

Software Requirements Specification

for

Box.com and eDefender Integration

Version 3.0.0 approved

Prepared by Branden Zamora, Jacqueline Molina, Luiz Perez Campos, Florian Haule, Leo Gallardo, Donovan Hatfield, Coby Alvarez, Jose Alvarado, Dat Nguyen, Brian McNaughton

Santa Barbara Public Defender's Office

December 8th, 2023

Table of Contents

Table of Contents.....	2
Revision History.....	3
1. Introduction.....	4
1.1 Purpose.....	4
1.2 Intended Audience and Reading Suggestions.....	4
1.3 Product Scope.....	5
1.4 Definitions, Acronyms, and Abbreviations.....	5
1.5 References.....	6
2. Overall Description.....	7
2.1 System Analysis.....	7
2.2 Product Perspective.....	7
2.3 Product Functions.....	8
2.4 User Classes and Characteristics.....	8
2.5 Operating Environment.....	9
2.6 Design and Implementation Constraints.....	9
2.7 User Documentation.....	9
2.8 Assumptions and Dependencies.....	9
2.9 Apportioning of Requirements.....	9
3. External Interface Requirements.....	10
3.1 User Interfaces.....	10
3.2 Hardware Interfaces.....	12
3.3 Software Interfaces.....	12
3.4 Communications Interfaces.....	12
4. Requirements Specification.....	13
4.1 Functional Requirements.....	13
4.2 External Interface Requirements.....	15
4.3 Logical Database Requirements.....	16
4.4 Design Constraints.....	16
5. Other Nonfunctional Requirements.....	17
5.1 Performance Requirements.....	17
5.2 Safety Requirements.....	17
5.3 Security Requirements.....	17
5.4 Software Quality Attributes.....	18
5.5 Business Rules.....	18
6. Legal and Ethical Considerations.....	19
Appendix A: Glossary.....	20
Appendix B: Analysis Models.....	21
Appendix C: To Be Determined List.....	23

Revision History

Name	Date	Reason For Changes	Version
Initial Setup	11/25/23		1.0.0
First Draft	12/02/23		2.0.0
Final Draft	12/08/23		3.0.0

1. Introduction

1.1 Purpose

This document is meant to outline the technical requirements needed by the developers who worked on the Box.com and eDefender Integration system. This document in particular includes insight on various types of requirements (functional and nonfunctional), functions, constraints; while simultaneously providing a short explanation for each component.

1.2 Intended Audience and Reading Suggestions

The intended audience for this document are the developers who are continuously working on maintaining this system and for users who aim to work with the system. In this specific instance, the audience ranges from CalStateLA students and staff to SBPD personnel.

The SRS document contains six distinct sections. The first section is the Introduction. This section contains an extremely high level description of the project scope; includes definitions, acronyms, abbreviations, and references to external documents. The second section is the Overall Description. This describes the problem the team is aiming to solve and the details of the approach that the team selected. Like the introduction, this section maintains a high level view of the project. The third section is the External Interface Requirements. This, again, is meant as a high level description. It describes how the project interacts with any and all external interfaces. The fourth section is the Requirements Specification. It contains all necessary software requirements needed for developers to accurately design the project. It is important to note that this section is not HOW things will be implemented, but WHAT will be implemented. The fifth section is Other Nonfunctional Requirements. This includes performance, safety, and security requirements. The sixth section is Legal and Ethical Considerations. This denotes any legal or ethical issues involved in the project.

If the reader would prefer a general overview of the project, it is recommended that they read the Introduction, Overall Description, and External Interface Requirements.

If the reader would prefer a technical overview, it is recommended that they read through the Requirements Specification, Nonfunctional Requirements, and peruse Appendix B: Analysis Models.

1.3 Product Scope

The Box.com/eDefender Integration system is a serverless cloud-based program which combines the use of three primary services: Amazon Web Services, Azure Video Indexer, and the Box.com cloud content management system. The goal of the project is to use visual analysis provided by Video Indexer to speed up the Public Defender's ability to review digital media evidence involving audio and/or video on the Box website during the initial discovery phase of a case.

Overall, the team promises to deliver a user-friendly and easy to navigate platform for the Santa Barbara Public Defender's Office.

1.4 Definitions, Acronyms, and Abbreviations

Term	Definition, Acronym, Abbreviation
SBPD	Santa Barbara Public Defender's Office
Box.com	Secure cloud content management system.
SRS	Software Requirements Specification
VI/Video Indexer	Microsoft Azure Video Indexer
AWS	Amazon Web Services

