Senior Design Final Report

Box.com/eDefender Integration

PUBLIC DEFENDER
SANTA BARBARA COUNTY

Version 1.0 - 05/12/2023

Team Members:
Mario Avila
Jimmy Castillo
Joseph Comeaux
Sherina Marquez
Ethan Ngo
Alexis Ponce
Karen Quan
Rosa Robles
Alexander Voison
James Yokley

Faculty Advisor:
Dr. Jung Soo Lim

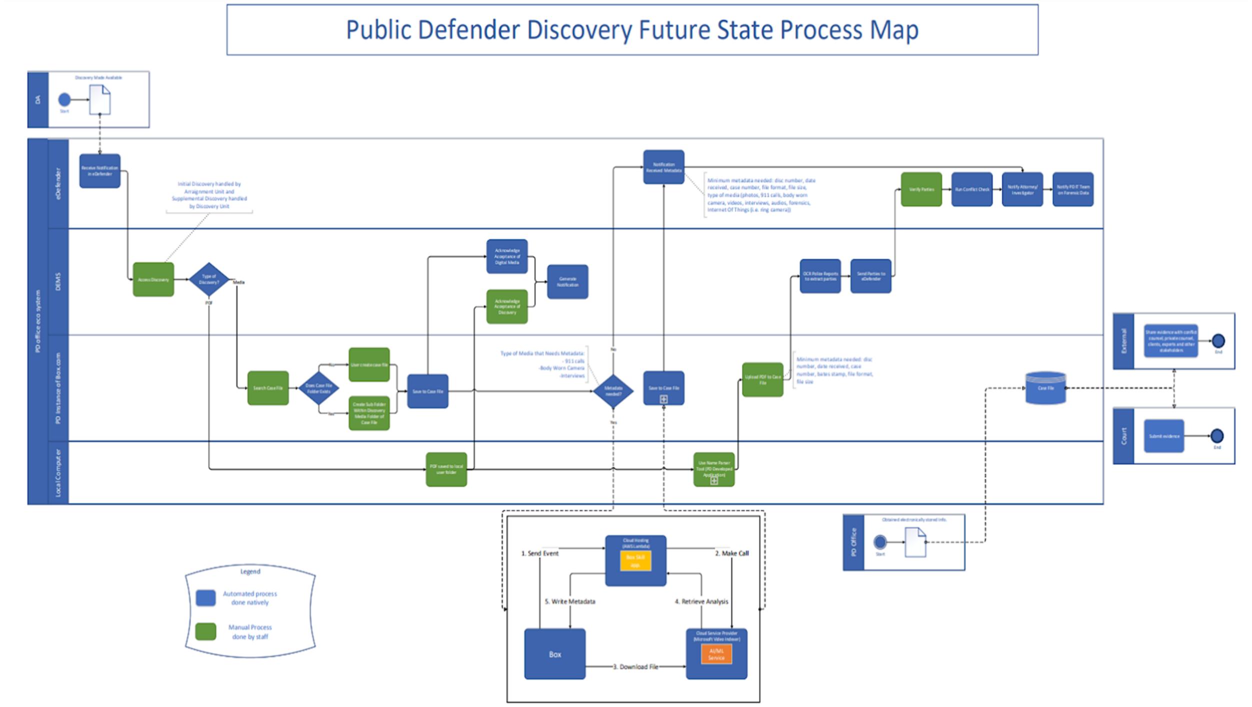
Liaisons:
Deepak Budwani
Bryan Burzon
Sarah Rothschild
Angella Stokke
Luis Ramirez

1.Introduction:

The sponsor for this project is the Santa Barbara Public Defender's Office (SPBD). They serve to protect and defend their client's rights through legal representation. In addition, they manually go through an abundant amount of evidence in various forms of media, such as videos from body-worn cameras, audio from 911 calls, documents, and more. Our task is to integrate eDefender, our sponsor's current file management system, and Box.com which will enable our sponsor's ability to review evidence in an efficient manner.

