

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

PA OVR Web API Specification

PA Online Voter Registration (PAOVR)

Version 1.0

Produced for:

PA OVR Web API Posting Entities

Produced by:

Pennsylvania Department of State

Harrisburg, PA

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

REVISION HISTORY

Date	Version	Description	Author
7/26/2016	0.0	Initial Publication	Moser
04/16/2017	0.1	Ready for Review	TenHuisen
04/21/2017	0.2	Feedback Incorporated, Added Release Notes and Project Timeline	TenHuisen
05/03/2017	0.3	Added batch vs interactive differentiation, field processing order, expected response codes, clarifications to technical description	TenHuisen
05/23/2017	1.0	Clarification in multi-language support, interactive mode, and batch mode with continueAppSubmit=1 and miscellaneous formatting and clarity issues.	TenHuisen

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

TABLE OF CONTENTS

INTRODUCTION	5
PA ONLINE VOTER REGISTRATION (PAOVR) PROCESS DESCRIPTION	5
PRODUCT OBJECTIVE	7
SPECIFICATION OBJECTIVES	7
INTENDED AUDIENCES.....	8
REFERENCES.....	8
SPECIFICATION OVERVIEW	8
<i>Language</i>	9
<i>Security</i>	9
CUSTOM VRAPP PROGRAMMER INTERFACES	10
REQUIRED FIELDS	10
PRECEDENCE	10
<i>Field Processing Order</i>	10
Missing Fields	10
Invalid Fields.....	10
Mode	11
Application Data (Required)	11
Application Data (Optional)	12
AUTHENTICATION.....	14
MODES OF OPERATION.....	15
<i>Submitting Registrations Interactively Online</i>	15
<i>Submitting Registrations in Batch</i>	17
REPORTING.....	17
ADMINISTRATIVE INTERFACE	18
HOSTING A PREVIOUSLY APPROVED CUSTOM VRAPP	18
FUNCTIONS.....	18
<i>GetCalls</i>	19
GetApplicationSetUp	19
GetLanguage.....	20
GetXMLTemplate	20
GetErrorValues	21
GetMunicipalities	21
<i>Post Call</i>	22
SetApplication	22

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

RESPONSES 24

Successful..... 24

Error..... 24

RELEASE NOTES 25

SANDBOX..... 27

APPENDICES 28

 ELECTRONIC SIGNATURE SPECIFICATIONS..... 28

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

INTRODUCTION

PA ONLINE VOTER REGISTRATION (PAOVR) PROCESS DESCRIPTION

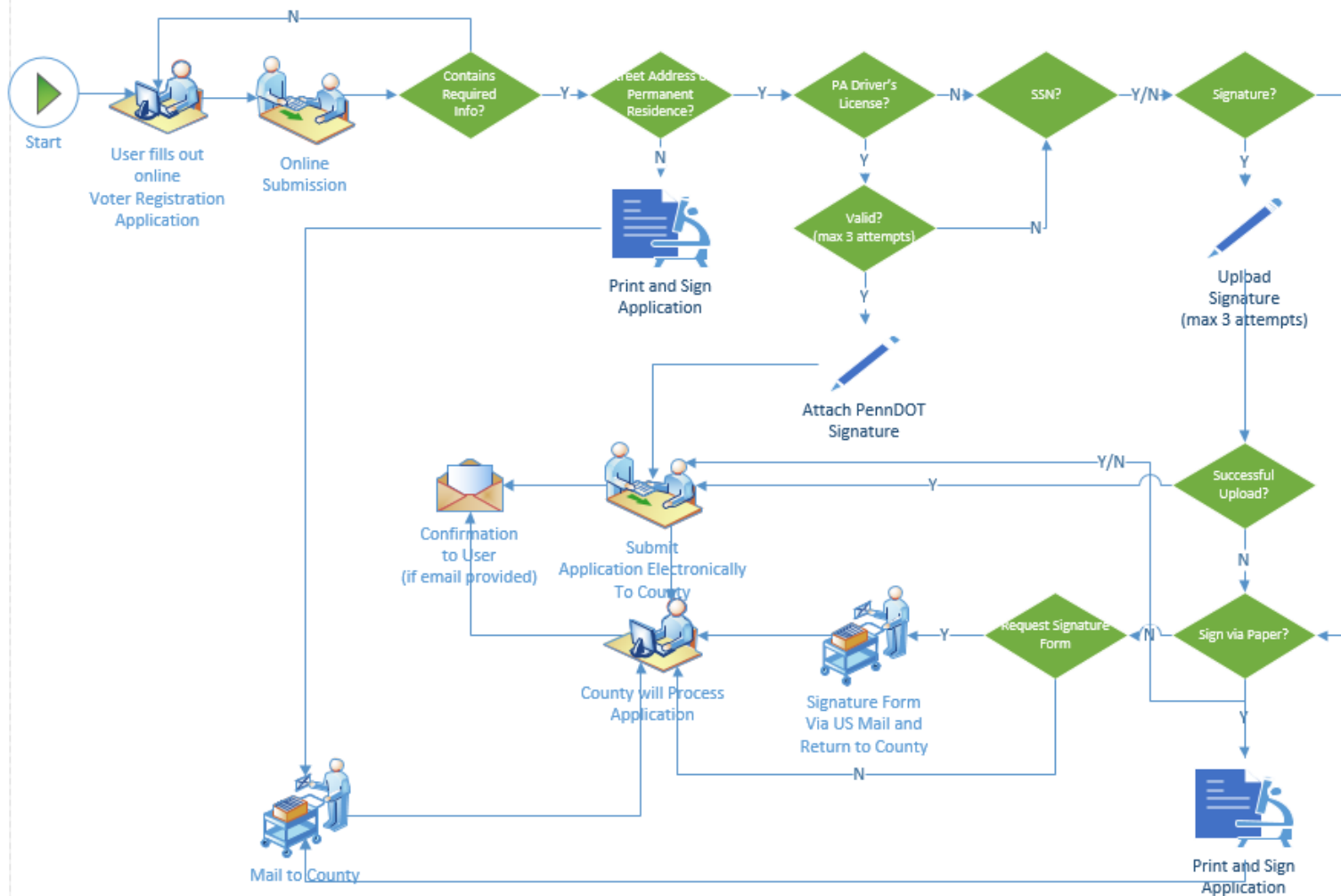
Upon launching its Online Voter Registration service in August of 2015, Pennsylvania's current Governor Tom Wolf said, "Online Voter Registration is about making the voting experience more convenient and more accessible ... It is about giving citizens an easier way to exercise their right to vote and establishing a clearer connection between the political system and the citizens. Online voter registration is secure; it improves accuracy and will reduce costs for counties by cutting down on time-consuming data entry."

