System Interface Diagrams and Specifications

California Department of Justice

AB 953: Racial and Identity Profiling Act of 2015

VERSION HISTORY

Version #	Date	Author	Key Differences
1.0	03/03/2017	Ray Birchler	Assembled document sections to provide deliverable template.
1.1	03/07/2017	Ray Birchler	Updated template based on DOJ DED review comment from Charles.
1.1a	05/19/2017	Ray Birchler	Assembled deliverable document.
1.1b	6/1/2017	Ray Birchler	Updated based on feedback from Lee and to adjust log record layouts to align with data model.
1.1c	06/07/2017	Ray Birchler	Updates based on comments from Technical and Infrastructure Team Members.
1.2	7/14/2017	Ray Birchler	Updated to provide alignment to version 1.2 of Data Model and Technical Data Dictionary.
1.3	8/4/2017	Ray Birchler	Updated interface spec to align with 6/19/2017 version of the DRAFT regulations.
1.3a	8/14/2017	Ray Birchler	Updated interface specifications to align with 7/25/2017 version of Draft Regulations.
1.4	8/28/2017	Ray Birchler	Updated to align with 8/1/2017 DRAFT regulation.
1.4b	11/20/2017	Ray Birchler	Applying cleanup changes to document in preparation for outreach meetings with LEA community.
1.4c	11/29/2017	Ray Birchler	Updated based on Peer Review
1.4c1	12/5/2017	Ray Birchler	Updated document to only contain data relevant to LEA outreach session.
1.4d	12/6/2017	Ray Birchler	Updated to replace Attachment C with Technical Data Dictionary

TERMS AND ACRONYMS

Terms	Definition
XML	eXtensible markup language
SSL	Secure Socket Layer
UML	Unified Modeling Language
REST	REpresentational State Transfer
JSON	JavaScript Object Notation
SFTP	Secure File Transport Protocol
SDCS	Stop Data Collection System

TABLE OF CONTENTS

TABLE OF TABLES

TABLE OF FIGURES

Introduction

The California Department of Justice (DOJ) is implementing the Racial and Identify Profiling Act of 2015 (AB 953). AB 953 requires all city and county local law enforcement agencies (LEAs), as well as the California Highway Patrol and peace officers of California state and university educational institutions, to collect perceived demographic and other detailed data on stops. Probation officers and officers in a custodial setting are excluded from this collection requirement. Once collected, LEAs are required to submit stop data to the DOJ. Additional information on AB 953 and the regulations is available at: https://oag.ca.gov/ab953.

Scope

This documentation details the Data Exchange Architecture and the System Interface Diagrams and Specifications that will be leveraged in the implementation of DOJ's Stop Data Collection System's (SDCS') internal and external interfaces. The document will approach the diagrams and documentation through the lens of services that are required to support the planned interfaces.

Data Exchange Architecture

The Data Exchange Architecture is a sub-component of the Application Architecture. However, for the SDCS implementation the Data Exchange Architecture is described as a separate architecture to more clearly articulate the dynamics of the architecture. This architecture will define the methods and services that will be utilized to integrate LEA data collection with the DOJ. Integration can either be through the RESTful web services or through the Secure File Transfer services. The following sections will describe how the architecture supports each of these interface methodologies.

Real-Time Data Exchanges

For real-time data exchanges, the standard data exchange protocol will be RESTful Web Services. The architecture regarding this data exchange type are described in the Web Service Interface Architecture and Data Validation sections below.

Batch Data Exchanges

For batch data exchanges, the standard data exchange protocol is secure file transfer protocol (SFTP) transmissions. The architecture with this data exchange type is described in the Batch External Interface Architecture section below.

This architecture is based on the System Interactions diagram from the System Architecture Diagrams documentation, as shown in Figure 1 below. The SDCS solution will have three (3) access channels. These access channels are shown in the blue boxes in the figure. Each of these will leverage the data exchange architectures differently. They are:

1. DOJ Hosted Web Application
2. DOJ Secure File Transfer Services
3. DOJ Hosted Web Services

The following is a brief description of how these three (3) access channels will interact with the Data Exchange Architecture.

1. The **DOJ Hosted Web Application** will be the access channel that is used for LEAs that chose not to develop their own data collection application. This application will NOT need to interact with the Central Data

Validation services to perform a variety of validations on the SDCS stop data prior to the data being placed into a “Successful Submission” status.

2. The **DOJ Secure File Transfer Services** will encompass a DOJ hosted SFTP service that LEAs can utilize when they have their own data collection application but chose not to leverage the DOJ hosted web services. This SFTP service will interact with DOJ Central Data Validation service for the Processing of the bulk stop data records that are uploaded to the FTP server.
3. The **DOJ Hosted Web Service** will allow LEA that have their own data collection application to send the resulting stop data to the DOJ in real-time through RESTful web service. The web service will interact with the Central Data Validation service to perform a variety of validations on the SDCS stop data prior to the data being placed into a “successful submission ” status.

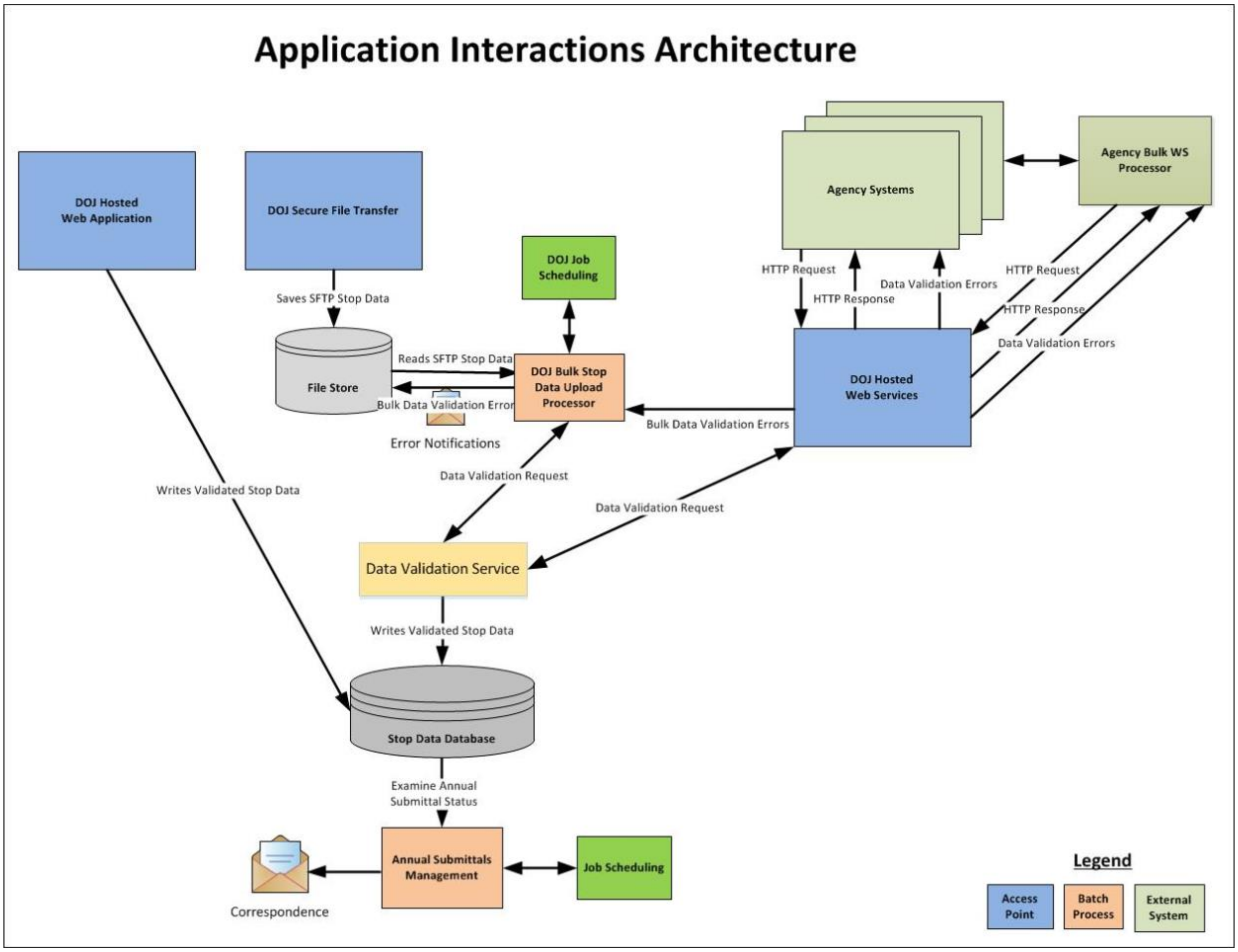


Figure 1 – Logical System Interactions

Reference Architecture

The following Figure 2 shows the core logical components of the Data Exchange Architecture.

