

Software Design Document

for

Box.com/eDefender Integration

Version 1.1 approved

Prepared by Joshua Cabrera, Marco De La Torre, Raul Gallegos, Daniel Guevara-Dominguez, Chuang Huang, Dang Le, Jesica Lopez De Leon, Shaocheng Shi, Sergio Tapia, Luke Williams

Santa Barbara Public Defender Office

October 7, 2021

Table of Contents

Revision History	3
1.1 Purpose	4
1.2 Document Conventions	4
1.3 Intended Audience and Reading Suggestions	4
1.4 System Overview	5
2. Design Considerations	5
2.1 Assumptions and Dependencies	6
2.2 General Constraints	6
2.3 Goals and Guidelines	6
2.4 Development Methods	7
3. Architectural Strategies	7
4. System Architecture	8
5. Policies and Tactics	10
5.1 Choice of which specific products used	10
5.2 Plans for ensuring requirements traceability	11
5.3 Plans for testing the software	11
6. Detailed System Design	12
6.1 Box	12
6.1.1 Responsibilities	12
6.1.2 Constraints	12
6.1.3 Composition	12
6.1.4 Uses/Interactions	12
6.1.5 Resources	12
6.1.6 Interface/Exports	12
6.2 Box Skills App	12
6.2.1 Responsibilities	12
Box Skills makes use of third-party AI/ML to get metadata within media files. Analyzes audio and video to produce a transcript of the speech detected in English and Spanish. Detects people in the videos using facial recognition.	12
6.2.2 Constraints	12
6.2.3 Composition	13
6.2.4 Uses/Interactions	13
6.2.5 Resources	13
6.2.6 Interface/Exports	13

6.3 eDefender	13
6.3.1 Responsibilities	13
6.3.2 Constraints	13
6.3.3 Composition	13
6.3.4 Uses/Interactions	13
6.3.5 Resources	13
6.3.6 Interface/Exports	13
6.4 Document Tag Parser App	13
6.4.1 Responsibilities	13
6.4.2 Constraints	14
6.4.3 Composition	14
6.4.4 Uses/Interactions	14
6.4.5 Resources	14
6.4.6 Interface/Exports	14
7. Detailed Lower level Component Design	14
8. Database Design	14
9. User Interface	15
9.1 Overview of User Interface	15
9.2 Screen Frameworks or Images	15
10. Requirements Validation and Verification	17
10.1 Functional Requirements	17
11. Glossary	21
12. References	21

Revision History

Name	Date	Reason For Changes	Version
Team	10/7/21	First Draft	1.0
Team	12/7/21	Update Project Design	1.1

1. Introduction

1.1 Purpose

The purpose of this software is to integrate Box.com with Santa Barbara Public Defender's office case management system, eDefender. Allowing for uploaded documents/files to be transcribed/translated. This is achieved by utilizing Azure Video Analyzer which provides artificial intelligence to transcribe audio and video files. It builds upon a project created by a previous team at CSULA in 2020 titled "Extension for Public Defender's Client Case Management System." This document is based on the software design document titled "Visual Analysis Using Cloud Services." In addition, a document tag parser app separate from the Box Skill app will be created to read the tag number on a document and rename the file based on SBPD's naming convention.

1.2 Document Conventions

N/A

1.3 Intended Audience and Reading Suggestions

This document is intended for software developers, project managers, testers, and users. This document covers the specifics of the Box.com and eDefender integration along with information from Azure Video Analyzer which is used to transcribe/translate files. In addition it covers the specifics of the document tag parser application.

Suggested reading sequence for each is the following:

- Developers - Understand the purpose and functions of the product to help during development; Recommended Reading: Section(s) 2, 4, 6-8
- Project Managers - Understand the purpose of the product and its functions; Recommended Reading: Section(s) 2-5
- Testers - Understand the purpose and requirements of the product to help during testing to guarantee intended function; Recommended Reading: Section(s) 2, 4, 5, 9

- Users - Understand the purpose of the product and its functions; Recommended Reading: Section(s) 1, 9

1.4 System Overview

Our system will allow for users to upload files to Box.com, which will trigger a custom Box Skill Application that will transcribe and translate files. Once successfully completed users will be able to view metadata from the file that consists of transcription, facial recognition, and timestamps. It will also include a notification system within the eDefender case management system to alert the handling attorney.