1.5 References

Box.com API Reference

<https://developer.box.com/reference/>

AWS v3 Documentation

<https://docs.aws.amazon.com/s3/>

Video Indexer Documentation

<https://learn.microsoft.com/en-us/azure/azure-video-indexer/>

Serverless Documentation

<https://www.serverless.com/framework/docs>

Ascent Project Homepage

<https://ascent.cysun.org/project/project/view/204>

[2022-2023] Github

https://github.com/James-Yokley/box.com_eDefender_integration

[2023-2024] Github

https://github.com/branden12/Box.com_eDefenderIntegration_23-24

2. Overall Description

2.1 System Analysis

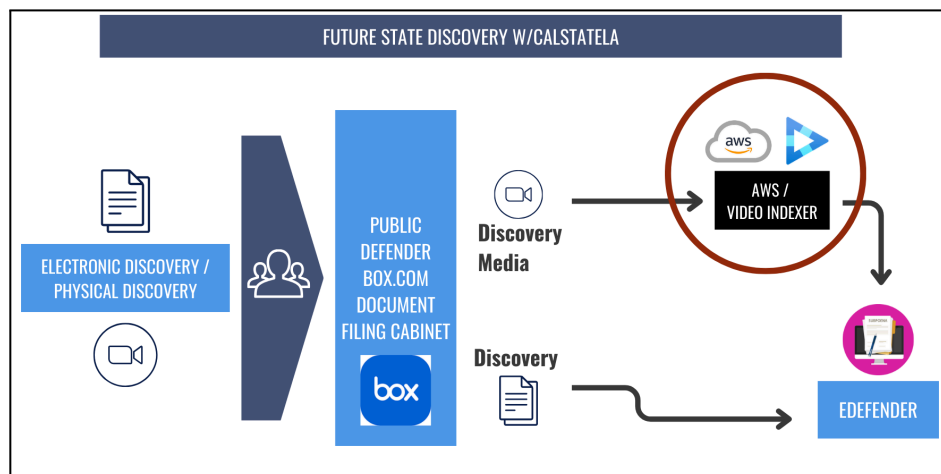
As previously mentioned, the aim of the project is to use visual analysis provided by Video Indexer to speed up the Public Defender's ability to review digital media evidence involving audio and/or video on the Box website during the initial discovery phase of a case. As of this moment, the amount of data incoming each month is growing exponentially, and if a solution isn't brought forth soon, the sheer amount of discovery media will overwhelm the faculty over at SBPD.

In order to achieve this goal, the team must work with various programs and APIs; including but not limited to: Video Indexer, Box Skills, Azure Custom Models, AWS Lambda, AWS S3, and additional AWS services.

The final product should emulate a transcribing software similar to many already on the market. One evident example being youtube and their automatic transcription service that they employ for subtitles.

2.2 Product Perspective

The software system integrates a variety of services: Serverless, Box, AWS, Video Indexer. This product is similar to an already existing product employed by the Los Angeles Public Defender's Office to automatically transcribe and tag discovery media uploaded to Box.com. While SBPD is looking for a product like the one mentioned, they would like to create a separate product in order to curate the software to the specific needs of the team. In our specific instance, we would like the digital transcription of discovery media, along with key words noted, face recognition, and an additional .docx transcription of the processed file.



2.3 Product Functions

2.3.1 Box Upload Event

- 2.3.1.1 Box Skills monitors file uploads to folders and in turn creates an 'upload event'
- 2.3.1.2 Box Skills event invokes AWS Lambda function through an Invocation URL

2.3.2 AWS Lambda Function & Services

- 2.3.2.1 Connects and authenticates event
- 2.3.2.2 Connects and authenticates Microsoft Video Indexer account
- 2.3.2.3 Stores event data in AWS S3 service
- 2.3.2.4 Receives new, processed, event from Video Indexer
- 2.3.2.5 Creates Box metadata cards using event from Video Indexer
- 2.3.2.6 Sends processed media back to original Box folder

2.3.3 Microsoft Azure Video Indexer

- 2.3.3.1 Utilizes machine learning to perform transcription and image recognition
- 2.3.3.1 Generates transcription and translation of audio
- 2.3.3.1 Generates timestamps of keywords and recognized faces

2.3.4 Document Parsing and Transcription

- 2.3.4.1 Receives metadata from Video Indexer to parse and transcribe
- 2.3.4.1 Parses metadata and returns a docx file to original Box folder

2.4 User Classes and Characteristics

2.4.1 Users

Users include lawyers or legal office professionals. These users would be the primary consumers of the product. They will be the ones uploading discovery media onto Box folders and waiting for the transcription and metadata to be generated.

2.4.2 System Developers/Administrators

System administrators are primarily in charge of maintaining and ensuring that everything runs smoothly on the backend. They will have access to the services in charge of handling all of the incoming data including administrator accounts. These users should have sufficient knowledge in handling code and fixing unforeseen errors.

2.5 Operating Environment

- 2.5.1 This software partially resides on Amazon Web Services
- 2.5.2 This software requires access to a Box.com account
- 2.5.3 This software requires access to a Video Indexer account
- 2.5.4 This software requires a stable internet connection.
- 2.5.5 This software runs on a Desktop or Laptop computer
- 2.5.6 This software is accessible by common web browsers (Safari, Chrome, etc.)

2.6 Design and Implementation Constraints

- 2.6.1 Authorization required to access Box.com folders
- 2.6.2 Operating costs of AWS
- 2.6.3 Operating costs of Video Indexer
- 2.6.4 Accuracy of transcription and keyword search varies and may require manual review
- 2.6.5 Accuracy of facial recognition is subject to errors and may require manual review

2.7 User Documentation

Developer Manuals will be provided for System Administrators who wish to set up the code environment. A User Manual will additionally be provided for individuals looking to use this system. All manuals will include step-by-step instructions and will be dispatched in a .pdf format.