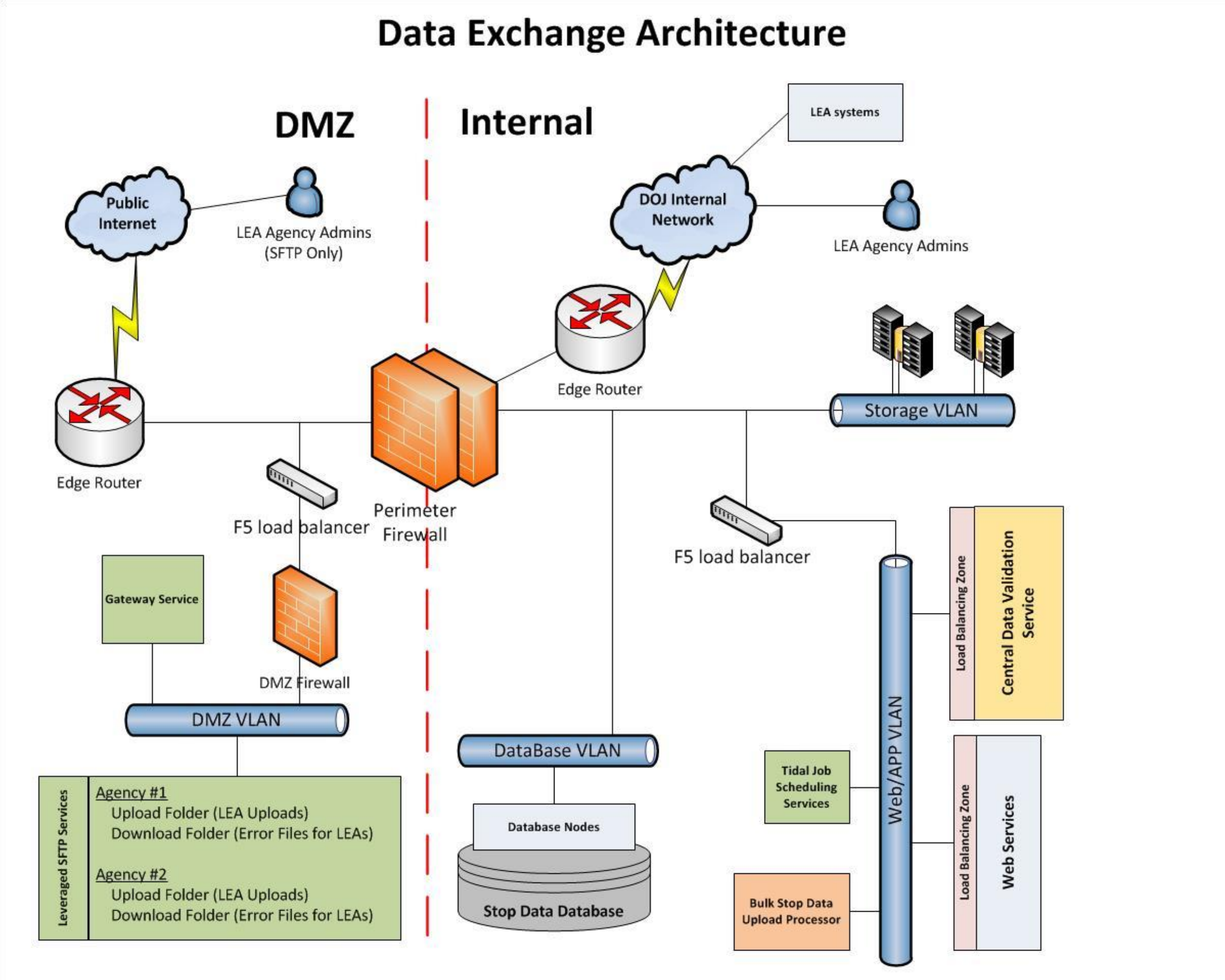


Figure 2 – Data Exchange Architecture

Supported Communication Protocol

In support of the Data Exchange Architecture, the SDCS includes services which are shown in the previous figures. The SDCS services are architected to support the synchronous communication protocol. Meaning, the SDCS services will complete the processing of a request from an agency before it accepts another request from the same agency. The architecture will accept requests from multiple agencies concurrently however, the processing with any single agency will follow the synchronous protocol.

The following figures demonstrate how this concept is realized in all of the SDCS access channels which are described in the Data Exchange Architecture.

Web Services

The following figure applies to the real-time Web Services access channel. As shown in Figure 3, each web service call from the agency is contained within a single communication thread between the agency system and the DOJ web service. This allows all processing of a transaction to complete before the next transaction is initiated. Meaning, all data validations and database activities are completed as a single unit of work. This facilitates better data consistency and error processing.