Since then, Pennsylvania has seen over 900,000 citizens register to vote through this type of online service. The process to register is quite simple, and relies on certifying individuals through either their Pennsylvania Driver's License or their signature. The process is outlined below:

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

PA OnLine Voter Registration Process

April 28, 2017



PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

PRODUCT OBJECTIVE

Historically, voter registration drives have been common across the nation. These drives use various methods to make voter registration “easier” and/or more “accessible” for eligible, but unregistered voters within each state, like Pennsylvania. Third-party registration organizations are moving towards the use of mobile devices to register voters but usually only if they can capture the voter data electronically for their own purposes.

The challenge then, is how can an approved Posting Entity and its associated Partner Organization(s) (aka those organizations managing voter registration drives) easily and efficiently transfer the application information into a Statewide Uniform Registry of Electors (SURE), such as in Pennsylvania? And in doing this, increase reliability, efficiency and quality of the information captured, while presenting “real-time” acknowledgement of the voter registration application to the applicant?

These and other metrics measuring the success rate and cost reductions have prompted the Department of State to develop an easy way to do business with the Commonwealth for voter registration drives by approved Posting Entities and their associated Partner Organization(s). In addition to registering new voters, the Department’s strategy further supports the capabilities of the drives to “maintain” already registered voters. Supported voter maintenance functions currently support the following changes:

- name
- address
- party affiliation
- requiring assistance to vote
- Federal of State Employee Registering in County of Last Residence

The goal of the Department’s strategy for electronic voter registration will not only significantly reduce the overhead in voter registration and record maintenance versus paper methods, but also increase voter registrations and the quality of the voter record data.

Following the models of industry giants, such as PayPal and Amazon, the Department of State developed a secure internet facing service to which approved Posting Entities can consume this web-based service from their custom voter registration applications (custom VRApp). Thus, this service is named the Pennsylvania Online Voter Registration Web Application Interface or PA OVR Web API.

SPECIFICATION OBJECTIVES

The goal of the PA OVR Web API specification is to provide an adequate document for interested entities to develop, test, and use the PA OVR Web API service for secure voter registration applications and voter record maintenance into the SURE system within Pennsylvania.

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

INTENDED AUDIENCES

The PA OVR Web API specification document is intended for the following Posting Entity's technical audiences:

- Architecture Team
- Development Team
- Test Team

REFERENCES

This document references and/or relies on the following documents:

- Field Definition Dictionary:
 - [PA OVR Web API Field Dictionary](#)
 - [XML Guidelines](#)
 - Must be registered on PA OVR Web API website
 - Under Web API Info menu
- Application (Custom VRApp) Approval Guidelines:
 - [Registration Website](#)
 - [PA OVR Web API Terms of Use](#)
- Test Plan:
 - Acceptance Test Plan (In development) **
- Project Documents:
 - [Sample PA OVR Web API Project Timeline](#)



- [Sample PA OVR Web API Project Plan](#)

SPECIFICATION OVERVIEW

The PA OVR Web API Specification Document is a reference document that outlines information and approval processes and the corresponding interface formats for the development, approval and utilization of this online service specifically associated with Voter Registration tasks within Pennsylvania.

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

The PA OVR Web API uses HTTP methods. Requests to PA OVR Web API should be formatted in JSON and the API's responses are returned, likewise, in JSON-formatted responses.

IMPORTANT: YOU CANNOT RUN THE SAMPLE REQUESTS IN THIS SPECIFICATION AS-IS. REPLACE CALL-SPECIFIC PARAMETERS, SUCH AS TOKENS AND IDS, WITH YOUR OWN VALUES.

LANGUAGE

Currently PA OVR Web API supports both English and Spanish submissions and responses, using US-ASCII (English) characters only. If a Posting Entity would desire to support additional languages, they would have to address this with the Department of State. Architecturally, the custom VRApp would be responsible to "translate" the posting of data and responses not currently supported under the current model, if approved by the Department.

SECURITY

Current security measures employ both the environment authorization key (per Posting Entity and Partner Organization) and calls using HTTPS and SSL protocols.

