the necessary data including the key fields and hash values for the data to be compared and send this to VoteCal using the SendSynchRequestedPayload operation of the Synchronization Integration Service. The hash values and the hashing process are described in Section 8.4 – Hashing Specifications. The synchronization information to be sent to VoteCal potentially can be a large data set, and transferring or processing this payload can lead to time outs while sending the message. To avoid this issue, the payload is divided in smaller self contained payloads, and sent to VoteCal in sequential messages. If one of the messages fails to be sent, it can be sent again. This asynchronous model decouples the process of sending the data with the actual synchronization process. Once the last message is sent, VoteCal needs to be notified that it can initiate the synchronization process. To accomplish this, the EMS sends a message using the EMS Outbound Integration service. This message contains information that allows VoteCal to initiate

the synchronization process. The VoteCal synchronization process (Synch Common Service) is then scheduled to run. The Synch Common Service sends a completed or cancelled message to the EMS depending on the result of the run.

8.2.2 EMS Initiated Synchronization

The EMSs uses the synchronization integration service (SynchronizationIntgSvc) to search or retrieve the synchronization results. VoteCal groups the results by voter data or precinct-district data.

Figure 5 – EMS Initiated Synch Process



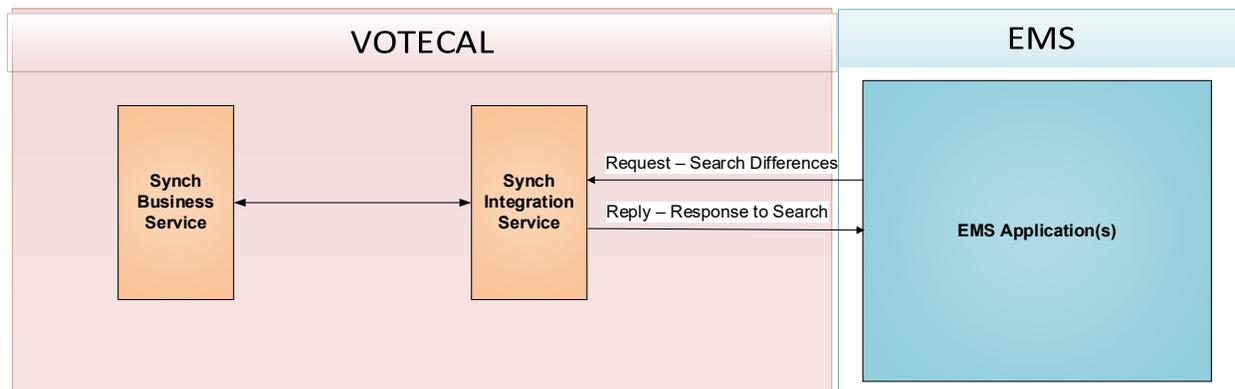
As shown in the diagram, to initiate the synchronization process the EMS invokes a request-reply synchronization operation in the synchronization integration service: Request Synch Check. The synchronization integration service in turn invokes the synchronization business service to send a synchronization request notification on the messaging business service. The messaging business service performs the processing indicated in Section 8.2.1 – SOS Initiated Synchronization and returns, as a result, a list of county ID and scheduled task item ID pairs to the synchronization business service. The synchronization business service passes the list of county ID and scheduled task item ID pairs to the synchronization integration service, which returns this information to the EMS.

8.2.3 EMS Synchronization Differences Search

The EMSs uses the synchronization integration service to search or retrieve the synchronization results. VoteCal also allows an authorized SOS Administrator to search or retrieve the same synchronization results. The user interface designs to support the SOS Administrator access to synchronization results is available in the DSD. Figure 6 – Synch Differences Search Process

illustrates the interaction between the EMS and VoteCal when the EMS searches or retrieves the synchronization differences.