The Box Skill application can be deployed to any server that can support a publicly available url such as serverless AWS lambda or Google cloud functions, as well as traditional servers on AWS and Firebase.

The document tag parser will be a small local application installed on each employee's system.

2. Design Considerations

Note that some of the dependencies and constraints have carried over from the original project, but some have changed.

2.1 Assumptions and Dependencies

- For the Box Skill App and Notification Service:
 - Node JS
 - Box SDK
 - AWS Lambda and S3 Bucket
 - Azure Video Analyzer
 - eDefender API
- For the Document Tag Parser:
 - Tesseract OCR Library
 - Pdf2image Library
 - Pypdf2 Library
 - PIL Image Processor Library

2.2 General Constraints

- When multiple people are speaking during a video or audio recording, transcription accuracy is unpredictable
- File security due to sensitive court documents and evidence
- eDefender
 - Notifications will need to be integrated into Santa Barbara Public Defender's environment, and additional work will be needed outside of the Box Skill App
- Document Tag Parser
 - Accuracy of file renaming depends on image to text conversion of tesseract machine learning algorithm
 - Documents vary based on file type and position of the tag number on each document. Multiple rounds of testing will be needed to ensure accurate file renaming.

2.3 Goals and Guidelines

- Deliver final project in April 2022
- Box Skill App
 - Ensure accurate transcription and ease of use within Box UI
- Document Tag Parser
 - Ensure accuracy of document renaming by handling different document file types appropriately

2.4 Development Methods

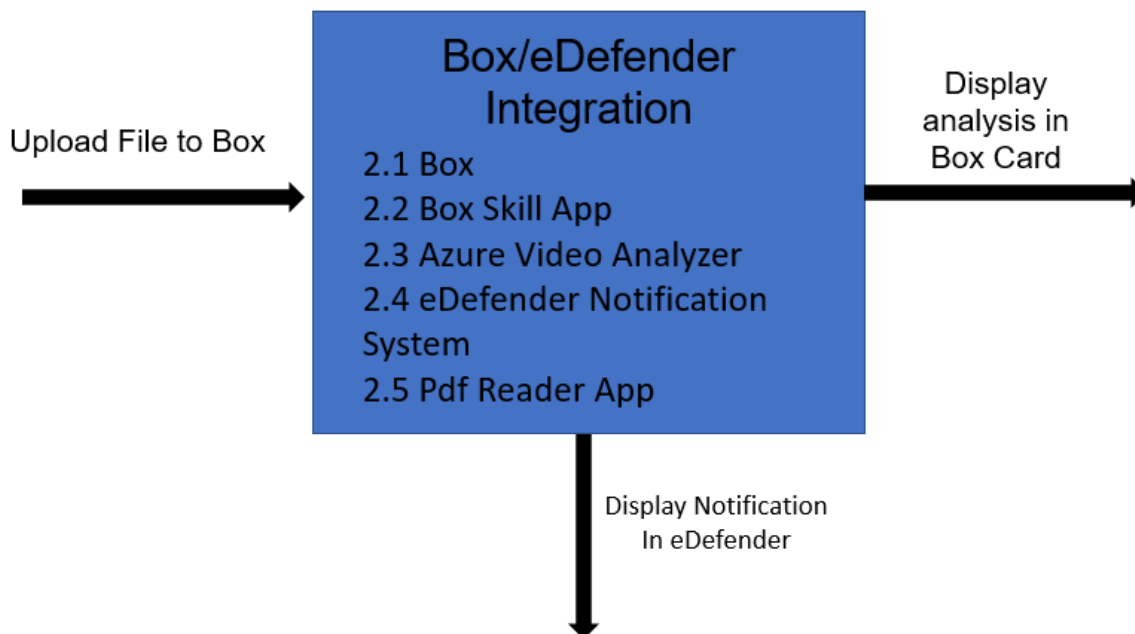
- We are using the Agile Development Methods