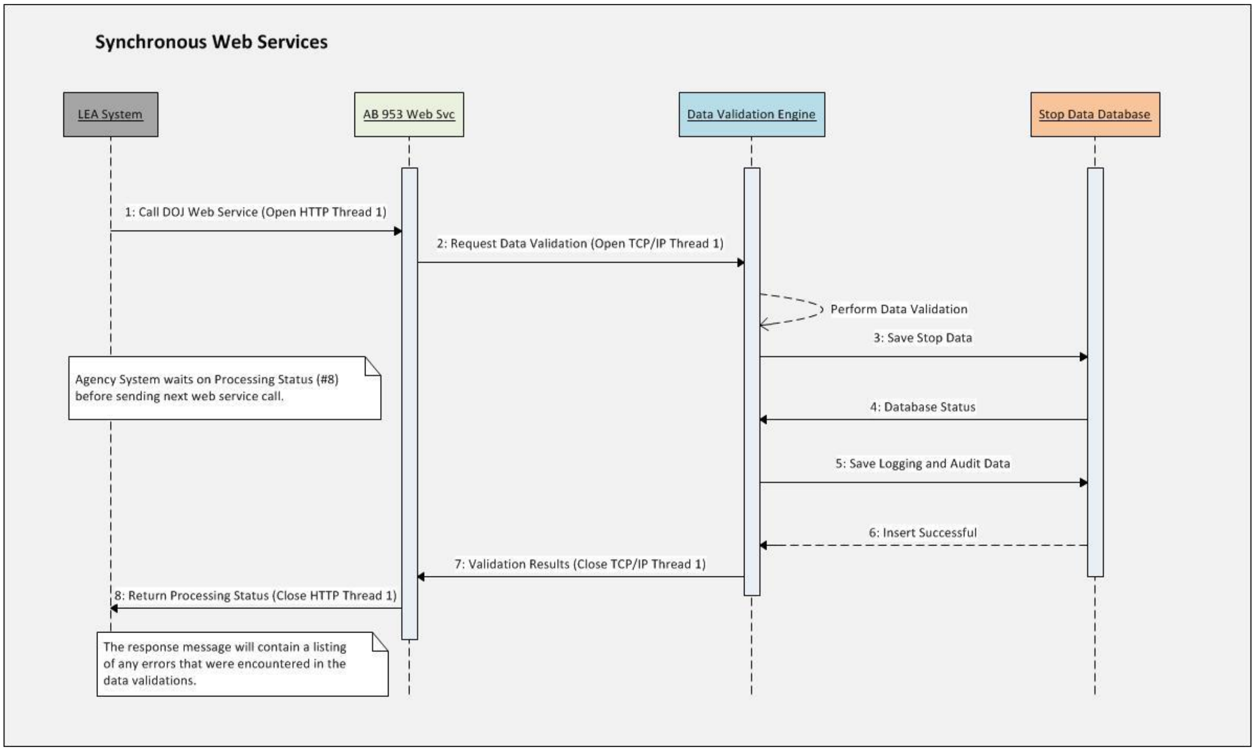
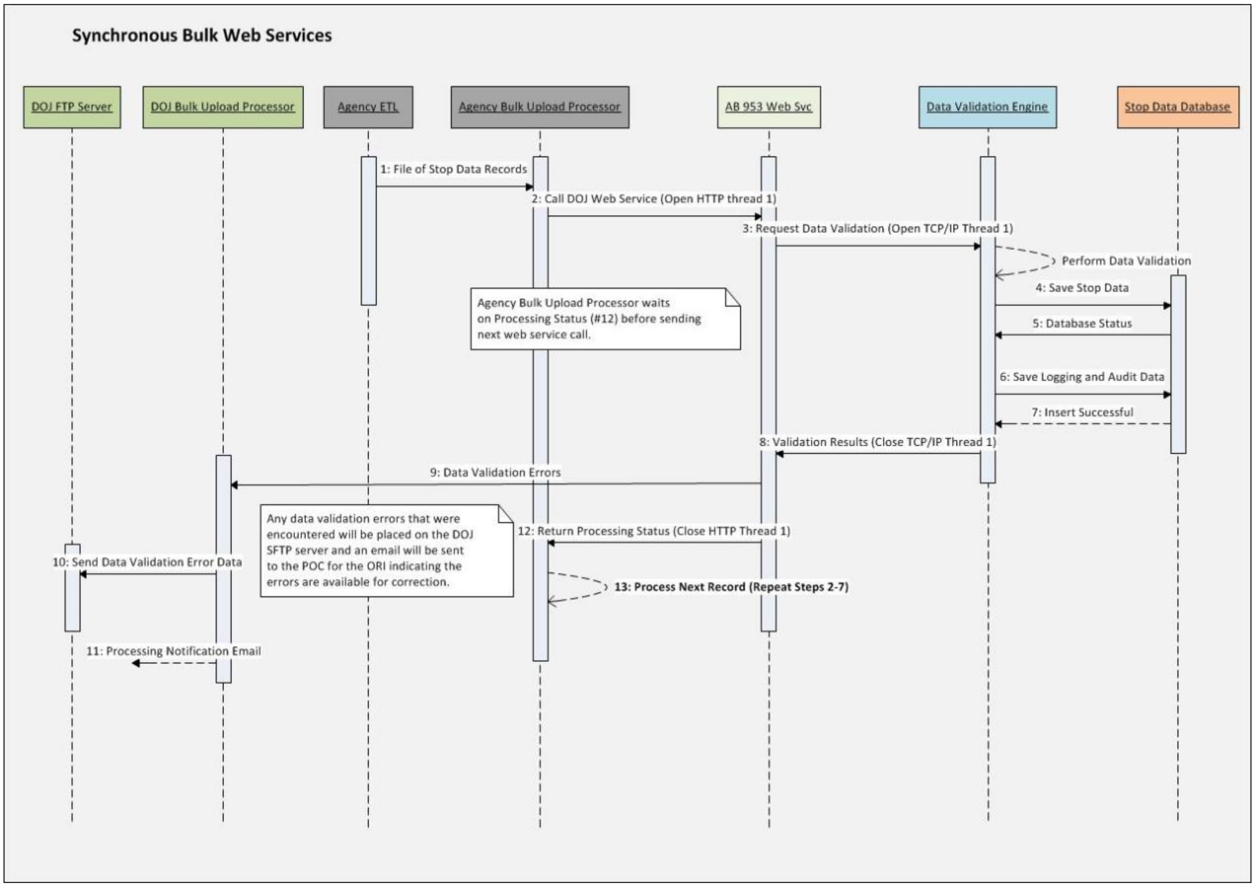

Synchronous Web Services
LEA System
AB 953 Web Svc
Data Validation Engine
Stop Data Database
1: Call DOJ Web Service (Open HTTP Thread 1)
2: Request Data Validation (Open TCP/IP Thread 1)
Perform Data Validation
3: Save Stop Data
Agency System waits on Processing Status (#8) before sending next web service call.
4: Database Status
5: Save Logging and Audit Data
6: Insert Successful
7: Validation Results (Close TCP/IP Thread 1)
8: Return Processing Status (Close HTTP Thread 1)
The response message will contain a listing of any errors that were encountered in the data validations.

Figure 3 – Web Service Synchronous Architecture

Bulk Web Services

The next figure applies to the Bulk Web Services access channel. Figure 4 is very similar to the real-time scenario above, with the exception that the agency is interacting with the DOJ web service from a bulk processing module instead of from a real-time application. In this scenario, because the calling entity is bulk processor, it is not reliable to only return the data validation errors to the calling entity. In addressing this issue, the bulk processing scenario will place a copy of the data validation errors on the DOJ FTP server and send an email to the agency point-of-contact (POC) to notify that errors were encountered in the Stop Data Record submissions. As with the real-time scenario, the communication thread with the agency remains open until all data validation, database processing, and data validation errors management activities are completed.


Synchronous Bulk Web Services
DOJ FTP Server
DOJ Bulk Upload Processor
Agency ETL
Agency Bulk Upload Processor
AB 953 Web Svc
Data Validation Engine
Stop Data Database
1: File of Stop Data Records
2: Call DOJ Web Service (Open HTTP thread 1)
3: Request Data Validation (Open TCP/IP Thread 1)
Perform Data Validation
4: Save Stop Data
5: Database Status
6: Save Logging and Audit Data
7: Insert Successful
8: Validation Results (Close TCP/IP Thread 1)
9: Data Validation Errors
10: Send Data Validation Error Data
11: Processing Notification Email
12: Return Processing Status (Close HTTP Thread 1)
13: Process Next Record (Repeat Steps 2-7)

Agency Bulk Upload Processor waits on Processing Status (#12) before sending next web service call.

Any data validation errors that were encountered will be placed on the DOJ SFTP server and an email will be sent to the POC for the ORI indicating the errors are available for correction.

Figure 4 – Bulk Web Service Synchronous Architecture

Secure File Transfer

The next figure applies to the SFTP access channel. Figure 5 is very similar to the Bulk Processing scenario above with the exception that the agency is interacting with the DOJ FTP service instead of interacting with the web services directly. In this scenario, because the calling entity is the SFTP service, it is not reliable to return the data validation errors to the calling entity. In addressing this issue, the DOJ bulk processing module will place a copy of the data validation errors on the DOJ FTP server and send an email to the agency POC to notify that errors were encountered. As with the bulk web service processing scenario, the communication thread with the data validation engine remains open until all data validation, database processing, and data validation errors management activities are completed.