1.1 Background:

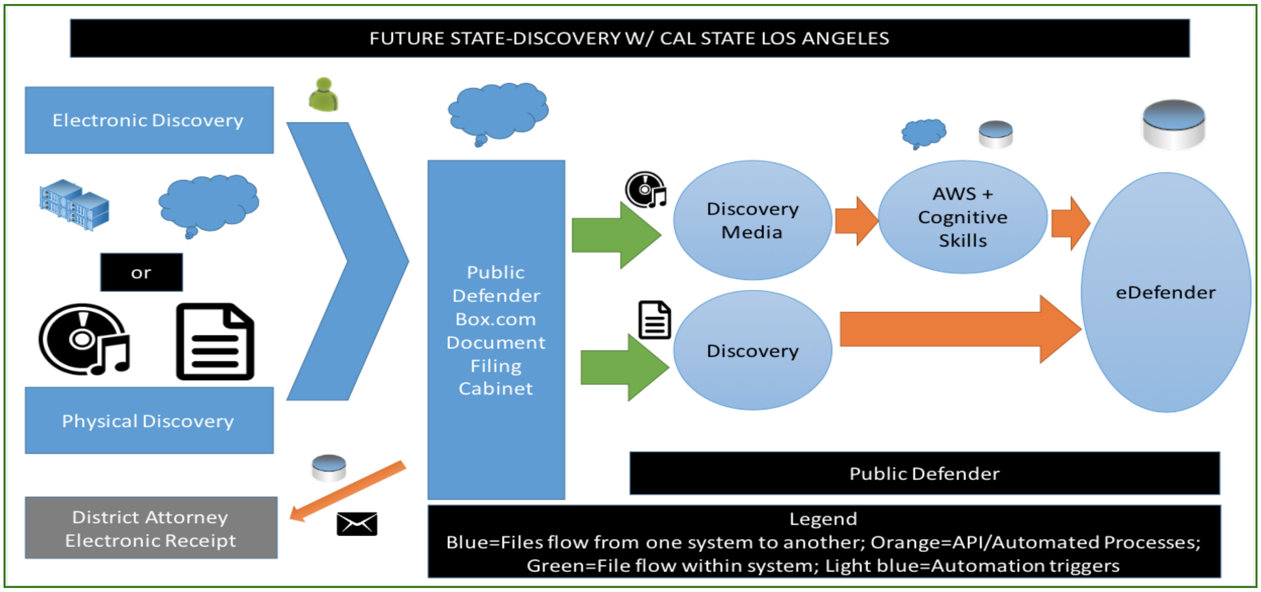
Discovery process details + images



Public Defender Discovery Future State Process Map
DA
Discovery Made Available
Start
PD office eco system
eDefender
DEMS
PD Instance of Box.com
Local Computer
Receive Notification in eDefender
Initial Discovery handled by Arraignment Unit and Supplemental Discovery handled by Discovery Unit
Access Discovery
Type of Discovery?
Media
PDF
Search Case File
Does Case File Folder Exists
No
Yes
User create case file
Create Sub Folder Within Discovery Media Folder of Case File
Save to Case File
Acknowledge Acceptance of Digital Media
Acknowledge Acceptance of Discovery
Generate Notification
PDF saved to local user folder
Type of Media that Needs Metadata:
- 911 calls
-Body Worn Camera
-Interviews
Metadata needed?
No
Yes
Notification Received Metadata
Minimum metadata needed: disc number, date received, case number, file format, file size, type of media (photos, 911 calls, body worn camera, videos, interviews, audios, forensics, Internet Of Things (i.e. ring camera))
Save to Case File
Use Name Parser Tool (PD Developed Application)
Upload PDF to Case File
Minimum metadata needed: disc number, date received, case number, bates stamp, file format, file size
OCR Police Reports to extract parties
Send Parties to eDefender
Verify Parties
Run Conflict Check
Notify Attorney/ Investigator
Notify PD IT Team on Forensic Data
Case File
External
Share evidence with conflict counsel, private counsel, clients, experts and other stakeholders
End
Court
Submit evidence
End
PD Office
Obtained electronically stored info.
Start
1. Send Event
Cloud Hosting (AWS Lambda)
Box Skill app
2. Make Call
5. Write Metadata
4. Retrieve Analysis
Box
3. Download File
Cloud Service Provider (Microsoft Video Indexer)
AI/ML Service
Legend
Automated process done natively
Manual Process done by staff