3. Architectural Strategies

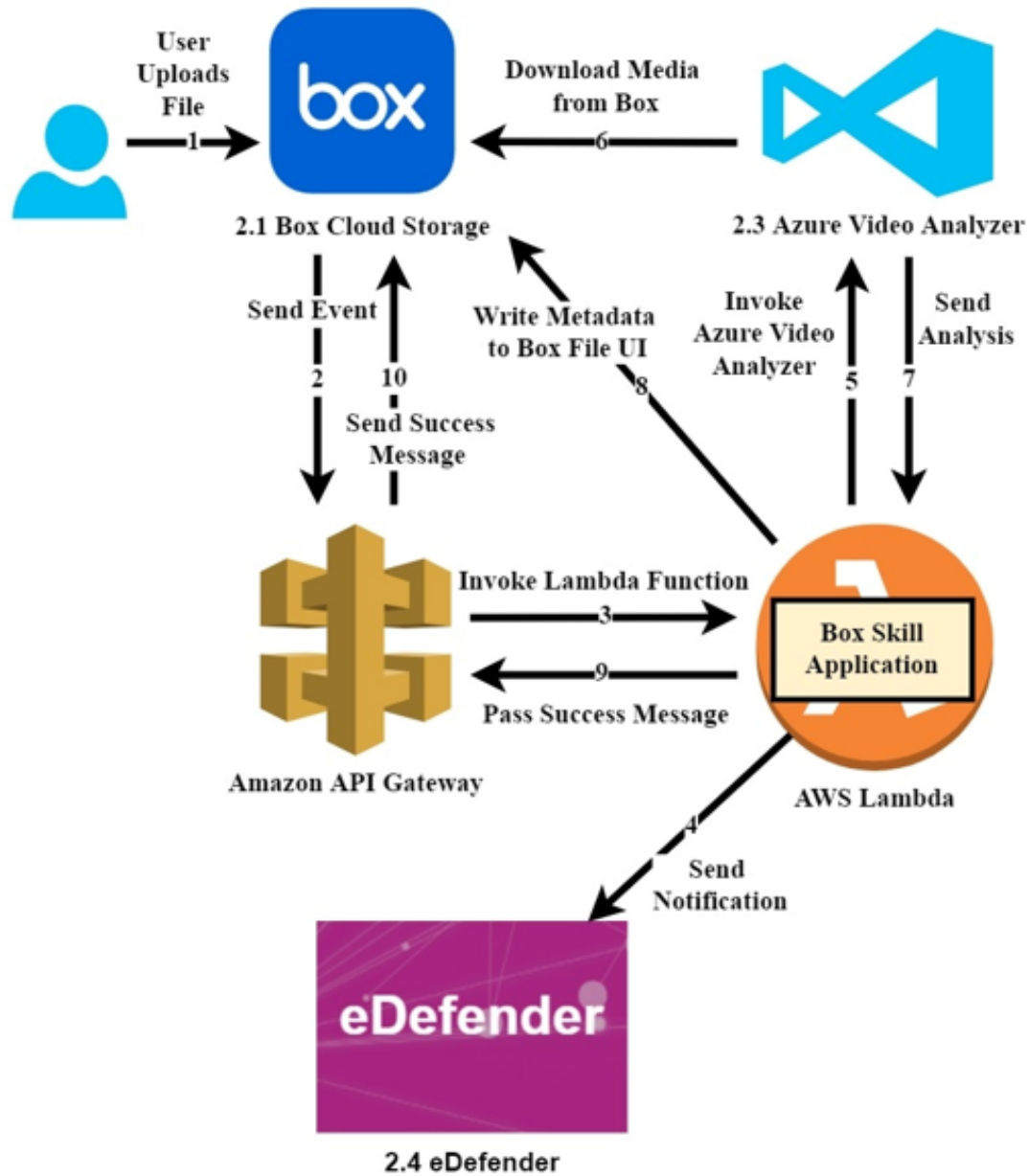
- We are building upon the existing design by the prior CSULA team, document titled “Visual Analysis Using Cloud Services”
- Reusing Original Box Skill App
 - Add eDefender Alert System
 - Azure Video Indexer was used by the prior team, this new version of the application will use the updated version, Azure Video Analyzer
 - Migrate our deployment of the application on our team’s lambda function to Santa Barbara Public Defender’s instance
- Document Tag Parser
 - Functions as a separate application from the Box Skill app that runs locally on a user’s machine
 - Combining the functionality of the libraries mentioned in 2.1
 - Refer to DFD Level 1 below for detailed architecture