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

CUSTOM VRAPP PROGRAMMER INTERFACES

REQUIRED FIELDS

Please refer to the PA OVR Web API Field Dictionary in the [REFERENCES](#) section of this document for a list of the required fields and values.

PRECEDENCE

In the design of any application that interfaces with a web service, it is important to know how the interface will expect and process the information. For this reason, we have included the order in which the PA OVR Web API will expect and process interactions with the Posting Entity's custom VRApp.

FIELD PROCESSING ORDER

PA OVR Web API has both a well-defined field processing order and a defined set of required and optional fields, denoted in the [PA OVR Web API Field Dictionary](#). It distinguishes each submission into several groups, processing each group by the following priorities:

1. Missing Fields – checking for existence of three main access fields to setup the authorized XML transfer
2. Invalid Fields – validating the values of these fields
3. Mode – existence and validation of the forthcoming XML submission mode
4. Application Data Phase
 - a. Primary Block – verifies existence and validation of primary fields (both required and optional) for most all submissions
 - b. Options Block – verifies existence and validation of options fields (both required and optional) that are used based on the prior field values

MISSING FIELDS

1. <MissingAccessKey>VR_WAPI_MissingAccessKey</MissingAccessKey>
2. <MissingAPIaction>VR_WAPI_MissingAPIaction</MissingAPIaction>
3. <MissingLanguage>VR_WAPI_MissingLanguage</MissingLanguage>

INVALID FIELDS

4. <InvalidAccessKey>VR_WAPI_InvalidAccessKey</InvalidAccessKey>
5. <InvalidAction>VR_WAPI_InvalidAction</InvalidAction>

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

MODE

6. <batch>VR_WAPI_InvalidAPIbatch</batch>

APPLICATION DATA (PRIMARY BLOCK)

7. <FirstName>VR_WAPI_MissingOVRfirstname</FirstName>
8. <LastName>VR_WAPI_MissingOVRlastname</LastName>
9. <DateOfBirth>VR_WAPI_InvalidOVRDOB</DateOfBirth>
10. <Phone>VR_WAPI_InvalidOVRphone</Phone>
11. <Email>VR_WAPI_InvalidOVRemail</Email>
12. <zipcode>VR_WAPI_InvalidOVRzipcode</zipcode>
13. <county>VR_WAPI_MissingOVRcounty</county>
14. <mailingzipcode>VR_WAPI_InvalidOVRmailingzipcode</mailingzipcode>
15. <drivers-license>VR_WAPI_InvalidOVRDLformat</drivers-license>
16. <ssn4>VR_WAPI_InvalidOVRSSNformat</ssn4>
17. <invalidsignaturestring>VR_WAPI_Invalidsignaturestring</invalidsignaturestring>
18. <invalidsignaturetype>VR_WAPI_Invalidsignaturetype</invalidsignaturetype>
19. <invalidsignaturesize>VR_WAPI_Invalidsignaturesize</invalidsignaturesize>
20. <invalidsignaturedimension>VR_WAPI_Invalidsignaturedimension</invalidsignaturedimension>
21. <invalidsignaturecontrast>VR_WAPI_Invalidsignaturecontrast</invalidsignaturecontrast>
22. <politicalparty>VR_WAPI_MissingOVRParty</politicalparty>
23. <needhelptovote></needhelptovote>
24. <typeofassistance></typeofassistance>
25. <preferredlanguage></preferredlanguage>
26. <voterregnumber></voterregnumber>
27. <previousregzip>VR_WAPI_InvalidOVRPreviouszipcode</previousregzip>
28. <previousregyear>VR_WAPI_invalidpreviousregyear</previousregyear>
29. <declaration1>VR_WAPI_MissingOVRdeclaration1</declaration1>
30. <assistedpersonname></assistedpersonname>
31. <assistedpersonAddress></assistedpersonAddress>
32. <assistedpersonphone>VR_WAPI_InvalidOVRAssistedpersonphone</assistedpersonphone>
33. <assistancedeclaration2></assistancedeclaration2>
34. <ispollworker></ispollworker>
35. <bilingualinterpreter></bilingualinterpreter>
36. <pollworkerspeaklang></pollworkerspeaklang>
37. <secondEmail>VR_WAPI_InvalidOVRsecondemail</secondEmail>

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

APPLICATION DATA (OPTIONS BLOCK)

After this, the PA OVR Web API checks for the following fields:

1. <isnewregistration></isnewregistration>
2. <name-update></name-update>
3. <address-update></address-update>
4. <ispartychange></ispartychange>
5. <isfederalvoter></isfederalvoter>
6. <united-states-citizen>VR_WAPI_MissingOVRisuscitizen</united-states-citizen>
7. <eighteen-on-election-day>VR_WAPI_MissingOVRisageover18</eighteen-on-election-day>
8. <otherpoliticalparty></otherpoliticalparty>
9. <donthavePermtOrResAddress></donthavePermtOrResAddress>
10. <streetaddress></streetaddress>
11. <streetaddress2></streetaddress2>
12. <unittype></unittype>
13. <unitnumber></unitnumber>
14. <city></city>
15. <previousreglastname></previousreglastname>
16. <previousregfirstname></previousregfirstname>
17. <previousregaddress></previousregaddress>
18. <previousregcity></previousregcity>
19. <previousregstate></previousregstate>
20. <previousregcounty></previousregcounty>
21. <politicalparty></politicalparty>
22. <assistedpersonname />
23. <assistedpersonAddress />
24. <assistedpersonphone />
25. <assistancedeclaration2 />
26. <donthavebothDLandSSN />
27. <drivers-license></drivers-license>
28. <signatureimage></signatureimage> (This depends on all the scenarios)
 - a. Same with DL too.