2.8 Assumptions and Dependencies

- 2.8.1 AWS is required to connect to Box.com and Video Indexer
- 2.8.2 Box Skills is required to contain an Invocation URL to connect to AWS
- 2.8.3 Video Indexer is required to perform transcription and facial recognition services
- 2.8.4 Code is required to return a .docx transcript in addition to digital video insights
- 2.8.5 Box is required for secure cloud storage

2.9 Apportioning of Requirements

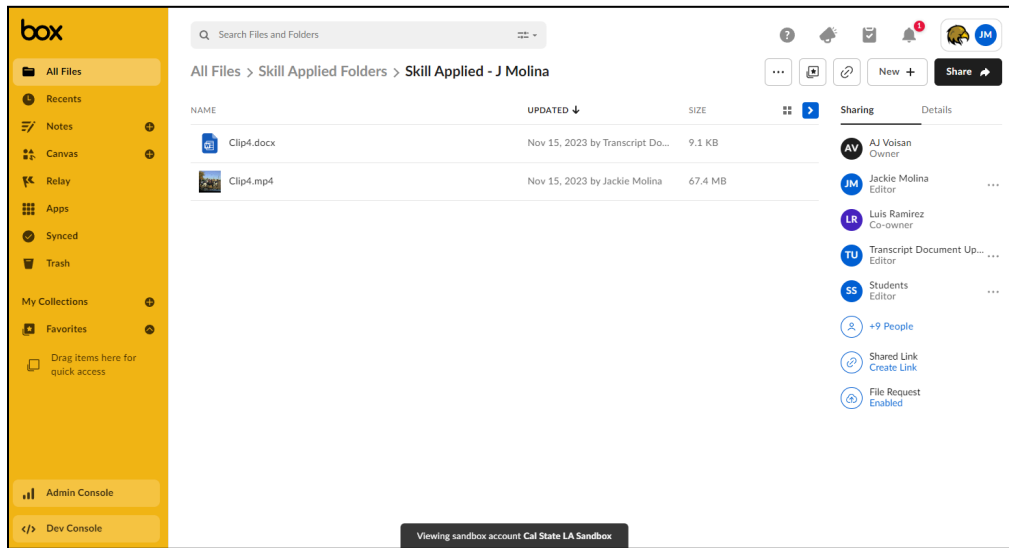
- 2.9.1 Custom Language Model (Language other than English)

3. External Interface Requirements

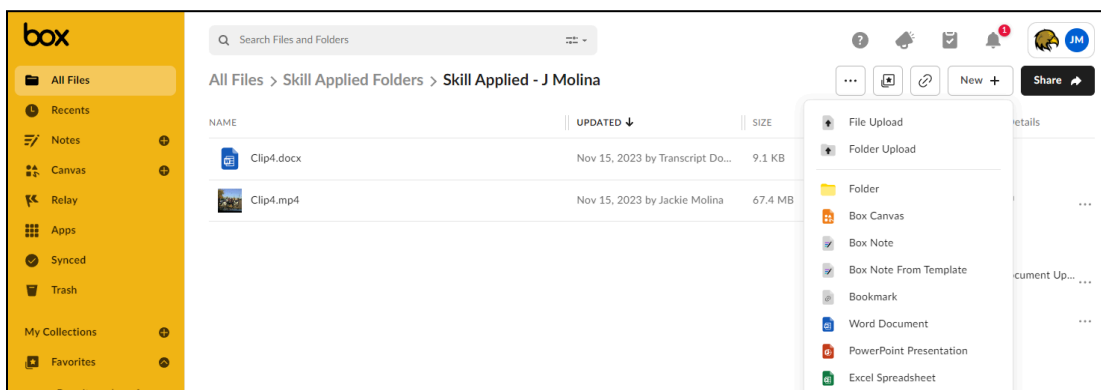
3.1 User Interfaces

A standard user will only be interacting through the Box.com provided interface. The video indexing is performed on cloud services that do not require the input of a standard user. To upload media, select a skill enabled folder, press the new button on the top right of the screen. Once upload is complete, wait for the processing to finish.