Figure 6 – Synch Differences Search Process



As shown in the diagram, the EMS requests synchronization differences for voter information or precinct-district data using the synchronization integration service. The search criteria for this service allow the EMS to retrieve differences by voter ID or synchronization area such as voter registration or voter participation history. In addition, all differences may be retrieved by providing only the county ID.

The synchronization differences include voter data and precinct-district data differences. The voter data differences include the list of voter IDs along with identifying the area or areas where differences occurred. The differences for voters also indicate voters that are only found in EMS county data or VoteCal data. The following lists the areas of voter data for the reported differences:

- Voter Registration
- Voter Participation History
- Images

The synchronization differences for precinct-district data include the precinct and district IDs along with an indication of those that are only found in the EMS county data or VoteCal data.

8.3 Synchronization Messages

This section describes the messages exchanged between VoteCal and the EMSs as a part of the synchronization process. The following services represent the interaction points for synchronization messaging between VoteCal and the EMSs:

- **EMSInboundIntgSvc** – The EMS inbound integration service allows the EMSs to pick up requests for data messages for a synchronization check as well as completed or cancelled status messages from a synchronization check.
- **EMSOutboundIntgSvc** – The EMS outbound integration service allows the EMSs to provide VoteCal with the data necessary to perform synchronization processing.
- **SynchronizationIntgSvc** – The synchronization integration service provides the EMSs with the entry point to request a synchronization check, sending synchronization

payloads that include the hash data from the County as well as to search the results of a synchronization check.

The EMSInboundIntgSvc service uses a common message structure for all message types and a payload that is specific to a given message type. The common message structure for the EMSInboundIntgSvc is located in the *VoteCal Service Design Common Contracts* document. The following message types are used when communicating with the EMSs regarding synchronization:

- **SynchRequested** – This message requests the data from the EMSs in order to perform the synchronization check.
- **SynchCompleted** – This message signals the EMSs about the completion of the synchronization check. This message also provides the EMSs with the list of voter IDs and precinct-district IDs that have differences.
- **SynchCancelled** – This message signals the EMSs about the cancellation of the synchronization check.

The payload for the above types of messages are detailed in the SynchronizationBusnSvc design document. It is important to note that the SynchCompleted messages contain the list of voter IDs and precinct-district IDs identified to have differences along with a message structure to support the detail behind the voter and precinct-district data. This represents a secondary stage that the EMSs then fills in the detail for the voter and precinct-district data and provides that detail in a return message to the EMSOutboundIntgSvc. This information along with VoteCal detail is persisted for analysis and reporting.

The EMSOutboundIntgSvc service accepts the data from the EMSs to be compared to the VoteCal data as a part of the synchronization check as well as the detailed data indicated above. The SynchronizationBusnSvc design document also defines the message structure for the EMS county data.

The SynchronizationIntgSvc design document details the message structures used by the EMSs when requesting a synchronization check or when searching the differences from a synchronization check.

8.4 Hashing Specifications

This section describes the specifications on the hashing mechanism in VoteCal. Hashing refers to the process that creates a fixed-length value from data of arbitrary length. This allows the system to generate fixed-length output data that acts as a shortened reference to the original data. When implemented by VoteCal and the EMSs, the resulting hash values are used to quickly compare data between the systems.

VoteCal aggregates or concatenates data values and then encrypts using a one-way algorithm. The output, or hash, is assumed unique to the encrypted values with varying degrees of certainty, depending on the algorithm used. The values passed in to the encryption algorithm must be in a standardized format, as the smallest difference (an extra whitespace character, an uppercase letter versus a lowercase letter) will cause the resulting hash to likewise be different. Hashes are calculated at the time that their underlying values are stored in the VoteCal OLTP database. VoteCal uses the Secure Hash Algorithm 1 (SHA-1) as the encryption algorithm, which produces a 160-bit hash value, expressed as a 40-digit hexadecimal number.