To help in the construct of the custom VRApp by a Posting Entity, a diagram is being presented to facilitate various technical teams in the development, testing and planning of an application focused on voter registration initiatives and integrating with the Department of State's Registry. The following UML sequence diagram shows how processes operate with one another and in what order within the context of the PA OVR Web API. It is a construct of the PA OVR Web API Message Sequence Chart that shows object interactions arranged in time sequence.

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

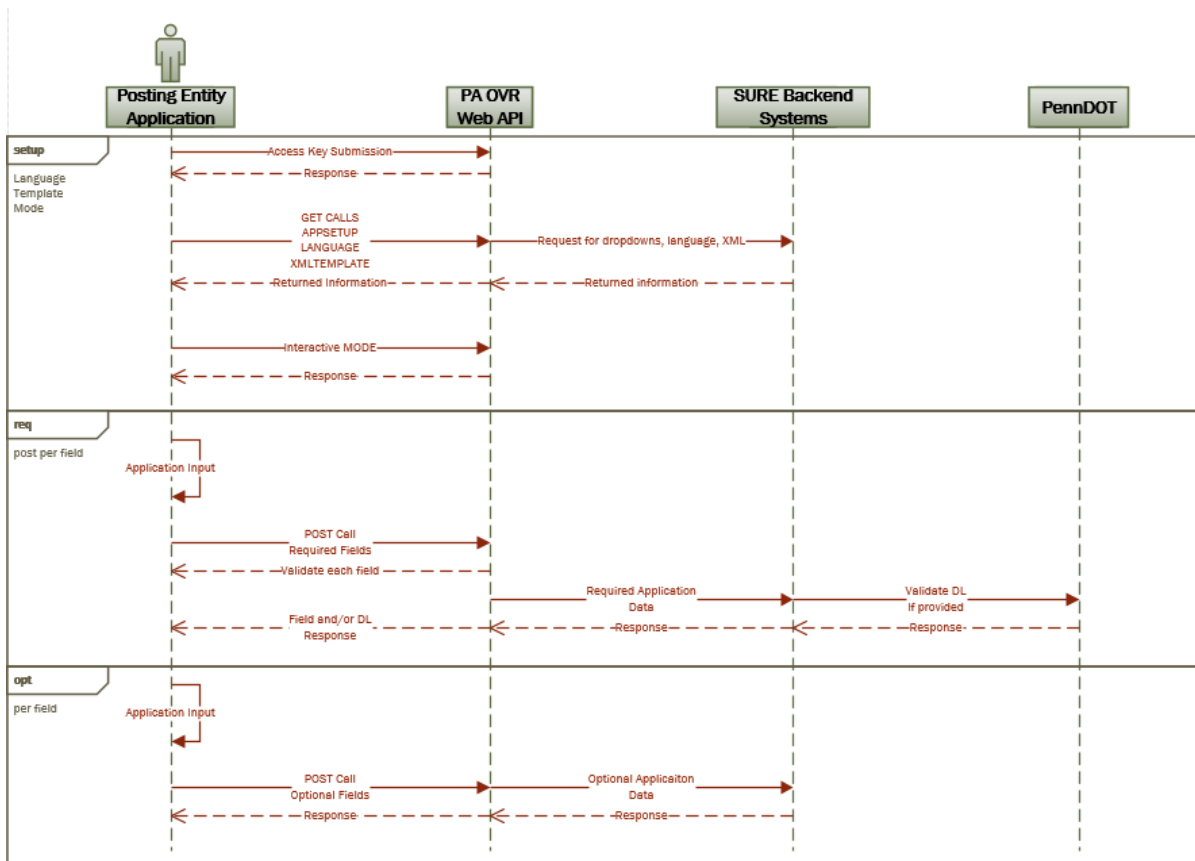


Figure 1: UML Sequence Diagram for PA OVR Web API - Interactive Mode

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

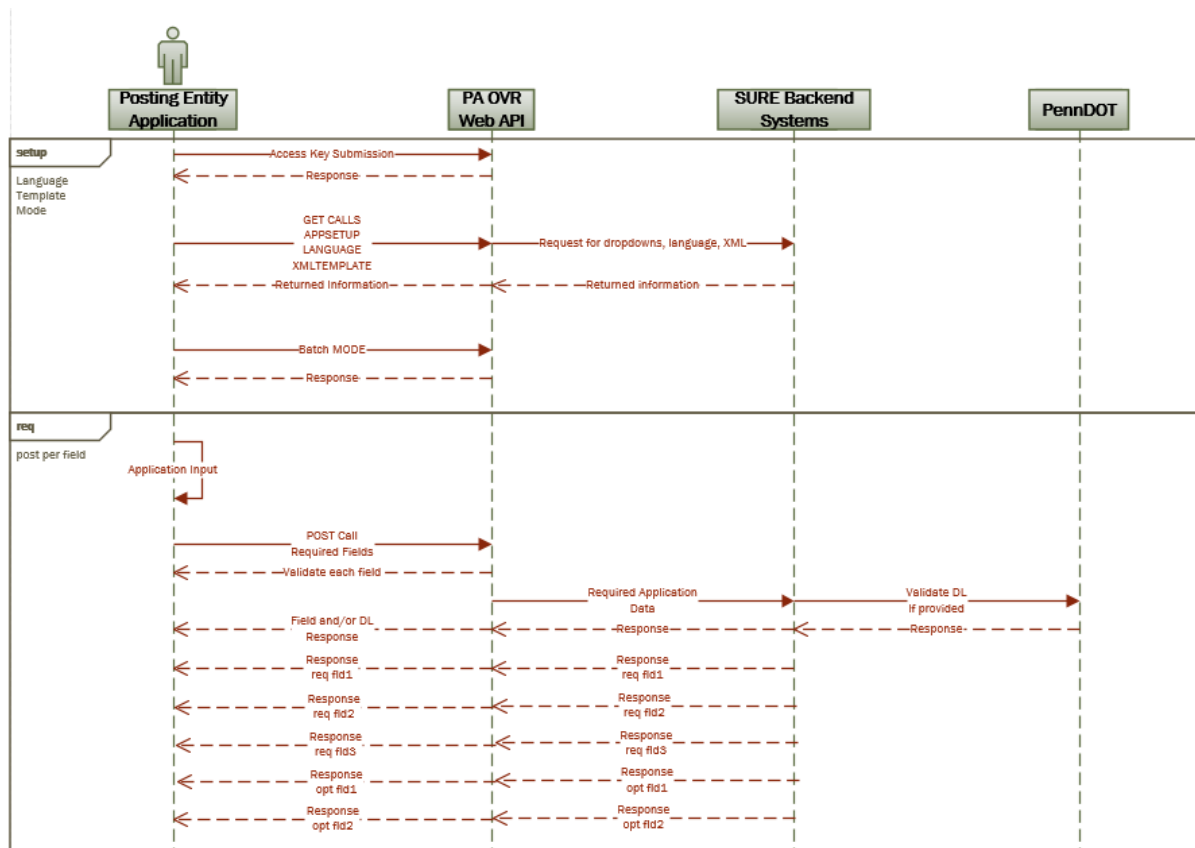


Figure 2: UML Sequence Diagram for PA OVR Web API - Batch Mode

AUTHENTICATION

To begin the process, requesting organizations will receive an authentication access key from the Department in the PA OVR Web API STAGING environment. Initial access to the STAGING environment is intended to aid in the development of the custom VRApp in the building of dropdowns with prepopulated information from the SURE system. Upon successful design and retrieval of the information, the STAGING access key will be temporarily disabled until end to end testing is approved and coordinated between the Department and the Posting Entity. All tests outlined in the test matrix (Test Plan referred in the [REFERENCES](#) section of this document) must be successful with documentation and with Department validation before an approved access key will be granted into the production environment (aka PROD). A PROD access key will be provided once the organization's custom VRApp has satisfied all testing requirements and the PA OVR Web API Terms of Use has been agreed to and completely executed via authorized signatures. The access key will be used to

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

authenticate each Posting Entity/Partner Organization for security and reporting purposes. The Department reserves the right to re-issue new PROD access keys with advanced notice. Re-issuance of access keys may be required due to security concerns.

An organization wishing to use PA OVR Web API must register with the Department of State's Bureau of Commissions, Elections and Legislation at <https://PAOVRWebAPI.votespa.com/SUREWebAPIAdmin/>

MODES OF OPERATION