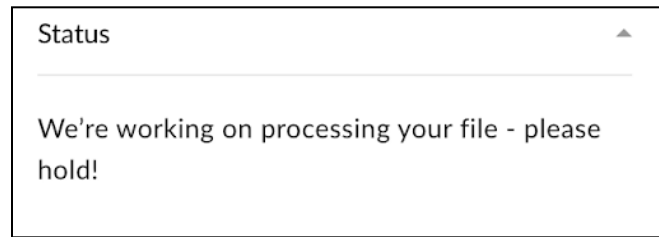
3.1.1 Box.com Folder Interface



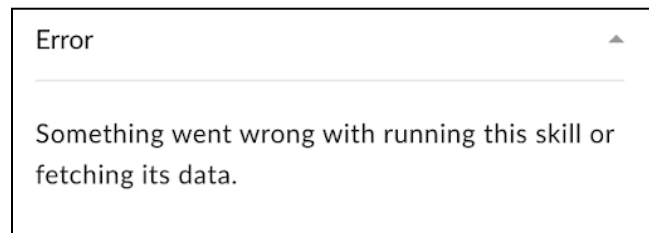
3.1.2 Box.com Upload Media Interface



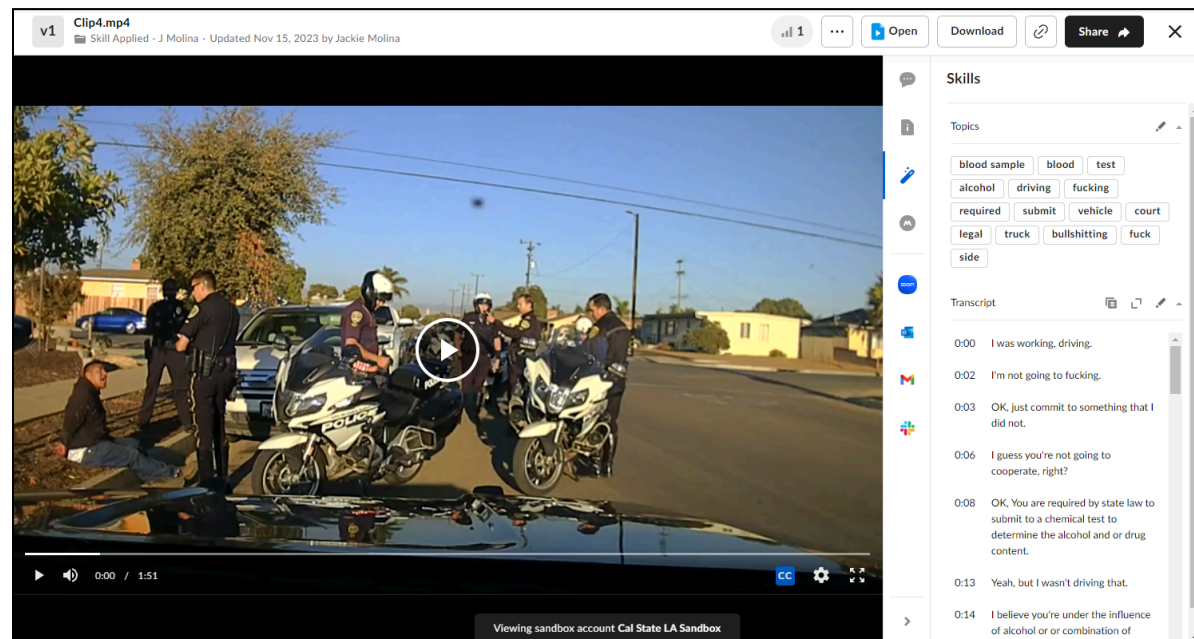
3.1.3 Box.com Processing Card



3.1.4 Box.com Error Card



3.1.5 Box.com Skill Insights



The image shows a screenshot of the Box.com Skill Insights interface. The main video player displays a scene with police officers and motorcycles. The video title is "Clip4.mp4" and it is associated with the skill "Skill Applied - J Molina". The video player includes a play button, a progress bar, and a volume icon. The right sidebar contains a "Skills" section with a list of topics and a transcript. The topics listed are: blood sample, blood, test, alcohol, driving, fucking, required, submit, vehicle, court, legal, truck, bullshitting, fuck, and side. The transcript shows a conversation between two people, with timestamps and text.

v1 Clip4.mp4
Skill Applied - J Molina · Updated Nov 15, 2023 by Jackie Molina

1 ... Open Download Share X

Skills

Topics

blood sample blood test
alcohol driving fucking
required submit vehicle court
legal truck bullshitting fuck
side