1.2 Design Objectives:

The main objective is for the user(s), who have access to the Box.com folder, to upload a video or audio file or folder. From there it will contact the Serverless endpoint which invokes AWS Lambda and triggers the Box Skills functions. It will traverse through each skill that we have and will return metadata, such as facial recognition and keywords. In addition to the metadata being returned, it will also return a generated transcript with 28 lines per page.


FUTURE STATE-DISCOVERY W/ CAL STATE LOS ANGELES
Electronic Discovery
or
Physical Discovery
District Attorney Electronic Receipt
Public Defender Box.com Document Filing Cabinet
Discovery Media
Discovery
AWS + Cognitive Skills
eDefender
Public Defender
Legend
Blue=Files flow from one system to another; Orange=API/Automated Processes;
Green=File flow within system; Light blue=Automation triggers

1.3 Design Benefits:

Our Box app helps the user go through video evidence like body cameras and audio evidence like recorded interviews more efficiently. The user may choose to focus on one person that appears in a video file, and our Box app allows the user to jump to where this person appears in the video. The transcript generated and topics also allows the user to search for relevant information without watching the whole video or listening to the whole audio. Additionally, this transcript can be used in the courtroom in which the line numbers on the transcript can be referred to in order to quickly ensure everyone is reading the same quote.

1.4 Achievements:

Over the academic school year we have completed a working application so that once a file is uploaded to a specified Box folder and is finished processing, a side panel will appear with faces of people who appear in the video(if it is a video), keywords from the file, and a transcript of the file. This will allow lawyers and legal representatives to quickly scan through the file and gain insights relevant to their case. A separate word document of the transcript will also be generated in the same folder, with the same name. The separately generated transcript mostly complies with the requirements and format set by our sponsor and will allow them to quickly refer to relevant parts of the transcript in court.

2. Related Programs:

2.1 Existing Solutions:

Other machine learning type solutions like Microsoft Azure and Box do exist, such as Clarifai and Amazon Rekognition. Compared to Microsoft Azure, which provides facial recognition, transcription, and keyword highlights, Clarifai and Amazon Rekognition only offer facial recognition. Additionally, our sponsors made us aware that there are no existing solutions catered exactly for the government, more specifically legal implementations for automating the generation of case metadata and insights.

2.2 Reused Solutions:

The application is developed in Node.js and utilizes the Box Skills SDK and Microsoft Azure Video Indexer API. The SDK allows us access to event triggers on Box.com folders as well as the ability to upload files and metadata through Box Skills cards. The Video Indexer API allows us to make processing requests for the file uploaded and can be configured for different languages and features, such as using facial recognition or creating only a transcript.

