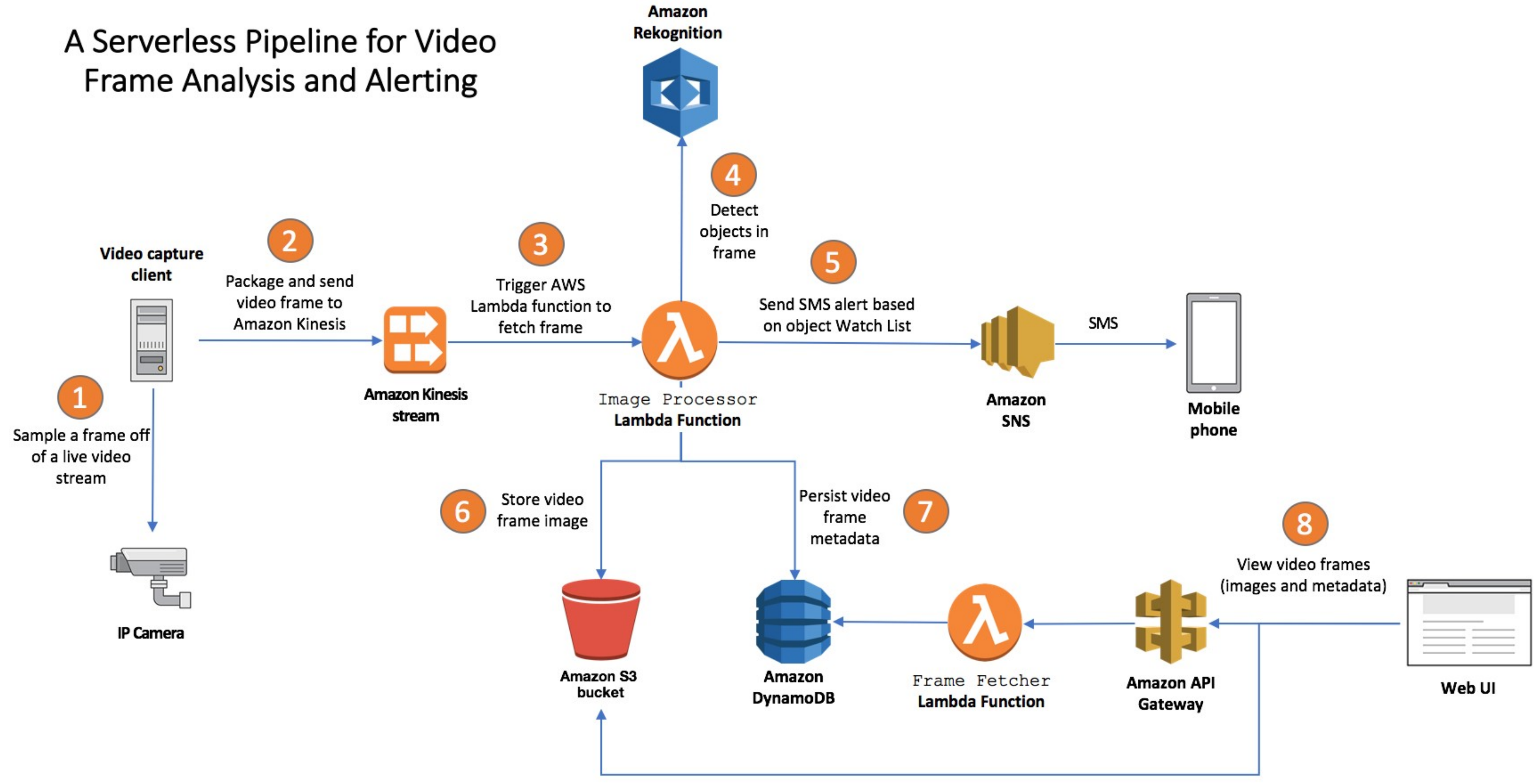
Near wild life

How we built it

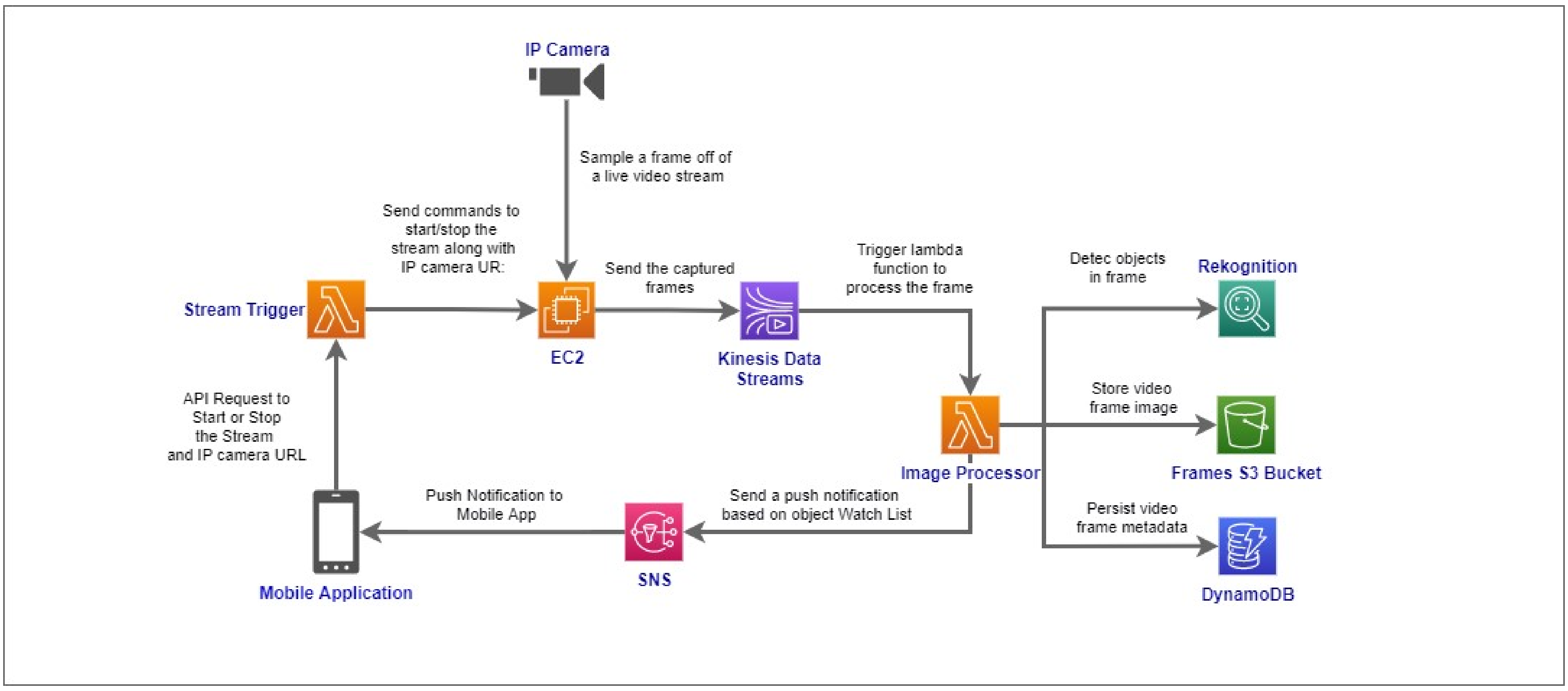
Since this was our first real exposure for all of us to AWS services. We started by searching for articles or projects that will help us to start ahead. And this led us to this project in AWS-Samples account.

This GitHub project was our perfect head start. It mainly streams videos from IP cameras to AWS, recognize frames by Rekognition according to user-configured labeling list, sends alerts.

Here below the GitHub project Flow Diagram:

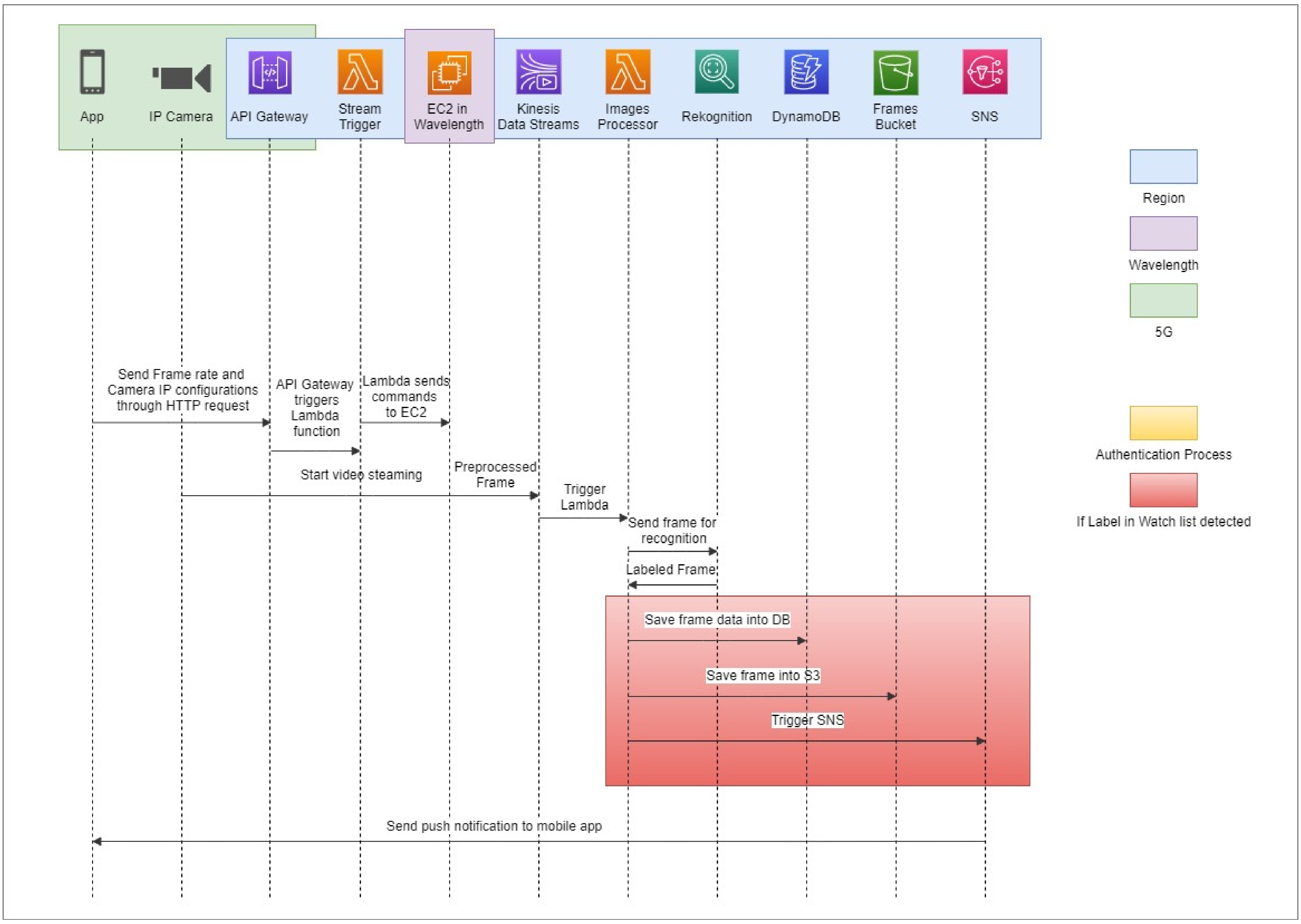

A Serverless Pipeline for Video Frame Analysis and Alerting
Amazon Rekognition
Video capture client
1
Sample a frame off of a live video stream
IP Camera
2
Package and send video frame to Amazon Kinesis
Amazon Kinesis stream
3
Trigger AWS Lambda function to fetch frame
4
Detect objects in frame
Image Processor Lambda Function
5
Send SMS alert based on object Watch List
Amazon SNS
SMS
Mobile phone
6
Store video frame image
Amazon S3 bucket
7
Persist video frame metadata
Amazon DynamoDB
Frame Fetcher Lambda Function
Amazon API Gateway
8
View video frames (images and metadata)
Web UI

But of course, it required modifications to fit our project idea. As seen in the below flow diagram.


IP Camera
Sample a frame off of a live video stream
Send commands to start/stop the stream along with IP camera UR:
Stream Trigger
EC2
Send the captured frames
Kinesis Data Streams
Trigger lambda function to process the frame
Detec objects in frame
Rekognition
Store video frame image
Frames S3 Bucket
Persist video frame metadata
DynamoDB
Image Processor
API Request to Start or Stop the Stream and IP camera URL
Push Notification to Mobile App
Send a push notification based on object Watch List
SNS
Mobile Application

We have changed/added the below:

1. The system is triggered from the mobile application; instead of being manually from local device.
2. Hosted the video capture client on EC2 instance that get triggered by a lambda function.
3. Added support for RTSP instead of HTTP/S streams.
4. A push notification is sent to the mobile application. Instead of checking for updates regularly from a web UI.



Challenges we ran into

- Wavelength connectivity while we're outside US
- Building the mobile application, as we had no experience in app development
- Deploying the video capture client to EC2 instance, and connect it with lambda function

Accomplishments that we're proud of

- Building our first architecture in AWS
- Building a functioning mobile application that gets the job done

What we learned

- There are different protocols for video streaming
- Flutter for App development
- Push Notification has different providers like FCM and APNs. It has large number of parameters that control the behavior of when and where it is being received.
- Amazon Rekognition has a large set of label types that almost cover every aspect.
- SNS can send to AWS services

What's next for Near wild

- Productionize it by adding accounts and authentication support
- Alert customers based on their geo-location (ex: bear detected near-by)
- Alerting customers by sending sound alerts generated by Amazon Alex.
- Support a wide range of cameras.
- Add support to stream the camera within the App