To accommodate various development techniques and support associated custom VRApp s, PA OVR Web API was developed to support two modes of operation, interactive and batch. Despite the terminology, the key differentiator between interactive and batch is the utilization of the POST calls and their corresponding responses. The table below outlines the key mode differentiators:

Mode	POST Call(s)	Expected Response
Batch	Submit all fields for an application within a single POST call (XML)	All responses to each field will be received in the order submitted/processed, but all at once.
Interactive	Submit each field separately within a single POST call (XML)	Response received per POST call/XML submission.

Table 1: Modes of Operation

This design then will allow custom VRApp developers to architect their interactive application using either interactive or batch modes within the PA OVR Web API, while providing a mechanism to submit multiple applications using batch mode.

NOTE: IT SHOULD BE NOTED THAT APPLICATIONS CAN BE SUBMITTED INTERACTIVELY USING EITHER THE INTERACTIVE OR BATCH MODE WITHIN THE PA OVR WEB API.

SUBMITTING REGISTRATIONS INTERACTIVELY ONLINE

Because both the PA OVR web application (<https://www.pavoterservices.pa.gov/Pages/VoterRegistrationApplication.aspx>) and PA OVR Web API services originated from the same design lineage, there are many similarities in their features and functions in capturing and processing voter registration applications. Although both focus on new voter registration applications and certain voter records maintenance functions, the main distinction between these services is subtle, yet very important. First while both integrate tightly with the Department's backend systems and includes the validation logic for PennDOT Driver's licenses, the PA OVR web application *currently* incorporates retry thresholds, such as one retry for PennDOT DL information correction and three signature upload

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

attempts. Conversely, the PA OVR Web API allows for retry thresholds as designed by the custom VRApp. Secondly, the design of the PA OVR Web API allows for submission flexibility, yet processing information similarly to the logic built into the PA OVR web application. So much that, the logic, efficiency and flow of a registration application should follow the precedent guidelines (see section [Precedence](#)) within the Posting Entity's custom VRApp. Likewise, there is currently no inherent mechanism to identify or correct the applicant's information upon a failure, unless the Posting Entity's custom VRApp is written to support this functionality.