4. System Architecture

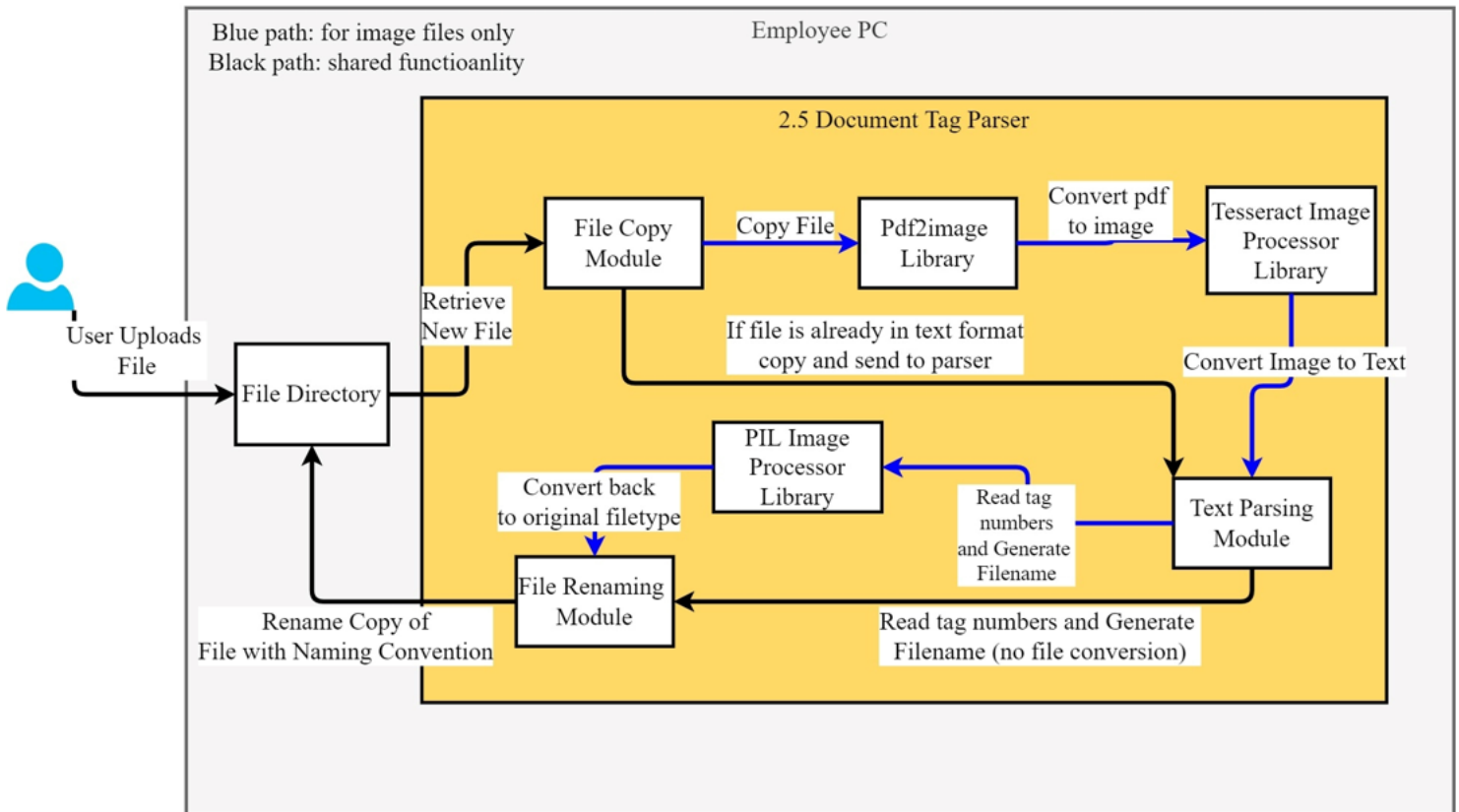
Box/eDefender Integration Level 0 DFD



Box/eDefender Integration DFD Level 1



Box/eDefender Integration DFD Level 1 Continued



5. Policies and Tactics

5.1 Choice of which specific products used

Box.com which is a cloud content management tool allows the Public Defender to be compliant with the CJIS/HIPAA requirements. This cloud content management tool also provides the custom Box skills application which enables the ability to apply third-party AI technologies to automatically analyze files as they are uploaded to Box.

Azure Video Analyzer which is a cloud application which analyses both visual and audio files by running different AI models. As the video analyzer runs an analysis on the files, the insights are extracted by the AI models that are then provided.

5.2 Plans for ensuring requirements traceability

N/A

5.3 Plans for testing the software

The Box Skills application will be tested with video file types all of which contain different languages in order to test the accuracy of the Video Indexer. The application will also be tested with videos of varying lengths to test the tokens.

Our plans for testing the Document Tag Parser software includes using example files provided by the Public Defender's office of Santa Barbara. Our Software uses the files which are PDF's, transforms them into images, via Tesseract OCR Library, in order for the program to translate the images into text. When the images are then in text format, this allows the program to search for the specific aspects we need from the files via "Regex" within our program. This will ensure that the program will find the two tag numbers we need in order to accomplish our objective which is renaming all files each with their two tag numbers included as well as the disc number that is provided for each file.

6. Detailed System Design

6.1 Box

6.1.1 Responsibilities

Allows users to upload and download files.

6.1.2 Constraints