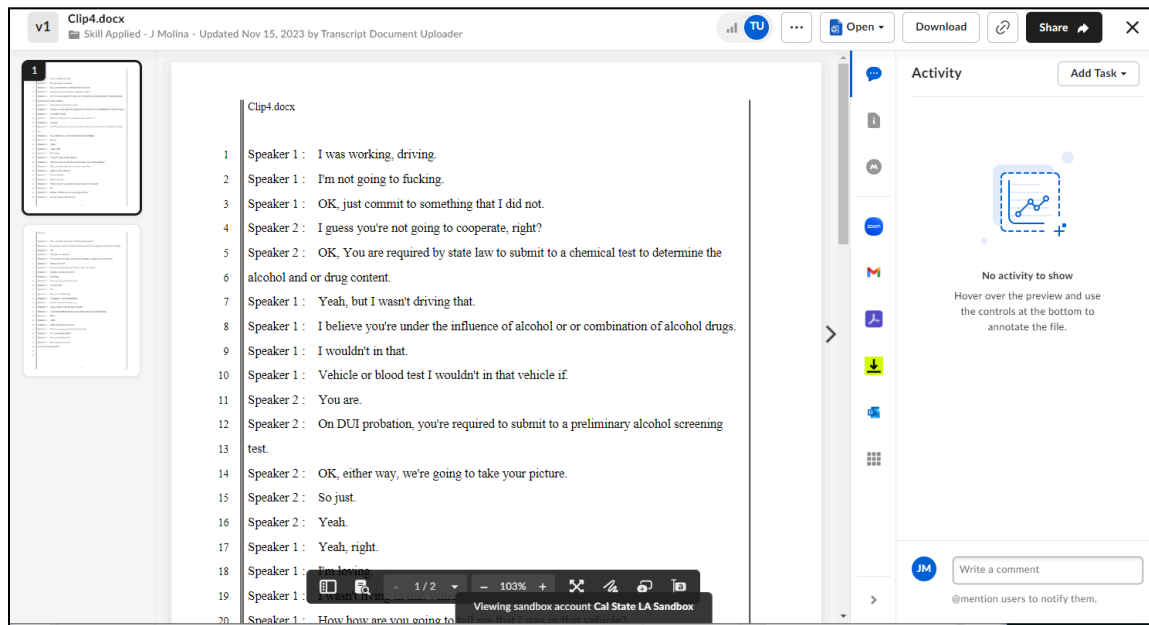
Transcript

0:00 I was working, driving.
0:02 I'm not going to fucking.
0:03 OK, just commit to something that I did not.
0:06 I guess you're not going to cooperate, right?
0:08 OK, You are required by state law to submit to a chemical test to determine the alcohol and or drug content.
0:13 Yeah, but I wasn't driving that.
0:14 I believe you're under the influence of alcohol or or combination of

0:00 / 1:51 CC

Viewing sandbox account Cal State LA Sandbox

3.1.6 Box.com Transcript



3.2 Hardware Interfaces

Not Applicable.

3.3 Software Interfaces

3.3.1 Internet Browser

3.3.1.1 Chrome (latest version)

3.3.1.2 Safari (latest version)

3.3.1.3 Microsoft Edge (latest versions)

3.3.2 Operating System

3.3.2.1 Windows (latest version)

3.3.2.2 macOS (two latest versions)

3.4 Communications Interfaces

3.4.1 Hypertext Transfer Protocol Secure (HTTPS) (GET)

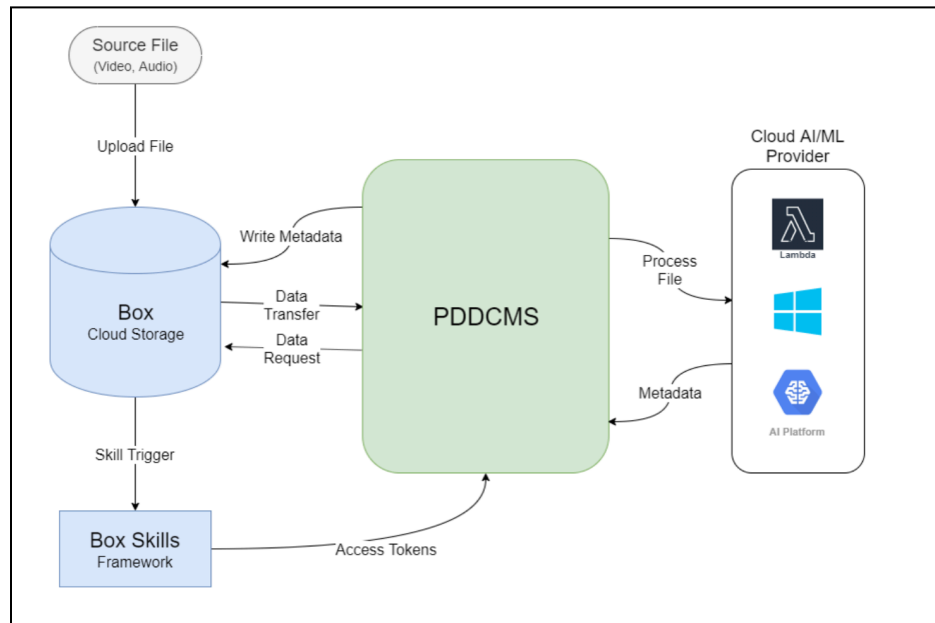
3.4.2 AWS Server Endpoint

3.4.3 Access Tokens (Video Indexer)

3.4.4 Security Keys (Box)

3.4.5 Additional Environment Variables required by the Lambda Function

4. Requirements Specification



4.1 Functional Requirements

4.1.1 Box

- 4.1.1.1 The user shall have valid Box credentials with access to SBPD folders
- 4.1.1.2 The user shall have permission to view a file along with its returned insights
- 4.1.1.3 The user shall have permissions to upload files to SBPD Box folders
- 4.1.1.4 The user shall have permission to remove files from SBPD Box folders
- 4.1.1.5 The user's given folder must be configured with a Box Skills Application

4.1.2 Box Skills (Developer)

- 4.1.2.1 The user shall have administrator credentials to configure Box Skills
- 4.1.2.2 The user shall have developer credentials to create Box Skills
- 4.1.2.3 The user must have created a Box Skills Application to run the program
- 4.1.2.4 The user administrator must apply the created skill to selected folders
- 4.1.2.5 The user must configure the Box Skills to a valid AWS invocation url
- 4.1.2.6 The user must configure the folder to trigger an event upon file upload
- 4.1.2.7 The user must provide access tokens to the invocation URL
- 4.1.2.8 The user must provide security keys to the lambda function