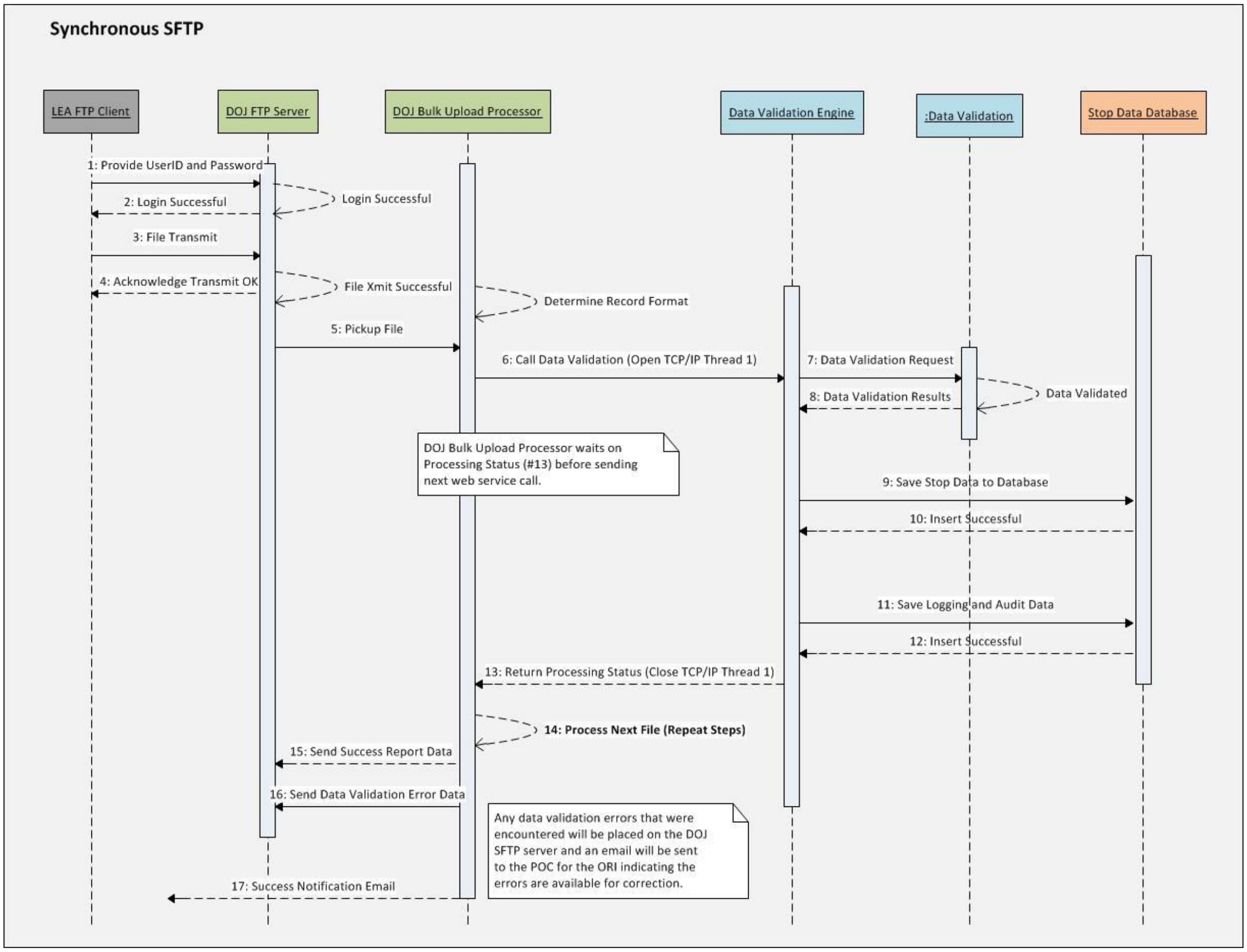

Synchronous SFTP
LEA FTP Client
DOJ FTP Server
DOJ Bulk Upload Processor
Data Validation Engine
:Data Validation
Stop Data Database
1: Provide UserID and Password
Login Successful
2: Login Successful
3: File Transmit
4: Acknowledge Transmit OK
File Xmit Successful
Determine Record Format
5: Pickup File
6: Call Data Validation (Open TCP/IP Thread 1)
7: Data Validation Request
Data Validated
8: Data Validation Results
DOJ Bulk Upload Processor waits on Processing Status (#13) before sending next web service call.
9: Save Stop Data to Database
10: Insert Successful
11: Save Logging and Audit Data
12: Insert Successful
13: Return Processing Status (Close TCP/IP Thread 1)
14: Process Next File (Repeat Steps)
15: Send Success Report Data
16: Send Data Validation Error Data
Any data validation errors that were encountered will be placed on the DOJ SFTP server and an email will be sent to the POC for the ORI indicating the errors are available for correction.
17: Success Notification Email

Figure 5 – FTP Synchronous Architecture

Central Data Validation Service

As depicted in Figures 1 and 2 the architecture will also provide for a Central Data Validation Service. This service will be shared by the DOJ Bulk Processing Application and the DOJ Hosted Web Service. The purpose of the central validation engine will be to provide a single method for validating the collected stop data in a consistent manner. No matter which of these access channels is leveraged, the stop data will all be validated using a single consistent methodology.

During the detailed design of the SDCS, it was determined this central service will be designed and exposed as a stand-alone Java Object that is called for data validations.

The SDCS Central Data Validation service will be a service that complies with the architectures that are documented for the SDCS. It will apply the required data validations and/or relational validations to the stop data and return the record along with the error messages to the calling application or service. It will create any logging or auditing records and will save stop data to the database.

Web Service Architecture

As shown in the figure above, the web service architecture will contain the DOJ Hosted Web Service. The Hosted Web Service will be leveraged by LEAs for real-time or bulk submissions of the SDCS Stop data.

DOJ Hosted Web Service

The **DOJ Hosted Web Service** will allow LEAs that have their own data collection application to send the resulting stop data to the DOJ in real-time through a RESTful web service. The web service will validate and store the stop data in the DOJ stop data database. Any errors or validation failures that are encountered by the web service will be returned to the LEA in real-time for resolution after the stop data has been saved in the database in a status to indicate that the Validation Failed. This access channel will leverage the Central Data Validation service to validate the stop data.

The DOJ Hosted Web service access channel will also be capable of supporting bulk web service calls from the agency systems. In this type of interaction, the LEA will send web services transactions to the web services engine in batch or bulk mode. This interaction from the DOJ system perspective will be a one stop record to one web service call model while it will be a bulk process from the LEA application perspective. Meaning, while the LEA will extract a batch of stop data records from their application they will need to make one call to the DOJ web service for each stop data record that was extracted.

In this scenario, there will not be an end-user to receive the data validation errors so the web services will leverage a DOJ batch process to construct and distribute the error messages to the DOJ SFTP server.

Web Service Architecture Constructs

Figure 6 represents the interface architecture for the real-time web service interactions between the SDCS and the LEAs. These architectures will apply to all of the categories of web services that are listed above.

These interfaces will be architected using RESTful web services. The architecture supports the standard interactions of web services architectures. The following figure demonstrates the interactions that the web services architecture will support.