In addition to the encryption algorithm, the hashing specifications include the column ordering and row ordering when rolling up multiple rows of data into a higher-level hash. Each collection of database columns to compare are hashed in a specific order and manner. The order of these columns is dependent on each collection. VoteCal converts each column to an upper-case string and trimmed of leading and trailing whitespace prior to concatenation. VoteCal retains the internal whitespace of a column with this process. Whitespace refers to both spaces and tabs. VoteCal converts null values, if they exist, to a zero-length empty string. VoteCal concatenates the values in order without delimiters.

VoteCal uses hashing of hashed values to see if entire groups of data have changed. These “rollups” function in a very similar fashion as the hashing of columns with one important caveat: since the hashing is taking place across rows instead of columns, there is no pre-defined order in which to concatenate the values for hashing. Therefore, VoteCal sorts the rows based on the fields and order (ascending or descending) as specified in the synchronization hash specifications document (Synchronization Hashing Specifications spreadsheet). The hash values from the SHA-1 encryption algorithm, concatenated together without delimiters, trimmed of leading and trailing whitespace are then stored in a 40-byte data-type.

The specific columns and rollup criteria are provided in the synchronization hash specifications referenced in Appendix D – EMS Requirements.

8.5 Synchronization Process Training for SOS Staff

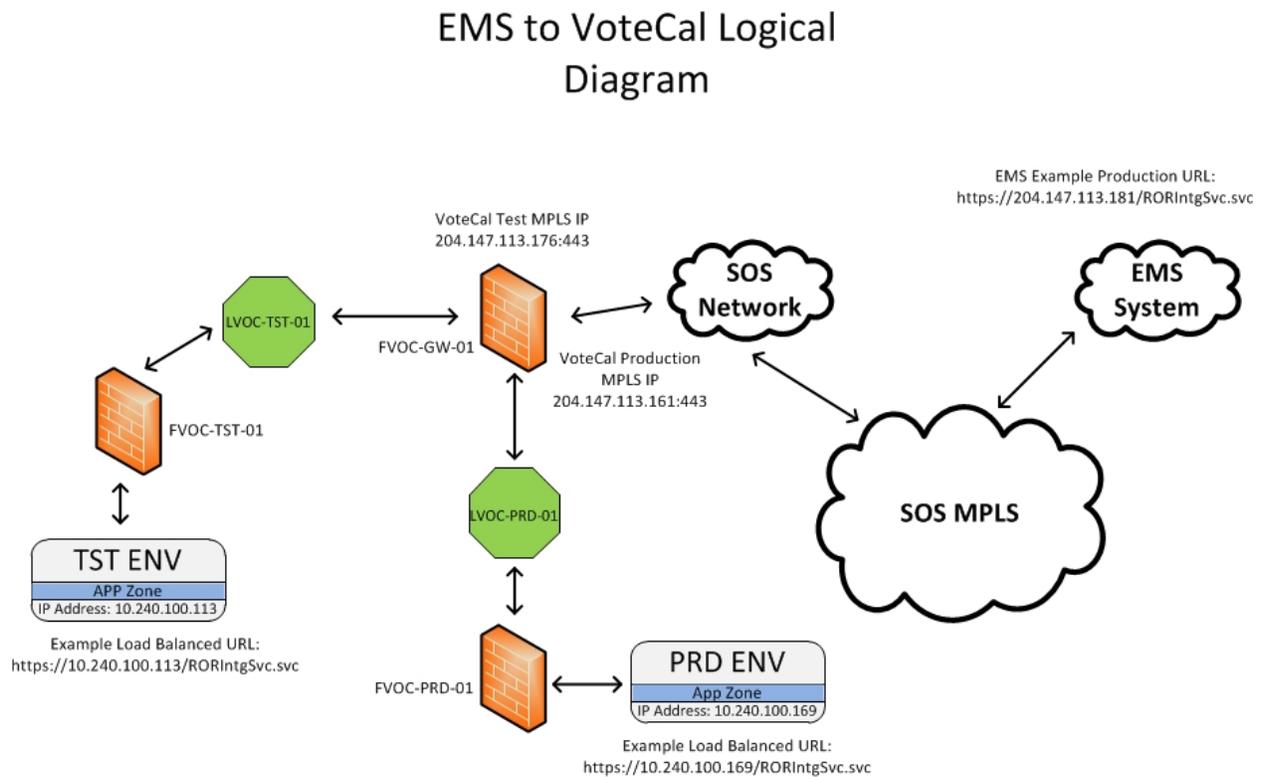
CGI provides training to SOS ITD and Elections Division on the mechanisms and procedures used to ensure continuing EMS compliance with VoteCal data requirements after deployment. Components of this training includes message monitoring to evaluate compliance with data standards and data validations. Training includes procedures for performing and evaluating data across systems using VoteCal’s synchronization check functionality and the release management methodology used by EMS Remediation Services contractors and the VoteCal Project Team to coordinate software releases. STrgP details the approach and plan for providing this component of training to SOS staff.