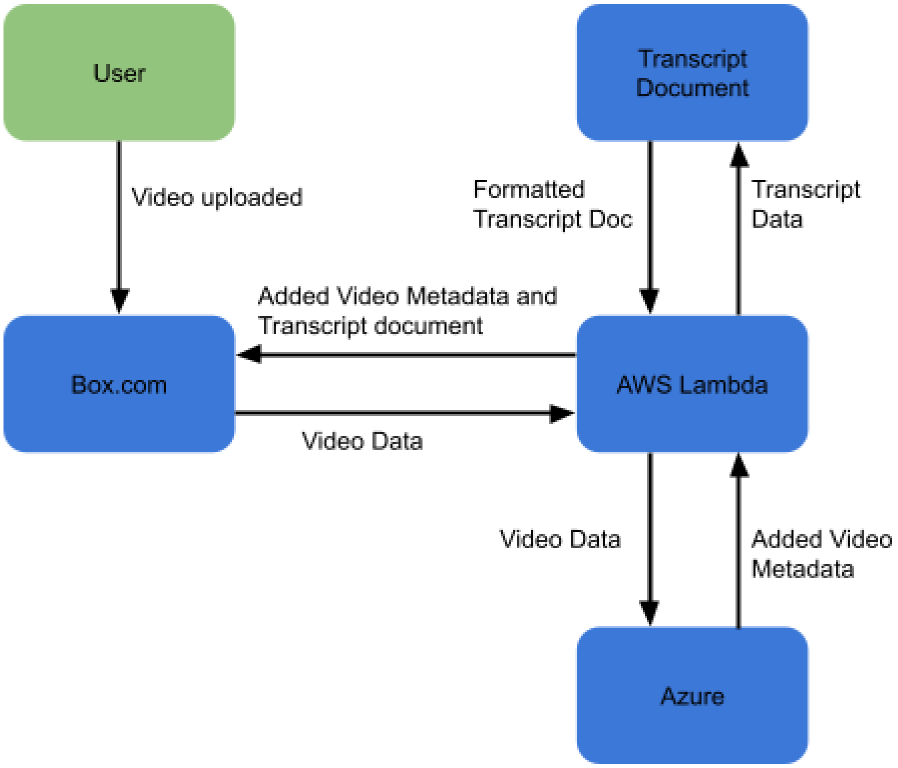
3.System Architecture:

3.1 Overview:

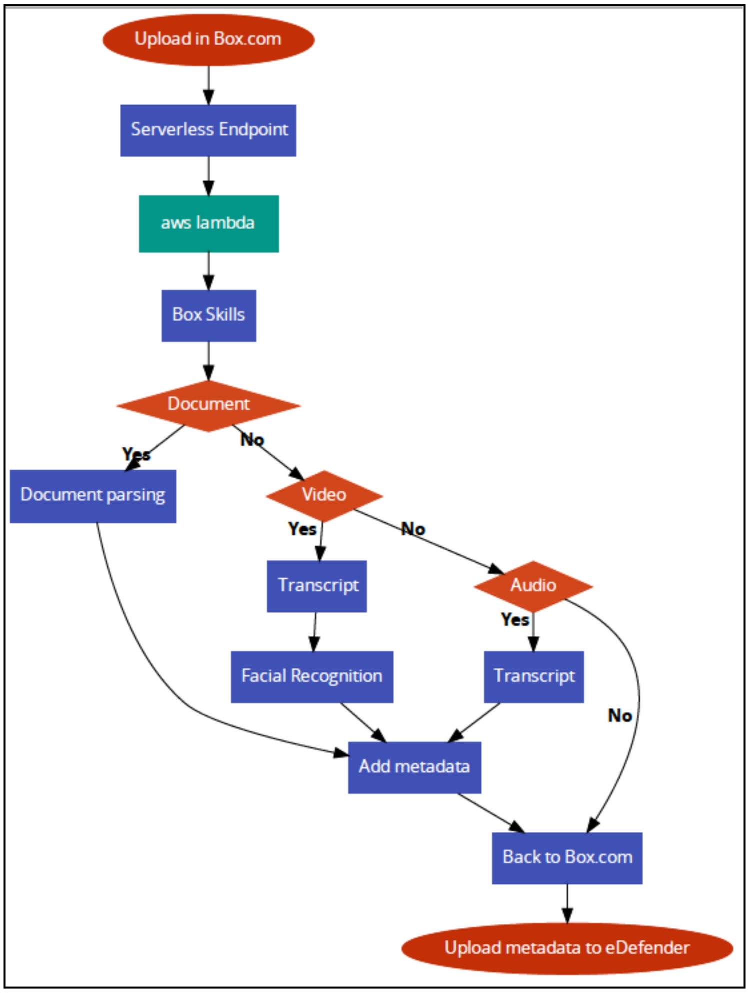
The application that we have built leverages several frameworks and APIs that will allow us to host our code using Serverless and run it using AWS Lambda. From there, a Box skills application is configured to notify the lambda function when new data has been updated. The responses generated by the Node.js code are then sent back to Box.com.