Applications that are submitted interactively with a valid PennDOT ID number or PA Driver's License number are to rely on the PA OVR Web API and other services to attach a digitized copy of the applicant's PennDOT signature to the voter registration application. This will be accomplished by the PA OVR Web API parsing the pertinent information (from the Posting Entity custom VRApp) and passing it to a PennDOT/HAVA Interface for validation of the PennDOT ID. Once a response is received by PennDOT (either an error or signature), the subsequent posting of application data will be committed to the SURE Portal database and an appropriate response is generated to the Posting Entity's VRApp via the PA OVR Web API per POST call.

If the applicant does not have a PennDOT Driver's License or cannot validate his/her PennDOT ID, the applicant should be presented the opportunity to submit the last four digits of their SSN and/or upload a digital image file of his/her signature. This functionality should be incorporated into the Posting Entity's custom VRApp logic. The submission of the voter registration application will follow the above description but instead of passing validation data to PennDOT, the uploaded signature will be processed by an imaging service within the SURE architecture. If the signature image file does not meet the PA OVR Web API standards or fails to properly process, the applicant may, at the discretion of the Posting Entity, have the option to print the registration form, sign it, and mail to their county voter registration office or to request a signature card be sent to their address for use in supplying a signature for their registration. This option can be built into the Posting Entity custom VRApp or relied on the SURE system to generate and mail Missing Signature Correspondence notifications (14-day rule).

Once an application is successfully submitted, the PA OVR Web API will provide a message back to the applicant that includes the application's Application ID number. If the application is successfully posted with a valid email address provided, the SURE system will notify the applicant of his/her registration via email with his/her Application ID. All further application and registration correspondence will be managed by the SURE system and its current workflow.

The following table outlines the PA OVR Web API supported flows in interactive mode:

Validation Parameter	Retries Supported?	Retries Supported?
	PA OVR Web API	PA OVR
PennDOT DL	No limit	3
Signature Upload	No limit	3

POSTING ENTITY'S CUSTOM VRAPP MUST ALSO SUPPORT RETRIES SUPPORTED BY PA OVR WEB API.

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

SUBMITTING REGISTRATIONS IN BATCH

While the goal of the Department is to certify applicants through their PennDOT ID, we do allow for applications with the last four digits of their SSN and/or digital signatures. Due to the nature of submitting multiple applications via batch, the PA OVR Web API requires multiple applications submission within a batch environment to utilize the PA OVR Web API batch mode. In batch mode, all pertinent application data is submitted within one POST call. Applications that are submitted in a batch process should allow the applicant to submit both a PennDOT ID/Driver's License number and a signature image file. Based on the precedent processing of the PA OVR Web API, it will first attempt to use the submitted PennDOT ID to retrieve a signature. If the PennDOT ID is validated, the PA OVR Web API will link the PennDOT signature to the application (and discard the submitted signature file). If the PennDOT ID cannot be validated, the system will then attempt to process the uploaded signature image file (if available) to determine if it meets signature quality standards. If neither process can capture the applicant's signature, the application will still be submitted into the SURE system, but be placed into a 14-day "hold" status, pending signature receipt. If a signature is not received after 14 days, the application will be placed into a "pending" state and the applicant will either receive a signature capture card from their county voter registration office or missing signature correspondence via the SURE system. Ultimately the applicant must supply a signature to their county voter registration office to successfully complete his or her registration.

Once an application is successfully submitted, the PA OVR Web API will provide a message back to the Posting Entity's custom VRAApp that includes an Application ID number. If a valid email address is provided, the applicant will receive an email with his/her Application ID number.

If the application is successfully posted with a valid email address provided, the SURE system will notify the applicant of his/her registration via email with his/her Application ID. All further application and registration correspondence will be managed by the SURE system and its current workflow.

NOTE: WHEN USING EITHER INTERACTIVE OR BATCH MODE WITH THE <continueAppSubmit>=1 FIELD, ALONG WITH PENNDOT DL AND/OR SIGNATURE UPLOAD, PA OVR WEB API WILL NOT RESPOND WITH A STATUS OF THE PENNDOT DL VERIFICATION SUCCESS NOR THE SIGNATURE UPLOAD STATUS.

REPORTING

Posting Entity organizations can generate a 6-month report by clicking a link in the organization's online PA OVR Web API account. This on-demand report will provide the status of all successfully submitted applications by transaction number within the 6-month reporting period. Below is an example of such a report:

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

Web API Application Status Report

	Application ID	Application Date	Status
	919958	11/03/2016	Approved
	919959	11/03/2016	Approved
	920767	11/06/2016	Approved
	920768	11/06/2016	Declined
	920769	11/06/2016	Pending
	945880	04/01/2017	Approved
	945883	04/01/2017	Declined
	945889	04/01/2017	Declined
Grand Total	8		

Figure 3: Sample Posting Entity Web API Status Report

ADMINISTRATIVE INTERFACE

Administrative account or profile functions can be performed by accessing the PA OVR Web API Admin site, located at <https://PAOVRWebAPI.votespa.com/SUREWebAPIAdmin>

HOSTING A PREVIOUSLY APPROVED CUSTOM VRAPP

Posting Entity organizations that host an approved custom VRApp from another source that has been tested previously will still need to pass all test criteria above to ensure the custom VRApp meets the Department of State's standards. This will apply to any hosting environment changes made by either the Posting Entity or its associated Partner Organization.