Users must have a Box account provided by SBPD.

6.1.3 Composition

N/A

6.1.4 Uses/Interactions

Provides security by only allowing signed in authorized users. Provides file management features.

6.1.5 Resources

Already existing file management web app Box.com

6.1.6 Interface/Exports

Website Box.com

6.2 Box Skills App

6.2.1 Responsibilities

Box Skills makes use of third-party AI/ML to get metadata within media files. Analyzes audio and video to produce a transcript of the speech detected in English and Spanish. Detects people in the videos using facial recognition.

6.2.2 Constraints

It is not its own application; it requires use of other APIs. The transcriptions are not 100% reliable.

6.2.3 Composition

N/A

6.2.4 Uses/Interactions

N/A

6.2.5 Resources

Uses AWS Lambda

6.2.6 Interface/Exports

Works within Box.com

6.3 eDefender

6.3.1 Responsibilities

File management system used by SBPD to organize case files. Alerts attorney when new discovery is received.

6.3.2 Constraints

Users must have authorization provided by SBPD. Currently a lot of manual work is being done by the SBPD support team. Our goal is to automate some of these tasks.

6.3.3 Composition

N/A

6.3.4 Uses/Interactions

Interacts with Box to get transcribed files. It will provide the files that need to be read by the Document Tag Parser App.

6.3.5 Resources

TBD

6.3.6 Interface/Exports

TBD

6.4 Document Tag Parser App

6.4.1 Responsibilities

The document tag parser accepts pdf, jpg, and png files and converts them into text files for parsing. It extracts the tag number from files and renames the tag file with the tag number included. The format of the tag number will be first page-last page_disc #_filename.

6.4.2 Constraints

There are three scenarios for files that need to be read, single page file, multiple page file, and non-stamped file. This makes it difficult to extract the tag number.

6.4.3 Composition

N/A

6.4.4 Uses/Interactions

Will be used by eDefender as a separate app.