3.2 Data Flow:

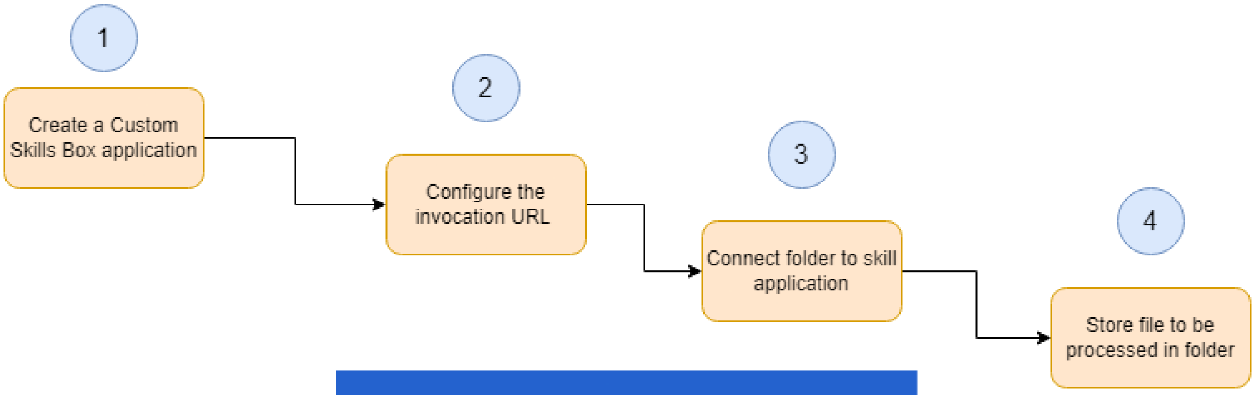
The data flow in our project is as shown in the figure below, starting with a user uploaded video. When the video or document is fully uploaded to Box.com, the site will check its extension type to confirm if the call needs to be triggered and then sends a call to AWS along with the data of the video. The data is now on AWS and is sent to Azure to be scanned by their algorithms. When this is done it is sent back to AWS. AWS will use the transcription data from azure in a new call to create the transcription document in the proper format, as this happens the metadata is sent back to Box.com to be added to the video. Simultaneously AWS is still working on the transcription document, which is then sent back when it is finished. Data on AWS is deleted after the calls are over, and can be altered in AWS settings. If we remember correctly from a previous meeting, according to the LA Public Defender Azure data must be scrubbed from the service manually, this may change in the future if they add an auto delete feature.

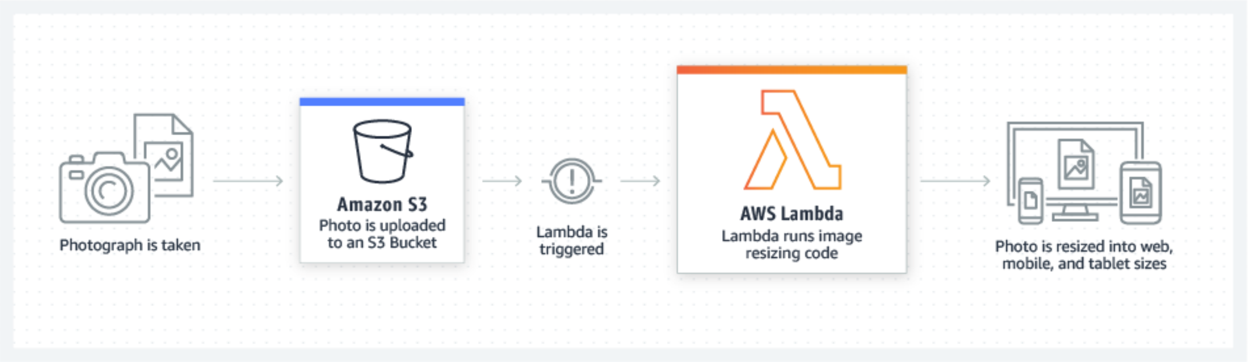

User
Video uploaded
Transcript Document
Formatted Transcript Doc
Transcript Data
Added Video Metadata and Transcript document
Box.com
AWS Lambda
Video Data
Video Data
Added Video Metadata
Azure