FUNCTIONS

Please note in each of the following calls, ***the AuthKey must be updated to your organization's authorization key.***

The general syntax for all PAOVR Web API functions will follow:

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

https://PAOVRWebAPI.beta.votespa.com/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=<XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX>&sysparm_action=<FUNCTION>&sysparm_Language=<#>

where:

<XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX> is the organization authentication key granted to the Posting Entity by the Department of State, specifically to the STAGING or PROD environment.

<FUNCTION> is one of the functional GETCALLS or POST CALL listed below

sysparm_Language=<#> is only included if listed as a functional parameter as outlined below

<#> is the functional parameter value as outlined below

GETCALLS

GETAPPLICATIONSETUP

Used to pull the drop-down values from DOS.

Parameter	Category	Value
AuthKey		<i>Your organization's auth. key</i>
Language	English	0
	Spanish	1

SYNTAX

Environment	Language	Syntax
STAGING	English	https://PAOVRWebAPI.beta.votespa.com/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=503296D9-C780-4A15-8803-1C829544258C&sysparm_action=GETAPPLICATIONSETUP&sysparm_Language=0
	Spanish	https://PAOVRWebAPI.beta.votespa.com/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=503296D9-C780-4A15-8803-1C829544258C&sysparm_action=GETAPPLICATIONSETUP&sysparm_Language=1
PROD	English	https://PAOVRWebAPI.votespa.com/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=503296D9

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

		-C780-4A15-8803-1C829544258C&sysparm_action=GETAPPLICATIONSETUP&sysparm_Language=0
	Spanish	https://PAOVRWebAPI.votespa.com/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=503296D9-C780-4A15-8803-1C829544258C&sysparm_action=GETAPPLICATIONSETUP&sysparm_Language=1

GETLANGUAGE

Used to determine the language to be displayed.

Parameter	Category	Value
AuthKey		<i>Your organization's auth. key</i>

SYNTAX

Environment	Syntax
STAGING	https://PAOVRWebAPI.beta.votespa.com/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=503296D9-C780-4A15-8803-1C829544258C&sysparm_action=GETLANGUAGES
PROD	https://PAOVRWebAPI.votespa.com/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=503296D9-C780-4A15-8803-1C829544258C&sysparm_action=GETLANGUAGES

GETXMLTEMPLATE

Used to identify the required XML tags.

Parameter	Category	Value
AuthKey		<i>Your organization's auth. key</i>

SYNTAX

Environment	Syntax
STAGING	https://PAOVRWebAPI.beta.votespa.com/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=503296D9-C780-4A15-8803-1C829544258C&sysparm_action=GETXMLTEMPLATE

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

PROD	https://PAOVRWebAPI.votespa.com/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=503296D9-C780-4A15-8803-1C829544258C&sysparm_action=GETXMLTEMPLATE
------	---

GETERRORVALUES

Used to retrieve the error code and error message pair in a JSON string.

Parameter	Category	Value
AuthKey		<i>Your organization's auth. key</i>
Language Values	English	0
	Spanish	1

SYNTAX

Environment	Language	Syntax
STAGING	English	https://PAOVRWebAPI.beta.votespa.com/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=503296D9-C780-4A15-8803-1C829544258C&sysparm_action=GETERRORVALUES&sysparm_Language=0
	Spanish	https://PAOVRWebAPI.beta.votespa.com/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=503296D9-C780-4A15-8803-1C829544258C&sysparm_action=GETERRORVALUES&sysparm_Language=1
PROD	English	https://PAOVRWebAPI.votespa.com/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=503296D9-C780-4A15-8803-1C829544258C&sysparm_action=GETERRORVALUES&sysparm_Language=0
	Spanish	https://PAOVRWebAPI.votespa.com/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=503296D9-C780-4A15-8803-1C829544258C&sysparm_action=GETERRORVALUES&sysparm_Language=1

GETMUNICIPALITIES

Used to return the municipalities in a given county. County names and corresponding Municipality Responses will be returned in English only.

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

Parameter	Category	Value
AuthKey		<i>Your organization's auth. key</i>
County		ADAMS through YORK

***NOTE: THE COUNTY NAME WILL NEED TO BE UPDATED ACCORDINGLY**

SYNTAX

Environment	Language	Syntax
STAGING	English	https://PAOVRWebAPI.beta.votespa.com/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=503296D9-C780-4A15-8803-1C829544258C&sysparm_action=GETMUNICIPALITIES&sysparm_County=ADAMS&sysparm_Language=0
PROD	English	https://PAOVRWebAPI.votespa.com/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=503296D9-C780-4A15-8803-1C829544258C&sysparm_action=GETMUNICIPALITIES&sysparm_County=ADAMS&sysparm_Language=0

POST CALL

SETAPPLICATION

Used to pass the label and value pairs in a JSON string for a registration application.