6.4.5 Resources

DTP relies on packages/dependencies defined in section 2.1

6.4.6 Interface/Exports

DTP provides the ability to automatically rename a file based on the naming convention defined by SBPD

7. Detailed Lower level Component Design

N/A

8. Database Design

N/A

9. User Interface

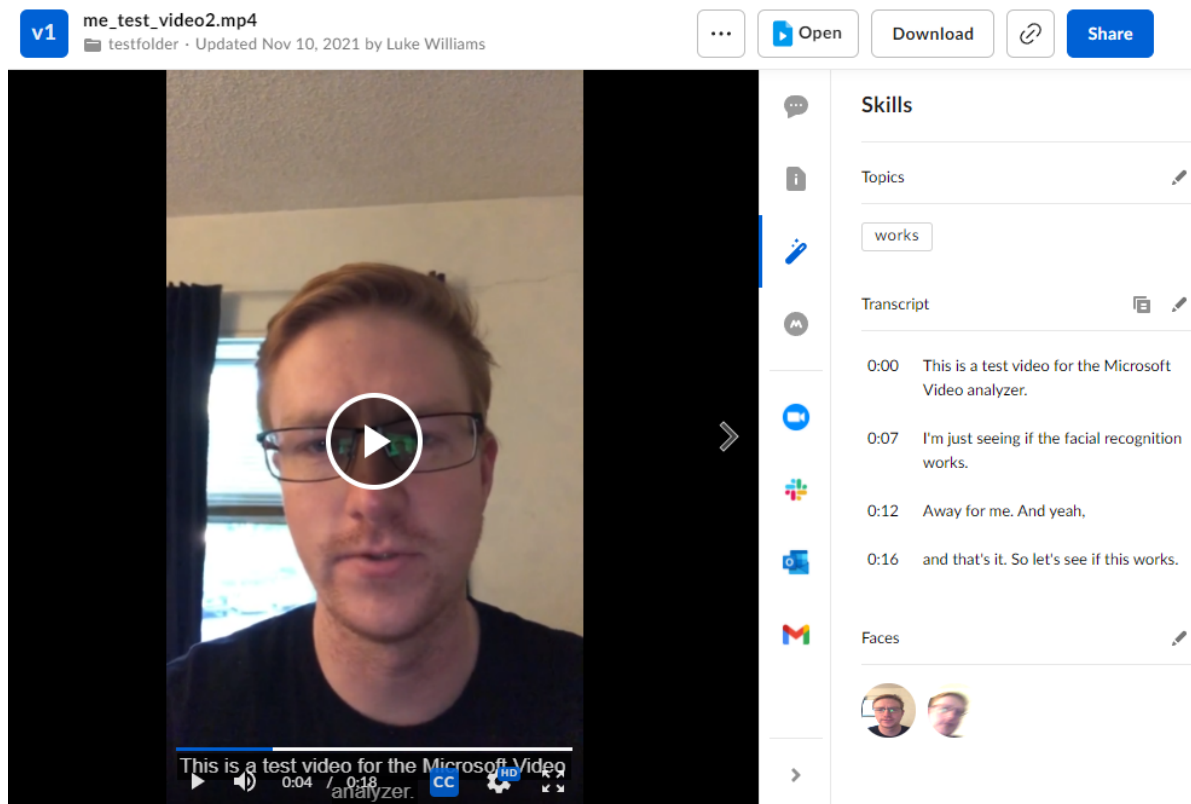
9.1 Overview of User Interface

Describe the functionality of the system from the user's perspective. Explain how the user will be able to use your system to complete all the expected features and the feedback information that will be displayed for the user. This is an overview of the UI and its use. The user manual will contain extensive detail about the actual use of the software.

9.2 Screen Frameworks or Images

These can be mockups or actual screenshots of the various UI screens and popups.

Box.com UI based on Azure Video Analyzer Results



Document Tag Parser UI Concept



The image shows a mockup of a web application window titled "Document Tag Parser". The window has a grey header bar with a red close button (X) in the top right corner. The main content area has a light green background. It features two input fields for file paths, each with a "Choose Directory" button. The first input field is labeled "Choose Input Path" and contains the text "C://Users/Me/docume.pdf". The second input field is labeled "Choose Output Path" and contains the text "C://Users/Me". Below these fields is a checkbox labeled "Output Path is same as input" which is checked. To the right of the checkbox is a white button labeled "Disc_05". Further right is a red button labeled "Process". At the bottom of the window, there is a "Welcome!" message followed by a bulleted list of two items: "This application supports the following file types: pdf, jpg, png" and "After clicking process, a copy of your input file will be made with the proper naming convention using the document tag numbers. It will appear in the chosen output folder."