3.3 Implementation:



3.4 Related Technologies:


1
Create a Custom Skills Box application
2
Configure the invocation URL
3
Connect folder to skill application
4
Store file to be processed in folder


Photograph is taken
Amazon S3
Photo is uploaded to an S3 Bucket
Lambda is triggered
AWS Lambda
Lambda runs image resizing code
Photo is resized into web, mobile, and tablet sizes

4. Conclusions:

Within these two semesters, our group has accomplished a lot. We feel that this will make a net positive impact to the workflow and time spent on cases for SBPD. With that said, there are a number of suggestions for improvements and expansions that can be done by future groups that we would have loved to implement, including advice for maintaining a good working pace.

4.1 Struggles:

The biggest issue during our journey was debugging, sometimes it was extremely difficult to pinpoint where the error was. When dealing with so many different frameworks and APIs, it became difficult to find where errors were occurring. Versions of technologies used, like node, should be noted as previous versions cause the application to work differently. There were some changes in code from the Box.com side that caused our own code to break which we eventually resolved when talking to a Box.com representative. Additionally, we had to spend some time correcting formatting for the generated transcript files and figuring out how to generate and save the transcript itself. Overall, becoming familiar with new technologies and how to make them work together was something we had to continually work at.

4.2 Results:

Once a file is uploaded to Box, facial recognition (if video), keyword topics, and a generated transcript will be attached to the corresponding file. A separate formatted transcript word file with the same name will be added to the same folder as well. This Word file is crucial for SBPD in that they often use these documents in court to refer to different parts of cases. The facial recognition allows users to jump to wherever the face appears in the video, without going through the whole video. Over the course of two semesters, we didn’t have enough time to look past implementing these features for anything other than English, but we are aware that this will be an important advancement in the future.

In its current state, only files in English will be processed and transcripts that are generated are not 100% accurate. Because the transcriptions are created by Microsoft Azure, there will have to be a conversation with them to determine how to best cater the natural language model they use to SBPD’s use case. Another thing to note is the cost to use Video Indexer is completely related to a video’s length, so a cost benefit analysis will have to be run by SBPD to determine when our application will be more beneficial to them than hiring a transcriber, typically this would be the case when speed is crucial.

4.3 Future:

Future groups that continue to work on this project may be interested in making a Box app or skill that processes files in foreign languages, especially Spanish or mixed English and Spanish. There may also be room for improvement regarding the accuracy of the transcription. It’s also possible the Box app or another future solution may be fine-tuned to recognize professional jargon. Future groups may also want to look into fixing the line numbering when a separate transcript is generated as extra empty lines may be generated at the end, leading to extra pages. We also strongly recommend asking for help or advice from the industry personnel SBPD has connections with; Only if the majority of the team are unable to get past an error, make sure to try all other solutions first, documentation is key and take notes on solutions and share with the group.

5. References:

- Serverless Framework documentation - https://www.serverless.com/framework/docs/providers/aws/cli-reference/deploy
- Box Skills documentation - http://developer.box.com/guides/applications/custom-skills/setup/
- Box.com documentation - http://support.box.com/hc/en-us/articles/360043696394-Create-New-Files-And-Folders
- AWS Lambda documentation - https://docs.aws.amazon.com/lambda/latest/dg/welcome.html
- Microsoft Azure Video Indexer documentation - https://vi.microsoft.com/en-us