9 Security and Network Connectivity

This section describes the security and network connectivity implementation between VoteCal and the EMSs.

9.1 Communication Exchange

Figure 7 – EMS to VoteCal Communication Path provides a visual representation of the network connectivity and communication exchange between the EMSs and VoteCal.

Figure 7 – EMS to VoteCal Communication Path

As depicted in the diagram, accessibility to VoteCal by the EMSs is facilitated through connections from the SOS Multiprotocol Label Switch (MPLS) network. VoteCal is connected to the SOS network located in Sacramento, CA and uses a firewall solution to process and evaluate data traffic permitted to access VoteCal. The EMSs initiate communication with the VoteCal production environment via a single internet protocol (IP) address which is the same for all counties. Data flows from the EMSs through the SOS MPLS cloud to a VoteCal virtual firewall (gateway firewall) located in the SOS Data Center. The VoteCal gateway firewall interrogates traffic flow, performs network address translation (NAT) to the VoteCal internal IP addressing schema, and routes traffic to the appropriate VoteCal load balancer. This load balancer applies processes to identify preferred policies and forwards data to the targeted VoteCal end-point server. Response server traffic is transmitted by the targeted VoteCal end-point server and exits VoteCal via the gateway firewall directing traffic back through the SOS MPLS network to each EMS. During Phase IV – Testing, the EMS Remediation Services contractors utilize the same communication path described above for appropriate routing to the test environment. VoteCal accommodates 64,329 concurrent connections across the EMSs. In the extremely unlikely event that this number is exceeded, the EMSs receive a connection timeout when attempting to access VoteCal services. It is advised that the EMSs proactively close open connections that are no longer being used to free up available connections.

9.2 IP Addressing and Port Numbering

The VoteCal production environment uses a single IP address as the destination for the EMSs. Additionally, the VoteCal test environment uses a single IP address as the destination for the EMS Remediation Services contractors during Phase IV – Testing. Publication of the IP address

to be used for access to the test environment by the EMS Remediation Services contractors is provided by Team CGI to the EMS Remediation Services contractors during Phase III – Development. Publication of the IP address to be used for access to the production environment by the EMSs is provided by Team CGI to the EMSs prior to Phase V – Pilot Deployment and Testing. All communications to VoteCal from the EMSs use a single port (port 443).

9.3 Communication Path Encryption

VoteCal secures the communication path between the EMSs and VoteCal by using Secure Socket Layer (SSL)/Transport Layer Security (TLS) encryption. SSL/TLS end points located in VoteCal utilize IIS web services running on the production integration servers. The following further specifies the SSL/TLS versioning and encryption policies used to secure the communication path between VoteCal and the EMSs:

- Cipher Suite – TLS1_DHE_RSA_AES_256_CBC_SHA
 - Version: SSLv3/TLS1.2
 - Public Key Exchange: Diffie-Hellman (DHE)
 - Authentication: RSA (2048 bit)
 - Encryption Type: Advanced Encryption Standard (AES) (256 bit)
 - Mode: Cipher-block Chaining (CBC)
 - Message Authentication Code (MAC) Hash: SHA-1 or SHA-256
- Certification Authority (CA): SOS required CA