Document Tag Parser

Choose Input Path

C://Users/Me/docume.pdf

Choose Directory

Choose Output Path

C://Users/Me

Choose Directory

Output Path is same as input

Disc_05

Process

Welcome!

- This application supports the following file types: pdf, jpg, png
- After clicking process, a copy of your input file will be made with the proper naming convention using the document tag numbers. It will appear in the chosen output folder.

9.3 User Interface Flow Model

N/A

10. Requirements Validation and Verification

10.1 Functional Requirements

BOX/e-Defender Integration:

1.1:Box		Met by
10.1-1.1-1	Box shall allow users upload and download files	Box.com
10.1-1.1-2	Box shall trigger BCS upon file upload to pre-configured folder	Box.com
10.1-1.1-3	Box shall send access token to hosting site(AWS)	Box.com
10.1-1.1-4	Box shall embed metadata to original video/audio	Box Skills
10.1-1.1-5	Box shall visualize metadata as transcript, face, and topic card	Box Skills

1.2:AWS		Met by
10.1-1.2-1	AWS shall host the Box skill application	AWS Lambda
10.1-1.2-2	AWS shall capture and parse all information of an event send by Box	AWS Lambda
10.1-1.2-3	AWS shall send an event to ML service(Azure Video Analyzer)	AWS Lambda
10.1-1.2-4	AWS shall send access token to ML service	AWS Lambda
10.1-1.2-5	AWS shall receive response from ML service	AWS Lambda

10.1-1.2-6	AWS shall retrieve metadata from ML service	AWS Lambda
10.1-1.2-7	AWS shall write metadata back to Box	AWS Lambda

1.3:ML service(AVA)		Met by
10.1-1.3-1	ML service shall listen for AWS call to process video/audio	Azure Video Analyzer
10.1-1.3-2	ML service shall use access token to retrieve file from Box	Azure Video Analyzer
10.1-1.3-3	ML service shall produce transcription of a file	Azure Video Analyzer
10.1-1.3-4	ML service shall perform language auto detection	Azure Video Analyzer
10.1-1.3-5	ML service shall recognize faces and keywords	Azure Video Analyzer
10.1-1.3-6	ML service shall provide timestamp for analyzed result	Azure Video Analyzer

Document Tag Parser(DTP):

1.4:Document Tag Parser(DTP)		Met by
10.1-1.4-1	DTP shall convert pdfs and images into text files for parsing	Tesseract Image Processor Library
10.1-1.4-2	DTP shall read the tag numbers on the first and last page of each document	Text Parsing Module

10.1-1.4-3	DTP shall rename the file based on the SBPD'S naming convention	File Renaming Module
10.1-1.4-4	DTP shall add the disc number based on the user's choice within the application	File Renaming Module
10.1-1.4-5	DTP shall not modify the original file	File Copy Module
10.1-1.4-6	DTP shall create new copies with the appropriate file naming convention	File Renaming Module

11. Glossary

- SBPD: Santa Barbara Public Defender Office
- SRS: Software Requirements Specification
- SDD: Software Design Document
- AI/ML: Artificial Intelligence/Machine Learning
- Box: Cloud Storage provider
- Box Skills: A framework used to provide customization for folder/files on Box.
- AWS: Amazon Web Services
- AWS Lambda: A serverless, event-driven compute service
- E-Defender: supplies case management software to courts, prosecutors, public defenders, probation and other justice agencies.
- BCS: Box Custom Skill
- AVA: Azure Video Indexer
- DTP: Document Tag Parser

12. References

- Extension for Public Defender's Client Case Management System CSULA team 2020-2021 <https://csns.cysun.org/department/cs/project/view?id=7248892>