4.1.3 BoxSDK (Developer)

- 4.1.3.1 The user must use BoxSDK to manipulate Box files
- 4.1.3.2 The user must use BoxSDK metadata cards to return to the standard user
- 4.1.3.3 The user must have valid access tokens and security keys

4.1.4 Video Indexer

- 4.1.4.1 The user shall have valid Microsoft Azure credentials
- 4.1.4.2 The user shall configure the credentials in the AWS environment
- 4.1.4.3 Video Indexer shall process uploaded media
- 4.1.4.4 Video Indexer shall return media metadata to AWS
- 4.1.4.5 Video Indexer shall communicate with AWS

4.1.5 Video Indexer Custom Model

- 4.1.5.1 The user shall have the necessary privileges and roles for the Custom Model
- 4.1.5.2 The Custom Model shall provide accurate subtitle and closed caption
- 4.1.5.3 The Custom Model should integrate with Azure Video Indexer API for customizing the language model
- 4.1.5.4 The Custom Model should support custom vocabulary and phrases needed by SBPD
- 4.1.5.5 The languages in the video content shall be supported

4.1.6 AWS Lambda

- 4.1.6.1 The user shall have valid AWS credentials
- 4.1.6.2 The user credentials must be configured within the Serverless Framework
- 4.1.6.3 The Lambda function must use Video Indexer credentials to authenticate
- 4.1.6.4 The Lambda function must use Box credentials to authenticate
- 4.1.6.5 The Lambda function must communicate with VI, Box, and S3.

4.1.7 AWS Secrets Manager

- 4.1.7.1 Secrets Manager shall securely store keys
- 4.1.7.2 Secrets Manager shall have control policies to manage who can access secrets
- 4.1.7.3 Secrets manager will allow users with requirements to access the information
- 4.1.7.4 Secrets manager will secure information within it
- 4.1.7.5 Secrets Manager will deny access to users without credentials

4.1.8 AWS S3

- 4.1.8.1 S3 must store data incoming from Box
- 4.1.8.2 S3 must store data incoming from Video Indexer
- 4.1.8.3 S3 must remove buckets once program has completed its cycle
- 4.1.8.4 S3 must automatically remove remaining buckets after a 30 day period

4.1.9 Transcription Function

- 4.1.9.1 The user shall have a valid JSON File from Microsoft AI Video Indexer
- 4.1.9.2 The program is required to format the JSON File
- 4.1.9.3 The program is required to highlight the confidence marking
- 4.1.9.4 The program is required to produce a transcribe document
- 4.1.9.5 The produced document is required to fill in all 28 lines

4.2 External Interface Requirements

4.2.1 Box User Interface

- 4.2.1.1 Users will interact directly with the provide Box interface
- 4.2.1.2 Users will be able to search files and folders by name
- 4.2.1.3 Users with valid permissions will be able to upload files to selected folders
- 4.2.1.4 Users will be able to interact with returned metadata from processing program

4.2.2 Box Administrator Interface (Developers)

- 4.2.2.1 Developers will be able to create a Box Skill application using the provided Box Developer Console
- 4.2.2.2 Developers will be able to authenticate their Box Skill application using the provided Box Admin Console
- 4.2.2.3 Developers will be able to configure and connect a Box Skill using a valid AWS invocation URL

4.2.3 AWS Interface (Developers)

- 4.2.3.1 Developers will be able to use the AWS Lambda service to create a Lambda function
- 4.2.3.2 Developers will be able to use AWS services to modify and test code
- 4.2.3.3 Developers must authenticate and connect to Video Indexer and Box within the Environment variables
- 4.2.3.4 Developers must use the metadata generated by Video Indexer to return to Box
- 4.2.3.5 Developers must use the access tokens to return metadata to Box
- 4.2.3.6 Developers must ensure the deletion of data under the AWS S3 service
- 4.2.3.7 Developers must authenticate incoming requests using provided security keys

4.2.4 Video Indexer Interface (Developers)

4.2.4.1 Developers will be able to create an account meant to manage and transcribe media uploaded onto Box

4.2.4.2 Developers must use the generated access tokens and ID to connect to the AWS function

4.2.4.3 Developers must ensure the security of the Video Indexer Account

4.3 Logical Database Requirements

Not Applicable.

All storage needs are satisfied through Box, AWS, and Microsoft Azure.

4.4 Design Constraints

4.4.1 All hardware is cloud based and maintained by either Box, AWS, or Microsoft Azure

4.4.2 Stable internet connection is required to upload or view files on Box

4.4.3 AWS Lambda outages will temporarily disable functionality until service returns

4.4.4 Azure outages will temporarily disable functionality until service returns

4.4.5 Computation cost for AWS and Azure is subject to change

4.4.6 Video Indexer has varying effectiveness in non-English languages

4.4.7 Video Indexer has varying effectiveness depending on given media quality

4.4.8 System administrators must have access to comprehensive documentation

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.1.1 Analysis time of media will take ~50% of the total length of the video

5.1.2 Accuracy of the AI model is dependent on clarity of video and audio. Users may see a reduction of accuracy for audio transcription and visual tagging.