9.4 Authentication

The VoteCal solution integrates with the existing Microsoft Active Directory (AD) implementation at the SOS Data Center to authenticate and authorize users and user roles to the VoteCal infrastructure platform. For the purposes of EMS authentication to VoteCal, each county EMS is assigned a unique AD service account based on the county in which it resides. All EMSs utilize their assigned service account to access VoteCal services. In addition, a county code attribute is identified for each EMS service account within the VoteCal AD to provide authorization privileges within the VoteCal application.

The SOS creates the password associated with each EMS service account and provides this information to the appropriate EMS Remediation Services contractor. Password complexity for the EMS service accounts consists of the following:

- Minimum password length of 18 characters
- Includes both uppercase and lowercase letters
- Includes numbers
- Includes special characters
- Passwords do not contain similar characters

For the initial implementation of VoteCal, the EMS service account password is set to not expire. The SOS plans to upgrade their AD infrastructure to a more recent version of Microsoft

AD providing fine-grained password capabilities. Following this upgrade, the EMS service account password expiration implementation is reviewed and adjusted as required by the SOS.

Appendix A

This appendix has been intentionally removed.

Appendix B – Standard Code Values

Please reference the VoteCal database for a complete listing of standard and common data values included in the VoteCal solution.

Standard code values are documented and maintained within the VoteCal database with a table for each set of codes.

Standard code values have been extracted from the database into a Microsoft Excel spreadsheet named vc-emsis-standard-values.xlsx.

Appendix C – Definitions and Acronyms

Table 3 – Definitions and Acronyms provides terms and descriptions of unique definitions and acronyms used in the EMSIS.

Table 3 – Definitions and Acronyms

TERM	ACRONYM	DEFINITION
Active Directory	AD	A directory service provided by Microsoft providing user authentication and authorization services.
Advanced Encryption Standard	AES	A standard specification for the encryption of electronic data.
Certification Authority	CA	An entity that issues digital certificates.
Cipher-block Chaining	CBC	A cryptographic mode of operation used to provide an information service such as authenticity or confidentiality.
Diffie-Hellman	DHE	A mode of exchanging cryptographic keys allowing two parties to establish a shared connection.
Internet Protocol	IP	A numeric label assigned to a device on a computer network.
Message Authentication Code	MAC	A short piece of information used to authenticate a message and to provide integrity and authenticity assurances on the message.
Multiprotocol Label Switching	MPLS	A mechanism in high performance telecommunications network that directs data from one network node to the next based on short path labels rather than long network addresses, avoiding complex lookups in a routing table.
Network Address Translation	NAT	A network protocol that allows multiple devices to connect to a network using the same public IP address.
Online Transaction Processing	OLTP	A type of database that facilitates and manages transaction-oriented applications.
Secure Hash Algorithm	SHA	A family of cryptographic hash functions published by the National Institute of Standards and Technology (NIST) as a U.S. Federal Information Processing Standard (FIPS).
Secure Socket Layer	SSL	A protocol designed to provide communication security over the internet.
Transport Layer	TLS	A protocol that is designed to provide communication security over the internet using

TERM	ACRONYM	DEFINITION
Security		certificates to encrypt transmitted data.
Web Service Definition Language	WSDL	An XML-based interface description language that is used for describing the functionality offered by a web service including how the service can be called, what parameters it expects, and what data structures it returns.

Appendix D – EMS Requirements

Please reference the *VoteCal EMS Requirements* work product (vc-emsis-ems-requirements.xlsx) related to this deliverable for a complete listing of EMS requirements included in the VoteCal solution, cross-referenced with remediation specifications. Please reference the “Introduction” tab (worksheet) for descriptions of each column of the work product, and an inventory of the supporting documents. This work product is in a Microsoft Excel spreadsheet. As the spreadsheet is extensive, it is located in the VoteCal Project Library.