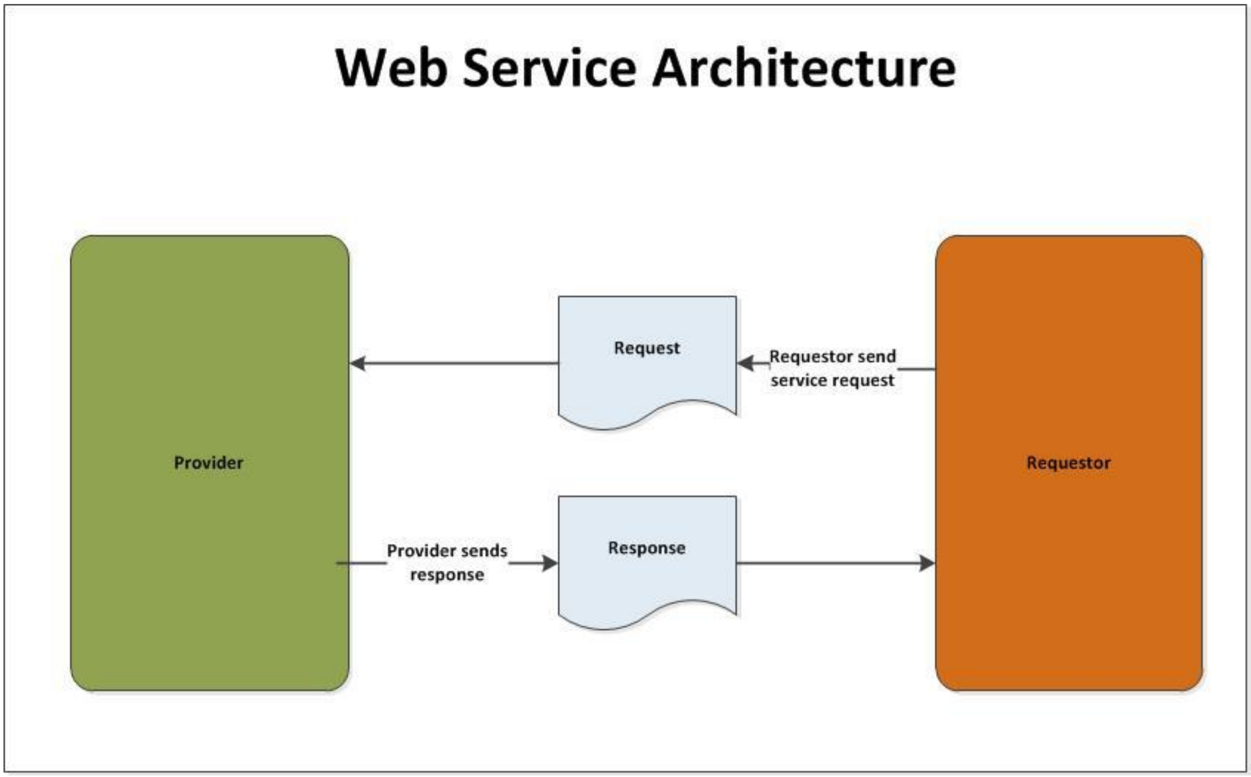


Figure 6 – Web Service Architecture

As shown this integration will use the standard interactions where (1) The requestor will send service requests to the provider and (2) the provider will respond to these requests with a response message.

The SDCS application will leverage RESTful web services as this is an industry best-practice architecture. The intent of this section is not to describe the services at the detailed design level, but to describe them from an architectural perspective. The more granular details regarding the request and reply contents for each web service can be found in the Interface Diagrams and Specifications Deliverable for SDCS.

Figure 7 below documents the architecture of the HTTP Request and HTTP Reply messages. The architecture leverages the constrains as defined in the REST architecture.

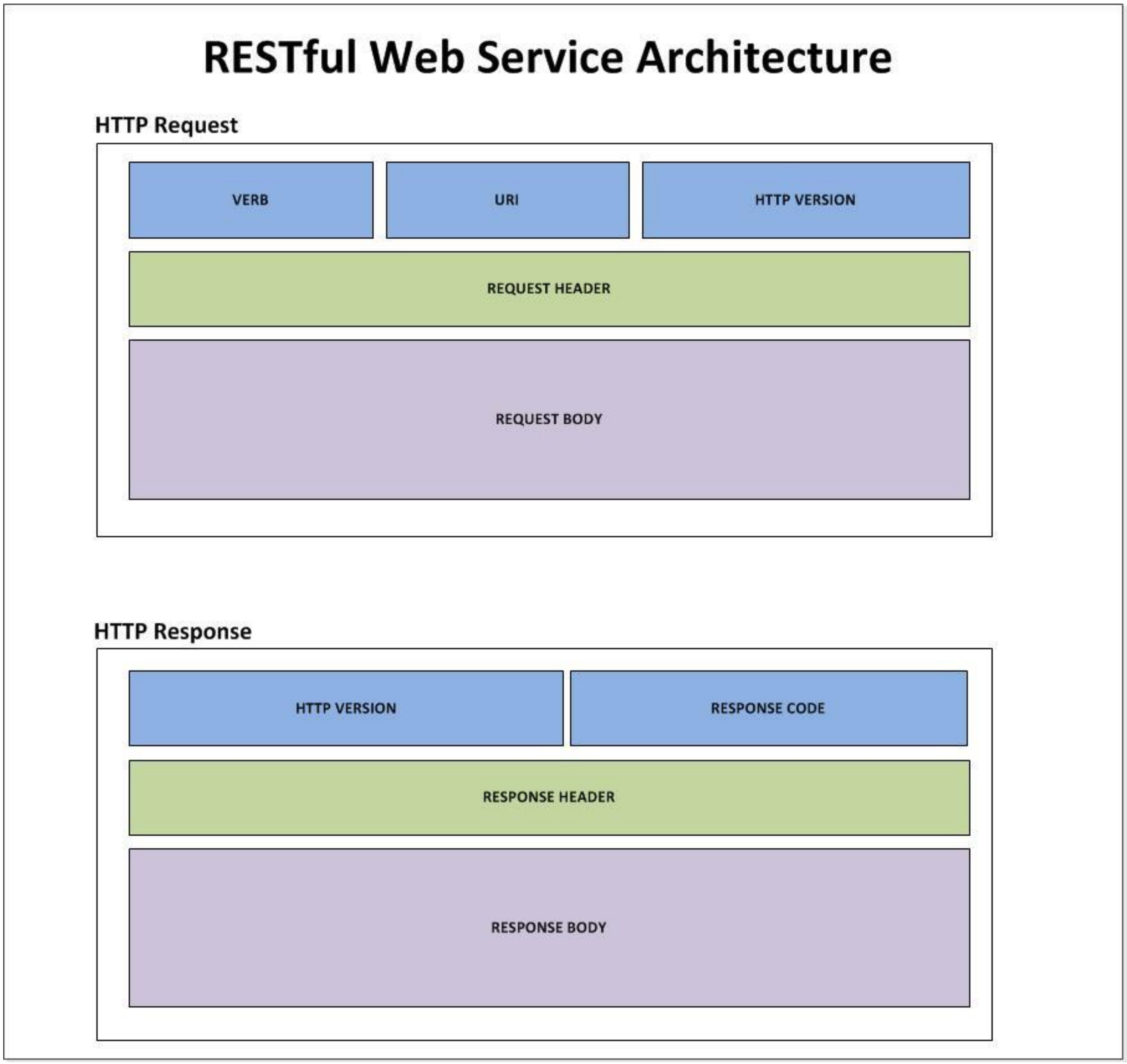


Figure 7 – RESTFul Service Architecture

Tables 1 and 2 below, contain a description of the components of the HTTP Request and HTTP Response messages that the architecture will support.

HTTP Request

Table 1 – Web Architecture HTTP Request Data Elements

Component	Description
Verb	Indicates the HTTP methods such as GET, POST, DELETE, PUT, etc.
URI	Uniform Resource Identifier (URI) to identify the resource on the server.
HTTP Version	Indicates the HTTP version. For example, HTTP v1.1.

Component	Description
Request Header	Contains metadata for the HTTP Request message as key-value pairs. For example, client (or browser) type, format supported by the client, format of the message body, cache settings, etc.
Request Body	Message content or Resource representation. This component will contain the detailed SDCS Stop Data in either XML or JSON format.

HTTP Response

Table 2 – Web Architecture HTTP Response Data Elements

Component	Description
Response Code	Indicates the Server status for the requested resource. For example, 404 means resource not found and 200 means response is ok.
HTTP Version	Indicates the HTTP version. For example, HTTP v1.1.
Response Header	Contains metadata for the HTTP Response message as key-value pairs. For example, content length, content type, response date, server type, etc.
Response Body	Response message content or Resource representation. This component may contain the detailed SDCS Stop Data.