5.1.3 The program is required to run smoothly and in a timely fashion; given that the user has a secure internet connection.

5.1.4 The program is required to ensure consistent data transfer when uploading or viewing media from the cloud content system.

5.1.5 The program is required to ensure client confidentiality.

5.1.6 The program is required to successfully upload, transcribe, and return media to users.

5.2 Safety Requirements

5.2.1 Biased AI

Software is heavily dependent on AI and may be biased in ways unintended by the original creators of the program.

5.2.2 Minute Inaccuracies

Software is heavily dependent on AI and may result in certain inaccuracies when tagging people/items or when transcribing speech.

5.3 Security Requirements

Security of external Microsoft Video Indexer and AWS accounts must be secured with strong passwords, ideally two factor authentication, that follow recommended safety standards.

Standard users should be aware of confidentiality and privacy laws surrounding the media they upload onto Box.

5.4 Software Quality Attributes

5.4.1 Usability

The website is interactive and responsive to any computer the user decides to view it on.

5.4.2 Availability

The program is up and functioning 99.99% of the time.

5.4.3 Maintainability

The source code for the program is easy to maintain, as well as easy to modify and update depending on the current needs of the software.

5.4.4 Robustness

The program is able to function well with different web browsers and network connection types.

5.4.5 Portability

The program is able to function on different platforms and operating systems.

5.5 Business Rules

Users are controlled and placed under certain restrictions to ensure the integrity of the system. Standard users should not be able to alter Box Skills, AWS services, or Video Indexer services. Administrators and Developers should have full access to the codebase and aforementioned services.

6. Legal and Ethical Considerations

6.1 Privacy and Security

6.1.1 Privacy and Security are critical aspects for SBPD to provide adequate disclosure for their clients

6.1.2 All laws regarding the handling of case discovery media must be followed

6.1.3 Access to case information should be limited to those with the necessary credentials

6.1.4 Disclosed information regarding clients, including personally identifiable information, should remain private

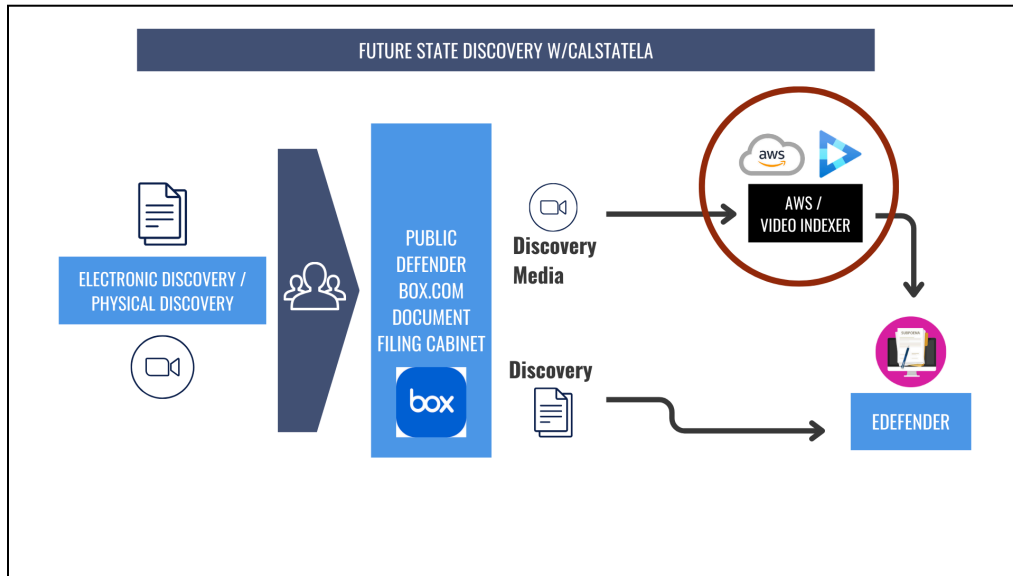
6.1.5 Credentials of all interconnected services must be tightly managed in order to ensure the safety of all information