SYNTAX

Environment	Language	Syntax
STAGING	English	https://PAOVRWebAPI.beta.votespa.com/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=503296D9-C780-4A15-8803-1C829544258C&sysparm_action=SETAPPLICATION&sysparm_Language=0
	Spanish	https://PAOVRWebAPI.beta.votespa.com/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=503296D9-C780-4A15-8803-1C829544258C&sysparm_action=SETAPPLICATION&sysparm_Language=1
PROD	English	https://PAOVRWebAPI.votespa.com/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=503296D9-C780-4A15-8803-1C829544258C&sysparm_action=SETAPPLICATION&sysparm_Language=0

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

		-C780-4A15-8803-1C829544258C&sysparm_action=SETAPPLICATION&sysparm_Language=0
	Spanish	https://PAOVRWebAPI.votespa.com/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=503296D9-C780-4A15-8803-1C829544258C&sysparm_action=SETAPPLICATION&sysparm_Language=1

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

RESPONSES

Based on initial testing, response times of 30 – 45 seconds are typical for non-signature upload applications, including PennDOT DL and signature retrieval applications. For applications including signature uploads, response times have been observed to be upwards of 1 minute.

We are working on improving these response times in the next version.

The default timeout currently set at the PA OVR Web API is 20 minutes.

All PA OVR Web API responses should be formatted in XML.

SUCCESSFUL

Successfully submitted applications transmitted to the Department of State have an associated application ID (system generated). Responses for successful applications will contain the application ID in the PA OVR Web API response. Here's a sample XML response for a successful transmission:

```
<RESPONSE><APPLICATIONID>81100</APPLICATIONID><APPLICATIONDATE>Jan 23
2017 2:53PM</APPLICATIONDATE><ERROR></ERROR></RESPONSE>
```

ERROR

For attempted transmissions that result in some type of error, you'll also receive an XML response with some information pertaining to the error. They'll contain an error code, which you will also receive a translation key for what each error code means. Here are some sample responses resulting from errors:

```
<RESPONSE><APPLICATIONID></APPLICATIONID><APPLICATIONDATE></APPLICATIONDATE><ERROR>VR_WA
PI_MissingOVRfirstname</ERROR></RESPONSE>
```

```
<RESPONSE><APPLICATIONID></APPLICATIONID><APPLICATIONDATE></APPLICATIONDATE><ERROR>VR_WA
PI_InvalidOVRDOB</ERROR></RESPONSE>
```

The text highlighted in red is the error code. A translation key for the error messages that is received back from the PA OVR Web API will enable your custom VRApp logic to determine appropriate responses to the applicant.

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

RELEASE NOTES

- ❖ June 23, 2016 (v1.0)
 - Online Processing
 - Dropdown values from OVR
 - Coding and Tags to NIST Standards
 - Reporting metrics added
- ❖ August 15, 2016 (v1.1)
 - Add Forgot Username/Pwd Link to Admin website
 - System generated temporary passwords with email notification
 - Application sources added to SURE VR from Web API
 - RD
 - SE
 - SF
 - SG
 - SH
 - SI
 - SJ
 - SK
 - SL
 - SM
 - SN
 - SO
 - Email notification to applicant upon registration
 - Email notification to applicant upon approval
 - Password reset link
 - Edit Web API vendor information in SURE Portal
 - On-demand reporting by county user
 - Web API admin pages
 - Batch Processing/Submission
 - Web API applications into separate batches in SURE VR (County)
 - Status of Web API application on SURE Portal
- ❖ September 9, 2016 (V1.1.1)
 - SURE Portal update status logic
- ❖ December 14, 2016 (v1.2)
 - Add instructions and requirements to Application Screen
 - Add digital signature to agreement when requesting access
- ❖ March 10, 2017 (v1.2.1)
 - Web API application stats (daily) in SURE VR
- ❖ April 12, 2017 (v1.3)

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

- Build upgraded with jQuery version 1.12.0

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

SANDBOX

All organizations will need to submit their custom VRApp to the Department of State for testing to ensure their online user interface and underlying custom VRApp collects all necessary data elements as outlined in the attached PA OVR Web API Field Definitions document below. The PA OVR Web API will be available to users in either interactive or batch mode. custom VRApp s must be tested and approved in either or both, interactive and batch, modes.

All testing will occur against the Department of State's staging environment. We require all custom VRApp s and organizations to successfully submit testing criteria before the Department begins the testing process in our production environment.

The Department of State's PA OVR Web API currently resides in two environments, STAGING and PROD. The STAGING environment is used for a variety of purposes, including:

- User Review of new features and ongoing bug fixes
- User Acceptance Testing of new features and ongoing bug fixes
- Deployment testing
- Demos
- Posting Entity testing and approval

Meanwhile, the PROD or production environment is for operational functions of LIVE data involving Voter Information and its associated systems.

PA OVR Web API Partner	Document ID: PAOVRSpec	Version: 1.0
PA OVR Web API Specification Document		Version Date: 05/23/2017

APPENDICES

ELECTRONIC SIGNATURE SPECIFICATIONS

The following specifications should be considered when capturing and submitting signatures as images. These specifications are meant to increase the performance of the PA OVR Web API and acceptance of digital signatures into the SURE system by enhancing their quality and other parameters:

- Signature capture and manipulation are the responsibility of the Posting Entity's custom VRAApp.
- Signatures should be formatted as TIF, TIFF, JPG, BMP or PNG
- Signatures can only be 80 high and 160 width (pixels)
- Signatures should be at least 96 dpi
- Signatures should not be rotated
- Signatures should be captured in black and white only (no grey tones or color)
- Datatype transmitted should be Base64