Batch Interface Architecture

The DOJ SFTP service will allow the LEA to upload bulk stop data submissions to the DOJ. Once these submissions have been completed and the file transmission is verified the stop data records will be passed to the DOJ bulk stop data upload processor module for processing, validation and storage in the stop data database. Any file transmission or stop data validation errors that are encountered will be sent to the LEA for error correction.

The expectation is that the SDCS SFTP services will leverage the existing DOJ SFTP services. The SDCS project can provide additional compute resources to this leveraged services if required.

As shown in Figures 1 and 2 above the SFTP server will provide secure file transfer services for the LEA Community. The SFTP service will also provide two landing zones for each LEA that subscribes to the services. The first will be an upload or INPUT folder and the second will be a download or OUTPUT folder.

The following is a description of each of these folders:

INPUT folder – this folder will be used by the LEAs who subscribe to the SFTP service to upload text based files that contains the Stop Collection Data records. There will be one record in the file for each stop data collection. The DOJ batch service that is noted in the figure above will poll this folder to determine when a new file submission has been received from the LEAs.

OUTPUT folder – this folder will be used by the batch service to log errors that were encountered while processing the stop data records. This folder will also be used by the LEAs to pickup these error files for error corrections.

Bulk Stop Data Upload Processor

There will be batch program and job scheduling functions deployed to route the SFTP data submissions to the DOJ Hosted Web Service for processing. These same batch services will be leveraged to pass the validation error data from the web service processing back to the SFTP server folders. An email notification is then generated to inform the LEA that some of the submitted stop data records received processing errors.

The following will be the typical processing flow for the SFTP file transmissions:

1. The LEA will upload a file onto the DOJ Hosted SFTP Server.
2. The DOJ Job Scheduler will discover that a new file has been placed on the SFTP server.
3. The DOJ Job Scheduler will leverage scripts to copy the data file to the folder that is utilized by the Bulk Stop Data Upload Processor module.
4. The DOJ job scheduler will run scripts to archive the data file as dictated in DOJ policy.
5. The Bulk Stop Data Upload Processor will process the records from the SFTP files one at a time and send the record to the DOJ hosted central validation engine for validation and processing.
6. The DOJ central validation engine will return to the Bulk Stop Data Upload Processor the status of each record that has been processed.
7. Records shall be validated and saved to the stop data database by the central validation engine in either a “Submission Successful” or “Submitted with errors” status depending on if any validation errors were encountered.
8. Any records that do not pass all of the data validations will be returned to the Bulk Stop Data Upload Processor along with the error messages.
9. After all records in the file(s) have been processed the Bulk Stop Data Upload Processor will consolidate any error records into a single file.
10. This file of the error records is then placed into the agency’s OUTPUT folder on the SFTP server.
11. An email message will be sent to the LEA POC, at the determined frequency, indicating that there were errors in the file(s) along with instructions on how to access and download the error files and messages.

System Interface Diagrams and Specifications

The following are the detailed interface specification for the real-time, bulk, and batch interfaces that the SDCS will provide. As these interfaces will share a common data structures attachments have been included that fully documents the data related to the SDCS Stop Data.

These data structures are included in **Attachment A - AB953 Interface XML** and **Attachment B - AB953 Interface JSON**. For LEAs that chose to use the XML or JSON file format it will be key that the uploaded records precisely follow the format in these attachments. Any records that do not follow the prescribed format will be rejected by the DOJ. The Stop Data Collection System will also have the ability to accept stop data record submission through the SFTP service in Excel or delimited text formats. The format of these text file formats is provided in **Attachment C – Excel-CSV Bulk Upload Format**. For LEAs that chose to use this file format it will be key that the uploaded records precisely follow this Excel or CSV format. Any records that do not follow the prescribed format will be rejected by the DOJ.

1. Central Data Validation Service

Operational Description

The SDCS will include a Central Data Validation service as described in the Data Exchange Architecture. The validation service will be utilized by both the Bulk Processing Application and Web Services to validate the SDCS stop data when the data is saved and submitted to the DOJ. For the SFTP services and the web service access channels, the data will be considered as submitted when it is transmitted to the DOJ by the LEAs. The Applications will initiate a call to the validation engine and it will return any errors that are encountered in the data to the calling entity. These errors will be data specific errors and as such will not be listed in the Error Handling section of the service descriptions below. These error codes will be defined and documented in the design specifications for the data validation service.

The data validation service will be built to accept the data from the XML data stream as defined in **Attachment A - AB953 Interface XML** or the JSON data stream as defined in **Attachment B - AB953 Interface JSON**. The service will apply the validation edits to the data elements as defined in the **Technical Data Dictionary.** Please refer to this attachment for the details regarding the data validation edits that will be applied. Once all of the data validations have been applied, a listing of the encountered errors codes and attribute names from the data stream will be returned to the calling entity.

The Central Data Validation service will be capable of performing data validations or relational data validations or both. This will be determined by the method that is used in the service execution message.

The data validation service will need to perform several functions to process the stop data. The following is a high-level list of the business functions the client-side service will need to perform.

1. Receive the call from the service requestor.
2. The Common Validation Service will then process the stop data and prepare a response message with any error codes for invalid or missing data. The service will use the rule in **Technical Data Dictionary** to validate the stop data record.
3. Prepare a response message and send it to the service requestor indicating the status of the data validations service execution.

Error Handling

Duplicate Record Checking

The data validation service will include duplicate record check as a FATAL error and will be conducted through a multi-level process to ensure the integrity of the SDCS stop database when the requested operation is CREATE or INSERT.

The following is the duplicate record checking logic that will be used when the requested operation is **INSERT or CREATE**: The first step will be to execute a query against the database using the Agency ORI and LEA Record ID as the search parameters. If the search finds a matching record in the database, a **FATAL** error will be generated and the new stop record will not be inserted into the database. If the search finds zero records with the ORI and LEA record ID combination, then a secondary check will be performed. The second query will use the Agency ORI, Officer UID, Stop Date, and Stop Time as search criteria. If this query finds one or more records in the database, then the new record will not be inserted into the database and a Duplicate Record **FATAL** error message will be returned to the LEA. It will then be up to the LEA to determine which of the records is valid and if needed, contact the DOJ for assistance to delete the duplicate record. Meaning, the duplicate records errors can not be corrected through the web service access channel when the secondary check fails.

Record Matching

The following is the record matching logic that will be used when the requested operation is **UPDATE**: execute a query against the database using the Agency ORI and LEA Record ID as the search parameters. If this search finds one records, then the record can be updated after the record status is checked of the existing record in the SDCS database. If the record cannot be found, a **FATAL** error will be generated and the record update will not be applied to the database. If the record exists and the status is any status other than "Submitted with Errors/incomplete" the update record will be rejected with a **FATAL** error as any records in a NFIA or Submitted to DOJ status are considered as un-editable.

Fatal Errors

The SDCS data validation service will generate several errors that are classified as fatal errors. The definition of a fatal error in the context of the SDCS web services are any errors that are encountered where the stop data record can not be saved to the database or the stop data record cannot be saved to the database without compromising the integrity of the database. The following table 3 lists the preliminary list of the fatal errors that can be generated.

Table 3 – Data Validation Fatal Errors

Name	Description
Invalid ORI	This error is generated when the ORI from the service call can not be validated against the ORI table in the database.
Invalid Core Data Attributes	This error is generated when the LEA Record ID, Officer UID, Stop Date, and Stop Time attributes are invalid on the transmitted stop data record from the service message.
Duplicate Stop Data Record	This error will be generated when the Insert/Create transaction determines the record is duplicate after all of the duplicate record identification logic has been applied or the record
Record Matching	This error will be generated when the Update transaction cannot identify the record to be updated in the database.
Update for Submitted Record	This error will be generated when a transaction is received for a stop data record that is in a "Successful Submission" or "NFIA" status in the database.