Appendix A: Glossary

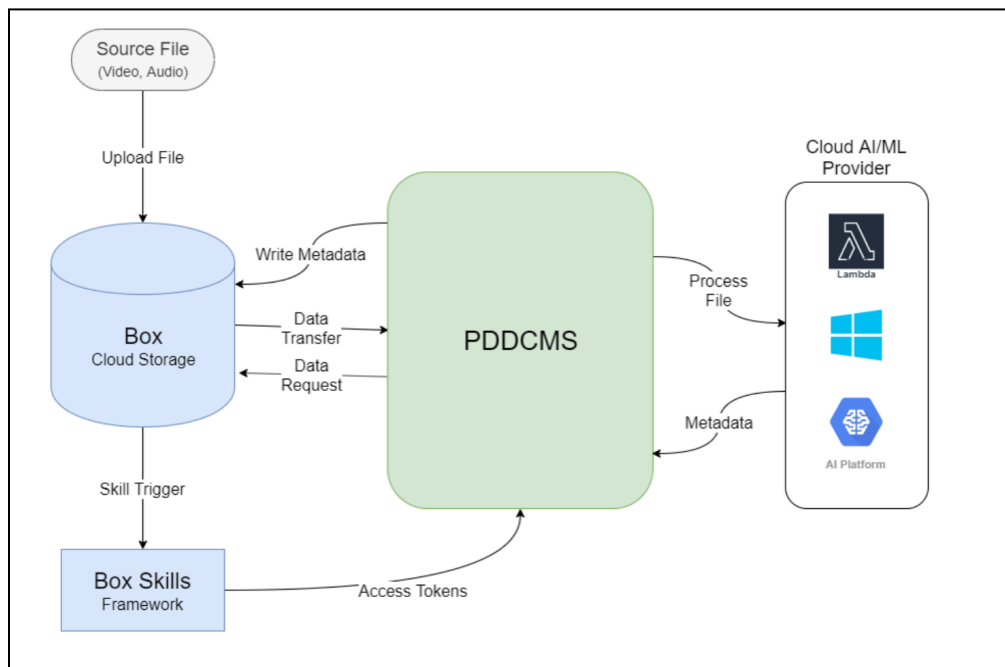
Not Applicable.

Appendix B: Analysis Models

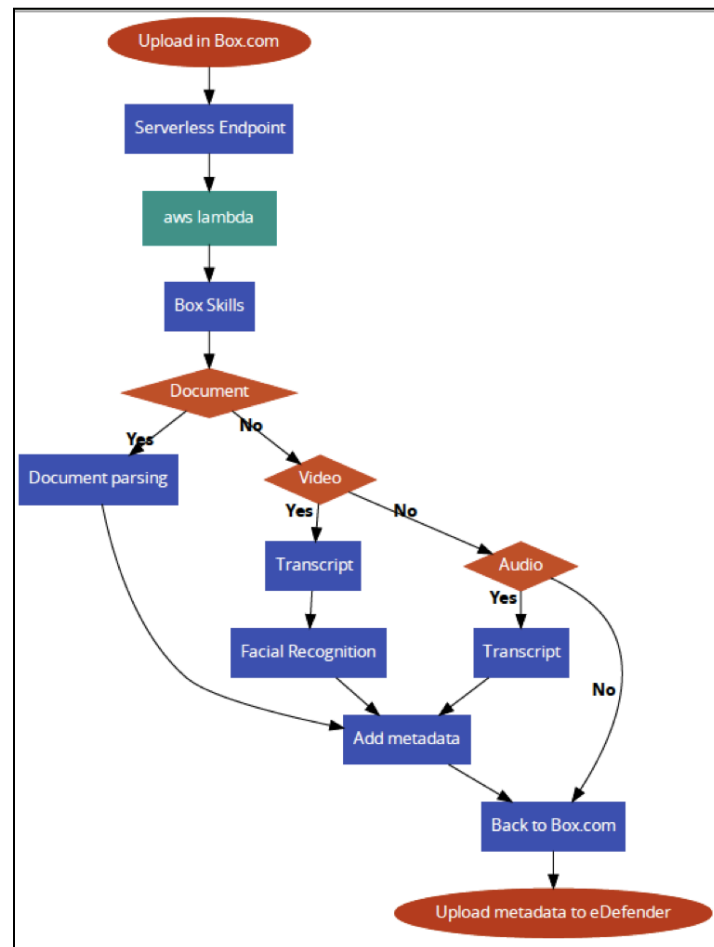
[Figure 1: Discovery Process using Cloud-Based Services]



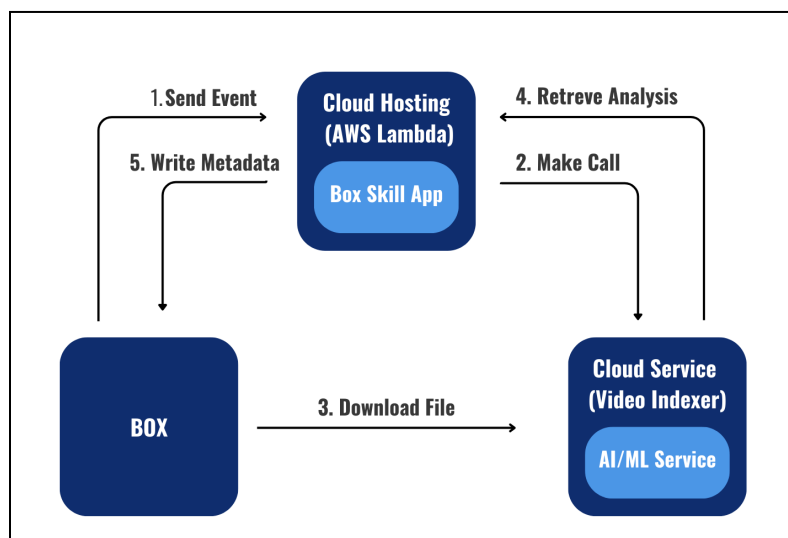
[Figure 2: Detailed View into the Program]



[Figure 3: Flow Diagram]



[Figure 4: Box Skills]



Appendix C: To Be Determined List

Not Applicable.