2. Submit Stop Data Web Service

Operational Description

The SDCS DOJ Hosted Web Service will allow LEAs that have their own data collection application to send the resulting stop data to the DOJ in real-time through a RESTful web service. The web service will process, validate and store the stop data in the SDCS. Any errors or validation failures that are encountered by the central validation service will be returned to the LEA in real-time for resolution. This service will leverage the Central Data Validation service to validate the stop data prior to the web service saving to the stop data database in a Successful Submission or Submitted with Errors status.

The format and contents of the error records can be found in the SDCS Data Model and the SDCS technical Data Dictionary. As such, the step in the sequence diagrams below titled Generate log and audit records will be generating all three log record types as appropriate to the specific scenario that is being discussed.

The following is a brief description of the processing flow for both the client side web service (LEA) and the server side web service (DOJ).

Client-side Service (LEA)

The client-side web service will need to perform several functions to prepare the HTTP Request message and process the HTTP Response Message. The following is a high-level list of the business functions the client-side service will need to perform.

1. Format the stop data into the XML or JSON format that is documented in **Attachment A – AB953 Interface XML** or that is documented in **Attachment B - AB953 Interface JSON**.
2. Format the HTTP Request message for a POST transaction as described later in this document.
3. Initiate a web service call the DOJ Hosted Web Service. The URL for this web service will be provided once it has been defined.
4. Process the HTTP Response message that is received from the DOJ. This message will contain a variety of response codes that will need to be handled. These codes will range from SUCCESS code to a wide variety of error codes.
5. Process the Error codes and determine the corrective actions to fix the issues with the stop data record. This may involve requesting missing or corrected data from the Officer.
6. Resend the web service request as a PUT transaction once the indicated error(s) have been addressed. This will re-execute functions until all errors have been addressed and the data has been successfully submitted to the database.

Server-side Services (DOJ)

The server-side DOJ Hosted web service will need to perform several functions to process the HTTP Request message and prepare the HTTP Response Message. The following is a high-level list of the business functions the client-side service will need to perform.

1. Receive the web service call from the client-side service.
2. Provide a receipt acknowledgement to the client-side service.
3. Validate the attributes in the HTTP Request message to ensure the message is valid. This will involve actions as validating the ORI # from the header to verify the message is from a valid LEA.
4. Extract the metadata from the request body and validate that if contains valid XML or JSON.
5. Prepare a Message for the Common Validation Service execution.
6. Initiate a call to the Common Validation Service.
7. The Common Validation Service will then process the stop data from the XML or JSON and prepare a response message with any error codes for invalid or missing data using the rule in **Technical Data Dictionary** to validate the stop data record.
8. In the event that the data is valid the stop data will be inserted into the stop data database in a “submitted” status. In the event that the stop data failed the data validations insert the data into the stop data database in a status to indicate that the data “Validation Failed”.
9. Assemble the required logging and auditing records for the stop data record and insert them into the stop data database.
10. Prepare a response message and send it to the client-side service indicating the status of the web service call.

Interaction Type

The DOJ hosted web service will be called to process both synchronous real-time and bulk initiated services. Based on this the service will have two access modes that both follow a real-time processing mode. It will process transactions from the LEA systems in a real-time mode and will accept service calls from the DOJ or LEA Batch Bulk Stop Data Processor in real-time to process the stop data.

The web service will support requests and responses communications that utilize the HTTPS protocol. Meaning, the web services will support the secure hypertext transfer protocol (HTTPS) in transferring encrypted information between the provider and requestor entities in the web service interactions.

Available Methods

Currently, the web service will provide two (2) methods to the subscriber. These methods are shown in the table below.

Table 4 – Stop Data Service Methods

Resource	Methods	URI	Description
LEA_New_Stop Collection_Data	POST	TBD	This method will allow the LEAs to submit a new SDCS stop data to the DOJ.
LEA_Updated_Stop Collection_Data	PUT	TBD	This method will allow the LEAs to submit an updated SDCS stop data record to the DOJ. The PUT verb can only be used for records in an “Submitted with Errors” status. PUT requests for a “submitted” record will be rejected.

HTTP Request

The following is the information that is currently planned to be included in the HTTP Request message for DOJ Hosted web service.

Table 5 – Stop Data Service HTTP Request Data Attributes

Part	Description	Required (Y/N)	DataType
Verb	PUT or POST	Y	String
URI	TBD		
HTTP Version	1.1	Y	String
Request Header	ORI #	Y	String
	LEARecordID	Y	String
	DOJRecordID	N	String
	Officer UID	Y	String
	Stop Date	Y	String
	Stop Time	Y	String

Part	Description	Required (Y/N)	DataType
	Browser type (Chrome, Edge, Safari, Firefox)	Y	String
	Resource Type (XML or JSON)	Y	String
	Calling Application (RMSWS, LEAFTP)	Y	String
Request Body	The request body will contain the XML or JSON representation of the SDCS Stop Data. Please refer to **Attachment A - AB953 Interface XML** and **Attachment B - AB953 Interface JSON** for the specific format and contents of this data stream.	Y	XML or JSON Data

HTTP Response

The following is the information that is currently planned to be included in the HTTP Response message for DOJ Hosted web service.

Table 6 – Stop Data Service HTTP Response Data Attributes

Part	Description	Required (Y/N)	DataType
Response Code	Please see error handling section	Y	String
HTTP Version	1.1	Y	String
Response Header	ORI #	Y	String
	LEARecordID	Y	String
	DOJRecordID	Y	String
	Officer UID	Y	String
	Stop Date	Y	String
	Stop Time	Y	String
	Resource Type (XML or JSON)	Y	String
	Browser type (Chrome, Edge, Safari, Firefox)	Y	String
Response Body	As the SDCS services utilize this message part for the storage of the variety of error messages that are raised during the validations of the stop data record.	Y	Metadata representation of the encounter errors by the central validation service.

Error Handling

This section will describe the error codes that are handled by the service and any Service Based Faults that the service is capable of raising. The following table contains the HTTP response codes that are supported by the web service. As such it will not contain the errors that are raised by the data validation service.

The Error Handling column will be utilized to describe whether and how the LEA's should handle the errors returned from the interface.

Table 7 – Stop Data Service HTTP Response Codes

HTTP Code	Name	Description	Error Handling
200	CREATED	Resource was successfully created using POST or PUT	No action required
304	NOT MODIFIED	Used for data validation errors where the stop data record has been saved in an “Submitted with Errors” status	This code will indicate that an error occurred during the validation of the stop data record in the central validation service.
400	BAD REQUEST	Invalid input in service call	The LEA will need to re-submit the service call with the required XML or JSON data in the message.
401	UNAUTHORIZED	The user is using an invalid or bad authentication token or codes	This will occur when the ORI # in the message header is invalid. The LEA will need to re-submit the message with a valid ORI #.
403	FORBIDDEN	User does not have access to the requested method.	The LEA will need to examine the submitted service call to validate it is utilizing the correct HTTP Request message attributes.
404	NOT FOUND	Requested method is not available or cannot be found	This error will be raise anytime the message header includes any verb other than PUT or POST. The LEA will need to only use the PUT or POST verb.
405	Method Not allowed	A request method is not supported for the requested resource; for example, a GET request on a form that requires data to be presented via POST, or a PUT request on a read-only resource.	This error will be raised when a PUT verb is used for a record in the database that is in a “successful submission ” or “NFIA” status.

HTTP Code	Name	Description	Error Handling
409	CONFLICT	Conflict situation while executing the method. For example, attempting to add a duplicate entry.	This error will be raised when an attempt is made to insert a duplicate stop data record.
500	INTERNAL SERVER ERROR	The server encountered an error or exception while executing the method.	This error will be raised when an un-expected error is encountered in the service call. The LEA should contact the DOJ technical support team for resolution.

Fatal Errors

The web service will generate several errors that are classified as fatal errors. The definition of a fatal error in the context of the SDCS web services are any errors that are encountered where the stop data record can not be saved to the database or the stop data record cannot be saved to the database without compromising the integrity of the database. The following table 3 lists the preliminary list of the fatal errors that can be generated.

Table 8 – Web Service Fatal Errors

Name	Description
Message Request Body Missing or Invalid	This error is generated when one of two situations occurs: 1) the message does not contain valid XML or JSON 2) the Message Request Body is empty
Invalid VERB in Request Message	This error will be generated when a VERB other than PUT or POST is passed in the Request Message.

Interface Security

The DOJ Hosted web service will be structured to fully comply with the DOJ standards. It is anticipated the while the service will not leverage WS-Security of UDDI that it will need to leverage a SSL cert to help ensure the security of the service.

3. Bulk Stop Data Processor Module

Operational Description

The SFTP service, through the DOJ Bulk Stop Data Upload Processor, will be capable of receiving stop data records in 4 record formats. These are:

1. XML data as defined in **Attachment A - AB953 Interface XML**
2. JSON data as defined in **Attachment B - AB953 Interface JSON**
3. Excel as defined in **Attachment C - Excel-CSV Bulk Upload Format**
4. Delimited-text format using the same overall format as **Attachment C - Excel-CSV Bulk Upload Format** with the allowance to using a "|" character as the data delimiter.

Any data that is received by the DOJ SFTP service that that does not comply to these record formats will be rejected by the DOJ and will not be saved to the DOJ stop data database. The bulk stop data upload processor module will be responsible for the management of the data that is transmitted via the SFTP services and for the submission of the data to the DOJ central validation service.

The following will be the high-level processing flow of this module when it is called by the SFTP job scheduler:

1. The DOJ Job Scheduler will initiate a session with the Bulk Stop Data Upload Processor module when a new file is placed on the SFTP server. The SDCS SFTP services will leverage the existing DOJ Tidal Enterprise Job Scheduling Services.
2. If the record format can not be identified as a supported format it will return an invalid file format error to the LEA.
3. If the records are identified as XML or JSON or Excel or delimited-text format the Bulk Stop Data Upload Processor will read the record(s) from the FTP file(s) one record at a time and send the record to the SDCS service for validation and processing.
4. The services will return to the Bulk Stop Data Upload Processor the status of each record that has been processed.
5. All records that pass the initial ORI and Data validations will be saved to the stop data database by the services in either a Successful Submission or Submitted with Errors status depending on the results on the requested data validations.
6. Any records that do not pass all of the data validations will be returned to the Bulk Stop Data Upload Processor along with the error messages.
7. After all records in the SFTP file(s) have been processed the Bulk Stop Data Upload Processor will consolidate any error records into a single file.
8. This file of the error records is then placed into Agency OUTPUT folder on the SFTP server.
9. An email message will be sent to the LEA, at the determined frequency, by the Bulk Stop Data Upload Processor indicating that there were errors in the file they uploaded and will be instructed the LEA how to download the error files.

The following will be the high-level processing flow of this module when it is called by the DOJ Hosted Web Services that were initiated by a bulk processing mode from the LEA:

1. Any records that do not pass all of the data validations in the web services when the web services are called from a bulk process by the LEA will be returned to the Bulk Stop Data Upload Processor along with the error messages from the DOJ Hosted Web Service.
2. The error records and error messages are then placed into Agencies OUTPUT folder on the SFTP server.
3. An email message will be sent to the LEA, at the determined frequency, by the Bulk Stop Data Upload Processor indicating that there were errors in the record that was included in the service call and they will be instructed how to download the error files.

Error Handling

Fatal Errors

The Bulk Stop data processor module will be capable of generating two fatal errors that will prevent it from sending the stop data record to the services for processing and validation. These conditions are listed in the following table.

Table 9 – Bulk Upload Processor Module Fatal Errors

Name	Description
SFTP Record type invalid	This error will be generated when the type of the records in the SFTP file cannot be identified as XML, JSON, or Excel or Delimited-text.
Conversion Error	This error is generated when a delimited text or Excel file is received in a SFTP submission. The error is generated when the routine that binds the delimited data or Excel data or XML or JSON to the transport object fails.

Application Log Data

The Bulk Upload Processor will generate a log file that documents the processing status of each file that was received for the LEA community through the FTP services. This file will be generated by the Bulk Data Upload Processor Module to log the status of each file that it processes. This file will be placed in the OUTPUT folder on the FTP server for the LEA to download and to conduct any file reconciliations processes that are required by the LEA.

Table 10 – Bulk Upload Processor Log Data Attributes

ID	Attribute	Description	DataType
1	ORI #	The ORI # of the LEA that transmitted the file.	String(9)
2	File Name	The physical filename of the file that was transmitted by the LEA.	String(30)
3	File Date	The date the file was received from the LEA	String(10)
4	Access Channel	This will always be set to FTP	String(3)
5	Processing Status	The status of processing the file. This will be a formatted string that includes the count of total records, records processed successfully, and error records.	String(75)
6	Processing date	The date the record was processed.	Date(10)
7	Processing time	The time the record was processed.	Time(8)
		TOTAL LENGTH	145 bytes

4. Submit Stop Data SFTP Service

Operational Description

The SFTP service will allow LEAs to upload bulk stop data submissions to the DOJ. Once these submissions have been completed and the file transmission is verified, the stop data records will be passed to the DOJ Bulk Stop Data Upload Processor that is described in the previous section for processing, validation and storage in the stop data database. Any file transmission or stop data validation errors will be sent to the LEA for resolution.

The SDCS SFTP service will leverage the existing DOJ SFTP services.

Interaction Type

This SFTP service will utilize a batch interaction type.

Record Layout and Definition

The SFTP services will leverage the XML or JSON format that is documented in **Attachment A – AB953 Interface XML** and **Attachment B - AB953 Interface JSON** or the Excel and Delimited Text format in **Attachment C - Excel-CSV Bulk Upload Format**. Please refer to these attachments for the format of the records that will be transmitted to the DOJ FTP service.

File Naming Convention

The SDCS SFTP services will support four file types. The following is a lists of the file naming conventions for each of the four file types:

1. XLSX for bulk processing
   a. Naming convention: <*YYYYMMDDHHMMSS*>_<*ORI*>_<*LEABatchID*>.xlsx or <*YYYYMMDDHHMMSS*>_<*ORI*>_<*LEABatchID*>.xls
2. CSV with "|" pipe delimiter for bulk processing
   a. Naming convention: <*YYYYMMDDHHMMSS*>_<*ORI*>_<*LEABatchID*>.csv
3. XML for individual stop record processing
   a. Naming convention: <*YYYYMMDDHHMMSS*>_<*ORI*>_<*LEARecordID*>.xml
4. JSON for individual stop record processing
   a. Naming convention: <*YYYYMMDDHHMMSS*>_<*ORI*>_<*LEARecordID*>.json

All file transmissions shall leverage these file naming conventions. Any files that do not utilize these file naming conventions will not be processed by the DOJ SFTP services.

Error Handling

The FTP access Channel will not generate any error messages outside of the standard errors that are generated by the SFTP software.

Interface Security

The SFTP services that are provided for the SDCS solution will leverage standard FTP logins. Each LEA that choses to leverage the FTP data access channel will be assigned a unique User ID and Password that they can use to perform the file transfers. However, any LEA that has NOT requested the FTP access channel will NOT be assigned a user ID and Password.

Application Log Data

The FTP Access Channel will generate a standard FTP log file as defined by the FTP software product that documents its processing status.